

Music Shop Geo&Alina

1.

În cadrul acestui proiect voi prezenta baza de date ce gestionează activitățile din cadrul unui music shop. În centru se afla produsele - precum instrumentele, aparatele de redare a sunetului sau alte auxiliare - ținând evidența magazinelor, depozitelor, lutierilor și transporturilor în cadrul cărora se află, precum și a vehiculelor și șoferilor ce se ocupă de transporturi, opririle intermediare ale acestora, angajații magazinelor și funcțiile lor.

Restricții:

Un produs nu se poate afla în același timp în două locații diferite (livrare, magazin, depozit, lutier), la fel cum un unic vehicul și un unic șofer se pot ocupa de un singur transport odată. Produsele aflate într-un depozit sau într-un magazin pot fi duse la un lutier aflat în același județ.

Un angajat lucrează la un singur magazin, la un magazin pot lucra mai mulți angajați.

Un angajat are o singură funcție, o funcție poate aparține mai multor angajați.

Un magazin se află într-o singură locație, într-o locație se poate afla un magazin.

Într-un magazin se pot afla mai multe produse, un produs se poate afla într-un magazin.

Un lutier se află într-o locație, într-o locație se poate afla un lutier.

Un produs poate fi la un lutier, un lutier poate repara mai multe produse.

Un produs poate fi într-un depozit, într-un depozit pot fi mai multe produse.

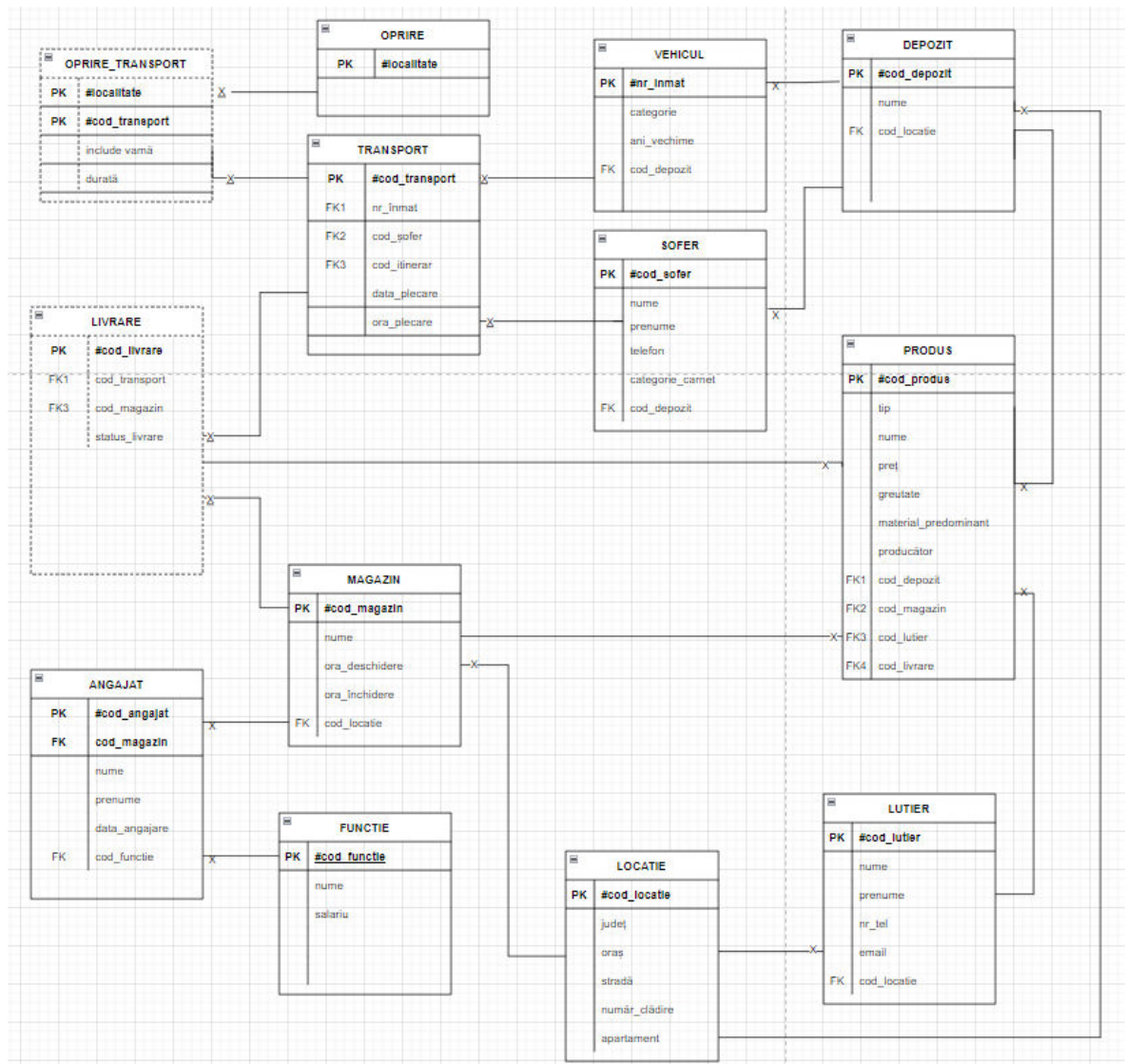
Un depozit se află într-o singură locație, într-o locație se poate afla un depozit.

Un depozit poate avea mai multe vehicule, un vehicul aparține unui singur depozit.

La un depozit pot lucra mai mulți șoferi, un șofer lucrează la un singur depozit.

Un transport poate face mai multe opriri, o oprire poate reține mai multe transporturi.

3.



4.

CREATE TABLE locatie

(cod_locatie NUMBER(10) NOT NULL,

judet VARCHAR2(20),

oras VARCHAR2(20),

strada VARCHAR2(20) ,

numar_cladire NUMBER(10),

apartament NUMBER(10),

CONSTRAINT pk_cod_locatie PRIMARY KEY (cod_locatie)

);

```
CREATE TABLE lutier
(cod_lutier NUMBER(10) NOT NULL,
nume VARCHAR2(20),
prenume VARCHAR2(20),
nr_tel NUMBER(10) ,
email VARCHAR2(20),
cod_locatie NUMBER(10) NOT NULL,
CONSTRAINT pk_cod_lutier PRIMARY KEY (cod_lutier),
CONSTRAINT fk_cod_locatie FOREIGN KEY (cod_locatie)
REFERENCES locatie(cod_locatie)
);
```

```
CREATE TABLE magazin
(cod_magazin NUMBER(10) NOT NULL,
nume VARCHAR2(20),
ora_deschidere NUMBER(4),
ora_inchidere NUMBER(4),
cod_locatie NUMBER(10) NOT NULL,
CONSTRAINT pk_cod_magazin PRIMARY KEY (cod_magazin),
CONSTRAINT fk1_cod_locatie FOREIGN KEY (cod_locatie)
REFERENCES locatie(cod_locatie),
CONSTRAINT chk_ora_deschidere CHECK (ora_deschidere >= 0 AND
ora_deschidere < 2400),
CONSTRAINT chk_ora_inchidere CHECK (ora_inchidere >= 0 AND ora_inchidere
< 2400)
);
```

```
CREATE TABLE depozit
(cod_depozit NUMBER(10) NOT NULL,
```

```
nume VARCHAR2(20),
cod_locatie NUMBER(10) NOT NULL,
CONSTRAINT pk_cod_depozit PRIMARY KEY (cod_depozit),
CONSTRAINT fk2_cod_locatie FOREIGN KEY (cod_locatie)
REFERENCES locatie(cod_locatie)
);
```

```
CREATE TABLE functie
(cod_functie NUMBER(10) NOT NULL,
nume VARCHAR2(20),
salariu NUMBER(10),
CONSTRAINT pk_cod_functie PRIMARY KEY (cod_functie),
CONSTRAINT chk_salariu CHECK (salariu > 0)
);
```

```
CREATE TABLE angajat
(cod_angajat NUMBER(10) NOT NULL,
nume VARCHAR2(20),
prenume VARCHAR2(20),
data_angajare DATE,
cod_magazin NUMBER(10) NOT NULL,
cod_functie NUMBER(10) NOT NULL,
CONSTRAINT pk_cod_angajat PRIMARY KEY (cod_angajat),
CONSTRAINT fk1_cod_magazin FOREIGN KEY (cod_magazin)
REFERENCES magazin(cod_magazin),
CONSTRAINT fk_cod_functie FOREIGN KEY (cod_functie)
REFERENCES functie (cod_functie)
);
```

```
CREATE TABLE vehicul
(nr_inmat VARCHAR2(20) NOT NULL,
categorie VARCHAR2(20) NOT NULL,
ani_vechime NUMBER(3),
cod_depozit NUMBER(10) NOT NULL,
CONSTRAINT pk_nr_inmat PRIMARY KEY (nr_inmat),
CONSTRAINT fk1_cod_depozit FOREIGN KEY (cod_depozit)
REFERENCES depozit(cod_depozit)
);
```

```
CREATE TABLE sofer
(cod_sofer NUMBER(10) NOT NULL,
nume VARCHAR2(20),
prenume VARCHAR2(20),
telefon NUMBER(20),
categorie_carnet VARCHAR2(20) NOT NULL,
cod_depozit NUMBER(10) NOT NULL,
CONSTRAINT pk_cod_sofer PRIMARY KEY (cod_sofer),
CONSTRAINT fk2_cod_depozit FOREIGN KEY (cod_depozit)
REFERENCES depozit(cod_depozit)
);
```

```
CREATE TABLE oprire
(localitate VARCHAR2(20) NOT NULL,
CONSTRAINT pk_localitate PRIMARY KEY (localitate)
);
```

```
CREATE TABLE transport
(cod_transport NUMBER(10) NOT NULL,
nr_inmat VARCHAR2(20) NOT NULL,
cod_sofer NUMBER(10) NOT NULL,
data_plecure DATE,
ora_plecure NUMBER(4),
CONSTRAINT pk_cod_transport PRIMARY KEY (cod_transport),
CONSTRAINT fk1_cod_sofer FOREIGN KEY (cod_sofer)
REFERENCES sofer(cod_sofer),
CONSTRAINT fk1_nr_inmat FOREIGN KEY (nr_inmat)
REFERENCES vehicul(nr_inmat),
CONSTRAINT chk1_ora_plecure CHECK (ora_plecure >= 0 AND ora_plecure <
2400)
);
```

```
CREATE TABLE oprire_transport
(cod_transport NUMBER(10) NOT NULL,
localitate VARCHAR2(20) NOT NULL,
include_vama VARCHAR2(2),
durata VARCHAR2(20),
CONSTRAINT pk_cod_transport_localitate PRIMARY KEY (cod_transport,
localitate),
CONSTRAINT fk_cod_transport FOREIGN KEY (cod_transport)
REFERENCES transport(cod_transport),
CONSTRAINT fk_localitate FOREIGN KEY (localitate)
REFERENCES oprire(localitate),
CONSTRAINT chk_include_vama CHECK (LOWER(include_vama) LIKE 'da' OR
LOWER(include_vama) LIKE 'nu')
);
```

```
CREATE TABLE livrare
(cod_livrare NUMBER(10) NOT NULL,
cod_transport NUMBER(10) NOT NULL,
cod_magazin NUMBER(10) NOT NULL,
status_livrare VARCHAR2(20) NOT NULL,
CONSTRAINT pk_cod_livrare PRIMARY KEY (cod_livrare),
CONSTRAINT fk1_cod_transport FOREIGN KEY (cod_transport)
REFERENCES transport(cod_transport),
CONSTRAINT fk2_cod_magazin FOREIGN KEY (cod_magazin)
REFERENCES magazin(cod_magazin)
);
```

```
CREATE TABLE produs
(cod_produs NUMBER(10) NOT NULL,
cod_depozit NUMBER(10),
cod_magazin NUMBER(10),
cod_lutier NUMBER(10),
cod_livrare NUMBER(10),
tip VARCHAR2(20),
nume VARCHAR2(20),
pret NUMBER(10),
greutate NUMBER(10),
material_predominant VARCHAR2(20),
producator VARCHAR2(20),
CONSTRAINT pk_cod_produs PRIMARY KEY (cod_produs),
CONSTRAINT fk4_cod_depozit FOREIGN KEY (cod_depozit)
REFERENCES depozit(cod_depozit),
```



```
CONSTRAINT fk4_cod_magazin FOREIGN KEY (cod_magazin)
REFERENCES magazin(cod_magazin),
CONSTRAINT fk4_cod_lutier FOREIGN KEY (cod_lutier)
REFERENCES lutier(cod_lutier),
CONSTRAINT fk4_cod_livrare FOREIGN KEY (cod_livrare)
REFERENCES livrare(cod_livrare),
CONSTRAINT chk4_pret CHECK (pret >= 0),
CONSTRAINT chk4_greutate CHECK (greutate >= 0),
CONSTRAINT chk_singur_loc CHECK (DECODE(cod_depozit, NULL, 0, 1) +
DECODE(cod_magazin, NULL, 0, 1) + DECODE(cod_lutier, NULL, 0, 1) +
DECODE(cod_livrare, NULL, 0, 1) <= 1)
);
```

```
ALTER TABLE produs
MODIFY cod_depozit DEFAULT NULL;
ALTER TABLE produs
MODIFY cod_magazin DEFAULT NULL;
ALTER TABLE produs
MODIFY cod_lutier DEFAULT NULL;
ALTER TABLE produs
MODIFY cod_livrare DEFAULT NULL;
ALTER TABLE livrare
MODIFY status_livrare DEFAULT 'pe drum';
```

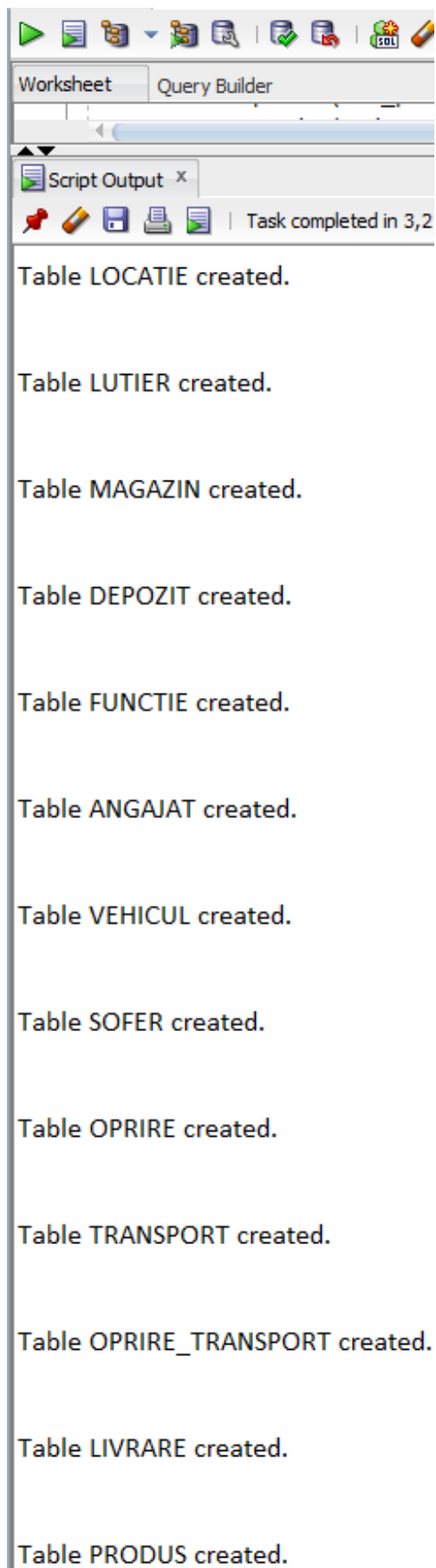


Table PRODUS created.

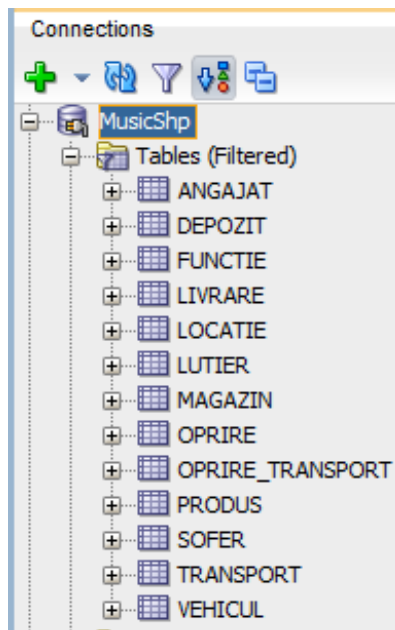
Table PRODUS altered.

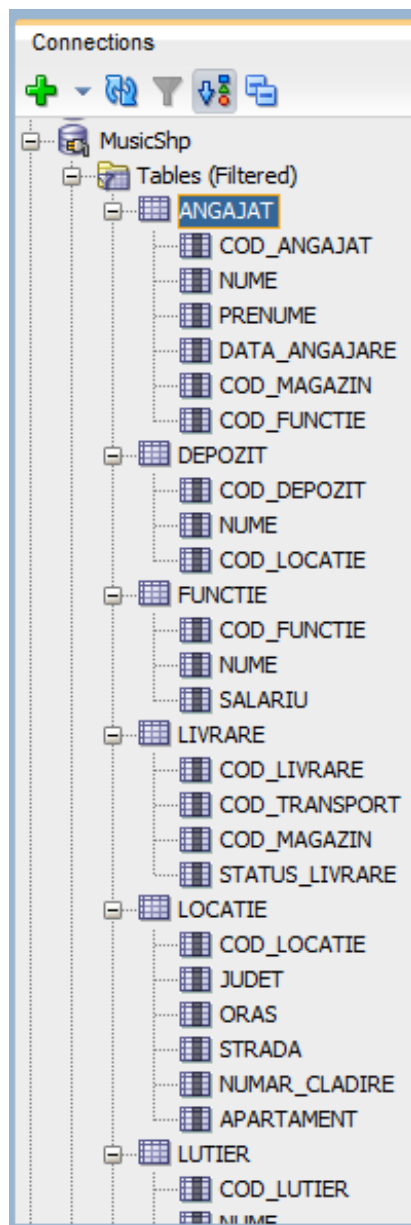
Table PRODUS altered.

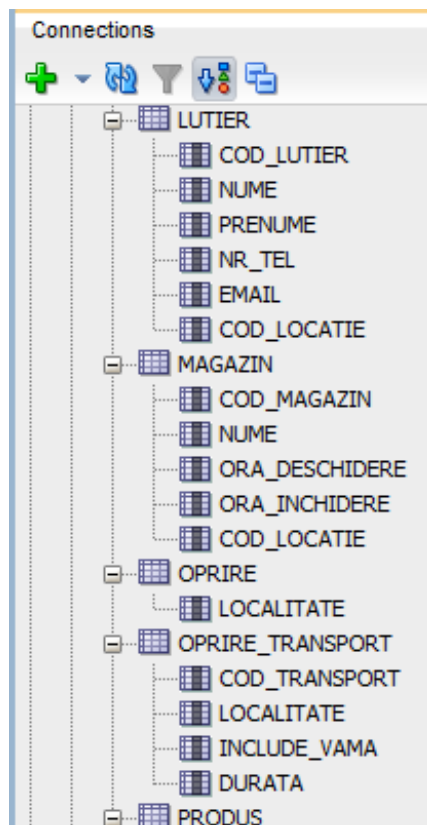
Table PRODUS altered.

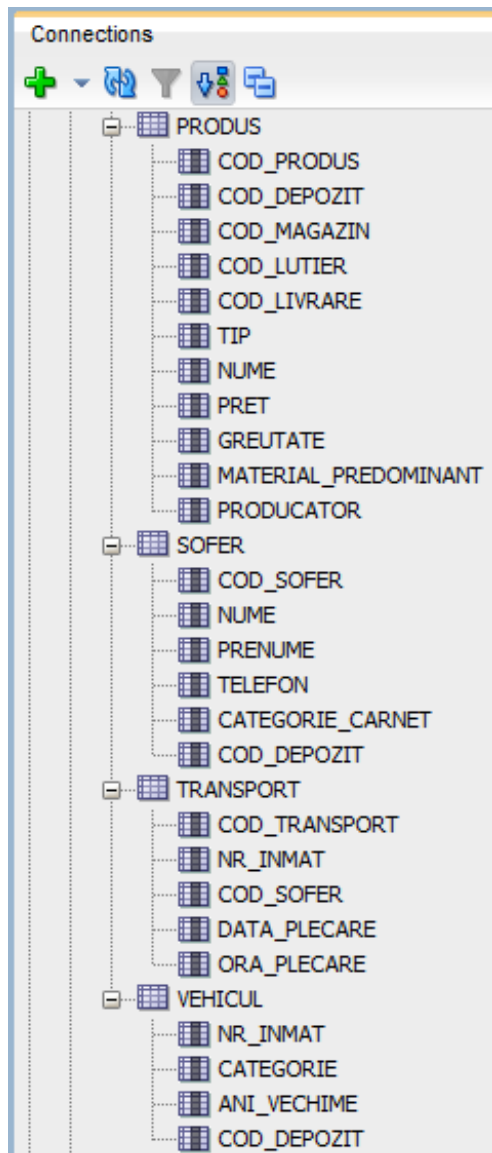
Table PRODUS altered.

Table LIVRARE altered.









5.

INSERT INTO locatie(cod_locatie, judet, oras, strada, numar_cladire, apartament) VALUES(1, 'Ilfov', 'Bucuresti', 'Strada Soarelui', 2, 6);

INSERT INTO locatie(cod_locatie, judet, oras, strada, numar_cladire, apartament) VALUES(2, 'Ilfov', 'Bucuresti', 'Calea Victoriei', 44, 15);

INSERT INTO locatie(cod_locatie, judet, oras, strada, numar_cladire, apartament) VALUES(3, 'Ilfov', 'Bucuresti', 'Strada Aurel Vlaicu', 63, 22);

INSERT INTO locatie(cod_locatie, judet, oras, strada, numar_cladire, apartament) VALUES(4, 'Ilfov', 'Bucuresti', 'Strada Dealului', 1, 19);

INSERT INTO locatie(cod_locatie, judet, oras, strada, numar_cladire, apartament) VALUES(5, 'Sibiu', 'Sibiu', 'Strada Ion Creanga', 23, 55);

```
INSERT INTO locatie(cod_locatie, judet, oras, strada, numar_cladire,
apartament) VALUES(6, 'Sibiu', 'Sibiu', 'Strada Morii', 50, 14);

INSERT INTO locatie(cod_locatie, judet, oras, strada, numar_cladire,
apartament) VALUES(7, 'Sibiu', 'Sibiu', 'Strada Rasaritului', 33, 33);

INSERT INTO locatie(cod_locatie, judet, oras, strada, numar_cladire,
apartament) VALUES(8, 'Constanta', 'Constanta', 'Strada Brezei', 53, 84);

INSERT INTO locatie(cod_locatie, judet, oras, strada, numar_cladire,
apartament) VALUES(9, 'Arges', 'Pitesti', 'Calea Mihai Viteazul', 52, 45);

INSERT INTO locatie(cod_locatie, judet, oras, strada, numar_cladire,
apartament) VALUES(10, 'Dolj', 'Craiova', 'Aleea Dumbavei', 12, 14);

INSERT INTO locatie(cod_locatie, judet, oras, strada, numar_cladire,
apartament) VALUES(11, 'Constanta', 'Constanta', 'Aleea Scoicilor', 1, 38);

INSERT INTO locatie(cod_locatie, judet, oras, strada, numar_cladire,
apartament) VALUES(12, 'Arges', 'Topoloveni', 'S-duta Topolovenilor', 2, 4);

INSERT INTO locatie(cod_locatie, judet, oras, strada, numar_cladire,
apartament) VALUES(13, 'Dolj', 'Craiova', 'Via Sarii', 1, 8);
```

```
INSERT INTO lutier(cod_lutier, nume, prenume, nr_tel, email, cod_locatie)
VALUES(1, 'Ionescu', 'Gigel', '0732103441', 'igigleandr@yahoo.com', 1);

INSERT INTO lutier(cod_lutier, nume, prenume, nr_tel, email, cod_locatie)
VALUES(6, 'Ionescu', 'Andrei', '0732214562', 'iandreei@yahoo.com', 1);

INSERT INTO lutier(cod_lutier, nume, prenume, nr_tel, email, cod_locatie)
VALUES(2, 'Popescu', 'Marcel', '0730286884', 'tituspoma@yahoo.com', 2);

INSERT INTO lutier(cod_lutier, nume, prenume, nr_tel, email, cod_locatie)
VALUES(3, 'Stroe', 'Andrei', '0734940265', 'stroe23@yahoo.com', 3);

INSERT INTO lutier(cod_lutier, nume, prenume, nr_tel, email, cod_locatie)
VALUES(4, 'Stoici', 'Alexandru', '0735273197', 'stoalex2@yahoo.com', 4);

INSERT INTO lutier(cod_lutier, nume, prenume, nr_tel, email, cod_locatie)
VALUES(5, 'Aioanei', 'Marius', '0730784294', 'aioaneity@google.com', 5);
```

```
INSERT INTO magazin(cod_magazin, nume, ora_deschidere, ora_inchidere,
cod_locatie) VALUES(1, 'Geo+Alina Soarelui', 800, 1900, 1);
```

```
INSERT INTO magazin(cod_magazin, nume, ora_deschidere, ora_inchidere,
cod_locatie) VALUES(2, 'Geo+Alina Morii', 800, 2000, 6);
```

```
INSERT INTO magazin(cod_magazin, nume, ora_deschidere, ora_inchidere,
cod_locatie) VALUES(3, 'Geo+Alina Tanta', 1200, 1600, 8);
```

```
INSERT INTO magazin(cod_magazin, nume, ora_deschidere, ora_inchidere,
cod_locatie) VALUES(4, 'Geo+Alina Viteazul', 800, 1900, 9);
```

```
INSERT INTO magazin(cod_magazin, nume, ora_deschidere, ora_inchidere,
cod_locatie) VALUES(5, 'Geo+Alina Dumbravii', 700, 1800, 10);
```

```
INSERT INTO depozit(cod_depozit, nume, cod_locatie) VALUES (1, 'Ilfov Stocks',
1);
```

```
INSERT INTO depozit(cod_depozit, nume, cod_locatie) VALUES (2, 'Depozit
Dealului', 4);
```

```
INSERT INTO depozit(cod_depozit, nume, cod_locatie) VALUES (3, 'ShineDepo',
7);
```

```
INSERT INTO depozit(cod_depozit, nume, cod_locatie) VALUES (4, 'Magazia lui
Marin', 11);
```

```
INSERT INTO depozit(cod_depozit, nume, cod_locatie) VALUES (5, 'Depozit
Blocului', 12);
```

```
INSERT INTO depozit(cod_depozit, nume, cod_locatie) VALUES (6, 'Depozit
Blocului', 12);
```

```
INSERT INTO depozit(cod_depozit, nume, cod_locatie) VALUES (7, 'Depozit599',
13);
```

```
INSERT INTO functie(cod_functie, nume, salariu) VALUES (1, 'Director Stocuri',
4400);
```

```
INSERT INTO functie(cod_functie, nume, salariu) VALUES (2, 'Agent Vanzari',
2800);
```

```
INSERT INTO functie(cod_functie, nume, salariu) VALUES (3, 'Manager Vanzari',
3700);
```

```
INSERT INTO functie(cod_functie, nume, salariu) VALUES (4, 'Specialist Electr.',
3200);
```



```
INSERT INTO functie(cod_functie, nume, salariu) VALUES (5, 'Resp. Intretinere', 2300);
```

```
INSERT INTO angajat(cod_angajat, cod_magazin, cod_functie, nume, prenume, data_angajare) VALUES (1, 1, 1, 'Pavelescu', 'Tiberiu', '22-04-2015');
```

```
INSERT INTO angajat(cod_angajat, cod_magazin, cod_functie, nume, prenume, data_angajare) VALUES (2, 2, 1, 'Raif', 'Andreea', '27-10-2020');
```

```
INSERT INTO angajat(cod_angajat, cod_magazin, cod_functie, nume, prenume, data_angajare) VALUES (3, 3, 1, 'Marcelo', 'Juan', '10-11-2018');
```

```
INSERT INTO angajat(cod_angajat, cod_magazin, cod_functie, nume, prenume, data_angajare) VALUES (4, 4, 1, 'Chitoi', 'Dragos', '16-04-2017');
```

```
INSERT INTO angajat(cod_angajat, cod_magazin, cod_functie, nume, prenume, data_angajare) VALUES (5, 5, 1, 'Pravale', 'Diana', '25-08-2012');
```

```
INSERT INTO vehicul(nr_inmat, categorie, ani_vechime, cod_depozit) VALUES ('DJ 131 ANA', 'BE', 7, 1);
```

```
INSERT INTO vehicul(nr_inmat, categorie, ani_vechime, cod_depozit) VALUES ('IF 50 PTR', 'C', 4, 3);
```

```
INSERT INTO vehicul(nr_inmat, categorie, ani_vechime, cod_depozit) VALUES ('IF 237 MSU', 'C', 1, 3);
```

```
INSERT INTO vehicul(nr_inmat, categorie, ani_vechime, cod_depozit) VALUES ('IF 09 TMY', 'C', 3, 4);
```

```
INSERT INTO vehicul(nr_inmat, categorie, ani_vechime, cod_depozit) VALUES ('SB 672 CMN', 'C', 5, 5);
```

```
INSERT INTO vehicul(nr_inmat, categorie, ani_vechime, cod_depozit) VALUES ('CT 116 MLG', 'BE', 4, 6);
```

```
INSERT INTO vehicul(nr_inmat, categorie, ani_vechime, cod_depozit) VALUES ('AG 198 IBY', 'C', 3, 7);
```

```
INSERT INTO sofer(cod_sofer, nume, prenume, telefon, categorie_carnet, cod_depozit) VALUES (1, 'Pirvu', 'George', 0773721895, 'BE', 1);
```

```
INSERT INTO sofer(cod_sofer, nume, prenume, telefon, categorie_carnet, cod_depozit) VALUES (2, 'Lupu', 'Toma', 0743265830, 'C', 3);
```

INSERT INTO sofer(cod_sofer, nume, prenume, telefon, categorie_carnet, cod_depozit) VALUES (3, 'Lupu', 'Vasile', 0243195285, 'C', 3);

INSERT INTO sofer(cod_sofer, nume, prenume, telefon, categorie_carnet, cod_depozit) VALUES (4, 'Niculescu', 'Petra', 0218933829, 'C', 4);

INSERT INTO sofer(cod_sofer, nume, prenume, telefon, categorie_carnet, cod_depozit) VALUES (5, 'Amzu', 'Nicu', 0246942666, 'C', 5);

INSERT INTO sofer(cod_sofer, nume, prenume, telefon, categorie_carnet, cod_depozit) VALUES (6, 'Cismiu', 'Andra', 0770883829, 'BE', 6);

INSERT INTO sofer(cod_sofer, nume, prenume, telefon, categorie_carnet, cod_depozit) VALUES (7, 'Lica', 'Tanase', 0040012421691, 'C', 7);

INSERT INTO oprire(localitate) VALUES ('Bais');

INSERT INTO oprire(localitate) VALUES ('Slatina');

INSERT INTO oprire(localitate) VALUES ('Lehliu Gara');

INSERT INTO oprire(localitate) VALUES ('Fetesti');

INSERT INTO oprire(localitate) VALUES ('Medgidia');

INSERT INTO oprire(localitate) VALUES ('Ramnicu Valcea');

INSERT INTO oprire(localitate) VALUES ('Alexandria');

INSERT INTO transport(cod_transport, nr_inmat, cod_sofer, data_plecure, ora_plecure) VALUES(1, 'AG 198 IBY', 7, '25-05-2021', 700);

INSERT INTO transport(cod_transport, nr_inmat, cod_sofer, data_plecure, ora_plecure) VALUES(2, 'AG 198 IBY', 7, '06-06-2021', 800);

INSERT INTO transport(cod_transport, nr_inmat, cod_sofer, data_plecure, ora_plecure) VALUES(3, 'DJ 131 ANA', 1, '30-05-2021', 930);

INSERT INTO transport(cod_transport, nr_inmat, cod_sofer, data_plecure, ora_plecure) VALUES(4, 'DJ 131 ANA', 1, '17-06-2021', NULL);

INSERT INTO transport(cod_transport, nr_inmat, cod_sofer, data_plecure, ora_plecure) VALUES(5, 'AG 198 IBY', 7, '28-06-2021', NULL);

INSERT INTO transport(cod_transport, nr_inmat, cod_sofer, data_plecure, ora_plecure) VALUES(6, 'DJ 131 ANA', 7, '19-05-2021', NULL);

```
INSERT INTO transport(cod_transport, nr_inmat, cod_sofer, data_plecare, ora_plecare) VALUES(7, 'IF 09 TMY', 4, '28-06-2021', NULL);
```

```
INSERT INTO transport(cod_transport, nr_inmat, cod_sofer, data_plecare, ora_plecare) VALUES(8, 'SB 672 CMN', 5, '01-07-2021', 1200);
```

```
INSERT INTO transport(cod_transport, nr_inmat, cod_sofer, data_plecare, ora_plecare) VALUES(9, 'SB 672 CMN', 5, '01-07-2021', 1930);
```

```
INSERT INTO transport(cod_transport, nr_inmat, cod_sofer, data_plecare, ora_plecare) VALUES(10, 'CT 116 MLG', 6, '01-07-2021', 2230);
```

```
INSERT INTO oprire_transport(cod_transport, localitate, include_vama, durata) VALUES (1, 'Bais', 'nu', NULL);
```

```
INSERT INTO oprire_transport(cod_transport, localitate, include_vama, durata) VALUES (1, 'Slatina', 'nu', 'aprox. 30 min.');
```

```
INSERT INTO oprire_transport(cod_transport, localitate, include_vama, durata) VALUES (2, 'Slatina', 'nu', 'aprox. 1h');
```

```
INSERT INTO oprire_transport(cod_transport, localitate, include_vama, durata) VALUES (3, 'Lehliu Gara', 'nu', NULL);
```

```
INSERT INTO oprire_transport(cod_transport, localitate, include_vama, durata) VALUES (3, 'Fetesti', 'nu', 'aprox. 30 min.');
```

```
INSERT INTO oprire_transport(cod_transport, localitate, include_vama, durata) VALUES (4, 'Lehliu Gara', 'nu', 'in jur de doua ore');
```

```
INSERT INTO oprire_transport(cod_transport, localitate, include_vama, durata) VALUES (4, 'Fetesti', 'nu', 'o ora');
```

```
INSERT INTO oprire_transport(cod_transport, localitate, include_vama, durata) VALUES (4, 'Medgidia', 'nu', '2 h - 2h 30');
```

```
INSERT INTO oprire_transport(cod_transport, localitate, include_vama, durata) VALUES (5, 'Ramnicu Valcea', 'nu', '2 ore cel mult 3');
```

```
INSERT INTO oprire_transport(cod_transport, localitate, include_vama, durata) VALUES (6, 'Alexandria', 'nu', 'in jur de 1h, 1h30');
```

```
INSERT INTO livrare(cod_livrare, cod_transport, cod_magazin) VALUES(1, 1, 4);
```

```
INSERT INTO livrare(cod_livrare, cod_transport, cod_magazin) VALUES(2, 2, 4);
```

```
INSERT INTO livrare(cod_livrare, cod_transport, cod_magazin) VALUES(3, 3, 3);
```

```
INSERT INTO livrare(cod_livrare, cod_transport, cod_magazin) VALUES(4, 4, 3);
INSERT INTO livrare(cod_livrare, cod_transport, cod_magazin) VALUES(5, 5, 5);
INSERT INTO livrare(cod_livrare, cod_transport, cod_magazin) VALUES(6, 6, 1);
INSERT INTO livrare(cod_livrare, cod_transport, cod_magazin) VALUES(7, 7, 1);
INSERT INTO livrare(cod_livrare, cod_transport, cod_magazin) VALUES(8, 8, 2);
INSERT INTO livrare(cod_livrare, cod_transport, cod_magazin) VALUES(9, 9, 4);
INSERT INTO livrare(cod_livrare, cod_transport, cod_magazin) VALUES(10, 10, 1);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,
cod_livrare) VALUES(1, 'instrument', 'bas acoustic', 1600, NULL, 'stejar',
'eBanmaltez', NULL, 4, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,
cod_livrare) VALUES(2, 'instrument', 'chitara el. fretless', 4400, NULL, 'artar',
'eBanmaltez', NULL, 4, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,
cod_livrare) VALUES(3, 'instrument', 'bas acoustic', 1600, NULL, 'stejar',
'eBanmaltez', NULL, 4, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,
cod_livrare) VALUES(4, 'instrument', 'bas acoustic', 1600, NULL, 'stejar',
'eBanmaltez', NULL, 4, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,
cod_livrare) VALUES(5, 'instrument', 'chitara el. DMustang', 1600, NULL,
'stejar', 'DnM Raider', NULL, 4, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,
cod_livrare) VALUES(6, 'instrument', 'chitara el. :Haver', 1600, 3.6, 'stejar',
'Hoover', NULL, 4, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,
```

```
cod_livrare) VALUES(7, 'instrument', 'drum set full iron', 1600, 34, 'otel',  
'Jerryhmann', NULL, 4, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,  
cod_livrare) VALUES(8, 'instrument', 'bas acoustic', 1600, NULL, 'stejar',  
'eBanmaltez', NULL, 3, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,  
cod_livrare) VALUES(9, 'instrument', 'drum set full iron', 1600, 34, 'otel',  
'Jerryhmann', NULL, 2, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,  
cod_livrare) VALUES(10, 'instrument', 'chitara el. DMustang', 1600, NULL,  
'stejar', 'DnM Raider', NULL, 2, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,  
cod_livrare) VALUES(11, 'instrument', 'chitara el. JumpingW', 2590, 3, 'frasin',  
'Harvey Benington', NULL, 1, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,  
cod_livrare) VALUES(12, 'instrument', 'chitara el. JumpingW', 2590, 3, 'frasin',  
'Harvey Benington', NULL, 1, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,  
cod_livrare) VALUES(13, 'audio', 'combo chitara 400V', 5000, 22, 'fier',  
'GraySun', NULL, 3, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,  
cod_livrare) VALUES(14, 'inregistrare/audio', 'interfata RedPumpKin', 2800, 1,  
'otel', 'Mathers', NULL, 2, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,  
cod_livrare) VALUES(15, 'instrument', 'bas acoustic', 1600, NULL, 'stejar',  
'eBanmaltez', 4, NULL, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,
```

```
cod_livrare) VALUES(16, 'instrument', 'chitara el. fretless', 4400, NULL, 'artar',  
'eBanmaltez', 4, NULL, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,  
cod_livrare) VALUES(17, 'instrument', 'bas acustic', 1600, NULL, 'stejar',  
'eBanmaltez', 4, NULL, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,  
cod_livrare) VALUES(18, 'instrument', 'bas acustic', 1600, NULL, 'stejar',  
'eBanmaltez', 4, NULL, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,  
cod_livrare) VALUES(19, 'instrument', 'chitara el. DMustang', 1600, NULL,  
'stejar', 'DnM Raider', 4, NULL, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,  
cod_livrare) VALUES(20, 'instrument', 'chitara el. :Haver', 1600, 3.6, 'stejar',  
'Hoover', 4, NULL, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,  
cod_livrare) VALUES(30, 'instrument', 'drum set full iron', 1600, 34, 'otel',  
'Jerryhmann', 4, NULL, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,  
cod_livrare) VALUES(31, 'instrument', 'bas acustic', 1600, NULL, 'stejar',  
'eBanmaltez', 3, NULL, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,  
cod_livrare) VALUES(32, 'instrument', 'drum set full iron', 1600, 34, 'otel',  
'Jerryhmann', 2, NULL, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,  
cod_livrare) VALUES(33, 'instrument', 'chitara el. DMustang', 1600, NULL,  
'stejar', 'DnM Raider', 2, NULL, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,
```

```
cod_livrare) VALUES(34, 'instrument', 'chitara el. JumpingW', 2590, 3, 'frasin',  
'Harvey Benington', 1, NULL, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,  
cod_livrare) VALUES(35, 'instrument', 'chitara el. JumpingW', 2590, 3, 'frasin',  
'Harvey Benington', 1, NULL, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,  
cod_livrare) VALUES(36, 'audio', 'combo chitara 400V', 5000, 22, 'fier',  
'GraySun', 3, NULL, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_depozit, cod_magazin, cod_lutier,  
cod_livrare) VALUES(37, 'inregistrare/audio', 'interfata RedPumpKin', 2800, 1,  
'otel', 'Mathers', 2, NULL, NULL, NULL);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_lutier) VALUES(21, 'instrument', 'bas  
acustic', 1600, NULL, 'stejar', 'eBanmaltez', 3);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_lutier) VALUES(22, 'instrument',  
'chitara el. fretless', 4400, NULL, 'artar', 'eBanmaltez', 2);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_lutier) VALUES(23, 'instrument', 'bas  
acustic', 1600, NULL, 'stejar', 'eBanmaltez', 1);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_lutier) VALUES(24, 'instrument', 'bas  
acustic', 1600, NULL, 'stejar', 'eBanmaltez', 5);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_lutier) VALUES(25, 'instrument',  
'chitara el. DMustang', 1600, NULL, 'stejar', 'DnM Raider', 1);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_lutier) VALUES(26, 'instrument',  
'chitara el. :Haver', 1600, 3.6, 'stejar', 'Hoover', 6);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_lutier) VALUES(27, 'instrument', 'drum  
set full iron', 1600, 34, 'otel', 'Jerryhmann', 2);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,
material_predominant, producator, cod_lutier) VALUES(28, 'instrument', 'bas
acustic', 1600, NULL, 'stejar', 'eBanmaltez', 3);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,
material_predominant, producator, cod_lutier) VALUES(29, 'instrument', 'drum
set full iron', 1600, 34, 'otel', 'Jerryhmann', 2);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,
material_predominant, producator, cod_lutier) VALUES(38, 'instrument',
'chitara el. DMustang', 1600, NULL, 'stejar', 'DnM Raider', 1);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,
material_predominant, producator, cod_lutier) VALUES(39, 'instrument',
'chitara el. JumpingW', 2590, 3, 'frasin', 'Harvey Benington', 2);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,
material_predominant, producator, cod_lutier) VALUES(40, 'instrument',
'chitara el. JumpingW', 2590, 3, 'frasin', 'Harvey Benington', 3);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,
material_predominant, producator, cod_lutier) VALUES(41, 'audio', 'combo
chitara 400V', 5000, 22, 'fier', 'GraySun', 6);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,
material_predominant, producator, cod_lutier) VALUES(42,
'inregistrare/audio', 'interfata RedPumpKin', 2800, 1, 'otel', 'Mathers', 6);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,
material_predominant, producator, cod_livrare) VALUES(43, 'instrument', 'bas
acustic', 1600, NULL, 'stejar', 'eBanmaltez', 3);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,
material_predominant, producator, cod_livrare) VALUES(44, 'instrument',
'chitara el. fretless', 4400, NULL, 'artar', 'eBanmaltez', 2);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,
material_predominant, producator, cod_livrare) VALUES(45, 'instrument', 'bas
acustic', 1600, NULL, 'stejar', 'eBanmaltez', 1);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,
material_predominant, producator, cod_livrare) VALUES(46, 'instrument', 'bas
acustic', 1600, NULL, 'stejar', 'eBanmaltez', 5);
```



```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_livrare) VALUES(47, 'instrument',  
'chitara el. DMustang', 1600, NULL, 'stejar', 'DnM Raider', 1);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_livrare) VALUES(48, 'instrument',  
'chitara el. :Haver', 1600, 3.6, 'stejar', 'Hoover', 6);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_livrare) VALUES(49, 'instrument',  
'drum set full iron', 1600, 34, 'otel', 'Jerryhmann', 2);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_livrare) VALUES(50, 'instrument', 'bas  
acustic', 1600, NULL, 'stejar', 'eBanmaltez', 3);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_livrare) VALUES(51, 'instrument',  
'drum set full iron', 1600, 34, 'otel', 'Jerryhmann', 2);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_livrare) VALUES(52, 'instrument',  
'chitara el. DMustang', 1600, NULL, 'stejar', 'DnM Raider', 1);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_livrare) VALUES(53, 'instrument',  
'chitara el. JumpingW', 2590, 3, 'frasin', 'Harvey Benington', 2);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_livrare) VALUES(54, 'instrument',  
'chitara el. JumpingW', 2590, 3, 'frasin', 'Harvey Benington', 3);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_livrare) VALUES(55, 'audio', 'combo  
chitara 400V', 5000, 22, 'fier', 'GraySun', 6);
```

```
INSERT INTO produs(cod_produș, tip, nume, pret, greutate,  
material_predominant, producator, cod_livrare) VALUES(56,  
'inregistrare/audio', 'interfata RedPumpKin', 2800, 1, 'otel', 'Mathers', 6);
```

```
COMMIT;
```

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

Commit complete.

PRODUS:

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	COD_PRODUS		NUMBER(10,0)	No		(null)		1		(null)
2	COD_DEPOZIT		NUMBER(10,0)	Yes		NULL		2		(null)
3	COD_MAGAZIN		NUMBER(10,0)	Yes		NULL		3		(null)
4	COD_LUTIER		NUMBER(10,0)	Yes		NULL		4		(null)
5	COD_LIVRARE		NUMBER(10,0)	Yes		NULL		5		(null)
6	TIP		VARCHAR2(20 BYTE)	Yes		(null)		6		(null)
7	NUME		VARCHAR2(20 BYTE)	Yes		(null)		7		(null)
8	PRET		NUMBER(10,0)	Yes		(null)		8		(null)
9	GREUTATE		NUMBER(10,0)	Yes		(null)		9		(null)
10	MATERIAL_PREDOMINANT		VARCHAR2(20 BYTE)	Yes		(null)		10		(null)
11	PRODUCATOR		VARCHAR2(20 BYTE)	Yes		(null)		11		(null)

	COD_PRODUS	COD_DEPOZIT	COD_MAGAZIN	COD_LUTIER	COD_LIVRARE	TIP	NUME	PRET	GREUTATE	MATERIAL_PREDOMINANT	PF
1	1	(null)	4	(null)	(null)	instrument	bas acustic	1600	(null)	stejar	eBa
2	2	(null)	4	(null)	(null)	instrument	chitara el. fretless	4400	(null)	artar	eBa
3	3	(null)	4	(null)	(null)	instrument	bas acustic	1600	(null)	stejar	eBa
4	4	(null)	4	(null)	(null)	instrument	bas acustic	1600	(null)	stejar	eBa
5	5	(null)	4	(null)	(null)	instrument	chitara el. DMustang	1600	(null)	stejar	Dnl
6	6	(null)	4	(null)	(null)	instrument	chitara el. :Haver	1600	4	stejar	Hoc
7	7	(null)	4	(null)	(null)	instrument	drum set full iron	1600	34	otel	Jern
8	8	(null)	3	(null)	(null)	instrument	bas acustic	1600	(null)	stejar	eBa
9	9	(null)	2	(null)	(null)	instrument	drum set full iron	1600	34	otel	Jern
10	10	(null)	2	(null)	(null)	instrument	chitara el. DMustang	1600	(null)	stejar	Dnl
11	11	(null)	1	(null)	(null)	instrument	chitara el. JumpingW	2590	3	frasin	Har
12	12	(null)	1	(null)	(null)	instrument	chitara el. JumpingW	2590	3	frasin	Har
13	13	(null)	3	(null)	(null)	audio	combo chitara 400V	5000	22	fier	Gra
14	14	(null)	2	(null)	(null)	inregistrare/audio	interfata RedPumpKin	2800	1	otel	Ma
15	15	4	(null)	(null)	(null)	instrument	bas acustic	1600	(null)	stejar	eBa
16	16	4	(null)	(null)	(null)	instrument	chitara el. fretless	4400	(null)	artar	eBa
17	17	4	(null)	(null)	(null)	instrument	bas acustic	1600	(null)	stejar	eBa
18	18	4	(null)	(null)	(null)	instrument	bas acustic	1600	(null)	stejar	eBa
19	19	4	(null)	(null)	(null)	instrument	chitara el. DMustang	1600	(null)	stejar	Dnl
20	20	4	(null)	(null)	(null)	instrument	chitara el. :Haver	1600	4	stejar	Hoc
21	30	4	(null)	(null)	(null)	instrument	drum set full iron	1600	34	otel	Jern
22	31	3	(null)	(null)	(null)	instrument	bas acustic	1600	(null)	stejar	eBa
23	32	2	(null)	(null)	(null)	instrument	drum set full iron	1600	34	otel	Jern
24	33	2	(null)	(null)	(null)	instrument	chitara el. DMustang	1600	(null)	stejar	Dnl
25	34	1	(null)	(null)	(null)	instrument	chitara el. JumpingW	2590	3	frasin	Har
26	35	1	(null)	(null)	(null)	instrument	chitara el. JumpingW	2590	3	frasin	Har
27	36	3	(null)	(null)	(null)	audio	combo chitara 400V	5000	22	fier	Gra
28	37	2	(null)	(null)	(null)	inregistrare/audio	interfata RedPumpKin	2800	1	otel	Ma
29	21	(null)	(null)	3	(null)	instrument	bas acustic	1600	(null)	stejar	eBa
30	22	(null)	(null)	2	(null)	instrument	chitara el. fretless	4400	(null)	artar	eBa
31	23	(null)	(null)	1	(null)	instrument	bas acustic	1600	(null)	stejar	eBa
32	24	(null)	(null)	5	(null)	instrument	bas acustic	1600	(null)	stejar	eBa
33	25	(null)	(null)	1	(null)	instrument	chitara el. DMustang	1600	(null)	stejar	Dnl
34	26	(null)	(null)	6	(null)	instrument	chitara el. :Haver	1600	4	stejar	Hoc
35	27	(null)	(null)	2	(null)	instrument	drum set full iron	1600	34	otel	Jern
36	28	(null)	(null)	3	(null)	instrument	bas acustic	1600	(null)	stejar	eBa
37	29	(null)	(null)	2	(null)	instrument	drum set full iron	1600	34	otel	Jern
38	38	(null)	(null)	1	(null)	instrument	chitara el. DMustang	1600	(null)	stejar	Dnl
39	39	(null)	(null)	2	(null)	instrument	chitara el. JumpingW	2590	3	frasin	Har
40	40	(null)	(null)	3	(null)	instrument	chitara el. JumpingW	2590	3	frasin	Har
41	41	(null)	(null)	6	(null)	audio	combo chitara 400V	5000	22	fier	Gra







LIVRARE:

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	COD_LIVRARE	NUMBER(10,0)	No	(null)	1 (null)	
2	COD_TRANSPORT	NUMBER(10,0)	No	(null)	2 (null)	
3	COD_MAGAZIN	NUMBER(10,0)	No	(null)	3 (null)	
4	STATUS_LIVRARE	VARCHAR2(20 BYTE)	No	'pe drum'	4 (null)	

	COD_LIVRARE	COD_TRANSPORT	COD_MAGAZIN	STATUS_LIVRARE
1	1	1		4 pe drum
2	2	2		4 pe drum
3	3	3		3 pe drum
4	4	4		3 pe drum
5	5	5		5 pe drum
6	6	6		1 pe drum
7	7	7		1 pe drum
8	8	8		2 pe drum
9	9	9		4 pe drum
10	10	10		1 pe drum

TRANSPORT:

Columns	Data	Model	Constraints	Grants	Statistics	Triggers	Flashback	Dependencies	Details	Partitions	Indexes	SQL
Actions...												
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS						
1	COD_TRANSPORT	NUMBER(10,0)	No	(null)	1 (null)							
2	NR_INMAT	VARCHAR2(20 BYTE)	No	(null)	2 (null)							
3	COD_SOFER	NUMBER(10,0)	No	(null)	3 (null)							
4	DATA_PLECARE	DATE	Yes	(null)	4 (null)							
5	ORA_PLECARE	NUMBER(4,0)	Yes	(null)	5 (null)							

Columns	Data	Model	Constraints	Grants	Statistics	Triggers	Flashback	Dependencies	Details	Partitions
<div> Sort.. Filter: <input type="text"/></div>										
	↕ COD_TRANSPORT	↕ NR_INMAT	↕ COD_SOFER	↕ DATA_PLECARE	↕ ORA_PLECARE					
1	1	AG 198 IBY	7	25-05-2021	700					
2	2	AG 198 IBY	7	06-06-2021	800					
3	3	DJ 131 ANA	1	30-05-2021	930					
4	4	DJ 131 ANA	1	17-06-2021	(null)					
5	5	AG 198 IBY	7	28-06-2021	(null)					
6	6	DJ 131 ANA	7	19-05-2021	(null)					
7	7	IF 09 TMY	4	28-06-2021	(null)					
8	8	SB 672 CMN	5	01-07-2021	1200					
9	9	SB 672 CMN	5	01-07-2021	1930					
10	10	CT 116 MLG	6	01-07-2021	2230					

OPRIRE:

Columns	Data	Model	Constraints	Grants	Statistics	Triggers	Flashback	Dependencies	Details	Partitions	Indexes	SQL
Actions...												
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS						
1	LOCALITATE	VARCHAR2(20 BYTE)	No	(null)	1 (null)							

Columns	Data	Model	Constraints
Sort..			
	LOCALITATE		
1	Bais		
2	Slatina		
3	Lehliu Gara		
4	Fetesti		
5	Medgidia		
6	Ramnicu Valcea		
7	Alexandria		

OPRIRE_TRANSPORT:

Columns	Data	Model	Constraints	Grants	Statistics	Triggers	Flashback	Dependencies	Details	Partitions	Indexes	SQL
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS						
1	COD_TRANSPORT	NUMBER(10,0)	No	(null)	1	(null)						
2	LOCALITATE	VARCHAR2(20 BYTE)	No	(null)	2	(null)						
3	INCLUDE_VAMA	VARCHAR2(2 BYTE)	Yes	(null)	3	(null)						
4	DURATA	VARCHAR2(20 BYTE)	Yes	(null)	4	(null)						

Columns	Data	Model	Constraints	Grants	Statistics	Triggers	Flashback	Dependencies	Details
	COD_TRANSPORT	LOCALITATE	INCLUDE_VAMA	DURATA					
1	1	Bais	nu	(null)					
2	1	Slatina	nu	aprox. 30 min.					
3	2	Slatina	nu	aprox. 1h					
4	3	Lehliu Gara	nu	(null)					
5	3	Fetesti	nu	aprox. 30 min.					
6	4	Lehliu Gara	nu	in jur de doua ore					
7	4	Fetesti	nu	o ora					
8	4	Medgidia	nu	2 h - 2h 30					
9	5	Ramnicu Valcea	nu	2 ore cel mult 3					
10	6	Alexandria	nu	in jur de 1h, 1h30					

6.

-- ex 6: Creati o procedura care primeste ca parametru de intrare un tabel indexat cu livrari si are parametrii de iesire doua tabele imbricate in care vor fi introduse codurile vehiculelor, respectiv soferilor care se ocupa de transporturile incheiate (ale caror ultime livrari au sosit acum). Modificati si livrarile de actualizat folosind procedura sosireLivrare (de la ex 7).

```
CREATE OR REPLACE TYPE varrayCoduriLivrari IS VARRAY(100) OF NUMBER(10);
```

```
/
```

```
CREATE OR REPLACE TYPE tblCoduriVehicule IS TABLE OF VARCHAR2(20);
```

```
/
```

```
CREATE OR REPLACE TYPE tblCoduriSoferi IS TABLE OF VARCHAR2(10);
```

```
/
```

```
CREATE OR REPLACE PROCEDURE elibereazaVehiculeSoferi(nrLivrari IN
NUMBER, coduriLivrari IN varrayCoduriLivrari, coduriVehicule OUT
tblCoduriVehicule, coduriSoferi OUT tblCoduriSoferi) AS
```

```

codTransport NUMBER;
BEGIN
FOR i IN coduriLivrari.FIRST..coduriLivrari.LAST LOOP
    sosireLivrare(coduriLivrari(i));      -- se incheie livrarea crt

    SELECT cod_transport
    INTO codTransport
    FROM livrare
    WHERE cod_livrare = coduriLivrari(i);

    SELECT DISTINCT v.nr_inmat
    BULK COLLECT INTO coduriVehicule
    FROM vehicul v, transport t
    WHERE t.cod_transport = codTransport -- transportul ce corespunde
livrarii crt
        AND t.nr_inmat = v.nr_inmat      -- join intre transport si vehicul
        AND t.cod_transport NOT IN      -- transport care sa nu fie in
mulimea transporturilor cu livrari neincheiate
        (SELECT t1.cod_transport
        FROM transport t1
        WHERE
            (SELECT COUNT(l.cod_livrare)
            FROM livrare l
            WHERE l.cod_transport = t.cod_transport      -- join intre livrare si
transport
            AND LOWER(l.status_livrare) NOT LIKE 'sosita' -- livrare neincheiata
            ) > 0
        );

```

```

SELECT DISTINCT s.cod_sofer
BULK COLLECT INTO coduriSoferi
FROM sofer s, transport t
WHERE t.cod_transport = codTransport -- transportul ce corespunde
livrarii crt
AND t.cod_sofer = s.cod_sofer          -- join intre transport si sofer
AND t.cod_transport NOT IN             -- transport care sa nu fie in
mulimea transporturilor cu livrari neincheiate
(SELECT t1.cod_transport
FROM transport t1
WHERE
(SELECT COUNT(l.cod_livrare)
FROM livrare l
WHERE l.cod_transport = t.cod_transport      -- join intre livrare si
transport
AND LOWER(l.status_livrare) NOT LIKE 'sosita' -- livrare neincheiata
) > 0
);
END LOOP;
END;
/

```

```

SELECT * FROM livrare;
SELECT * FROM transport;
SELECT cod_produs produs, cod_livrare livrare, cod_magazin magazin FROM
produs;

```

```

DECLARE
coduriLivrari varrayCoduriLivrari := varrayCoduriLivrari();
coduriVehicule tblCoduriVehicule := tblCoduriVehicule();

```

```
coduriSoferi tblCoduriSoferi := tblCoduriSoferi();
```

```
BEGIN
```

```
SELECT cod_livrare
```

```
BULK COLLECT INTO coduriLivrari
```

```
FROM livrare
```

```
WHERE cod_livrare IN (1,2,3, 5,6, 7);
```

```
elibereazaVehiculeSoferi(4, coduriLivrari, coduriVehicule, coduriSoferi);
```

```
DBMS_OUTPUT.PUT('Nr inmatriculare vehicule eliberate: ');
```

```
FOR i IN coduriVehicule.FIRST..coduriVehicule.LAST LOOP
```

```
    DBMS_OUTPUT.PUT(coduriVehicule(i) || ' ');
```

```
END LOOP;
```

```
DBMS_OUTPUT.NEW_LINE;
```

```
DBMS_OUTPUT.PUT('Coduri soferi eliberati: ');
```

```
FOR i IN coduriSoferi.FIRST..coduriSoferi.LAST LOOP
```

```
    DBMS_OUTPUT.PUT(coduriSoferi(i) || ' ');
```

```
END LOOP;
```

```
DBMS_OUTPUT.NEW_LINE;
```


```
END;
```

```
/
```

```
SELECT * FROM livrare;
```

```
SELECT cod_produs produs, cod_livrare livrare, cod_magazin magazin FROM  
produs;
```


Dbms Output x

 Buffer Size: 20000

MusicShp x

Produsul cu codul 45 a fost livrat la magazinul cu codul 4
Produsul cu codul 47 a fost livrat la magazinul cu codul 4
Produsul cu codul 52 a fost livrat la magazinul cu codul 4
Livrarea cu codul 1 s-a incheiat
Produsul cu codul 44 a fost livrat la magazinul cu codul 4
Produsul cu codul 49 a fost livrat la magazinul cu codul 4
Produsul cu codul 51 a fost livrat la magazinul cu codul 4
Produsul cu codul 53 a fost livrat la magazinul cu codul 4
Livrarea cu codul 2 s-a incheiat
Produsul cu codul 43 a fost livrat la magazinul cu codul 3
Produsul cu codul 50 a fost livrat la magazinul cu codul 3
Produsul cu codul 54 a fost livrat la magazinul cu codul 3
Livrarea cu codul 3 s-a incheiat
Produsul cu codul 46 a fost livrat la magazinul cu codul 5
Livrarea cu codul 5 s-a incheiat
Produsul cu codul 48 a fost livrat la magazinul cu codul 1
Produsul cu codul 55 a fost livrat la magazinul cu codul 1
Produsul cu codul 56 a fost livrat la magazinul cu codul 1
Livrarea cu codul 6 s-a incheiat
Livrarea cu codul 7 s-a incheiat
Nr inmatriculare vehicule eliberate: IF 09 TMY
Coduri soferi eliberati: 4

Type VARRAYCODURILIVRARI compiled

Type TBLCODURIVEHICULE compiled

Type TBLCODURISOFERI compiled

Procedure ELIBEREAZAVEHICULESOFERI compiled

COD_LIVRARE COD_TRANSPORT COD_MAGAZIN STATUS_LIVRARE

1	1	4 pe drum
2	2	4 pe drum
3	3	3 pe drum
4	4	3 pe drum
5	5	5 pe drum
6	6	1 pe drum
7	7	1 pe drum
8	8	2 pe drum
9	9	4 pe drum
10	10	1 pe drum

10 rows selected.

Script Output x

Task completed in 0,027 seconds

COD_TRANSPORT NR_INMAT	COD_SOFER DATA_PLECA	ORA_PLECARE
1 AG 198 IBY	7 25-05-2021	700
2 AG 198 IBY	7 06-06-2021	800
3 DJ 131 ANA	1 30-05-2021	930
4 DJ 131 ANA	1 17-06-2021	
5 AG 198 IBY	7 28-06-2021	
6 DJ 131 ANA	7 19-05-2021	
7 IF 09 TMY	4 28-06-2021	
8 SB 672 CMN	5 01-07-2021	1200
9 SB 672 CMN	5 01-07-2021	1930
10 CT 116 MLG	6 01-07-2021	2230

10 rows selected.

PRODUS	LIVRARE	MAGAZIN
1	4	
2	4	
3	4	
4	4	
5	4	
6	4	
7	4	
8	3	
9	2	
10	2	
11	1	
12	1	
13	3	
14	2	
15		
16		
17		
18		
19		
20		
30		
31		
32		
33		
34		
35		
36		
37		
21		

22
23
24
25

PRODUS	LIVRARE	MAGAZIN
--------	---------	---------

26		
27		
28		
29		
38		
39		
40		
41		
42		
43	3	
44	2	

PRODUS	LIVRARE	MAGAZIN
--------	---------	---------

45	1	
46	5	
47	1	
48	6	
49	2	
50	3	
51	2	
52	1	
53	2	
54	3	
55	6	

PRODUS	LIVRARE	MAGAZIN
--------	---------	---------

56	6	
----	---	--

Script Output x

 | Task completed in 0,027 seconds

56 rows selected.

PL/SQL procedure successfully completed.

COD_LIVRARE COD_TRANSPORT COD_MAGAZIN STATUS_LIVRARE

1	1	4	sosita
2	2	4	sosita
3	3	3	sosita
4	4	3	pe drum
5	5	5	sosita
6	6	1	sosita
7	7	1	sosita
8	8	2	pe drum
9	9	4	pe drum
10	10	1	pe drum






10 rows selected.

PRODUS	LIVRARE	MAGAZIN
1	4	
2	4	
3	4	
4	4	
5	4	
6	4	
7	4	
8	3	
9	2	
10	2	
11	1	

PRODUS	LIVRARE	MAGAZIN
12	1	
13	3	
14	2	
15		
16		
17		
18		
19		
20		
30		
31		

PRODUS	LIVRARE	MAGAZIN
32		
33		
34		
35		
36		
37		
21		

Script Output x

 | Task completed in 0,027s

22

23

24

25

PRODUS	LIVRARE	MAGAZIN
26		
27		
28		
29		
38		
39		
40		
41		
42		
43	3	
44	4	

PRODUS	LIVRARE	MAGAZIN
45	4	
46	5	
47	4	
48	1	
49	4	
50	3	
51	4	
52	4	
53	4	
54	3	
55	1	

PRODUS	LIVRARE	MAGAZIN
56	1	

56 rows selected.

-- ex 7: Creati o procedură care primește ca parametru de intrare codul unei livrări și îi setează status_livrare ca 'sosita', iar pentru fiecare produs care are cod_livrare egal cu parametrul dat, setează cod_livrare cu NULL și cod_magazin cu codul magazinului ce corespunde livrării în cauză. Se va folosi un cursor pentru a reține produsele ce trebuie actualizate.

```
CREATE OR REPLACE PROCEDURE sosireLivrare(codLivrare IN NUMBER) AS
```

```
    TYPE tabelCoduriProduse IS TABLE OF NUMBER;
```

```
    coduriProduse  tabelCoduriProduse :=tabelCoduriProduse();
```

```
    codMagazin  NUMBER;
```

```
    codProdusCrt  NUMBER;
```

```
    CURSOR c IS
```

```
        SELECT cod_produs
```

```
        BULK COLLECT INTO coduriProduse
```

```
        FROM produs
```

```
        WHERE cod_livrare = codLivrare;
```

```
BEGIN
```

```
    IF (codLivrare IS NULL)
```

```
        THEN
```

```
            DBMS_OUTPUT.PUT_LINE('sosireLivrare: Parametrul dat nu poate fi NULL!');
```

```
            RETURN;
```

```
        END IF;
```

```
        SELECT cod_magazin
```

```
        INTO codMagazin
```

```
        FROM livrare
```

```
        WHERE cod_livrare = codLivrare;
```

```
UPDATE livrare
SET status_livrare = 'sosita'
WHERE cod_livrare = codLivrare;
```

```
OPEN c;
LOOP
    FETCH c INTO codProdusCrt;
    EXIT WHEN c%NOTFOUND;
```

```
UPDATE produs
SET cod_livrare = NULL, cod_magazin = codMagazin
WHERE cod_produs = codProdusCrt;

DBMS_OUTPUT.PUT_LINE('Produsul cu codul ' || codProdusCrt || ' a fost
livrat la magazinul cu codul ' || codMagazin );

END LOOP;
CLOSE c;
```

```
DBMS_OUTPUT.PUT_LINE('Livrarea cu codul ' || codLivrare || ' s-a incheiat');
```

```
EXCEPTION

    WHEN NO_DATA_FOUND THEN

        DBMS_OUTPUT.PUT_LINE('sosireLivrare: Nu exista o livrare cu codul ' ||
codLivrare || '!');

END;

/
```

```
SELECT cod_produs produs, NVL(cod_magazin, 0) magazin, NVL(cod_livrare, 0)
livrare, nume FROM produs;

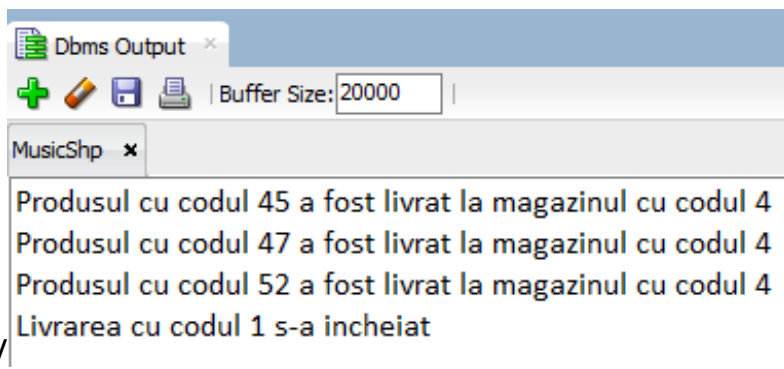
SELECT * FROM livrare;
```

```
EXECUTE sosireLivrare(1);
```

```
/
```

```
SELECT cod_produș produs, NVL(cod_magazin, 0) magazin, NVL(cod_livrare, 0)  
livrare, nume FROM produs;
```

```
SELECT * FROM LIVRARE;
```



Procedure SOSIRELIVRARE compiled

PRODUS	MAGAZIN	LIVRARE NUME
--------	---------	--------------

1	4	0 bas acustic
2	4	0 chitara el. fretless
3	4	0 bas acustic
4	4	0 bas acustic
5	4	0 chitara el. DMustang
6	4	0 chitara el. :Haver
7	4	0 drum set full iron
8	3	0 bas acustic
9	2	0 drum set full iron
10	2	0 chitara el. DMustang
11	1	0 chitara el. JumpingW

PRODUS	MAGAZIN	LIVRARE NUME
--------	---------	--------------

12	1	0 chitara el. JumpingW
13	3	0 combo chitara 400V
14	2	0 interfata RedPumpKin
15	0	0 bas acustic
16	0	0 chitara el. fretless
17	0	0 bas acustic
18	0	0 bas acustic
19	0	0 chitara el. DMustang
20	0	0 chitara el. :Haver
30	0	0 drum set full iron
31	0	0 bas acustic

PRODUS	MAGAZIN	LIVRARE NUME
--------	---------	--------------

32	0	0 drum set full iron
33	0	0 chitara el. DMustang
34	0	0 chitara el. JumpingW

34	0	0 chitara el. JumpingW
35	0	0 chitara el. JumpingW
36	0	0 combo chitara 400V
37	0	0 interfata RedPumpKin
21	0	0 bas acoustic
22	0	0 chitara el. fretless
23	0	0 bas acoustic
24	0	0 bas acoustic
25	0	0 chitara el. DMustang

PRODUS	MAGAZIN	LIVRARE NUME
--------	---------	--------------

26	0	0 chitara el. :Haver
27	0	0 drum set full iron
28	0	0 bas acoustic
29	0	0 drum set full iron
38	0	0 chitara el. DMustang
39	0	0 chitara el. JumpingW
40	0	0 chitara el. JumpingW
41	0	0 combo chitara 400V
42	0	0 interfata RedPumpKin
43	0	3 bas acoustic
44	0	2 chitara el. fretless

PRODUS	MAGAZIN	LIVRARE NUME
--------	---------	--------------

45	0	1 bas acoustic
46	0	5 bas acoustic
47	0	1 chitara el. DMustang
48	0	6 chitara el. :Haver
49	0	2 drum set full iron
50	0	3 bas acoustic
51	0	2 drum set full iron
52	0	1 chitara el. DMustang
53	0	2 chitara el. JumpingW
54	0	3 chitara el. JumpingW
55	0	6 combo chitara 400V

55 0 6 combo chitara 400V

PRODUS	MAGAZIN	LIVRARE NUME
--------	---------	--------------

56	0	6 interfata RedPumpKin
----	---	------------------------

56 rows selected.

COD_LIVRARE	COD_TRANSPORT	COD_MAGAZIN	STATUS_LIVRARE
-------------	---------------	-------------	----------------

1	1	4 pe drum
2	2	4 pe drum
3	3	3 pe drum
4	4	3 pe drum
5	5	5 pe drum
6	6	1 pe drum
7	7	1 pe drum
8	8	2 pe drum
9	9	4 pe drum
10	10	1 pe drum

10 rows selected.

PL/SQL procedure successfully completed.

PRODUS	MAGAZIN	LIVRARE NUME
--------	---------	--------------

1	4	0 bas acoustic
2	4	0 chitara el. fretless
3	4	0 bas acoustic
4	4	0 bas acoustic
5	4	0 chitara el. DMustang
6	4	0 chitara el. :Haver
7	4	0 drum set full iron

Script Output x

Task completed in 0,038 seconds

7	4	0 drum set full iron
8	3	0 bas acustic
9	2	0 drum set full iron
10	2	0 chitara el. DMustang
11	1	0 chitara el. JumpingW

PRODUS	MAGAZIN	LIVRARE NUME
--------	---------	--------------

12	1	0 chitara el. JumpingW
13	3	0 combo chitara 400V
14	2	0 interfata RedPumpKin
15	0	0 bas acustic
16	0	0 chitara el. fretless
17	0	0 bas acustic
18	0	0 bas acustic
19	0	0 chitara el. DMustang
20	0	0 chitara el. :Haver
30	0	0 drum set full iron
31	0	0 bas acustic

PRODUS	MAGAZIN	LIVRARE NUME
--------	---------	--------------

32	0	0 drum set full iron
33	0	0 chitara el. DMustang
34	0	0 chitara el. JumpingW
35	0	0 chitara el. JumpingW
36	0	0 combo chitara 400V
37	0	0 interfata RedPumpKin
21	0	0 bas acustic
22	0	0 chitara el. fretless
23	0	0 bas acustic
24	0	0 bas acustic
25	0	0 chitara el. DMustang

PRODUS	MAGAZIN	LIVRARE NUME
--------	---------	--------------

26	0	0 chitara el. :Haver
----	---	----------------------

PRODUS	MAGAZIN	LIVRARE NUME
26	0	0 chitara el. :Haver
27	0	0 drum set full iron
28	0	0 bas acoustic
29	0	0 drum set full iron
38	0	0 chitara el. DMustang
39	0	0 chitara el. JumpingW
40	0	0 chitara el. JumpingW
41	0	0 combo chitara 400V
42	0	0 interfata RedPumpKin
43	0	3 bas acoustic
44	0	2 chitara el. fretless

PRODUS	MAGAZIN	LIVRARE NUME
45	4	0 bas acoustic
46	0	5 bas acoustic
47	4	0 chitara el. DMustang
48	0	6 chitara el. :Haver
49	0	2 drum set full iron
50	0	3 bas acoustic
51	0	2 drum set full iron
52	4	0 chitara el. DMustang
53	0	2 chitara el. JumpingW
54	0	3 chitara el. JumpingW
55	0	6 combo chitara 400V

PRODUS	MAGAZIN	LIVRARE NUME
56	0	6 interfata RedPumpKin

56 rows selected.

COD_LIVRARE	COD_TRANSPORT	COD_MAGAZIN	STATUS_LIVRARE
1	1	4	acustic

COD_LIVRARE	COD_TRANSPORT	COD_MAGAZIN	STATUS_LIVRARE
1	1	4	sosita
2	2	4	pe drum
3	3	3	pe drum
4	4	3	pe drum
5	5	5	pe drum
6	6	1	pe drum
7	7	1	pe drum
8	8	2	pe drum
9	9	4	pe drum
10	10	1	pe drum

10 rows selected.

8.

-- ex 8: Creati o functie ce primeste ca parametru de intrare codul unui produs si returneaza codul unui lutier aflat in acelasi judet in care se gaseste magazinul in care sta produsul dat. Daca produsul nu se afla intr-un magazin sau daca in judetul respectiv nu este niciun lutier, functia va returna -1.

```
CREATE OR REPLACE FUNCTION lutierLocal(codProdus IN NUMBER)
```

```
RETURN lutier.cod_lutier%TYPE AS
```

```
TYPE coduriLutieri IS TABLE OF lutier.cod_lutier%TYPE;
```

```
judetLocatie locatie.judet%TYPE;
```

```
codLutier lutier.cod_lutier%TYPE;
```

```
tblCoduriLutieri coduriLutieri := coduriLutieri();
```

```
BEGIN
```

```
SELECT l.judet
```

```
INTO judetLocatie
```

```
FROM produs p, magazin m, locatie l
```

```
WHERE p.cod_produs = codProdus -- alegem produsul cu codul dat
```

```
AND m.cod_magazin = p.cod_magazin -- join intre magazin si produs
AND l.cod_locatie = m.cod_locatie;    -- join intre locatie si magazin
```

```
SELECT lut.cod_lutier
```

```
INTO codLutier
```

```
FROM lutier lut, locatie loc
```

```
WHERE loc.judet = judetLocatie      -- judetul gasit
```

```
AND lut.cod_locatie = loc.cod_locatie; -- join intre lutier si locatie
```

```
RETURN codLutier;
```

```
EXCEPTION
```

```
    WHEN NO_DATA_FOUND THEN
```

```
        DBMS_OUTPUT.PUT_LINE('lutierLocal: Atentie! In produs, cod_magazin
nu trebuie sa fie null. Posibil sa nu fie un lutier in judetul magazinului.');
```

```
        RETURN -1;
```

```
    WHEN TOO_MANY_ROWS THEN
```

```
        DBMS_OUTPUT.PUT_LINE('lutierLocal: Mai multi lutieri sunt la
dispozitie. A fost returnat primul gasit.');
```

```
        SELECT lut.cod_lutier
```

```
        BULK COLLECT INTO tblCoduriLutieri
```

```
        FROM lutier lut, locatie loc
```

```
        WHERE loc.judet = judetLocatie      -- judetul gasit
```

```
        AND lut.cod_locatie = loc.cod_locatie; -- join intre lutier si locatie
```

```
        codLutier := tblCoduriLutieri(1);
```

```
        RETURN codLutier;
```

END lutierLocal;

/

```
SELECT p.cod_produs produs, m.cod_magazin magazin, l.judet judet
FROM produs p, magazin m, locatie l
WHERE p.cod_magazin IS NOT NULL
AND m.cod_magazin = p.cod_magazin
AND l.cod_locatie = m.cod_locatie;
```

```
SELECT lut.cod_lutier lutier, loc.judet judet
FROM lutier lut, locatie loc
WHERE loc.cod_locatie = lut.cod_locatie;
```

DECLARE

TYPE coduriLutieri IS VARRAY(4) OF lutier.cod_lutier%TYPE;

varrayCoduriLutieri coduriLutieri:=coduriLutieri();

codLutierLocal lutier.cod_lutier%TYPE;

BEGIN

varrayCoduriLutieri.extend();

varrayCoduriLutieri.extend();

varrayCoduriLutieri.extend();

varrayCoduriLutieri.extend();

varrayCoduriLutieri(1):= 10; -- un lutier in judet

varrayCoduriLutieri(2):= 11; -- mai multi lutieri in judet (too many rows)

varrayCoduriLutieri(3):= 13; -- niciun lutier in judet (no data found)

varrayCoduriLutieri(4):= 15; -- cod magazin null (no data found)

```
FOR i IN varrayCoduriLutieri.FIRST..varrayCoduriLutieri.LAST LOOP
    codLutierLocal := lutierLocal(varrayCoduriLutieri(i));

    IF codLutierLocal > -1
        THEN
            DBMS_OUTPUT.PUT_LINE('Pentru produsul cu codul ' ||
varrayCoduriLutieri(i) || ', codul unui lutier din judet: ' || codLutierLocal);
        ELSE
            DBMS_OUTPUT.PUT_LINE('Pentru produsul cu codul ' ||
varrayCoduriLutieri(i) || ', eroare intampinata! Vedeti detalii mai sus');
        END IF;
        DBMS_OUTPUT.NEW_LINE();
    END LOOP;
END;
```

Function LUTIERLOCAL compiled

PRODUS	MAGAZIN JUDET
--------	---------------

1	4 Arges
2	4 Arges
3	4 Arges
4	4 Arges
5	4 Arges
6	4 Arges
7	4 Arges
8	3 Constanta
9	2 Sibiu
10	2 Sibiu
11	1 Ilfov

PRODUS	MAGAZIN JUDET
--------	---------------

PRODUS	MAGAZIN JUDET
--------	---------------

12	1 Ilfov
13	3 Constanta
14	2 Sibiu

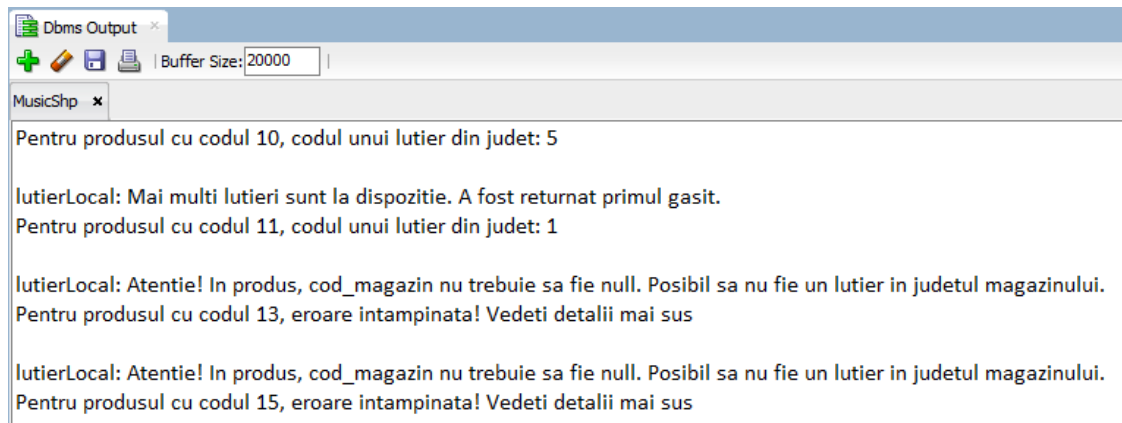
14 rows selected.

LUTIER JUDET

1 Ilfov
6 Ilfov
2 Ilfov
3 Ilfov
4 Ilfov
5 Sibiu

6 rows selected.

PL/SQL procedure successfully completed.



9.

-- ex 9: Creati o procedura care primeste un parametru de intrare pentru codul unui produs si are un parametru de iesire pentru numele depozitului in care se afla acesta si doi parametri de iesire pentru numarul de inmatriculare al unui vehicul, respectiv codul unui sofer, ale caror depozite coincid cu cel al depozitului in care se afla produsul.

Vehiculul si soferul indicati nu vor avea un transport care sa aiba livrari neinchiate.

Daca produsul dat nu exista sau nu are un cod_depozit nenul sau daca nu se gaseste o pereche disponibila de vehicul si sofer, parametrii de iesire vor fi setati toti cu -1.

```
CREATE OR REPLACE PROCEDURE soferiVehiculeDisponibile (codProdus IN
produs.cod_produs%TYPE, numeDepozit OUT depozit.nume%TYPE,
codVehiculDisponibil OUT vehicul.nr_inmat%TYPE, codSoferDisponibil OUT
sofer.cod_sofer%TYPE) AS
```

```
    TYPE perecheVehiculSofer IS RECORD
```

```
    (nrInmat    vehicul.nr_inmat%TYPE,
```

```
    codSofer    sofer.cod_sofer%TYPE
```

```
    );
```

```
    TYPE tabelPerecheVehiculSofer IS TABLE OF perecheVehiculSofer;
```

```
    perechiDisponibile    tabelPerecheVehiculSofer := tabelPerecheVehiculSofer();
```

```
BEGIN
```

```

WITH transporturiNeincheiate AS
    (SELECT t.cod_transport codT
    FROM transport t
    WHERE
        (SELECT COUNT(l.cod_livrare)
        FROM livrare l
        WHERE l.cod_transport = t.cod_transport      -- join intre livrare si
transport
        AND LOWER(l.status_livrare) NOT LIKE 'sosita' -- livrare neincheiata
        ) > 0
    )
SELECT DISTINCT d.nume, v.nr_inmat, s.cod_sofer
INTO numeDepozit, codVehiculDisponibil, codSoferDisponibil
FROM produs p, depozit d, vehicul v, sofer s
WHERE p.cod_produs = codProdus  -- produsul cautat
AND d.cod_depozit = p.cod_depozit -- join intre depozit si produs
AND v.cod_depozit = d.cod_depozit -- join intre vehicul si depozit
AND s.cod_depozit = d.cod_depozit -- join intre sofer si depozit
AND v.nr_inmat NOT IN
    (SELECT v1.nr_inmat      -- vehiculele cu transporturi neincheiate
    FROM vehicul v1, transport t
    WHERE t.nr_inmat = v.nr_inmat      -- join intre transport si
vehicul
    AND t.cod_transport IN
        (SELECT codT      -- multimea transporturilor cu livrari
neincheiate
        FROM transporturiNeincheiate
        )
    )
)

```

```

AND s.cod_sofer NOT IN
    (SELECT s1.cod_sofer          -- soferii cu transporturi neinchiate
    FROM sofer s1, transport t
    WHERE t.cod_sofer = s1.cod_sofer      -- join intre transport si
sofer
    AND t.cod_transport IN
        (SELECT codT          -- multimea transporturilor cu livrari
neinchiate
        FROM transporturiNeinchiate
        )
    );

```

EXCEPTION

```

    WHEN NO_DATA_FOUND THEN

        DBMS_OUTPUT.PUT_LINE('soferiVehiculeDisponibile: Nu am gasit
perechi disponibile de vehicule si soferi pentru produsul cu codul ' ||
codProdus || ' sau cod_depozit este NULL sau produsul dat nu exista. ');

        numeDepozit := -1;

        codVehiculDisponibil := -1;

        codSoferDisponibil:= -1;

    WHEN TOO_MANY_ROWS THEN

        DBMS_OUTPUT.PUT_LINE('soferiVehiculeDisponibile: S-au gasit mai
multe perechi disponibile de vehicule si soferi pentru produsul cu codul ' ||
codProdus);

        WITH transporturiNeinchiate AS

        (SELECT t.cod_transport codT

        FROM transport t

        WHERE

            (SELECT COUNT(l.cod_livrare)

```



```

FROM livrare l
WHERE l.cod_transport = t.cod_transport      -- join intre livrare si
transport
AND LOWER(l.status_livrare) NOT LIKE 'sosita' -- livrare neincheiata
) > 0
)
SELECT DISTINCT v.nr_inmat, s.cod_sofer
BULK COLLECT INTO perechiDisponibile
FROM produs p, depozit d, vehicul v, sofer s
WHERE p.cod_produs = codProdus  -- produsul cautat
AND d.cod_depozit = p.cod_depozit -- join intre depozit si produs
AND v.cod_depozit = d.cod_depozit -- join intre vehicul si depozit
AND s.cod_depozit = d.cod_depozit -- join intre sofer si depozit
AND v.nr_inmat NOT IN
    (SELECT v1.nr_inmat          -- vehiculele cu transporturi
neincheiate
    FROM vehicul v1, transport t
    WHERE t.nr_inmat = v.nr_inmat      -- join intre transport si
vehicul
    AND t.cod_transport IN
        (SELECT codT          -- multimea transporturilor cu livrari
neincheiate
        FROM transporturiNeincheiate
        )
    )
AND s.cod_sofer NOT IN
    (SELECT s1.cod_sofer        -- soferii cu transporturi neincheiate
    FROM sofer s1, transport t
    WHERE t.cod_sofer = s1.cod_sofer      -- join intre transport si
sofer

```

```

        AND t.cod_transport IN
            (SELECT codT
              -- multimea transporturilor cu livrari
neincheiate
              FROM transporturiNeincheiate
            )
    );

```

```

codVehiculDisponibil := perechiDisponibile(1).nrInmat;

```

```

codSoferDisponibil:= perechiDisponibile(1).codSofer;

```

```

SELECT d.numa
INTO numeDepozit
FROM depozit d, sofer s
WHERE s.cod_sofer = codSoferDisponibil -- soferul gasit
AND d.cod_depozit = s.cod_depozit;    -- join intre depozit si sofer

```

```

WHEN OTHERS THEN

```

```

    DBMS_OUTPUT.PUT_LINE('soferiVehiculeDisponibile: Alta exceptie
pentru produsul cu codul ' || codProdus);

```

```

END;

```

```

/

```

```

SELECT cod_produs produs, cod_depozit depozit FROM produs;

```

```

DECLARE

```

```

    TYPE coduriProduse IS VARRAY(5) OF produs.cod_produs%TYPE;

```

```

    varrayCoduriProduse coduriProduse:=coduriProduse();

```

```

    codV  vehicul.nr_inmat%TYPE;

```

```

    codS  sofer.cod_sofer%TYPE;

```

```

    numeD depozit.nume%TYPE;
BEGIN
    FOR i IN 1..5 LOOP
        varrayCoduriProduse.extend();
    END LOOP;

    varrayCoduriProduse(1) := 14;
    varrayCoduriProduse(2) := 15;
    varrayCoduriProduse(3) := 35;
    varrayCoduriProduse(4) := 36;
    varrayCoduriProduse(5) := 37;

    FOR i IN varrayCoduriProduse.FIRST..varrayCoduriProduse.LAST LOOP
        soferiVehiculeDisponibile(varrayCoduriProduse(i), numeD, codV, codS);

        IF codS >= 0
            THEN
                DBMS_OUTPUT.PUT_LINE('Pentru produsul cu codul ' ||
varrayCoduriProduse(i) || ': ');
                DBMS_OUTPUT.PUT_LINE('  nume depozit: ' || numeD);
                DBMS_OUTPUT.PUT_LINE('  nr inmat vehicul disponibil: ' || codV);
                DBMS_OUTPUT.PUT_LINE('  cod sofer disponibil: ' || codS);
                DBMS_OUTPUT.NEW_LINE();
            ELSE
                DBMS_OUTPUT.NEW_LINE();
            END IF;
        END LOOP;
    END;
/

```

Procedure SOFERIVEHICULEDISPONIBILE compiled

PRODUS	DEPOZIT
--------	---------

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	

PRODUS	DEPOZIT
--------	---------

12	
13	
14	
15	4
16	4
17	4
18	4
19	4
20	4
30	4
31	3

PRODUS	DEPOZIT
--------	---------

32	2
33	2
34	1

34 1
35 1
36 3
37 2
21
22
23
24
25

PRODUS DEPOZIT

26
27
28 |
29
38
39
40
41
42
43
44

PRODUS DEPOZIT

45
46
47
48
49
50
51
52
53
54
55

PRODUS DEPOZIT

56

56 rows selected.

PL/SQL procedure successfully completed.

(Poză mai mare, extrasă din poza anterioară):

10.

-- ex 10: Creati un trigger de tip LMD la nivel de comanda cu executare dupa, pentru update-uri pe coloana status_livrare a tabelului livrare.

Triggerul va calcula pretul maxim al unui produs aflat intr-un magazin si va actualiza preturile tuturor produselor aflate in magazine si avand acel pret maxim, prin reducere cu 5%.

```
CREATE OR REPLACE TRIGGER ofertaProdus
```

```
    AFTER UPDATE OF status_livrare ON livrare
```

```
DECLARE
```

```
    TYPE coduriProduse IS TABLE OF produs.cod_produs%TYPE;
```

```
    tblCoduriProduse  coduriProduse := coduriProduse();
```

```
    pretMaxim  produs.pret%TYPE;
```

```
    pretCrt  produs.pret%TYPE;
```

```
    codMagazinCrt  produs.cod_magazin%TYPE;
```

```
BEGIN
```

```
    SELECT MAX(pret)
```

```
    INTO pretMaxim
```

```
    FROM produs
```

```
    WHERE cod_magazin IS NOT NULL;
```

```
    SELECT cod_produs
```

```
    BULK COLLECT INTO tblCoduriProduse
```

```
    FROM produs
```

```
    WHERE pret = pretMaxim
```

```
    AND cod_magazin IS NOT NULL;
```

```
    FOR i IN tblCoduriProduse.FIRST..tblCoduriProduse.LAST LOOP
```

```
        SELECT pret, cod_magazin
```

```
        INTO pretCrt, codMagazinCrt
```

```
        FROM produs
```

```
        WHERE cod_produs = tblCoduriProduse(i);
```

```
        DBMS_OUTPUT.PUT_LINE('Produsul cu codul ' || tblCoduriProduse(i) || ',  
aflat in magazinul cu codul ' || codMagazinCrt || ', este redus prin oferta de la '  
|| pretCrt || ' lei la ' || 95/100*pretCrt || ' lei!');
```

```
    END LOOP;
```

```
FORALL i IN tblCoduriProduse.FIRST..tblCoduriProduse.LAST
```

```
    UPDATE produs
```

```
        SET pret = 95/100 * pret
```

```
        WHERE cod_produs = tblCoduriProduse(i);
```

```
EXCEPTION
```

```
    WHEN NO_DATA_FOUND THEN
```

```
        DBMS_OUTPUT.PUT_LINE('ofertaProdus: Eroare no_data_found; este  
posibil sa nu existe niciun produs in magazine.');
```

```
END;
```

```
/
```

```
SELECT cod_produs produs, NVL(cod_magazin, 0) magazin, pret,  
NVL(cod_livrare, 0) livrare, nume FROM produs;
```

```
SELECT * FROM livrare;
```

```
EXECUTE sosireLivrare(2);
```

```
/
```

```
SELECT cod_produs produs, NVL(cod_magazin, 0) magazin, pret,  
NVL(cod_livrare, 0) livrare, nume FROM produs;
```

```
SELECT * FROM LIVRARE;
```


Trigger OFERTAPRODUS compiled

PRODUS	MAGAZIN	PRET	LIVRARE NUME
<hr/>			
1	4	1600	0 bas acoustic
2	4	4400	0 chitara el. fretless
3	4	1600	0 bas acoustic
4	4	1600	0 bas acoustic
5	4	1600	0 chitara el. DMustang
6	4	1600	0 chitara el. :Haver
7	4	1600	0 drum set full iron
8	3	1600	0 bas acoustic
9	2	1600	0 drum set full iron
10	2	1600	0 chitara el. DMustang
11	1	2590	0 chitara el. JumpingW
<hr/>			
PRODUS	MAGAZIN	PRET	LIVRARE NUME
<hr/>			
12	1	2590	0 chitara el. JumpingW
13	3	5000	0 combo chitara 400V
14	2	2800	0 interfata RedPumpKin
15	0	1600	0 bas acoustic
16	0	4400	0 chitara el. fretless
17	0	1600	0 bas acoustic
18	0	1600	0 bas acoustic
19	0	1600	0 chitara el. DMustang
20	0	1600	0 chitara el. :Haver
30	0	1600	0 drum set full iron
31	0	1600	0 bas acoustic
<hr/>			
PRODUS	MAGAZIN	PRET	LIVRARE NUME
<hr/>			
32	0	1600	0 drum set full iron
33	0	1600	0 chitara el. DMustang
34	0	2590	0 chitara el. JumpingW
<hr/>			

34	0	2590	0 chitara el. JumpingW
35	0	2590	0 chitara el. JumpingW
36	0	5000	0 combo chitara 400V
37	0	2800	0 interfata RedPumpKin
21	0	1600	0 bas acustic
22	0	4400	0 chitara el. fretless
23	0	1600	0 bas acustic
24	0	1600	0 bas acustic
25	0	1600	0 chitara el. DMustang

PRODUS	MAGAZIN	PRET	LIVRARE NUME
--------	---------	------	--------------

26	0	1600	0 chitara el. :Haver
27	0	1600	0 drum set full iron
28	0	1600	0 bas acustic
29	0	1600	0 drum set full iron
38	0	1600	0 chitara el. DMustang
39	0	2590	0 chitara el. JumpingW
40	0	2590	0 chitara el. JumpingW
41	0	5000	0 combo chitara 400V
42	0	2800	0 interfata RedPumpKin
43	0	1600	3 bas acustic
44	0	4400	2 chitara el. fretless

PRODUS	MAGAZIN	PRET	LIVRARE NUME
--------	---------	------	--------------

45	0	1600	1 bas acustic
46	0	1600	5 bas acustic
47	0	1600	1 chitara el. DMustang
48	0	1600	6 chitara el. :Haver
49	0	1600	2 drum set full iron
50	0	1600	3 bas acustic
51	0	1600	2 drum set full iron
52	0	1600	1 chitara el. DMustang
53	0	2590	2 chitara el. JumpingW
54	0	2590	3 chitara el. JumpingW
55	0	5000	6 combo chitara 400V

```
Script Output x
Task completed in 0,919 seconds

55      0      5000      6 combo chitara 400V

PRODUS  MAGAZIN  PRET  LIVRARE NUME
-----
56      0      2800      6 interfata RedPumpKin

56 rows selected.

COD_LIVRARE COD_TRANSPORT COD_MAGAZIN STATUS_LIVRARE
-----
1          1          4 pe drum
2          2          4 pe drum
3          3          3 pe drum
4          4          3 pe drum
5          5          5 pe drum
6          6          1 pe drum
7          7          1 pe drum
8          8          2 pe drum
9          9          4 pe drum
10         10          1 pe drum

10 rows selected.

PL/SQL procedure successfully completed.
```

```
Dbms Output x
Buffer Size: 20000

MusicShp x
Produsul cu codul 13, aflat in magazinul cu codul 3, este redus prin oferta de la 5000 lei la 4750 lei!
Produsul cu codul 44 a fost livrat la magazinul cu codul 4
Produsul cu codul 49 a fost livrat la magazinul cu codul 4
Produsul cu codul 51 a fost livrat la magazinul cu codul 4
Produsul cu codul 53 a fost livrat la magazinul cu codul 4
Livrarea cu codul 2 s-a incheiat
```

Script Output x			
Task completed in 0,919 seconds			
PRODUS	MAGAZIN	PRET	LIVRARE NUME
1	4	1600	0 bas acoustic
2	4	4400	0 chitara el. fretless
3	4	1600	0 bas acoustic
4	4	1600	0 bas acoustic
5	4	1600	0 chitara el. DMustang
6	4	1600	0 chitara el. :Haver
7	4	1600	0 drum set full iron
8	3	1600	0 bas acoustic
9	2	1600	0 drum set full iron
10	2	1600	0 chitara el. DMustang
11	1	2590	0 chitara el. JumpingW
PRODUS	MAGAZIN	PRET	LIVRARE NUME
12	1	2590	0 chitara el. JumpingW
13	3	4750	0 combo chitara 400V
14	2	2800	0 interfata RedPumpKin
15	0	1600	0 bas acoustic
16	0	4400	0 chitara el. fretless
17	0	1600	0 bas acoustic
18	0	1600	0 bas acoustic
19	0	1600	0 chitara el. DMustang
20	0	1600	0 chitara el. :Haver
30	0	1600	0 drum set full iron
31	0	1600	0 bas acoustic
PRODUS	MAGAZIN	PRET	LIVRARE NUME
32	0	1600	0 drum set full iron
33	0	1600	0 chitara el. DMustang
34	0	2590	0 chitara el. JumpingW
35	0	2590	0 chitara el. JumpingW
36	0	5000	0 combo chitara 400V
37	0	2800	0 interfata RedPumpKin
21	0	1600	0 bas acoustic

Script Output x			
Task completed in 0,919 seconds			
21	0	1600	0 bas acoustic
22	0	4400	0 chitara el. fretless
23	0	1600	0 bas acoustic
24	0	1600	0 bas acoustic
25	0	1600	0 chitara el. DMustang
PRODUS	MAGAZIN	PRET	LIVRARE NUME
26	0	1600	0 chitara el. :Haver
27	0	1600	0 drum set full iron
28	0	1600	0 bas acoustic
29	0	1600	0 drum set full iron
38	0	1600	0 chitara el. DMustang
39	0	2590	0 chitara el. JumpingW
40	0	2590	0 chitara el. JumpingW
41	0	5000	0 combo chitara 400V
42	0	2800	0 interfata RedPumpKin
43	0	1600	3 bas acoustic
44	4	4400	0 chitara el. fretless
PRODUS	MAGAZIN	PRET	LIVRARE NUME
45	0	1600	1 bas acoustic
46	0	1600	5 bas acoustic
47	0	1600	1 chitara el. DMustang
48	0	1600	6 chitara el. :Haver
49	4	1600	0 drum set full iron
50	0	1600	3 bas acoustic
51	4	1600	0 drum set full iron
52	0	1600	1 chitara el. DMustang
53	4	2590	0 chitara el. JumpingW
54	0	2590	3 chitara el. JumpingW
55	0	5000	6 combo chitara 400V
PRODUS	MAGAZIN	PRET	LIVRARE NUME
56	0	2800	6 interfata RedPumpKin

56 0 2800 6 interfata RedPumpKin

56 rows selected.

COD_LIVRARE COD_TRANSPORT COD_MAGAZIN STATUS_LIVRARE

1	1	4 pe drum
2	2	4 sosita
3	3	3 pe drum
4	4	3 pe drum
5	5	5 pe drum
6	6	1 pe drum
7	7	1 pe drum
8	8	2 pe drum
9	9	4 pe drum
10	10	1 pe drum

10 rows selected.

11.

-- ex 11: Creati un trigger de tip LMD la nivel de linie cu executare inainte, pentru update-uri pe coloana status_livrare a tabelului livrare.

Triggerul va adauga intr-un depozit anume (adica va insera in tabelul produs) cate un produs nou pentru fiecare produs din livrarea ce se va actualiza.

Produsele noi vor prelua de la cele livrate: tip, nume, pret, greutate, material_predominant, producator

Depozitul se va alege in felul urmator: prioritatea cea mai mare o au depozitele din acelasi judet cu magazinul la care ajunge livrarea, apoi mai au prioritate depozitele cu un numar cat mai mic de produse (calculat inainte de actualizarile triggerului).

```
SELECT m.cod_magazin, loc.judet
FROM magazin m, locatie loc
WHERE m.cod_locatie = loc.cod_locatie;
```

```
SELECT d.cod_depozit, loc.judet
FROM depozit d, locatie loc
```

```
WHERE d.cod_locatie = loc.cod_locatie;
```

```
CREATE SEQUENCE seqCoduriProduce
```

```
START WITH 57
```

```
INCREMENT BY 1
```

```
MINVALUE 1
```

```
MAXVALUE 999999999
```

```
CYCLE;
```

```
CREATE OR REPLACE TRIGGER completareStocDepozit
```

```
BEFORE UPDATE OF status_livrare ON livrare
```

```
FOR EACH ROW
```

```
DECLARE
```

```
TYPE produseNoi IS TABLE OF produs%ROWTYPE;
```

```
TYPE coduriDepozite IS TABLE OF depozit.cod_depozit%TYPE;
```

```
tblProduseNoi produseNoi := produseNoi();
```

```
tblCoduriDepozite coduriDepozite := coduriDepozite();
```

```
codLivrare livrare.cod_livrare%TYPE;
```

```
codDepozit depozit.cod_depozit%TYPE;
```

```
codMagazin livrare.cod_magazin%TYPE;
```

```
BEGIN
```

```
codLivrare := :OLD.cod_livrare;
```

```
codMagazin := :OLD.cod_magazin;
```

```
-- retinem depozitele din acelasi judet cu magazinul la care ajunge livrarea  
curenta:
```

```
SELECT d.cod_depozit
```

```
BULK COLLECT INTO tblCoduriDepozite
```

```

FROM locatie loc, depozit d
WHERE loc.judet =                -- locatiile aflate in judetul curent
    (SELECT loc.judet judetulCrt
     FROM magazin m, locatie loc
     WHERE m.cod_magazin = codMagazin -- magazinul la care ajunge livrarea
curenta
     AND loc.cod_locatie = m.cod_locatie) -- join intre locatie si magazin
(adica locatia magazinului)
AND d.cod_locatie = loc.cod_locatie -- join intre depozit si locatie (adica
depozitul de la locatie)
ORDER BY                        -- depozitele cu mai putine produse au
prioritate
    (SELECT COUNT(p.cod_produș)
     FROM produs p
     WHERE p.cod_depozit = d.cod_depozit -- join intre produs si depozit
    );

-- daca nu am gasit un depozit in acelasi judet, consideram toate depozitele
IF tblCoduriDepozite.COUNT = 0
THEN
    SELECT d.cod_depozit
    BULK COLLECT INTO tblCoduriDepozite
    FROM depozit d
    ORDER BY                        -- depozitele cu mai putine produse au
prioritate
        (SELECT COUNT(p.cod_produș)
         FROM produs p
         WHERE p.cod_depozit = d.cod_depozit -- join intre produs si depozit
        );
END IF;

```



```
codDepozit := tblCoduriDepozite(1); -- retinem doar primul depozit (cu
prioritatea cea mai mare) din cate avem la dispozitie
```

```
SELECT seqCoduriProduse.NEXTVAL, codDepozit, NULL, NULL, NULL, p.tip,
p.nume, p.pret, p.greutate, p.material_predominant, p.producator
```

```
BULK COLLECT INTO tblProduseNoi
```

```
FROM produs p
```

```
WHERE p.cod_livrare = codLivrare;
```

```
FOR i IN tblProduseNoi.FIRST..tblProduseNoi.LAST LOOP
```

```
-- inseram in produs o linie cu toate datele din tblProduseNoi(i):
```

```
INSERT INTO produs VALUES(tblProduseNoi(i).cod_produs,
tblProduseNoi(i).cod_depozit, tblProduseNoi(i).cod_magazin,
tblProduseNoi(i).cod_lutier, tblProduseNoi(i).cod_livrare,
```

```
tblProduseNoi(i).tip, tblProduseNoi(i).nume, tblProduseNoi(i).pret,
tblProduseNoi(i).greutate, tblProduseNoi(i).material_predominant,
tblProduseNoi(i).producator);
```

```
DBMS_OUTPUT.PUT_LINE('In depozitul cu codul ' ||
tblProduseNoi(i).cod_depozit || ' a fost adaugat produsul cu codul ' ||
tblProduseNoi(i).cod_produs);
```

```
END LOOP;
```

```
EXCEPTION
```

```
WHEN NO_DATA_FOUND THEN
```

```
DBMS_OUTPUT.PUT_LINE('completareStocDepozit: Eroare
no_data_found; este posibil sa nu fie produse in livrarea respectiva');
```

```
END;
```

```
/
```

```
SELECT *
```

```
FROM produs;
```

```
EXECUTE sosireLivrare(1);
```

```
/
```

```
SELECT *
```

```
FROM produs;
```



COD_PRODUS	COD_DEPOZIT	COD_MAGAZIN	COD_LUTIER	COD_LIVRARE	TIP	NUME	PRET	GREUTATE	MATERIAL_PREDOMINANT	PRODUCATOR
------------	-------------	-------------	------------	-------------	-----	------	------	----------	----------------------	------------

Produsele, inainte de declansarea triggerului si de actualizarea livrarii 1:

Script Output x										
Task completed in 0,141 seconds										
COD_PRODUS	COD_DEPOZIT	COD_MAGAZIN	COD_LUTIER	COD_LIVRARE	TIP	NUME	PRET	GREUTATE	MATERIAL_PREDOMINANT	PRODUCATOR
32	2		instrument	drum set full iron	1600	34 otel	Jerryhmann			
33	2		instrument	chitara el. DMustang	1600	stejar	DnM Raider			
34	1		instrument	chitara el. JumpingW	2590	3 frasin	Harvey Benington			
35	1		instrument	chitara el. JumpingW	2590	3 frasin	Harvey Benington			
36	3		audio	combo chitara 400V	5000	22 fier	GraySun			
37	2		inregistrare/audio	interfata RedPumpKin	2800	1 otel	Mathers			
21		3	instrument	bas acustic	1600	stejar	eBanmaltez			
22		2	instrument	chitara el. fretless	4400	artar	eBanmaltez			
23		1	instrument	bas acustic	1600	stejar	eBanmaltez			
24		5	instrument	bas acustic	1600	stejar	eBanmaltez			
25		1	instrument	chitara el. DMustang	1600	stejar	DnM Raider			
COD_PRODUS	COD_DEPOZIT	COD_MAGAZIN	COD_LUTIER	COD_LIVRARE	TIP	NUME	PRET	GREUTATE	MATERIAL_PREDOMINANT	PRODUCATOR
26		6	instrument	chitara el. :Haver	1600	4 stejar	Hoover			
27		2	instrument	drum set full iron	1600	34 otel	Jerryhmann			
28		3	instrument	bas acustic	1600	stejar	eBanmaltez			
29		2	instrument	drum set full iron	1600	34 otel	Jerryhmann			
38		1	instrument	chitara el. DMustang	1600	stejar	DnM Raider			
39		2	instrument	chitara el. JumpingW	2590	3 frasin	Harvey Benington			
40		3	instrument	chitara el. JumpingW	2590	3 frasin	Harvey Benington			
41		6	audio	combo chitara 400V	5000	22 fier	GraySun			
42		6	inregistrare/audio	interfata RedPumpKin	2800	1 otel	Mathers			
43		3	instrument	bas acustic	1600	stejar	eBanmaltez			
44		2	instrument	chitara el. fretless	4400	artar	eBanmaltez			
COD_PRODUS	COD_DEPOZIT	COD_MAGAZIN	COD_LUTIER	COD_LIVRARE	TIP	NUME	PRET	GREUTATE	MATERIAL_PREDOMINANT	PRODUCATOR
45		1	instrument	bas acustic	1600	stejar	eBanmaltez			
46		5	instrument	bas acustic	1600	stejar	eBanmaltez			
47		1	instrument	chitara el. DMustang	1600	stejar	DnM Raider			
48		6	instrument	chitara el. :Haver	1600	4 stejar	Hoover			
49		2	instrument	drum set full iron	1600	34 otel	Jerryhmann			
50		3	instrument	bas acustic	1600	stejar	eBanmaltez			
51		2	instrument	drum set full iron	1600	34 otel	Jerryhmann			

Script Output x										
Task completed in 0,141 seconds										
51		2	instrument	drum set full iron	1600	34 otel	Jerryhmann			
52		1	instrument	chitara el. DMustang	1600	stejar	DnM Raider			
53		2	instrument	chitara el. JumpingW	2590	3 frasin	Harvey Benington			
54		3	instrument	chitara el. JumpingW	2590	3 frasin	Harvey Benington			
55		6	audio	combo chitara 400V	5000	22 fier	GraySun			
COD_PRODUS	COD_DEPOZIT	COD_MAGAZIN	COD_LUTIER	COD_LIVRARE	TIP	NUME	PRET	GREUTATE	MATERIAL_PREDOMINANT	PRODUCATOR
56		6	inregistrare/audio	interfata RedPumpKin	2800	1 otel	Mathers			

56 rows selected.

PL/SQL procedure successfully completed.

Produsele, dupa declansarea triggerului si actualizarea livrarii 1:

Script Output x							
Task completed in 0,141 seconds							
COD_PRODUS	COD_DEPOZIT	COD_MAGAZIN	COD_LUTIER	COD_LIVRARE	TIP	NUME	PRET GREUTATE
1	4	instrument	bas acustic	1600	stejar	eBanmaltez	
2	4	instrument	chitara el. fretless	4400	artar	eBanmaltez	
3	4	instrument	bas acustic	1600	stejar	eBanmaltez	
4	4	instrument	bas acustic	1600	stejar	eBanmaltez	
5	4	instrument	chitara el. DMustang	1600	stejar	DnM Raider	
6	4	instrument	chitara el. :Haver	1600	4 stejar	Hoover	
7	4	instrument	drum set full iron	1600	34 otel	Jerryhmann	
8	3	instrument	bas acustic	1600	stejar	eBanmaltez	
9	2	instrument	drum set full iron	1600	34 otel	Jerryhmann	
10	2	instrument	chitara el. DMustang	1600	stejar	DnM Raider	
11	1	instrument	chitara el. JumpingW	2590	3 frasin	Harvey Benington	
COD_PRODUS	COD_DEPOZIT	COD_MAGAZIN	COD_LUTIER	COD_LIVRARE	TIP	NUME	PRET GREUTATE
12	1	instrument	chitara el. JumpingW	2590	3 frasin	Harvey Benington	
13	3	audio	combo chitara 400V	5000	22 fier	GraySun	
14	2	inregistrare/audio	interfata RedPumpKin	2800	1 otel	Mathers	
15	4	instrument	bas acustic	1600	stejar	eBanmaltez	
16	4	instrument	chitara el. fretless	4400	artar	eBanmaltez	
17	4	instrument	bas acustic	1600	stejar	eBanmaltez	
18	4	instrument	bas acustic	1600	stejar	eBanmaltez	
19	4	instrument	chitara el. DMustang	1600	stejar	DnM Raider	
20	4	instrument	chitara el. :Haver	1600	4 stejar	Hoover	
30	4	instrument	drum set full iron	1600	34 otel	Jerryhmann	
31	3	instrument	bas acustic	1600	stejar	eBanmaltez	
COD_PRODUS	COD_DEPOZIT	COD_MAGAZIN	COD_LUTIER	COD_LIVRARE	TIP	NUME	PRET GREUTATE
32	2	instrument	drum set full iron	1600	34 otel	Jerryhmann	
33	2	instrument	chitara el. DMustang	1600	stejar	DnM Raider	
34	1	instrument	chitara el. JumpingW	2590	3 frasin	Harvey Benington	
35	1	instrument	chitara el. JumpingW	2590	3 frasin	Harvey Benington	
36	3	audio	combo chitara 400V	5000	22 fier	GraySun	
37	2	inregistrare/audio	interfata RedPumpKin	2800	1 otel	Mathers	

Script Output x							
Task completed in 0,141 seconds							
37	2		inregistrare/audio	interfata RedPumpKin	2800	1 otel	Mathers
21		3	instrument	bas acustic	1600	stejar	eBanmaltez
22		2	instrument	chitara el. fretless	4400	artar	eBanmaltez
23		1	instrument	bas acustic	1600	stejar	eBanmaltez
24		5	instrument	bas acustic	1600	stejar	eBanmaltez
25		1	instrument	chitara el. DMustang	1600	stejar	DnM Raider
COD_PRODUS COD_DEPOZIT COD_MAGAZIN COD_LUTIER COD_LIVRARE TIP NUME PRET GREUTATE							
26		6	instrument	chitara el. :Haver	1600	4 stejar	Hoover
27		2	instrument	drum set full iron	1600	34 otel	Jerryhmann
28		3	instrument	bas acustic	1600	stejar	eBanmaltez
29		2	instrument	drum set full iron	1600	34 otel	Jerryhmann
38		1	instrument	chitara el. DMustang	1600	stejar	DnM Raider
39		2	instrument	chitara el. JumpingW	2590	3 frasin	Harvey Benington
40		3	instrument	chitara el. JumpingW	2590	3 frasin	Harvey Benington
41		6	audio	combo chitara 400V	5000	22 fier	GraySun
42		6	inregistrare/audio	interfata RedPumpKin	2800	1 otel	Mathers
43		3	instrument	bas acustic	1600	stejar	eBanmaltez
44		2	instrument	chitara el. fretless	4400	artar	eBanmaltez
COD_PRODUS COD_DEPOZIT COD_MAGAZIN COD_LUTIER COD_LIVRARE TIP NUME PRET GREUTATE							
45		4	instrument	bas acustic	1600	stejar	eBanmaltez
46		5	instrument	bas acustic	1600	stejar	eBanmaltez
47		4	instrument	chitara el. DMustang	1600	stejar	DnM Raider
48		6	instrument	chitara el. :Haver	1600	4 stejar	Hoover
49		2	instrument	drum set full iron	1600	34 otel	Jerryhmann
50		3	instrument	bas acustic	1600	stejar	eBanmaltez
51		2	instrument	drum set full iron	1600	34 otel	Jerryhmann
52		4	instrument	chitara el. DMustang	1600	stejar	DnM Raider
53		2	instrument	chitara el. JumpingW	2590	3 frasin	Harvey Benington
54		3	instrument	chitara el. JumpingW	2590	3 frasin	Harvey Benington
55		6	audio	combo chitara 400V	5000	22 fier	GraySun
55		6	audio	combo chitara 400V	5000	22 fier	GraySun
COD_PRODUS COD_DEPOZIT COD_MAGAZIN COD_LUTIER COD_LIVRARE TIP NUME PRET GREI							
56		6	inregistrare/audio	interfata RedPumpKin	2800	1 otel	Mathers
57	6		instrument	bas acustic	1600	stejar	eBanmaltez
58	6		instrument	chitara el. DMustang	1600	stejar	DnM Raider
59	6		instrument	chitara el. DMustang	1600	stejar	DnM Raider

59 rows selected.

Dbms Output x	
Buffer Size: 20000	
MusicShp x	
In depozitul cu codul 6 a fost adaugat produsul cu codul 57	
In depozitul cu codul 6 a fost adaugat produsul cu codul 58	
In depozitul cu codul 6 a fost adaugat produsul cu codul 59	
Produsul cu codul 45 a fost livrat la magazinul cu codul 4	
Produsul cu codul 47 a fost livrat la magazinul cu codul 4	
Produsul cu codul 52 a fost livrat la magazinul cu codul 4	
Livrarea cu codul 1 s-a incheiat	

12.

-- ex 12: Creati un trigger la nivel de schema pentru drop, cu executare inainte.

Triggerul verifica daca obiectul de sters este tabelul produs si apoi, in functie de valoarea unui boolean definit intr-un pachet separat pentru operarea permisiunilor, triggerul fie va permite stergerea tabelului, fie va ridica o eroare difinita de utilizator.

```
CREATE OR REPLACE PACKAGE pkgPermisuni AS
```

```
    permisiuneDropProdus  BOOLEAN := TRUE;
```

```
    PROCEDURE setPermisuneDropProdus (val IN BOOLEAN);
```

```
END pkgPermisuni;
```

```
/
```

```
CREATE OR REPLACE PACKAGE BODY pkgPermisuni AS
```

```
    PROCEDURE setPermisuneDropProdus (val IN BOOLEAN) IS
```

```
    BEGIN
```

```
        permisiuneDropProdus := val;
```

```
        IF permisiuneDropProdus
```

```
            THEN
```

```
                DBMS_OUTPUT.PUT_LINE('permisiuneDropProdus a fost setata la TRUE');
```

```
            ELSE
```

```
                DBMS_OUTPUT.PUT_LINE('permisiuneDropProdus a fost setata la FALSE');
```

```
            END IF;
```

```
        END;
```

```
END pkgPermisuni;
```

```
/
```

```
CREATE OR REPLACE TRIGGER verificareStergere
```

```

BEFORE DROP ON SCHEMA

DECLARE

    permisiunea BOOLEAN;

BEGIN

    IF LOWER(SYS.DICTIONARY_OBJ_NAME) LIKE 'produs'

        THEN

            permisiunea := pkgPermisiuni.permisiuneDropProdus;


            IF permisiunea = FALSE

                THEN

                    DBMS_OUTPUT.PUT_LINE('verificareStergere: (' || SYSDATE || ') Nu
a fost permisa stergerea tabelului produs.');
```

RAISE_APPLICATION_ERROR(-20998, 'verificareStergere: Nu a fost
permisa stergerea tabelului produs! Setati permisiuneDropProdus la TRUE si
incercati iar!');

```

                ELSE

                    DBMS_OUTPUT.PUT_LINE('verificareStergere: (' || SYSDATE || ')
Tabelulul produs a fost sters cu succes.');
```

END IF;

```

            END IF;

        END IF;

    END;

/

EXECUTE pkgPermisiuni.setPermisiuneDropProdus(FALSE);

/

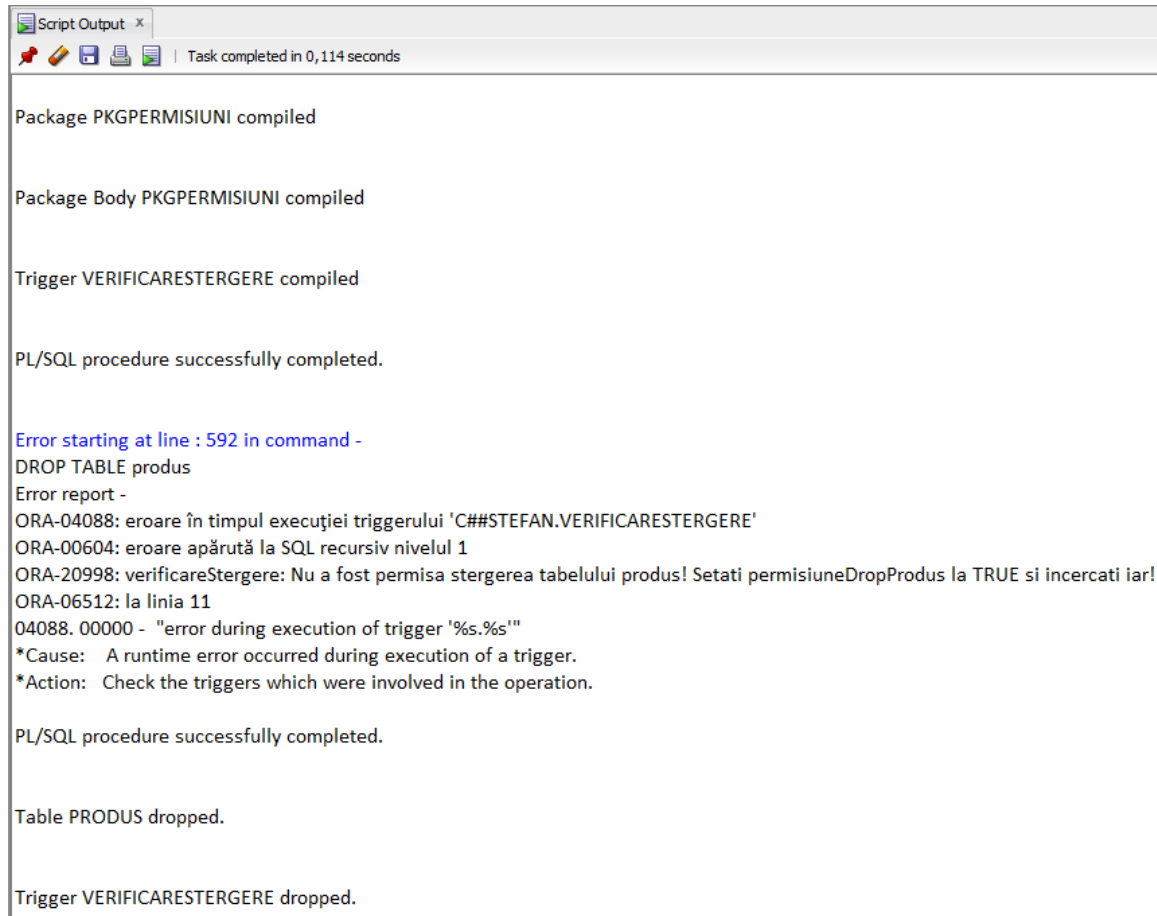
DROP TABLE produs;

EXECUTE pkgPermisiuni.setPermisiuneDropProdus(TRUE);

/

```

DROP TABLE produs;



Script Output x

Task completed in 0,114 seconds

Package PKGPERMISIUNI compiled

Package Body PKGPERMISIUNI compiled

Trigger VERIFICARESTERGERE compiled

PL/SQL procedure successfully completed.

Error starting at line : 592 in command -
DROP TABLE produs
Error report -
ORA-04088: eroare în timpul execuției triggerului 'C##STEFAN.VERIFICARESTERGERE'
ORA-00604: eroare apărută la SQL recursiv nivelul 1
ORA-20998: verificareStergere: Nu a fost permisa stergerea tabelului produs! Setati permisiuneDropProdus la TRUE si incercati iar!
ORA-06512: la linia 11
04088. 00000 - "error during execution of trigger '%s.%s'"
*Cause: A runtime error occurred during execution of a trigger.
*Action: Check the triggers which were involved in the operation.

PL/SQL procedure successfully completed.

Table PRODUS dropped.

Trigger VERIFICARESTERGERE dropped.

(Bucata din poza de deasupra, cu scris mai mare):

Script Output x

Task completed in 0,114 seconds

Package PKGPERMISIUNI compiled

Package Body PKGPERMISIUNI compiled

Trigger VERIFICARESTERGERE compiled

PL/SQL procedure successfully completed.

Error starting at line : 592 in command -
DROP TABLE produs
Error report -
ORA-04088: eroare în timpul execuției triggerului 'C##STEFAN.VERIFICARESTERGERE'
ORA-00604: eroare apărută la SQL recursiv nivelul 1
ORA-20998: verificareStergere: Nu a fost permisa stergerea tabelului produs! Setati permisiuneD
ORA-06512: la linia 11
04088. 00000 - "error during execution of trigger '%s.%s'"
*Cause: A runtime error occurred during execution of a trigger.
*Action: Check the triggers which were involved in the operation.

PL/SQL procedure successfully completed.

Table PRODUS dropped.

Trigger VERIFICARESTERGERE dropped.

Dbms Output x

Buffer Size: 20000

MusicShp x

permisiuneDropProdus a fost setata la FALSE

verificareStergere: (04-01-2022) Nu a fost permisa stergerea tabelului produs.

permisiuneDropProdus a fost setata la TRUE

verificareStergere: (04-01-2022) Tabelul produs a fost sters cu succes.

13.

-- ex 13: Creati un pachet musicShopExtensii ce va cuprinde toate tipurile, functiile si procedurile create pana acum pentru operarea bazei de date MusicShp.

```
CREATE OR REPLACE PACKAGE musicShopExtensii AS
```

```
-- de la ex 7:
```

```
PROCEDURE sosireLivrare(codLivrare IN NUMBER);
```

```
-- de la ex 9:
```

```
PROCEDURE soferiVehiculeDisponibile (codProdus IN  
produs.cod_produs%TYPE, numeDepozit OUT depozit.nume%TYPE,  
codVehiculDisponibil OUT vehicul.nr_inmat%TYPE, codSoferDisponibil OUT  
sofer.cod_sofer%TYPE);
```

```
-- de la ex 6:
```

```
TYPE varrayCoduriLivrari IS VARRAY(100) OF NUMBER(10);
```

```
TYPE tblCoduriVehicule IS TABLE OF VARCHAR2(20);
```

```
TYPE tblCoduriSoferi IS TABLE OF VARCHAR2(10);
```

```
PROCEDURE elibereazaVehiculeSoferi(nrLivrari IN NUMBER, coduriLivrari IN  
musicShopExtensii.varrayCoduriLivrari, coduriVehicule OUT  
musicShopExtensii.tblCoduriVehicule,  
coduriSoferi OUT musicShopExtensii.tblCoduriSoferi);
```

```
-- de la ex 8:
```

```
FUNCTION lutierLocal(codProdus IN NUMBER)
```

```
RETURN lutier.cod_lutier%TYPE;
```

```
END musicShopExtensii;
```

```
/
```

```
CREATE OR REPLACE PACKAGE BODY musicShopExtensii AS
```

```
-- de la ex 7:
```

```
PROCEDURE sosireLivrare(codLivrare IN NUMBER) IS
```

```
TYPE tabelCoduriProduse IS TABLE OF NUMBER;
```

```
coduriProduse tabelCoduriProduse :=tabelCoduriProduse();
```

```
codMagazin NUMBER;
```

```
codProdusCrt NUMBER;
```

```
CURSOR c IS
```

```
    SELECT cod_produs
```

```
    BULK COLLECT INTO coduriProduse
```

```
    FROM produs
```

```
    WHERE cod_livrare = codLivrare;
```

```
BEGIN
```

```
    IF (codLivrare IS NULL)
```

```
        THEN
```

```
            DBMS_OUTPUT.PUT_LINE('sosireLivrare: Parametrul dat nu poate fi  
NULL!');
```

```
            RETURN;
```

```
        END IF;
```

```
    SELECT cod_magazin
```

```
    INTO codMagazin
```

```
    FROM livrare
```

```
    WHERE cod_livrare = codLivrare;
```

```
    UPDATE livrare
```

```
    SET status_livrare = 'sosita'
```

```
    WHERE cod_livrare = codLivrare;
```

```

OPEN c;

LOOP

    FETCH c INTO codProdusCrt;

    EXIT WHEN c%NOTFOUND;


    UPDATE produs

    SET cod_livrare = NULL, cod_magazin = codMagazin

    WHERE cod_produs = codProdusCrt;

    DBMS_OUTPUT.PUT_LINE('Produsul cu codul ' || codProdusCrt || ' a
fost livrat la magazinul cu codul ' || codMagazin );

END LOOP;

CLOSE c;


    DBMS_OUTPUT.PUT_LINE('Livrarea cu codul ' || codLivrare || ' s-a
incheiat');


EXCEPTION

    WHEN NO_DATA_FOUND THEN

        DBMS_OUTPUT.PUT_LINE('sosireLivrare: Nu exista o livrare cu codul
' || codLivrare || '!');

END;


-- de la ex 9:

PROCEDURE soferiVehiculeDisponibile (codProdus IN
produs.cod_produs%TYPE, numeDepozit OUT depozit.nume%TYPE,
codVehiculDisponibil OUT vehicul.nr_inmat%TYPE, codSoferDisponibil OUT
sofer.cod_sofer%TYPE) AS

```

```

TYPE perecheVehiculSofer IS RECORD
(nrInmat  vehicul.nr_inmat%TYPE,
codSofer  sofer.cod_sofer%TYPE
);

TYPE tabelPerecheVehiculSofer IS TABLE OF perecheVehiculSofer;

perechiDisponibile tabelPerecheVehiculSofer :=
tabelPerecheVehiculSofer();

BEGIN

WITH transporturiNeincheiate AS
  (SELECT t.cod_transport codT
   FROM transport t
   WHERE
     (SELECT COUNT(l.cod_livrare)
      FROM livrare l
      WHERE l.cod_transport = t.cod_transport      -- join intre livrare si
transport
      AND LOWER(l.status_livrare) NOT LIKE 'sosita' -- livrare neincheiata
     ) > 0
  )

SELECT DISTINCT d.nume, v.nr_inmat, s.cod_sofer
INTO numeDepozit, codVehiculDisponibil, codSoferDisponibil
FROM produs p, depozit d, vehicul v, sofer s
WHERE p.cod_produs = codProdus  -- produsul cautat
AND d.cod_depozit = p.cod_depozit -- join intre depozit si produs
AND v.cod_depozit = d.cod_depozit -- join intre vehicul si depozit

```

```

AND s.cod_depozit = d.cod_depozit -- join intre sofer si depozit

AND v.nr_inmat NOT IN

    (SELECT v1.nr_inmat          -- vehiculele cu transporturi
neincheiate

    FROM vehicul v1, transport t

    WHERE t.nr_inmat = v.nr_inmat          -- join intre transport si
vehicul

    AND t.cod_transport IN

        (SELECT codT          -- multimea transporturilor cu livrari
neincheiate

        FROM transporturiNeincheiate

        )

    )

AND s.cod_sofer NOT IN

    (SELECT s1.cod_sofer          -- soferii cu transporturi neincheiate

    FROM sofer s1, transport t

    WHERE t.cod_sofer = s1.cod_sofer          -- join intre transport si
sofer

    AND t.cod_transport IN

        (SELECT codT          -- multimea transporturilor cu livrari
neincheiate

        FROM transporturiNeincheiate

        )

    );

EXCEPTION

    WHEN NO_DATA_FOUND THEN

```

```
DBMS_OUTPUT.PUT_LINE('soferiVehiculeDisponibile: Nu am gasit  
perechi disponibile de vehicule si soferi pentru produsul cu codul ' ||  
codProdus || ' sau cod_depozit este NULL sau produsul dat nu exista.' );
```

```
numeDepozit := -1;
```

```
codVehiculDisponibil := -1;
```

```
codSoferDisponibil:= -1;
```

```
WHEN TOO_MANY_ROWS THEN
```

```
DBMS_OUTPUT.PUT_LINE('soferiVehiculeDisponibile: S-au gasit mai  
multe perechi disponibile de vehicule si soferi pentru produsul cu codul ' ||  
codProdus);
```

```
WITH transporturiNeincheiate AS
```

```
(SELECT t.cod_transport codT
```

```
FROM transport t
```

```
WHERE
```

```
(SELECT COUNT(l.cod_livrare)
```

```
FROM livrare l
```

```
WHERE l.cod_transport = t.cod_transport      -- join intre livrare si  
transport
```

```
AND LOWER(l.status_livrare) NOT LIKE 'sosita' -- livrare neincheiata  
) > 0
```

```
)
```

```
SELECT DISTINCT v.nr_inmat, s.cod_sofer
```

```
BULK COLLECT INTO perechiDisponibile
```

```
FROM produs p, depozit d, vehicul v, sofer s
```

```
WHERE p.cod_produs = codProdus  -- produsul cautat
```

```
AND d.cod_depozit = p.cod_depozit -- join intre depozit si produs
```

```
AND v.cod_depozit = d.cod_depozit -- join intre vehicul si depozit
```

```

AND s.cod_depozit = d.cod_depozit -- join intre sofer si depozit
AND v.nr_inmat NOT IN
    (SELECT v1.nr_inmat          -- vehiculele cu transporturi
neincheiate
    FROM vehicul v1, transport t
    WHERE t.nr_inmat = v.nr_inmat          -- join intre transport
si vehicul
    AND t.cod_transport IN
        (SELECT codT          -- multimea transporturilor cu
livrari neincheiate
        FROM transporturiNeincheiate
        )
    )
AND s.cod_sofer NOT IN
    (SELECT s1.cod_sofer          -- soferii cu transporturi
neincheiate
    FROM sofer s1, transport t
    WHERE t.cod_sofer = s1.cod_sofer          -- join intre transport
si sofer
    AND t.cod_transport IN
        (SELECT codT          -- multimea transporturilor cu
livrari neincheiate
        FROM transporturiNeincheiate
        )
    );

codVehiculDisponibil := perechiDisponibile(1).nrInmat;
codSoferDisponibil:= perechiDisponibile(1).codSofer;

SELECT d.num

```



```

        INTO numeDepozit
        FROM depozit d, sofer s
        WHERE s.cod_sofer = codSoferDisponibil -- soferul gasit
        AND d.cod_depozit = s.cod_depozit;      -- join intre depozit si sofer

    WHEN OTHERS THEN

        DBMS_OUTPUT.PUT_LINE('soferiVehiculeDisponibile: Alta exceptie
        pentru produsul cu codul ' || codProdus);

    END;

```

-- de la ex 6:

```

    PROCEDURE elibereazaVehiculeSoferi(nrLivrari IN NUMBER, coduriLivrari IN
    musicShopExtensii.varrayCoduriLivrari, coduriVehicule OUT
    musicShopExtensii.tblCoduriVehicule,
    coduriSoferi OUT musicShopExtensii.tblCoduriSoferi) IS
        codTransport  NUMBER;
    BEGIN
        FOR i IN coduriLivrari.FIRST..coduriLivrari.LAST LOOP
            musicShopExtensii.sosireLivrare(coduriLivrari(i));      -- se incheie
            livrarea crt

```

```

        SELECT cod_transport
        INTO codTransport
        FROM livrare
        WHERE cod_livrare = coduriLivrari(i);

```

```

        SELECT DISTINCT v.nr_inmat

```

```

BULK COLLECT INTO coduriVehicule
FROM vehicul v, transport t
WHERE t.cod_transport = codTransport -- transportul ce corespunde
livrarii crt
AND t.nr_inmat = v.nr_inmat          -- join intre transport si vehicul
AND t.cod_transport NOT IN          -- transport care sa nu fie in
mulimea transporturilor cu livrari neincheiate
(SELECT t1.cod_transport
FROM transport t1
WHERE
(SELECT COUNT(l.cod_livrare)
FROM livrare l
WHERE l.cod_transport = t.cod_transport      -- join intre livrare si
transport
AND LOWER(l.status_livrare) NOT LIKE 'sosita' -- livrare neincheiata
) > 0
);

```

```

SELECT DISTINCT s.cod_sofer
BULK COLLECT INTO coduriSoferi
FROM sofer s, transport t
WHERE t.cod_transport = codTransport -- transportul ce corespunde
livrarii crt
AND t.cod_sofer = s.cod_sofer        -- join intre transport si sofer
AND t.cod_transport NOT IN          -- transport care sa nu fie in
mulimea transporturilor cu livrari neincheiate
(SELECT t1.cod_transport
FROM transport t1

```

```

WHERE
    (SELECT COUNT(l.cod_livrare)
     FROM livrare l
     WHERE l.cod_transport = t.cod_transport      -- join intre livrare si
transport
     AND LOWER(l.status_livrare) NOT LIKE 'sosita' -- livrare neincheiata
    ) > 0
);
END LOOP;
END;

```

-- de la ex 8:

```

FUNCTION lutierLocal(codProdus IN NUMBER)
RETURN lutier.cod_lutier%TYPE IS lutierul lutier.cod_lutier%TYPE;
    TYPE coduriLutieri  IS TABLE OF lutier.cod_lutier%TYPE;

    judetLocatie  locatie.judet%TYPE;
    codLutier      lutier.cod_lutier%TYPE;
    tblCoduriLutieri  coduriLutieri := coduriLutieri();
BEGIN
    SELECT l.judet
    INTO judetLocatie
    FROM produs p, magazin m, locatie l
    WHERE p.cod_produs = codProdus      -- alegem produsul cu codul dat
    AND m.cod_magazin = p.cod_magazin  -- join intre magazin si produs
    AND l.cod_locatie = m.cod_locatie;  -- join intre locatie si magazin

```

```

SELECT lut.cod_lutier
INTO codLutier
FROM lutier lut, locatie loc
WHERE loc.judet = judetLocatie      -- judetul gasit
AND lut.cod_locatie = loc.cod_locatie; -- join intre lutier si locatie

RETURN codLutier;

EXCEPTION

    WHEN NO_DATA_FOUND THEN

        DBMS_OUTPUT.PUT_LINE('lutierLocal: Atentie! In produs,
cod_magazin nu trebuie sa fie null. Posibil sa nu fie un lutier in judetul
magazinului.');
```

```

        RETURN -1;

    WHEN TOO_MANY_ROWS THEN

        DBMS_OUTPUT.PUT_LINE('lutierLocal: Mai multi lutieri sunt la
dispozitie. A fost returnat primul gasit.');
```

```

        SELECT lut.cod_lutier
        BULK COLLECT INTO tblCoduriLutieri
        FROM lutier lut, locatie loc
        WHERE loc.judet = judetLocatie      -- judetul gasit
        AND lut.cod_locatie = loc.cod_locatie; -- join intre lutier si locatie

        codLutier := tblCoduriLutieri(1);

        RETURN codLutier;

    END lutierLocal;

END musicShopExtensii;
```

/

```
SELECT * FROM livrare;
```

```
SELECT * FROM transport;
```

```
SELECT cod_produs produs, cod_livrare livrare, cod_magazin magazin FROM  
produs;
```

```
DECLARE
```

```
    coduriLivrari    musicShopExtensii.varrayCoduriLivrari :=  
musicShopExtensii.varrayCoduriLivrari();
```

```
    coduriVehicule  musicShopExtensii.tblCoduriVehicule  :=  
musicShopExtensii.tblCoduriVehicule();
```

```
    coduriSoferi    musicShopExtensii.tblCoduriSoferi    :=  
musicShopExtensii.tblCoduriSoferi();
```

```
BEGIN
```

```
    SELECT cod_livrare
```

```
    BULK COLLECT INTO coduriLivrari
```

```
    FROM livrare
```

```
    WHERE cod_livrare IN (1,2,3, 5,6, 7);
```

```
    musicShopExtensii.elibereazaVehiculeSoferi(4, coduriLivrari, coduriVehicule,  
coduriSoferi);
```

```
    DBMS_OUTPUT.PUT('Nr inmatriculare vehicule eliberate: ');
```

```
    FOR i IN coduriVehicule.FIRST..coduriVehicule.LAST LOOP
```

```
        DBMS_OUTPUT.PUT(coduriVehicule(i) || ' ');
```

END LOOP;

DBMS_OUTPUT.NEW_LINE;

DBMS_OUTPUT.PUT('Coduri soferi eliberati: ');

FOR i IN coduriSoferi.FIRST..coduriSoferi.LAST LOOP

DBMS_OUTPUT.PUT(coduriSoferi(i) || ' ');

END LOOP;

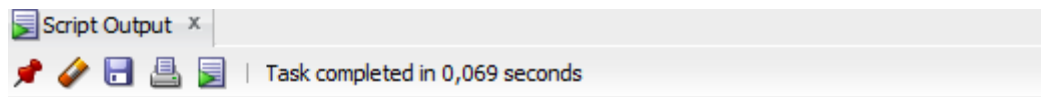
DBMS_OUTPUT.NEW_LINE;

END;

/

SELECT * FROM livrare;

SELECT cod_produs produs, cod_livrare livrare, cod_magazin magazin FROM
produs;



Package MUSICSHOPEXTENSII compiled

Package Body MUSICSHOPEXTENSII compiled

COD_LIVRARE COD_TRANSPORT COD_MAGAZIN STATUS_LIVRARE

1	1	4 pe drum
2	2	4 pe drum
3	3	3 pe drum
4	4	3 pe drum
5	5	5 pe drum
6	6	1 pe drum
7	7	1 pe drum
8	8	2 pe drum
9	9	4 pe drum
10	10	1 pe drum

10 rows selected.

COD_TRANSPORT NR_INMAT COD_SOFER DATA_PLECA ORA_PLEcare

1 AG 198 IBY	7 25-05-2021	700
2 AG 198 IBY	7 06-06-2021	800
3 DJ 131 ANA	1 30-05-2021	930
4 DJ 131 ANA	1 17-06-2021	
5 AG 198 IBY	7 28-06-2021	
6 DJ 131 ANA	7 19-05-2021	
7 IF 09 TMY	4 28-06-2021	
8 SB 672 CMN	5 01-07-2021	1200
9 SB 672 CMN	5 01-07-2021	1930
10 CT 116 MLG	6 01-07-2021	2230

10 rows selected.

PRODUS	LIVRARE	MAGAZIN
--------	---------	---------

1	4
2	4
3	4
4	4
5	4
6	4
7	4
8	3
9	2
10	2
11	1

PRODUS	LIVRARE	MAGAZIN
--------	---------	---------

12	1
13	3
14	2
15	
16	
17	
18	
19	
20	
30	
31	

PRODUS	LIVRARE	MAGAZIN
--------	---------	---------

32	
33	
34	
35	
36	
37	
21	

Script Output x

Task completed in 0,0

22
23
24
25

PRODUS	LIVRARE	MAGAZIN
--------	---------	---------

26		
27		
28		
29		
38		
39		
40		
41		
42		
43	3	
44	2	

PRODUS	LIVRARE	MAGAZIN
--------	---------	---------

45	1	
46	5	
47	1	
48	6	
49	2	
50	3	
51	2	
52	1	
53	2	
54	3	
55	6	

PRODUS	LIVRARE	MAGAZIN
--------	---------	---------

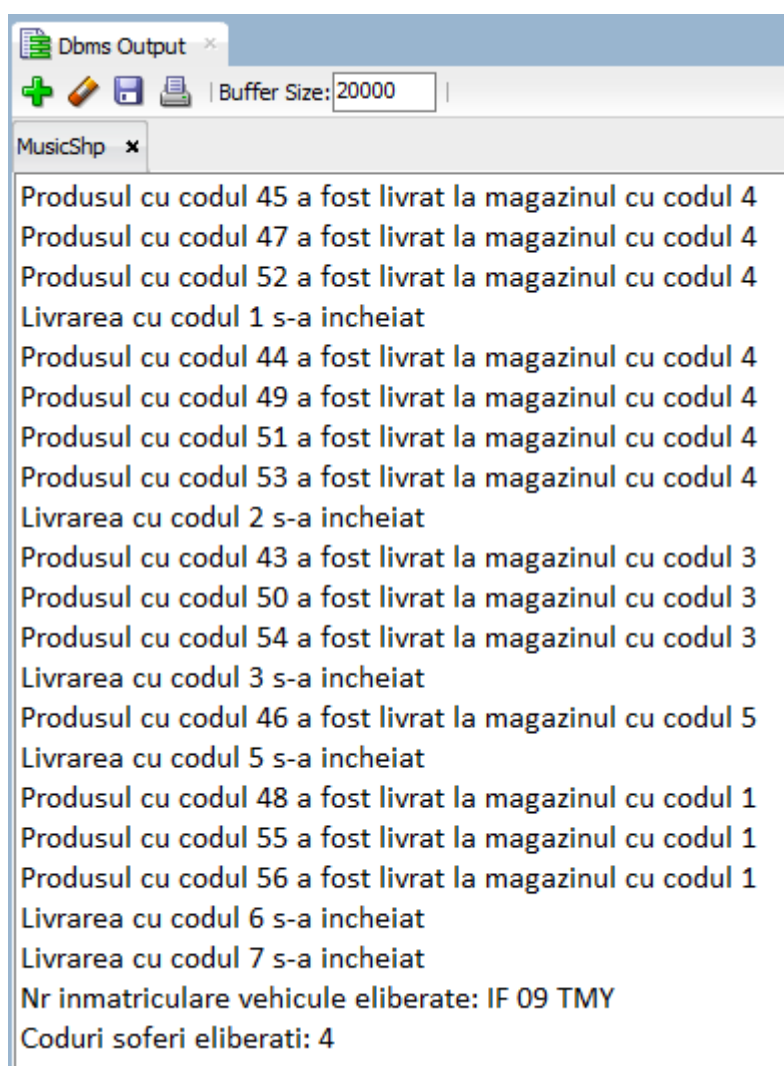
56	6	
----	---	--

Script Output x

Task completed in 0,069 seconds

56 rows selected.

PL/SQL procedure successfully completed.



Dbms Output x

+ | Buffer Size: 20000 |

MusicShp x

Produsul cu codul 45 a fost livrat la magazinul cu codul 4
Produsul cu codul 47 a fost livrat la magazinul cu codul 4
Produsul cu codul 52 a fost livrat la magazinul cu codul 4
Livrarea cu codul 1 s-a incheiat
Produsul cu codul 44 a fost livrat la magazinul cu codul 4
Produsul cu codul 49 a fost livrat la magazinul cu codul 4
Produsul cu codul 51 a fost livrat la magazinul cu codul 4
Produsul cu codul 53 a fost livrat la magazinul cu codul 4
Livrarea cu codul 2 s-a incheiat
Produsul cu codul 43 a fost livrat la magazinul cu codul 3
Produsul cu codul 50 a fost livrat la magazinul cu codul 3
Produsul cu codul 54 a fost livrat la magazinul cu codul 3
Livrarea cu codul 3 s-a incheiat
Produsul cu codul 46 a fost livrat la magazinul cu codul 5
Livrarea cu codul 5 s-a incheiat
Produsul cu codul 48 a fost livrat la magazinul cu codul 1
Produsul cu codul 55 a fost livrat la magazinul cu codul 1
Produsul cu codul 56 a fost livrat la magazinul cu codul 1
Livrarea cu codul 6 s-a incheiat
Livrarea cu codul 7 s-a incheiat
Nr inmatriculare vehicule eliberate: IF 09 TMY
Coduri soferi eliberati: 4

Script Output x

Task completed in 0,069 seconds

COD_LIVRARE COD_TRANSPORT COD_MAGAZIN STATUS_LIVRARE

1	1	4 sosita
2	2	4 sosita
3	3	3 sosita
4	4	3 pe drum
5	5	5 sosita
6	6	1 sosita
7	7	1 sosita
8	8	2 pe drum
9	9	4 pe drum
10	10	1 pe drum

10 rows selected.






PRODUS	LIVRARE	MAGAZIN

1	4	
2	4	
3	4	
4	4	
5	4	
6	4	
7	4	
8	3	
9	2	
10	2	
11	1	

PRODUS	LIVRARE	MAGAZIN

12	1	
13	3	
14	2	
15		
16		

Script Output x



Task completed in 0,069 seconds

16
17
18
19
20
30
31

PRODUS	LIVRARE	MAGAZIN
--------	---------	---------

32		
33		
34		
35		
36		
37		
21		
22		
23		
24		
25		

PRODUS	LIVRARE	MAGAZIN
--------	---------	---------

26		
27		
28		
29		
38		
39		
40		
41		
42		
43	3	
44	4	

PRODUS	LIVRARE	MAGAZIN
--------	---------	---------

PRODUS	LIVRARE	MAGAZIN
45	4	
46	5	
47	4	
48	1	
49	4	
50	3	
51	4	
52	4	
53	4	
54	3	
55	1	

PRODUS	LIVRARE	MAGAZIN
56	1	

56 rows selected.