Proiect SGBD

Lant de Magazine de Instrumente Muzicale

Radoi Dragos Cosmin

Grupa 244

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

Proiectul meu va prezenta modelul unei baze de date care reprezinta un lant de magazine deinstrumente muzicale. Magazinele vor avea mai multe marci de instrumente muzicale(furnizate decompaniile respective) sau echipamente pentru acestea si un Id care reprezinta locatia magazinului. Ca angajati, magazinele vor avea fiecare unul sau mai multi vanzatori, un tehnician care se ocupa cu reparatul intrumentelor aduse de clienti si un manager. Angajatii vor avea fiecare un cod si un email de munca. Clientii, odata ce isi fac un cont pe site-ul magazinului, pot crea comenzi pentru orice produs (daca nu este in stoc, livrarea o sa dureze mai mult). Livrarea va prezenta instrumentul, marca, durata de livrare si locatia din care vine.

Locatie – Magazin(One to many): Prezenta unui magazin intr-o locatie

Magazin – Angajati(One to many): Prezenta mai multor angajati magazin

Magazin – Clienti(Many to many): Prezenta mai multor clienti in mai multe magazine

Magazin – Instrumente(Many to many): Prezenta mai multor intrumente in mai multe

magazine

Instrumente – Brand(Many to one): Fiecare intrument poate o marca diferita fata de alt

intrument, dar mai multe intrumente pot avea acelasi brand

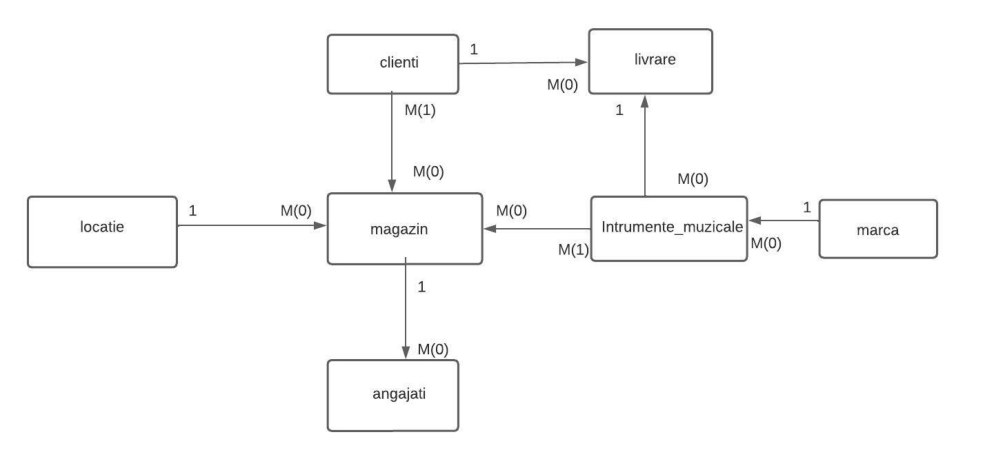
Intrumente – Livrare(Many to one): Mai multe intrumente se pot incadra intr-o singura

livrare

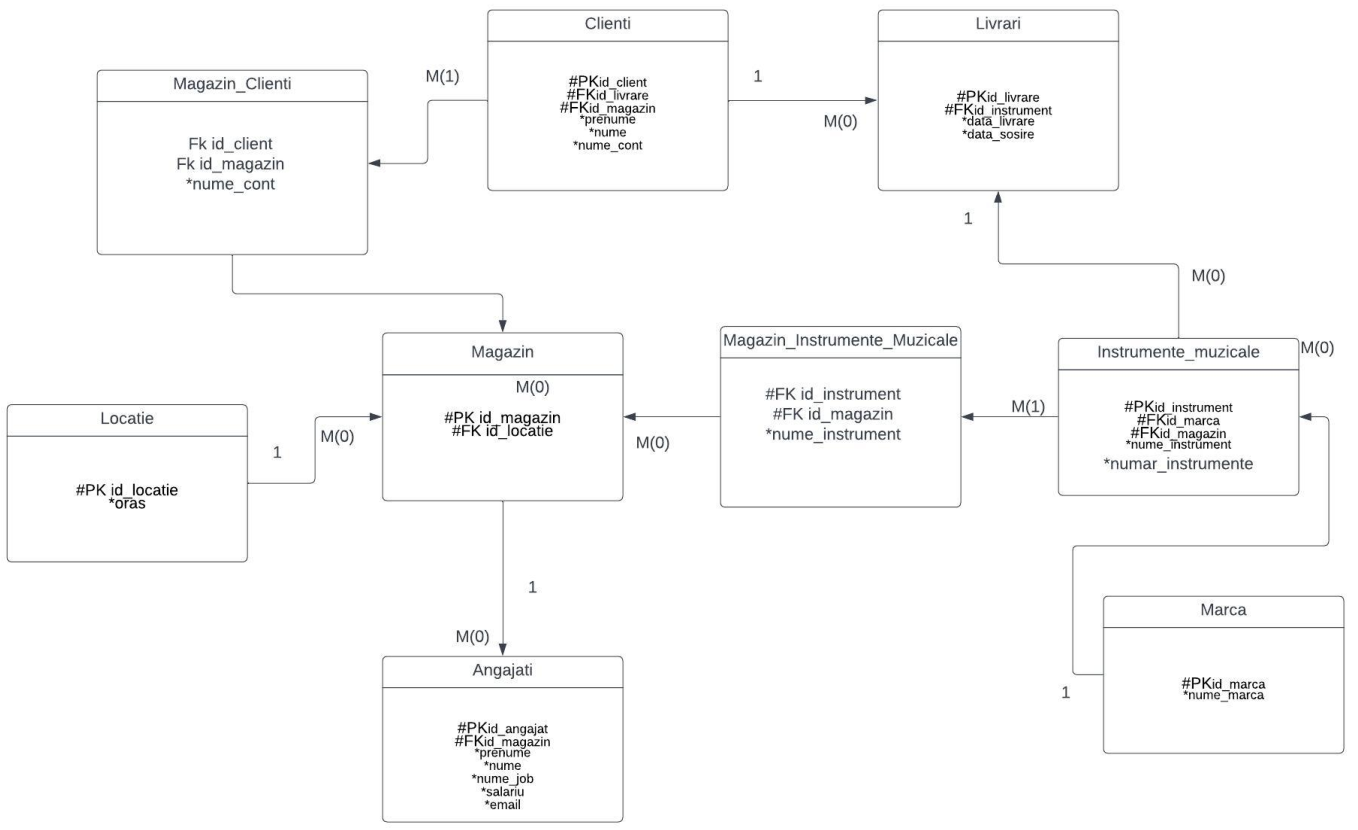
Clienti – Livrare(One to many): Un client poate avea mai multe livrari

2)Realizați diagrama entitate-relație (ERD): entitățile, relațiile și atributele trebuie definite în limba

română (vezi curs SGBD / model de diagrama ERD; nu se va accepta alt format).



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.



4)Implementați în Oracle diagrama conceptuală realizată: definiți toate tabelele, definind toate

constrângerile de integritate necesare (chei primare, cheile externe etc).

--locatie

CREATE TABLE locatie(id\_locatie NUMBER(4), oras VARCHAR(20));

ALTER TABLE locatie

ADD CONSTRAINT PK\_locatie PRIMARY KEY(id\_locatie);

--magazin

CREATE TABLE magazin(id\_magazin NUMBER(4), id\_locatie NUMBER(4));

ALTER TABLE magazin

ADD CONSTRAINT PK\_magazin PRIMARY KEY(id\_magazin)

ADD CONSTRAINT FK\_locatie2 FOREIGN KEY (id\_locatie) REFERENCES locatie(id\_locatie);

--angajati

CREATE TABLE angajati(id\_angajat NUMBER(4), id\_magazin NUMBER(4), prenume VARCHAR2(20),

nume VARCHAR2(20) NOT NULL, nume\_job VARCHAR2(15), salariu NUMBER(8, 2) NOT NULL, email CHAR(15));

ALTER TABLE angajati

ADD CONSTRAINT PK\_angajati PRIMARY KEY(id\_angajat)

ADD CONSTRAINT FK\_magazin2 FOREIGN KEY (id\_magazin) REFERENCES magazin(id\_magazin);

--marca

CREATE TABLE marca(id\_marca NUMBER(4), nume\_marca VARCHAR2(15));

ALTER TABLE marca

ADD CONSTRAINT PK\_marca PRIMARY KEY(id\_marca);

--instrumente muzicale

CREATE TABLE instrumente\_muzicale(id\_instrument NUMBER(4), id\_marca NUMBER(4), id\_magazin NUMBER(4), nume\_instrument VARCHAR2(15));

ALTER TABLE instrumente\_muzicale

ADD CONSTRAINT PK\_instrument PRIMARY KEY(id\_instrument)

ADD CONSTRAINT FK\_magazin3 FOREIGN KEY (id\_magazin) REFERENCES magazin(id\_magazin)

ADD CONSTRAINT FK\_marca2 FOREIGN KEY (id\_marca) REFERENCES marca(id\_marca);

--livrare

CREATE TABLE livrare(id\_livrare NUMBER(4), id\_instrument NUMBER(4), data\_livrare DATE, data\_sosire DATE);

ALTER TABLE livrare

ADD CONSTRAINT PK\_livrare PRIMARY KEY(id\_livrare)

ADD CONSTRAINT FK\_instrument2 FOREIGN KEY(id\_instrument) REFERENCES instrumente\_muzicale(id\_instrument);

--clienti

CREATE TABLE clienti(id\_client NUMBER(4), id\_livrare NUMBER(4), id\_magazin NUMBER(4),

prenume VARCHAR2(20), nume VARCHAR2(20), nume\_cont CHAR(15));

ALTER TABLE clienti

ADD CONSTRAINT PK\_client PRIMARY KEY(id\_client)

ADD CONSTRAINT FK\_livrare2 FOREIGN KEY(id\_livrare) REFERENCES livrare(id\_livrare)

ADD CONSTRAINT FK\_magazin4 FOREIGN KEY(id\_magazin) REFERENCES magazin(id\_magazin);

--magazin-clienti

CREATE TABLE magazin\_clienti(id\_client NUMBER(4), id\_magazin NUMBER(4), nume\_cont

CHAR(15));

ALTER TABLE magazin\_clienti

ADD CONSTRAINT FK\_ID\_Magazin\_Clienti FOREIGN KEY(id\_client) REFERENCES

clienti(id\_client)

ADD CONSTRAINT FK\_ID\_Clienti\_Magazin FOREIGN KEY(id\_magazin) REFERENCES

magazin(id\_magazin);

--magazin-instrumente-muzicale

CREATE TABLE magazin\_instrumente\_muzicale(id\_instrument NUMBER(4), id\_magazin

NUMBER(4), nume\_instrument VARCHAR(15));

ALTER TABLE magazin\_instrumente\_muzicale

ADD CONSTRAINT FK\_ID\_Magazin\_Instrumente FOREIGN KEY(id\_magazin) REFERENCES

magazin(id\_magazin)

ADD CONSTRAINT FK\_ID\_Instrumente\_Magazin FOREIGN KEY(id\_instrument)

REFERENCES instrumente\_muzicale(id\_instrument);

5)Adăugați informații coerente în tabelele create (minim 5 înregistrări pentru fiecare entitate

independentă; minim 10 înregistrări pentru tabela asociativă).

CREATE TABLE locatie(id\_locatie NUMBER(4), oras VARCHAR(20));

ALTER TABLE locatie

ADD CONSTRAINT PK\_locatie PRIMARY KEY(id\_locatie);

INSERT INTO locatie (id\_locatie, oras)

VALUES(1, 'New York');

INSERT INTO locatie (id\_locatie, oras)

VALUES(2, 'London');

INSERT INTO locatie (id\_locatie, oras)

VALUES(3, 'Paris');

INSERT INTO locatie (id\_locatie, oras)

VALUES(4, 'Tokyo');

INSERT INTO locatie (id\_locatie, oras)

VALUES(5, 'Sydney');

INSERT INTO locatie (id\_locatie, oras)

VALUES(6, 'Rome');

INSERT INTO locatie (id\_locatie, oras)

VALUES(7, 'Barcelona');

INSERT INTO locatie (id\_locatie, oras)

VALUES(8, 'Berlin');

INSERT INTO locatie (id\_locatie, oras)

VALUES(9, 'Mumbai');

INSERT INTO locatie (id\_locatie, oras)

VALUES(10, 'Cairo');

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

CREATE TABLE magazin(id\_magazin NUMBER(4), id\_locatie NUMBER(4));

ALTER TABLE magazin

ADD CONSTRAINT PK\_magazin PRIMARY KEY(id\_magazin)

ADD CONSTRAINT FK\_locatie2 FOREIGN KEY (id\_locatie) REFERENCES locatie(id\_locatie);

INSERT INTO magazin (id\_magazin, id\_locatie)

VALUES(101, 1);

INSERT INTO magazin (id\_magazin, id\_locatie)

VALUES(102, 2);

INSERT INTO magazin (id\_magazin, id\_locatie)

VALUES(103, 3);

INSERT INTO magazin (id\_magazin, id\_locatie)

VALUES(104, 4);

INSERT INTO magazin (id\_magazin, id\_locatie)

VALUES(105, 5);

INSERT INTO magazin (id\_magazin, id\_locatie)

VALUES(106, 6);

INSERT INTO magazin (id\_magazin, id\_locatie)

VALUES(107, 7);

INSERT INTO magazin (id\_magazin, id\_locatie)

VALUES(108, 8);

INSERT INTO magazin (id\_magazin, id\_locatie)

VALUES(109, 9);

INSERT INTO magazin (id\_magazin, id\_locatie)

VALUES(110, 10);

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

CREATE TABLE angajati(id\_angajat NUMBER(4), id\_magazin NUMBER(4), prenume VARCHAR2(20),

nume VARCHAR2(20) NOT NULL, nume\_job VARCHAR2(15), salariu NUMBER(8, 2) NOT NULL, email CHAR(15));

ALTER TABLE angajati

ADD CONSTRAINT PK\_angajati PRIMARY KEY(id\_angajat)

ADD CONSTRAINT FK\_magazin2 FOREIGN KEY (id\_magazin) REFERENCES magazin(id\_magazin);

INSERT INTO angajati (id\_angajat, id\_magazin, prenume, nume, nume\_job, salariu, email)

VALUES(1001, 101, 'John', 'Smith', 'manager', 50000, 'john@gmail.com');

INSERT INTO angajati (id\_angajat, id\_magazin, prenume, nume, nume\_job, salariu, email)

VALUES(1002, 102, 'Emily', 'Johnson', 'tehnician', 40000, 'emi@gmail.com');

INSERT INTO angajati (id\_angajat, id\_magazin, prenume, nume, nume\_job, salariu, email)

VALUES(1003, 103, 'Sarah', 'Brown', 'manager', 50000, 'sarah@gmail.com');

INSERT INTO angajati (id\_angajat, id\_magazin, prenume, nume, nume\_job, salariu, email)

VALUES(1004, 104, 'Michael', 'Williams', 'vanzator', 30000, 'mike@gmail.com');

INSERT INTO angajati (id\_angajat, id\_magazin, prenume, nume, nume\_job, salariu, email)

VALUES(1005, 105, 'David', 'Davis', 'tehnician', 40000, 'david@gmail.com');

INSERT INTO angajati (id\_angajat, id\_magazin, prenume, nume, nume\_job, salariu, email)

VALUES(1006, 106, 'Jessica', 'Miller', 'vanzator', 30000, 'mill@gmail.com');

INSERT INTO angajati (id\_angajat, id\_magazin, prenume, nume, nume\_job, salariu, email)

VALUES(1007, 107, 'Matthew', 'Wilson', 'manager', 50000, 'matt@gmail.com');

INSERT INTO angajati (id\_angajat, id\_magazin, prenume, nume, nume\_job, salariu, email)

VALUES(1008, 108, 'Olivia', 'Anderson', 'tehnician', 40000, 'olie@gmail.com');

INSERT INTO angajati (id\_angajat, id\_magazin, prenume, nume, nume\_job, salariu, email)

VALUES(1009, 109, 'Ethan', 'Taylor', 'vanzator', 30000, 'ethan@gmail.com');

INSERT INTO angajati (id\_angajat, id\_magazin, prenume, nume, nume\_job, salariu, email)

VALUES(1010, 110, 'Ava', 'Martinez', 'manager', 50000, 'ava@gmail.com');

INSERT INTO angajati (id\_angajat, id\_magazin, prenume, nume, nume\_job, salariu, email)

VALUES(1011, 101, 'Horia', 'Mihai', 'tehnician', 40000, 'horia@gmail.com');

INSERT INTO angajati (id\_angajat, id\_magazin, prenume, nume, nume\_job, salariu, email)

VALUES(1012, 101, 'Nick', 'Hincu', 'vanzator', 30000, 'Nick@gmail.com');

INSERT INTO angajati (id\_angajat, id\_magazin, prenume, nume, nume\_job, salariu, email)

VALUES(1013, 101, 'Cristi', 'Andrei', 'tehnician', 40000, 'Cris@gmail.com');

INSERT INTO angajati (id\_angajat, id\_magazin, prenume, nume, nume\_job, salariu, email)

VALUES(1014, 107, 'Andrei', 'Cherciu', 'manager', 50000, 'Andii@gmail.com');

INSERT INTO angajati (id\_angajat, id\_magazin, prenume, nume, nume\_job, salariu, email)

VALUES(1015, 107, 'Vlad', 'Tudose', 'vanzator', 30000, 'Vlad@gmail.com');

INSERT INTO angajati (id\_angajat, id\_magazin, prenume, nume, nume\_job, salariu, email)

VALUES(1016, 107, 'Alin', 'Iosif', 'tehnician', 40000, 'Alin@gmail.com');

INSERT INTO angajati (id\_angajat, id\_magazin, prenume, nume, nume\_job, salariu, email)

VALUES(1017, 109, 'Alex', 'Iosif', 'magager', 50000, 'Alex@gmail.com');

INSERT INTO angajati (id\_angajat, id\_magazin, prenume, nume, nume\_job, salariu, email)

VALUES(1018, 110, 'David', 'Ranaciu', 'vanzator', 30000, 'David@gmail.com');

INSERT INTO angajati (id\_angajat, id\_magazin, prenume, nume, nume\_job, salariu, email)

VALUES(1019, 103, 'Robert', 'Trifu', 'tehnician', 40000, 'Rob@gmail.com');

INSERT INTO angajati (id\_angajat, id\_magazin, prenume, nume, nume\_job, salariu, email)

VALUES(1020, 103, 'Filip', 'Dima', 'vanzator', 30000, 'Filip@gmail.com');

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

CREATE TABLE marca(id\_marca NUMBER(4), nume\_marca VARCHAR2(15));

ALTER TABLE marca

ADD CONSTRAINT PK\_marca PRIMARY KEY(id\_marca);

INSERT INTO marca(id\_marca, nume\_marca)

VALUES(201, 'Gibson');

INSERT INTO marca(id\_marca, nume\_marca)

VALUES(202, 'Fender');

INSERT INTO marca(id\_marca, nume\_marca)

VALUES(203, 'Pearl');

INSERT INTO marca(id\_marca, nume\_marca)

VALUES(204, 'Yamaha');

INSERT INTO marca(id\_marca, nume\_marca)

VALUES(205, 'Stradivarius');

INSERT INTO marca(id\_marca, nume\_marca)

VALUES(206, 'Selmer');

INSERT INTO marca(id\_marca, nume\_marca)

VALUES(207, 'Bach');

INSERT INTO marca(id\_marca, nume\_marca)

VALUES(208, 'Muramatsu');

INSERT INTO marca(id\_marca, nume\_marca)

VALUES(209, 'Ibanez');

INSERT INTO marca(id\_marca, nume\_marca)

VALUES(210, 'Roland');

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

CREATE TABLE instrumente\_muzicale(id\_instrument NUMBER(4), id\_marca NUMBER(4), id\_magazin NUMBER(4), nume\_instrument VARCHAR2(15), numar\_instrumente NUMBER(4));

ALTER TABLE instrumente\_muzicale

ADD CONSTRAINT PK\_instrument PRIMARY KEY(id\_instrument)

ADD CONSTRAINT FK\_magazin3 FOREIGN KEY (id\_magazin) REFERENCES magazin(id\_magazin)

ADD CONSTRAINT FK\_marca2 FOREIGN KEY (id\_marca) REFERENCES marca(id\_marca);

INSERT INTO instrumente\_muzicale (id\_instrument, id\_marca, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(301, 201, 101, 'Chit. Acustica', 23);

INSERT INTO instrumente\_muzicale (id\_instrument, id\_marca, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(302, 202, 102, 'Chit. Electrica', 57);

INSERT INTO instrumente\_muzicale (id\_instrument, id\_marca, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(303, 203, 103, 'Tobe', 14);

INSERT INTO instrumente\_muzicale (id\_instrument, id\_marca, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(304, 204, 104, 'Pian', 11);

INSERT INTO instrumente\_muzicale (id\_instrument, id\_marca, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(305, 205, 105, 'Vioara', 54);

INSERT INTO instrumente\_muzicale (id\_instrument, id\_marca, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(306, 206, 106, 'Saxofon', 67);

INSERT INTO instrumente\_muzicale (id\_instrument, id\_marca, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(307, 207, 107, 'Trompeta', 45);

INSERT INTO instrumente\_muzicale (id\_instrument, id\_marca, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(308, 208, 108, 'Flaut', 67);

INSERT INTO instrumente\_muzicale (id\_instrument, id\_marca, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(309, 209, 109, 'Chitara Bas', 34);

INSERT INTO instrumente\_muzicale (id\_instrument, id\_marca, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(310, 210, 110, 'Clape', 55);

INSERT INTO instrumente\_muzicale (id\_instrument, id\_marca, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(311, 201, 103, 'Cello', 66);

INSERT INTO instrumente\_muzicale (id\_instrument, id\_marca, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(312, 202, 110, 'Banjo', 13);

INSERT INTO instrumente\_muzicale (id\_instrument, id\_marca, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(313, 203, 107, 'Harpa', 9);

INSERT INTO instrumente\_muzicale (id\_instrument, id\_marca, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(314, 204, 106, 'Mandolina', 24);

INSERT INTO instrumente\_muzicale (id\_instrument, id\_marca, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(315, 205, 110, 'Acordeon', 9);

INSERT INTO instrumente\_muzicale (id\_instrument, id\_marca, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(316, 206, 107, 'Trombon', 14);

INSERT INTO instrumente\_muzicale (id\_instrument, id\_marca, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(317, 207, 101, 'Clarinet', 39);

INSERT INTO instrumente\_muzicale (id\_instrument, id\_marca, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(318, 208, 102, 'Oboi', 15);

INSERT INTO instrumente\_muzicale (id\_instrument, id\_marca, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(319, 209, 109, 'Cimpoi', 7);

INSERT INTO instrumente\_muzicale (id\_instrument, id\_marca, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(320, 210, 107, 'Sitar', 33);

commit;

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

CREATE TABLE livrare(id\_livrare NUMBER(4), id\_instrument NUMBER(4), data\_livrare DATE, data\_sosire DATE);

ALTER TABLE livrare

ADD CONSTRAINT PK\_livrare PRIMARY KEY(id\_livrare)

ADD CONSTRAINT FK\_instrument2 FOREIGN KEY(id\_instrument) REFERENCES instrumente\_muzicale(id\_instrument);

INSERT INTO livrare (id\_livrare, id\_instrument, data\_livrare, data\_sosire)

VALUES(1234, 301, TO\_DATE('2023-05-10', 'YYYY-MM-DD'), TO\_DATE('2023-05-15', 'YYYY-MM-DD'));

INSERT INTO livrare (id\_livrare, id\_instrument, data\_livrare, data\_sosire)

VALUES(5678, 302, TO\_DATE('2023-05-11', 'YYYY-MM-DD'), TO\_DATE('2023-05-16', 'YYYY-MM-DD'));

INSERT INTO livrare (id\_livrare, id\_instrument, data\_livrare, data\_sosire)

VALUES(9876, 303, TO\_DATE('2023-05-12', 'YYYY-MM-DD'), TO\_DATE('2023-05-17', 'YYYY-MM-DD'));

INSERT INTO livrare (id\_livrare, id\_instrument, data\_livrare, data\_sosire)

VALUES(5432, 304, TO\_DATE('2023-05-13', 'YYYY-MM-DD'), TO\_DATE(' 2023-05-18', 'YYYY-MM-DD'));

INSERT INTO livrare (id\_livrare, id\_instrument, data\_livrare, data\_sosire)

VALUES(2468, 305, TO\_DATE('2023-05-14', 'YYYY-MM-DD'), TO\_DATE(' 2023-05-19', 'YYYY-MM-DD'));

INSERT INTO livrare (id\_livrare, id\_instrument, data\_livrare, data\_sosire)

VALUES(1357, 306, TO\_DATE('2023-05-15', 'YYYY-MM-DD'), TO\_DATE(' 2023-05-20', 'YYYY-MM-DD'));

INSERT INTO livrare (id\_livrare, id\_instrument, data\_livrare, data\_sosire)

VALUES(8642, 307, TO\_DATE('2023-05-16', 'YYYY-MM-DD'), TO\_DATE(' 2023-05-21', 'YYYY-MM-DD'));

INSERT INTO livrare (id\_livrare, id\_instrument, data\_livrare, data\_sosire)

VALUES(9753, 308, TO\_DATE('2023-05-17', 'YYYY-MM-DD'), TO\_DATE(' 2023-05-22', 'YYYY-MM-DD'));

INSERT INTO livrare (id\_livrare, id\_instrument, data\_livrare, data\_sosire)

VALUES(7531, 309, TO\_DATE('2023-05-18', 'YYYY-MM-DD'), TO\_DATE(' 2023-05-23', 'YYYY-MM-DD'));

INSERT INTO livrare (id\_livrare, id\_instrument, data\_livrare, data\_sosire)

VALUES(1029, 310, TO\_DATE('2023-05-19', 'YYYY-MM-DD'), TO\_DATE(' 2023-05-24', 'YYYY-MM-DD'));

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

CREATE TABLE clienti(id\_client NUMBER(4), id\_livrare NUMBER(4), id\_magazin NUMBER(4),

prenume VARCHAR2(20), nume VARCHAR2(20), nume\_cont CHAR(15));

ALTER TABLE clienti

ADD CONSTRAINT PK\_client PRIMARY KEY(id\_client)

ADD CONSTRAINT FK\_livrare2 FOREIGN KEY(id\_livrare) REFERENCES livrare(id\_livrare)

ADD CONSTRAINT FK\_magazin4 FOREIGN KEY(id\_magazin) REFERENCES magazin(id\_magazin);

INSERT INTO clienti(id\_client, id\_livrare, id\_magazin, prenume, nume, nume\_cont)

VALUES(3011, 1234, 110, 'Ethan', 'Lewis', 'ethanlewis234');

INSERT INTO clienti(id\_client, id\_livrare, id\_magazin, prenume, nume, nume\_cont)

VALUES(3012, 5678, 101, 'Emma', 'Lee', 'emmalee567');

INSERT INTO clienti(id\_client, id\_livrare, id\_magazin, prenume, nume, nume\_cont)

VALUES(3013, 9876, 102, 'Daniel', 'Walker', 'danielwalker890');

INSERT INTO clienti(id\_client, id\_livrare, id\_magazin, prenume, nume, nume\_cont)

VALUES(3014, 5432, 103, 'Mia', 'Green', 'miagreen123');

INSERT INTO clienti(id\_client, id\_livrare, id\_magazin, prenume, nume, nume\_cont)

VALUES(3015, 9876, 104, 'Alexander', 'Hall', 'alexanderhall');

INSERT INTO clienti(id\_client, id\_livrare, id\_magazin, prenume, nume, nume\_cont)

VALUES(3016, 5432, 105, 'Sophia', 'Turner', 'sophiaturner789');

INSERT INTO clienti(id\_client, id\_livrare, id\_magazin, prenume, nume, nume\_cont)

VALUES(3017, 2468, 106, 'William', 'White', 'williamwhite321');

INSERT INTO clienti(id\_client, id\_livrare, id\_magazin, prenume, nume, nume\_cont)

VALUES(3018, 8642, 107, 'Charlotte', 'Harris', 'charlotteh654');

INSERT INTO clienti(id\_client, id\_livrare, id\_magazin, prenume, nume, nume\_cont)

VALUES(3019, 7531, 108, 'James', 'King', 'jamesking987');

INSERT INTO clienti(id\_client, id\_livrare, id\_magazin, prenume, nume, nume\_cont)

VALUES(3020, 1029, 109, 'Harper', 'Scott', 'harperscott012');

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

CREATE TABLE magazin\_clienti(id\_client NUMBER(4), id\_magazin NUMBER(4), nume\_cont

CHAR(15));

ALTER TABLE magazin\_clienti

ADD CONSTRAINT FK\_ID\_Magazin\_Clienti FOREIGN KEY(id\_client) REFERENCES

clienti(id\_client)

ADD CONSTRAINT FK\_ID\_Clienti\_Magazin FOREIGN KEY(id\_magazin) REFERENCES

magazin(id\_magazin);

INSERT INTO magazin\_clienti(id\_client, id\_magazin, nume\_cont)

VALUES(3011, 110, 'ethanlewis234');

INSERT INTO magazin\_clienti(id\_client, id\_magazin, nume\_cont)

VALUES(3012, 101, 'emmalee567');

INSERT INTO magazin\_clienti(id\_client, id\_magazin, nume\_cont)

VALUES(3013, 102, 'danielwalker890');

INSERT INTO magazin\_clienti(id\_client, id\_magazin, nume\_cont)

VALUES(3014, 103, 'miagreen123');

INSERT INTO magazin\_clienti(id\_client, id\_magazin, nume\_cont)

VALUES(3015, 104, 'alexanderhall');

INSERT INTO magazin\_clienti(id\_client, id\_magazin, nume\_cont)

VALUES(3016, 105, 'sophiaturner789');

INSERT INTO magazin\_clienti(id\_client, id\_magazin, nume\_cont)

VALUES(3017, 106, 'williamwhite321');

INSERT INTO magazin\_clienti(id\_client, id\_magazin, nume\_cont)

VALUES(3018, 107, 'charlotteh654');

INSERT INTO magazin\_clienti(id\_client, id\_magazin, nume\_cont)

VALUES(3019, 108, 'jamesking987');

INSERT INTO magazin\_clienti(id\_client, id\_magazin, nume\_cont)

VALUES(3020, 109, 'harperscott012');

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

CREATE TABLE magazin\_instrumente\_muzicale(id\_instrument NUMBER(4), id\_magazin

NUMBER(4), nume\_instrument VARCHAR(15), numar\_instrumente NUMBER(4));

ALTER TABLE magazin\_instrumente\_muzicale

ADD CONSTRAINT FK\_ID\_Magazin\_Instrumente FOREIGN KEY(id\_magazin) REFERENCES

magazin(id\_magazin)

ADD CONSTRAINT FK\_ID\_Instrumente\_Magazin FOREIGN KEY(id\_instrument)

REFERENCES instrumente\_muzicale(id\_instrument);

INSERT INTO magazin\_instrumente\_muzicale(id\_instrument, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(301, 101, 'Chit. Acustica', 23);

INSERT INTO magazin\_instrumente\_muzicale(id\_instrument, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(302, 102, 'Chit. Electrica', 57);

INSERT INTO magazin\_instrumente\_muzicale(id\_instrument, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(303, 103, 'Tobe', 14);

INSERT INTO magazin\_instrumente\_muzicale(id\_instrument, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(304, 104, 'Pian', 11);

INSERT INTO magazin\_instrumente\_muzicale(id\_instrument, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(305, 105, 'Vioara', 54);

INSERT INTO magazin\_instrumente\_muzicale(id\_instrument, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(306, 106, 'Saxofon', 67);

INSERT INTO magazin\_instrumente\_muzicale(id\_instrument, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(307, 107, 'Trompeta', 45);

INSERT INTO magazin\_instrumente\_muzicale(id\_instrument, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(308, 108, 'Flaut', 67);

INSERT INTO magazin\_instrumente\_muzicale(id\_instrument, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(309, 109, 'Chitara Bas', 34);

INSERT INTO magazin\_instrumente\_muzicale(id\_instrument, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(310, 110, 'Clape', 55);

INSERT INTO magazin\_instrumente\_muzicale(id\_instrument, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(311, 103, 'Cello', 66);

INSERT INTO magazin\_instrumente\_muzicale(id\_instrument, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(312, 110 ,'Banjo', 13);

INSERT INTO magazin\_instrumente\_muzicale(id\_instrument, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(313, 107, 'Harpa', 9);

INSERT INTO magazin\_instrumente\_muzicale(id\_instrument, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(314, 106, 'Mandolina', 24);

INSERT INTO magazin\_instrumente\_muzicale(id\_instrument, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(315, 110, 'Acordeon', 9);

INSERT INTO magazin\_instrumente\_muzicale(id\_instrument, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(316, 107, 'Trombon', 14);

INSERT INTO magazin\_instrumente\_muzicale(id\_instrument, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(317, 101, 'Clarinet', 39);

INSERT INTO magazin\_instrumente\_muzicale(id\_instrument, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(318, 102, 'Oboi', 15);

INSERT INTO magazin\_instrumente\_muzicale(id\_instrument, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(319, 109, 'Cimpoi', 7);

INSERT INTO magazin\_instrumente\_muzicale(id\_instrument, id\_magazin, nume\_instrument, numar\_instrumente)

VALUES(320, 107, 'Sitar', 33);

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

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.

Scrieti un program care, sa introduca intr-un tablou indexat informatii despre instrumentele muzicale si numarul acestora in stoc si afisati informatiile acestea, sa genereze un tablou imbricat care sa afiseze lista celor 5 cei mai platiti angajati si ordonatii crescatori in functie de salariu si afisati pentru fiecare angajat din lista numele si id-ul magazinului unde lucreaza si creati un vector care sa contina o lista ordonata numeric crescator a id-urilor magazinelor in care lucreaza toti angajatii.

CREATE OR REPLACE PROCEDURE Informatii\_Angajati

IS

TYPE t\_instrumente\_radoi IS record (

NumeInstrument instrumente\_muzicale.nume\_instrument%TYPE,

NumarStoc instrumente\_muzicale.numar\_instrumente%TYPE

);

TYPE tablou\_indexat\_radoi IS TABLE OF t\_instrumente\_radoi INDEX BY PLS\_INTEGER;

t\_indexat tablou\_indexat\_radoi;

TYPE tablou\_imbricat\_radoi IS TABLE OF angajati%ROWTYPE;

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

TYPE tablou\_vector\_radoi IS VARRAY(100) OF angajati.id\_magazin%TYPE;

t\_vector\_magazin tablou\_vector\_radoi := tablou\_vector\_radoi();

BEGIN

for informatii in (select i.nume\_instrument, i.numar\_instrumente from instrumente\_muzicale i) loop

t\_indexat(t\_indexat.count + 1).NumeInstrument := informatii.nume\_instrument;

t\_indexat(t\_indexat.count).NumarStoc := informatii.numar\_instrumente;

end loop;

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

DBMS\_OUTPUT.PUT\_LINE(t\_indexat(i).NumeInstrument || ' are in stoc un numar de ' || t\_indexat(i).NumarStoc || ' exemplare' );

end loop;

DBMS\_OUTPUT.NEW\_LINE;

for informatii in (select \* from angajati order by salariu desc) loop

t\_imbricat.extend;

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

exit when t\_imbricat.count = 5;

end loop;

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

DBMS\_OUTPUT.PUT\_LINE(t\_imbricat(i).nume || ' lucreaza in magazinul ' || t\_imbricat(i).id\_magazin);

end loop;

DBMS\_OUTPUT.NEW\_LINE;

FOR informatii IN (SELECT DISTINCT id\_magazin FROM angajati ORDER BY id\_magazin) LOOP

IF t\_vector\_magazin.COUNT < t\_vector\_magazin.LIMIT THEN

t\_vector\_magazin.EXTEND;

t\_vector\_magazin(t\_vector\_magazin.COUNT) := informatii.id\_magazin;

END IF;

END LOOP;

FOR i IN 1..t\_vector\_magazin.COUNT LOOP

DBMS\_OUTPUT.PUT\_LINE('Magazinul: ' || t\_vector\_magazin(i));

END LOOP;

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('A aparut o eroare: ' || SQLERRM);

END Informatii\_Angajati;

/

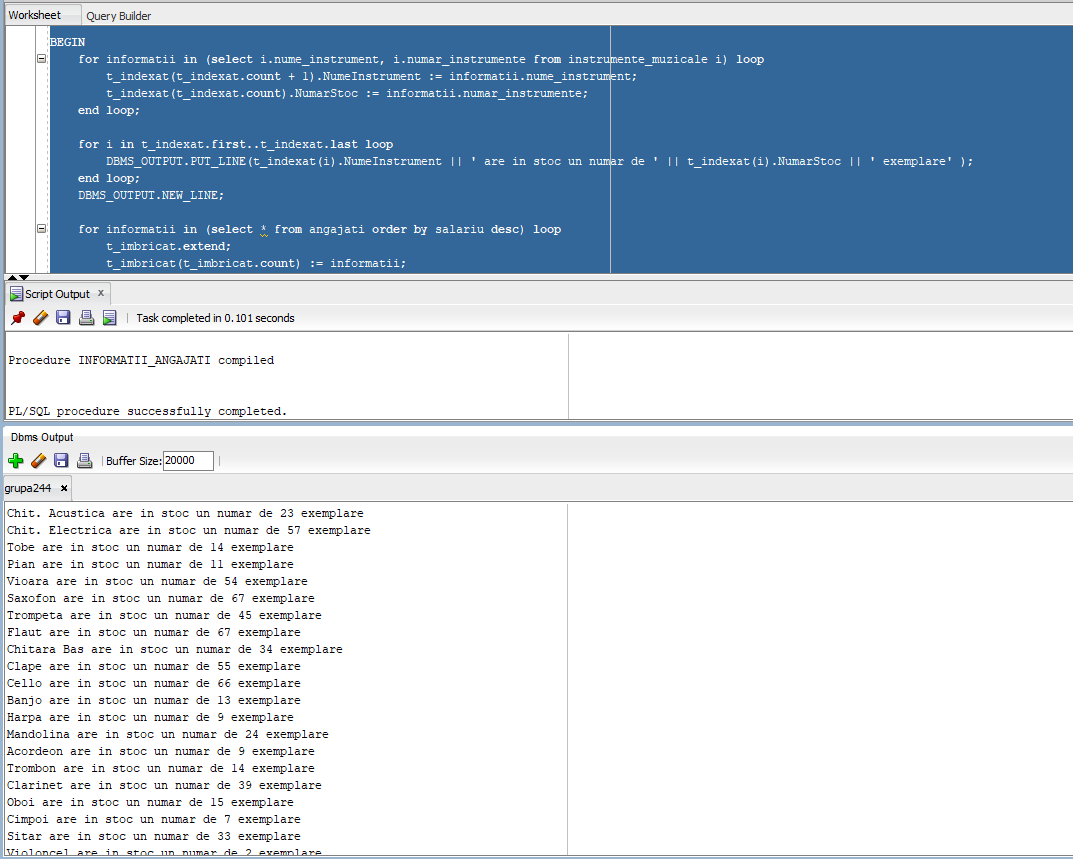
BEGIN

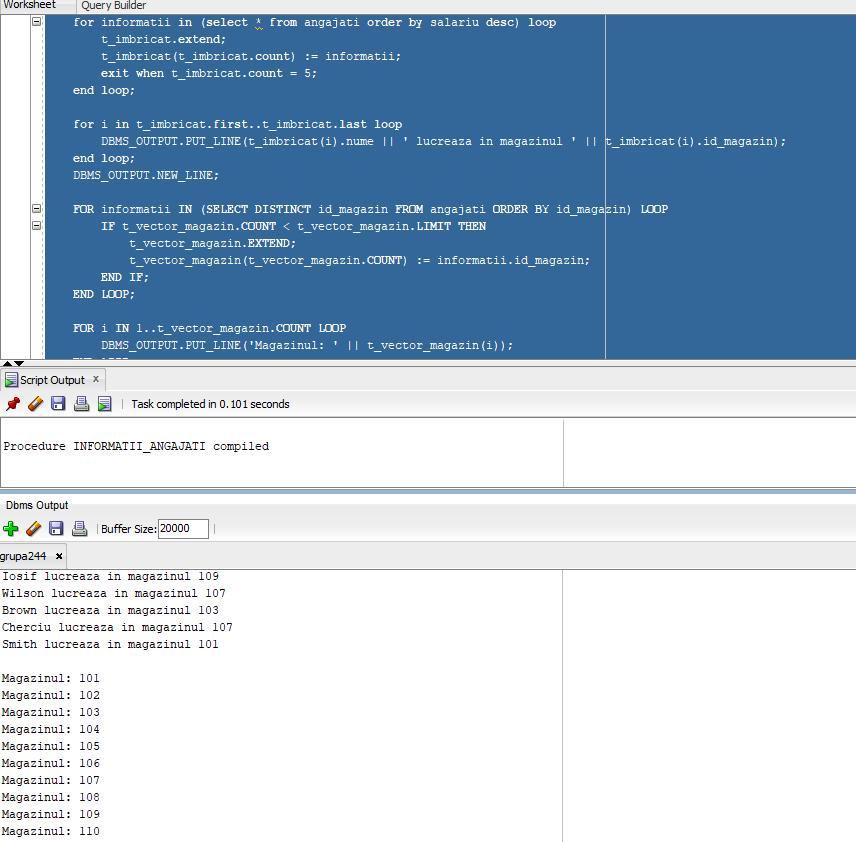
Informatii\_Angajati;

END;

/

Tabloul indexat:





Tabloul imbricat si vectorul:

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.

Scrieti un program care sa introduca intr-un cursor id-ul si numele marcilor si sa introduca intr-un cursor parametrizat numele si id-ul tuturor instrumentelor. Afisati toate marcile si sub fiecare marca sa afisati instrumentele care au acel tip de marca

create or replace procedure cursoare\_radoi is

cursor cursor\_dragos is select id\_marca, nume\_marca

from marca;

cursor cursor\_cosmin (param marca.id\_marca%type) is

select nume\_instrument, id\_instrument

from instrumente\_muzicale

where id\_marca = param;

d\_id\_marca marca.id\_marca%type;

d\_nume\_marca marca.nume\_marca%type;

begin

open cursor\_dragos;

loop

fetch cursor\_dragos into d\_id\_marca, d\_nume\_marca;

exit when cursor\_dragos%notfound;

dbms\_output.put\_line('Marca ' || d\_nume\_marca);

for i in cursor\_cosmin(d\_id\_marca) loop

dbms\_output.put\_line(i.nume\_instrument);

end loop;

end loop;

end cursoare\_radoi;

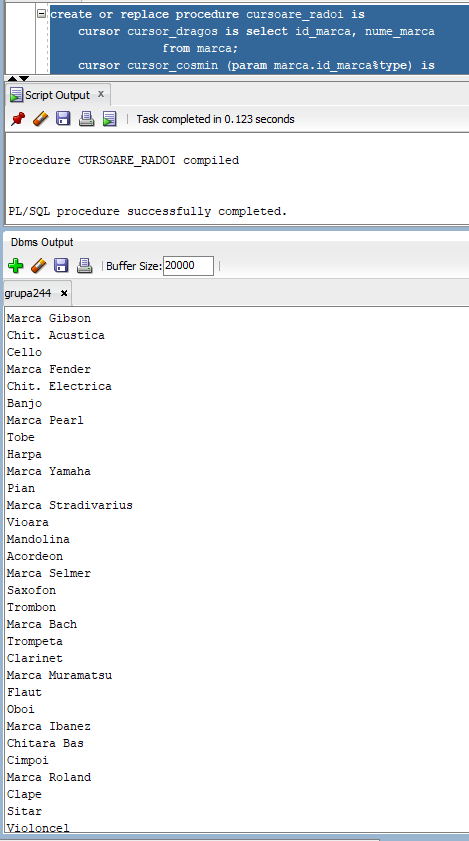
/

begin

cursoare\_radoi;

end;

/



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 definite.

Definiți minim 2 excepții proprii. Apelați subprogramul astfel încât să evidențiați toate cazurile

definite și tratate.

Scrieti un subprogram care afiseaza orasul in care se afla instrumentul mentionat, iar instrumentul sa aiba intre 20 si 50 de exemplare in magazin. Apelati subprogramul astfel incat sa fie tratate toate cazurile definite si tratate.

--Ex\_8

create or replace function cautare\_locatie\_instrument

(v\_nume\_instrument instrumente\_muzicale.nume\_instrument%TYPE DEFAULT 'Tobe')

return VARCHAR is

nume\_oras locatie.oras%type;

numar\_instrumente instrumente\_muzicale.numar\_instrumente%type;

exceptie\_1 EXCEPTION;

exceptie\_2 EXCEPTION;

begin

select oras, i.numar\_instrumente into nume\_oras, numar\_instrumente

from instrumente\_muzicale i, magazin m, locatie l

where nume\_instrument = v\_nume\_instrument

and i.id\_magazin = m.id\_magazin

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

if numar\_instrumente < 20 then

raise exceptie\_1;

elsif numar\_instrumente > 50 then

raise exceptie\_2;

else

return nume\_oras;

end if;

EXCEPTION

WHEN exceptie\_1 THEN

DBMS\_OUTPUT.PUT\_LINE('Magazinul nu are destule instrumente');

WHEN exceptie\_2 THEN

DBMS\_OUTPUT.PUT\_LINE('Magazinul are prea multe instrumente');

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20000, 'Nu exista instrumente cu numele dat');

WHEN OTHERS THEN

RAISE\_APPLICATION\_ERROR(-20001,'Alta eroare!');

end cautare\_locatie\_instrument;

/

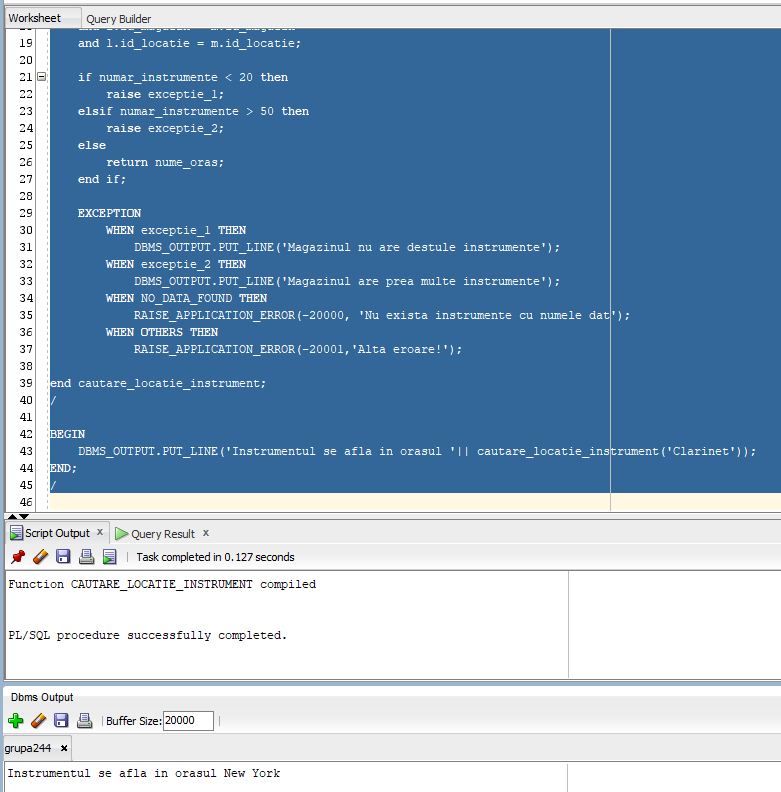
BEGIN

DBMS\_OUTPUT.PUT\_LINE('Instrumentul se afla in orasul '|| cautare\_locatie\_instrument('Saxofon'));

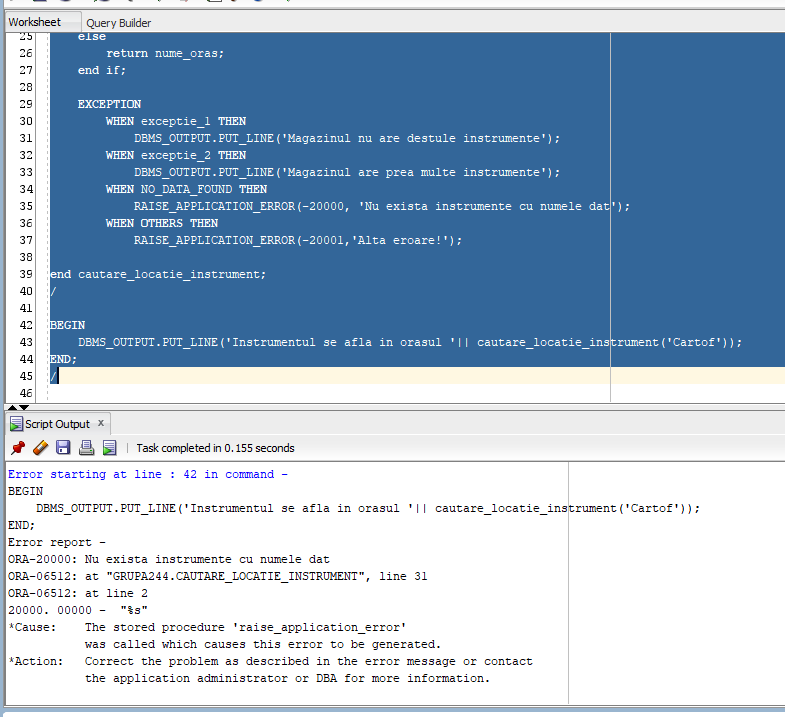
END;

/

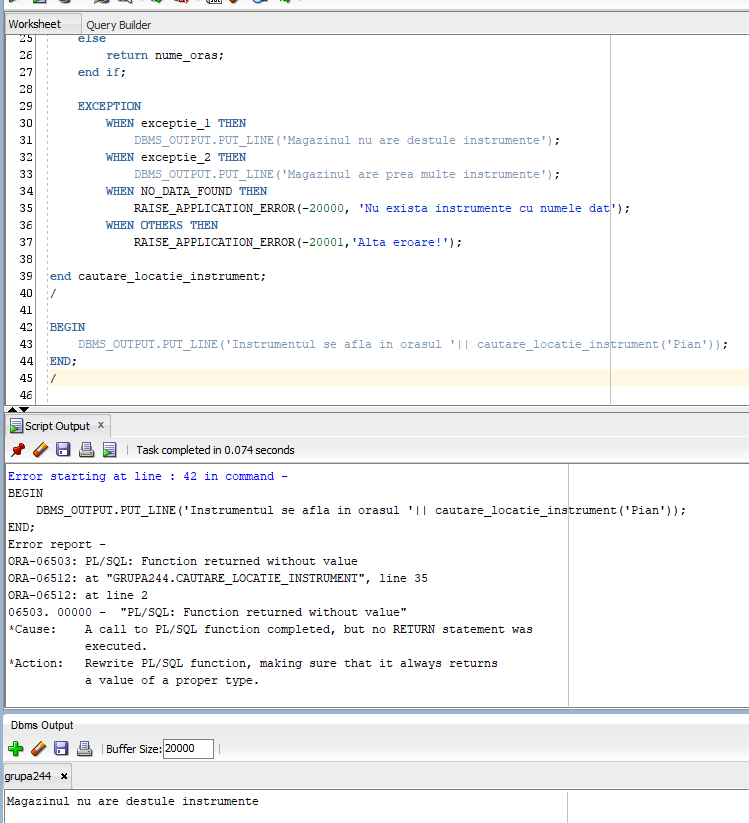
Cazul in care exista instrumentul:



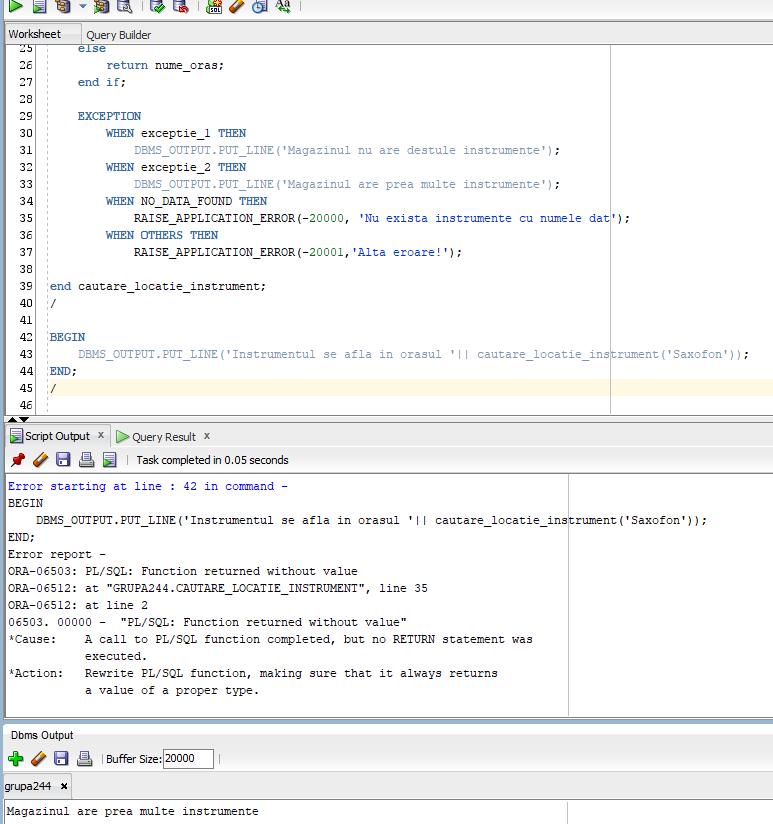
Cazul in care nu exista instrumentul:



Cazul in care exista, dar are sub 20 de exemplare in magazin



Cazul in care exista, dar are peste 50 de instrumente



9)Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat

independent de tip procedură care să utilizeze într-o singură comandă SQL 5 dintre tabelele

definite. Tratați toate excepțiile care pot apărea, incluzând excepțiile NO\_DATA\_FOUND și

TOO\_MANY\_ROWS. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

Scrieti un subprogram stocat independent de tip procedura care sa afiseze numele orasului unde sa afla instrumentele cu o anumita marca.

create or replace procedure oras\_marca

(v\_marca marca.nume\_marca%TYPE)

is

city locatie.oras%TYPE;

begin

select l.oras into city

from locatie l, magazin m, magazin\_instrumente\_muzicale mi, instrumente\_muzicale i, marca ma

where v\_marca = nume\_marca

and ma.id\_marca = i.id\_marca

and i.nume\_instrument = mi.nume\_instrument

and i.id\_instrument = mi.id\_instrument

and mi.id\_magazin = m.id\_magazin

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

DBMS\_OUTPUT.PUT\_LINE('Orasul in care se afla marca mentionata este '|| city);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20000, 'Nu exista orase in care se afla instrumente cu marca data');

WHEN TOO\_MANY\_ROWS THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Exista mai multe orase in care se afla instrumente cu marca data');

WHEN OTHERS THEN

RAISE\_APPLICATION\_ERROR(-20002,'Alta eroare!');

end oras\_marca;

/

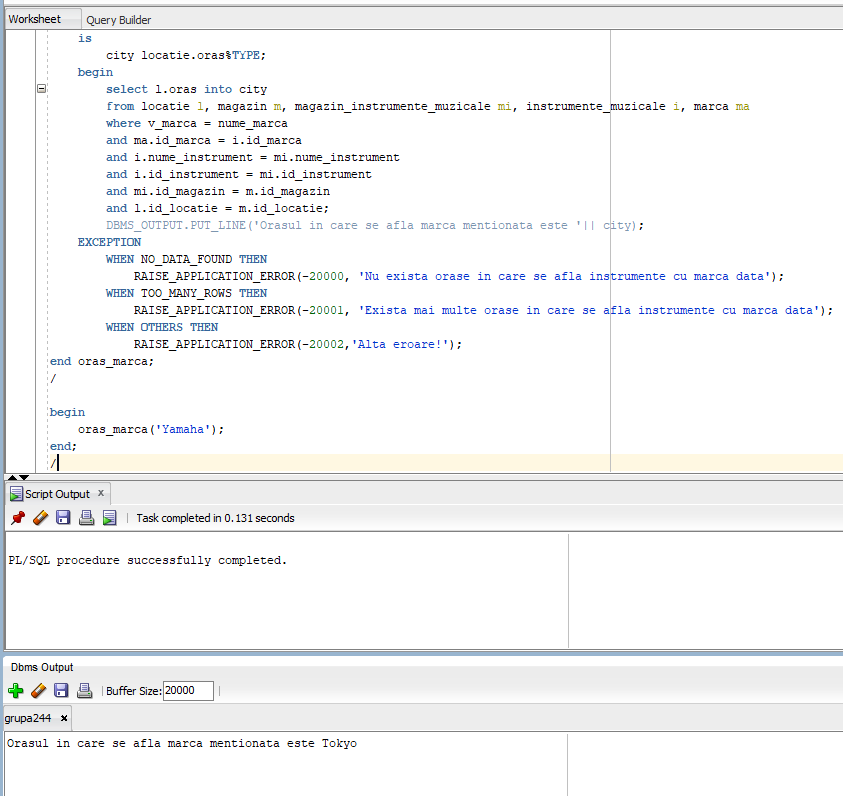
begin

oras\_marca('Yamaha');

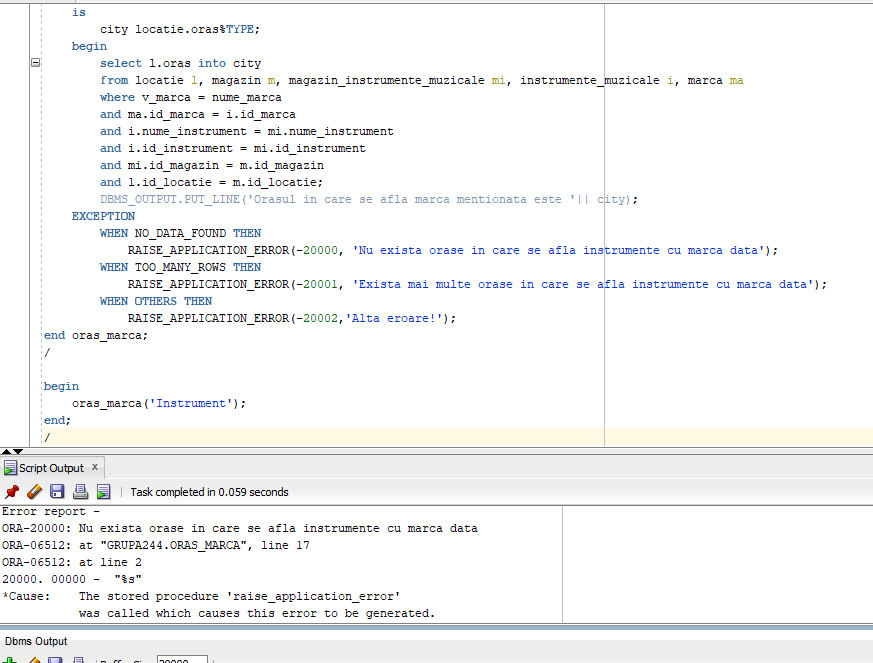
end;

/

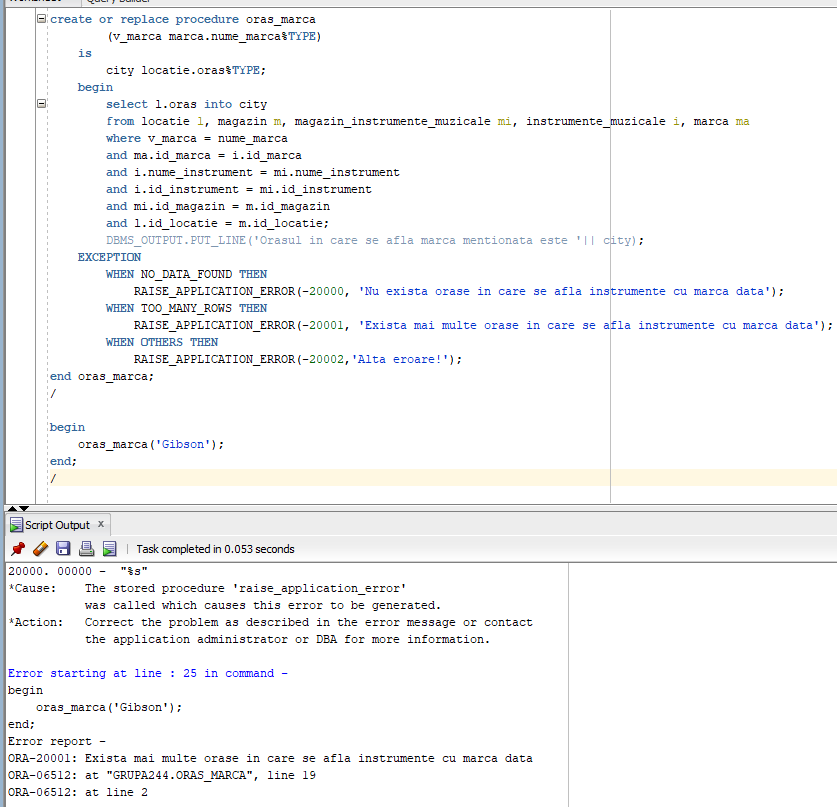
Cazul in care instrumentul exista:



Cazul in care instrumentul nu exista:



Cazul TOO\_MANY\_ROWS:



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

Definiți un declanșator care să permită lucrul asupra tabelului marca (INSERT, UPDATE,

DELETE) doar în intervalul de ore 12:00 - 13:00, de luni până sâmbătă.

CREATE OR REPLACE TRIGGER triggering\_radoi

BEFORE INSERT OR UPDATE OR DELETE ON marca

BEGIN

IF (TO\_CHAR(SYSDATE,'D') = 1)

OR (TO\_CHAR(SYSDATE,'HH24') NOT BETWEEN 12 AND 13)

THEN

RAISE\_APPLICATION\_ERROR(-20001,'tabelul nu poate fi actualizat');

END IF;

END;

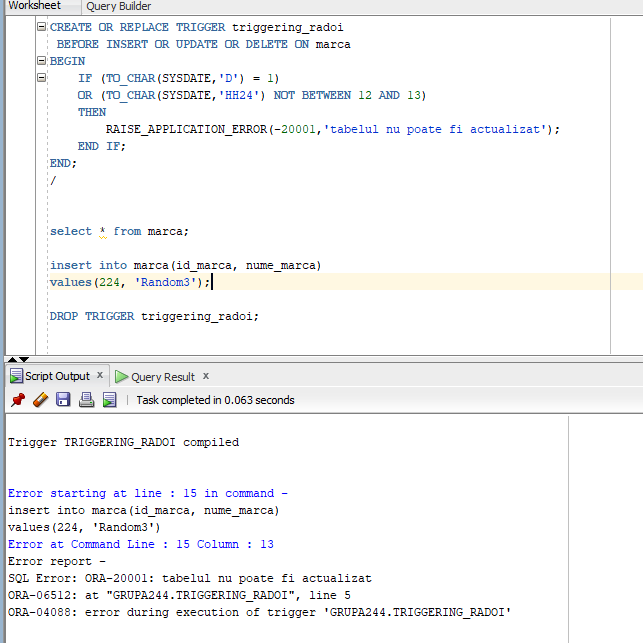
/

select \* from marca;

insert into marca(id\_marca, nume\_marca)

values(223, 'Random2');

DROP TRIGGER triggering\_radoi;



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

Definiți un declanșator prin care să nu se permită marirea numarului de instrumente din tabelul instrumente\_muzicale.

select \* from instrumente\_muzicale;

CREATE OR REPLACE TRIGGER triggering\_radoi

BEFORE UPDATE OF numar\_instrumente ON instrumente\_muzicale

FOR EACH ROW

BEGIN

IF (:NEW.numar\_instrumente > :OLD.numar\_instrumente) THEN

RAISE\_APPLICATION\_ERROR(-20002,'numarul de instrumente nu poate fi marit');

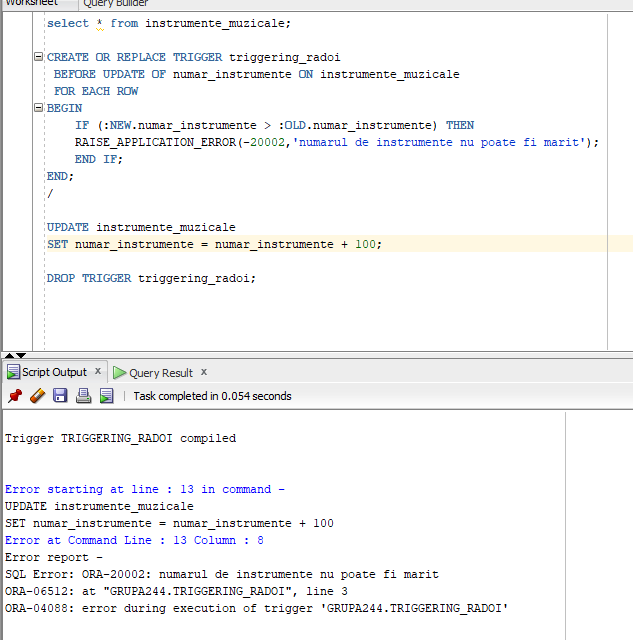
END IF;

END;

/

UPDATE instrumente\_muzicale

SET numar\_instrumente = numar\_instrumente + 100;

DROP TRIGGER triggering\_radoi;

12)Definiți un trigger de tip LDD. Declanșați trigger-ul.

Creati un trigger care confirma modificarea unui tabel.

CREATE OR REPLACE TRIGGER trigger\_radoi\_3

AFTER ALTER ON SCHEMA

BEGIN

dbms\_output.put\_line('Un tabel a fost modificat: ' || SYS.DICTIONARY\_OBJ\_NAME);

END;

/

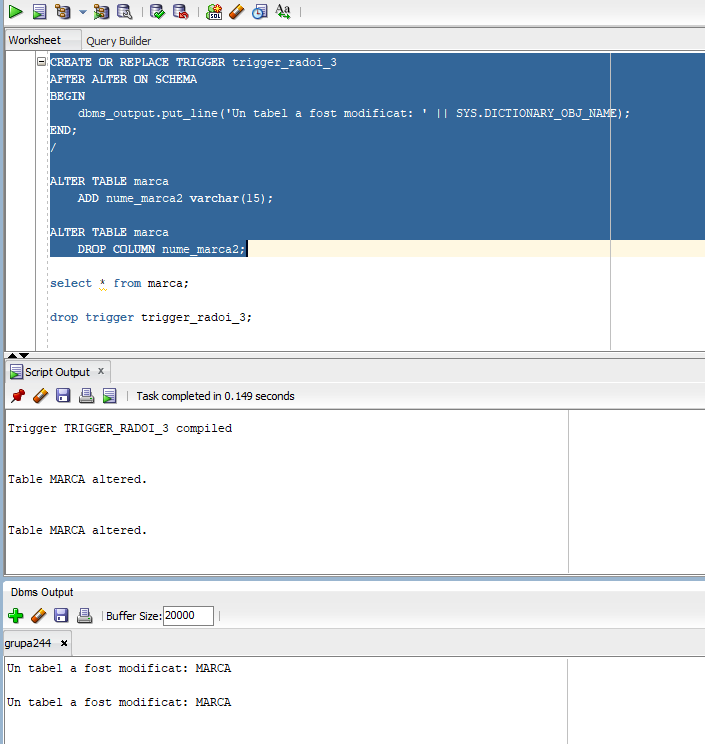
ALTER TABLE marca

ADD nume\_marca2 varchar(15);

ALTER TABLE marca

DROP COLUMN nume\_marca2;

select \* from marca;

drop trigger trigger\_radoi\_3;