

Baza de date

„FonoDesign”

Nițe Dan-Alexandru

Grupa 244, Seria 24

Anul universitar 2024-2025

Cuprins

1.Prezentarea pe scurt a bazei de date..................................................................................2

2.Realizarea diagramei entitate-relație corespunzătoare bazei de date................................5

3.Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație................6

4.Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea..............7

6. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze toate cele 3 tipuri de colecții studiate.......................46

7. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat, dependent de celălalt cursor.........................................49

8. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele create. Tratați toate excepțiile care pot apărea, incluzând excepțiile predefinite NO\_DATA\_FOUND și TOO\_MANY\_ROWS.................................................................53

9. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip procedură care să aibă minim 2 parametri și să utilizeze într-o singură comandă SQL 5 dintre tabelele create...................................................................56

10. Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.................59

11. Definiți un trigger de tip LMD la nivel de linie. Declanșați trigger-ul........................60

12. Definiți un trigger de tip LDD. Declanșați trigger-ul..................................................61

Baza de date a unei firme care se ocupă de producerea si comercializarea difuzoarelor acustice

**Tema proiectului:** Dezvoltarea unei baze de date destinată să optimizeze gestionarea si organizarea datelor associate cu sistemul de fabricație și distribuție a difuzoarelor. Aceasta va include detalii despre produse, componente, loturi de producție, clienți, comenzi și expediții. Produsele sunt clasificate în diverse modele, fiecare cu specificareii tehnice proprii. Componentele necesare pentru fabricația difuzoarelor provin de la diferiți furnizori și sunt gestionate într-un inventar detaliat.

Loturile de producție sunt planificate pentru a optimiza fluxul de fabricație și sunt asociate cu date specifice de începere și terminare. Comenzile de la clienți sunt procesate în sistem, fiecare comandă fiind urmărită de la plasare până la finalizare, inclusiv detaliile expedițiilor și statusul acestora. Fiecare expediție este asociată cu un vehicul din departamentul de transport și un curier, asigurându-se că produsele ajung la destinație în mod eficient.

### 1.Prezentarea pe scurt a bazei de date

Această bază de date este destinată să optimizeze gestionarea și organizarea datelor asociate cu sistemul de fabricație și distribuție a difuzoarelor. Modelul include tabele care stochează informații despre furnizori, produse, componente, inventar, loturi de producție, clienți, comenzi și expediții.

#### Entitățile și relațiile principale sunt:

* **Furnizori**: Stochează informații despre furnizorii de componente, inclusiv nume, adresă și email.
* **Componente**: Include detalii despre componentele utilizate în fabricația difuzoarelor, fiecare componentă fiind asociată cu un furnizor specific.
* **Produs**: Conține informații despre produsele finale, inclusiv numele, descrierea și tipul produsului.
* **Inventar**: Găzduiește date despre stocurile de componente, inclusiv cantitatea disponibilă și cea comandată.
* **Loturi\_Producție**: Gestionează planificarea producției, inclusiv datele de începere și terminare ale fiecărui lot.
* **Client**: Înregistrează informații despre clienți, inclusiv numele, prenumele, adresa, telefonul și emailul.
* **Comenzi**: Urmărește comenzile de la clienți, inclusiv data comenzii, totalul și statusul.
* **Detalii\_Comenzi**: Include detalii despre produsele comandate, cantitatea și prețul per unitate.
* **Expedieri**: Gestionează expedițiile comenzilor, inclusiv data expedierii, data estimată de livrare și statusul.
* **Departament\_Transport**: Conține informații despre vehiculele din departamentul de transport, inclusiv modelul, marca, numărul de licență și statusul.
* **Curier**: Stochează detalii despre curieri, inclusiv numele, prenumele, telefonul și emailul, precum și vehiculul alocat.

Utilitatea acestei baze de date constă în îmbunătățirea eficienței gestionării și organizării datelor referitoare la fabricația și distribuția difuzoarelor.

#### Beneficiile includ:

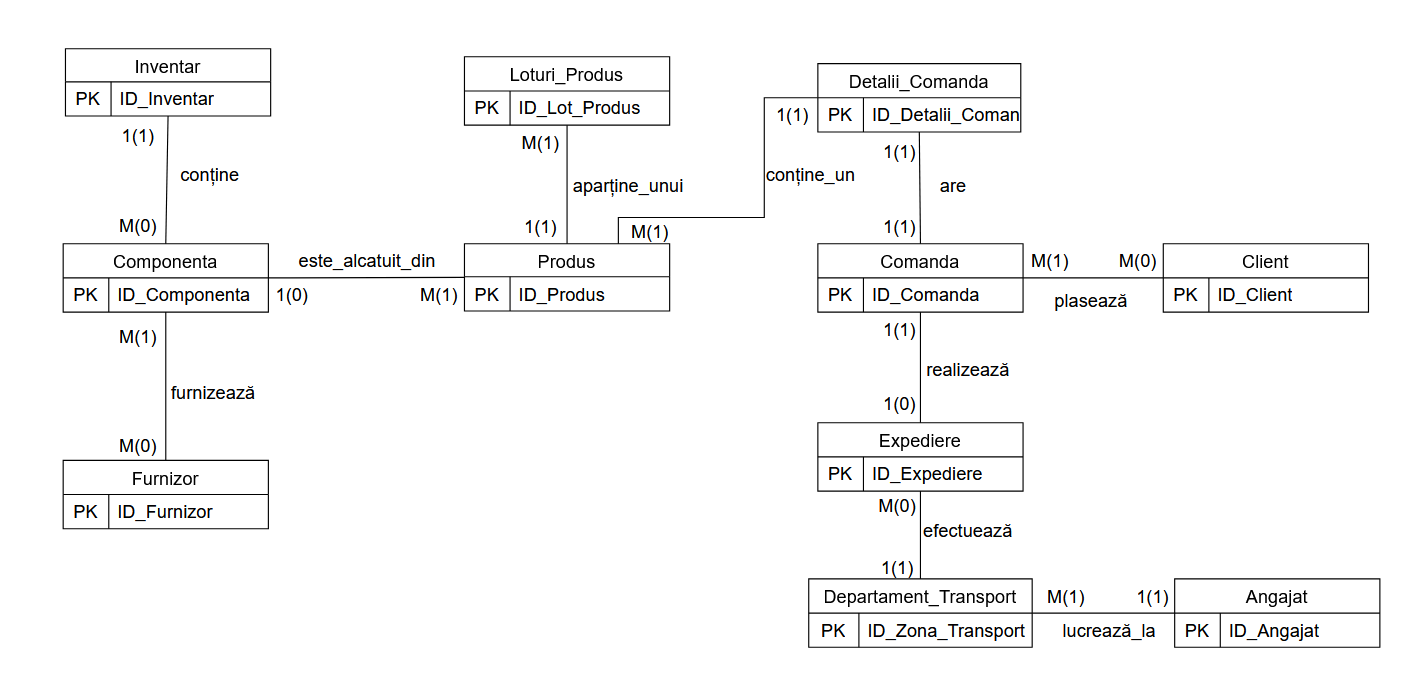
* **Optimizarea producției**: Planificarea eficientă a loturilor de producție pentru a maximiza fluxul de fabricație și a reduce timpul de inactivitate.
* **Gestionarea inventarului**: Monitorizarea stocurilor de componente pentru a preveni supra-stocarea sau deficitul de materiale.
* **Urmărirea comenzilor**: Gestionarea completă a comenzilor de la plasare până la finalizare, incluzând expedițiile.
* **Eficientizarea transportului**: Alocarea eficientă a vehiculelor și curierilor pentru a asigura livrarea promptă a produselor.
* **Rapoarte detaliate**: Generarea de rapoarte privind producția, stocurile, comenzile și expedițiile, utile pentru analiza performanței și luarea deciziilor informate.

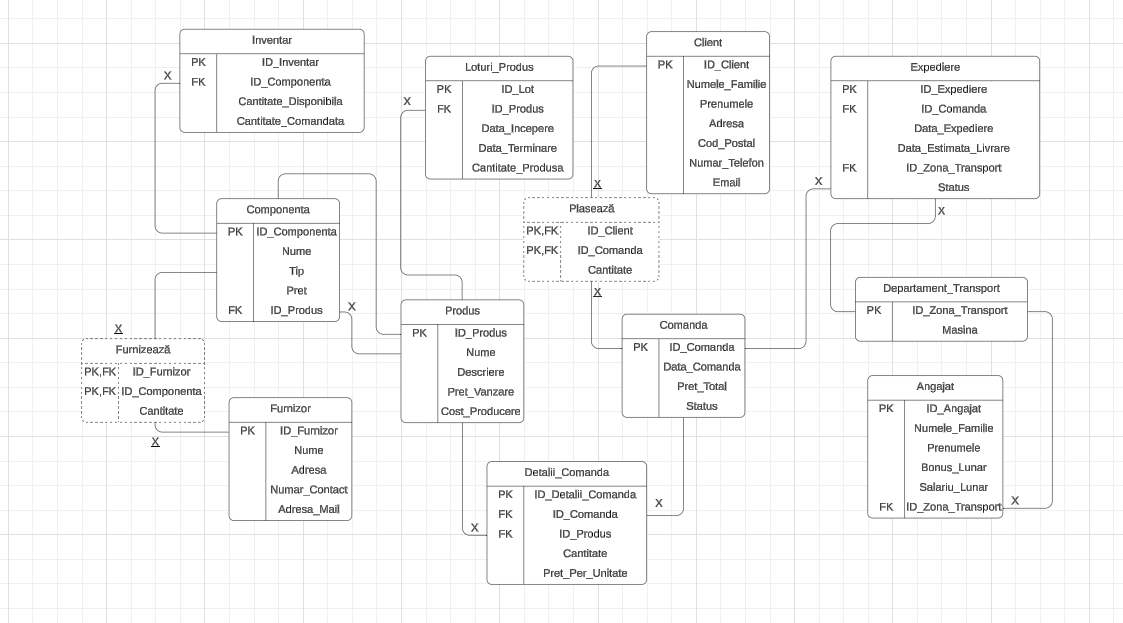
Funcționalitățile bazei de date includ:

* **Stocarea și gestionarea datelor**: Informații detaliate despre furnizori, produse, componente și inventar.
* **Planificarea producției**: Gestionarea loturilor de producție, incluzând datele de începere și terminare.
* **Urmărirea comenzilor**: Înregistrarea și monitorizarea comenzilor de la clienți, inclusiv detaliile expedițiilor și statusul acestora.
* **Administrarea transportului**: Gestionarea vehiculelor și curierilor, incluzând alocarea vehiculelor pentru expediții.
* **Generarea rapoartelor**: Crearea de rapoarte detaliate despre producție, inventar, comenzi și expediții.

Am realizat și testat proiectul folosind Oracle Database 21c pe un sistem cu Windows 11 Pro, dispunând de 16 GB RAM. Pentru administrarea și interogarea bazei de date, am utilizat mediul de dezvoltare Oracle SQL Developer.

2.Realizarea diagramei entitate-relație corespunzătoare bazei de date



3.Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație

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

Codul sursă SQL:

CREATE SEQUENCE Furnizor\_SEQ START WITH 1;

CREATE SEQUENCE Componenta\_SEQ START WITH 1;

CREATE SEQUENCE Produs\_SEQ START WITH 1;

CREATE SEQUENCE Loturi\_Produs\_SEQ START WITH 1;

CREATE SEQUENCE Inventar\_SEQ START WITH 1;

CREATE SEQUENCE Client\_SEQ START WITH 1;

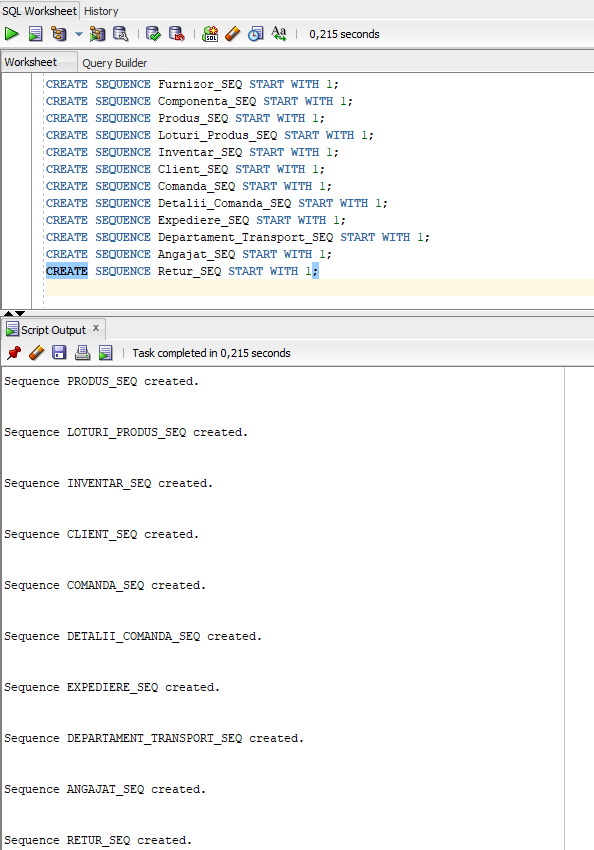
CREATE SEQUENCE Comanda\_SEQ START WITH 1;

CREATE SEQUENCE Detalii\_Comanda\_SEQ START WITH 1;

CREATE SEQUENCE Expediere\_SEQ START WITH 1;

CREATE SEQUENCE Departament\_Transport\_SEQ START WITH 1;

CREATE SEQUENCE Angajat\_SEQ START WITH 1;

PrintScreen cu execuția și cu rezultatul:

4.Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative; maxim 30 de înregistrări în fiecare tabel).

Pentru tabelul **Furnizor**:

*Codul SQL:*

CREATE TABLE Furnizor (

ID\_Furnizor NUMBER DEFAULT Furnizor\_SEQ.NEXTVAL PRIMARY KEY,

Nume VARCHAR2(100),

Adresa VARCHAR2(255),

Numar\_Contact VARCHAR2(15),

Adresa\_Mail VARCHAR2(100)

);

INSERT INTO Furnizor (Nume, Adresa, Numar\_Contact, Adresa\_Mail) VALUES

('SC Sunet Premium SRL', 'Strada Muzicii, Nr. 10', '0722345678', 'contact@sunetpremium.com');

INSERT INTO Furnizor (Nume, Adresa, Numar\_Contact, Adresa\_Mail) VALUES

('Acoustic Solutions', 'Bulevardul Armoniei, Nr. 23', '0722345679', 'info@acousticsolutions.com');

INSERT INTO Furnizor (Nume, Adresa, Numar\_Contact, Adresa\_Mail) VALUES

('High Fidelity Acoustics', 'Calea Melodiei, Nr. 45', '0722345680', 'support@highfidelity.com');

INSERT INTO Furnizor (Nume, Adresa, Numar\_Contact, Adresa\_Mail) VALUES

('SC Amplify SRL', 'Strada Bass, Nr. 7', '0722345681', 'contact@amplify.com');

INSERT INTO Furnizor (Nume, Adresa, Numar\_Contact, Adresa\_Mail) VALUES

('VibraSound Inc.', 'Bulevardul Frecvențelor, Nr. 33', '0722345682', 'info@vibrasound.com');

INSERT INTO Furnizor (Nume, Adresa, Numar\_Contact, Adresa\_Mail) VALUES

('Pure Audio Systems', 'Aleea Decibelilor, Nr. 9', '0722345683', 'support@pureaudio.com');

INSERT INTO Furnizor (Nume, Adresa, Numar\_Contact, Adresa\_Mail) VALUES

('Waveform Electronics', 'Strada Rezonanței, Nr. 56', '0722345684', 'contact@waveform.com');

INSERT INTO Furnizor (Nume, Adresa, Numar\_Contact, Adresa\_Mail) VALUES

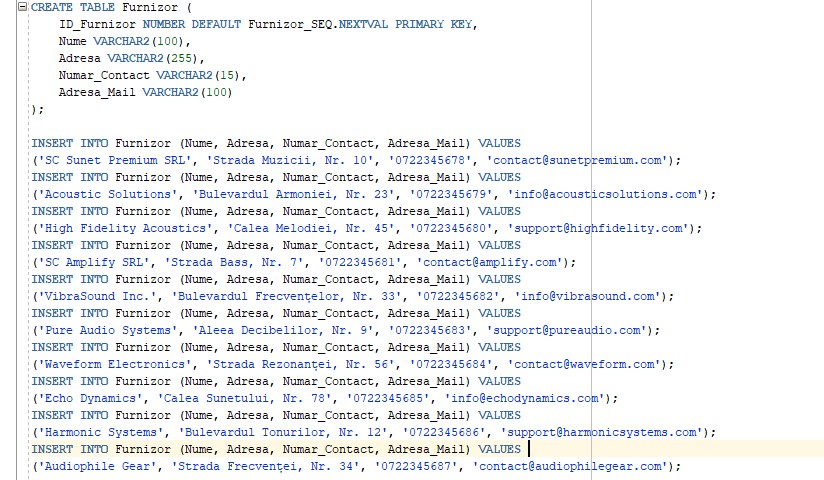
('Echo Dynamics', 'Calea Sunetului, Nr. 78', '0722345685', 'info@echodynamics.com');

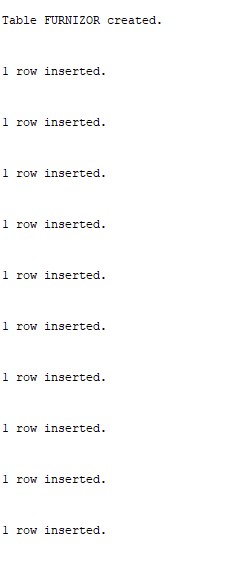
INSERT INTO Furnizor (Nume, Adresa, Numar\_Contact, Adresa\_Mail) VALUES

('Harmonic Systems', 'Bulevardul Tonurilor, Nr. 12', '0722345686', 'support@harmonicsystems.com');

INSERT INTO Furnizor (Nume, Adresa, Numar\_Contact, Adresa\_Mail) VALUES

('Audiophile Gear', 'Strada Frecvenței, Nr. 34', '0722345687', 'contact@audiophilegear.com');





Pentru tabelul **Produs**:

*Codul SQL:*

CREATE TABLE Produs (

ID\_Produs NUMBER DEFAULT Produs\_SEQ.NEXTVAL PRIMARY KEY,

Nume VARCHAR2(100),

Descriere CLOB,

Pret\_Vanzare NUMBER(10, 2),

Cost\_Productie NUMBER(10, 2)

);

INSERT INTO Produs (Nume, Descriere, Pret\_Vanzare, Cost\_Productie) VALUES

('EchoMaster', 'Difuzor de înaltă performanță, ideal pentru utilizare profesională.', 1200.00, 800.00);

INSERT INTO Produs (Nume, Descriere, Pret\_Vanzare, Cost\_Productie) VALUES

('SoundWave', 'Difuzor compact, perfect pentru sisteme audio casnice.', 800.00, 500.00);

INSERT INTO Produs (Nume, Descriere, Pret\_Vanzare, Cost\_Productie) VALUES

('BassBoom', 'Difuzor de medii, optimizat pentru claritatea sunetului.', 950.00, 600.00);

INSERT INTO Produs (Nume, Descriere, Pret\_Vanzare, Cost\_Productie) VALUES

('TrebleMax', 'Difuzor de bas cu răspuns rapid și profund.', 1300.00, 900.00);

INSERT INTO Produs (Nume, Descriere, Pret\_Vanzare, Cost\_Productie) VALUES

('PortablePro', 'Difuzor portabil, cu baterie încorporată și conectivitate Bluetooth.', 700.00, 400.00);

INSERT INTO Produs (Nume, Descriere, Pret\_Vanzare, Cost\_Productie) VALUES

('OutdoorX', 'Difuzor de exterior, rezistent la intemperii.', 1100.00, 700.00);

INSERT INTO Produs (Nume, Descriere, Pret\_Vanzare, Cost\_Productie) VALUES

('AllSound', 'Sistem audio all-in-one, cu amplificator integrat.', 1500.00, 1000.00);

INSERT INTO Produs (Nume, Descriere, Pret\_Vanzare, Cost\_Productie) VALUES

('MegaSound', 'Difuzor de coloană, pentru utilizare în spații mari.', 2000.00, 1300.00);

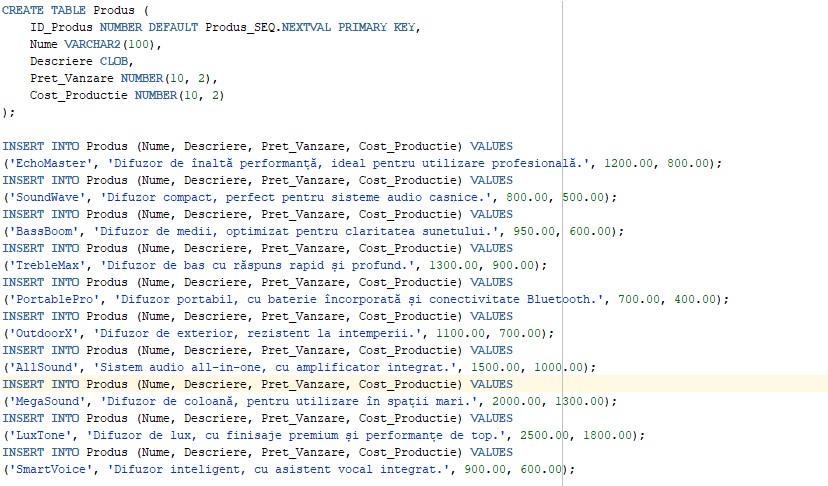
INSERT INTO Produs (Nume, Descriere, Pret\_Vanzare, Cost\_Productie) VALUES

('LuxTone', 'Difuzor de lux, cu finisaje premium și performanțe de top.', 2500.00, 1800.00);

INSERT INTO Produs (Nume, Descriere, Pret\_Vanzare, Cost\_Productie) VALUES

('SmartVoice', 'Difuzor inteligent, cu asistent vocal integrat.', 900.00, 600.00);

PrintScreen cu codul si cu rezultatul executiei:





Pentru tabelul **Componente**:

*Codul SQL:*

CREATE TABLE Componenta (

ID\_Componenta NUMBER DEFAULT Componenta\_SEQ.NEXTVAL PRIMARY KEY,

Nume VARCHAR2(100),

Tip VARCHAR2(50),

Pret NUMBER(10, 2),

ID\_Furnizor NUMBER,

ID\_Produs NUMBER,

CONSTRAINT fk\_furnizor FOREIGN KEY (ID\_Furnizor) REFERENCES Furnizor(ID\_Furnizor),

CONSTRAINT fk\_produs FOREIGN KEY (ID\_Produs) REFERENCES Produs(ID\_Produs)

);

INSERT INTO Componenta (Nume, Tip, Pret, ID\_Furnizor, ID\_Produs) VALUES

('Difuzor de înalte', 'Tweeter', 150.00, 1, 1);

INSERT INTO Componenta (Nume, Tip, Pret, ID\_Furnizor, ID\_Produs) VALUES

('Difuzor de medii', 'Midrange', 200.00, 2, 2);

INSERT INTO Componenta (Nume, Tip, Pret, ID\_Furnizor, ID\_Produs) VALUES

('Difuzor de bas', 'Woofer', 250.00, 3, 3);

INSERT INTO Componenta (Nume, Tip, Pret, ID\_Furnizor, ID\_Produs) VALUES

('Crossover pasiv', 'Crossover', 100.00, 4, 4);

INSERT INTO Componenta (Nume, Tip, Pret, ID\_Furnizor, ID\_Produs) VALUES

('Amplificator intern', 'Amplifier', 300.00, 5, 5);

INSERT INTO Componenta (Nume, Tip, Pret, ID\_Furnizor, ID\_Produs) VALUES

('Cablu de conexiune', 'Cable', 50.00, 1, 2);

INSERT INTO Componenta (Nume, Tip, Pret, ID\_Furnizor, ID\_Produs) VALUES

('Carcasă difuzor', 'Cabinet', 400.00, 2, 3);

INSERT INTO Componenta (Nume, Tip, Pret, ID\_Furnizor, ID\_Produs) VALUES

('Panou de control', 'Control Panel', 350.00, 3, 4);

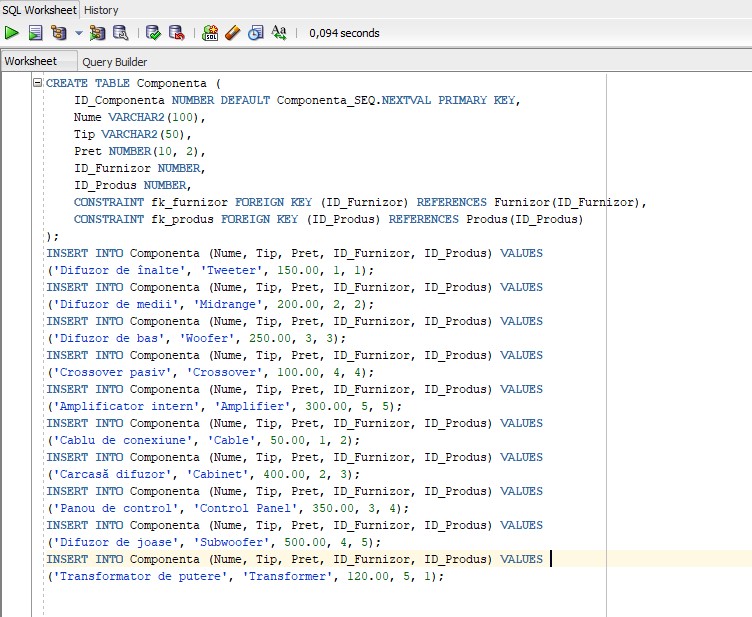
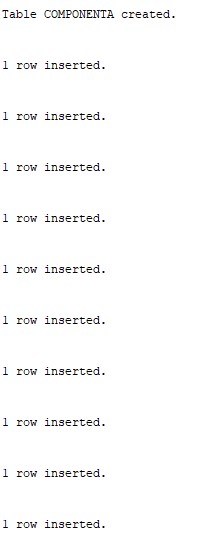
INSERT INTO Componenta (Nume, Tip, Pret, ID\_Furnizor, ID\_Produs) VALUES

('Difuzor de joase', 'Subwoofer', 500.00, 4, 5);

INSERT INTO Componenta (Nume, Tip, Pret, ID\_Furnizor, ID\_Produs) VALUES

('Transformator de putere', 'Transformer', 120.00, 5, 1);

*PrintScreen cu codul si cu rezultatul executiei lui:*



Pentru tabelul Loturi\_Produs:

*Codul SQL:*

CREATE TABLE Loturi\_Produs (

ID\_Lot NUMBER DEFAULT Loturi\_Produs\_SEQ.NEXTVAL PRIMARY KEY,

ID\_Produs NUMBER,

Data\_Incepere DATE,

Data\_Terminare DATE,

Cantitate\_Produsa NUMBER,

CONSTRAINT fk\_produs\_loturi\_produs FOREIGN KEY (ID\_Produs) REFERENCES Produs(ID\_Produs)

);

INSERT INTO Loturi\_Produs (ID\_Produs, Data\_Incepere, Data\_Terminare, Cantitate\_Produsa) VALUES

(1, TO\_DATE('01-01-2023', 'DD-MM-YYYY'), TO\_DATE('10-01-2023', 'DD-MM-YYYY'), 100);

INSERT INTO Loturi\_Produs (ID\_Produs, Data\_Incepere, Data\_Terminare, Cantitate\_Produsa) VALUES

(2, TO\_DATE('01-02-2023', 'DD-MM-YYYY'), TO\_DATE('10-02-2023', 'DD-MM-YYYY'), 150);

INSERT INTO Loturi\_Produs (ID\_Produs, Data\_Incepere, Data\_Terminare, Cantitate\_Produsa) VALUES

(3, TO\_DATE('01-03-2023', 'DD-MM-YYYY'), TO\_DATE('10-03-2023', 'DD-MM-YYYY'), 200);

INSERT INTO Loturi\_Produs (ID\_Produs, Data\_Incepere, Data\_Terminare, Cantitate\_Produsa) VALUES

(4, TO\_DATE('01-04-2023', 'DD-MM-YYYY'), TO\_DATE('10-04-2023', 'DD-MM-YYYY'), 250);

INSERT INTO Loturi\_Produs (ID\_Produs, Data\_Incepere, Data\_Terminare, Cantitate\_Produsa) VALUES

(5, TO\_DATE('01-05-2023', 'DD-MM-YYYY'), TO\_DATE('10-05-2023', 'DD-MM-YYYY'), 300);

INSERT INTO Loturi\_Produs (ID\_Produs, Data\_Incepere, Data\_Terminare, Cantitate\_Produsa) VALUES

(6, TO\_DATE('01-06-2023', 'DD-MM-YYYY'), TO\_DATE('10-06-2023', 'DD-MM-YYYY'), 350);

INSERT INTO Loturi\_Produs (ID\_Produs, Data\_Incepere, Data\_Terminare, Cantitate\_Produsa) VALUES

(7, TO\_DATE('01-07-2023', 'DD-MM-YYYY'), TO\_DATE('10-07-2023', 'DD-MM-YYYY'), 400);

INSERT INTO Loturi\_Produs (ID\_Produs, Data\_Incepere, Data\_Terminare, Cantitate\_Produsa) VALUES

(8, TO\_DATE('01-08-2023', 'DD-MM-YYYY'), TO\_DATE('10-08-2023', 'DD-MM-YYYY'), 450);

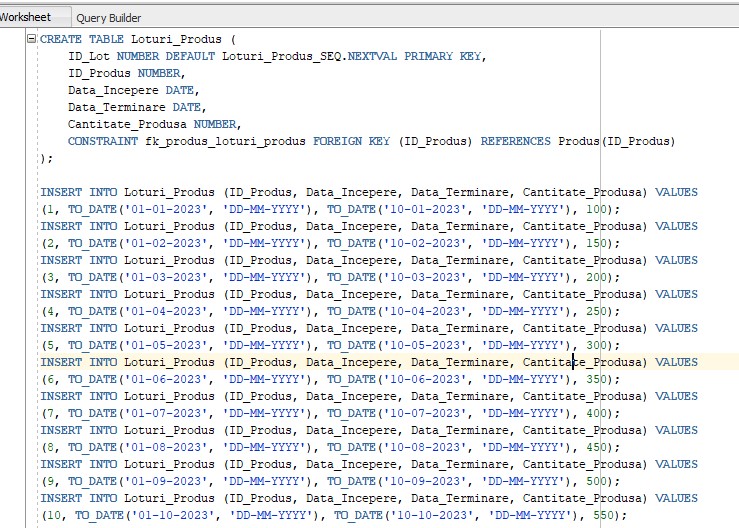
INSERT INTO Loturi\_Produs (ID\_Produs, Data\_Incepere, Data\_Terminare, Cantitate\_Produsa) VALUES

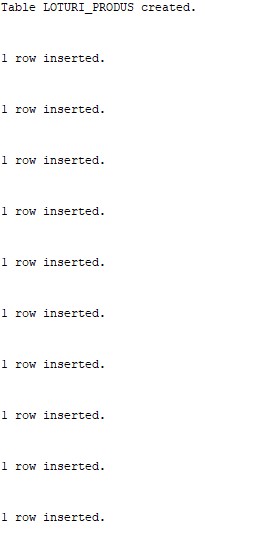
(9, TO\_DATE('01-09-2023', 'DD-MM-YYYY'), TO\_DATE('10-09-2023', 'DD-MM-YYYY'), 500);

INSERT INTO Loturi\_Produs (ID\_Produs, Data\_Incepere, Data\_Terminare, Cantitate\_Produsa) VALUES

(10, TO\_DATE('01-10-2023', 'DD-MM-YYYY'), TO\_DATE('10-10-2023', 'DD-MM-YYYY'), 550);

*PrintScreen cu codul si cu rezultatul executiei lui:*





Pentru tabelul **Inventar**:

*Codul SQL:*

CREATE TABLE Inventar (

ID\_Inventar NUMBER DEFAULT Inventar\_SEQ.NEXTVAL PRIMARY KEY,

ID\_Componenta NUMBER,

Cantitate\_Disponibila NUMBER,

Cantitate\_Comandata NUMBER,

CONSTRAINT fk\_componenta\_inventar FOREIGN KEY (ID\_Componenta) REFERENCES Componenta(ID\_Componenta)

);

INSERT INTO Inventar (ID\_Componenta, Cantitate\_Disponibila, Cantitate\_Comandata) VALUES

(1, 500, 100);

INSERT INTO Inventar (ID\_Componenta, Cantitate\_Disponibila, Cantitate\_Comandata) VALUES

(2, 600, 150);

INSERT INTO Inventar (ID\_Componenta, Cantitate\_Disponibila, Cantitate\_Comandata) VALUES

(3, 700, 200);

INSERT INTO Inventar (ID\_Componenta, Cantitate\_Disponibila, Cantitate\_Comandata) VALUES

(4, 800, 250);

INSERT INTO Inventar (ID\_Componenta, Cantitate\_Disponibila, Cantitate\_Comandata) VALUES

(5, 900, 300);

INSERT INTO Inventar (ID\_Componenta, Cantitate\_Disponibila, Cantitate\_Comandata) VALUES

(6, 400, 100);

INSERT INTO Inventar (ID\_Componenta, Cantitate\_Disponibila, Cantitate\_Comandata) VALUES

(7, 350, 150);

INSERT INTO Inventar (ID\_Componenta, Cantitate\_Disponibila, Cantitate\_Comandata) VALUES

(8, 450, 200);

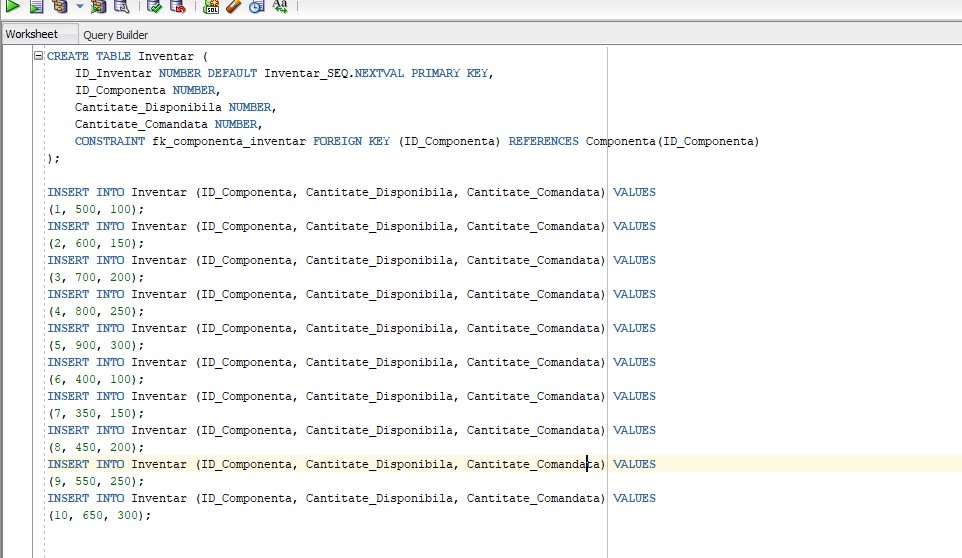
INSERT INTO Inventar (ID\_Componenta, Cantitate\_Disponibila, Cantitate\_Comandata) VALUES

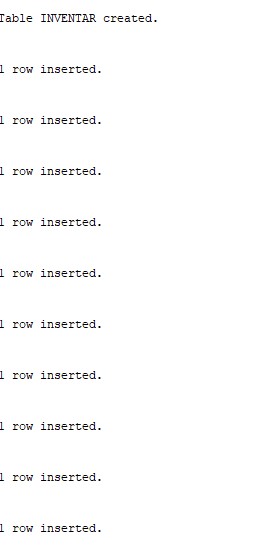
(9, 550, 250);

INSERT INTO Inventar (ID\_Componenta, Cantitate\_Disponibila, Cantitate\_Comandata) VALUES

(10, 650, 300);

*PrintScreen cu codul si cu rezultatul executiei lui:*





Pentru tabelul Client:

*Codul SQL:*

CREATE TABLE Client (

ID\_Client NUMBER DEFAULT Client\_SEQ.NEXTVAL PRIMARY KEY,

Numele\_Familie VARCHAR2(100),

Prenumele VARCHAR2(100),

Adresa VARCHAR2(255),

Cod\_Postal VARCHAR2(10),

Numar\_Telefon VARCHAR2(15),

Email VARCHAR2(100)

);

INSERT INTO Client (Numele\_Familie, Prenumele, Adresa, Cod\_Postal, Numar\_Telefon, Email) VALUES

('Popescu', 'Ion', 'Strada Fictiva, Nr. 10', '010101', '0712345678', 'ion.popescu@example.com');

INSERT INTO Client (Numele\_Familie, Prenumele, Adresa, Cod\_Postal, Numar\_Telefon, Email) VALUES

('Ionescu', 'Maria', 'Bulevardul Inventat, Nr. 23', '020202', '0723456789', 'maria.ionescu@example.com');

INSERT INTO Client (Numele\_Familie, Prenumele, Adresa, Cod\_Postal, Numar\_Telefon, Email) VALUES

('Georgescu', 'Andrei', 'Calea Imaginara, Nr. 45', '030303', '0734567890', 'andrei.georgescu@example.com');

INSERT INTO Client (Numele\_Familie, Prenumele, Adresa, Cod\_Postal, Numar\_Telefon, Email) VALUES

('Vasilescu', 'Ana', 'Strada Noua, Nr. 7', '040404', '0745678901', 'ana.vasilescu@example.com');

INSERT INTO Client (Numele\_Familie, Prenumele, Adresa, Cod\_Postal, Numar\_Telefon, Email) VALUES

('Dumitrescu', 'Dan', 'Aleea Fictiunii, Nr. 33', '050505', '0756789012', 'dan.dumitrescu@example.com');

INSERT INTO Client (Numele\_Familie, Prenumele, Adresa, Cod\_Postal, Numar\_Telefon, Email) VALUES

('Marinescu', 'Elena', 'Piata Inchipuita, Nr. 5', '060606', '0767890123', 'elena.marinescu@example.com');

INSERT INTO Client (Numele\_Familie, Prenumele, Adresa, Cod\_Postal, Numar\_Telefon, Email) VALUES

('Stanciu', 'Florin', 'Parcul Inexistent, Nr. 12', '070707', '0778901234', 'florin.stanciu@example.com');

INSERT INTO Client (Numele\_Familie, Prenumele, Adresa, Cod\_Postal, Numar\_Telefon, Email) VALUES

('Radu', 'Ioana', 'Bulevardul Fantaziei, Nr. 19', '080808', '0789012345', 'ioana.radu@example.com');

INSERT INTO Client (Numele\_Familie, Prenumele, Adresa, Cod\_Postal, Numar\_Telefon, Email) VALUES

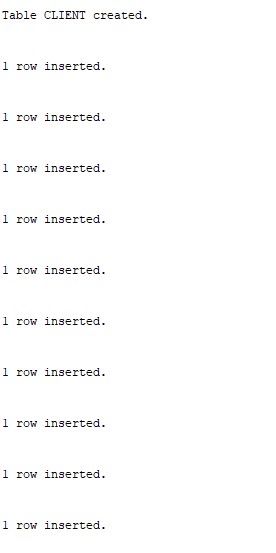
('Preda', 'Mihai', 'Strada Misterului, Nr. 21', '090909', '0790123456', 'mihai.preda@example.com');

INSERT INTO Client (Numele\_Familie, Prenumele, Adresa, Cod\_Postal, Numar\_Telefon, Email) VALUES

('Iliescu', 'Carmen', 'Aleea Iluziilor, Nr. 27', '100101', '0701234567', 'carmen.iliescu@example.com');

*PrintScreen cu codul si cu rezultatul executiei lui:*





Pentru tabelul **Comanda**:

*Codul SQL:*

CREATE TABLE Comanda (

ID\_Comanda NUMBER DEFAULT Comanda\_SEQ.NEXTVAL PRIMARY KEY,

Data\_Comanda DATE,

Pret\_Total NUMBER(10, 2),

Status VARCHAR2(50)

);

INSERT INTO Comanda (Data\_Comanda, Pret\_Total, Status) VALUES

(1, TO\_DATE('01-01-2023', 'DD-MM-YYYY'), 1000.00, 'procesată');

INSERT INTO Comanda (Data\_Comanda, Pret\_Total, Status) VALUES

(TO\_DATE('02-01-2023', 'DD-MM-YYYY'), 1200.00, 'expediată');

INSERT INTO Comanda (Data\_Comanda, Pret\_Total, Status) VALUES

(TO\_DATE('03-01-2023', 'DD-MM-YYYY'), 1400.00, 'finalizată');

INSERT INTO Comanda (Data\_Comanda, Pret\_Total, Status) VALUES

(TO\_DATE('04-01-2023', 'DD-MM-YYYY'), 1600.00, 'procesată');

INSERT INTO Comanda (Data\_Comanda, Pret\_Total, Status) VALUES

(TO\_DATE('05-01-2023', 'DD-MM-YYYY'), 1800.00, 'expediată');

INSERT INTO Comanda (Data\_Comanda, Pret\_Total, Status) VALUES

(TO\_DATE('06-01-2023', 'DD-MM-YYYY'), 2000.00, 'finalizată');

INSERT INTO Comanda (Data\_Comanda, Pret\_Total, Status) VALUES

(TO\_DATE('07-01-2023', 'DD-MM-YYYY'), 2200.00, 'procesată');

INSERT INTO Comanda (Data\_Comanda, Pret\_Total, Status) VALUES

(TO\_DATE('08-01-2023', 'DD-MM-YYYY'), 2400.00, 'expediată');

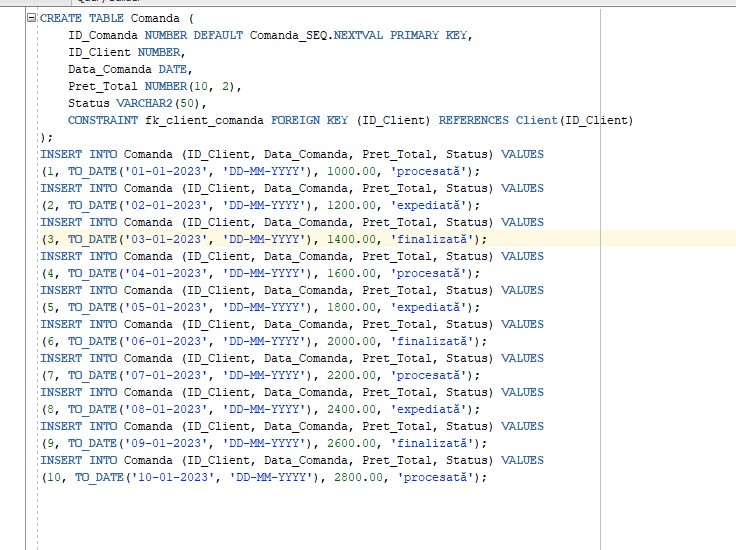
INSERT INTO Comanda (Data\_Comanda, Pret\_Total, Status) VALUES

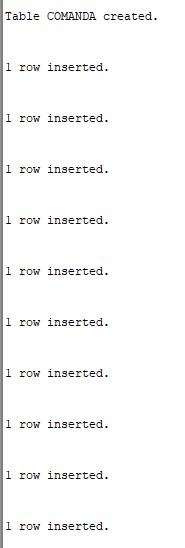
(TO\_DATE('09-01-2023', 'DD-MM-YYYY'), 2600.00, 'finalizată');

INSERT INTO Comanda (Data\_Comanda, Pret\_Total, Status) VALUES

(1TO\_DATE('10-01-2023', 'DD-MM-YYYY'), 2800.00, 'procesată');

*PrintScreen cu codul si cu rezultatul executiei lui:*





Pentru tabelul **Detalii\_Comanda**:

*Codul SQL:*

CREATE TABLE Detalii\_Comanda (

ID\_Detalii\_Comanda NUMBER DEFAULT Detalii\_Comanda\_SEQ.NEXTVAL PRIMARY KEY,

ID\_Comanda NUMBER,

ID\_Produs NUMBER,

Cantitate NUMBER,

Pret\_Per\_Unitate NUMBER(10, 2),

CONSTRAINT fk\_comanda\_detalii FOREIGN KEY (ID\_Comanda) REFERENCES Comanda(ID\_Comanda),

CONSTRAINT fk\_produs\_detalii FOREIGN KEY (ID\_Produs) REFERENCES Produs(ID\_Produs)

);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(1, 1, 10, 120.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(1, 2, 5, 160.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(2, 3, 8, 130.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(2, 4, 4, 110.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(3, 5, 12, 125.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(3, 1, 6, 100.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(4, 2, 9, 105.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(4, 3, 7, 150.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(5, 4, 10, 200.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(5, 5, 3, 250.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(6, 1, 5, 120.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(6, 2, 7, 160.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(7, 3, 9, 130.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(7, 4, 8, 110.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(8, 5, 6, 125.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(8, 1, 4, 100.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(9, 2, 3, 105.00);

INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(9, 3, 2, 150.00);

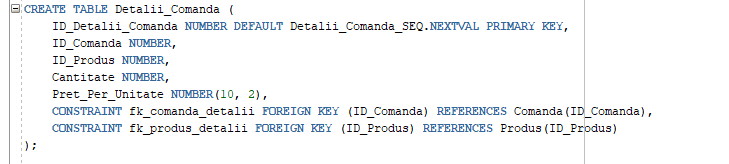
INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

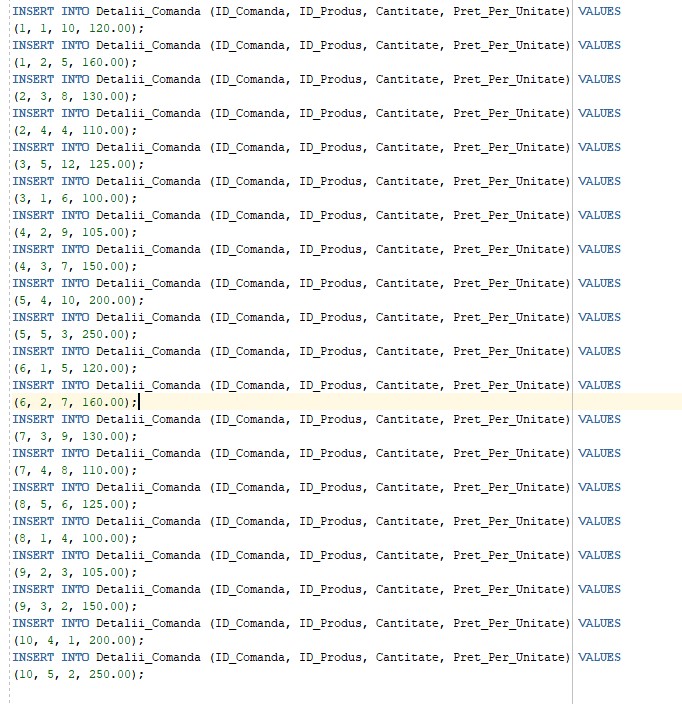
(10, 4, 1, 200.00);

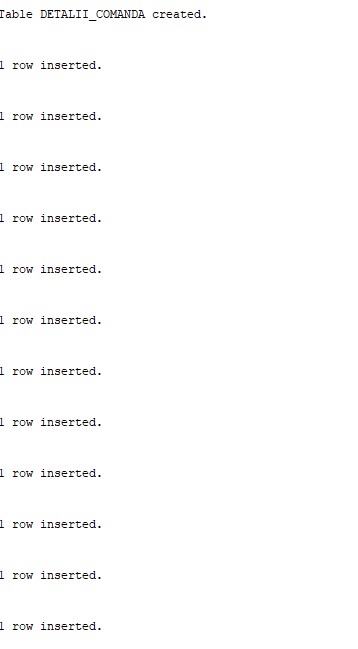
INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate) VALUES

(10, 5, 2, 250.00);

*PrintScreen cu codul si cu rezultatul executiei lui:*

**





Pentru tabelul **Expediere**:

*Codul SQL:*

CREATE TABLE Expediere (

ID\_Expediere NUMBER DEFAULT Expediere\_SEQ.NEXTVAL PRIMARY KEY,

ID\_Comanda NUMBER,

Data\_Expediere DATE,

Data\_Estimata\_Livrare DATE,

ID\_Zona\_Transport NUMBER,

Status VARCHAR2(50),

CONSTRAINT fk\_comanda FOREIGN KEY (ID\_Comanda) REFERENCES Comanda(ID\_Comanda),

CONSTRAINT fk\_zona\_transport FOREIGN KEY (ID\_Zona\_Transport) REFERENCES Departament\_Transport(ID\_Zona\_Transport)

);

INSERT INTO Expediere (ID\_Comanda, Data\_Expediere, Data\_Estimata\_Livrare, ID\_Zona\_Transport, Status) VALUES

(1, TO\_DATE('11-01-2023', 'DD-MM-YYYY'), TO\_DATE('15-01-2023', 'DD-MM-YYYY'), 1, 'în curs de livrare');

INSERT INTO Expediere (ID\_Comanda, Data\_Expediere, Data\_Estimata\_Livrare, ID\_Zona\_Transport, Status) VALUES

(2, TO\_DATE('12-01-2023', 'DD-MM-YYYY'), TO\_DATE('16-01-2023', 'DD-MM-YYYY'), 2, 'livrat');

INSERT INTO Expediere (ID\_Comanda, Data\_Expediere, Data\_Estimata\_Livrare, ID\_Zona\_Transport, Status) VALUES

(3, TO\_DATE('13-01-2023', 'DD-MM-YYYY'), TO\_DATE('17-01-2023', 'DD-MM-YYYY'), 3, 'în curs de livrare');

INSERT INTO Expediere (ID\_Comanda, Data\_Expediere, Data\_Estimata\_Livrare, ID\_Zona\_Transport, Status) VALUES

(4, TO\_DATE('14-01-2023', 'DD-MM-YYYY'), TO\_DATE('18-01-2023', 'DD-MM-YYYY'), 4, 'livrat');

INSERT INTO Expediere (ID\_Comanda, Data\_Expediere, Data\_Estimata\_Livrare, ID\_Zona\_Transport, Status) VALUES

(5, TO\_DATE('15-01-2023', 'DD-MM-YYYY'), TO\_DATE('19-01-2023', 'DD-MM-YYYY'), 5, 'în curs de livrare');

INSERT INTO Expediere (ID\_Comanda, Data\_Expediere, Data\_Estimata\_Livrare, ID\_Zona\_Transport, Status) VALUES

(6, TO\_DATE('16-01-2023', 'DD-MM-YYYY'), TO\_DATE('20-01-2023', 'DD-MM-YYYY'), 1, 'livrat');

INSERT INTO Expediere (ID\_Comanda, Data\_Expediere, Data\_Estimata\_Livrare, ID\_Zona\_Transport, Status) VALUES

(7, TO\_DATE('17-01-2023', 'DD-MM-YYYY'), TO\_DATE('21-01-2023', 'DD-MM-YYYY'), 2, 'în curs de livrare');

INSERT INTO Expediere (ID\_Comanda, Data\_Expediere, Data\_Estimata\_Livrare, ID\_Zona\_Transport, Status) VALUES

(8, TO\_DATE('18-01-2023', 'DD-MM-YYYY'), TO\_DATE('22-01-2023', 'DD-MM-YYYY'), 3, 'livrat');

INSERT INTO Expediere (ID\_Comanda, Data\_Expediere, Data\_Estimata\_Livrare, ID\_Zona\_Transport, Status) VALUES

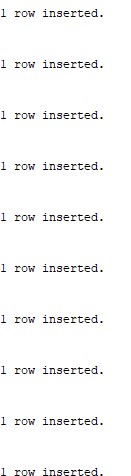
(9, TO\_DATE('19-01-2023', 'DD-MM-YYYY'), TO\_DATE('23-01-2023', 'DD-MM-YYYY'), 4, 'în curs de livrare');

INSERT INTO Expediere (ID\_Comanda, Data\_Expediere, Data\_Estimata\_Livrare, ID\_Zona\_Transport, Status) VALUES

(10, TO\_DATE('20-01-2023', 'DD-MM-YYYY'), TO\_DATE('24-01-2023', 'DD-MM-YYYY'), 5, 'livrat');

*PrintScreen cu codul si cu rezultatul executiei lui:*





Pentru tabelul **Departament\_Transport**:

*Codul SQL:*

CREATE TABLE Departament\_Transport (

ID\_Zona\_Transport NUMBER DEFAULT Departament\_Transport\_SEQ.NEXTVAL PRIMARY KEY,

Masina VARCHAR2(50)

);

INSERT INTO Departament\_Transport (Masina) VALUES

('Camion 1');

INSERT INTO Departament\_Transport (Masina) VALUES

('Camion 2');

INSERT INTO Departament\_Transport (Masina) VALUES

('Camion 3');

INSERT INTO Departament\_Transport (Masina) VALUES

('Camion 4');

INSERT INTO Departament\_Transport (Masina) VALUES

('Camion 5');

INSERT INTO Departament\_Transport (Masina) VALUES

('Furgonetă 1');

INSERT INTO Departament\_Transport (Masina) VALUES

('Furgonetă 2');

INSERT INTO Departament\_Transport (Masina) VALUES

('Furgonetă 3');

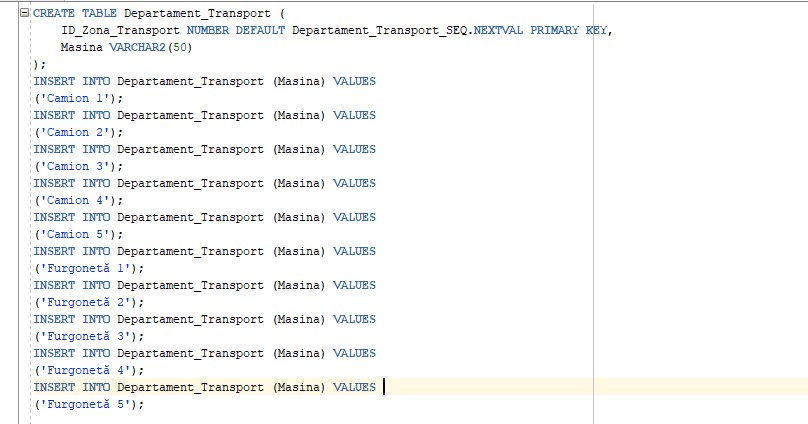
INSERT INTO Departament\_Transport (Masina) VALUES

('Furgonetă 4');

INSERT INTO Departament\_Transport (Masina) VALUES

('Furgonetă 5');

*PrintScreen cu codul si cu rezultatul executiei lui:*





Pentru tabelul Angajat:

*Codul SQL:*

CREATE TABLE Angajat (

ID\_Angajat NUMBER DEFAULT Angajat\_SEQ.NEXTVAL PRIMARY KEY,

Numele\_Familie VARCHAR2(100),

Prenumele VARCHAR2(100),

Bonus\_Lunar NUMBER(10, 2),

Salariu\_Lunar NUMBER(10, 2),

ID\_Zona\_Transport NUMBER,

CONSTRAINT fk\_zona\_transport\_angajat FOREIGN KEY (ID\_Zona\_Transport) REFERENCES Departament\_Transport(ID\_Zona\_Transport)

);

INSERT INTO Angajat (Numele\_Familie, Prenumele, Bonus\_Lunar, Salariu\_Lunar, ID\_Zona\_Transport) VALUES

('Popescu', 'Alex', 100.00, 3000.00, 1);

INSERT INTO Angajat (Numele\_Familie, Prenumele, Bonus\_Lunar, Salariu\_Lunar, ID\_Zona\_Transport) VALUES

('Ionescu', 'Cristina', 150.00, 3200.00, 2);

INSERT INTO Angajat (Numele\_Familie, Prenumele, Bonus\_Lunar, Salariu\_Lunar, ID\_Zona\_Transport) VALUES

('Georgescu', 'Mihai', 200.00, 3400.00, 3);

INSERT INTO Angajat (Numele\_Familie, Prenumele, Bonus\_Lunar, Salariu\_Lunar, ID\_Zona\_Transport) VALUES

('Vasilescu', 'Elena', 250.00, 3600.00, 4);

INSERT INTO Angajat (Numele\_Familie, Prenumele, Bonus\_Lunar, Salariu\_Lunar, ID\_Zona\_Transport) VALUES

('Dumitrescu', 'Adrian', 300.00, 3800.00, 5);

INSERT INTO Angajat (Numele\_Familie, Prenumele, Bonus\_Lunar, Salariu\_Lunar, ID\_Zona\_Transport) VALUES

('Marinescu', 'Raluca', 180.00, 3100.00, 6);

INSERT INTO Angajat (Numele\_Familie, Prenumele, Bonus\_Lunar, Salariu\_Lunar, ID\_Zona\_Transport) VALUES

('Stanciu', 'Florin', 170.00, 3300.00, 7);

INSERT INTO Angajat (Numele\_Familie, Prenumele, Bonus\_Lunar, Salariu\_Lunar, ID\_Zona\_Transport) VALUES

('Radu', 'Ioana', 190.00, 3500.00, 8);

INSERT INTO Angajat (Numele\_Familie, Prenumele, Bonus\_Lunar, Salariu\_Lunar, ID\_Zona\_Transport) VALUES

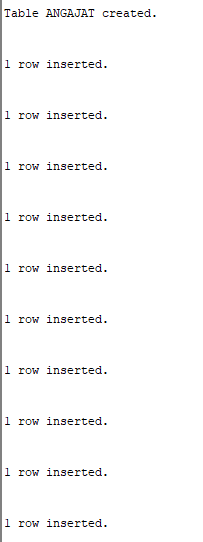
('Preda', 'Mihai', 220.00, 3700.00, 9);

INSERT INTO Angajat (Numele\_Familie, Prenumele, Bonus\_Lunar, Salariu\_Lunar, ID\_Zona\_Transport) VALUES

('Iliescu', 'Carmen', 210.00, 3900.00, 10);

*PrintScreen cu codul si cu rezultatul executiei lui:*





Pentru tabelul **Departament\_Transport**:

*Codul SQL:*

CREATE TABLE Departament\_Transport (

ID\_Zona\_Transport NUMBER DEFAULT Departament\_Transport\_SEQ.NEXTVAL PRIMARY KEY,

Masina VARCHAR2(50)

);

INSERT INTO Departament\_Transport (Masina) VALUES

('Camion 1');

INSERT INTO Departament\_Transport (Masina) VALUES

('Camion 2');

INSERT INTO Departament\_Transport (Masina) VALUES

('Camion 3');

INSERT INTO Departament\_Transport (Masina) VALUES

('Camion 4');

INSERT INTO Departament\_Transport (Masina) VALUES

('Camion 5');

INSERT INTO Departament\_Transport (Masina) VALUES

('Furgonetă 1');

INSERT INTO Departament\_Transport (Masina) VALUES

('Furgonetă 2');

INSERT INTO Departament\_Transport (Masina) VALUES

('Furgonetă 3');

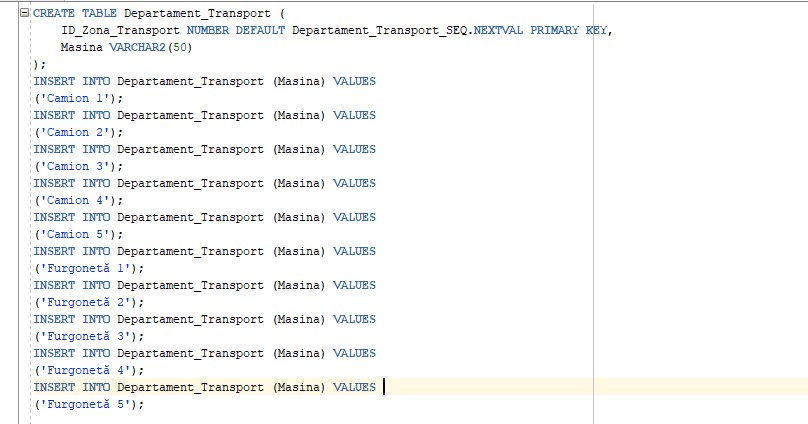
INSERT INTO Departament\_Transport (Masina) VALUES

('Furgonetă 4');

INSERT INTO Departament\_Transport (Masina) VALUES

('Furgonetă 5');

*PrintScreen cu codul si cu rezultatul executiei lui:*





Pentru tabelul **Furnizeaza** (asociativ)**:**

*Codul SQL:*

CREATE TABLE Furnizeaza (

ID\_Furnizor NUMBER,

ID\_Componenta NUMBER,

Cantitate NUMBER,

CONSTRAINT fk\_furnizeaza\_furnizor FOREIGN KEY (ID\_Furnizor) REFERENCES Furnizor(ID\_Furnizor),

CONSTRAINT fk\_furnizeaza\_componenta FOREIGN KEY (ID\_Componenta) REFERENCES Componenta(ID\_Componenta),

PRIMARY KEY (ID\_Furnizor, ID\_Componenta)

);

INSERT INTO Furnizeaza (ID\_Furnizor, ID\_Componenta, Cantitate) VALUES

(1, 1, 100);

INSERT INTO Furnizeaza (ID\_Furnizor, ID\_Componenta, Cantitate) VALUES

(2, 2, 150);

INSERT INTO Furnizeaza (ID\_Furnizor, ID\_Componenta, Cantitate) VALUES

(3, 3, 200);

INSERT INTO Furnizeaza (ID\_Furnizor, ID\_Componenta, Cantitate) VALUES

(4, 4, 250);

INSERT INTO Furnizeaza (ID\_Furnizor, ID\_Componenta, Cantitate) VALUES

(5, 5, 300);

INSERT INTO Furnizeaza (ID\_Furnizor, ID\_Componenta, Cantitate) VALUES

(6, 6, 100);

INSERT INTO Furnizeaza (ID\_Furnizor, ID\_Componenta, Cantitate) VALUES

(7, 7, 150);

INSERT INTO Furnizeaza (ID\_Furnizor, ID\_Componenta, Cantitate) VALUES

(8, 8, 200);

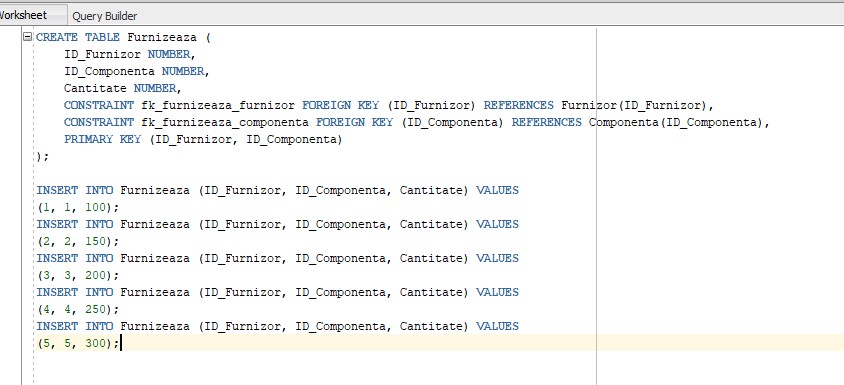
INSERT INTO Furnizeaza (ID\_Furnizor, ID\_Componenta, Cantitate) VALUES

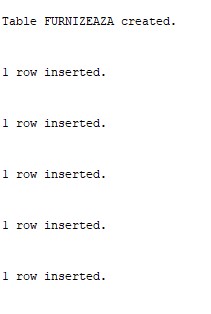
(9, 9, 250);

INSERT INTO Furnizeaza (ID\_Furnizor, ID\_Componenta, Cantitate) VALUES

(10, 10, 300);

*PrintScreen cu codul si cu rezultatul executiei lui:*





Pentru tabelul **Plaseaza** (asociativ)**:**

*Codul SQL:*

CREATE TABLE Plaseaza (

ID\_Client NUMBER,

ID\_Comanda NUMBER,

Data\_Comanda DATE,

CONSTRAINT fk\_plaseaza\_client FOREIGN KEY (ID\_Client) REFERENCES Client(ID\_Client),

CONSTRAINT fk\_plaseaza\_comanda FOREIGN KEY (ID\_Comanda) REFERENCES Comanda(ID\_Comanda),

PRIMARY KEY (ID\_Client, ID\_Comanda)

);

INSERT INTO Plaseaza (ID\_Client, ID\_Comanda, Data\_Comanda) VALUES

(1, 1, TO\_DATE('01-01-2023', 'DD-MM-YYYY'));

INSERT INTO Plaseaza (ID\_Client, ID\_Comanda, Data\_Comanda) VALUES

(2, 2, TO\_DATE('02-01-2023', 'DD-MM-YYYY'));

INSERT INTO Plaseaza (ID\_Client, ID\_Comanda, Data\_Comanda) VALUES

(3, 3, TO\_DATE('03-01-2023', 'DD-MM-YYYY'));

INSERT INTO Plaseaza (ID\_Client, ID\_Comanda, Data\_Comanda) VALUES

(4, 4, TO\_DATE('04-01-2023', 'DD-MM-YYYY'));

INSERT INTO Plaseaza (ID\_Client, ID\_Comanda, Data\_Comanda) VALUES

(5, 5, TO\_DATE('05-01-2023', 'DD-MM-YYYY'));

INSERT INTO Plaseaza (ID\_Client, ID\_Comanda, Data\_Comanda) VALUES

(6, 6, TO\_DATE('06-01-2023', 'DD-MM-YYYY'));

INSERT INTO Plaseaza (ID\_Client, ID\_Comanda, Data\_Comanda) VALUES

(7, 7, TO\_DATE('07-01-2023', 'DD-MM-YYYY'));

INSERT INTO Plaseaza (ID\_Client, ID\_Comanda, Data\_Comanda) VALUES

(8, 8, TO\_DATE('08-01-2023', 'DD-MM-YYYY'));

INSERT INTO Plaseaza (ID\_Client, ID\_Comanda, Data\_Comanda) VALUES

(9, 9, TO\_DATE('09-01-2023', 'DD-MM-YYYY'));

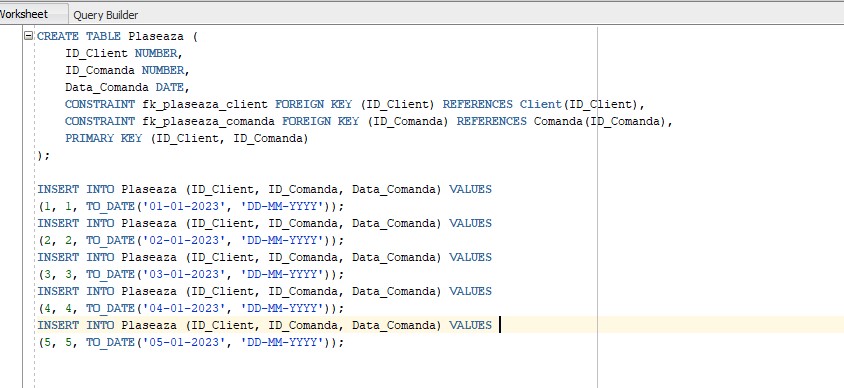
INSERT INTO Plaseaza (ID\_Client, ID\_Comanda, Data\_Comanda) VALUES

(10, 10, TO\_DATE('10-01-2023', 'DD-MM-YYYY'));

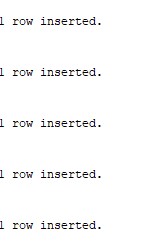
INSERT INTO Plaseaza (ID\_Client, ID\_Comanda, Data\_Comanda) VALUES

(1, 11, TO\_DATE('01-02-2023', 'DD-MM-YYYY'));

*PrintScreen cu codul si cu rezultatul executiei lui:*







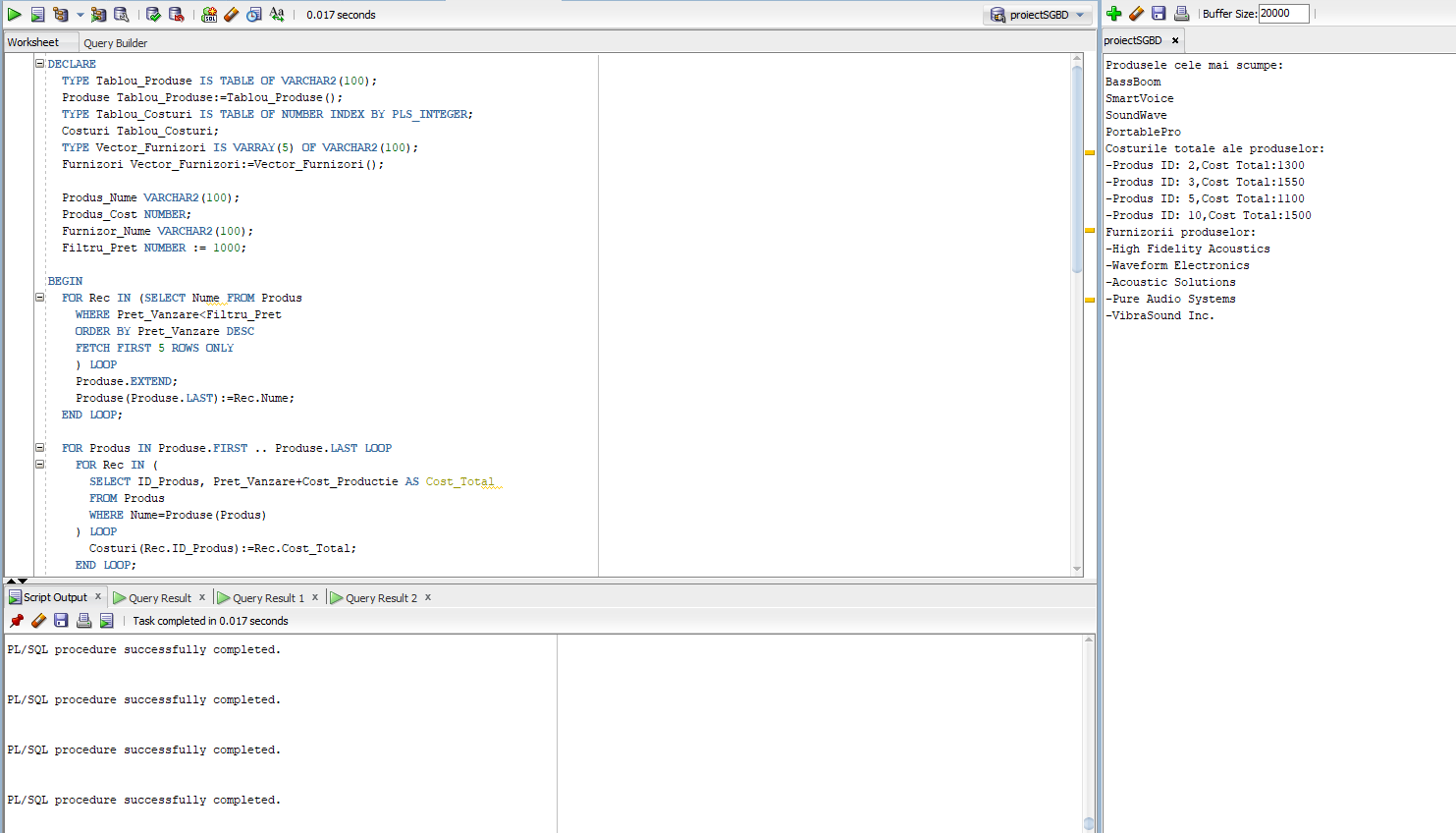
6. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze toate cele 3 tipuri de colecții studiate. Apelați subprogramul.

**Problema formulată în limbaj natural:**

Scrieți un subprogram stocat independent care să identifice cele mai scumpe 5 produse cu un preț de vânzare sub o valoare specificată, să calculeze costurile lor totale (preț de vânzare + cost de producție) și să determine furnizorii acestor produse. La final, afișați produsele, costurile totale și furnizorii identificați.

Codul PL/SQL:

|  |
| --- |
| DECLARE  TYPE Tablou\_Produse IS TABLE OF VARCHAR2(100);  Produse Tablou\_Produse:=Tablou\_Produse();  TYPE Tablou\_Costuri IS TABLE OF NUMBER INDEX BY PLS\_INTEGER;  Costuri Tablou\_Costuri;  TYPE Vector\_Furnizori IS VARRAY(5) OF VARCHAR2(100);  Furnizori Vector\_Furnizori:=Vector\_Furnizori();  Produs\_Nume VARCHAR2(100);  Produs\_Cost NUMBER;  Furnizor\_Nume VARCHAR2(100);  Filtru\_Pret NUMBER := 1000;  BEGIN  FOR Rec IN (SELECT Nume FROM Produs  WHERE Pret\_Vanzare<Filtru\_Pret  ORDER BY Pret\_Vanzare DESC  FETCH FIRST 5 ROWS ONLY  ) LOOP  Produse.EXTEND;  Produse(Produse.LAST):=Rec.Nume;  END LOOP;  FOR Produs IN Produse.FIRST .. Produse.LAST LOOP  FOR Rec IN (  SELECT ID\_Produs, Pret\_Vanzare+Cost\_Productie AS Cost\_Total  FROM Produs  WHERE Nume=Produse(Produs)  ) LOOP  Costuri(Rec.ID\_Produs):=Rec.Cost\_Total;  END LOOP;  END LOOP;  FOR Produs IN Produse.FIRST .. Produse.LAST LOOP  FOR Rec IN (  SELECT DISTINCT F.Nume AS Furnizor\_Nume  FROM Furnizor F  JOIN Furnizeaza Fu ON F.ID\_Furnizor = Fu.ID\_Furnizor  JOIN Componenta C ON Fu.ID\_Componenta = C.ID\_Componenta  JOIN Produs P ON C.ID\_Produs=P.ID\_Produs  WHERE P.Nume=Produse(Produs)  ) LOOP  IF Furnizori.COUNT<Furnizori.LIMIT THEN  Furnizori.EXTEND;  Furnizori(Furnizori.COUNT):=Rec.Furnizor\_Nume;  ELSE  EXIT;  END IF;  END LOOP;  END LOOP;  DBMS\_OUTPUT.PUT\_LINE('Produsele cele mai scumpe:');  FOR I IN Produse.FIRST .. Produse.LAST LOOP  DBMS\_OUTPUT.PUT\_LINE(Produse(I));  END LOOP;  DBMS\_OUTPUT.PUT\_LINE('Costurile totale ale produselor:');  FOR I IN Costuri.FIRST .. Costuri.LAST LOOP  IF Costuri.EXISTS(I) THEN  DBMS\_OUTPUT.PUT\_LINE('-Produs ID: '||I||',Cost Total:'||Costuri(I));  END IF;  END LOOP;  IF Furnizori.COUNT > 0 THEN  DBMS\_OUTPUT.PUT\_LINE('Furnizorii produselor:');  FOR I IN 1 .. Furnizori.COUNT LOOP  DBMS\_OUTPUT.PUT\_LINE('-'||Furnizori(I));  END LOOP;  ELSE  DBMS\_OUTPUT.PUT\_LINE('Nu exista furnizori.');  END IF;  END;  / |

PrintScreen cu rularea codului:  


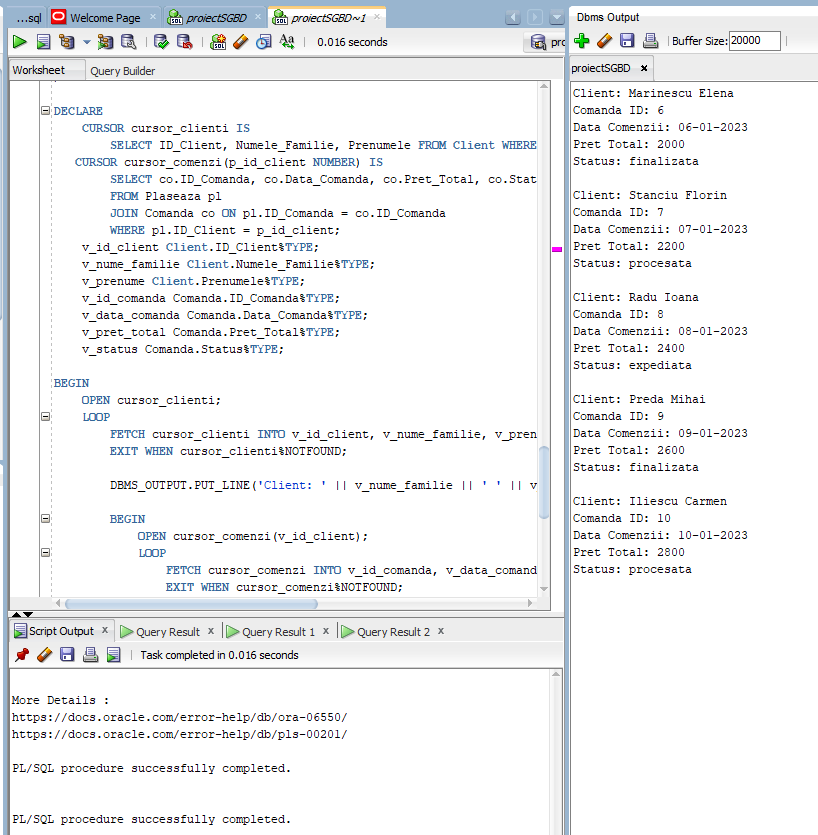
7. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat, dependent de celălalt cursor. Apelați subprogramul.

**Problema formulată în limbaj natural:**

Pentru clientii cu index-ul 11 până la 15 din baza de date, afișați numele complet (numele de familie și prenumele) și detaliile fiecărei comenzi plasate de acesta. Pentru fiecare comandă, afișați data plasării comenzii, prețul total și statusul comenzii.

Codul PL/SQL pentru rezolvarea problemei:

|  |
| --- |
| DECLARE      CURSOR cursor\_clienti IS          SELECT ID\_Client, Numele\_Familie, Prenumele FROM Client WHERE ID\_Client BETWEEN 11 AND 15;     CURSOR cursor\_comenzi(p\_id\_client NUMBER) IS  SELECT co.ID\_Comanda, co.Data\_Comanda, co.Pret\_Total, co.Status  FROM Plaseaza pl  JOIN Comanda co ON pl.ID\_Comanda = co.ID\_Comanda  WHERE pl.ID\_Client = p\_id\_client;      v\_id\_client Client.ID\_Client%TYPE;      v\_nume\_familie Client.Numele\_Familie%TYPE;      v\_prenume Client.Prenumele%TYPE;      v\_id\_comanda Comanda.ID\_Comanda%TYPE;      v\_data\_comanda Comanda.Data\_Comanda%TYPE;      v\_pret\_total Comanda.Pret\_Total%TYPE;      v\_status Comanda.Status%TYPE;    BEGIN      OPEN cursor\_clienti;      LOOP          FETCH cursor\_clienti INTO v\_id\_client, v\_nume\_familie, v\_prenume;          EXIT WHEN cursor\_clienti%NOTFOUND;            DBMS\_OUTPUT.PUT\_LINE('Client: ' || v\_nume\_familie || ' ' || v\_prenume);            BEGIN              OPEN cursor\_comenzi(v\_id\_client);              LOOP                  FETCH cursor\_comenzi INTO v\_id\_comanda, v\_data\_comanda, v\_pret\_total, v\_status;                  EXIT WHEN cursor\_comenzi%NOTFOUND;                    DBMS\_OUTPUT.PUT\_LINE('Comanda ID: ' || v\_id\_comanda);                  DBMS\_OUTPUT.PUT\_LINE('Data Comenzii: ' || TO\_CHAR(v\_data\_comanda, 'DD-MM-YYYY'));                  DBMS\_OUTPUT.PUT\_LINE('Pret Total: ' || v\_pret\_total);                  DBMS\_OUTPUT.PUT\_LINE('Status: ' || v\_status);              END LOOP;              CLOSE cursor\_comenzi;          EXCEPTION              WHEN OTHERS THEN                  DBMS\_OUTPUT.PUT\_LINE('Eroare la procesarea comenzilor pentru client ' || v\_nume\_familie || ': ' || SQLERRM);                  IF cursor\_comenzi%ISOPEN THEN                      CLOSE cursor\_comenzi;                  END IF;          END;            DBMS\_OUTPUT.PUT\_LINE(' ');      END LOOP;      CLOSE cursor\_clienti;  EXCEPTION      WHEN OTHERS THEN          DBMS\_OUTPUT.PUT\_LINE('eroare: ' || SQLERRM);          IF cursor\_clienti%ISOPEN THEN              CLOSE cursor\_clienti;          END IF;  END; |



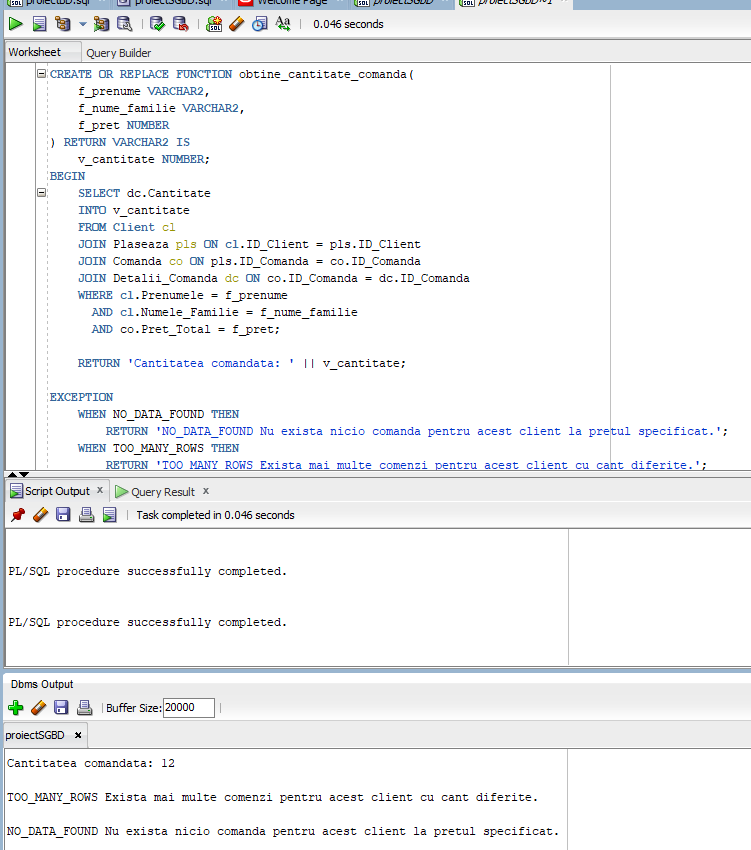
8. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele create. Tratați toate excepțiile care pot apărea, incluzând excepțiile predefinite NO\_DATA\_FOUND și TOO\_MANY\_ROWS.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

Cerința în limbaj natural: Pentru un anumit client, identificat prin prenume și nume de familie, precum și pentru o anumită valoare a prețului total al comenzii, să se returneze cantitatea totală comandată (din tabelele *Comanda* și *Detalii\_Comanda*). În cazul în care nu există nicio comandă asociată cu parametrii dați, să se afișeze un mesaj corespunzător (NO\_DATA\_FOUND), iar dacă sunt găsite mai multe comenzi care îndeplinesc condițiile, să se afișeze mesajul de eroare (TOO\_MANY\_ROWS).

Codul PL/SQL:

|  |
| --- |
| CREATE OR REPLACE FUNCTION obtine\_cantitate\_comanda(  f\_prenume VARCHAR2,  f\_nume\_familie VARCHAR2,  f\_pret NUMBER  ) RETURN VARCHAR2 IS  v\_cantitate NUMBER;  BEGIN  SELECT dc.Cantitate  INTO v\_cantitate  FROM Client cl  JOIN Plaseaza pls ON cl.ID\_Client = pls.ID\_Client  JOIN Comanda co ON pls.ID\_Comanda = co.ID\_Comanda  JOIN Detalii\_Comanda dc ON co.ID\_Comanda = dc.ID\_Comanda  WHERE cl.Prenumele = f\_prenume  AND cl.Numele\_Familie = f\_nume\_familie  AND co.Pret\_Total = f\_pret;  RETURN 'Cantitatea comandata: ' || v\_cantitate;  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  RETURN 'NO\_DATA\_FOUND Nu exista nicio comanda pentru acest client la pretul specificat.';  WHEN TOO\_MANY\_ROWS THEN  RETURN 'TOO\_MANY\_ROWS Exista mai multe comenzi pentru acest client cu cant diferite.';  WHEN OTHERS THEN  RETURN 'eroare neasteptata.';  END obtine\_cantitate\_comanda;  /  INSERT INTO Comanda (Data\_Comanda, Pret\_Total, Status)  VALUES (TO\_DATE('11-JAN-23', 'DD-MON-YY'), 1000, 'procesata');  INSERT INTO Comanda (Data\_Comanda, Pret\_Total, Status)  VALUES (TO\_DATE('12-JAN-23', 'DD-MON-YY'), 1000, 'procesata');  INSERT INTO Plaseaza (ID\_Client, ID\_Comanda, Data\_Comanda)  VALUES (1, 12, TO\_DATE('11-JAN-23', 'DD-MON-YY'));  INSERT INTO Plaseaza (ID\_Client, ID\_Comanda, Data\_Comanda)  VALUES (2, 13, TO\_DATE('12-JAN-23', 'DD-MON-YY'));  INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate)  VALUES (12, 1, 5, 200);  INSERT INTO Detalii\_Comanda (ID\_Comanda, ID\_Produs, Cantitate, Pret\_Per\_Unitate)  VALUES (13, 1, 10, 100);  BEGIN  DBMS\_OUTPUT.PUT\_LINE(obtine\_cantitate\_comanda('Ion', 'Popescu', 1000));  END;  /  BEGIN  DBMS\_OUTPUT.PUT\_LINE(obtine\_cantitate\_comanda('Maria', 'Ionescu', 500));  END;  /  BEGIN  DBMS\_OUTPUT.PUT\_LINE(obtine\_cantitate\_comanda('Andrei', 'Georgescu', 1400));  END;  / |

PrintScreen cu rularea codului:

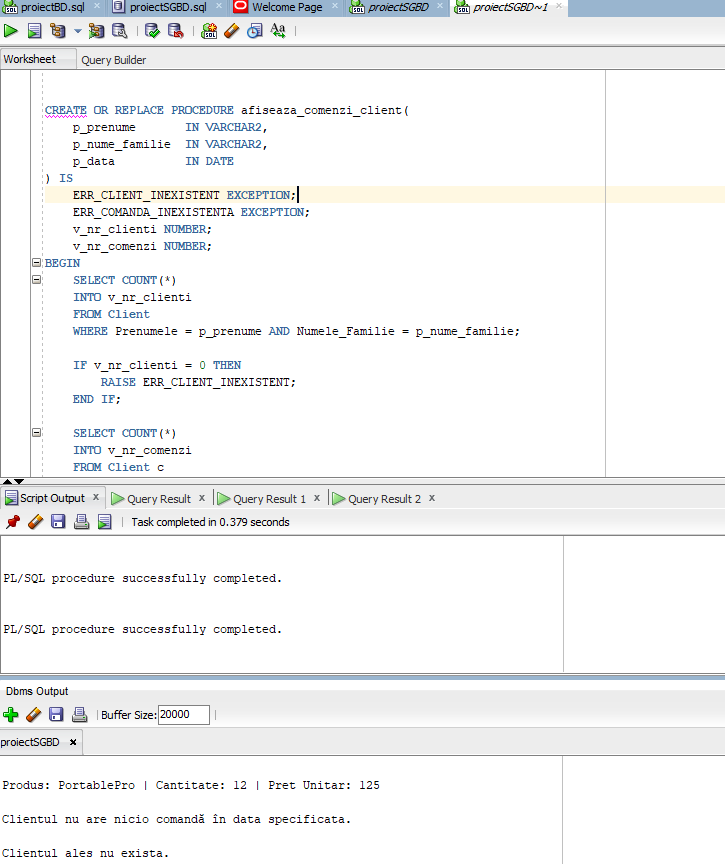


9. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip procedură care să aibă minim 2 parametri și să utilizeze într-o singură comandă SQL 5 dintre tabelele create. Definiți minim 2 excepții proprii, altele decât cele predefinite la nivel de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

Cerința în limbaj natural: Pentru un client identificat prin prenume și nume de familie, precum și pentru o dată specificată, procedura trebuie să verifice dacă acel client a plasat o comandă în data respectivă. În cazul în care clientul nu există, să afișeze un mesaj de eroare corespunzător. Dacă clientul există, însă nu are nicio comandă plasată la data respectivă, să afișeze un alt mesaj de eroare. Altfel, pentru fiecare comandă găsită la data respectivă, să afișeze produsele comandate, cantitatea cât și prețul unitar.

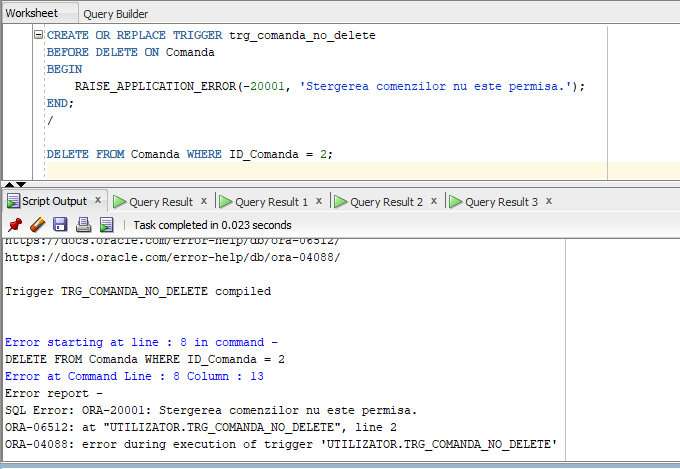
Codul PL/SQL:

|  |
| --- |
| CREATE OR REPLACE PROCEDURE afiseaza\_comenzi\_client(  p\_prenume IN VARCHAR2,  p\_nume\_familie IN VARCHAR2,  p\_data IN DATE  ) IS  ERR\_CLIENT\_INEXISTENT EXCEPTION;  ERR\_COMANDA\_INEXISTENTA EXCEPTION;  v\_nr\_clienti NUMBER;  v\_nr\_comenzi NUMBER;  BEGIN  SELECT COUNT(\*)  INTO v\_nr\_clienti  FROM Client  WHERE Prenumele = p\_prenume AND Numele\_Familie = p\_nume\_familie;  IF v\_nr\_clienti = 0 THEN  RAISE ERR\_CLIENT\_INEXISTENT;  END IF;  SELECT COUNT(\*)  INTO v\_nr\_comenzi  FROM Client c  JOIN Plaseaza pl ON c.ID\_Client = pl.ID\_Client  JOIN Comanda co ON pl.ID\_Comanda = co.ID\_Comanda  WHERE c.Prenumele = p\_prenume  AND c.Numele\_Familie = p\_nume\_familie  AND TRUNC(co.Data\_Comanda) = TRUNC(p\_data);  IF v\_nr\_comenzi = 0 THEN  RAISE ERR\_COMANDA\_INEXISTENTA;  END IF;  FOR comanda IN (  SELECT p.Nume AS Nume\_Produs,  dc.Cantitate,  dc.Pret\_Per\_Unitate  FROM Client c  JOIN Plaseaza pl ON c.ID\_Client = pl.ID\_Client  JOIN Comanda co ON pl.ID\_Comanda = co.ID\_Comanda  JOIN Detalii\_Comanda dc ON co.ID\_Comanda = dc.ID\_Comanda  JOIN Produs p ON dc.ID\_Produs = p.ID\_Produs  JOIN Componenta comp ON comp.ID\_Produs = p.ID\_Produs  WHERE c.Prenumele = p\_prenume  AND c.Numele\_Familie = p\_nume\_familie  AND TRUNC(co.Data\_Comanda) = TRUNC(p\_data)  GROUP BY p.Nume, dc.Cantitate, dc.Pret\_Per\_Unitate  ) LOOP  DBMS\_OUTPUT.PUT\_LINE('Produs: ' || comanda.Nume\_Produs ||  ' | Cantitate: ' || comanda.Cantitate ||  ' | Pret Unitar: ' || comanda.Pret\_Per\_Unitate);  END LOOP;  EXCEPTION  WHEN ERR\_CLIENT\_INEXISTENT THEN  DBMS\_OUTPUT.PUT\_LINE('Clientul ales nu exista.');  WHEN ERR\_COMANDA\_INEXISTENTA THEN  DBMS\_OUTPUT.PUT\_LINE('Clientul nu are nicio comandă în data specificata.');  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('Unexpected error: ' || SQLERRM);  END afiseaza\_comenzi\_client;  /  INSERT INTO Plaseaza (ID\_Client, ID\_Comanda, Data\_Comanda)  VALUES (3, 12, TO\_DATE('11-JAN-23', 'DD-MON-YY'));  INSERT INTO Plaseaza (ID\_Client, ID\_Comanda, Data\_Comanda)  VALUES (3, 13, TO\_DATE('12-JAN-23', 'DD-MON-YY'));  BEGIN  afiseaza\_comenzi\_client('Andrei', 'Georgescu', '03-JAN-23');  END;  /  BEGIN  afiseaza\_comenzi\_client('Andrei', 'Georgescu', '01-JAN-23');  END;  /  BEGIN  afiseaza\_comenzi\_client('Alex', 'Nite', TO\_DATE('03-JAN-23', 'DD-MON-YYYY'));  END;  / |



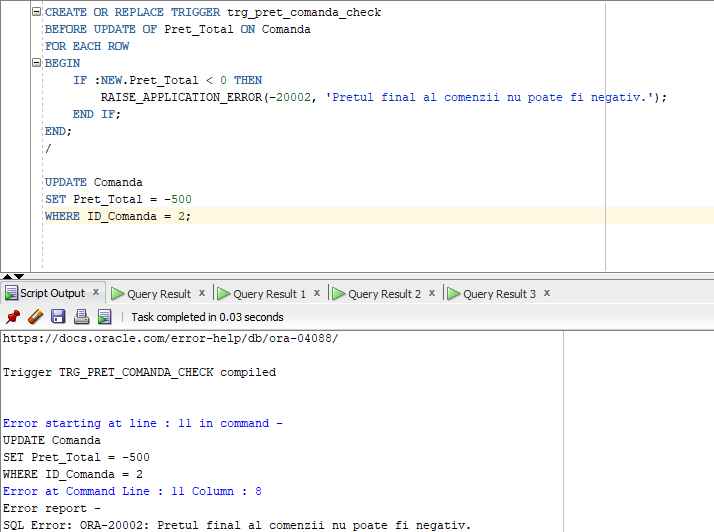
10. Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.

|  |
| --- |
| CREATE OR REPLACE TRIGGER trg\_comanda\_no\_delete  BEFORE DELETE ON Comanda  BEGIN  RAISE\_APPLICATION\_ERROR(-20001, 'Stergerea comenzilor nu este permisa.');  END;  /  DELETE FROM Comanda WHERE ID\_Comanda = 2; |



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

|  |
| --- |
| CREATE OR REPLACE TRIGGER trg\_pret\_comanda\_check  BEFORE UPDATE OF Pret\_Total ON Comanda  FOR EACH ROW  BEGIN  IF :NEW.Pret\_Total < 0 THEN  RAISE\_APPLICATION\_ERROR(-20002, 'Pretul final al comenzii nu poate fi negativ.');  END IF;  END;  /  UPDATE Comanda  SET Pret\_Total = -500  WHERE ID\_Comanda = 2; |



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

|  |
| --- |
| CREATE OR REPLACE TRIGGER trg\_restrict\_ldd  BEFORE DROP OR ALTER ON SCHEMA  DECLARE  v\_tabele\_restrictionate SYS.ODCIVARCHAR2LIST := SYS.ODCIVARCHAR2LIST('CLIENT', 'COMANDA', 'PRODUS');  BEGIN  FOR i IN 1..v\_tabele\_restrictionate.COUNT LOOP  IF ORA\_DICT\_OBJ\_NAME = v\_tabele\_restrictionate(i) THEN  RAISE\_APPLICATION\_ERROR(-20003, 'Alter si drop nu sunt permise pe tabelul ' || ORA\_DICT\_OBJ\_NAME || '.');  END IF;  END LOOP;  END;  /  DROP TABLE Client; |

