FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

UNIVERSITATEA DIN BUCUREȘTI

OPRAN ANDREI  
GRUPA 242  
2024 - 2025

SISTEM DE GESTIUNE  
A UNEI BAZE DE DATE

DEALER AUTO

**CUPRINS**

[**1. Introducere** 2](#_Toc187442240)

[Tema aleasă 2](#_Toc187442241)

[Infrastructura utilizată pentru implementare 2](#_Toc187442242)

[**2. Cerința 2 - Diagrama entitate-relație** 3](#_Toc187442243)

[**3. Cerința 3 - Diagrama conceptuală** 4](#_Toc187442244)

[**4. Cerința 4 - Implementați în Oracle diagrama conceptuală realizată** 5](#_Toc187442245)

[**5. Cerința 5 - Adăugați informații coerente în tabelele create** 10](#_Toc187442246)

[**6. Cerința 6 - Formulați în limbaj natural o problemă (3 tipuri de colecții)** 19](#_Toc187442247)

[**7. Cerința 7 - Formulați în limbaj natural o problemă (2 tipuri diferite de cursoare)** 24](#_Toc187442248)

[**8. Cerința 8 - Formulați în limbaj natural o problemă (să utilizeze într-o singură comandă SQL 3 dintre tabele)** 28](#_Toc187442249)

[**9. Cerința 9 - Formulați în limbaj natural o problemă (să utilizeze într-o singură comandă SQL 5 dintre tabele)** 32](#_Toc187442250)

[**10. Cerința 10 - Definiți un trigger de tip LMD la nivel de comandă** 40](#_Toc187442251)

[**11. Cerința 11 - Definiți un trigger de tip LMD la nivel de linie** 43](#_Toc187442252)

[**12. Cerința 12 - Definiți un trigger de tip LDD** 47](#_Toc187442253)

# **1. Introducere**

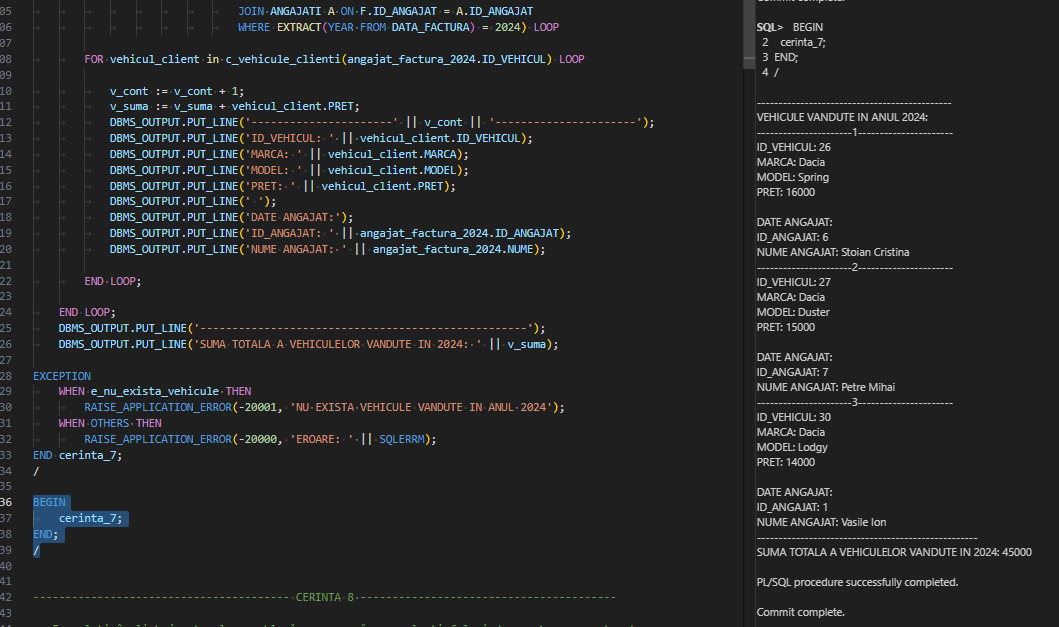
## Tema aleasă

Proiectul consta într-o baza de date destinata unui Dealer auto, respectiv un sistem de gestiune a acesteia, prin intermediul căreia se pot administra diverse informații utile pentru o buna desfășurare a activității de comerț.

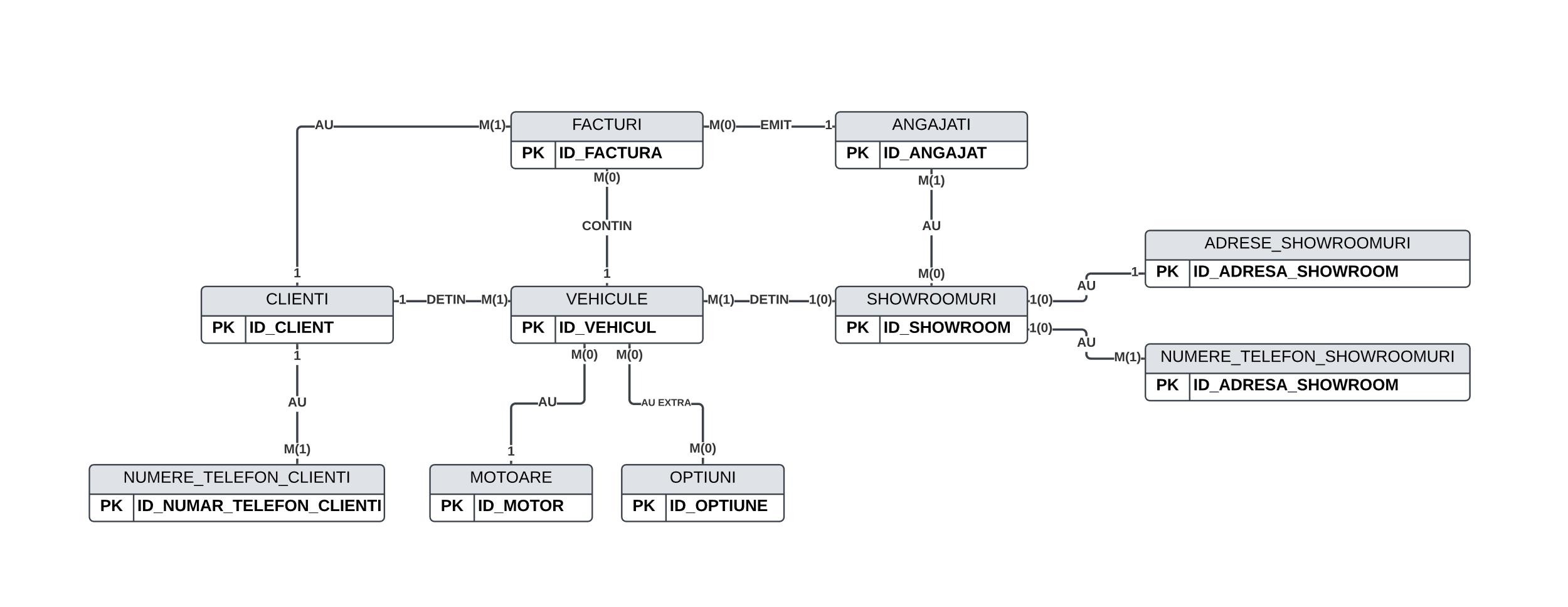
Baza de date include tabele pentru clienți, showroom-uri, vehicule, motoare, opțiuni extra pentru vehicule, facturi, angajați și numere de telefon.

## Infrastructura utilizată pentru implementare

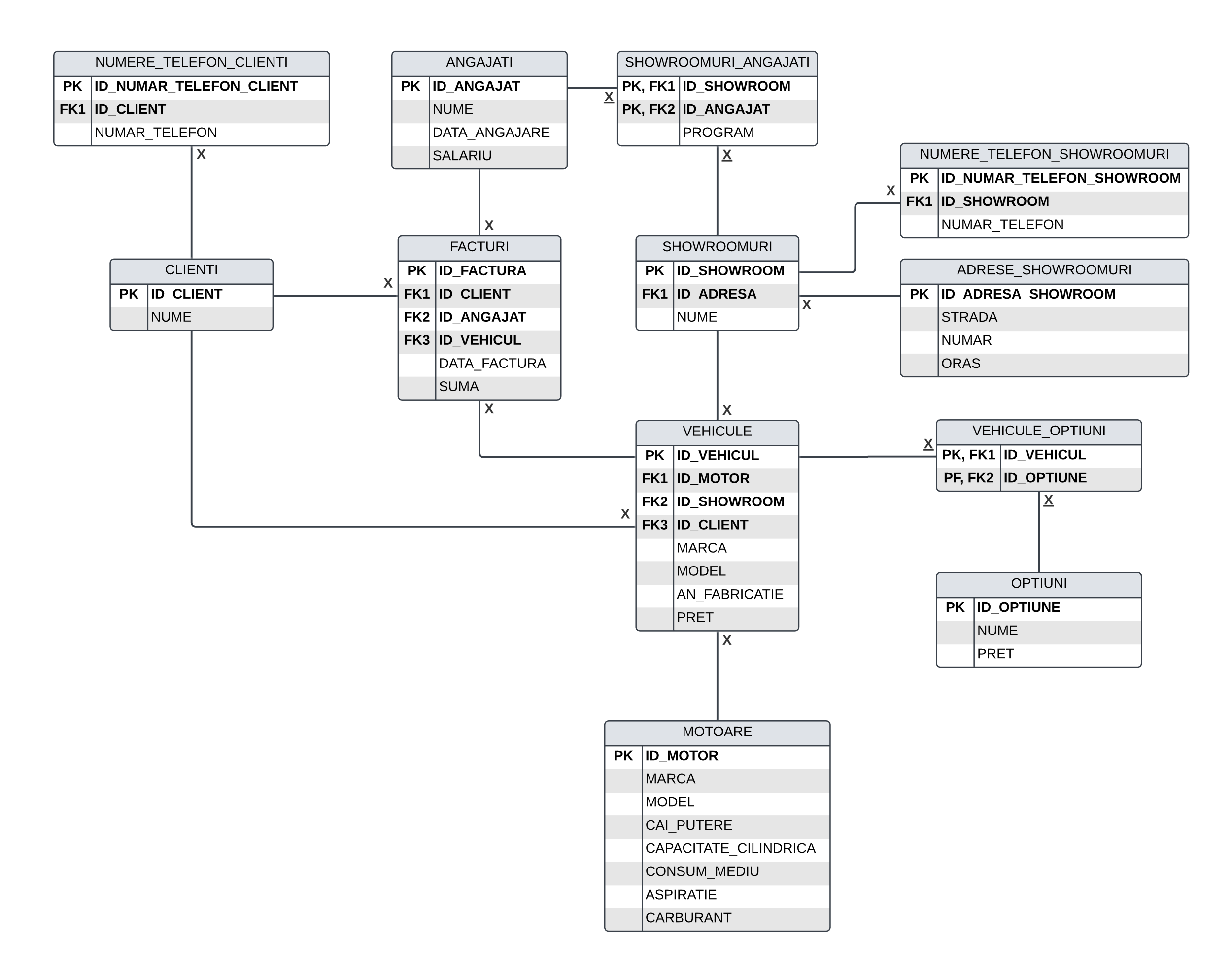
* Versiune SGBD: Oracle Database 21c Express Edition for Windows x64
* RAM alocat: 4 GB
* Sistem operare: Windows 11, version 24H2 instalat local (fără mașină virtuala)
* IDE: Visual Studio Code cu extensia ”Oracle Developer Tools for VS Code (SQL and PLSQL)”
* Observație: Pentru ca am folosit Visual Studio Code, in print-screen-urile cu rularea exercițiilor „Script Output” si „Dbms Output” se afla in aceeași fereastră, fata de SQL Developer unde exista o fereastra separata pentru amândouă. Exemplu:



# **2. Cerința 2 - Diagrama entitate-relație**

****

# **3. Cerința 3 - Diagrama conceptuală**



# **4. Cerința 4 - Implementați în Oracle diagrama conceptuală realizată**

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

**Cod SQL:**

CREATE TABLE CLIENTI (

    ID\_CLIENT NUMBER(6) PRIMARY KEY,

    NUME VARCHAR2(256)

);

CREATE TABLE ADRESE\_SHOWROOMURI (

    ID\_ADRESA\_SHOWROOM NUMBER(6) PRIMARY KEY,

    STRADA VARCHAR2(256),

    NUMAR NUMBER(6),

    ORAS VARCHAR2(256)

);

CREATE TABLE MOTOARE (

    ID\_MOTOR NUMBER(6) PRIMARY KEY,

    MARCA VARCHAR2(256),

    MODEL VARCHAR2(256),

    CAI\_PUTERE NUMBER(6),

    CAPACITATE\_CILINDRICA NUMBER(6),

    CONSUM\_MEDIU NUMBER(6),

    ASPIRATIE VARCHAR2(256),

    CARBURANT VARCHAR2(256)

);

CREATE TABLE OPTIUNI (

    ID\_OPTIUNE NUMBER(6) PRIMARY KEY,

    NUME VARCHAR2(256),

    PRET NUMBER(6)

);

CREATE TABLE SHOWROOMURI (

    ID\_SHOWROOM NUMBER(6) PRIMARY KEY,

    ID\_ADRESA NUMBER(6),

    NUME VARCHAR2(256),

    FOREIGN KEY (ID\_ADRESA) REFERENCES ADRESE\_SHOWROOMURI(ID\_ADRESA\_SHOWROOM)

);

CREATE TABLE ANGAJATI (

    ID\_ANGAJAT NUMBER(6) PRIMARY KEY,

    NUME VARCHAR2(256),

    DATA\_ANGAJARE DATE,

    SALARIU NUMBER(6)

);

CREATE TABLE VEHICULE (

    ID\_VEHICUL NUMBER(6) PRIMARY KEY,

    ID\_MOTOR NUMBER(6),

    ID\_SHOWROOM NUMBER(6),

    ID\_CLIENT NUMBER(6),

    MARCA VARCHAR2(256),

    MODEL VARCHAR2(256),

    AN\_FABRICATIE NUMBER(6),

    PRET NUMBER(6),

    FOREIGN KEY (ID\_MOTOR) REFERENCES MOTOARE(ID\_MOTOR),

    FOREIGN KEY (ID\_SHOWROOM) REFERENCES SHOWROOMURI(ID\_SHOWROOM),

    FOREIGN KEY (ID\_CLIENT) REFERENCES CLIENTI(ID\_CLIENT),

    CONSTRAINT CHECK\_ID\_CLIENT\_SHOWROOM CHECK (

        (ID\_CLIENT IS NULL AND ID\_SHOWROOM IS NOT NULL)

        OR (ID\_CLIENT IS NOT NULL AND ID\_SHOWROOM IS NULL)

        OR (ID\_CLIENT IS NULL AND ID\_SHOWROOM IS NULL)

    )

);

CREATE TABLE FACTURI (

    ID\_FACTURA NUMBER(6) PRIMARY KEY,

    ID\_CLIENT NUMBER(6) NOT NULL,

    ID\_ANGAJAT NUMBER(6) NOT NULL,

    ID\_VEHICUL NUMBER(6) NOT NULL,

    DATA\_FACTURA DATE,

    SUMA NUMBER(6),

    FOREIGN KEY (ID\_CLIENT) REFERENCES CLIENTI(ID\_CLIENT),

    FOREIGN KEY (ID\_ANGAJAT) REFERENCES ANGAJATI(ID\_ANGAJAT),

    FOREIGN KEY (ID\_VEHICUL) REFERENCES VEHICULE(ID\_VEHICUL)

);

CREATE TABLE NUMERE\_TELEFON\_SHOWROOMURI (

    ID\_NUMAR\_TELEFON\_SHOWROOM NUMBER(6) PRIMARY KEY,

    ID\_SHOWROOM NUMBER(6),

    NUMAR\_TELEFON VARCHAR2(256),

    CONSTRAINT CHECK\_NUMAR\_TELEFON\_SHOWROOM CHECK

    (

        REGEXP\_LIKE(NUMAR\_TELEFON, '^07[0-9]{2} [0-9]{3} [0-9]{3}$')

    ),

    FOREIGN KEY (ID\_SHOWROOM) REFERENCES SHOWROOMURI(ID\_SHOWROOM)

);

CREATE TABLE NUMERE\_TELEFON\_CLIENTI (

    ID\_NUMAR\_TELEFON\_CLIENT NUMBER(6) PRIMARY KEY,

    ID\_CLIENT NUMBER(6),

    NUMAR\_TELEFON VARCHAR2(256),

    CONSTRAINT CHECK\_NUMAR\_TELEFON\_CLIENT CHECK

    (

        REGEXP\_LIKE(NUMAR\_TELEFON, '^07[0-9]{2} [0-9]{3} [0-9]{3}$')

    ),

    FOREIGN KEY (ID\_CLIENT) REFERENCES CLIENTI(ID\_CLIENT)

);

CREATE TABLE SHOWROOMURI\_ANGAJATI (

    ID\_SHOWROOM NUMBER(6),

    ID\_ANGAJAT NUMBER(6),

    PROGRAM VARCHAR2(256),

    PRIMARY KEY (ID\_SHOWROOM, ID\_ANGAJAT),

    FOREIGN KEY (ID\_SHOWROOM) REFERENCES SHOWROOMURI(ID\_SHOWROOM),

    FOREIGN KEY (ID\_ANGAJAT) REFERENCES ANGAJATI(ID\_ANGAJAT)

);

CREATE TABLE VEHICULE\_OPTIUNI (

    ID\_VEHICUL NUMBER(6),

    ID\_OPTIUNE NUMBER(6),

    PRIMARY KEY (ID\_VEHICUL, ID\_OPTIUNE),

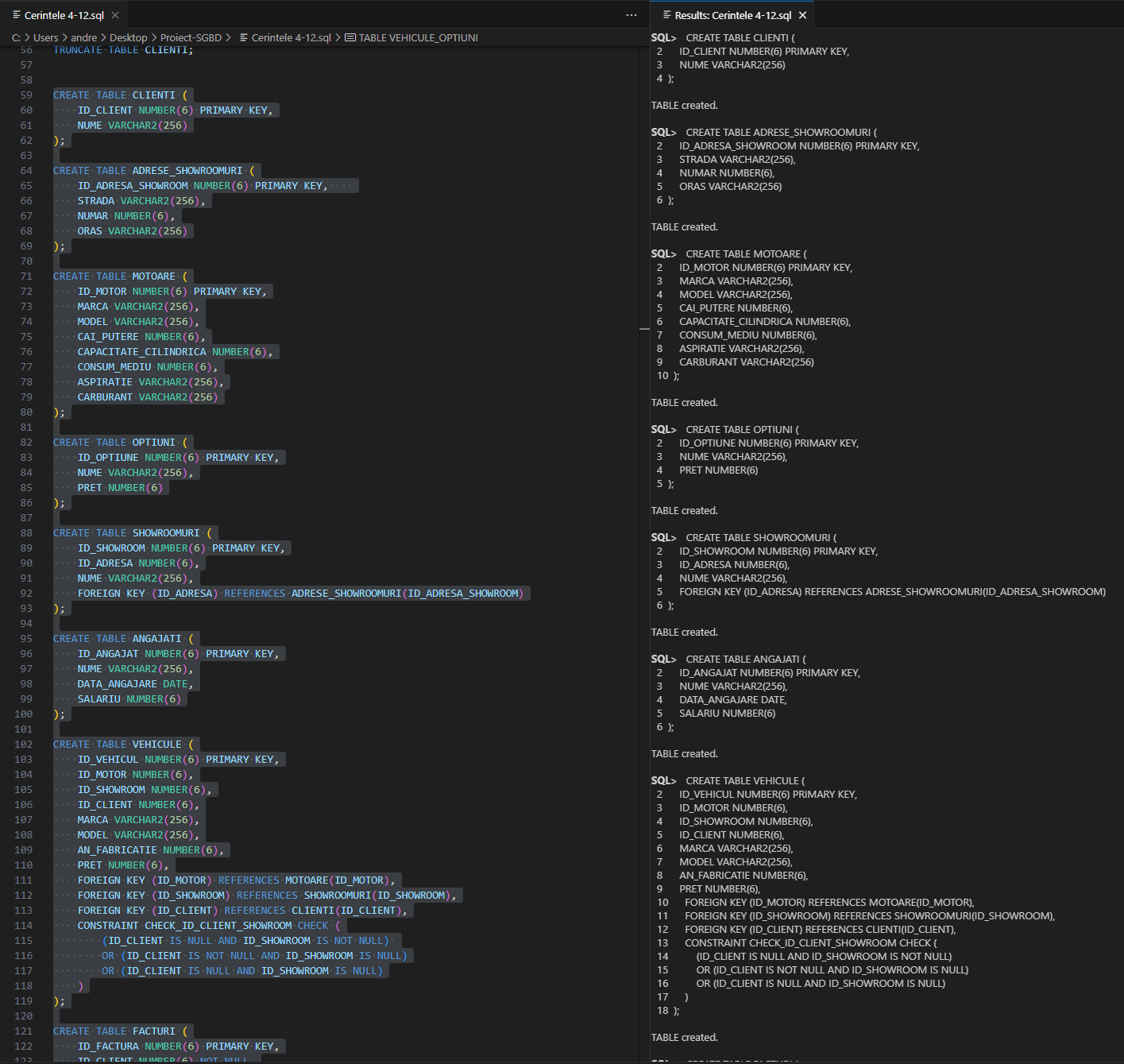
    FOREIGN KEY (ID\_VEHICUL) REFERENCES VEHICULE(ID\_VEHICUL),

    FOREIGN KEY (ID\_OPTIUNE) REFERENCES OPTIUNI(ID\_OPTIUNE)

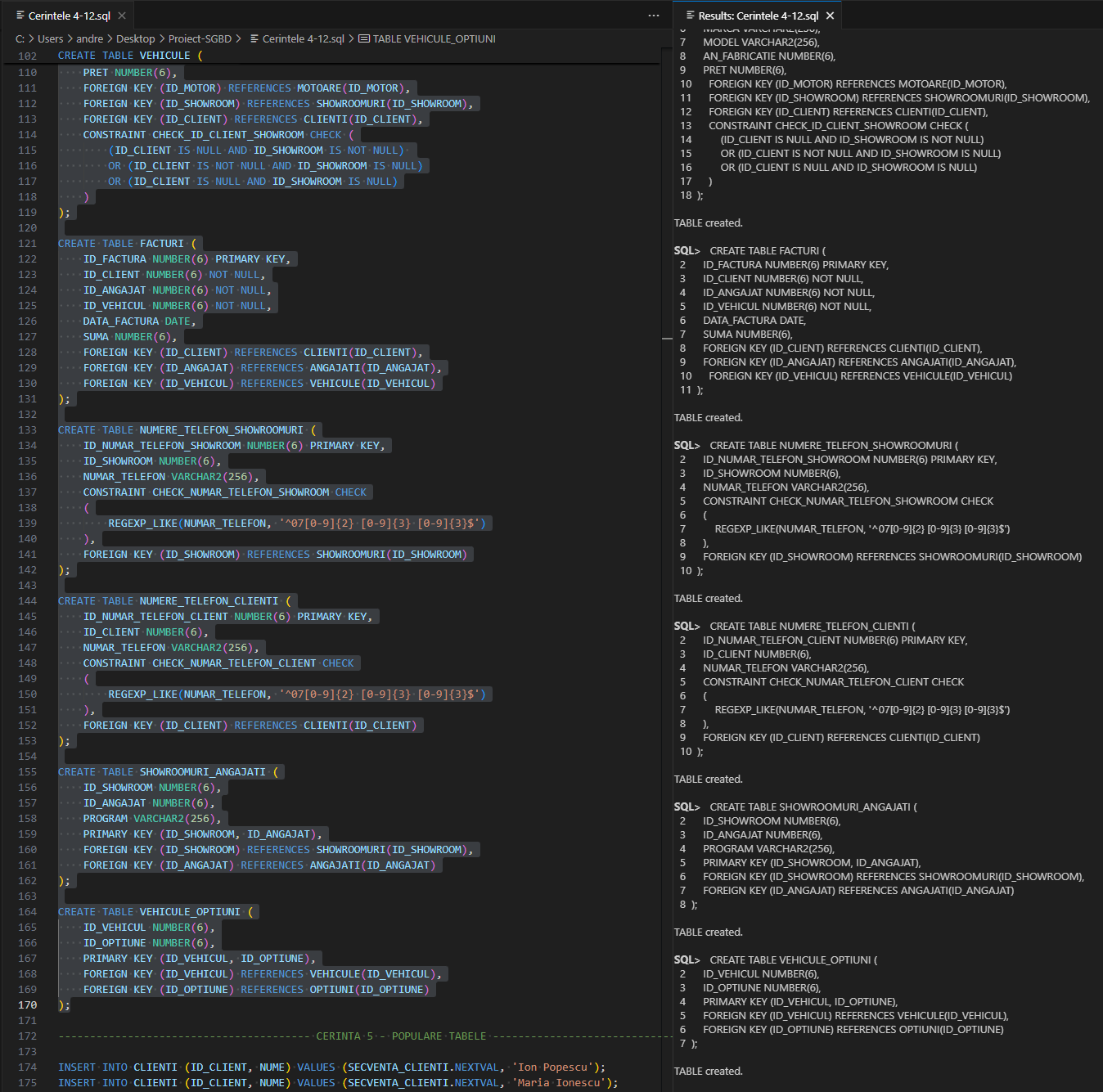
);

**Rezultat rulare:**

Poza 1/2:

****

Poza 2/2:

****

# **5. Cerința 5 - Adăugați informații coerente în tabelele create**

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

**Cod SQL:**

INSERT INTO CLIENTI (ID\_CLIENT, NUME) VALUES (SECVENTA\_CLIENTI.NEXTVAL, 'Ion Popescu');

INSERT INTO CLIENTI (ID\_CLIENT, NUME) VALUES (SECVENTA\_CLIENTI.NEXTVAL, 'Maria Ionescu');

INSERT INTO CLIENTI (ID\_CLIENT, NUME) VALUES (SECVENTA\_CLIENTI.NEXTVAL, 'Vasile Georgescu');

INSERT INTO CLIENTI (ID\_CLIENT, NUME) VALUES (SECVENTA\_CLIENTI.NEXTVAL, 'Elena Mihai');

INSERT INTO CLIENTI (ID\_CLIENT, NUME) VALUES (SECVENTA\_CLIENTI.NEXTVAL, 'Cristina Stoica');

INSERT INTO ADRESE\_SHOWROOMURI (ID\_ADRESA\_SHOWROOM, STRADA, NUMAR, ORAS) VALUES (SECVENTA\_ADRESE\_SHOWROOM.NEXTVAL, 'Soseaua Colentina', 33, 'Bucuresti');

INSERT INTO ADRESE\_SHOWROOMURI (ID\_ADRESA\_SHOWROOM, STRADA, NUMAR, ORAS) VALUES (SECVENTA\_ADRESE\_SHOWROOM.NEXTVAL, 'Strada Republicii', 45, 'Cluj-Napoca');

INSERT INTO ADRESE\_SHOWROOMURI (ID\_ADRESA\_SHOWROOM, STRADA, NUMAR, ORAS) VALUES (SECVENTA\_ADRESE\_SHOWROOM.NEXTVAL, 'Bulevardul Basarabia', 34, 'Bucuresti');

INSERT INTO ADRESE\_SHOWROOMURI (ID\_ADRESA\_SHOWROOM, STRADA, NUMAR, ORAS) VALUES (SECVENTA\_ADRESE\_SHOWROOM.NEXTVAL, 'Strada Unirii', 23, 'Timisoara');

INSERT INTO ADRESE\_SHOWROOMURI (ID\_ADRESA\_SHOWROOM, STRADA, NUMAR, ORAS) VALUES (SECVENTA\_ADRESE\_SHOWROOM.NEXTVAL, 'Strada Libertatii', 56, 'Constanta');

INSERT INTO ADRESE\_SHOWROOMURI (ID\_ADRESA\_SHOWROOM, STRADA, NUMAR, ORAS) VALUES (SECVENTA\_ADRESE\_SHOWROOM.NEXTVAL, 'Strada Independentei', 78, 'Craiova');

INSERT INTO ADRESE\_SHOWROOMURI (ID\_ADRESA\_SHOWROOM, STRADA, NUMAR, ORAS) VALUES (SECVENTA\_ADRESE\_SHOWROOM.NEXTVAL, 'Strada Victoriei', 89, 'Brasov');

INSERT INTO ADRESE\_SHOWROOMURI (ID\_ADRESA\_SHOWROOM, STRADA, NUMAR, ORAS) VALUES (SECVENTA\_ADRESE\_SHOWROOM.NEXTVAL, 'Strada Mihai Viteazu', 101, 'Galati');

INSERT INTO ADRESE\_SHOWROOMURI (ID\_ADRESA\_SHOWROOM, STRADA, NUMAR, ORAS) VALUES (SECVENTA\_ADRESE\_SHOWROOM.NEXTVAL, 'Strada Stefan cel Mare', 67, 'Ploiesti');

INSERT INTO ADRESE\_SHOWROOMURI (ID\_ADRESA\_SHOWROOM, STRADA, NUMAR, ORAS) VALUES (SECVENTA\_ADRESE\_SHOWROOM.NEXTVAL, 'Strada Tudor Vladimirescu', 34, 'Oradea');

INSERT INTO MOTOARE (ID\_MOTOR, MARCA, MODEL, CAI\_PUTERE, CAPACITATE\_CILINDRICA, CONSUM\_MEDIU, ASPIRATIE, CARBURANT)

VALUES (SECVENTA\_MOTOARE.NEXTVAL, 'Renault', 'K9K', 110, 1461, 4.8, 'Turbo', 'Diesel');

INSERT INTO MOTOARE (ID\_MOTOR, MARCA, MODEL, CAI\_PUTERE, CAPACITATE\_CILINDRICA, CONSUM\_MEDIU, ASPIRATIE, CARBURANT)

VALUES (SECVENTA\_MOTOARE.NEXTVAL, 'Renault', 'H5Ft', 140, 1333, 5.3, 'Turbo', 'Benzina');

INSERT INTO MOTOARE (ID\_MOTOR, MARCA, MODEL, CAI\_PUTERE, CAPACITATE\_CILINDRICA, CONSUM\_MEDIU, ASPIRATIE, CARBURANT)

VALUES (SECVENTA\_MOTOARE.NEXTVAL, 'Renault', 'F4R', 170, 1998, 7.5, 'Natural', 'Benzina');

INSERT INTO MOTOARE (ID\_MOTOR, MARCA, MODEL, CAI\_PUTERE, CAPACITATE\_CILINDRICA, CONSUM\_MEDIU, ASPIRATIE, CARBURANT)

VALUES (SECVENTA\_MOTOARE.NEXTVAL, 'Renault', 'M9R', 150, 1995, 6.0, 'Turbo', 'Diesel');

INSERT INTO MOTOARE (ID\_MOTOR, MARCA, MODEL, CAI\_PUTERE, CAPACITATE\_CILINDRICA, CONSUM\_MEDIU, ASPIRATIE, CARBURANT)

VALUES (SECVENTA\_MOTOARE.NEXTVAL, 'Dacia', 'K7M', 90, 1598, 7.0, 'Natural', 'Benzina');

INSERT INTO MOTOARE (ID\_MOTOR, MARCA, MODEL, CAI\_PUTERE, CAPACITATE\_CILINDRICA, CONSUM\_MEDIU, ASPIRATIE, CARBURANT)

VALUES (SECVENTA\_MOTOARE.NEXTVAL, 'Dacia', 'K4M', 105, 1598, 6.5, 'Natural', 'Benzina');

INSERT INTO MOTOARE (ID\_MOTOR, MARCA, MODEL, CAI\_PUTERE, CAPACITATE\_CILINDRICA, CONSUM\_MEDIU, ASPIRATIE, CARBURANT)

VALUES (SECVENTA\_MOTOARE.NEXTVAL, 'Dacia', 'H4Bt', 130, 1332, 5.8, 'Turbo', 'Benzina');

INSERT INTO MOTOARE (ID\_MOTOR, MARCA, MODEL, CAI\_PUTERE, CAPACITATE\_CILINDRICA, CONSUM\_MEDIU, ASPIRATIE, CARBURANT)

VALUES (SECVENTA\_MOTOARE.NEXTVAL, 'Renault', 'R9M', 130, 1598, 4.5, 'Turbo', 'Diesel');

INSERT INTO MOTOARE (ID\_MOTOR, MARCA, MODEL, CAI\_PUTERE, CAPACITATE\_CILINDRICA, CONSUM\_MEDIU, ASPIRATIE, CARBURANT)

VALUES (SECVENTA\_MOTOARE.NEXTVAL, 'Dacia', 'H5Ht', 150, 1332, 5.5, 'Turbo', 'Benzina');

INSERT INTO MOTOARE (ID\_MOTOR, MARCA, MODEL, CAI\_PUTERE, CAPACITATE\_CILINDRICA, CONSUM\_MEDIU, ASPIRATIE, CARBURANT)

VALUES (SECVENTA\_MOTOARE.NEXTVAL, 'Renault', 'K4J', 75, 1390, 6.8, 'Natural', 'Benzina');

INSERT INTO OPTIUNI (ID\_OPTIUNE, NUME, PRET) VALUES (SECVENTA\_OPTIUNI.NEXTVAL, 'Navigatie', 1200);

INSERT INTO OPTIUNI (ID\_OPTIUNE, NUME, PRET) VALUES (SECVENTA\_OPTIUNI.NEXTVAL, 'Scaune incalzite', 800);

INSERT INTO OPTIUNI (ID\_OPTIUNE, NUME, PRET) VALUES (SECVENTA\_OPTIUNI.NEXTVAL, 'Climatizare automata', 1500);

INSERT INTO OPTIUNI (ID\_OPTIUNE, NUME, PRET) VALUES (SECVENTA\_OPTIUNI.NEXTVAL, 'Sistem audio premium', 1000);

INSERT INTO OPTIUNI (ID\_OPTIUNE, NUME, PRET) VALUES (SECVENTA\_OPTIUNI.NEXTVAL, 'Faruri LED', 900);

INSERT INTO OPTIUNI (ID\_OPTIUNE, NUME, PRET) VALUES (SECVENTA\_OPTIUNI.NEXTVAL, 'Senzori parcare', 600);

INSERT INTO OPTIUNI (ID\_OPTIUNE, NUME, PRET) VALUES (SECVENTA\_OPTIUNI.NEXTVAL, 'Camera marsarier', 700);

INSERT INTO OPTIUNI (ID\_OPTIUNE, NUME, PRET) VALUES (SECVENTA\_OPTIUNI.NEXTVAL, 'Pilot automat', 1100);

INSERT INTO OPTIUNI (ID\_OPTIUNE, NUME, PRET) VALUES (SECVENTA\_OPTIUNI.NEXTVAL, 'Jante aliaj', 1300);

INSERT INTO OPTIUNI (ID\_OPTIUNE, NUME, PRET) VALUES (SECVENTA\_OPTIUNI.NEXTVAL, 'Trapa panoramica', 2000);

INSERT INTO SHOWROOMURI (ID\_SHOWROOM, ID\_ADRESA, NUME)

VALUES (SECVENTA\_SHOWROOM.NEXTVAL, 1, 'Dacia Bucuresti Colentina');

INSERT INTO SHOWROOMURI (ID\_SHOWROOM, ID\_ADRESA, NUME)

VALUES (SECVENTA\_SHOWROOM.NEXTVAL, 2, 'Dacia Cluj-Napoca');

INSERT INTO SHOWROOMURI (ID\_SHOWROOM, ID\_ADRESA, NUME)

VALUES (SECVENTA\_SHOWROOM.NEXTVAL, 3, 'Dacia Bucuresti Basarabiei');

INSERT INTO SHOWROOMURI (ID\_SHOWROOM, ID\_ADRESA, NUME)

VALUES (SECVENTA\_SHOWROOM.NEXTVAL, 4, 'Dacia Timisoara');

INSERT INTO SHOWROOMURI (ID\_SHOWROOM, ID\_ADRESA, NUME)

VALUES (SECVENTA\_SHOWROOM.NEXTVAL, 5, 'Dacia Constanta');

INSERT INTO ANGAJATI (ID\_ANGAJAT, NUME, DATA\_ANGAJARE, SALARIU)

VALUES (SECVENTA\_ANGAJATI.NEXTVAL, 'Vasile Ion', TO\_DATE('2020-01-15', 'YYYY-MM-DD'), 3500);

INSERT INTO ANGAJATI (ID\_ANGAJAT, NUME, DATA\_ANGAJARE, SALARIU)

VALUES (SECVENTA\_ANGAJATI.NEXTVAL, 'Mihai Ana', TO\_DATE('2019-06-10', 'YYYY-MM-DD'), 4200);

INSERT INTO ANGAJATI (ID\_ANGAJAT, NUME, DATA\_ANGAJARE, SALARIU)

VALUES (SECVENTA\_ANGAJATI.NEXTVAL, 'Popa George', TO\_DATE('2018-03-20', 'YYYY-MM-DD'), 3800);

INSERT INTO ANGAJATI (ID\_ANGAJAT, NUME, DATA\_ANGAJARE, SALARIU)

VALUES (SECVENTA\_ANGAJATI.NEXTVAL, 'Ionescu Maria', TO\_DATE('2021-11-05', 'YYYY-MM-DD'), 3100);

INSERT INTO ANGAJATI (ID\_ANGAJAT, NUME, DATA\_ANGAJARE, SALARIU)

VALUES (SECVENTA\_ANGAJATI.NEXTVAL, 'Dumitrescu Andrei', TO\_DATE('2017-09-12', 'YYYY-MM-DD'), 4500);

INSERT INTO ANGAJATI (ID\_ANGAJAT, NUME, DATA\_ANGAJARE, SALARIU)

VALUES (SECVENTA\_ANGAJATI.NEXTVAL, 'Stoian Cristina', TO\_DATE('2022-02-28', 'YYYY-MM-DD'), 3300);

INSERT INTO ANGAJATI (ID\_ANGAJAT, NUME, DATA\_ANGAJARE, SALARIU)

VALUES (SECVENTA\_ANGAJATI.NEXTVAL, 'Petre Mihai', TO\_DATE('2016-07-19', 'YYYY-MM-DD'), 4700);

INSERT INTO ANGAJATI (ID\_ANGAJAT, NUME, DATA\_ANGAJARE, SALARIU)

VALUES (SECVENTA\_ANGAJATI.NEXTVAL, 'Radu Elena', TO\_DATE('2020-05-23', 'YYYY-MM-DD'), 3600);

INSERT INTO ANGAJATI (ID\_ANGAJAT, NUME, DATA\_ANGAJARE, SALARIU)

VALUES (SECVENTA\_ANGAJATI.NEXTVAL, 'Badea Alexandru', TO\_DATE('2019-08-14', 'YYYY-MM-DD'), 3900);

INSERT INTO ANGAJATI (ID\_ANGAJAT, NUME, DATA\_ANGAJARE, SALARIU)

VALUES (SECVENTA\_ANGAJATI.NEXTVAL, 'Marinescu Daniela', TO\_DATE('2018-12-01', 'YYYY-MM-DD'), 3400);

-- VEHICULE SHOWROOM

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 1, 1, NULL, 'Dacia', 'Duster', 2024, 15000);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 2, 1, NULL, 'Dacia', 'Sandero', 2024, 12000);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 3, 2, NULL, 'Dacia', 'Logan', 2023, 13000);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 4, 3, NULL, 'Dacia', 'Lodgy', 2024, 14000);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 5, 4, NULL, 'Dacia', 'Dokker', 2024, 13500);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 6, 5, NULL, 'Dacia', 'Spring', 2024, 16000);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 7, 1, NULL, 'Dacia', 'Duster', 2024, 15000);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 8, 2, NULL, 'Dacia', 'Sandero', 2025, 12000);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 9, 3, NULL, 'Dacia', 'Logan', 2023, 13000);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 10, 4, NULL, 'Dacia', 'Lodgy', 2023, 14000);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 1, 1, NULL, 'Dacia', 'Duster', 2024, 15000);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 2, 1, NULL, 'Dacia', 'Sandero', 2024, 12000);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 3, 2, NULL, 'Dacia', 'Logan', 2025, 13000);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 4, 3, NULL, 'Dacia', 'Lodgy', 2024, 14000);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 5, 4, NULL, 'Dacia', 'Dokker', 2024, 13500);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 6, 5, NULL, 'Dacia', 'Spring', 2024, 16000);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 7, 1, NULL, 'Dacia', 'Duster', 2023, 15000);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 8, 2, NULL, 'Dacia', 'Sandero', 2023, 12000);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 9, 3, NULL, 'Dacia', 'Logan', 2024, 13000);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 10, 4, NULL, 'Dacia', 'Lodgy', 2025, 14000);

-- VEHICULE CLIENTI

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 1, NULL, 1, 'Dacia', 'Duster', 2021, 15000);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 2, NULL, 2, 'Dacia', 'Sandero', 2023, 12000);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 3, NULL, 3, 'Dacia', 'Logan', 2023, 13000);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 4, NULL, 4, 'Dacia', 'Lodgy', 2022, 14000);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 5, NULL, 5, 'Dacia', 'Dokker', 2019, 13500);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 6, NULL, 1, 'Dacia', 'Spring', 2024, 16000);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 7, NULL, 2, 'Dacia', 'Duster', 2024, 15000);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 8, NULL, 3, 'Dacia', 'Sandero', 2022, 12000);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 9, NULL, 4, 'Dacia', 'Logan', 2021, 13000);

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (SECVENTA\_VEHICULE.NEXTVAL, 10, NULL, 1, 'Dacia', 'Lodgy', 2024, 14000);

INSERT INTO FACTURI (ID\_FACTURA, ID\_CLIENT, ID\_ANGAJAT, ID\_VEHICUL, DATA\_FACTURA, SUMA)

VALUES (SECVENTA\_FACTURI.NEXTVAL, 1, 1, 21, TO\_DATE('2021-07-01', 'YYYY-MM-DD'), 15000);

INSERT INTO FACTURI (ID\_FACTURA, ID\_CLIENT, ID\_ANGAJAT, ID\_VEHICUL, DATA\_FACTURA, SUMA)

VALUES (SECVENTA\_FACTURI.NEXTVAL, 2, 2, 22, TO\_DATE('2023-08-15', 'YYYY-MM-DD'), 12000);

INSERT INTO FACTURI (ID\_FACTURA, ID\_CLIENT, ID\_ANGAJAT, ID\_VEHICUL, DATA\_FACTURA, SUMA)

VALUES (SECVENTA\_FACTURI.NEXTVAL, 3, 3, 23, TO\_DATE('2023-09-10', 'YYYY-MM-DD'), 13000);

INSERT INTO FACTURI (ID\_FACTURA, ID\_CLIENT, ID\_ANGAJAT, ID\_VEHICUL, DATA\_FACTURA, SUMA)

VALUES (SECVENTA\_FACTURI.NEXTVAL, 4, 4, 24, TO\_DATE('2022-10-05', 'YYYY-MM-DD'), 14000);

INSERT INTO FACTURI (ID\_FACTURA, ID\_CLIENT, ID\_ANGAJAT, ID\_VEHICUL, DATA\_FACTURA, SUMA)

VALUES (SECVENTA\_FACTURI.NEXTVAL, 5, 5, 25, TO\_DATE('2019-11-20', 'YYYY-MM-DD'), 13500);

INSERT INTO FACTURI (ID\_FACTURA, ID\_CLIENT, ID\_ANGAJAT, ID\_VEHICUL, DATA\_FACTURA, SUMA)

VALUES (SECVENTA\_FACTURI.NEXTVAL, 1, 6, 26, TO\_DATE('2024-12-01', 'YYYY-MM-DD'), 16000);

INSERT INTO FACTURI (ID\_FACTURA, ID\_CLIENT, ID\_ANGAJAT, ID\_VEHICUL, DATA\_FACTURA, SUMA)

VALUES (SECVENTA\_FACTURI.NEXTVAL, 2, 7, 27, TO\_DATE('2024-12-15', 'YYYY-MM-DD'), 15000);

INSERT INTO FACTURI (ID\_FACTURA, ID\_CLIENT, ID\_ANGAJAT, ID\_VEHICUL, DATA\_FACTURA, SUMA)

VALUES (SECVENTA\_FACTURI.NEXTVAL, 3, 8, 28, TO\_DATE('2022-01-10', 'YYYY-MM-DD'), 12000);

INSERT INTO FACTURI (ID\_FACTURA, ID\_CLIENT, ID\_ANGAJAT, ID\_VEHICUL, DATA\_FACTURA, SUMA)

VALUES (SECVENTA\_FACTURI.NEXTVAL, 4, 1, 29, TO\_DATE('2021-02-05', 'YYYY-MM-DD'), 13000);

INSERT INTO FACTURI (ID\_FACTURA, ID\_CLIENT, ID\_ANGAJAT, ID\_VEHICUL, DATA\_FACTURA, SUMA)

VALUES (SECVENTA\_FACTURI.NEXTVAL, 1, 1, 30, TO\_DATE('2024-03-20', 'YYYY-MM-DD'), 14000);

INSERT INTO NUMERE\_TELEFON\_SHOWROOMURI (ID\_NUMAR\_TELEFON\_SHOWROOM, ID\_SHOWROOM, NUMAR\_TELEFON)

VALUES (SECVENTA\_NUMERE\_TELEFON\_SHOWROOM.NEXTVAL, 1, '0725 426 589');

INSERT INTO NUMERE\_TELEFON\_SHOWROOMURI (ID\_NUMAR\_TELEFON\_SHOWROOM, ID\_SHOWROOM, NUMAR\_TELEFON)

VALUES (SECVENTA\_NUMERE\_TELEFON\_SHOWROOM.NEXTVAL, 1, '0754 113 636');

INSERT INTO NUMERE\_TELEFON\_SHOWROOMURI (ID\_NUMAR\_TELEFON\_SHOWROOM, ID\_SHOWROOM, NUMAR\_TELEFON)

VALUES (SECVENTA\_NUMERE\_TELEFON\_SHOWROOM.NEXTVAL, 2, '0726 214 577');

INSERT INTO NUMERE\_TELEFON\_SHOWROOMURI (ID\_NUMAR\_TELEFON\_SHOWROOM, ID\_SHOWROOM, NUMAR\_TELEFON)

VALUES (SECVENTA\_NUMERE\_TELEFON\_SHOWROOM.NEXTVAL, 2, '0732 355 518');

INSERT INTO NUMERE\_TELEFON\_SHOWROOMURI (ID\_NUMAR\_TELEFON\_SHOWROOM, ID\_SHOWROOM, NUMAR\_TELEFON)

VALUES (SECVENTA\_NUMERE\_TELEFON\_SHOWROOM.NEXTVAL, 3, '0727 446 289');

INSERT INTO NUMERE\_TELEFON\_SHOWROOMURI (ID\_NUMAR\_TELEFON\_SHOWROOM, ID\_SHOWROOM, NUMAR\_TELEFON)

VALUES (SECVENTA\_NUMERE\_TELEFON\_SHOWROOM.NEXTVAL, 3, '0754 566 491');

INSERT INTO NUMERE\_TELEFON\_SHOWROOMURI (ID\_NUMAR\_TELEFON\_SHOWROOM, ID\_SHOWROOM, NUMAR\_TELEFON)

VALUES (SECVENTA\_NUMERE\_TELEFON\_SHOWROOM.NEXTVAL, 4, '0723 638 941');

INSERT INTO NUMERE\_TELEFON\_SHOWROOMURI (ID\_NUMAR\_TELEFON\_SHOWROOM, ID\_SHOWROOM, NUMAR\_TELEFON)

VALUES (SECVENTA\_NUMERE\_TELEFON\_SHOWROOM.NEXTVAL, 4, '0754 749 020');

INSERT INTO NUMERE\_TELEFON\_SHOWROOMURI (ID\_NUMAR\_TELEFON\_SHOWROOM, ID\_SHOWROOM, NUMAR\_TELEFON)

VALUES (SECVENTA\_NUMERE\_TELEFON\_SHOWROOM.NEXTVAL, 5, '0739 481 395');

INSERT INTO NUMERE\_TELEFON\_SHOWROOMURI (ID\_NUMAR\_TELEFON\_SHOWROOM, ID\_SHOWROOM, NUMAR\_TELEFON)

VALUES (SECVENTA\_NUMERE\_TELEFON\_SHOWROOM.NEXTVAL, 5, '0704 719 578');

INSERT INTO NUMERE\_TELEFON\_CLIENTI (ID\_NUMAR\_TELEFON\_CLIENT, ID\_CLIENT, NUMAR\_TELEFON)

VALUES (SECVENTA\_NUMERE\_TELEFON\_CLIENT.NEXTVAL, 1, '0793 765 902');

INSERT INTO NUMERE\_TELEFON\_CLIENTI (ID\_NUMAR\_TELEFON\_CLIENT, ID\_CLIENT, NUMAR\_TELEFON)

VALUES (SECVENTA\_NUMERE\_TELEFON\_CLIENT.NEXTVAL, 1, '0794 832 384');

INSERT INTO NUMERE\_TELEFON\_CLIENTI (ID\_NUMAR\_TELEFON\_CLIENT, ID\_CLIENT, NUMAR\_TELEFON)

VALUES (SECVENTA\_NUMERE\_TELEFON\_CLIENT.NEXTVAL, 2, '0748 200 555');

INSERT INTO NUMERE\_TELEFON\_CLIENTI (ID\_NUMAR\_TELEFON\_CLIENT, ID\_CLIENT, NUMAR\_TELEFON)

VALUES (SECVENTA\_NUMERE\_TELEFON\_CLIENT.NEXTVAL, 2, '0774 195 594');

INSERT INTO NUMERE\_TELEFON\_CLIENTI (ID\_NUMAR\_TELEFON\_CLIENT, ID\_CLIENT, NUMAR\_TELEFON)

VALUES (SECVENTA\_NUMERE\_TELEFON\_CLIENT.NEXTVAL, 3, '0758 102 333');

INSERT INTO NUMERE\_TELEFON\_CLIENTI (ID\_NUMAR\_TELEFON\_CLIENT, ID\_CLIENT, NUMAR\_TELEFON)

VALUES (SECVENTA\_NUMERE\_TELEFON\_CLIENT.NEXTVAL, 3, '0759 392 101');

INSERT INTO NUMERE\_TELEFON\_CLIENTI (ID\_NUMAR\_TELEFON\_CLIENT, ID\_CLIENT, NUMAR\_TELEFON)

VALUES (SECVENTA\_NUMERE\_TELEFON\_CLIENT.NEXTVAL, 4, '0754 393 121');

INSERT INTO NUMERE\_TELEFON\_CLIENTI (ID\_NUMAR\_TELEFON\_CLIENT, ID\_CLIENT, NUMAR\_TELEFON)

VALUES (SECVENTA\_NUMERE\_TELEFON\_CLIENT.NEXTVAL, 4, '0754 102 303');

INSERT INTO NUMERE\_TELEFON\_CLIENTI (ID\_NUMAR\_TELEFON\_CLIENT, ID\_CLIENT, NUMAR\_TELEFON)

VALUES (SECVENTA\_NUMERE\_TELEFON\_CLIENT.NEXTVAL, 5, '0749 584 431');

INSERT INTO NUMERE\_TELEFON\_CLIENTI (ID\_NUMAR\_TELEFON\_CLIENT, ID\_CLIENT, NUMAR\_TELEFON)

VALUES (SECVENTA\_NUMERE\_TELEFON\_CLIENT.NEXTVAL, 5, '0755 492 120');

INSERT INTO SHOWROOMURI\_ANGAJATI (ID\_SHOWROOM, ID\_ANGAJAT, PROGRAM) VALUES (1, 1, 'Luni-Miercuri 9-17');

INSERT INTO SHOWROOMURI\_ANGAJATI (ID\_SHOWROOM, ID\_ANGAJAT, PROGRAM) VALUES (1, 2, 'Luni-Vineri 9-17');

INSERT INTO SHOWROOMURI\_ANGAJATI (ID\_SHOWROOM, ID\_ANGAJAT, PROGRAM) VALUES (2, 3, 'Luni-Miercuri 9-17');

INSERT INTO SHOWROOMURI\_ANGAJATI (ID\_SHOWROOM, ID\_ANGAJAT, PROGRAM) VALUES (2, 4, 'Luni-Vineri 9-17');

INSERT INTO SHOWROOMURI\_ANGAJATI (ID\_SHOWROOM, ID\_ANGAJAT, PROGRAM) VALUES (3, 5, 'Luni-Miercuri 9-17');

INSERT INTO SHOWROOMURI\_ANGAJATI (ID\_SHOWROOM, ID\_ANGAJAT, PROGRAM) VALUES (3, 6, 'Luni-Vineri 9-17');

INSERT INTO SHOWROOMURI\_ANGAJATI (ID\_SHOWROOM, ID\_ANGAJAT, PROGRAM) VALUES (4, 7, 'Luni-Miercuri 9-17');

INSERT INTO SHOWROOMURI\_ANGAJATI (ID\_SHOWROOM, ID\_ANGAJAT, PROGRAM) VALUES (4, 8, 'Luni-Vineri 9-17');

INSERT INTO SHOWROOMURI\_ANGAJATI (ID\_SHOWROOM, ID\_ANGAJAT, PROGRAM) VALUES (5, 9, 'Luni-Miercuri 9-17');

INSERT INTO SHOWROOMURI\_ANGAJATI (ID\_SHOWROOM, ID\_ANGAJAT, PROGRAM) VALUES (5, 10, 'Luni-Vineri 9-17');

INSERT INTO SHOWROOMURI\_ANGAJATI (ID\_SHOWROOM, ID\_ANGAJAT, PROGRAM) VALUES (1, 3, 'Joi-Vineri 9-17');

INSERT INTO SHOWROOMURI\_ANGAJATI (ID\_SHOWROOM, ID\_ANGAJAT, PROGRAM) VALUES (2, 5, 'Joi-Vineri 9-17');

INSERT INTO SHOWROOMURI\_ANGAJATI (ID\_SHOWROOM, ID\_ANGAJAT, PROGRAM) VALUES (3, 7, 'Joi-Vineri 9-17');

INSERT INTO SHOWROOMURI\_ANGAJATI (ID\_SHOWROOM, ID\_ANGAJAT, PROGRAM) VALUES (4, 9, 'Joi-Vineri 9-17');

INSERT INTO SHOWROOMURI\_ANGAJATI (ID\_SHOWROOM, ID\_ANGAJAT, PROGRAM) VALUES (5, 1, 'Joi-Vineri 9-17');

INSERT INTO VEHICULE\_OPTIUNI (ID\_VEHICUL, ID\_OPTIUNE) VALUES (11, 1);

INSERT INTO VEHICULE\_OPTIUNI (ID\_VEHICUL, ID\_OPTIUNE) VALUES (11, 2);

INSERT INTO VEHICULE\_OPTIUNI (ID\_VEHICUL, ID\_OPTIUNE) VALUES (12, 3);

INSERT INTO VEHICULE\_OPTIUNI (ID\_VEHICUL, ID\_OPTIUNE) VALUES (12, 4);

INSERT INTO VEHICULE\_OPTIUNI (ID\_VEHICUL, ID\_OPTIUNE) VALUES (13, 5);

INSERT INTO VEHICULE\_OPTIUNI (ID\_VEHICUL, ID\_OPTIUNE) VALUES (13, 6);

INSERT INTO VEHICULE\_OPTIUNI (ID\_VEHICUL, ID\_OPTIUNE) VALUES (14, 7);

INSERT INTO VEHICULE\_OPTIUNI (ID\_VEHICUL, ID\_OPTIUNE) VALUES (14, 8);

INSERT INTO VEHICULE\_OPTIUNI (ID\_VEHICUL, ID\_OPTIUNE) VALUES (15, 9);

INSERT INTO VEHICULE\_OPTIUNI (ID\_VEHICUL, ID\_OPTIUNE) VALUES (15, 10);

INSERT INTO VEHICULE\_OPTIUNI (ID\_VEHICUL, ID\_OPTIUNE) VALUES (16, 1);

INSERT INTO VEHICULE\_OPTIUNI (ID\_VEHICUL, ID\_OPTIUNE) VALUES (16, 2);

INSERT INTO VEHICULE\_OPTIUNI (ID\_VEHICUL, ID\_OPTIUNE) VALUES (17, 3);

INSERT INTO VEHICULE\_OPTIUNI (ID\_VEHICUL, ID\_OPTIUNE) VALUES (17, 4);

INSERT INTO VEHICULE\_OPTIUNI (ID\_VEHICUL, ID\_OPTIUNE) VALUES (18, 5);

INSERT INTO VEHICULE\_OPTIUNI (ID\_VEHICUL, ID\_OPTIUNE) VALUES (18, 6);

INSERT INTO VEHICULE\_OPTIUNI (ID\_VEHICUL, ID\_OPTIUNE) VALUES (19, 7);

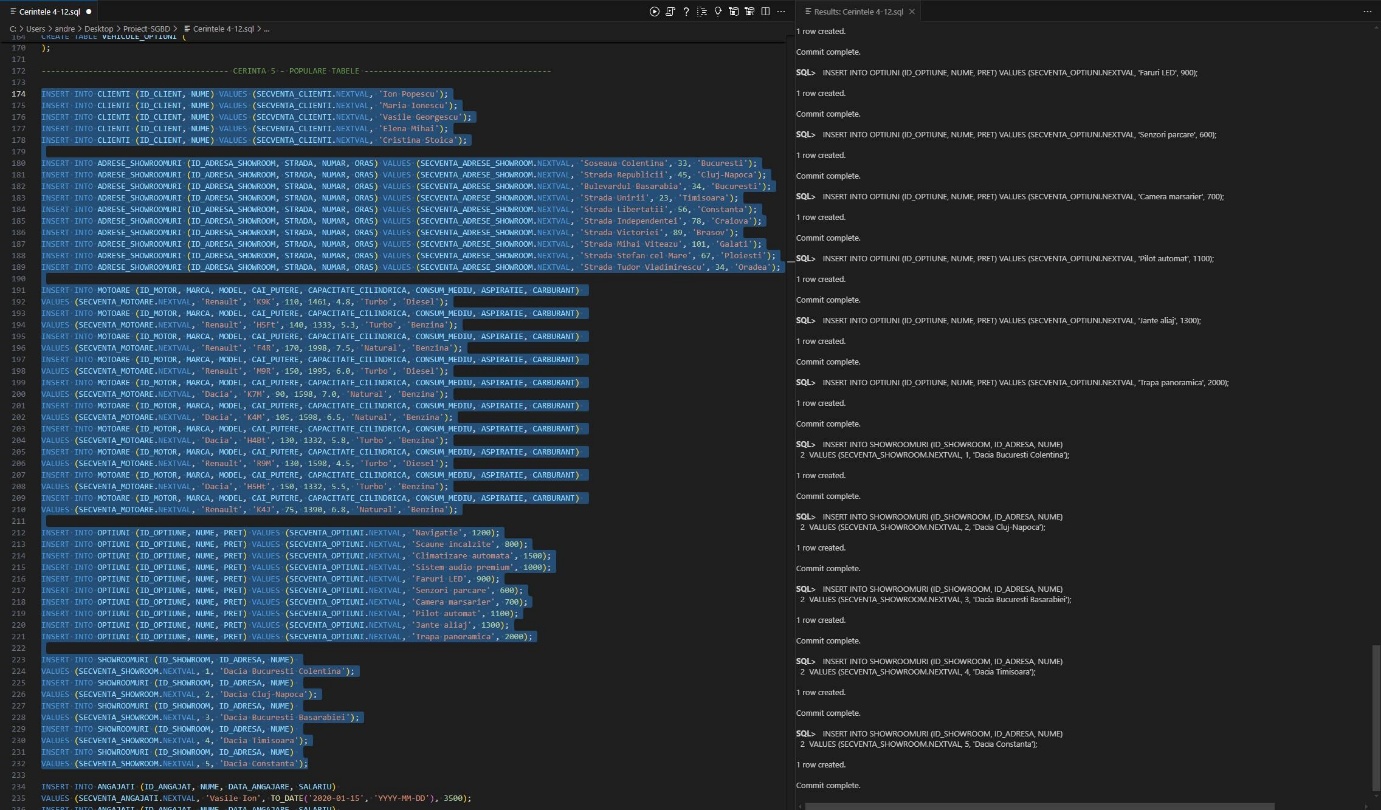
INSERT INTO VEHICULE\_OPTIUNI (ID\_VEHICUL, ID\_OPTIUNE) VALUES (19, 8);

INSERT INTO VEHICULE\_OPTIUNI (ID\_VEHICUL, ID\_OPTIUNE) VALUES (20, 9);

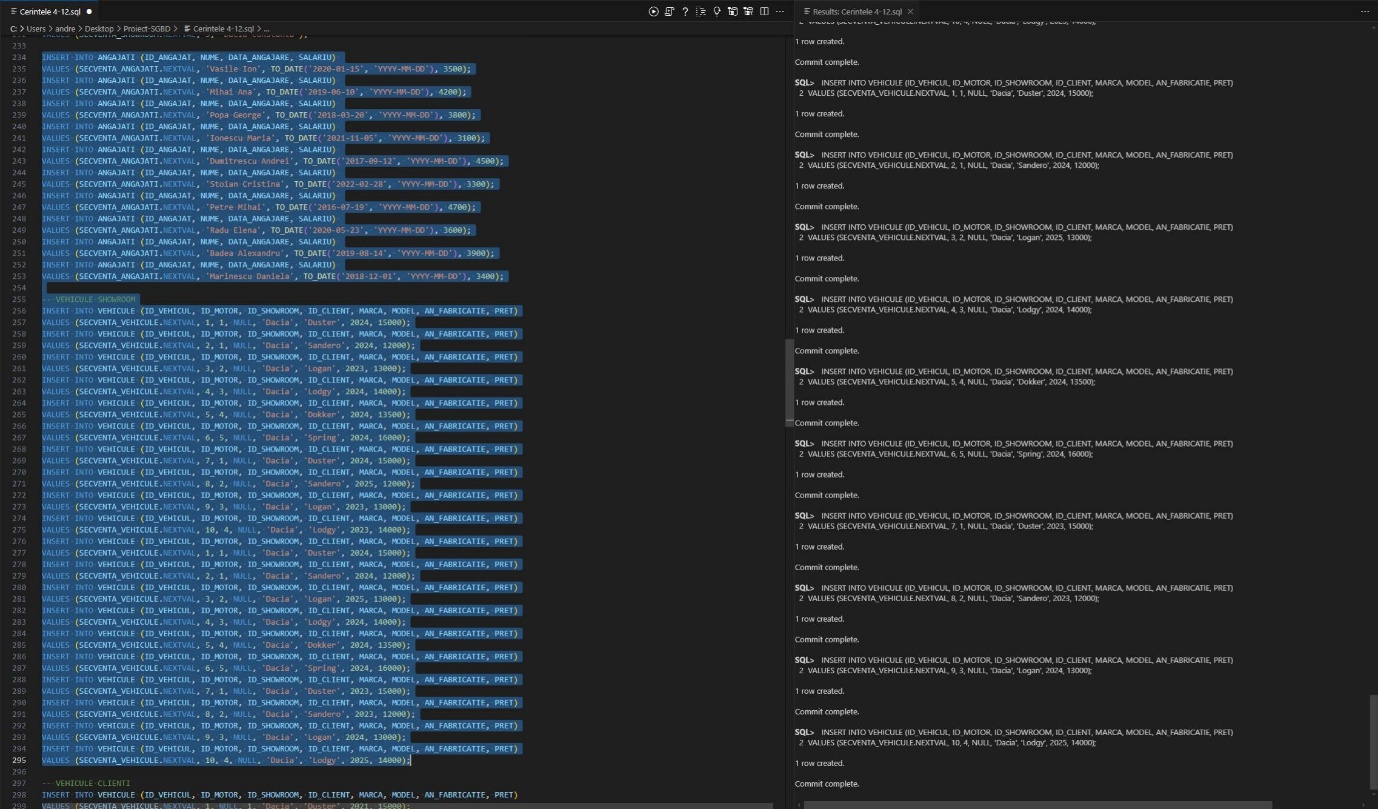
INSERT INTO VEHICULE\_OPTIUNI (ID\_VEHICUL, ID\_OPTIUNE) VALUES (20, 10);

**Rezultat rulare:**

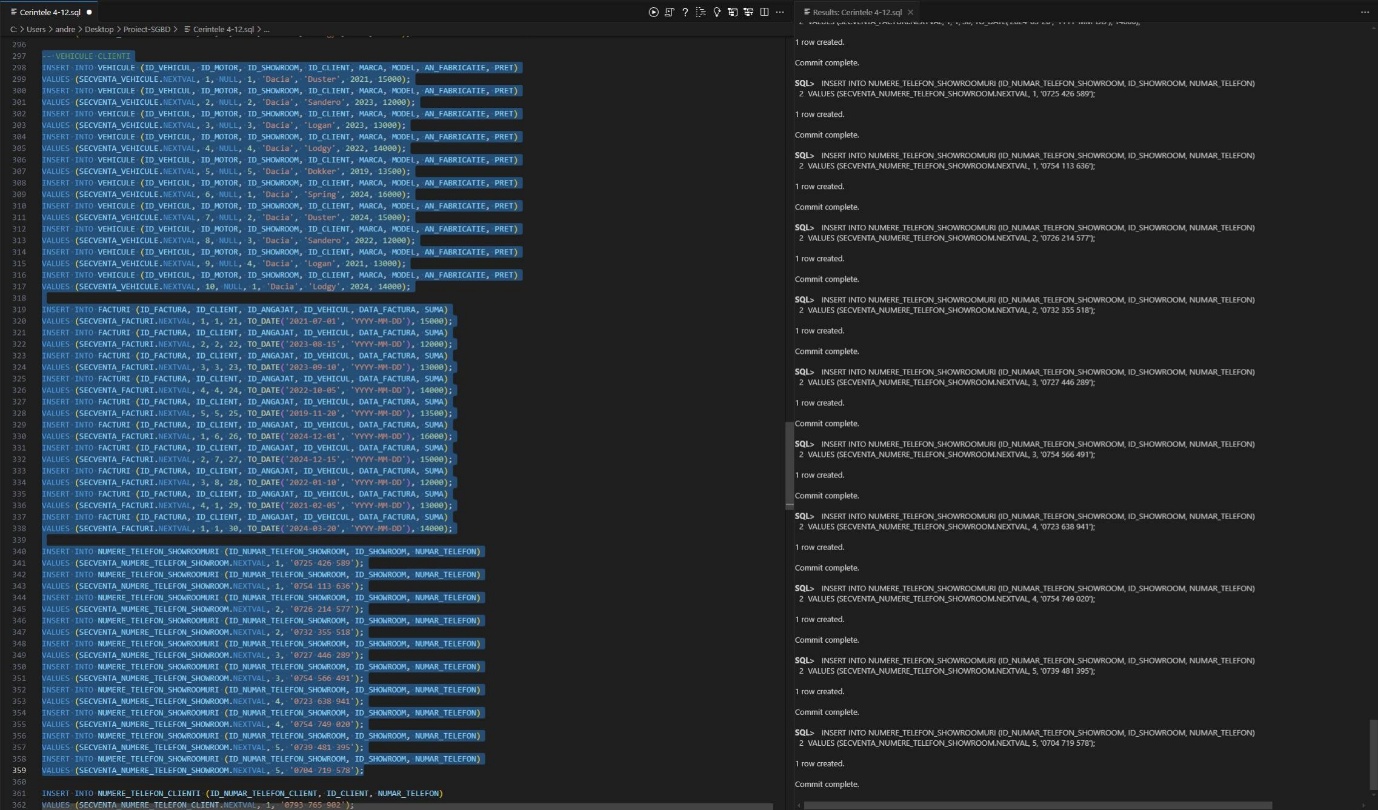
Poza 1/4:



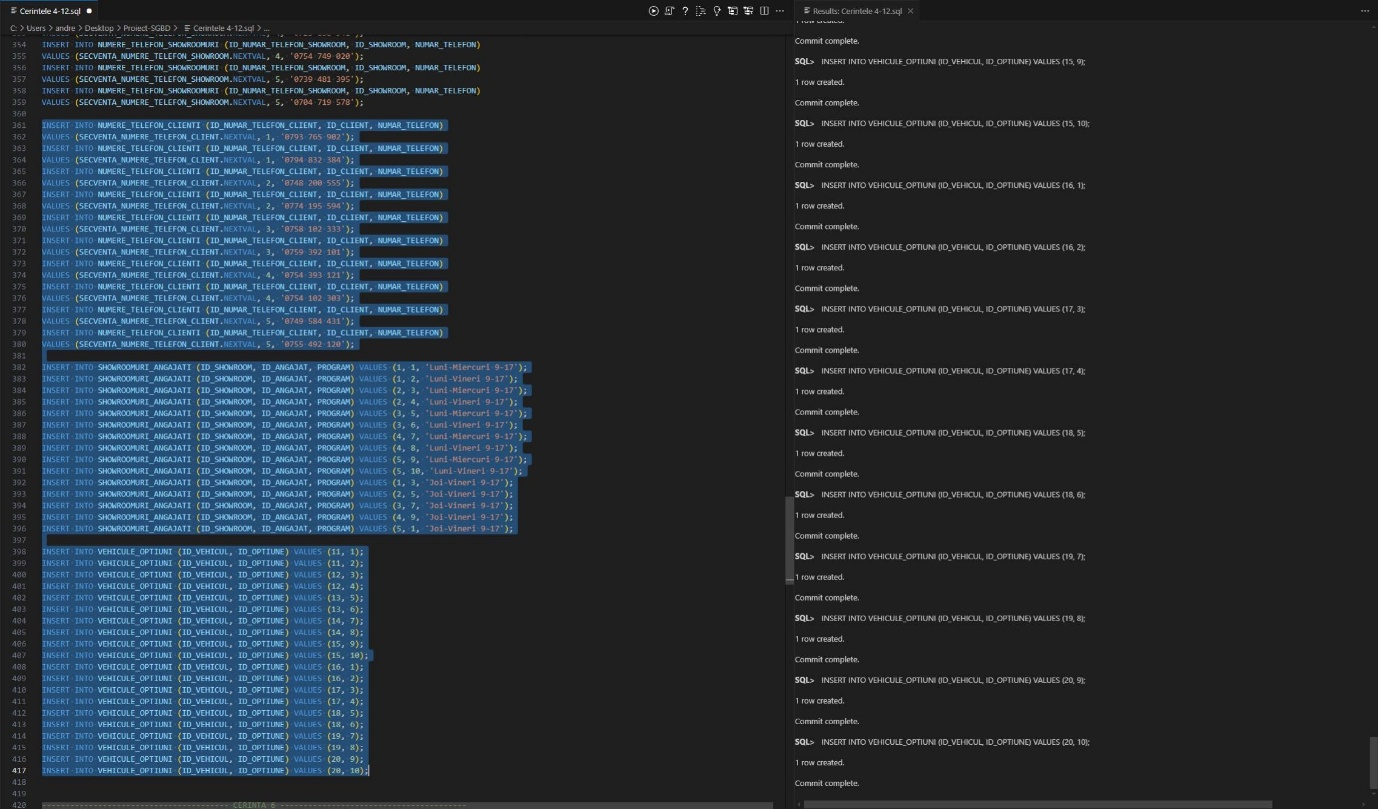
Poza 2/4:



Poza 3/4:



Poza 4/4:



# **6. Cerința 6 - Formulați în limbaj natural o problemă (3 tipuri de colecții)**

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.

**Enunț problema:**

Afișați pentru fiecare angajat:

* ID\_ANGAJAT
* Totalul sumelor facturilor emise de acesta
* Numărul de facturi emise de acesta

Rezultatul va fi ordonat descrescător după totalul sumelor facturilor emise de acesta.

**Cod PL/SQL:**

CREATE OR REPLACE PROCEDURE cerinta\_6 IS

    TYPE tip\_tablou\_indexat IS TABLE OF ANGAJATI.ID\_ANGAJAT%TYPE INDEX BY PLS\_INTEGER;

    TYPE tip\_tablou\_imbricat IS TABLE OF FACTURI.SUMA%TYPE INDEX BY PLS\_INTEGER;

    TYPE tip\_vector is VARRAY(100) OF NUMBER(6);

    t\_id\_angajati tip\_tablou\_indexat;

    t\_suma\_facturi tip\_tablou\_imbricat;

    t\_numar\_facturi tip\_vector := tip\_vector();

    v\_suma NUMBER(6) := 0;

    v\_contor NUMBER(6) := 0;

    aux\_id\_angajat ANGAJATI.ID\_ANGAJAT%TYPE;

    aux\_suma\_factura FACTURI.SUMA%TYPE;

    aux\_numar\_facturi NUMBER(6);

BEGIN

    -- ID\_ANGAJATI

    SELECT

        ID\_ANGAJAT

    BULK COLLECT INTO t\_id\_angajati

    FROM ANGAJATI;

    -- SUME FACTURI

    FOR i IN 1..t\_id\_angajati.COUNT LOOP

        SELECT

            NVL(SUM(SUMA), 0)

        INTO v\_suma

        FROM FACTURI

        WHERE ID\_ANGAJAT = t\_id\_angajati(i);

        t\_suma\_facturi(i) := v\_suma;

    END LOOP;

    -- NUMAR FACTURI

    FOR i IN 1..t\_id\_angajati.COUNT LOOP

SELECT

            COUNT(\*)

        INTO v\_contor

        FROM FACTURI

        WHERE ID\_ANGAJAT = t\_id\_angajati(i);

        t\_numar\_facturi.EXTEND;

        t\_numar\_facturi(i) := v\_contor;

    END LOOP;

    -- SORTARE DESCRESCATOR

    FOR i IN 1..t\_id\_angajati.COUNT LOOP

        FOR j IN i+1..t\_id\_angajati.COUNT LOOP

            IF t\_suma\_facturi(i) < t\_suma\_facturi(j) THEN

                aux\_id\_angajat := t\_id\_angajati(i);

                t\_id\_angajati(i) := t\_id\_angajati(j);

                t\_id\_angajati(j) := aux\_id\_angajat;

                aux\_suma\_factura := t\_suma\_facturi(i);

                t\_suma\_facturi(i) := t\_suma\_facturi(j);

                t\_suma\_facturi(j) := aux\_suma\_factura;

                aux\_numar\_facturi := t\_numar\_facturi(i);

                t\_numar\_facturi(i) := t\_numar\_facturi(j);

                t\_numar\_facturi(j) := aux\_numar\_facturi;

            END IF;

        END LOOP;

    END LOOP;

    -- AFISARE

    FOR i IN 1..t\_id\_angajati.COUNT LOOP

        DBMS\_OUTPUT.PUT\_LINE('--------------------------------------------------------------------');

        DBMS\_OUTPUT.PUT\_LINE('ID\_ANGAJAT: ' || t\_id\_angajati(i));

        DBMS\_OUTPUT.PUT\_LINE('TOTALUL SUMELOR FACTURILOR EMISE DE ACESTA: ' || t\_suma\_facturi(i));

        DBMS\_OUTPUT.PUT\_LINE('NUMARUL DE FACTURI EMISE DE ACESTA: ' || t\_numar\_facturi(i));

    END LOOP;

END;

/

BEGIN

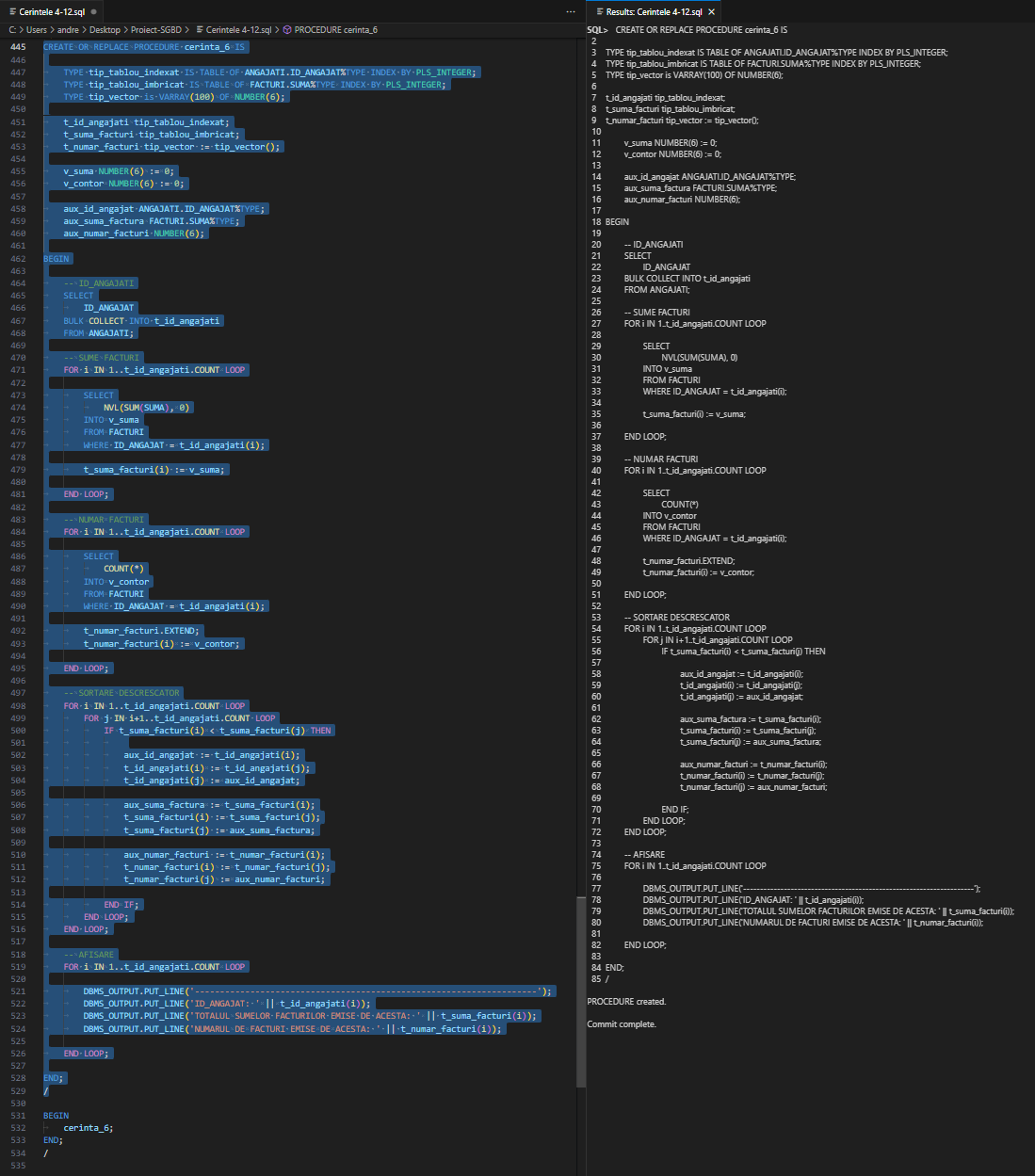
    cerinta\_6;

END;

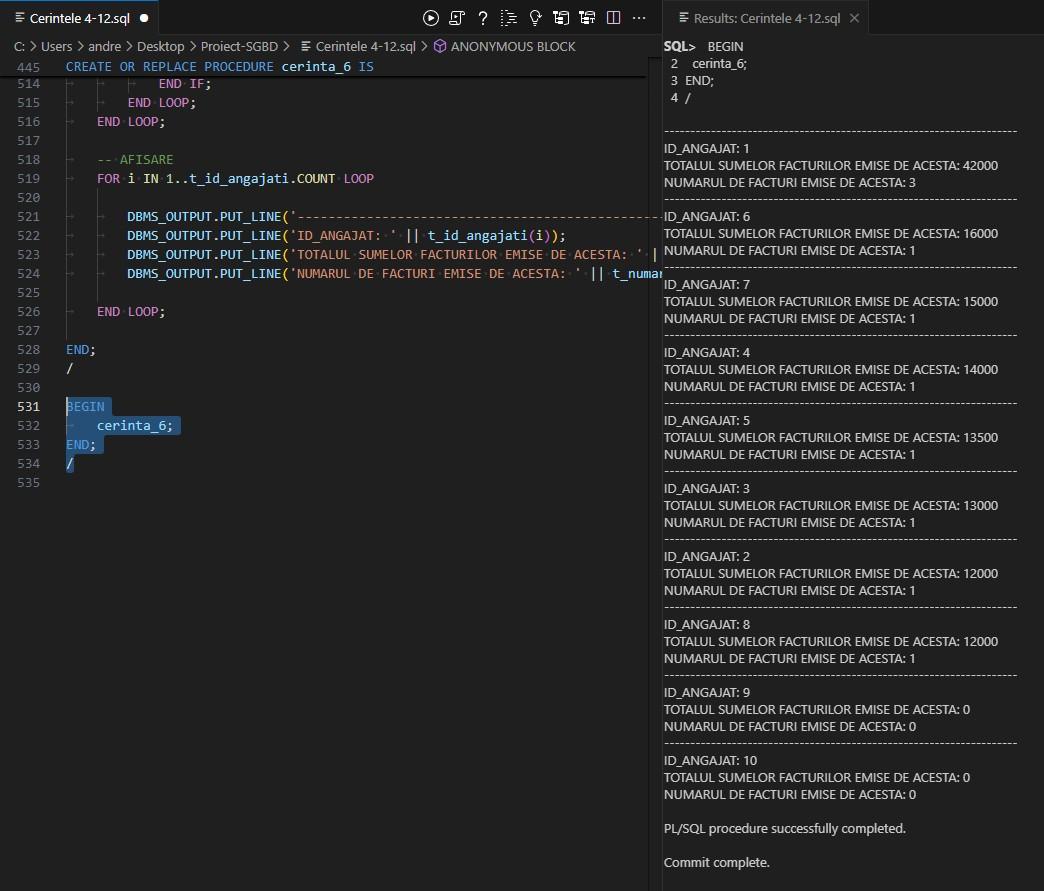
/

**Rezultat rulare:**

Poza 1/2:

****

Poza 2/2:



# **7. Cerința 7 - Formulați în limbaj natural o problemă (2 tipuri diferite de cursoare)**

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.

Sa se afișeze vehiculele deținute de către clienți care au fost vândute in anul 2024 (anul vânzării se regăsește in factura). Pentru fiecare vehicul sa se afișeze si angajatul care a efectuat vânzarea. La final sa se afișeze suma totala a vehiculelor vândute in 2024.

**Cod PL/SQL:**

CREATE OR REPLACE PROCEDURE cerinta\_7 IS

    CURSOR c\_vehicule\_clienti (p\_id\_vehicul\_factura FACTURI.ID\_VEHICUL%TYPE) IS

        SELECT

            ID\_VEHICUL,

            MARCA,

            MODEL,

            PRET

        FROM VEHICULE

        WHERE ID\_VEHICUL = p\_id\_vehicul\_factura;

    v\_cont NUMBER(6) := 0;

    v\_suma NUMBER(6) := 0;

    v\_nr\_vehicule\_clienti NUMBER(6) := 0;

    e\_nu\_exista\_vehicule EXCEPTION;

BEGIN

    -- verificare existenta vechiule vandute in 2024

    SELECT

        COUNT(\*)

    INTO v\_nr\_vehicule\_clienti

    FROM FACTURI

    WHERE EXTRACT(YEAR FROM DATA\_FACTURA) = 2024;

    IF v\_nr\_vehicule\_clienti = 0 THEN

        RAISE e\_nu\_exista\_vehicule;

    END IF;

DBMS\_OUTPUT.PUT\_LINE('---------------------------------------------');

    DBMS\_OUTPUT.PUT\_LINE('VEHICULE VANDUTE IN ANUL 2024:');

    FOR angajat\_factura\_2024 in (SELECT

                                    F.ID\_VEHICUL,

                                    A.ID\_ANGAJAT,

                                    A.NUME

                                FROM FACTURI F

                                JOIN ANGAJATI A ON F.ID\_ANGAJAT = A.ID\_ANGAJAT

                                WHERE EXTRACT(YEAR FROM DATA\_FACTURA) = 2024) LOOP

        FOR vehicul\_client in c\_vehicule\_clienti(angajat\_factura\_2024.ID\_VEHICUL) LOOP

            v\_cont := v\_cont + 1;

            v\_suma := v\_suma + vehicul\_client.PRET;

            DBMS\_OUTPUT.PUT\_LINE('----------------------' || v\_cont || '----------------------');

            DBMS\_OUTPUT.PUT\_LINE('ID\_VEHICUL: ' || vehicul\_client.ID\_VEHICUL);

            DBMS\_OUTPUT.PUT\_LINE('MARCA: ' || vehicul\_client.MARCA);

            DBMS\_OUTPUT.PUT\_LINE('MODEL: ' || vehicul\_client.MODEL);

            DBMS\_OUTPUT.PUT\_LINE('PRET: ' || vehicul\_client.PRET);

            DBMS\_OUTPUT.PUT\_LINE(' ');

            DBMS\_OUTPUT.PUT\_LINE('DATE ANGAJAT:');

            DBMS\_OUTPUT.PUT\_LINE('ID\_ANGAJAT: ' || angajat\_factura\_2024.ID\_ANGAJAT);

            DBMS\_OUTPUT.PUT\_LINE('NUME ANGAJAT: ' || angajat\_factura\_2024.NUME);

        END LOOP;

    END LOOP;

    DBMS\_OUTPUT.PUT\_LINE('---------------------------------------------------');

    DBMS\_OUTPUT.PUT\_LINE('SUMA TOTALA A VEHICULELOR VANDUTE IN 2024: ' || v\_suma);

EXCEPTION

    WHEN e\_nu\_exista\_vehicule THEN

        RAISE\_APPLICATION\_ERROR(-20001, 'NU EXISTA VEHICULE VANDUTE IN ANUL 2024');

    WHEN OTHERS THEN

        RAISE\_APPLICATION\_ERROR(-20000, 'EROARE: ' || SQLERRM);

END cerinta\_7;

/

BEGIN

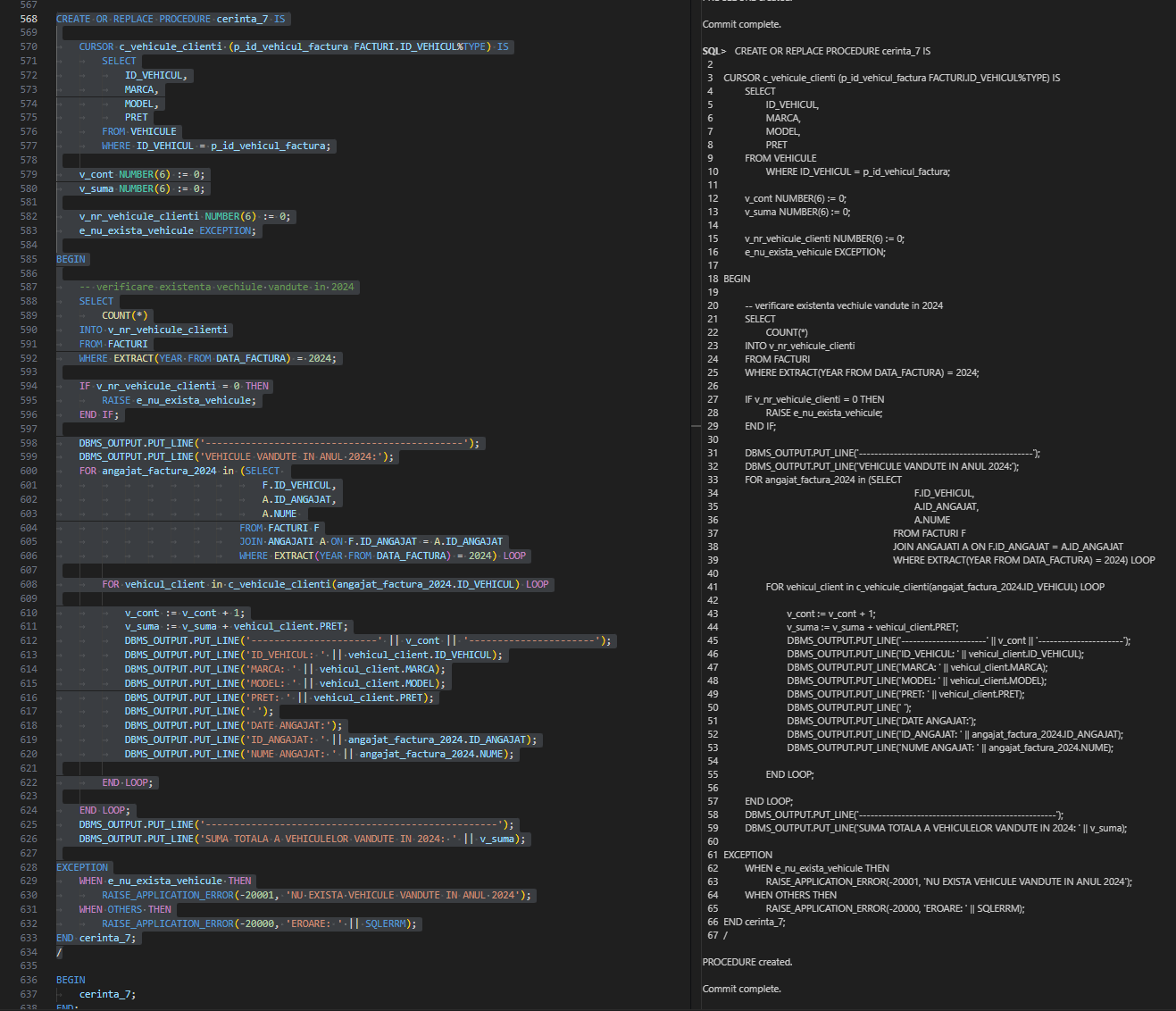
    cerinta\_7;

END;

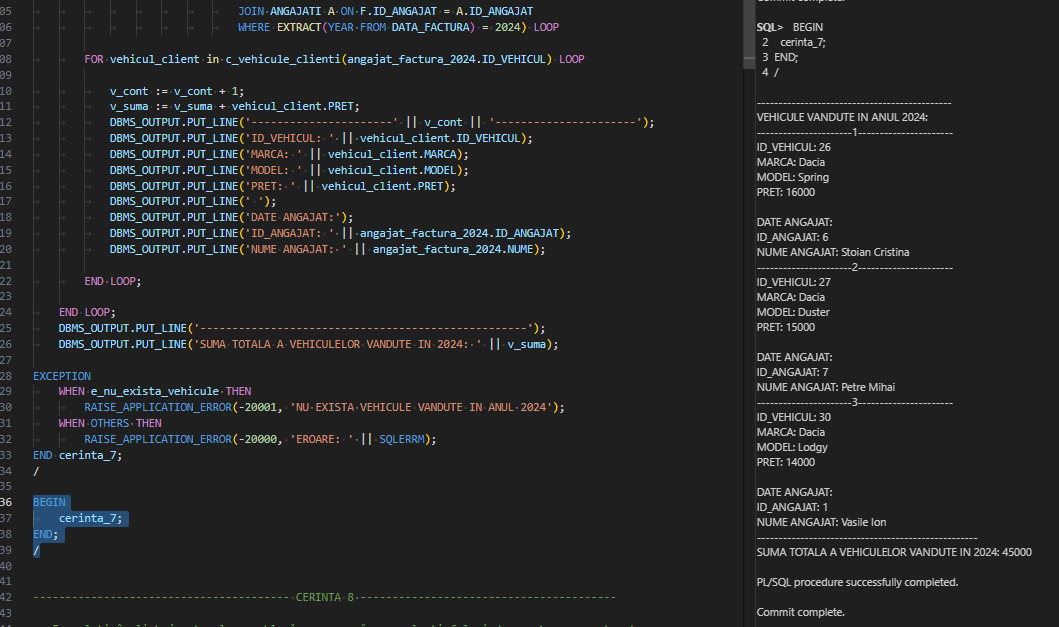
/

**Rezultat rulare:**

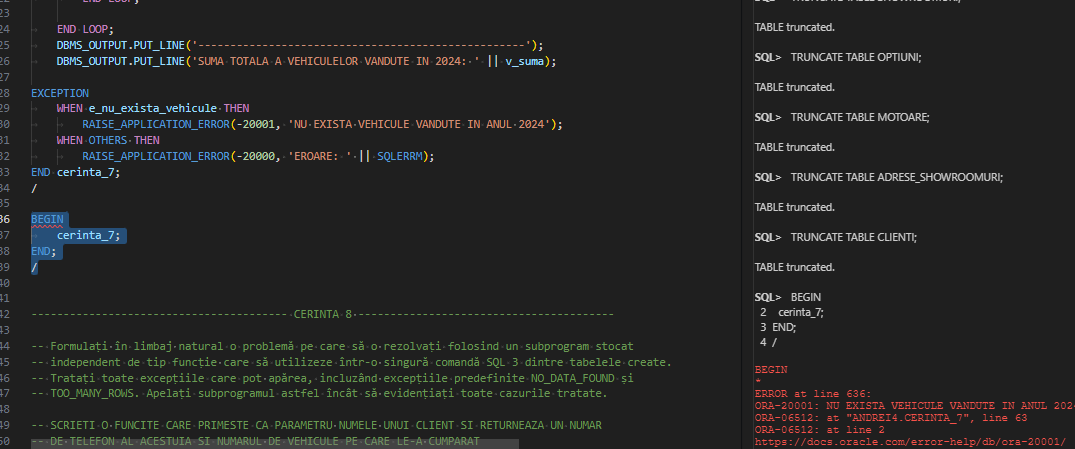
Poza 1/2:



Poza 2/2:



Poza excepție (după TRUNCATE TABLE VEHICULE):



# **8. Cerința 8 - Formulați în limbaj natural o problemă (să utilizeze într-o singură comandă SQL 3 dintre tabele)**

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.

Scrieți o funcție care primește ca parametru numele unui client si returnează un număr de telefon al acestuia si numărul de vehicule pe care le-a cumpărat. Daca nu exista clienți cu acel nume, se va afișa un mesaj. Daca exista cel puțin 2 clienți cu același nume, se va afișa un mesaj.

**Cod PL/SQL:**

CREATE OR REPLACE FUNCTION cerinta\_8(p\_nume\_client IN CLIENTI.NUME%TYPE) RETURN VARCHAR2 IS

    v\_nr\_clienti NUMBER(6) := 0;

    v\_numar\_telefon NUMERE\_TELEFON\_CLIENTI.NUMAR\_TELEFON%TYPE;

    v\_nr\_vehicule NUMBER(6) := 0;

    v\_rezultat VARCHAR2(256);

    -- EXCEPTII PROPRII

    e\_nu\_exista\_clienti EXCEPTION;

    e\_prea\_multi\_clienti EXCEPTION;

BEGIN

    SELECT

        COUNT(\*)

    INTO v\_nr\_clienti

    FROM CLIENTI

    WHERE UPPER(NUME) = UPPER('p\_nume\_client');

    -- EXCEPTII PROPRII

    -- IF v\_nr\_clienti = 0 THEN

    --  RAISE e\_nu\_exista\_clienti;

    -- ELSIF v\_nr\_clienti > 1 THEN

    --  RAISE e\_prea\_multi\_clienti;

    -- END IF;

    SELECT

        (SELECT

            NTC.NUMAR\_TELEFON

        FROM NUMERE\_TELEFON\_CLIENTI NTC

        WHERE C.ID\_CLIENT = NTC.ID\_CLIENT AND ROWNUM = 1

        ) AS NUMAR\_TELEFON,

        COUNT(V.ID\_VEHICUL) AS NUMAR\_VEHICULE

    INTO v\_numar\_telefon, v\_nr\_vehicule

    FROM CLIENTI C

    LEFT JOIN VEHICULE V ON C.ID\_CLIENT = V.ID\_CLIENT

    WHERE UPPER(C.NUME) = UPPER(p\_nume\_client)

    GROUP BY C.ID\_CLIENT;

    v\_rezultat := 'Clientul cu numele ' || p\_nume\_client || ' are numarul de telefon ' || v\_numar\_telefon || ' si detine ' || v\_nr\_vehicule || ' vehicule.';

    RETURN v\_rezultat;

EXCEPTION

    -- EXCEPTII PROPRII

    -- WHEN e\_nu\_exista\_clienti THEN

    --  RAISE\_APPLICATION\_ERROR(-20001, 'NU EXISTA CLIENTI CU ACEST NUME');

    -- WHEN e\_prea\_multi\_clienti THEN

    --  RAISE\_APPLICATION\_ERROR(-20002, 'EXISTA MAI MULTI CLIENTI CU ACEST NUME');

    WHEN TOO\_MANY\_ROWS THEN

        RAISE\_APPLICATION\_ERROR(-20004, 'TOO\_MANY\_ROWS ' || SQLERRM);

    WHEN NO\_DATA\_FOUND THEN

        RAISE\_APPLICATION\_ERROR(-20003, 'NO\_DATA\_FOUND ' || SQLERRM);

    WHEN OTHERS THEN

        RAISE\_APPLICATION\_ERROR(-20000, 'EROARE: ' || SQLERRM);

END cerinta\_8;

/

DECLARE

    v\_rezultat VARCHAR2(256);

BEGIN

    v\_rezultat := cerinta\_8('ION POPESCU');

    DBMS\_OUTPUT.PUT\_LINE(v\_rezultat);

    v\_rezultat := cerinta\_8('CLIENT DUPLICAT');

    DBMS\_OUTPUT.PUT\_LINE(v\_rezultat);

    v\_rezultat := cerinta\_8('CLIENT INEXISTENT');

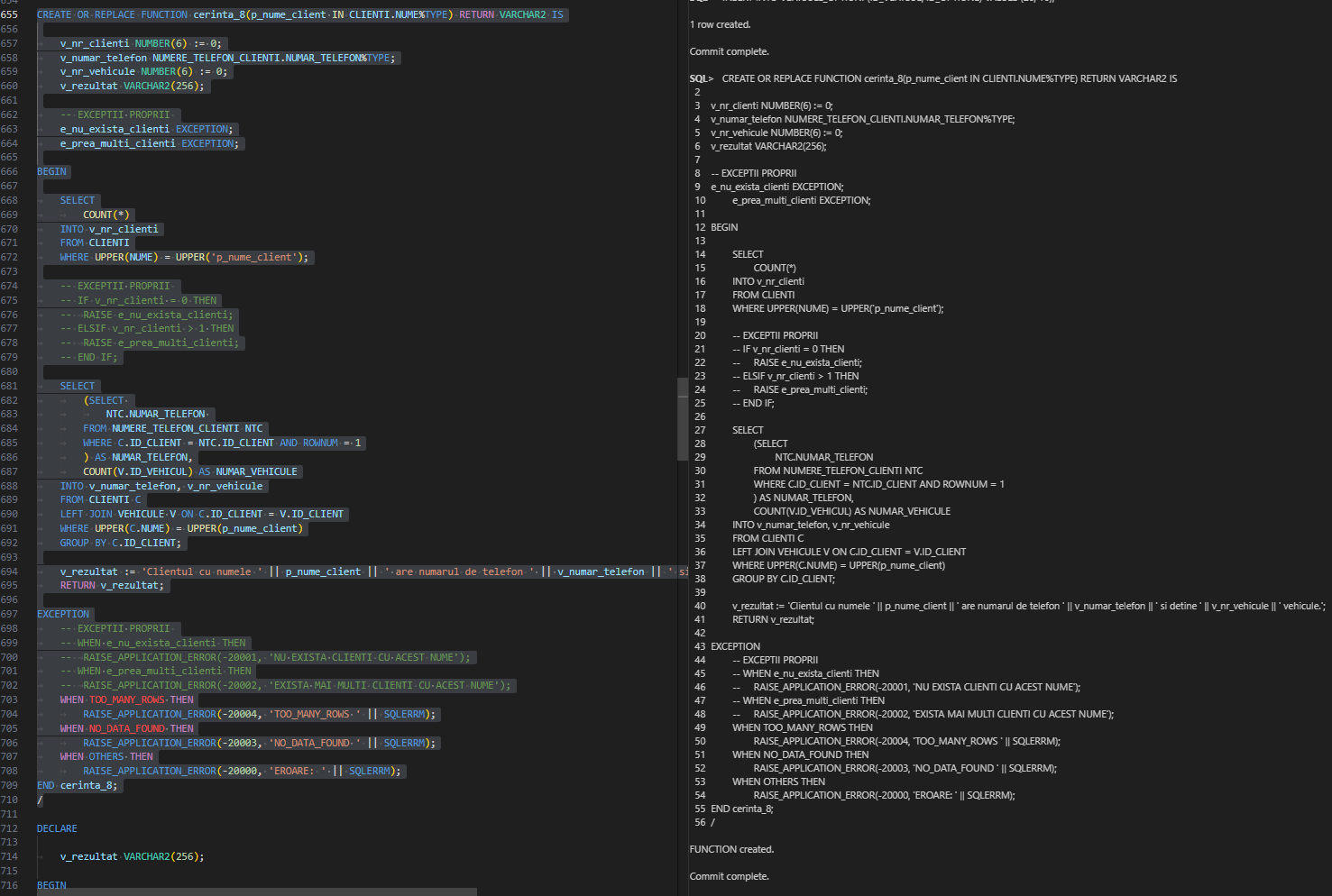
    DBMS\_OUTPUT.PUT\_LINE(v\_rezultat);

END;

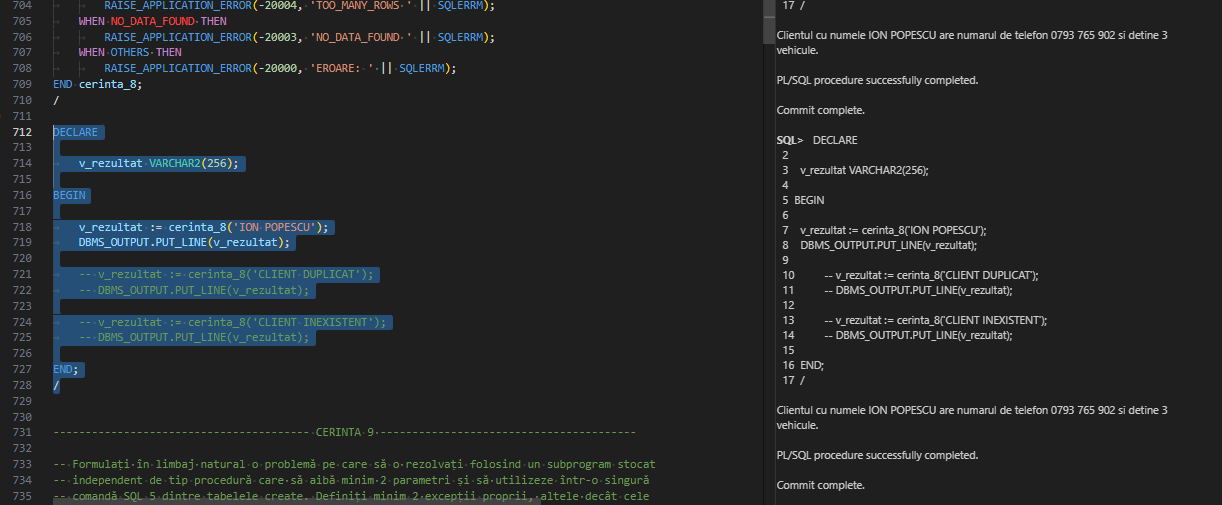
/

**Rezultat rulare:**

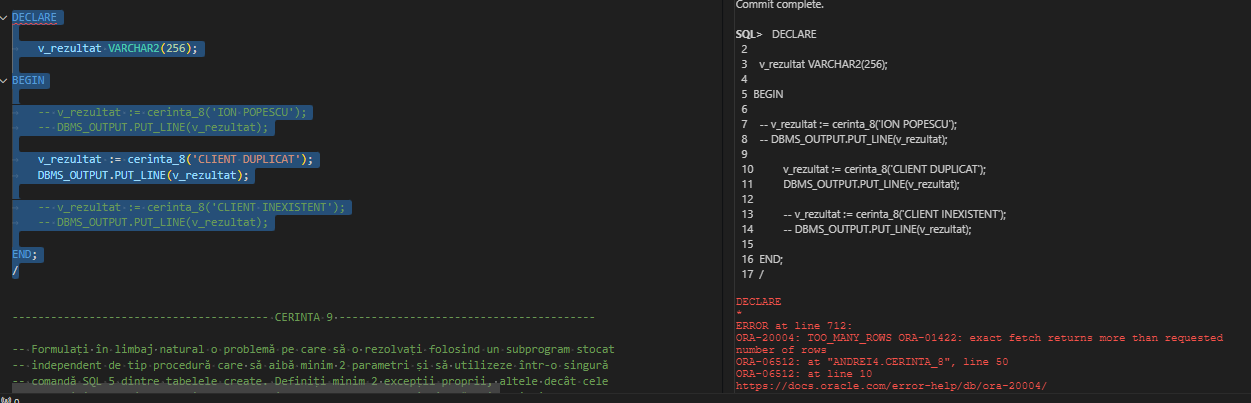
Poza 1/4:



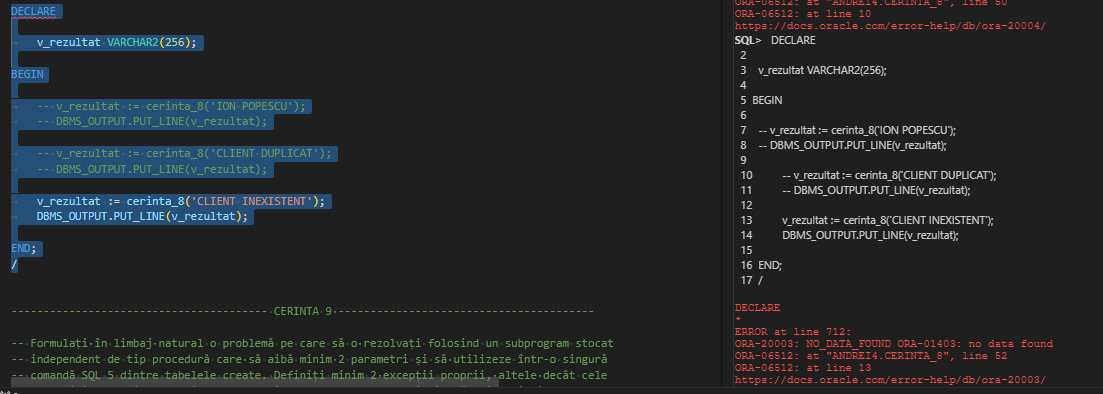
Poza 2/4:



Poza 3/4 (TOO\_MANY\_ROWS):



Poza 4/4 (NO\_DATA\_FOUND):



# **9. Cerința 9 - Formulați în limbaj natural o problemă (să utilizeze într-o singură comandă SQL 5 dintre tabele)**

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.

Definiți o procedură care primește ca parametri ID\_VEHICUL, MARCA și MODELUL unui vehicul. Procedura afișează anul fabricației și prețul vehiculului, informații despre motorul vehiculului, o listă cu opțiunile extra ale vehiculului, date despre client dacă vehiculul este deținut de către un client, altfel se afișează date despre showroom. Dacă nu există vehicule cu această marcă și model, se va afișa un mesaj corespunzător. Dacă nu are opțiuni extra, atunci se va afișa un mesaj corespunzător.

**Cod PL/SQL:**

CREATE OR REPLACE PROCEDURE cerinta\_9(p\_id\_vehicul IN VEHICULE.ID\_VEHICUL%TYPE, p\_marca IN VEHICULE.MARCA%TYPE, p\_model IN VEHICULE.MARCA%TYPE) IS

    TYPE r\_cerinta\_9 IS RECORD (

        id\_vehicul VEHICULE.ID\_VEHICUL%TYPE,

        marca VEHICULE.MARCA%TYPE,

        model VEHICULE.MODEL%TYPE,

        an\_fabricatie VEHICULE.AN\_FABRICATIE%TYPE,

        pret VEHICULE.PRET%TYPE,

        id\_motor MOTOARE.ID\_MOTOR%TYPE,

        cai\_putere MOTOARE.CAI\_PUTERE%TYPE,

        capacitate\_cilindrica MOTOARE.CAPACITATE\_CILINDRICA%TYPE,

        nume\_client CLIENTI.NUME%TYPE,

        -- SAU

        nume\_showroom SHOWROOMURI.NUME%TYPE,

        optiuni VARCHAR2(2048)

    );

    v\_r\_cerinta\_9 r\_cerinta\_9;

    v\_nr\_vehicule NUMBER(6) := 0;

    v\_nr\_optiuni NUMBER(6) := 0;

    e\_nu\_exista\_vehicule EXCEPTION;

    e\_nu\_are\_optiuni EXCEPTION;

BEGIN

    -- EXTRAGERE NUMARUL DE VEHICULE CU COMBINATIA p\_id\_vehicul, p\_marca, p\_model

    SELECT

        COUNT(\*)

    INTO v\_nr\_vehicule

    FROM VEHICULE

    WHERE ID\_VEHICUL = p\_id\_vehicul AND UPPER(MARCA) = UPPER(p\_marca)

        AND UPPER(MODEL) = UPPER(p\_model);

    IF v\_nr\_vehicule = 0 THEN

        RAISE e\_nu\_exista\_vehicule;

    END IF;

    -- EXTRAGERE NUMAR OPTIUNI

    SELECT

        COUNT(\*)

    INTO v\_nr\_optiuni

    FROM VEHICULE\_OPTIUNI

    WHERE ID\_VEHICUL = p\_id\_vehicul;

    SELECT

        V.ID\_VEHICUL,

        V.MARCA,

        V.MODEL,

        V.AN\_FABRICATIE,

        V.PRET,

        M.ID\_MOTOR,

        M.CAI\_PUTERE,

        M.CAPACITATE\_CILINDRICA,

        C.NUME,

        S.NUME,

        LISTAGG(O.NUME, ', ') WITHIN GROUP (ORDER BY O.ID\_OPTIUNE)

    INTO v\_r\_cerinta\_9.id\_vehicul, v\_r\_cerinta\_9.marca, v\_r\_cerinta\_9.model,

            v\_r\_cerinta\_9.an\_fabricatie, v\_r\_cerinta\_9.pret, v\_r\_cerinta\_9.id\_motor,

            v\_r\_cerinta\_9.cai\_putere, v\_r\_cerinta\_9.capacitate\_cilindrica,

            v\_r\_cerinta\_9.nume\_client, v\_r\_cerinta\_9.nume\_showroom, v\_r\_cerinta\_9.optiuni

    FROM VEHICULE V

    JOIN MOTOARE M ON V.ID\_MOTOR = M.ID\_MOTOR

    LEFT JOIN CLIENTI C ON V.ID\_CLIENT = C.ID\_CLIENT

    LEFT JOIN SHOWROOMURI S ON V.ID\_SHOWROOM = S.ID\_SHOWROOM

    LEFT JOIN VEHICULE\_OPTIUNI VO ON V.ID\_VEHICUL = VO.ID\_VEHICUL

    LEFT JOIN OPTIUNI O ON VO.ID\_OPTIUNE = O.ID\_OPTIUNE

    WHERE V.ID\_VEHICUL = p\_id\_vehicul AND UPPER(V.MARCA) = UPPER(p\_marca)

            AND UPPER(V.MODEL) = UPPER(p\_model)

    GROUP BY V.ID\_VEHICUL, V.MARCA, V.MODEL, V.AN\_FABRICATIE, V.PRET, M.ID\_MOTOR,

                M.CAI\_PUTERE, M.CAPACITATE\_CILINDRICA, C.NUME, S.NUME;

    -- AFISARE VEHICUL

    DBMS\_OUTPUT.PUT\_LINE('------------------ DATE VEHICUL ------------------');

    DBMS\_OUTPUT.PUT\_LINE('ID\_VEHICUL: ' || v\_r\_cerinta\_9.id\_vehicul);

    DBMS\_OUTPUT.PUT\_LINE('MARCA: ' || v\_r\_cerinta\_9.marca);

    DBMS\_OUTPUT.PUT\_LINE('MODEL: ' || v\_r\_cerinta\_9.model);

    DBMS\_OUTPUT.PUT\_LINE('AN FABRICATIE: ' || v\_r\_cerinta\_9.an\_fabricatie);

    DBMS\_OUTPUT.PUT\_LINE('PRET: ' || v\_r\_cerinta\_9.pret);

    -- AFISARE MOTOR

    DBMS\_OUTPUT.PUT\_LINE(' ');

    DBMS\_OUTPUT.PUT\_LINE('------------------ DATE MOTOR --------------------');

    DBMS\_OUTPUT.PUT\_LINE('ID\_MOTOR: ' || v\_r\_cerinta\_9.id\_motor);

    DBMS\_OUTPUT.PUT\_LINE('CAI PUTERE: ' || v\_r\_cerinta\_9.cai\_putere);

    DBMS\_OUTPUT.PUT\_LINE('CAPACITATE CILINDRICA: ' || v\_r\_cerinta\_9.capacitate\_cilindrica);

    -- AFISARE OPTIUNI

    BEGIN

    DBMS\_OUTPUT.PUT\_LINE(' ');

        DBMS\_OUTPUT.PUT\_LINE('------------------ DATE OPTIUNI --------------------');

        IF v\_nr\_optiuni = 0 THEN

            RAISE e\_nu\_are\_optiuni;

        ELSE

            DBMS\_OUTPUT.PUT\_LINE('OPTIUNI EXTRA: ' || v\_r\_cerinta\_9.optiuni);

        END IF;

    EXCEPTION

        WHEN e\_nu\_are\_optiuni THEN

            DBMS\_OUTPUT.PUT\_LINE('VEHICULUL NU ARE OPTIUNI EXTRA');

    END;

    -- AFISARE CLIENT/SHOWROOM

    IF v\_r\_cerinta\_9.nume\_client IS NOT NULL THEN

        DBMS\_OUTPUT.PUT\_LINE(' ');

        DBMS\_OUTPUT.PUT\_LINE('------------------ DATE CLIENT ------------------');

        DBMS\_OUTPUT.PUT\_LINE('NUME CLIENT: ' || v\_r\_cerinta\_9.nume\_client);

    ELSE

        DBMS\_OUTPUT.PUT\_LINE(' ');

        DBMS\_OUTPUT.PUT\_LINE('------------------ DATE SHOWROOM ---------------');

        DBMS\_OUTPUT.PUT\_LINE('NUME SHOWROOM: ' || v\_r\_cerinta\_9.nume\_showroom);

    END IF;

EXCEPTION

    WHEN e\_nu\_exista\_vehicule THEN

        RAISE\_APPLICATION\_ERROR(-20001, 'NU EXISTA VEHICULE CU ACEST ID, MARCA SI MODEL');

    WHEN OTHERS THEN

        RAISE\_APPLICATION\_ERROR(-20000, 'EROARE: ' || SQLERRM);

END cerinta\_9;

/

BEGIN

    cerinta\_9(1, 'DACIA', 'DUSTER');

    DBMS\_OUTPUT.PUT\_LINE(' ');

    DBMS\_OUTPUT.PUT\_LINE(' ');

    DBMS\_OUTPUT.PUT\_LINE('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

    DBMS\_OUTPUT.PUT\_LINE(' ');

    DBMS\_OUTPUT.PUT\_LINE(' ');

    cerinta\_9(11, 'DACIA', 'DUSTER');

    DBMS\_OUTPUT.PUT\_LINE(' ');

    DBMS\_OUTPUT.PUT\_LINE(' ');

    DBMS\_OUTPUT.PUT\_LINE('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

    DBMS\_OUTPUT.PUT\_LINE(' ');

    DBMS\_OUTPUT.PUT\_LINE(' ');

    cerinta\_9(21, 'DACIA', 'DUSTER');

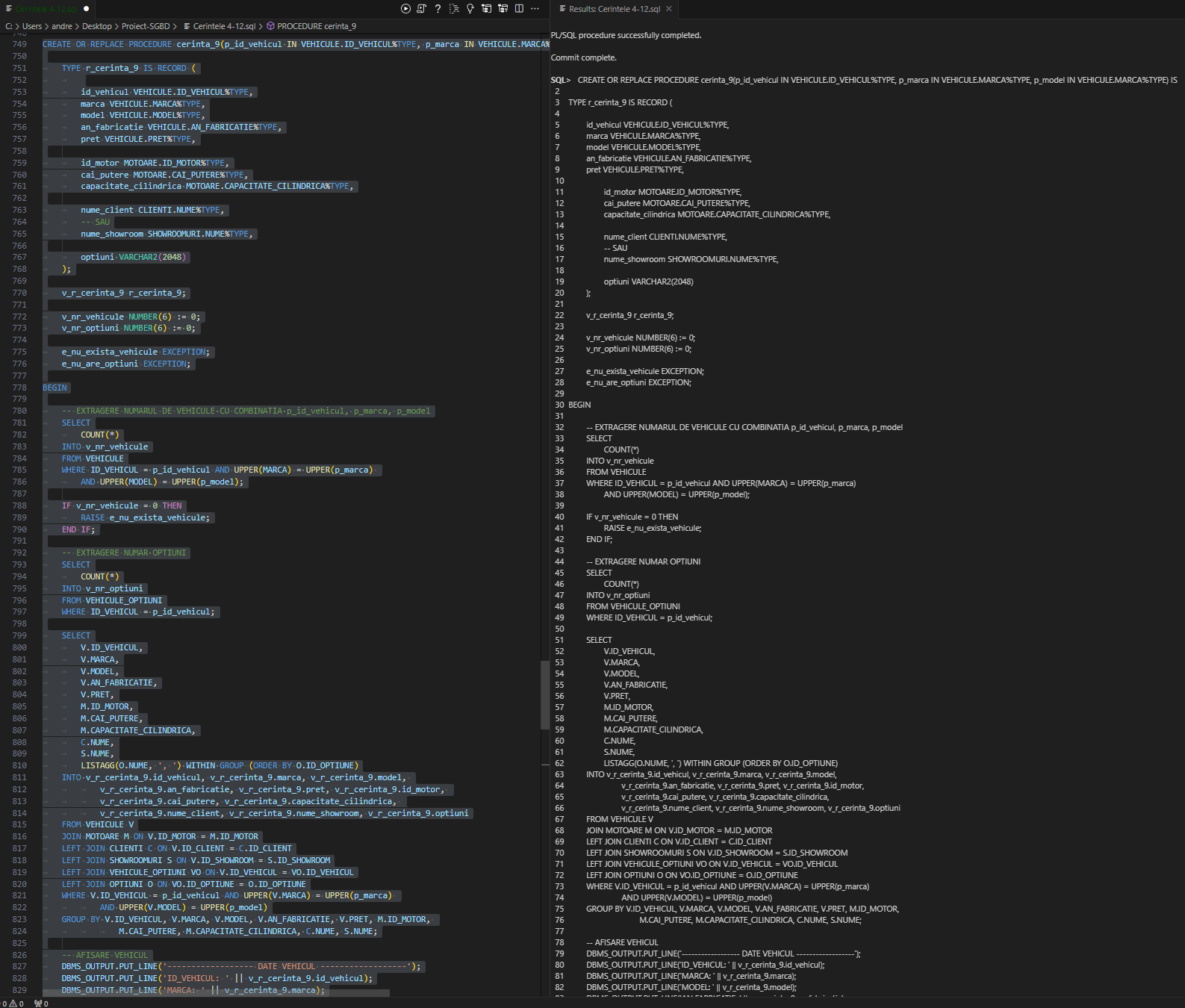
    cerinta\_9(-1, 'VEHICUL', 'INEXISTENT');

END;

/

**Rezultat rulare:**

Poza 1/4:



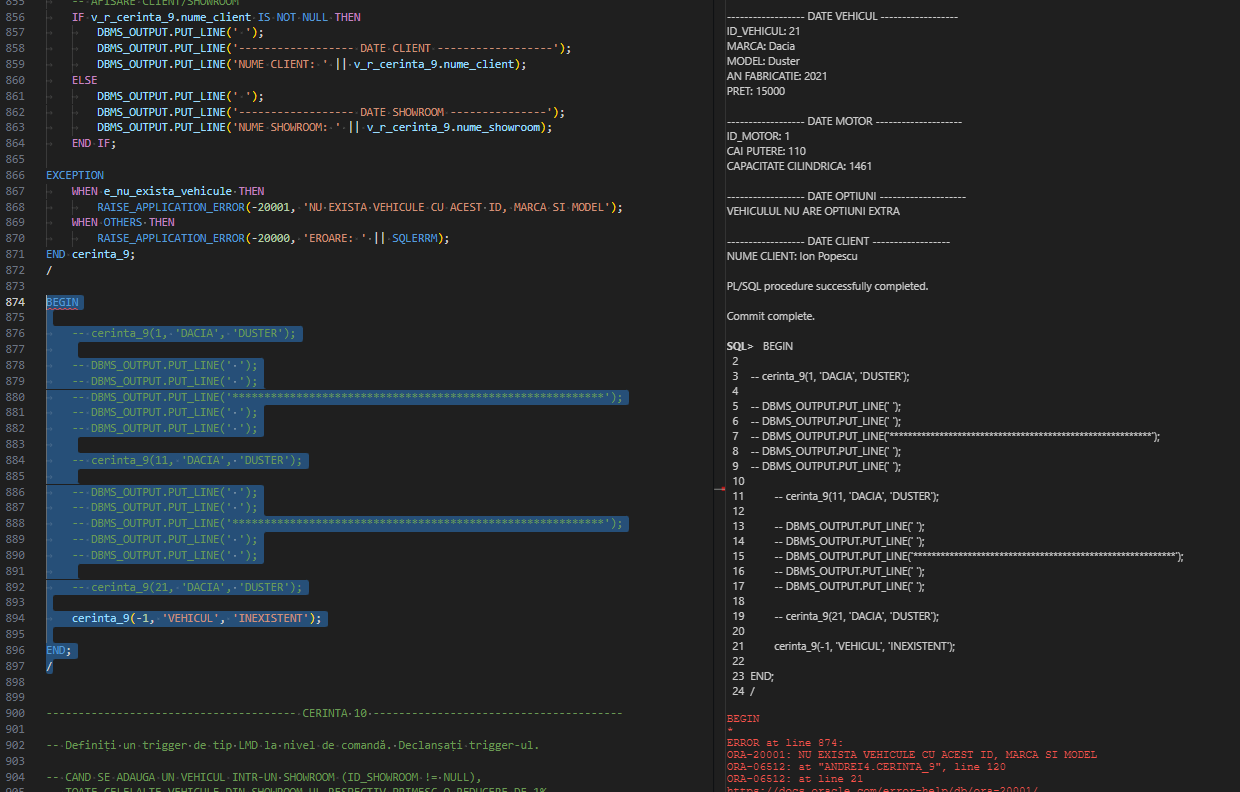
Poza 2/4:



Poza 3/4:



Poza 4/4 (vehicul inexistent):



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

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

Când se adaugă un vehicul într-un showroom (Vehicul cu ID\_SHOWROOM != NULL), toate celelalte vehicule din showroom-ul respectiv primesc o reducere de 1%. Noul vehicul adăugat nu primește reducere.

**Cod PL/SQL:**

CREATE OR REPLACE TRIGGER trigger\_cerinta\_10

AFTER INSERT ON VEHICULE

DECLARE

    v\_id\_showroom VEHICULE.ID\_SHOWROOM%TYPE;

BEGIN

    DBMS\_OUTPUT.PUT\_LINE(' ');

    DBMS\_OUTPUT.PUT\_LINE('-------------------- DECLANSARE TRIGGER CERINTA 10 -----------');

    DBMS\_OUTPUT.PUT\_LINE(' ');

    SELECT

        ID\_SHOWROOM

    INTO v\_id\_showroom

    FROM VEHICULE

    WHERE ID\_VEHICUL = (SELECT

                            MAX(ID\_VEHICUL)

                        FROM VEHICULE);

    IF v\_id\_showroom IS NOT NULL THEN -- altfel este vehiculul unui client

        UPDATE VEHICULE

        SET PRET = PRET \* 0.99

        WHERE ID\_SHOWROOM = v\_id\_showroom

        AND ID\_VEHICUL != (SELECT

                              MAX(id\_vehicul)

                           FROM vehicule);

    END IF;

END trigger\_cerinta\_10;

/

DROP TRIGGER trigger\_cerinta\_10;

-- EXEMPLU

SELECT ID\_VEHICUL, PRET FROM VEHICULE WHERE ID\_SHOWROOM = 1;

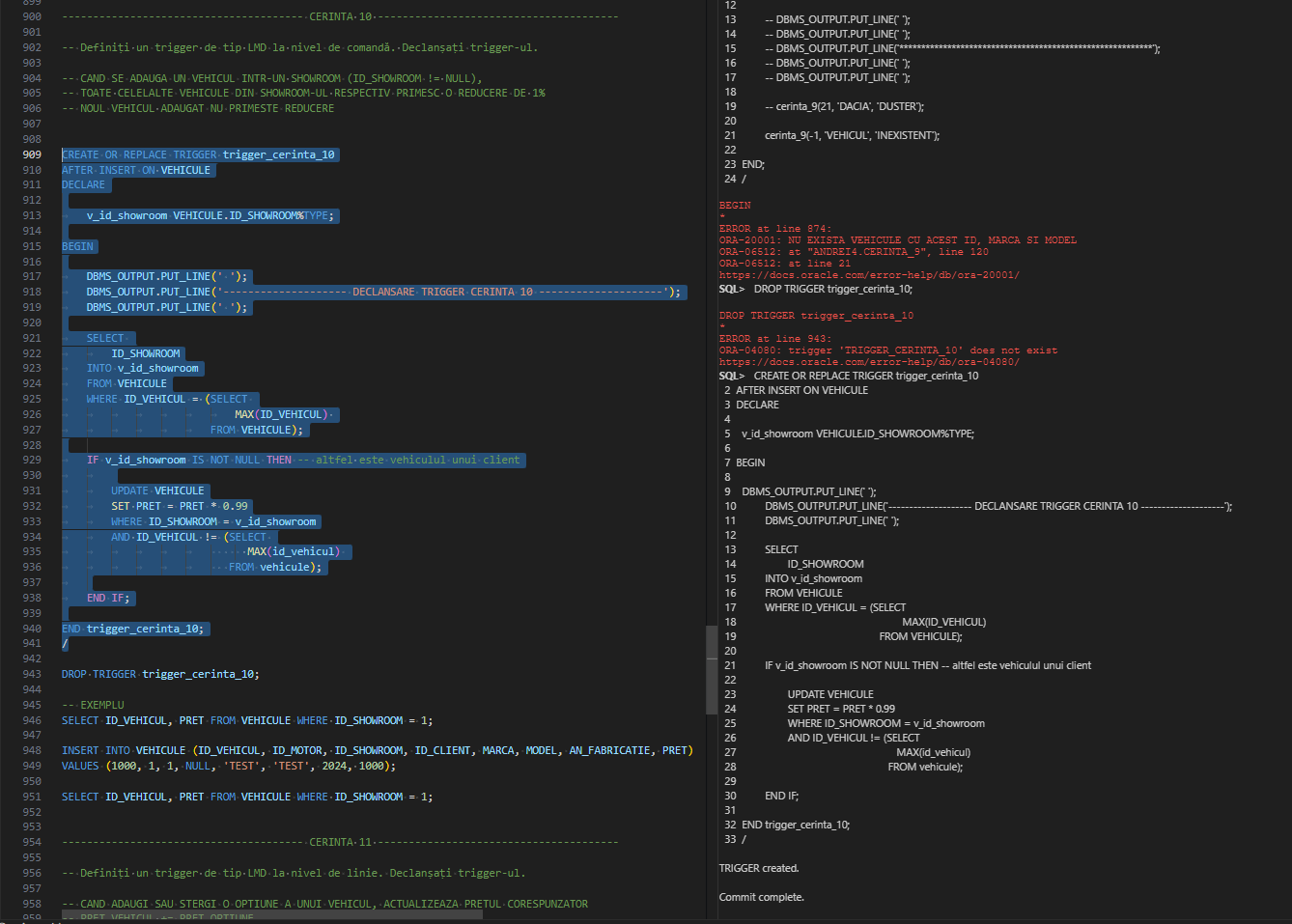
INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (1000, 1, 1, NULL, 'TEST', 'TEST', 2024, 1000);

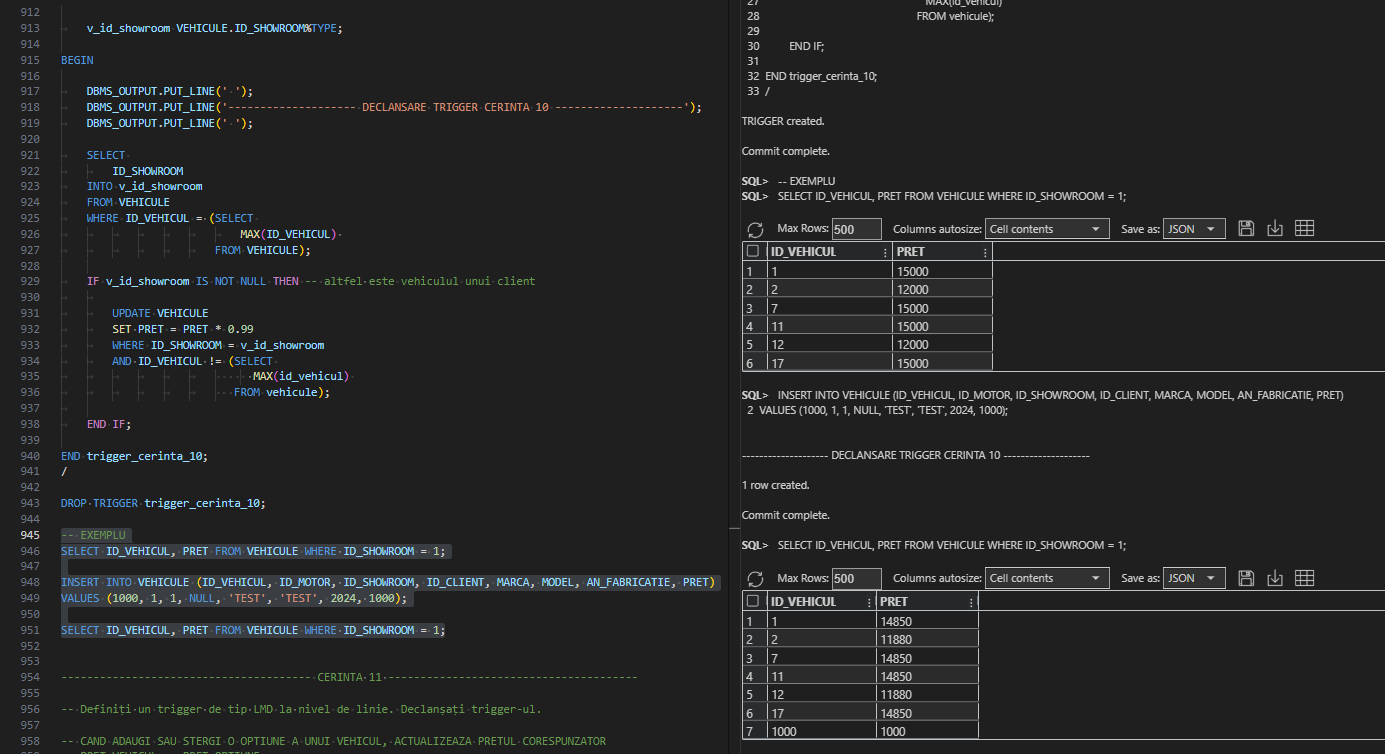
SELECT ID\_VEHICUL, PRET FROM VEHICULE WHERE ID\_SHOWROOM = 1;

**Rezultat rulare:**

Poza 1/2:



Poza 2/2:



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

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

Când adaugi sau ștergi o opțiune a unui vehicul, actualizează prețul corespunzător. Prețul vehiculului se modifică astfel: PRET\_VEHICUL += (-=) PRET\_OPTIUNE.

**Cod PL/SQL:**

CREATE OR REPLACE TRIGGER trigger\_cerinta\_11

AFTER INSERT OR DELETE ON vehicule\_optiuni

FOR EACH ROW

DECLARE

    v\_pret\_optiune OPTIUNI.PRET%TYPE;

BEGIN

    DBMS\_OUTPUT.PUT\_LINE(' ');

    DBMS\_OUTPUT.PUT\_LINE('-------------------- DECLANSARE TRIGGER CERINTA 11 -----------');

    DBMS\_OUTPUT.PUT\_LINE(' ');

    IF INSERTING THEN

        SELECT

            PRET

        INTO v\_pret\_optiune

        FROM OPTIUNI

        WHERE ID\_OPTIUNE = :NEW.ID\_OPTIUNE;

        UPDATE VEHICULE

        SET PRET = PRET + v\_pret\_optiune

        WHERE ID\_VEHICUL = :NEW.ID\_VEHICUL;

    ELSIF DELETING THEN

        SELECT

            PRET

        INTO v\_pret\_optiune

        FROM OPTIUNI

        WHERE ID\_OPTIUNE = :OLD.ID\_OPTIUNE;

        UPDATE vehicule

        SET pret = pret - v\_pret\_optiune

        WHERE id\_vehicul = :OLD.id\_vehicul;

    END IF;

END trigger\_cerinta\_11;

/

DROP TRIGGER trigger\_cerinta\_11;

-- REINITIALIZARE EXEMPLE

DELETE FROM VEHICULE\_OPTIUNI WHERE ID\_VEHICUL = 1000 AND ID\_OPTIUNE = 100;

DELETE FROM VEHICULE WHERE ID\_VEHICUL = 1000;

DELETE FROM OPTIUNI WHERE ID\_OPTIUNE = 100;

-- ADAUGARE OPTIUNE LA UN VEHICUL

INSERT INTO VEHICULE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)

VALUES (1000, 1, 1, NULL, 'TEST', 'TEST', 2024, 1000);

SELECT ID\_VEHICUL, PRET FROM VEHICULE WHERE ID\_VEHICUL = 1000;

INSERT INTO OPTIUNI (ID\_OPTIUNE, NUME, PRET) VALUES (100, 'OPTIUNE TEST', 100);

SELECT \* FROM OPTIUNI WHERE ID\_OPTIUNE = 100;

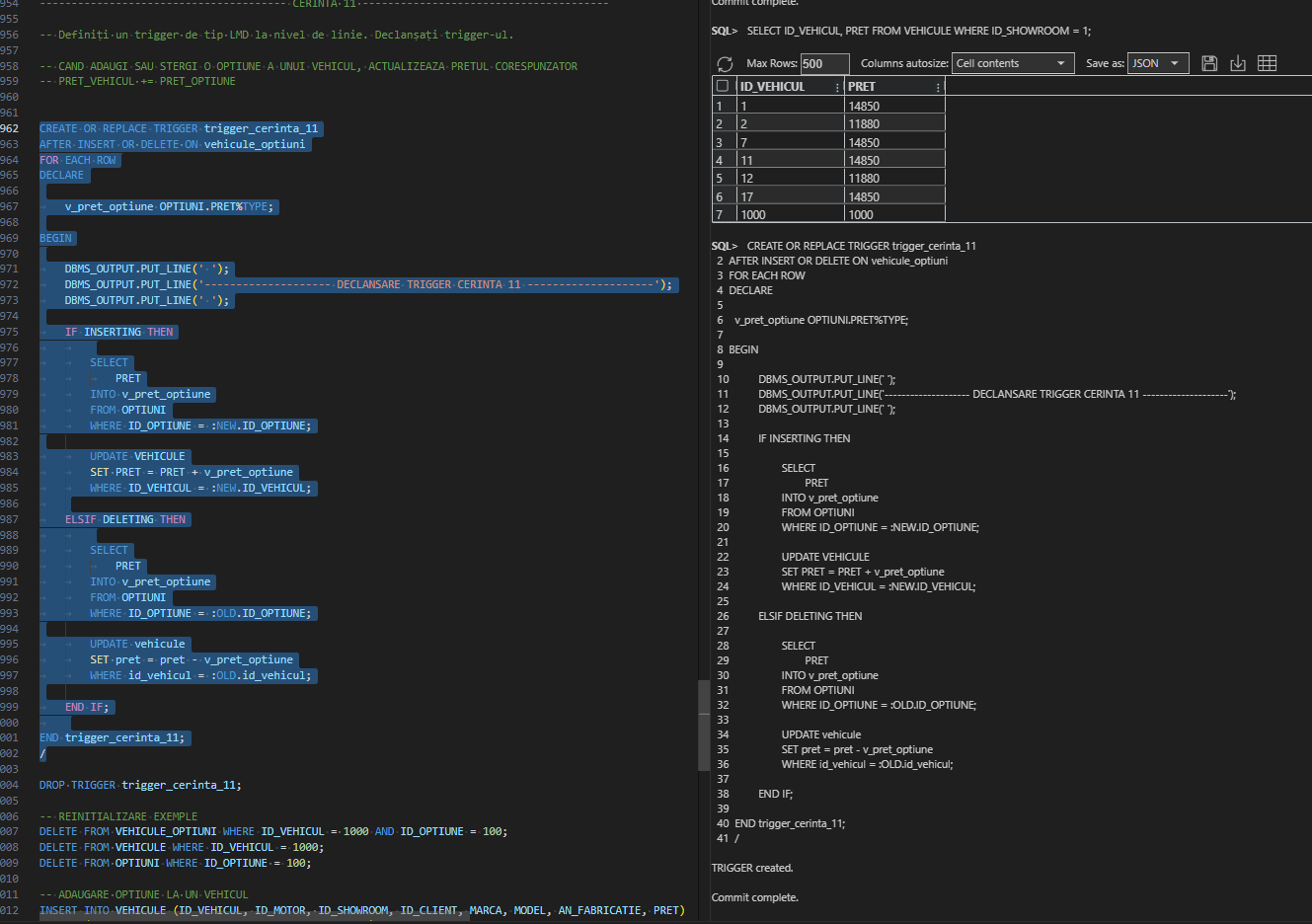
INSERT INTO VEHICULE\_OPTIUNI (ID\_VEHICUL, ID\_OPTIUNE) VALUES (1000, 100);

SELECT \* FROM VEHICULE\_OPTIUNI WHERE ID\_VEHICUL = 1000 AND ID\_OPTIUNE = 100;

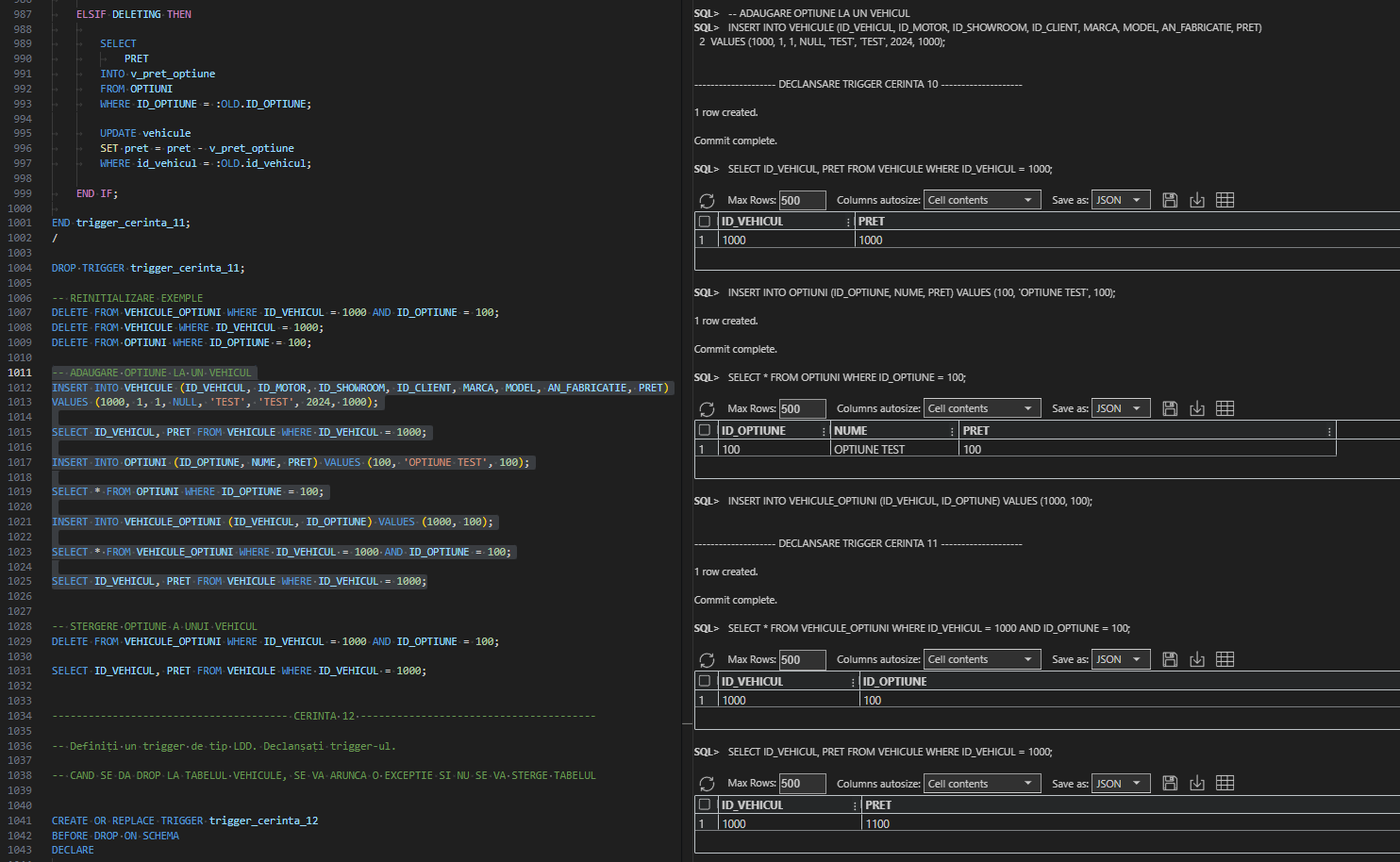
SELECT ID\_VEHICUL, PRET FROM VEHICULE WHERE ID\_VEHICUL = 1000;

**Rezultat rulare:**

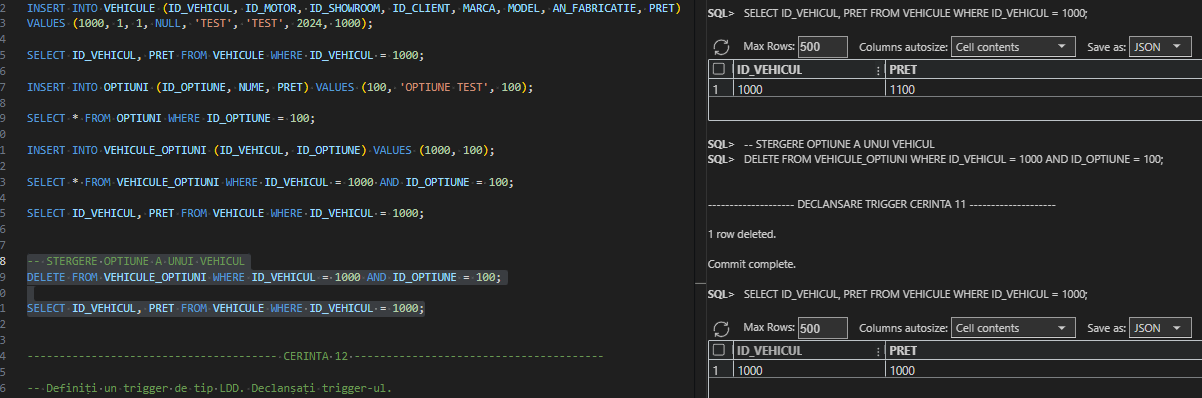
Poza 1/3:



Poza 2/3 (adăugare opțiune la un vehicul):



Poza 3/3 (ștergere opțiune a unui vehicul):



# **12. Cerința 12 - Definiți un trigger de tip LDD**

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

Când se dă DROP la tabelul VEHICULE, se va arunca o excepție și tabelul nu se va șterge.

**Cod PL/SQL:**

CREATE OR REPLACE TRIGGER trigger\_cerinta\_12

BEFORE DROP ON SCHEMA

DECLARE

    e\_stergere\_tabel\_vehicule EXCEPTION;

BEGIN

    IF UPPER(DICTIONARY\_OBJ\_NAME) = 'VEHICULE' THEN

        RAISE e\_stergere\_tabel\_vehicule;

    END IF;

EXCEPTION

    WHEN e\_stergere\_tabel\_vehicule THEN

        RAISE\_APPLICATION\_ERROR(-20999, 'TABELUL <<VEHICULE>> ESTE UN TABEL DEFINITIV');

END trigger\_cerinta\_12;

/

DROP TRIGGER trigger\_cerinta\_12;

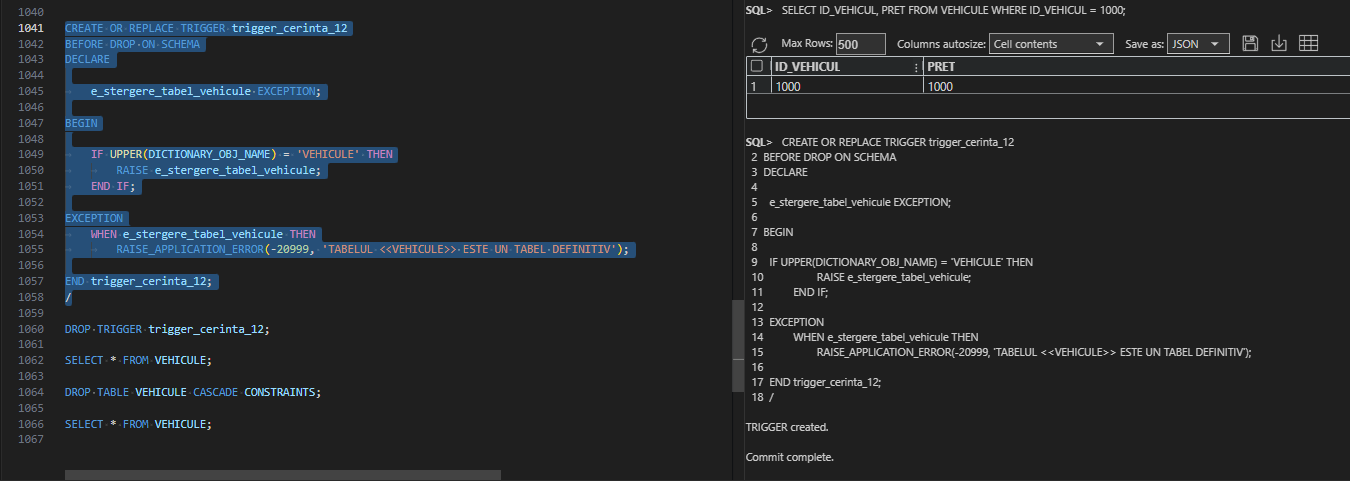
SELECT \* FROM VEHICULE;

DROP TABLE VEHICULE CASCADE CONSTRAINTS;

SELECT \* FROM VEHICULE;

**Rezultat rulare:**

Poza 1/2:



Poza 2/2:

