



# SISTEM DE GESTIUNE A UNEI BAZE DE DATE

DEALER AUTO

OPRAN ANDREI

GRUPA 242  
2024 - 2025

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

UNIVERSITATEA DIN BUCUREȘTI

## CUPRINS

<b>1. Introducere .....</b>	<b>2</b>
Tema aleasă.....	2
Infrastructura utilizată pentru implementare .....	2
<b>2. Cerința 2 - Diagrama entitate-relație .....</b>	<b>3</b>
<b>3. Cerința 3 - Diagrama conceptuală.....</b>	<b>4</b>
<b>4. Cerința 4 - Implementați în Oracle diagrama conceptuală realizată .....</b>	<b>5</b>
<b>5. Cerința 5 - Adăugați informații coerente în tabelele create .....</b>	<b>10</b>
<b>6. Cerința 6 - Formulați în limbaj natural o problemă (3 tipuri de colecții).....</b>	<b>19</b>
<b>7. Cerința 7 - Formulați în limbaj natural o problemă (2 tipuri diferite de cursoare) .....</b>	<b>24</b>
<b>8. Cerința 8 - Formulați în limbaj natural o problemă (să utilizeze într-o singură comandă SQL 3 dintre tabele).....</b>	<b>28</b>
<b>9. Cerința 9 - Formulați în limbaj natural o problemă (să utilizeze într-o singură comandă SQL 5 dintre tabele).....</b>	<b>32</b>
<b>10. Cerința 10 - Definiți un trigger de tip LMD la nivel de comandă .....</b>	<b>40</b>
<b>11. Cerința 11 - Definiți un trigger de tip LMD la nivel de linie.....</b>	<b>43</b>
<b>12. Cerința 12 - Definiți un trigger de tip LDD .....</b>	<b>47</b>

# 1. Introducere

## Tema aleasă

Proiectul consta într-o baza de date destinată unui Dealer auto, respectiv un sistem de gestiune a acesteia, prin intermediul căreia se pot administra diverse informații utile pentru o bună 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 virtuală)
- IDE: Visual Studio Code cu extensia "Oracle Developer Tools for VS Code (SQL and PLSQL)"
- **Observație:** Pentru ca am folosit [Visual Studio Code](#), în print-screen-urile cu rularea exercițiilor „Script Output” și „Dbms Output” se afla în aceeași fereastră, fata de [SQL Developer](#) unde există o fereastră separată pentru amândouă. Exemplu:

The screenshot displays a PL/SQL script in the left pane and its execution output in the right pane. The script, named 'cerinta\_7', is a procedure that iterates through vehicles sold in 2024, printing details like vehicle ID, brand, model, price, and employee information. It also calculates a total sum of all vehicle prices for 2024. The output on the right shows the results of these iterations, listing three vehicles with their respective details and a final total sum of 45000. The script concludes with a 'PL/SQL procedure successfully completed.' message and a 'Commit complete.' status.

```
35 JOIN_ANGAJATI A ON F.ID_ANGAJAT = A.ID_ANGAJAT
36 WHERE EXTRACT(YEAR FROM DATA_FACTURA) = 2024) LOOP
37
38 FOR vehicul_client in c_vehicule_clienti(angajat_factura_2024.ID_VEHICUL) LOOP
39
40     v_cont := v_cont + 1;
41     v_suma := v_suma + vehicul_client.PRET;
42     DBMS_OUTPUT.PUT_LINE('-----' || v_cont || '-----');
43     DBMS_OUTPUT.PUT_LINE('ID_VEHICUL: ' || vehicul_client.ID_VEHICUL);
44     DBMS_OUTPUT.PUT_LINE('MARCA: ' || vehicul_client.MARCA);
45     DBMS_OUTPUT.PUT_LINE('MODEL: ' || vehicul_client.MODEL);
46     DBMS_OUTPUT.PUT_LINE('PRET: ' || vehicul_client.PRET);
47     DBMS_OUTPUT.PUT_LINE('');
48     DBMS_OUTPUT.PUT_LINE('DATE_ANGAJAT:');
49     DBMS_OUTPUT.PUT_LINE('ID_ANGAJAT: ' || angajat_factura_2024.ID_ANGAJAT);
50     DBMS_OUTPUT.PUT_LINE('NUME_ANGAJAT: ' || angajat_factura_2024.NUME);
51
52 END LOOP;
53
54 END LOOP;
55 DBMS_OUTPUT.PUT_LINE('-----');
56 DBMS_OUTPUT.PUT_LINE('SUMA TOTALA A VEHICULELOR VANDUTE IN 2024: ' || v_suma);
57
58 EXCEPTION
59 WHEN e_nu_exista_vehicule THEN
60     RAISE_APPLICATION_ERROR(-20001, 'NU EXISTA VEHICULE VANDUTE IN ANUL 2024');
61 WHEN OTHERS THEN
62     RAISE_APPLICATION_ERROR(-20000, 'EROARE: ' || SQLERRM);
63 END cerinta_7;
64 /
65
66 BEGIN
67     cerinta_7;
68 END;
69 /
```

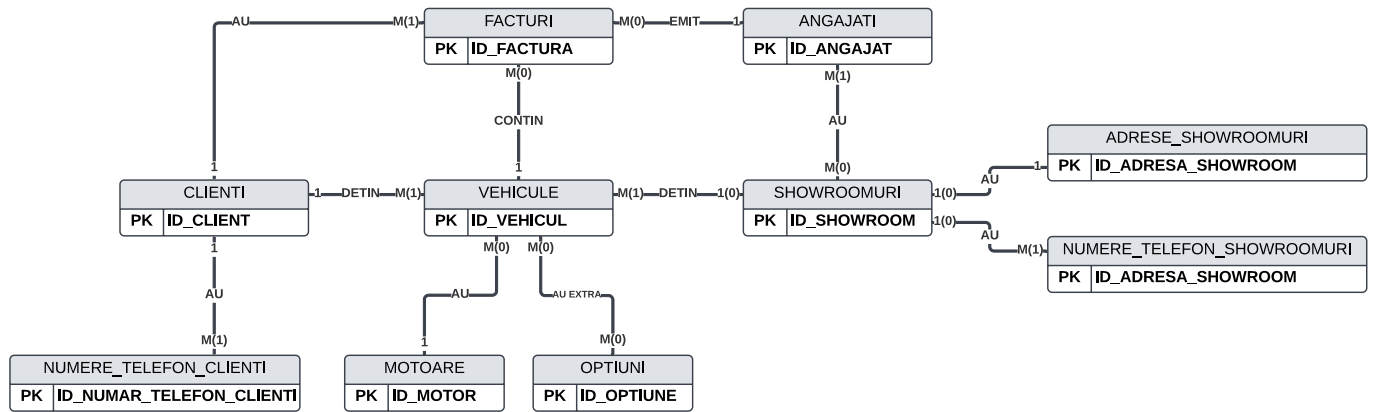
SQL> BEGIN  
2 cerinta\_7;  
3 END;  
4 /

VEHICULE VANDUTE IN ANUL 2024:

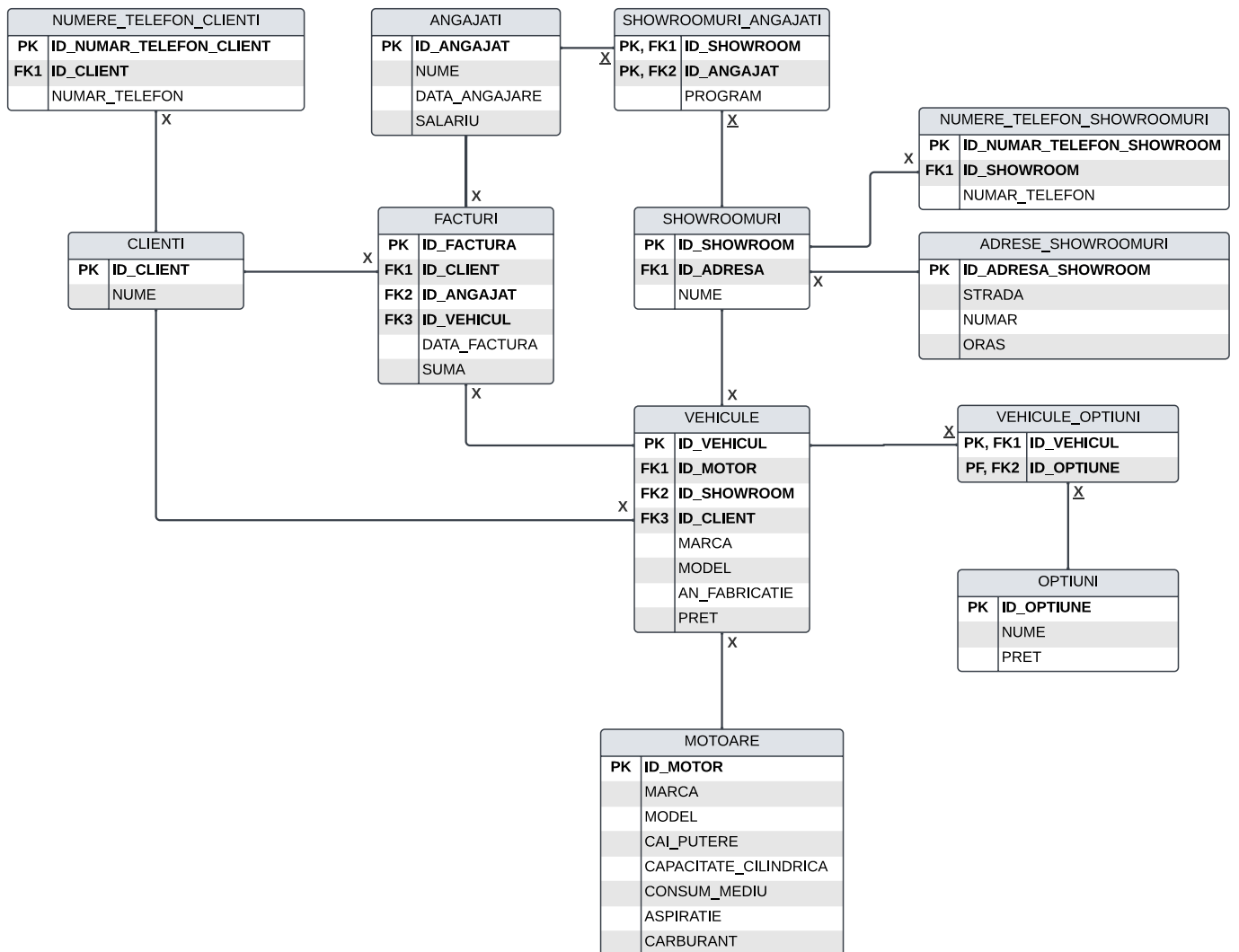
-----1-----  
ID\_VEHICUL: 26  
MARCA: Dacia  
MODEL: Spring  
PRET: 16000  
  
DATE\_ANGAJAT:  
ID\_ANGAJAT: 6  
NUME\_ANGAJAT: Stoian Cristina  
  
-----2-----  
ID\_VEHICUL: 27  
MARCA: Dacia  
MODEL: Duster  
PRET: 15000  
  
DATE\_ANGAJAT:  
ID\_ANGAJAT: 7  
NUME\_ANGAJAT: Petre Mihai  
  
-----3-----  
ID\_VEHICUL: 30  
MARCA: Dacia  
MODEL: Lodgy  
PRET: 14000  
  
DATE\_ANGAJAT:  
ID\_ANGAJAT: 1  
NUME\_ANGAJAT: Vasile Ion  
  
SUMA TOTALA A VEHICULELOR VANDUTE IN 2024: 45000

PL/SQL procedure successfully completed.  
Commit complete.

## 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:

```

C:\Users> andre > Desktop > Proiect-SGBD > Cerintele 4-12.sql > TABLE VEHICULE_OPTIUNI
56 TRUNCATE TABLE CLIENTI;
57
58
59 CREATE TABLE CLIENTI (
60     ID_CLIENT NUMBER(6) PRIMARY KEY,
61     NUME VARCHAR2(256)
62 );
63
64 CREATE TABLE ADRESE_SHOWROOMURI (
65     ID_ADRESA_SHOWROOM NUMBER(6) PRIMARY KEY,
66     STRADA VARCHAR2(256),
67     NUMAR NUMBER(6),
68     ORAS VARCHAR2(256)
69 );
70
71 CREATE TABLE MOTOARE (
72     ID_MOTOR NUMBER(6) PRIMARY KEY,
73     MARCA VARCHAR2(256),
74     MODEL VARCHAR2(256),
75     CAL_PUTERE NUMBER(6),
76     CAPACITATE_CILINDRICA NUMBER(6),
77     CONSUM_MEDIU NUMBER(6),
78     ASPIRATIE VARCHAR2(256),
79     CARBURANT VARCHAR2(256)
80 );
81
82 CREATE TABLE OPTIUNI (
83     ID_OPTIUNE NUMBER(6) PRIMARY KEY,
84     NUME VARCHAR2(256),
85     PRET NUMBER(6)
86 );
87
88 CREATE TABLE SHOWROOMURI (
89     ID_SHOWROOM NUMBER(6) PRIMARY KEY,
90     ID_ADRESA NUMBER(6),
91     NUME VARCHAR2(256),
92     FOREIGN KEY (ID_ADRESA) REFERENCES ADRESE_SHOWROOMURI (ID_ADRESA_SHOWROOM)
93 );
94
95 CREATE TABLE ANGAJATI (
96     ID_ANGAJAT NUMBER(6) PRIMARY KEY,
97     NUME VARCHAR2(256),
98     DATA_ANGAJARE DATE,
99     SALARIU NUMBER(6)
100 );
101
102 CREATE TABLE VEHICULE (
103     ID_VEHICUL NUMBER(6) PRIMARY KEY,
104     ID_MOTOR NUMBER(6),
105     ID_SHOWROOM NUMBER(6),
106     ID_CLIENT NUMBER(6),
107     MARCA VARCHAR2(256),
108     MODEL VARCHAR2(256),
109     AN_FABRICATIE NUMBER(6),
110     PRET NUMBER(6),
111     FOREIGN KEY (ID_MOTOR) REFERENCES MOTOARE (ID_MOTOR),
112     FOREIGN KEY (ID_SHOWROOM) REFERENCES SHOWROOMURI (ID_SHOWROOM),
113     FOREIGN KEY (ID_CLIENT) REFERENCES CLIENTI (ID_CLIENT),
114     CONSTRAINT CHECK_ID_CLIENT_SHOWROOM CHECK (
115         (ID_CLIENT IS NULL AND ID_SHOWROOM IS NOT NULL)
116         OR (ID_CLIENT IS NOT NULL AND ID_SHOWROOM IS NULL)
117         OR (ID_CLIENT IS NULL AND ID_SHOWROOM IS NULL)
118     )
119 );
120
121 CREATE TABLE FACTURI (
122     ID_FACTURA NUMBER(6) PRIMARY KEY,
123     ID_CLIENT NUMBER(6) NOT NULL
124 );
125
Results: Cerintele 4-12.sql
SQL> CREATE TABLE CLIENTI (
2 ID_CLIENT NUMBER(6) PRIMARY KEY,
3 NUME VARCHAR2(256)
4 );
TABLE created.
SQL> CREATE TABLE ADRESE_SHOWROOMURI (
2 ID_ADRESA_SHOWROOM NUMBER(6) PRIMARY KEY,
3 STRADA VARCHAR2(256),
4 NUMAR NUMBER(6),
5 ORAS VARCHAR2(256)
6 );
TABLE created.
SQL> CREATE TABLE MOTOARE (
2 ID_MOTOR NUMBER(6) PRIMARY KEY,
3 MARCA VARCHAR2(256),
4 MODEL VARCHAR2(256),
5 CAL_PUTERE NUMBER(6),
6 CAPACITATE_CILINDRICA NUMBER(6),
7 CONSUM_MEDIU NUMBER(6),
8 ASPIRATIE VARCHAR2(256),
9 CARBURANT VARCHAR2(256)
10 );
TABLE created.
SQL> CREATE TABLE OPTIUNI (
2 ID_OPTIUNE NUMBER(6) PRIMARY KEY,
3 NUME VARCHAR2(256),
4 PRET NUMBER(6)
5 );
TABLE created.
SQL> CREATE TABLE SHOWROOMURI (
2 ID_SHOWROOM NUMBER(6) PRIMARY KEY,
3 ID_ADRESA NUMBER(6),
4 NUME VARCHAR2(256),
5 FOREIGN KEY (ID_ADRESA) REFERENCES ADRESE_SHOWROOMURI (ID_ADRESA_SHOWROOM)
6 );
TABLE created.
SQL> CREATE TABLE ANGAJATI (
2 ID_ANGAJAT NUMBER(6) PRIMARY KEY,
3 NUME VARCHAR2(256),
4 DATA_ANGAJARE DATE,
5 SALARIU NUMBER(6)
6 );
TABLE created.
SQL> CREATE TABLE VEHICULE (
2 ID_VEHICUL NUMBER(6) PRIMARY KEY,
3 ID_MOTOR NUMBER(6),
4 ID_SHOWROOM NUMBER(6),
5 ID_CLIENT NUMBER(6),
6 MARCA VARCHAR2(256),
7 MODEL VARCHAR2(256),
8 AN_FABRICATIE NUMBER(6),
9 PRET NUMBER(6),
10 FOREIGN KEY (ID_MOTOR) REFERENCES MOTOARE (ID_MOTOR),
11 FOREIGN KEY (ID_SHOWROOM) REFERENCES SHOWROOMURI (ID_SHOWROOM),
12 FOREIGN KEY (ID_CLIENT) REFERENCES CLIENTI (ID_CLIENT),
13 CONSTRAINT CHECK_ID_CLIENT_SHOWROOM CHECK (
14 (ID_CLIENT IS NULL AND ID_SHOWROOM IS NOT NULL)
15 OR (ID_CLIENT IS NOT NULL AND ID_SHOWROOM IS NULL)
16 OR (ID_CLIENT IS NULL AND ID_SHOWROOM IS NULL)
17 )
18 );
TABLE created.
```

Poza 2/2:

```

C: > Users > andre > Desktop > Proiect-SGBD > Cerintele 4-12.sql > TABLE VEHICULE_OPTIUNI

102 CREATE TABLE VEHICULE (
110     PRET NUMBER(6),
111     FOREIGN KEY (ID_MOTOR) REFERENCES MOTOARE(ID_MOTOR),
112     FOREIGN KEY (ID_SHOWROOM) REFERENCES SHOWROOMURI(ID_SHOWROOM),
113     FOREIGN KEY (ID_CLIENT) REFERENCES CLIENTI(ID_CLIENT),
114     CONSTRAINT CHECK_ID_CLIENT_SHOWROOM CHECK (
115         (ID_CLIENT IS NULL AND ID_SHOWROOM IS NOT NULL)
116         OR (ID_CLIENT IS NOT NULL AND ID_SHOWROOM IS NULL)
117         OR (ID_CLIENT IS NULL AND ID_SHOWROOM IS NULL)
118     );
119 );
120
121 CREATE TABLE FACTURI (
122     ID_FACTURA NUMBER(6) PRIMARY KEY,
123     ID_CLIENT NUMBER(6) NOT NULL,
124     ID_ANGAJAT NUMBER(6) NOT NULL,
125     ID_VEHICUL NUMBER(6) NOT NULL,
126     DATA_FACTURA DATE,
127     SUMA NUMBER(6),
128     FOREIGN KEY (ID_CLIENT) REFERENCES CLIENTI(ID_CLIENT),
129     FOREIGN KEY (ID_ANGAJAT) REFERENCES ANGAJATI(ID_ANGAJAT),
130     FOREIGN KEY (ID_VEHICUL) REFERENCES VEHICULE(ID_VEHICUL)
131 );
132
133 CREATE TABLE NUMERE_TELEFON_SHOWROOMURI (
134     ID_NUMAR_TELEFON_SHOWROOM NUMBER(6) PRIMARY KEY,
135     ID_SHOWROOM NUMBER(6),
136     NUMAR_TELEFON VARCHAR2(256),
137     CONSTRAINT CHECK_NUMAR_TELEFON_SHOWROOM CHECK
138     (
139         REGEXP_LIKE(NUMAR_TELEFON, '^07[0-9]{2}[0-9]{3}[0-9]{3}$')
140     ),
141     FOREIGN KEY (ID_SHOWROOM) REFERENCES SHOWROOMURI(ID_SHOWROOM)
142 );
143
144 CREATE TABLE NUMERE_TELEFON_CLIENTI (
145     ID_NUMAR_TELEFON_CLIENT NUMBER(6) PRIMARY KEY,
146     ID_CLIENT NUMBER(6),
147     NUMAR_TELEFON VARCHAR2(256),
148     CONSTRAINT CHECK_NUMAR_TELEFON_CLIENT CHECK
149     (
150         REGEXP_LIKE(NUMAR_TELEFON, '^07[0-9]{2}[0-9]{3}[0-9]{3}$')
151     ),
152     FOREIGN KEY (ID_CLIENT) REFERENCES CLIENTI(ID_CLIENT)
153 );
154
155 CREATE TABLE SHOWROOMURI_ANGAJATI (
156     ID_SHOWROOM NUMBER(6),
157     ID_ANGAJAT NUMBER(6),
158     PROGRAM VARCHAR2(256),
159     PRIMARY KEY (ID_SHOWROOM, ID_ANGAJAT),
160     FOREIGN KEY (ID_SHOWROOM) REFERENCES SHOWROOMURI(ID_SHOWROOM),
161     FOREIGN KEY (ID_ANGAJAT) REFERENCES ANGAJATI(ID_ANGAJAT)
162 );
163
164 CREATE TABLE VEHICULE_OPTIUNI (
165     ID_VEHICUL NUMBER(6),
166     ID_OPTIUNE NUMBER(6),
167     PRIMARY KEY (ID_VEHICUL, ID_OPTIUNE),
168     FOREIGN KEY (ID_VEHICUL) REFERENCES VEHICULE(ID_VEHICUL),
169     FOREIGN KEY (ID_OPTIUNE) REFERENCES OPTIUNI(ID_OPTIUNE)
170 );
171
172 ----- CERINTA 5 - POPULARE TABELE -----
173
174 INSERT INTO CLIENTI (ID_CLIENT, NUME) VALUES (SECVENTA_CLIENTI.NEXTVAL, 'Ion Popescu');
175 INSERT INTO CLIENTI (ID_CLIENT, NUME) VALUES (SECVENTA_CLIENTI.NEXTVAL, 'Maria Ionescu');
```

```

Results: Cerintele 4-12.sql X
7 MODEL VARCHAR2(256),
8 AN_FABRICATIE NUMBER(6),
9 PRET NUMBER(6),
10 FOREIGN KEY (ID_MOTOR) REFERENCES MOTOARE(ID_MOTOR),
11 FOREIGN KEY (ID_SHOWROOM) REFERENCES SHOWROOMURI(ID_SHOWROOM),
12 FOREIGN KEY (ID_CLIENT) REFERENCES CLIENTI(ID_CLIENT),
13 CONSTRAINT CHECK_ID_CLIENT_SHOWROOM CHECK (
14     (ID_CLIENT IS NULL AND ID_SHOWROOM IS NOT NULL)
15     OR (ID_CLIENT IS NOT NULL AND ID_SHOWROOM IS NULL)
16     OR (ID_CLIENT IS NULL AND ID_SHOWROOM IS NULL)
17 );
18 ;

TABLE created.

SQL> CREATE TABLE FACTURI (
2 ID_FACTURA NUMBER(6) PRIMARY KEY,
3 ID_CLIENT NUMBER(6) NOT NULL,
4 ID_ANGAJAT NUMBER(6) NOT NULL,
5 ID_VEHICUL NUMBER(6) NOT NULL,
6 DATA_FACTURA DATE,
7 SUMA NUMBER(6),
8 FOREIGN KEY (ID_CLIENT) REFERENCES CLIENTI(ID_CLIENT),
9 FOREIGN KEY (ID_ANGAJAT) REFERENCES ANGAJATI(ID_ANGAJAT),
10 FOREIGN KEY (ID_VEHICUL) REFERENCES VEHICULE(ID_VEHICUL)
11 );

TABLE created.

SQL> CREATE TABLE NUMERE_TELEFON_SHOWROOMURI (
2 ID_NUMAR_TELEFON_SHOWROOM NUMBER(6) PRIMARY KEY,
3 ID_SHOWROOM NUMBER(6),
4 NUMAR_TELEFON VARCHAR2(256),
5 CONSTRAINT CHECK_NUMAR_TELEFON_SHOWROOM CHECK
6 (
7     REGEXP_LIKE(NUMAR_TELEFON, '^07[0-9]{2}[0-9]{3}[0-9]{3}$')
8 ),
9 FOREIGN KEY (ID_SHOWROOM) REFERENCES SHOWROOMURI(ID_SHOWROOM)
10 );

TABLE created.

SQL> CREATE TABLE NUMERE_TELEFON_CLIENTI (
2 ID_NUMAR_TELEFON_CLIENT NUMBER(6) PRIMARY KEY,
3 ID_CLIENT NUMBER(6),
4 NUMAR_TELEFON VARCHAR2(256),
5 CONSTRAINT CHECK_NUMAR_TELEFON_CLIENT CHECK
6 (
7     REGEXP_LIKE(NUMAR_TELEFON, '^07[0-9]{2}[0-9]{3}[0-9]{3}$')
8 ),
9 FOREIGN KEY (ID_CLIENT) REFERENCES CLIENTI(ID_CLIENT)
10 );

TABLE created.

SQL> CREATE TABLE SHOWROOMURI_ANGAJATI (
2 ID_SHOWROOM NUMBER(6),
3 ID_ANGAJAT NUMBER(6),
4 PROGRAM VARCHAR2(256),
5 PRIMARY KEY (ID_SHOWROOM, ID_ANGAJAT),
6 FOREIGN KEY (ID_SHOWROOM) REFERENCES SHOWROOMURI(ID_SHOWROOM),
7 FOREIGN KEY (ID_ANGAJAT) REFERENCES ANGAJATI(ID_ANGAJAT)
8 );

TABLE created.

SQL> CREATE TABLE VEHICULE_OPTIUNI (
2 ID_VEHICUL NUMBER(6),
3 ID_OPTIUNE NUMBER(6),
4 PRIMARY KEY (ID_VEHICUL, ID_OPTIUNE),
5 FOREIGN KEY (ID_VEHICUL) REFERENCES VEHICULE(ID_VEHICUL),
6 FOREIGN KEY (ID_OPTIUNE) REFERENCES OPTIUNI(ID_OPTIUNE)
7 );

TABLE created.
```

## 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);

```

## -- VEHICLE SHOWROOM

[illegible]

```

-- 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_TEFON_SHOWROOMURI (ID_NUMAR_TEFON_SHOWROOM, ID_SHOWROOM, NUMAR_TEFON)
VALUES (SECVENTA_NUMERE_TEFON_SHOWROOM.NEXTVAL, 1, '0725 426 589');
INSERT INTO NUMERE_TEFON_SHOWROOMURI (ID_NUMAR_TEFON_SHOWROOM, ID_SHOWROOM, NUMAR_TEFON)
VALUES (SECVENTA_NUMERE_TEFON_SHOWROOM.NEXTVAL, 1, '0754 113 636');
INSERT INTO NUMERE_TEFON_SHOWROOMURI (ID_NUMAR_TEFON_SHOWROOM, ID_SHOWROOM, NUMAR_TEFON)
VALUES (SECVENTA_NUMERE_TEFON_SHOWROOM.NEXTVAL, 2, '0726 214 577');
INSERT INTO NUMERE_TEFON_SHOWROOMURI (ID_NUMAR_TEFON_SHOWROOM, ID_SHOWROOM, NUMAR_TEFON)
VALUES (SECVENTA_NUMERE_TEFON_SHOWROOM.NEXTVAL, 2, '0732 355 518');
INSERT INTO NUMERE_TEFON_SHOWROOMURI (ID_NUMAR_TEFON_SHOWROOM, ID_SHOWROOM, NUMAR_TEFON)
VALUES (SECVENTA_NUMERE_TEFON_SHOWROOM.NEXTVAL, 3, '0727 446 289');
INSERT INTO NUMERE_TEFON_SHOWROOMURI (ID_NUMAR_TEFON_SHOWROOM, ID_SHOWROOM, NUMAR_TEFON)
VALUES (SECVENTA_NUMERE_TEFON_SHOWROOM.NEXTVAL, 3, '0754 566 491');
INSERT INTO NUMERE_TEFON_SHOWROOMURI (ID_NUMAR_TEFON_SHOWROOM, ID_SHOWROOM, NUMAR_TEFON)
VALUES (SECVENTA_NUMERE_TEFON_SHOWROOM.NEXTVAL, 4, '0723 638 941');
INSERT INTO NUMERE_TEFON_SHOWROOMURI (ID_NUMAR_TEFON_SHOWROOM, ID_SHOWROOM, NUMAR_TEFON)
VALUES (SECVENTA_NUMERE_TEFON_SHOWROOM.NEXTVAL, 4, '0754 749 020');
INSERT INTO NUMERE_TEFON_SHOWROOMURI (ID_NUMAR_TEFON_SHOWROOM, ID_SHOWROOM, NUMAR_TEFON)
VALUES (SECVENTA_NUMERE_TEFON_SHOWROOM.NEXTVAL, 5, '0739 481 395');
INSERT INTO NUMERE_TEFON_SHOWROOMURI (ID_NUMAR_TEFON_SHOWROOM, ID_SHOWROOM, NUMAR_TEFON)
VALUES (SECVENTA_NUMERE_TEFON_SHOWROOM.NEXTVAL, 5, '0704 719 578');

INSERT INTO NUMERE_TEFON_CLIENTI (ID_NUMAR_TEFON_CLIENT, ID_CLIENT, NUMAR_TEFON)
VALUES (SECVENTA_NUMERE_TEFON_CLIENT.NEXTVAL, 1, '0793 765 902');
INSERT INTO NUMERE_TEFON_CLIENTI (ID_NUMAR_TEFON_CLIENT, ID_CLIENT, NUMAR_TEFON)
VALUES (SECVENTA_NUMERE_TEFON_CLIENT.NEXTVAL, 1, '0794 832 384');
INSERT INTO NUMERE_TEFON_CLIENTI (ID_NUMAR_TEFON_CLIENT, ID_CLIENT, NUMAR_TEFON)
VALUES (SECVENTA_NUMERE_TEFON_CLIENT.NEXTVAL, 2, '0748 200 555');
INSERT INTO NUMERE_TEFON_CLIENTI (ID_NUMAR_TEFON_CLIENT, ID_CLIENT, NUMAR_TEFON)
VALUES (SECVENTA_NUMERE_TEFON_CLIENT.NEXTVAL, 2, '0774 195 594');
INSERT INTO NUMERE_TEFON_CLIENTI (ID_NUMAR_TEFON_CLIENT, ID_CLIENT, NUMAR_TEFON)
VALUES (SECVENTA_NUMERE_TEFON_CLIENT.NEXTVAL, 3, '0758 102 333');
INSERT INTO NUMERE_TEFON_CLIENTI (ID_NUMAR_TEFON_CLIENT, ID_CLIENT, NUMAR_TEFON)
VALUES (SECVENTA_NUMERE_TEFON_CLIENT.NEXTVAL, 3, '0759 392 101');
INSERT INTO NUMERE_TEFON_CLIENTI (ID_NUMAR_TEFON_CLIENT, ID_CLIENT, NUMAR_TEFON)
VALUES (SECVENTA_NUMERE_TEFON_CLIENT.NEXTVAL, 4, '0754 393 121');
INSERT INTO NUMERE_TEFON_CLIENTI (ID_NUMAR_TEFON_CLIENT, ID_CLIENT, NUMAR_TEFON)
VALUES (SECVENTA_NUMERE_TEFON_CLIENT.NEXTVAL, 4, '0754 102 303');
INSERT INTO NUMERE_TEFON_CLIENTI (ID_NUMAR_TEFON_CLIENT, ID_CLIENT, NUMAR_TEFON)
VALUES (SECVENTA_NUMERE_TEFON_CLIENT.NEXTVAL, 5, '0749 584 431');
INSERT INTO NUMERE_TEFON_CLIENTI (ID_NUMAR_TEFON_CLIENT, ID_CLIENT, NUMAR_TEFON)
VALUES (SECVENTA_NUMERE_TEFON_CLIENT.NEXTVAL, 5, '0755 492 120');

```



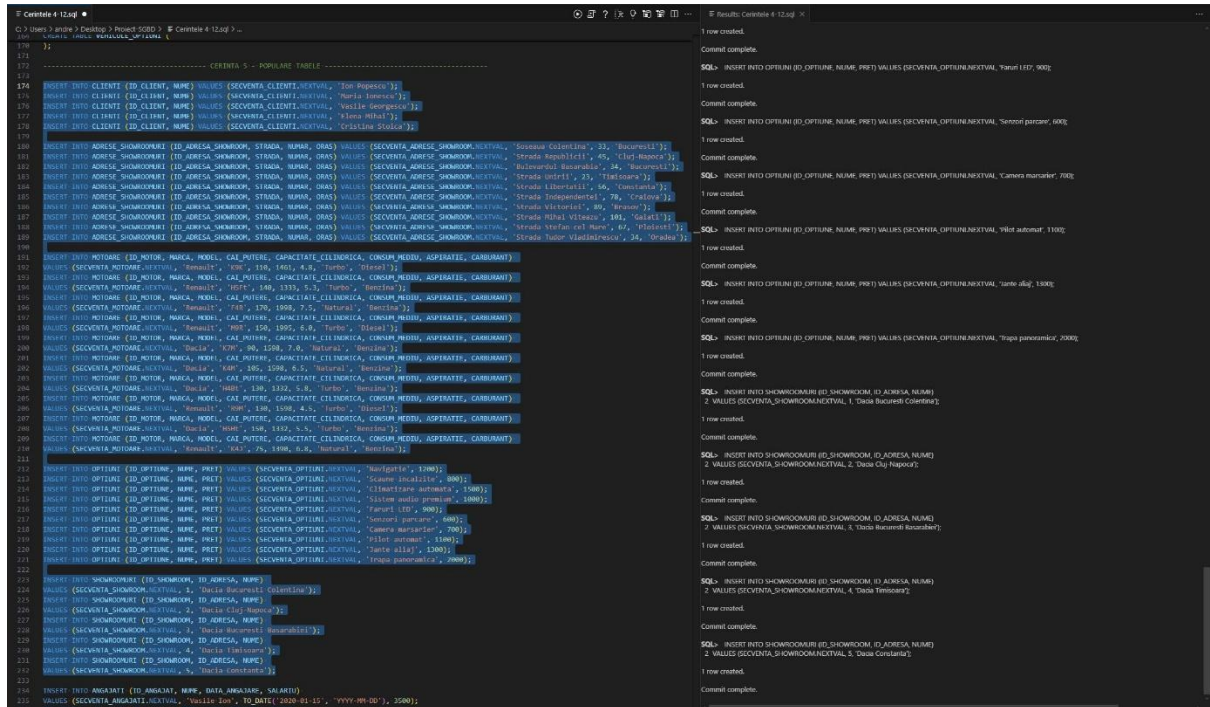
```

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');

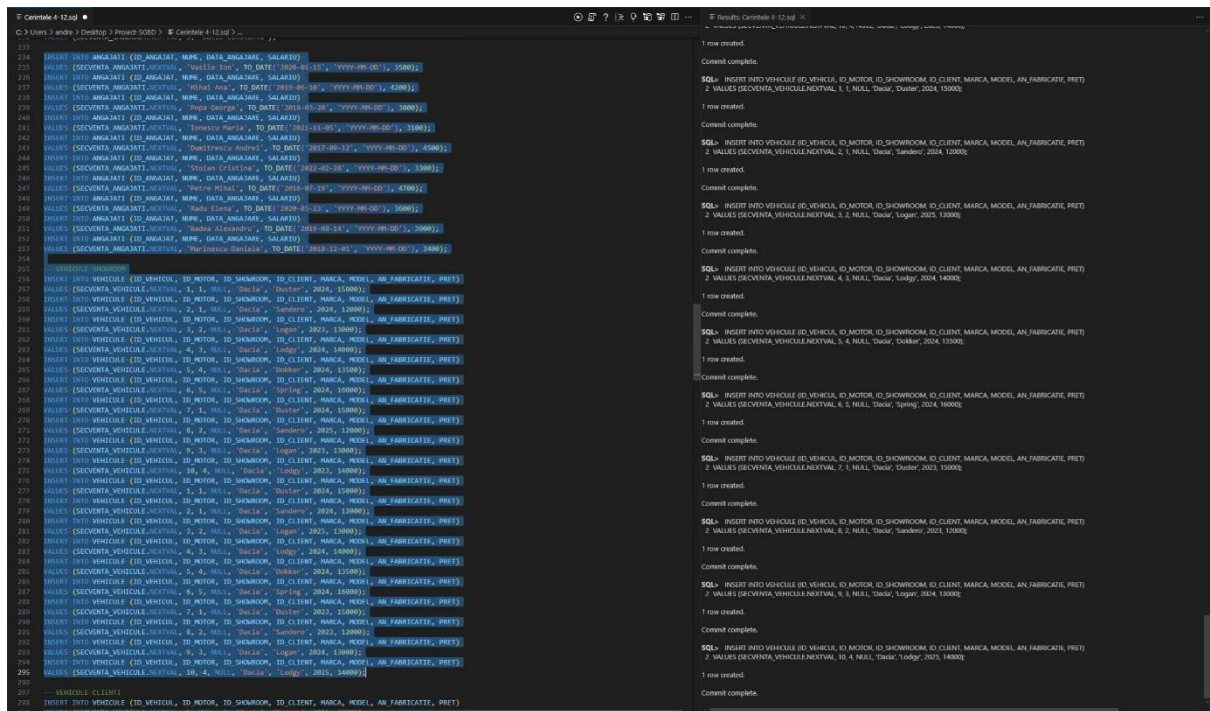
INSERT INTO VEHICULE_OPTIUNI (ID_VEHICUL, ID_OPTIUNE) VALUES (11, 1);
INSERT INTO VEHICULE_OPTIUNI (ID_VEHICUL, ID_OPTIUNE) VALUES (11, 2);
INSERT INTO VEHICULE_OPTIUNI (ID_VEHICUL, ID_OPTIUNE) VALUES (12, 3);
INSERT INTO VEHICULE_OPTIUNI (ID_VEHICUL, ID_OPTIUNE) VALUES (12, 4);
INSERT INTO VEHICULE_OPTIUNI (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);

```

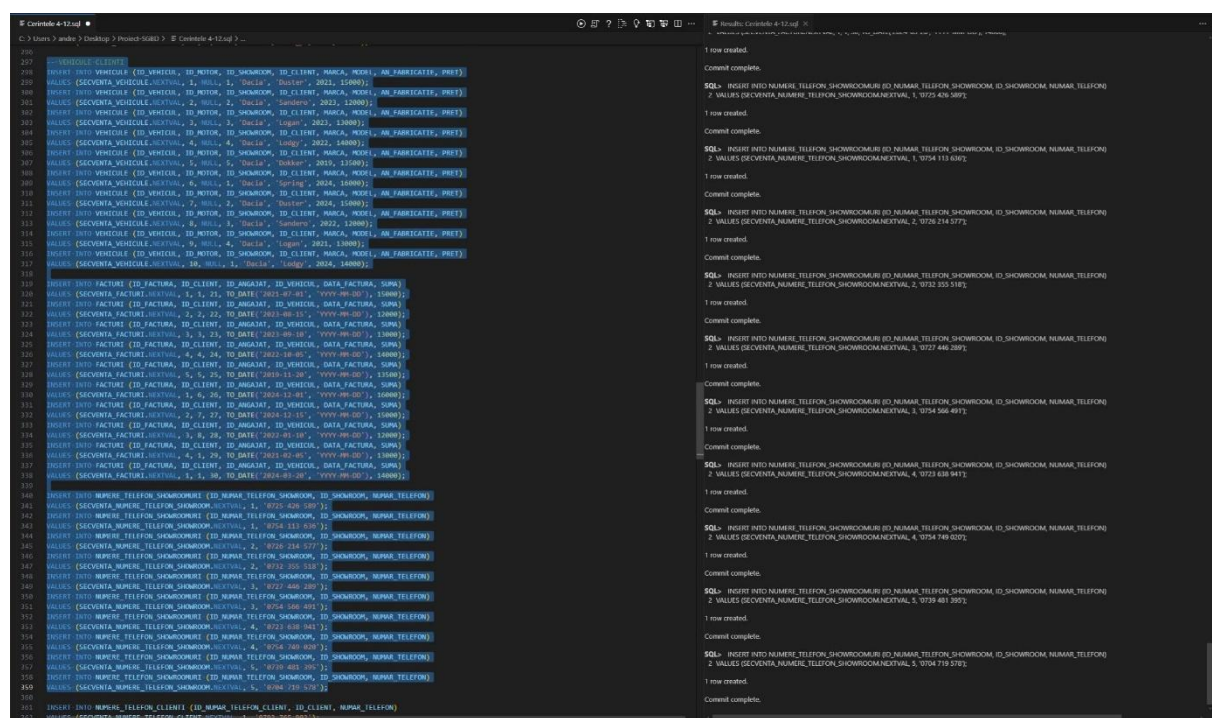
Poza 1/4:



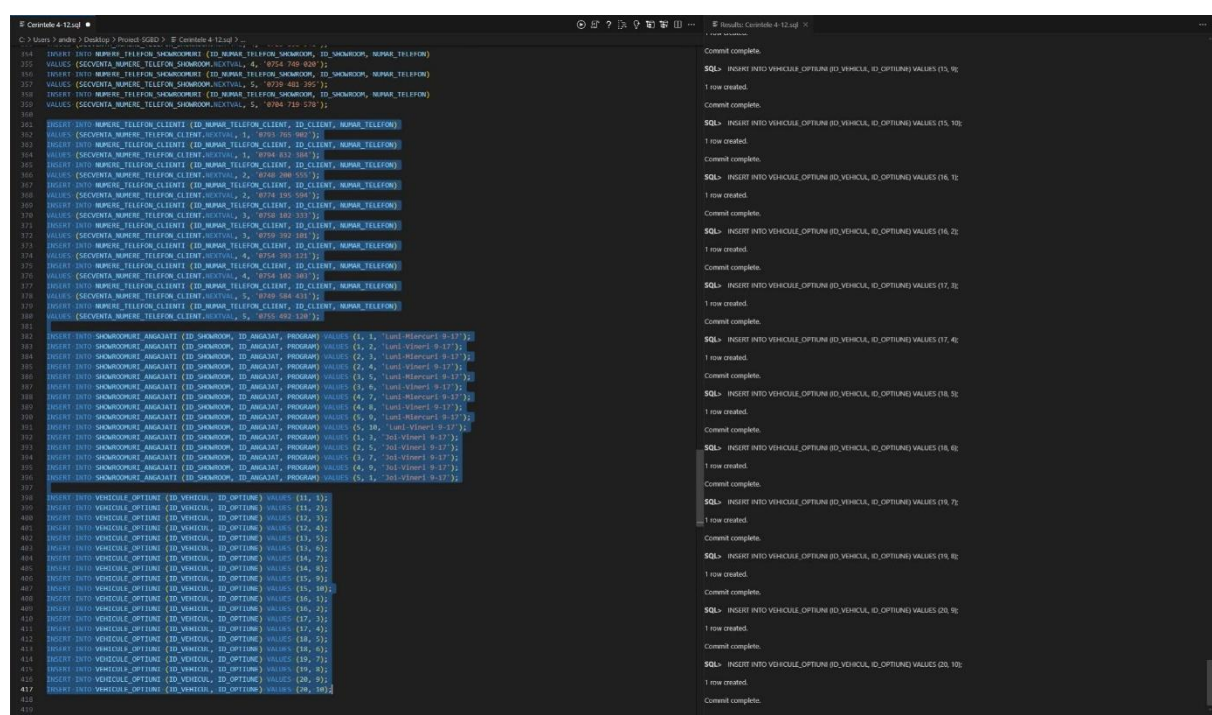
## Poza 2/4:



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;
    TYPE tip_vector is VARRAY(100) OF NUMBER(6);

    t_id_angajati tip_tablou_indexat;
    t_suma_facturi tip_tablou_imbricat := 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);

    e_zero_angajati EXCEPTION;
BEGIN
    -- ID_ANGAJATI
    SELECT
        ID_ANGAJAT
    BULK COLLECT INTO t_id_angajati
    FROM ANGAJATI;
    IF t_id_angajati.COUNT = 0 THEN
        RAISE e_zero_angajati;
    END IF;
    -- 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.EXTEND;
        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 DESCRESATOR
    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;
    DBMS_OUTPUT.PUT_LINE('-----');

EXCEPTION
    WHEN e_zero_angajati THEN
        RAISE_APPLICATION_ERROR(-20001, 'NU EXISTA ANGAJATI IN BAZA DE DATE');
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20000, 'EROARE: ' || SQLERRM);
END cerinta_6;
/

BEGIN
    cerinta_6;
END;
/

```



## Rezultat rulare:

Poza 1/2:

```
PL/SQL procedure successfully completed.

Commit complete.

SQL> CREATE OR REPLACE PROCEDURE cerinta_6 IS
2
3  TYPE tip_tablou_indexat IS TABLE OF ANGAJATI_ID_ANGAJATI%TYPE INDEX BY PLS_INTEGER;
4  TYPE tip_tablou_imbricat IS TABLE OF FACTURI_SUMA%TYPE;
5  TYPE tip_vector IS VARRAY(100) OF NUMBER(6);
6
7  t_id_angajati tip_tablou_indexat;
8  t_suma_facturi tip_tablou_imbricat := tip_tablou_imbricat();
9  t_numar_facturi tip_vector := tip_vector();
10
11  v_suma NUMBER(6) := 0;
12  v_contor NUMBER(6) := 0;
13
14  aux_id_angajat ANGAJATI_ID_ANGAJATI%TYPE;
15  aux_suma_factura FACTURI_SUMA%TYPE;
16  aux_numar_facturi NUMBER(6);
17
18  e_zero_angajati EXCEPTION;
19
20 BEGIN
21
22  -- ID_ANGAJATI
23  SELECT
24    ID_ANGAJAT
25  BULK COLLECT INTO t_id_angajati
26  FROM ANGAJATI;
27
28  IF t_id_angajati.COUNT = 0 THEN
29    RAISE e_zero_angajati;
30  END IF;
31
32  -- SUME FACTURI
33  FOR i IN 1..t_id_angajati.COUNT LOOP
34
35    SELECT
36      NVL(SUM(SUMA), 0)
37    INTO v_suma
38    FROM FACTURI
39    WHERE ID_ANGAJAT = t_id_angajati(i);
40
41    t_suma_facturi.EXTEND;
42    t_suma_facturi(i) := v_suma;
43
44  END LOOP;
45
46  -- NUMAR FACTURI
47  FOR i IN 1..t_id_angajati.COUNT LOOP
48
49    SELECT
50      COUNT(*)
51    INTO v_contor
52    FROM FACTURI
53    WHERE ID_ANGAJAT = t_id_angajati(i);
54
55    t_numar_facturi.EXTEND;
56    t_numar_facturi(i) := v_contor;
57
58  END LOOP;
59
60  -- SORTARE DESCRESCTOR
61  FOR i IN 1..t_id_angajati.COUNT LOOP
62    FOR j IN i+1..t_id_angajati.COUNT LOOP
63      IF t_suma_facturi(i) < t_suma_facturi(j) THEN
64
65        aux_id_angajat := t_id_angajati(i);
66        t_id_angajati(i) := t_id_angajati(j);
67        t_id_angajati(j) := aux_id_angajat;
68
69        aux_suma_factura := t_suma_facturi(i);
70        t_suma_facturi(i) := t_suma_facturi(j);
71        t_suma_facturi(j) := aux_suma_factura;
72
73        aux_numar_facturi := t_numar_facturi(i);
74        t_numar_facturi(i) := t_numar_facturi(j);
75        t_numar_facturi(j) := aux_numar_facturi;
76
77      END IF;
78    END LOOP;
79  END LOOP;
80
81  -- AFISARE
82  FOR i IN 1..t_id_angajati.COUNT LOOP
83
84    DBMS_OUTPUT.PUT_LINE('-----');
85    DBMS_OUTPUT.PUT_LINE('ID_ANGAJAT: ' || t_id_angajati(i));
86    DBMS_OUTPUT.PUT_LINE('TOTALUL SUMELOR FACTURILOR EMISE DE ACESTA: ' || t_suma_facturi(i));
87    DBMS_OUTPUT.PUT_LINE('NUMARUL DE FACTURI EMISE DE ACESTA: ' || t_numar_facturi(i));
88
89  END LOOP;
90  DBMS_OUTPUT.PUT_LINE('-----');
91
92 EXCEPTION
93  WHEN e_zero_angajati THEN
94    RAISE_APPLICATION_ERROR(-20001, 'NU EXISTA ANGAJATI IN BAZA DE DATE');
95  WHEN OTHERS THEN
96    RAISE_APPLICATION_ERROR(-20000, 'EROARE: ' || SQLERRM);
97 END cerinta_6;
98 /

PROCEDURE created.

Commit complete.
```

## Poza 2/2:

```

522         t_suma_facturi(j) := aux_suma_factura;
523     --
524     aux_numar_facturi := t_numar_facturi(i);
525     t_numar_facturi(i) := t_numar_facturi(j);
526     t_numar_facturi(j) := aux_numar_facturi;
527 --
528     END IF;
529     END LOOP;
530 END LOOP;
531
532 -- AFISARE
533 FOR i IN 1..t_id_angajati.COUNT LOOP
534
535     DBMS_OUTPUT.PUT_LINE('-----');
536     DBMS_OUTPUT.PUT_LINE('ID ANGAJAT: ' || t_id_angajati(i));
537     DBMS_OUTPUT.PUT_LINE('TOTALUL SUMELOR FACTURILOR EMISE DE ACESTA: ' || t_suma_facturi(i));
538     DBMS_OUTPUT.PUT_LINE('NUMARUL DE FACTURI EMISE DE ACESTA: ' || t_numar_facturi(i));
539
540     END LOOP;
541     DBMS_OUTPUT.PUT_LINE('-----');
542
543 EXCEPTION
544     WHEN e_zero_angajati THEN
545         RAISE_APPLICATION_ERROR(-20001, 'NU EXISTA ANGAJATI IN BAZA DE DATE');
546     WHEN OTHERS THEN
547         RAISE_APPLICATION_ERROR(-20000, 'EROARE: ' || SQLERRM);
548 END cerinta_6;
549 /
550
551 BEGIN
552     cerinta_6;
553 END;
554 /
555
556 ----- CERINTA 7 -----
557
558 -- Formulati in limbaj natural o problema pe care sa o rezolvati folosind un subprogram stocat
559 -- independent care sa utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor
560 -- parametrizat, dependent de celalalt cursor. Apelati subprogramul.
561
562 -- SA SE AFISEZE VEHICULELE DETINUTE DE CATRE CLIENTII CARE AU FOST VANDUTE IN ANUL 2024
563 -- (ANUL VANZARII SE REGASESTE IN FACTURA)
564 -- LA FIECARE VEHICUL SA SE AFISEZE SI ANGAJATUL CARE A EFECTUAT VANZAREA
565 -- LA FINAL SA SE AFISEZE SUMA TOTALA A VEHICULELOR VANDUTE IN 2024
566
567

```

```

SQL> BEGIN
2   cerinta_6;
3 END;
4 /
-----
ID_ANGAJAT: 1
TOTALUL SUMELOR FACTURILOR EMISE DE ACESTA: 42000
NUMARUL DE FACTURI EMISE DE ACESTA: 3
-----
ID_ANGAJAT: 6
TOTALUL SUMELOR FACTURILOR EMISE DE ACESTA: 16000
NUMARUL DE FACTURI EMISE DE ACESTA: 1
-----
ID_ANGAJAT: 7
TOTALUL SUMELOR FACTURILOR EMISE DE ACESTA: 15000
NUMARUL DE FACTURI EMISE DE ACESTA: 1
-----
ID_ANGAJAT: 4
TOTALUL SUMELOR FACTURILOR EMISE DE ACESTA: 14000
NUMARUL DE FACTURI EMISE DE ACESTA: 1
-----
ID_ANGAJAT: 5
TOTALUL SUMELOR FACTURILOR EMISE DE ACESTA: 13500
NUMARUL DE FACTURI EMISE DE ACESTA: 1
-----
ID_ANGAJAT: 3
TOTALUL SUMELOR FACTURILOR EMISE DE ACESTA: 13000
NUMARUL DE FACTURI EMISE DE ACESTA: 1
-----
ID_ANGAJAT: 2
TOTALUL SUMELOR FACTURILOR EMISE DE ACESTA: 12000
NUMARUL DE FACTURI EMISE DE ACESTA: 1
-----
ID_ANGAJAT: 8
TOTALUL SUMELOR FACTURILOR EMISE DE ACESTA: 12000
NUMARUL DE FACTURI EMISE DE ACESTA: 1
-----
ID_ANGAJAT: 9
TOTALUL SUMELOR FACTURILOR EMISE DE ACESTA: 0
NUMARUL DE FACTURI EMISE DE ACESTA: 0
-----
ID_ANGAJAT: 10
TOTALUL SUMELOR FACTURILOR EMISE DE ACESTA: 0
NUMARUL DE FACTURI EMISE DE ACESTA: 0
-----
PL/SQL procedure successfully completed.
Commit complete.

```



## 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:

```
567 CREATE OR REPLACE PROCEDURE cerinta_7 IS
568
569     CURSOR c_vehicule_clienti (p_id_vehicul_factura FACTURI.ID_VEHICUL%TYPE) IS
570     SELECT
571         ID_VEHICUL,
572         MARCA,
573         MODEL,
574         PRET
575     FROM VEHICULE
576     WHERE ID_VEHICUL = p_id_vehicul_factura;
577
578     v_cont NUMBER(6) := 0;
579     v_suma NUMBER(6) := 0;
580
581     v_nr_vehicule_clienti NUMBER(6) := 0;
582     e_nu_exista_vehicule EXCEPTION;
583
584 BEGIN
585
586     -- verificare existenta vehicule vandute in 2024
587     SELECT
588         COUNT(*)
589     INTO v_nr_vehicule_clienti
590     FROM FACTURI
591     WHERE EXTRACT(YEAR FROM DATA_FACTURA) = 2024;
592
593     IF v_nr_vehicule_clienti = 0 THEN
594         RAISE e_nu_exista_vehicule;
595     END IF;
596
597     DBMS_OUTPUT.PUT_LINE('----- VANDUTE IN ANUL 2024:');
598     DBMS_OUTPUT.PUT_LINE('VEHICULE VANDUTE IN ANUL 2024:');
599     FOR angajat_factura_2024 IN (SELECT
600         F.ID_VEHICUL,
601         A.ID_ANGAJAT,
602         A.NUME
603     FROM FACTURI F
604     JOIN ANGAJATI A ON F.ID_ANGAJAT = A.ID_ANGAJAT
605     WHERE EXTRACT(YEAR FROM DATA_FACTURA) = 2024) LOOP
606
607         FOR vehicul_client IN c_vehicule_clienti(angajat_factura_2024.ID_VEHICUL) LOOP
608
609             v_cont := v_cont + 1;
610             v_suma := v_suma + vehicul_client.PRET;
611             DBMS_OUTPUT.PUT_LINE('-----' || v_cont || '-----');
612             DBMS_OUTPUT.PUT_LINE('ID_VEHICUL: ' || vehicul_client.ID_VEHICUL);
613             DBMS_OUTPUT.PUT_LINE('MARCA: ' || vehicul_client.MARCA);
614             DBMS_OUTPUT.PUT_LINE('MODEL: ' || vehicul_client.MODEL);
615             DBMS_OUTPUT.PUT_LINE('PRET: ' || vehicul_client.PRET);
616             DBMS_OUTPUT.PUT_LINE(' ');
617             DBMS_OUTPUT.PUT_LINE('DATE ANGAJAT: ');
618             DBMS_OUTPUT.PUT_LINE('ID_ANGAJAT: ' || angajat_factura_2024.ID_ANGAJAT);
619             DBMS_OUTPUT.PUT_LINE('NUME ANGAJAT: ' || angajat_factura_2024.NUME);
620
621         END LOOP;
622     END LOOP;
623
624     DBMS_OUTPUT.PUT_LINE('-----');
625     DBMS_OUTPUT.PUT_LINE('SUMA TOTALA A VEHICULELOR VANDUTE IN 2024: ' || v_suma);
626
627 EXCEPTION
628     WHEN e_nu_exista_vehicule THEN
629         RAISE_APPLICATION_ERROR(-20001, 'NU EXISTA VEHICULE VANDUTE IN ANUL 2024');
630     WHEN OTHERS THEN
631         RAISE_APPLICATION_ERROR(-20000, 'EROARE: ' || SQLERRM);
632 END cerinta_7;
633
634 BEGIN
635     cerinta_7;
636 END;
```

```
Commit complete.
SQL> CREATE OR REPLACE PROCEDURE cerinta_7 IS
2
3     CURSOR c_vehicule_clienti (p_id_vehicul_factura FACTURI.ID_VEHICUL%TYPE) IS
4     SELECT
5         ID_VEHICUL,
6         MARCA,
7         MODEL,
8         PRET
9     FROM VEHICULE
10     WHERE ID_VEHICUL = p_id_vehicul_factura;
11
12     v_cont NUMBER(6) := 0;
13     v_suma NUMBER(6) := 0;
14
15     v_nr_vehicule_clienti NUMBER(6) := 0;
16     e_nu_exista_vehicule EXCEPTION;
17
18 BEGIN
19
20     -- verificare existenta vehicule vandute in 2024
21     SELECT
22         COUNT(*)
23     INTO v_nr_vehicule_clienti
24     FROM FACTURI
25     WHERE EXTRACT(YEAR FROM DATA_FACTURA) = 2024;
26
27     IF v_nr_vehicule_clienti = 0 THEN
28         RAISE e_nu_exista_vehicule;
29     END IF;
30
31     DBMS_OUTPUT.PUT_LINE('----- VANDUTE IN ANUL 2024:');
32     DBMS_OUTPUT.PUT_LINE('VEHICULE VANDUTE IN ANUL 2024:');
33     FOR angajat_factura_2024 IN (SELECT
34         F.ID_VEHICUL,
35         A.ID_ANGAJAT,
36         A.NUME
37     FROM FACTURI F
38     JOIN ANGAJATI A ON F.ID_ANGAJAT = A.ID_ANGAJAT
39     WHERE EXTRACT(YEAR FROM DATA_FACTURA) = 2024) LOOP
40
41         FOR vehicul_client IN c_vehicule_clienti(angajat_factura_2024.ID_VEHICUL) LOOP
42
43             v_cont := v_cont + 1;
44             v_suma := v_suma + vehicul_client.PRET;
45             DBMS_OUTPUT.PUT_LINE('-----' || v_cont || '-----');
46             DBMS_OUTPUT.PUT_LINE('ID_VEHICUL: ' || vehicul_client.ID_VEHICUL);
47             DBMS_OUTPUT.PUT_LINE('MARCA: ' || vehicul_client.MARCA);
48             DBMS_OUTPUT.PUT_LINE('MODEL: ' || vehicul_client.MODEL);
49             DBMS_OUTPUT.PUT_LINE('PRET: ' || vehicul_client.PRET);
50             DBMS_OUTPUT.PUT_LINE(' ');
51             DBMS_OUTPUT.PUT_LINE('DATE ANGAJAT: ');
52             DBMS_OUTPUT.PUT_LINE('ID_ANGAJAT: ' || angajat_factura_2024.ID_ANGAJAT);
53             DBMS_OUTPUT.PUT_LINE('NUME ANGAJAT: ' || angajat_factura_2024.NUME);
54
55         END LOOP;
56     END LOOP;
57
58     DBMS_OUTPUT.PUT_LINE('-----');
59     DBMS_OUTPUT.PUT_LINE('SUMA TOTALA A VEHICULELOR VANDUTE IN 2024: ' || v_suma);
60
61 EXCEPTION
62     WHEN e_nu_exista_vehicule THEN
63         RAISE_APPLICATION_ERROR(-20001, 'NU EXISTA VEHICULE VANDUTE IN ANUL 2024');
64     WHEN OTHERS THEN
65         RAISE_APPLICATION_ERROR(-20000, 'EROARE: ' || SQLERRM);
66 END cerinta_7;
67 /
68
69 PROCEDURE created.
70
71 Commit complete.
```

Poza 2/2:

```

95 JOIN ANGAJATI A ON F.ID_ANGAJAT = A.ID_ANGAJAT
96 WHERE EXTRACT(YEAR FROM DATA_FACTURA) = 2024) LOOP
97
98     FOR vehicul_client in c_vehicule_clienti(angajat_factura_2024.ID_VEHICUL) LOOP
99
100         v_cont := v_cont + 1;
101         v_suma := v_suma + vehicul_client.PRET;
102         DBMS_OUTPUT.PUT_LINE('-----' || v_cont || '-----');
103         DBMS_OUTPUT.PUT_LINE('ID_VEHICUL: ' || vehicul_client.ID_VEHICUL);
104         DBMS_OUTPUT.PUT_LINE('MARCA: ' || vehicul_client.MARCA);
105         DBMS_OUTPUT.PUT_LINE('MODEL: ' || vehicul_client.MODEL);
106         DBMS_OUTPUT.PUT_LINE('PRET: ' || vehicul_client.PRET);
107         DBMS_OUTPUT.PUT_LINE(' ');
108         DBMS_OUTPUT.PUT_LINE('DATE_ANGAJAT: ');
109         DBMS_OUTPUT.PUT_LINE('ID_ANGAJAT: ' || angajat_factura_2024.ID_ANGAJAT);
110         DBMS_OUTPUT.PUT_LINE('NUME_ANGAJAT: ' || angajat_factura_2024.NUME);
111
112     END LOOP;
113
114 END LOOP;
115 DBMS_OUTPUT.PUT_LINE('-----');
116 DBMS_OUTPUT.PUT_LINE('SUMA TOTALA A VEHICULELOR VANDUTE IN 2024: ' || v_suma);
117
118 EXCEPTION
119 WHEN e_nu_exista_vehicule THEN
120     RAISE_APPLICATION_ERROR(-20001, 'NU EXISTA VEHICULE VANDUTE IN ANUL 2024');
121 WHEN OTHERS THEN
122     RAISE_APPLICATION_ERROR(-20000, 'EROARE: ' || SQLERRM);
123 END cerinta_7;
124 /
125
126 BEGIN
127     cerinta_7;
128 END;
129 /
130
131 ----- CERINTA 8 -----
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
265
```

## 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 și returnează un număr de telefon al acestuia și numărul de vehicule pe care le-a cumpărat. Dacă nu există clienți cu acel nume, se va afișa un mesaj. Dacă există 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_VEICUL) AS NUMAR_VEICULE
INTO v_numar_telefon, v_nr_vehicule
FROM CLIENTI C
LEFT JOIN VEICULE 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:

```
655 CREATE OR REPLACE FUNCTION cerinta_8(p_nume_client IN CLIENTI.NUME%TYPE) RETURN VARCHAR2 IS
656
657     v_nr_clienti NUMBER(6) := 0;
658     v_numar_telefon NUMERE_TELEFON.CLIENTI.NUMAR_TELEFON%TYPE;
659     v_nr_vehicule NUMBER(6) := 0;
660     vRezultat VARCHAR2(256);
661
662     -- EXCEPTII-PROPRII
663     e_nu_exista_clienti EXCEPTION;
664     e_prea_multi_clienti EXCEPTION;
665
666 BEGIN
667
668     SELECT
669         COUNT(*)
670     INTO v_nr_clienti
671     FROM CLIENTI
672     WHERE UPPER(NUME) = UPPER('p_nume_client');
673
674     -- EXCEPTII-PROPRII
675     -- IF v_nr_clienti = 0 THEN
676     --     RAISE e_nu_exista_clienti;
677     -- ELSEIF v_nr_clienti > 1 THEN
678     --     RAISE e_prea_multi_clienti;
679     -- END IF;
680
681     SELECT
682         (SELECT
683             NTC.NUMAR_TELEFON
684             FROM NUMERE_TELEFON CLIENTI NTC
685             WHERE C.ID_CLIENT = NTC.ID_CLIENT AND ROWNUM = 1
686             ) AS NUMAR_TELEFON,
687         COUNT(V.ID_VEHICUL) AS NUMAR_VEHICULE
688     INTO v_numar_telefon, v_nr_vehicule
689     FROM CLIENTI C
690     LEFT JOIN VEHICULE V ON C.ID_CLIENT = V.ID_CLIENT
691     WHERE UPPER(C.NUME) = UPPER(p_nume_client)
692     GROUP BY C.ID_CLIENT;
693
694     vRezultat := 'Clientul cu numele ' || p_nume_client || ' are numarul de telefon ' || v_numar_telefon || ' si
695     RETURN vRezultat;
696
697 EXCEPTION
698     -- EXCEPTII-PROPRII
699     -- WHEN e_nu_exista_clienti THEN
700     --     RAISE_APPLICATION_ERROR(-20001, 'NU-EXISTA-CLIENTI-CU-ACEST-NUME');
701     -- WHEN e_prea_multi_clienti THEN
702     --     RAISE_APPLICATION_ERROR(-20002, 'EXISTA-MAI-MULTI-CLIENTI-CU-ACEST-NUME');
703     -- WHEN TOO_MANY_ROWS THEN
704     --     RAISE_APPLICATION_ERROR(-20004, 'TOO-MANY-ROWS' || SQLERRM);
705     -- WHEN NO_DATA_FOUND THEN
706     --     RAISE_APPLICATION_ERROR(-20003, 'NO-DATA-FOUND' || SQLERRM);
707     -- WHEN OTHERS THEN
708     --     RAISE_APPLICATION_ERROR(-20000, 'EROARE: ' || SQLERRM);
709 END cerinta_8;
710 /
711
712 DECLARE
713
714     vRezultat VARCHAR2(256);
715
716 BEGIN
717
718     vRezultat := cerinta_8('ION POPOESCU');
719     DBMS_OUTPUT.PUT_LINE(vRezultat);
720
721     -- vRezultat := cerinta_8('CLIENT-DUPLICAT');
722     -- DBMS_OUTPUT.PUT_LINE(vRezultat);
723
724     -- vRezultat := cerinta_8('CLIENT-INEXISTENT');
725     -- DBMS_OUTPUT.PUT_LINE(vRezultat);
726
727 END;
728 /
729
730 ----- CERINTA 9 -----
731
732
733 -- Formulati in limba natural o problema pe care sa o rezolvati folosind un subprogram stocat
734 -- independent de tip procedura care sa aiba minim 2 parametri si sa utilizeze intr-o singura
735 -- comanda SQL 5 dintre tabelele create. Definiti minim 2 exceptii proprii, altele decat cele
```

```
1 row created.
Commit complete.
SQL> CREATE OR REPLACE FUNCTION cerinta_8(p_nume_client IN CLIENTI.NUME%TYPE) RETURN VARCHAR2 IS
2
3     v_nr_clienti NUMBER(6) := 0;
4     v_numar_telefon NUMERE_TELEFON.CLIENTI.NUMAR_TELEFON%TYPE;
5     v_nr_vehicule NUMBER(6) := 0;
6     vRezultat VARCHAR2(256);
7
8     -- EXCEPTII-PROPRII
9     e_nu_exista_clienti EXCEPTION;
10    e_prea_multi_clienti EXCEPTION;
11
12 BEGIN
13
14     SELECT
15         COUNT(*)
16     INTO v_nr_clienti
17     FROM CLIENTI
18     WHERE UPPER(NUME) = UPPER('p_nume_client');
19
20     -- EXCEPTII-PROPRII
21     -- IF v_nr_clienti = 0 THEN
22     --     RAISE e_nu_exista_clienti;
23     -- ELSEIF v_nr_clienti > 1 THEN
24     --     RAISE e_prea_multi_clienti;
25     -- END IF;
26
27     SELECT
28         (SELECT
29             NTC.NUMAR_TELEFON
30             FROM NUMERE_TELEFON CLIENTI NTC
31             WHERE C.ID_CLIENT = NTC.ID_CLIENT AND ROWNUM = 1
32             ) AS NUMAR_TELEFON,
33         COUNT(V.ID_VEHICUL) AS NUMAR_VEHICULE
34     INTO v_numar_telefon, v_nr_vehicule
35     FROM CLIENTI C
36     LEFT JOIN VEHICULE V ON C.ID_CLIENT = V.ID_CLIENT
37     WHERE UPPER(C.NUME) = UPPER(p_nume_client)
38     GROUP BY C.ID_CLIENT;
39
40     vRezultat := 'Clientul cu numele ' || p_nume_client || ' are numarul de telefon ' || v_numar_telefon || ' si detine ' || v_nr_vehicule || ' vehicule';
41     RETURN vRezultat;
42
43 EXCEPTION
44     -- EXCEPTII-PROPRII
45     -- WHEN e_nu_exista_clienti THEN
46     --     RAISE_APPLICATION_ERROR(-20001, 'NU-EXISTA-CLIENTI-CU-ACEST-NUME');
47     -- WHEN e_prea_multi_clienti THEN
48     --     RAISE_APPLICATION_ERROR(-20002, 'EXISTA-MAI-MULTI-CLIENTI-CU-ACEST-NUME');
49     -- WHEN TOO_MANY_ROWS THEN
50     --     RAISE_APPLICATION_ERROR(-20004, 'TOO-MANY-ROWS' || SQLERRM);
51     -- WHEN NO_DATA_FOUND THEN
52     --     RAISE_APPLICATION_ERROR(-20003, 'NO-DATA-FOUND' || SQLERRM);
53     -- WHEN OTHERS THEN
54     --     RAISE_APPLICATION_ERROR(-20000, 'EROARE: ' || SQLERRM);
55 END cerinta_8;
56 /
FUNCTION created.
Commit complete.
```

Poza 2/4:

```
704     RAISE_APPLICATION_ERROR(-20004, 'TOO-MANY-ROWS' || SQLERRM);
705     WHEN NO_DATA_FOUND THEN
706         RAISE_APPLICATION_ERROR(-20003, 'NO-DATA-FOUND' || SQLERRM);
707     WHEN OTHERS THEN
708         RAISE_APPLICATION_ERROR(-20000, 'EROARE: ' || SQLERRM);
709 END cerinta_8;
710 /
711
712 DECLARE
713
714     vRezultat VARCHAR2(256);
715
716 BEGIN
717
718     vRezultat := cerinta_8('ION POPOESCU');
719     DBMS_OUTPUT.PUT_LINE(vRezultat);
720
721     -- vRezultat := cerinta_8('CLIENT-DUPLICAT');
722     -- DBMS_OUTPUT.PUT_LINE(vRezultat);
723
724     -- vRezultat := cerinta_8('CLIENT-INEXISTENT');
725     -- DBMS_OUTPUT.PUT_LINE(vRezultat);
726
727 END;
728 /
729
730 ----- CERINTA 9 -----
731
732
733 -- Formulati in limba natural o problema pe care sa o rezolvati folosind un subprogram stocat
734 -- independent de tip procedura care sa aiba minim 2 parametri si sa utilizeze intr-o singura
735 -- comanda SQL 5 dintre tabelele create. Definiti minim 2 exceptii proprii, altele decat cele
```

```
17 /
Clientul cu numele ION POPOESCU are numarul de telefon 0793 765 902 si detine 3 vehicule.
PL/SQL procedure successfully completed.
Commit complete.
SQL> DECLARE
2
3     vRezultat VARCHAR2(256);
4
5 BEGIN
6
7     vRezultat := cerinta_8('ION POPOESCU');
8     DBMS_OUTPUT.PUT_LINE(vRezultat);
9
10    -- vRezultat := cerinta_8('CLIENT DUPLICAT');
11    -- DBMS_OUTPUT.PUT_LINE(vRezultat);
12
13    -- vRezultat := cerinta_8('CLIENT INEXISTENT');
14    -- DBMS_OUTPUT.PUT_LINE(vRezultat);
15
16 END;
17 /
Clientul cu numele ION POPOESCU are numarul de telefon 0793 765 902 si detine 3 vehicule.
PL/SQL procedure successfully completed.
Commit complete.
```

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

<pre> 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; /  ----- CERINTA 9 -----  -- Formulati in limbaj natural o problema pe care sa o rezolvati folosind un subprogram stocat -- independent de tip procedura care sa aiba minim 2 parametri si sa utilizeze intr-o singura -- comanda SQL 5 dintre tabelele create. Definiti minim 2 exceptii proprii, altele decit cele </pre>	<pre> Commit complete.  SQL&gt; DECLARE 2 3   v_rezultat VARCHAR2(256); 4 5 BEGIN 6 7   -- v_rezultat := cerinta_8('ION POPESCU'); 8   -- DBMS_OUTPUT.PUT_LINE(v_rezultat); 9 10  v_rezultat := cerinta_8('CLIENT DUPLICAT'); 11  DBMS_OUTPUT.PUT_LINE(v_rezultat); 12 13  -- v_rezultat := cerinta_8('CLIENT INEXISTENT'); 14  -- DBMS_OUTPUT.PUT_LINE(v_rezultat); 15 16 END; 17 /  DECLARE * ERROR at line 712: ORA-20004: TOO_MANY_ROWS ORA-01422: exact fetch returns more than requested number of rows ORA-06512: at "ANDREI4.CERINTA_8", line 50 ORA-06512: at line 10 ORA-06512: at line 10 https://docs.oracle.com/error-help/db/ora-20004/ </pre>
--	--

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

<pre> 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; /  ----- CERINTA 9 -----  -- Formulati in limbaj natural o problema pe care sa o rezolvati folosind un subprogram stocat -- independent de tip procedura care sa aiba minim 2 parametri si sa utilizeze intr-o singura -- comanda SQL 5 dintre tabelele create. Definiti minim 2 exceptii proprii, altele decit cele </pre>	<pre> ORA-06512: at "ANDREI4.CERINTA_8", line 50 ORA-06512: at line 10 https://docs.oracle.com/error-help/db/ora-20004/ SQL&gt; DECLARE 2 3   v_rezultat VARCHAR2(256); 4 5 BEGIN 6 7   -- v_rezultat := cerinta_8('ION POPESCU'); 8   -- DBMS_OUTPUT.PUT_LINE(v_rezultat); 9 10  -- v_rezultat := cerinta_8('CLIENT DUPLICAT'); 11  -- DBMS_OUTPUT.PUT_LINE(v_rezultat); 12 13  v_rezultat := cerinta_8('CLIENT INEXISTENT'); 14  DBMS_OUTPUT.PUT_LINE(v_rezultat); 15 16 END; 17 /  DECLARE * ERROR at line 712: ORA-20003: NO DATA FOUND ORA-01403: no data found ORA-06512: at "ANDREI4.CERINTA_8", line 52 ORA-06512: at line 13 https://docs.oracle.com/error-help/db/ora-20003/ </pre>
--	---



## 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_vehicle NUMBER(6) := 0;
v_nr_optiuni NUMBER(6) := 0;

e_nu_exista_vehicle 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_vehicle
    FROM VEHICULE
    WHERE ID_VEHICUL = p_id_vehicul AND UPPER(MARCA) = UPPER(p_marca)
        AND UPPER(MODEL) = UPPER(p_model);

    IF v_nr_vehicle = 0 THEN
        RAISE e_nu_exista_vehicle;
    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.numa_client, v_r_cerinta_9.numa_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.num_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.num_client);
ELSE
    DBMS_OUTPUT.PUT_LINE(' ');
    DBMS_OUTPUT.PUT_LINE('----- DATE SHOWROOM -----');
    DBMS_OUTPUT.PUT_LINE('NUME SHOWROOM: ' || v_r_cerinta_9.num_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:

```
G:\Users\andre\Desktop> Project-SCGD > E Cerintele 4-12.sql > PROCEDURE cerinta_9
179 CREATE OR REPLACE PROCEDURE cerinta_9(p_id_vehicul IN VEHICLE.ID_VEHICULTYPE, p_marca IN VEHICLE.MARCA)
180
181 TYPE r_cerinta_9 IS RECORD (
182
183     id_vehicul VEHICLE.ID_VEHICULTYPE,
184     marca VEHICLE.MARCA%TYPE,
185     model VEHICLE.MODEL%TYPE,
186     an_fabricatie VEHICLE.AN_FABRICATIENTYPE,
187     pret VEHICLE.PRET%TYPE,
188
189     id_motor MOTOARE.ID_MOTOR%TYPE,
190     cai_putere MOTOARE.CAI_PUTERENTYPE,
191     capacitate_cilindrica MOTOARE.CAPACITATE_CILINDRICATYPE,
192
193     nume_client CLIENTI.NUMENTYPE,
194     -- SAU
195     nume_showroom SHOWROOMURI.NUMENTYPE,
196
197     optiuni VARCHAR2(2048)
198 );
199
200 v_r_cerinta_9 r_cerinta_9;
201
202 v_nr_vehicule NUMBER(6) := 0;
203 v_nr_optiuni NUMBER(6) := 0;
204
205 e_nu_exista_vehicule EXCEPTION;
206 e_nu_are_optiuni EXCEPTION;
207
208 BEGIN
209
210     -- EXTRAGERE NUMARUL DE VEHICULE CU COMBINATIA p_id_vehicul, p_marca, p_model
211     SELECT
212         COUNT(*)
213     INTO v_nr_vehicule
214     FROM VEHICLE
215     WHERE ID_VEHICUL = p_id_vehicul AND UPPER(MARCA) = UPPER(p_marca)
216           AND UPPER(MODEL) = UPPER(p_model);
217
218     IF v_nr_vehicule = 0 THEN
219         RAISE e_nu_exista_vehicule;
220     END IF;
221
222     -- EXTRAGERE NUMAR OPTIUNI
223     SELECT
224         COUNT(*)
225     INTO v_nr_optiuni
226     FROM VEHICLE_OPTIUNI
227     WHERE ID_VEHICUL = p_id_vehicul;
228
229     SELECT
230         V.ID_VEHICUL,
231         V.MARCA,
232         V.MODEL,
233         V.AN_FABRICATIE,
234         V.PRET,
235         M.ID_MOTOR,
236         M.CAI_PUTERE,
237         M.CAPACITATE_CILINDRICA,
238         C.NUME,
239         S.NUME,
240         LISTAGG(C.NUME, ',' ) WITHIN GROUP (ORDER BY O.ID_OPTIUNE)
241     INTO v_r_cerinta_9.id_vehicul, v_r_cerinta_9.marca, v_r_cerinta_9.model,
242         v_r_cerinta_9.an_fabricatie, v_r_cerinta_9.pret, v_r_cerinta_9.id_motor,
243         v_r_cerinta_9.cai_putere, v_r_cerinta_9.capacitate_cilindrica,
244         v_r_cerinta_9.nume_client, v_r_cerinta_9.nume_showroom, v_r_cerinta_9.optiuni
245     FROM VEHICLE V
246     JOIN MOTOARE M ON V.ID_MOTOR = M.ID_MOTOR
247     LEFT JOIN CLIENTI C ON V.ID_CLIENT = C.ID_CLIENT
248     LEFT JOIN SHOWROOMURI S ON V.ID_SHOWROOM = S.ID_SHOWROOM
249     LEFT JOIN VEHICLE_OPTIUNI VO ON V.ID_VEHICUL = VO.ID_VEHICUL
250     LEFT JOIN OPTIUNI O ON VO.ID_OPTIUNE = O.ID_OPTIUNE
251     WHERE V.ID_VEHICUL = p_id_vehicul AND UPPER(V.MARCA) = UPPER(p_marca)
252           AND UPPER(V.MODEL) = UPPER(p_model)
253     GROUP BY V.ID_VEHICUL, V.MARCA, V.MODEL, V.AN_FABRICATIE, V.PRET, M.ID_MOTOR,
254           M.CAI_PUTERE, M.CAPACITATE_CILINDRICA, C.NUME, S.NUME;
255
256     -- AFISARE VEHICUL
257     DBMS_OUTPUT.PUT_LINE('----- DATE VEHICUL -----');
258     DBMS_OUTPUT.PUT_LINE('ID_VEHICUL: ' || v_r_cerinta_9.id_vehicul);
259     DBMS_OUTPUT.PUT_LINE('MARCA: ' || v_r_cerinta_9.marca);
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

```

C:\Users\andre\Desktop> Proiect-SGBD > Cerintele 4-12.sql > PROCEDURE cerinta_9

749 CREATE OR REPLACE PROCEDURE cerinta_9(p_id_vehicul IN VEHICLE.ID_VEHICUL%TYPE, p_marca IN VEHICLE.MARCA%TYPE)
750 AS
751     v_id_vehicul V.ID_VEHICUL,
752     v_marca V.MARCA,
753     v_model V.MODEL,
754     v_an_fabricatie V.AN_FABRICATIE,
755     v_pret V.PRET,
756     m_id_motor M.ID_MOTOR,
757     m_cai_putere M.CAI_PUTERE,
758     m_capacitate_cilindrica M.CAPACITATE_CILINDRICA,
759     s_nume S.NUME,
760     LISTAGG(S.NUME, ',' WITHIN GROUP (ORDER BY O.ID_OPTIUNE)
761 INTO v_r_cerinta_9_id_vehicul, v_r_cerinta_9_marca, v_r_cerinta_9_model,
762     v_r_cerinta_9_an_fabricatie, v_r_cerinta_9_pret, v_r_cerinta_9_id_motor,
763     v_r_cerinta_9_cai_putere, v_r_cerinta_9_capacitate_cilindrica,
764     v_r_cerinta_9_nume_client, v_r_cerinta_9_nume_showroom, v_r_cerinta_9_optiuni
765 FROM VEHICLE V
766 JOIN MOTOARE M ON V.ID_MOTOR = M.ID_MOTOR
767 LEFT JOIN CLIENTI C ON V.ID_CLIENT = C.ID_CLIENT
768 LEFT JOIN SHOWROOMURI S ON V.ID_SHOWROOM = S.ID_SHOWROOM
769 LEFT JOIN VEHICLE_OPTIUNI VO ON V.ID_VEHICUL = VO.ID_VEHICUL
770 LEFT JOIN OPTIUNI O ON VO.ID_OPTIUNE = O.ID_OPTIUNE
771 WHERE V.ID_VEHICUL = p_id_vehicul AND UPPER(V.MARCA) = UPPER(p_marca)
772 AND UPPER(V.MODEL) = UPPER(p_model)
773 GROUP BY V.ID_VEHICUL, V.MARCA, V.MODEL, V.AN_FABRICATIE, V.PRET, M.ID_MOTOR,
774     M.CAI_PUTERE, M.CAPACITATE_CILINDRICA, C.NUME, S.NUME;
775
776 -- AFISARE-VEHICUL
777 DBMS_OUTPUT.PUT_LINE('----- DATE VEHICUL -----');
778 DBMS_OUTPUT.PUT_LINE('ID_VEHICUL: ' || v_r_cerinta_9_id_vehicul);
779 DBMS_OUTPUT.PUT_LINE('MARCA: ' || v_r_cerinta_9_marca);
780 DBMS_OUTPUT.PUT_LINE('MODEL: ' || v_r_cerinta_9_model);
781 DBMS_OUTPUT.PUT_LINE('AN_FABRICATIE: ' || v_r_cerinta_9_an_fabricatie);
782 DBMS_OUTPUT.PUT_LINE('PRET: ' || v_r_cerinta_9_pret);
783
784 -- AFISARE-MOTOR
785 DBMS_OUTPUT.PUT_LINE('----- DATE MOTOR -----');
786 DBMS_OUTPUT.PUT_LINE('ID_MOTOR: ' || v_r_cerinta_9_id_motor);
787 DBMS_OUTPUT.PUT_LINE('CAI_PUTERE: ' || v_r_cerinta_9_cai_putere);
788 DBMS_OUTPUT.PUT_LINE('CAPACITATE_CILINDRICA: ' || v_r_cerinta_9_capacitate_cilindrica);
789
790 -- AFISARE-OPTIUNI
791 BEGIN
792     DBMS_OUTPUT.PUT_LINE('----- DATE OPTIUNI -----');
793     IF v_nr_optiuni = 0 THEN
794         RAISE e_nu_are_optiuni;
795     ELSE
796         DBMS_OUTPUT.PUT_LINE('OPTIUNI EXTRA: ' || v_r_cerinta_9_optiuni);
797     END IF;
798 EXCEPTION
799     WHEN e_nu_are_optiuni THEN
800         DBMS_OUTPUT.PUT_LINE('VEHICULUL NU ARE OPTIUNI EXTRA');
801 END;
802
803 -- AFISARE-CLIENT/SHOWROOM
804 IF v_r_cerinta_9_nume_client IS NOT NULL THEN
805     DBMS_OUTPUT.PUT_LINE('----- DATE CLIENT -----');
806     DBMS_OUTPUT.PUT_LINE('NUME_CLIENT: ' || v_r_cerinta_9_nume_client);
807 ELSE
808     DBMS_OUTPUT.PUT_LINE('----- DATE SHOWROOM -----');
809     DBMS_OUTPUT.PUT_LINE('NUME_SHOWROOM: ' || v_r_cerinta_9_nume_showroom);
810 END IF;
811
812 EXCEPTION
813     WHEN e_nu_exista_vehicule THEN
814         RAISE_APPLICATION_ERROR(-20001, 'NU-EXISTA-VEHICULE CU ACEST-ID, MARCA SI MODEL');
815     WHEN OTHERS THEN
816         RAISE_APPLICATION_ERROR(-20000, 'EROARE: ' || SQLERRM);
817 END cerinta_9;
818 /
819
820 BEGIN
821     cerinta_9(1, 'DACIA', 'DUSTER');
822
823     DBMS_OUTPUT.PUT_LINE(' ');
824     DBMS_OUTPUT.PUT_LINE(' ');
825     DBMS_OUTPUT.PUT_LINE(' ');
826
827 COMMIT;
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

749 CREATE OR REPLACE PROCEDURE cerinta_9(p_id_vehicul IN VEHICUL.ID_VEHICUL%TYPE, p_marca IN VEHICUL.MARCA
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1
```

## Poza 4/4 (vehicul inexistent):

```

855 -- IF IDBMS_CLIENT=SHOWROOM
856 IF v_r_cerinta_9.ume_client IS NOT NULL THEN
857 DBMS_OUTPUT.PUT_LINE('.');
858 DBMS_OUTPUT.PUT_LINE('----- DATE CLIENT -----');
859 DBMS_OUTPUT.PUT_LINE('NUME CLIENT: ' || v_r_cerinta_9.ume_client);
860 ELSE
861 DBMS_OUTPUT.PUT_LINE('.');
862 DBMS_OUTPUT.PUT_LINE('----- DATE SHOWROOM -----');
863 DBMS_OUTPUT.PUT_LINE('NUME SHOWROOM: ' || v_r_cerinta_9.ume_showroom);
864 END IF;
865
866 EXCEPTION
867 WHEN e_nu_exista_vehicule THEN
868 RAISE_APPLICATION_ERROR(-20001, 'NU EXISTA VEHICULE CU ACEST ID, MARCA SI MODEL');
869 WHEN OTHERS THEN
870 RAISE_APPLICATION_ERROR(-20000, 'ERORARE: ' || SQLERRM);
871 END cerinta_9;
872 /
873
874 BEGIN
875
876 cerinta_9(1, 'DACIA', 'DUSTER');
877
878 DBMS_OUTPUT.PUT_LINE('.');
879 DBMS_OUTPUT.PUT_LINE('.');
880 DBMS_OUTPUT.PUT_LINE('-----');
881 DBMS_OUTPUT.PUT_LINE('.');
882 DBMS_OUTPUT.PUT_LINE('.');
883
884 cerinta_9(11, 'DACIA', 'DUSTER');
885
886 DBMS_OUTPUT.PUT_LINE('.');
887 DBMS_OUTPUT.PUT_LINE('.');
888 DBMS_OUTPUT.PUT_LINE('-----');
889 DBMS_OUTPUT.PUT_LINE('.');
890 DBMS_OUTPUT.PUT_LINE('.');
891
892 cerinta_9(21, 'DACIA', 'DUSTER');
893
894 cerinta_9(-1, 'VEHICUL', 'INEXISTENT');
895
896 END;
897 /
898
899 ----- CERINTA-10 -----
900
901 -- Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.
902
903 -- CAND SE ADAUGA UN VEHICUL INTR-UN SHOWROOM (ID_SHOWROOM != NULL),
904 -- TOATE CELE ALTE VEHICULE DIN ACELASI SHOWROOM DEVIN AUTOMATIC VEHICULE DE TIP

```

```

----- DATE VEHICUL -----
ID_VEHICUL: 21
MARCA: Dacia
MODEL: Duster
AN FABRICATIE: 2021
PRET: 15000

----- DATE MOTOR -----
ID_MOTOR: 1
CAI PUTERE: 110
CAPACITATE CILINDRICA: 1461

----- DATE OPTIUNI -----
VEHICULUL NU ARE OPTIUNI EXTRA

----- DATE CLIENT -----
NUME CLIENT: Ion Popescu

PL/SQL procedure successfully completed.

Commit complete.

SQL> BEGIN
2
3 -- cerinta_9(1, 'DACIA', 'DUSTER');
4
5 -- DBMS_OUTPUT.PUT_LINE('.');
6 -- DBMS_OUTPUT.PUT_LINE('.');
7 -- DBMS_OUTPUT.PUT_LINE('-----');
8 -- DBMS_OUTPUT.PUT_LINE('.');
9 -- DBMS_OUTPUT.PUT_LINE('.');
10
11 -- cerinta_9(11, 'DACIA', 'DUSTER');
12
13 -- DBMS_OUTPUT.PUT_LINE('.');
14 -- DBMS_OUTPUT.PUT_LINE('.');
15 -- DBMS_OUTPUT.PUT_LINE('-----');
16 -- DBMS_OUTPUT.PUT_LINE('.');
17 -- DBMS_OUTPUT.PUT_LINE('.');
18
19 -- cerinta_9(21, 'DACIA', 'DUSTER');
20
21 cerinta_9(-1, 'VEHICUL', 'INEXISTENT');
22
23 END;
24 /

BEGIN
*
ERROR at line 874:
ORA-20001: NU EXISTA VEHICULE CU ACEST ID, MARCA SI MODEL
ORA-06512: at "ANDREI4.CERINTA_9", line 120
ORA-06512: at line 21

```



## 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:

```
899 ----- CERINTA 10 -----
900
901 --Definiti un trigger de tip DML la nivel de comanda. Declansati trigger-ul.
902
903 -- CAND SE ADAUGA UN VEHICUL INTR-UN SHOWROOM (ID_SHOWROOM != NULL),
904 -- DATELE CELELALTE VEHICULE DIN SHOWROOM-UL RESPECTIV PRIMESC O REDUCERE DE 1%
905 -- NOUL VEHICUL ADAUGAT NU PRIMESTE REDUCERE
906
907
908
909 CREATE OR REPLACE TRIGGER trigger_cerinta_10
910 AFTER INSERT ON VEHICLE
911 DECLARE
912     v_id_showroom VEHICLE.ID_SHOWROOM%TYPE;
913
914 BEGIN
915     DBMS_OUTPUT.PUT_LINE(' ');
916     DBMS_OUTPUT.PUT_LINE('----- DECLANSARE TRIGGER CERINTA 10 -----');
917     DBMS_OUTPUT.PUT_LINE(' ');
918
919     SELECT
920         ID_SHOWROOM
921     INTO v_id_showroom
922     FROM VEHICLE
923     WHERE ID_VEHICUL = (SELECT
924                         MAX(ID_VEHICUL)
925                         FROM VEHICLE);
926
927     IF v_id_showroom IS NOT NULL THEN -- altfel este vehiculul unui client
928
929         UPDATE VEHICLE
930         SET PRET = PRET * 0.99
931         WHERE ID_SHOWROOM = v_id_showroom
932         AND ID_VEHICUL != (SELECT
933                          MAX(id_vehicul)
934                          FROM vehicle);
935
936     END IF;
937
938 END trigger_cerinta_10;
939 /
940
941 DROP TRIGGER trigger_cerinta_10;
942
943 -- EXEMPLU
944 SELECT ID_VEHICUL, PRET FROM VEHICLE WHERE ID_SHOWROOM = 1;
945
946 INSERT INTO VEHICLE (ID_VEHICUL, ID_MOTOR, ID_SHOWROOM, ID_CLIENT, MARCA, MODEL, AN_FABRICATIE, PRET)
947 VALUES (1000, 1, 1, NULL, 'TEST', 'TEST', 2024, 1000);
948
949 SELECT ID_VEHICUL, PRET FROM VEHICLE WHERE ID_SHOWROOM = 1;
950
951 ----- CERINTA 11 -----
952
953 --Definiti un trigger de tip DML la nivel de linie. Declansati trigger-ul.
954
955 -- CAND ADAUGI SAU STERGI O OPTIUNE A UNUI VEHICUL, ACTUALIZEAZA PRETUL CORESPUNZATOR
956
```

```
12
13 -- DBMS_OUTPUT.PUT_LINE(' ');
14 -- DBMS_OUTPUT.PUT_LINE(' ');
15 -- DBMS_OUTPUT.PUT_LINE('----- DECLANSARE TRIGGER CERINTA 10 -----');
16 -- DBMS_OUTPUT.PUT_LINE(' ');
17 -- DBMS_OUTPUT.PUT_LINE(' ');
18
19 -- cerinta_9(21, 'DACIA', 'DUSTER');
20
21 cerinta_9(1, 'VEHICUL', 'INEXISTENT');
22
23 END;
24 /
25
26 BEGIN
27 ERROR at line 874:
28 ORA-20001: NO EXISTA VEHICULE CU ACEST ID, MARCA SI MODEL
29 ORA-06512: at "ANDREI14.CERINTA_9", line 120
30 ORA-06512: at line 21
31 https://docs.oracle.com/error-help/db/ora-20001/
32 SQL> DROP TRIGGER trigger_cerinta_10;
33
34 DROP TRIGGER trigger_cerinta_10
35 *
36 ERROR at line 943:
37 ORA-04080: trigger 'TRIGGER_CERINTA_10' does not exist
38 https://docs.oracle.com/error-help/db/ora-04080/
39 SQL> CREATE OR REPLACE TRIGGER trigger_cerinta_10
40 2 AFTER INSERT ON VEHICLE
41 3 DECLARE
42 4
43 5     v_id_showroom VEHICLE.ID_SHOWROOM%TYPE;
44 6
45 7 BEGIN
46 8
47 9     DBMS_OUTPUT.PUT_LINE(' ');
48 10    DBMS_OUTPUT.PUT_LINE('----- DECLANSARE TRIGGER CERINTA 10 -----');
49 11    DBMS_OUTPUT.PUT_LINE(' ');
50 12
51 13    SELECT
52 14        ID_SHOWROOM
53 15    INTO v_id_showroom
54 16    FROM VEHICLE
55 17    WHERE ID_VEHICUL = (SELECT
56 18                        MAX(ID_VEHICUL)
57 19                        FROM VEHICLE);
58 20
59 21    IF v_id_showroom IS NOT NULL THEN -- altfel este vehiculul unui client
60 22
61 23        UPDATE VEHICLE
62 24        SET PRET = PRET * 0.99
63 25        WHERE ID_SHOWROOM = v_id_showroom
64 26        AND ID_VEHICUL != (SELECT
65 27                          MAX(id_vehicul)
66 28                          FROM vehicle);
69 29
70 30    END IF;
71 31
72 32 END trigger_cerinta_10;
73 33 /
74
75 TRIGGER created.
76
77 Commit complete.
```

Poza 2/2:

```
912     v_id_showroom VEHICLE.ID_SHOWROOM%TYPE;
913
914 BEGIN
915     DBMS_OUTPUT.PUT_LINE(' ');
916     DBMS_OUTPUT.PUT_LINE('----- DECLANSARE TRIGGER CERINTA 10 -----');
917     DBMS_OUTPUT.PUT_LINE(' ');
918
919     SELECT
920         ID_SHOWROOM
921     INTO v_id_showroom
922     FROM VEHICLE
923     WHERE ID_VEHICUL = (SELECT
924                         MAX(ID_VEHICUL)
925                         FROM VEHICLE);
926
927     IF v_id_showroom IS NOT NULL THEN -- altfel este vehiculul unui client
928
929         UPDATE VEHICLE
930         SET PRET = PRET * 0.99
931         WHERE ID_SHOWROOM = v_id_showroom
932         AND ID_VEHICUL != (SELECT
933                          MAX(id_vehicul)
934                          FROM vehicle);
935
936     END IF;
937
938 END trigger_cerinta_10;
939 /
940
941 DROP TRIGGER trigger_cerinta_10;
942
943 -- EXEMPLU
944 SELECT ID_VEHICUL, PRET FROM VEHICLE WHERE ID_SHOWROOM = 1;
945
946 INSERT INTO VEHICLE (ID_VEHICUL, ID_MOTOR, ID_SHOWROOM, ID_CLIENT, MARCA, MODEL, AN_FABRICATIE, PRET)
947 VALUES (1000, 1, 1, NULL, 'TEST', 'TEST', 2024, 1000);
948
949 SELECT ID_VEHICUL, PRET FROM VEHICLE WHERE ID_SHOWROOM = 1;
950
951 ----- CERINTA 11 -----
952
953 --Definiti un trigger de tip DML la nivel de linie. Declansati trigger-ul.
954
955 -- CAND ADAUGI SAU STERGI O OPTIUNE A UNUI VEHICUL, ACTUALIZEAZA PRETUL CORESPUNZATOR
956
```

```
27
28     ID_VEHICUL
29     FROM vehicle);
30
31 END IF;
32
33 END trigger_cerinta_10;
34 /
35
36 TRIGGER created.
37
38 Commit complete.
39
40 SQL> -- EXEMPLU
41 SQL> SELECT ID_VEHICUL, PRET FROM VEHICLE WHERE ID_SHOWROOM = 1;
42
43 Max Rows: 500 Columns autosize: Cell contents: Save as: JSON
44
45 ID_VEHICUL PRET
46 1 15000
47 2 12000
48 3 15000
49 4 11 15000
50 5 12 12000
51 6 17 15000
52
53 SQL> INSERT INTO VEHICLE (ID_VEHICUL, ID_MOTOR, ID_SHOWROOM, ID_CLIENT, MARCA, MODEL, AN_FABRICATIE, PRET)
54 2 VALUES (1000, 1, 1, NULL, 'TEST', 'TEST', 2024, 1000);
55
56 ----- DECLANSARE TRIGGER CERINTA 10 -----
57
58 1 row created.
59
60 Commit complete.
61
62 SQL> SELECT ID_VEHICUL, PRET FROM VEHICLE WHERE ID_SHOWROOM = 1;
63
64 Max Rows: 500 Columns autosize: Cell contents: Save as: JSON
65
66 ID_VEHICUL PRET
67 1 14850
68 2 11880
69 3 7 14850
70 4 11 14850
71 5 12 11880
72 6 17 14850
73 7 1000 1000
```

## 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:  $PRET\_VEHICUL \pm PRET\_OPTIUNE$ .

### 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:

```
354 ----- CERINTA 11 -----
355
356 -- Definiți un trigger de tip UMD la nivel de linie. Declanșati trigger-ul.
357
358 -- CAND ADAUGI SAU STERGI O OPTIUNE A UNUI VEHICUL, ACTUALIZEAZA PRETUL CORESPUNZATOR
359 -- PRET_VEHICUL += PRET_OPTIUNE
360
361
362 CREATE OR REPLACE TRIGGER trigger_cerinta_11
363 AFTER INSERT OR DELETE ON vehicle_optiuni
364 FOR EACH ROW
365 DECLARE
366     v_pret_optiune OPTIUNI.PRET%TYPE;
367
368 BEGIN
369
370     DBMS_OUTPUT.PUT_LINE(' ');
371     DBMS_OUTPUT.PUT_LINE('----- DECLANSARE TRIGGER CERINTA 11 -----');
372     DBMS_OUTPUT.PUT_LINE(' ');
373
374     IF INSERTING THEN
375
376         SELECT
377             PRET
378         INTO v_pret_optiune
379         FROM OPTIUNI
380         WHERE ID_OPTIUNE = :NEW.ID_OPTIUNE;
381
382         UPDATE VEHICULE
383         SET PRET = PRET + v_pret_optiune
384         WHERE ID_VEHICUL = :NEW.ID_VEHICUL;
385
386     ELSIF DELETING THEN
387
388         SELECT
389             PRET
390         INTO v_pret_optiune
391         FROM OPTIUNI
392         WHERE ID_OPTIUNE = :OLD.ID_OPTIUNE;
393
394         UPDATE VEHICULE
395         SET PRET = PRET - v_pret_optiune
396         WHERE ID_VEHICUL = :OLD.ID_VEHICUL;
397
398     END IF;
399
400 END trigger_cerinta_11;
401 /
402
403 DROP TRIGGER trigger_cerinta_11;
404
405 -- REINITIALIZARE EXEMPLE
406 DELETE FROM VEHICULE_OPTIUNI WHERE ID_VEHICUL = 1000 AND ID_OPTIUNE = 100;
407 DELETE FROM VEHICULE WHERE ID_VEHICUL = 1000;
408 DELETE FROM OPTIUNI WHERE ID_OPTIUNE = 100;
409
410 -- ADAUGARE OPTIUNE LA UN VEHICUL
411 INSERT INTO VEHICULE (ID_VEHICUL, ID_MOTOR, ID_SHOWROOM, ID_CLIENT, MARCA, MODEL, AN_FABRICATIE, PRET)
412
```

Commit complete.

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

ID_VEHICUL	PRET
1	14850
2	11880
3	14850
4	14850
5	11880
6	14850
7	1000

```
SQL> CREATE OR REPLACE TRIGGER trigger_cerinta_11
2 AFTER INSERT OR DELETE ON vehicle_optiuni
3 FOR EACH ROW
4 DECLARE
5
6     v_pret_optiune OPTIUNI.PRET%TYPE;
7
8 BEGIN
9
10     DBMS_OUTPUT.PUT_LINE(' ');
11     DBMS_OUTPUT.PUT_LINE('----- DECLANSARE TRIGGER CERINTA 11 -----');
12     DBMS_OUTPUT.PUT_LINE(' ');
13
14     IF INSERTING THEN
15
16         SELECT
17             PRET
18         INTO v_pret_optiune
19         FROM OPTIUNI
20         WHERE ID_OPTIUNE = :NEW.ID_OPTIUNE;
21
22         UPDATE VEHICULE
23         SET PRET = PRET + v_pret_optiune
24         WHERE ID_VEHICUL = :NEW.ID_VEHICUL;
25
26     ELSIF DELETING THEN
27
28         SELECT
29             PRET
30         INTO v_pret_optiune
31         FROM OPTIUNI
32         WHERE ID_OPTIUNE = :OLD.ID_OPTIUNE;
33
34         UPDATE VEHICULE
35         SET PRET = PRET - v_pret_optiune
36         WHERE ID_VEHICUL = :OLD.ID_VEHICUL;
37
38     END IF;
39
40 END trigger_cerinta_11;
41 /
42
43 TRIGGER created.
44
45 Commit complete.
```

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

```
987 -- ELSEIF DELETING THEN
988 --
989 -- SELECT
990 --     PRET
991 -- INTO v_pret_optiune
992 -- FROM OPTIUNI
993 -- WHERE ID_OPTIUNE = :OLD.ID_OPTIUNE;
994 --
995 -- UPDATE vehicle
996 -- SET pret = pret - v_pret_optiune
997 -- WHERE id_vehicul = :OLD.id_vehicul;
998 --
999 -- END IF;
1000 --
1001 END trigger_cerinta_11;
1002 /
1003
1004 DROP TRIGGER trigger_cerinta_11;
1005
1006 -- REINITIALIZARE EXEMPLE
1007 DELETE FROM VEHICLE_OPTIUNI WHERE ID_VEHICUL = 1000 AND ID_OPTIUNE = 100;
1008 DELETE FROM VEHICLE WHERE ID_VEHICUL = 1000;
1009 DELETE FROM OPTIUNI WHERE ID_OPTIUNE = 100;
1010
1011 -- ADĂUGARE OPTIUNE LA UN VEHICUL
1012 INSERT INTO VEHICLE (ID_VEHICUL, ID_MOTOR, ID_SHOWROOM, ID_CLIENT, MARCA, MODEL, AN_FABRICATIE, PRET)
1013 VALUES (1000, 1, 1, NULL, 'TEST', 'TEST', 2024, 1000);
1014
1015 SELECT ID_VEHICUL, PRET FROM VEHICLE WHERE ID_VEHICUL = 1000;
1016
1017 INSERT INTO OPTIUNI (ID_OPTIUNE, NUME, PRET) VALUES (100, 'OPTIUNE TEST', 100);
1018
1019 SELECT * FROM OPTIUNI WHERE ID_OPTIUNE = 100;
1020
1021 INSERT INTO VEHICLE_OPTIUNI (ID_VEHICUL, ID_OPTIUNE) VALUES (1000, 100);
1022
1023 SELECT * FROM VEHICLE_OPTIUNI WHERE ID_VEHICUL = 1000 AND ID_OPTIUNE = 100;
1024
1025 SELECT ID_VEHICUL, PRET FROM VEHICLE WHERE ID_VEHICUL = 1000;
1026
1027 -- STERGERE OPTIUNE A UNUI VEHICUL
1028 DELETE FROM VEHICLE_OPTIUNI WHERE ID_VEHICUL = 1000 AND ID_OPTIUNE = 100;
1029
1030 SELECT ID_VEHICUL, PRET FROM VEHICLE WHERE ID_VEHICUL = 1000;
1031
1032 ----- CERINTA 12 -----
1033
1034 -- Definiți un trigger de tip DDL. Declarați trigger-ul.
1035
1036 -- CÂND SE DA DROP LA TABELUL VEHICULE, SE VA ARIUNCA O EXCEPȚIE SI NU SE VA STERGE TABELUL
1037
1038 CREATE OR REPLACE TRIGGER trigger_cerinta_12
1039 BEFORE DROP ON SCHEMA
1040 DECLARE
```

SQL> -- ADĂUGARE OPTIUNE LA UN VEHICUL  
SQL> INSERT INTO VEHICLE (ID\_VEHICUL, ID\_MOTOR, ID\_SHOWROOM, ID\_CLIENT, MARCA, MODEL, AN\_FABRICATIE, PRET)  
2 VALUES (1000, 1, 1, NULL, 'TEST', 'TEST', 2024, 1000);

----- DECLANSARE TRIGGER CERINTA 10 -----

1 row created.

Commit complete.

SQL> SELECT ID\_VEHICUL, PRET FROM VEHICLE WHERE ID\_VEHICUL = 1000;

ID_VEHICUL	PRET
1 1000	1000

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

1 row created.

Commit complete.

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

ID_OPTIUNE	NUME	PRET
1 100	OPTIUNE TEST	100

SQL> INSERT INTO VEHICLE\_OPTIUNI (ID\_VEHICUL, ID\_OPTIUNE) VALUES (1000, 100);

----- DECLANSARE TRIGGER CERINTA 11 -----

1 row created.

Commit complete.

SQL> SELECT \* FROM VEHICLE\_OPTIUNI WHERE ID\_VEHICUL = 1000 AND ID\_OPTIUNE = 100;

ID_VEHICUL	ID_OPTIUNE
1 1000	100

SQL> SELECT ID\_VEHICUL, PRET FROM VEHICLE WHERE ID\_VEHICUL = 1000;

ID_VEHICUL	PRET
1 1000	1100

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

```
2 INSERT INTO VEHICLE (ID_VEHICUL, ID_MOTOR, ID_SHOWROOM, ID_CLIENT, MARCA, MODEL, AN_FABRICATIE, PRET)
3 VALUES (1000, 1, 1, NULL, 'TEST', 'TEST', 2024, 1000);
4
5 SELECT ID_VEHICUL, PRET FROM VEHICLE WHERE ID_VEHICUL = 1000;
6
7 INSERT INTO OPTIUNI (ID_OPTIUNE, NUME, PRET) VALUES (100, 'OPTIUNE TEST', 100);
8
9 SELECT * FROM OPTIUNI WHERE ID_OPTIUNE = 100;
10
11 INSERT INTO VEHICLE_OPTIUNI (ID_VEHICUL, ID_OPTIUNE) VALUES (1000, 100);
12
13 SELECT * FROM VEHICLE_OPTIUNI WHERE ID_VEHICUL = 1000 AND ID_OPTIUNE = 100;
14
15 SELECT ID_VEHICUL, PRET FROM VEHICLE WHERE ID_VEHICUL = 1000;
16
17 -- STERGERE OPTIUNE A UNUI VEHICUL
18 DELETE FROM VEHICLE_OPTIUNI WHERE ID_VEHICUL = 1000 AND ID_OPTIUNE = 100;
19
20 SELECT ID_VEHICUL, PRET FROM VEHICLE WHERE ID_VEHICUL = 1000;
21
22 ----- CERINTA 12 -----
23
24 -- Definiți un trigger de tip DDL. Declarați trigger-ul.
```

SQL> SELECT ID\_VEHICUL, PRET FROM VEHICLE WHERE ID\_VEHICUL = 1000;

ID_VEHICUL	PRET
1 1000	1100

SQL> -- STERGERE OPTIUNE A UNUI VEHICUL  
SQL> DELETE FROM VEHICLE\_OPTIUNI WHERE ID\_VEHICUL = 1000 AND ID\_OPTIUNE = 100;

----- DECLANSARE TRIGGER CERINTA 11 -----

1 row deleted.

Commit complete.

SQL> SELECT ID\_VEHICUL, PRET FROM VEHICLE WHERE ID\_VEHICUL = 1000;

ID_VEHICUL	PRET
1 1000	1000

## 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:

```
1040
1041 CREATE OR REPLACE TRIGGER trigger_cerinta_12
1042 BEFORE DROP ON SCHEMA
1043 DECLARE
1044
1045     e_stergere_tabel_vehicule EXCEPTION;
1046
1047 BEGIN
1048
1049     IF UPPER(DICTIONARY_OBJ_NAME) = 'VEHICULE' THEN
1050         RAISE e_stergere_tabel_vehicule;
1051     END IF;
1052
1053 EXCEPTION
1054 WHEN e_stergere_tabel_vehicule THEN
1055     RAISE_APPLICATION_ERROR(-20999, 'TABELUL <<VEHICULE>> ESTE UN TABEL DEFINITIV');
1056
1057 END trigger_cerinta_12;
1058
1059
1060 DROP TRIGGER trigger_cerinta_12;
1061
1062 SELECT * FROM VEHICLE;
1063
1064 DROP TABLE VEHICLE CASCADE CONSTRAINTS;
1065
1066 SELECT * FROM VEHICLE;
1067
```

SQL> SELECT ID\_VEHICUL, PRET FROM VEHICLE WHERE ID\_VEHICUL = 1000;

ID_VEHICUL	PRET
1 1000	1000

```
SQL> CREATE OR REPLACE TRIGGER trigger_cerinta_12
2 BEFORE DROP ON SCHEMA
3 DECLARE
4
5     e_stergere_tabel_vehicule EXCEPTION;
6
7 BEGIN
8
9     IF UPPER(DICTIONARY_OBJ_NAME) = 'VEHICULE' THEN
10         RAISE e_stergere_tabel_vehicule;
11     END IF;
12
13 EXCEPTION
14 WHEN e_stergere_tabel_vehicule THEN
15     RAISE_APPLICATION_ERROR(-20999, 'TABELUL <<VEHICULE>> ESTE UN TABEL DEFINITIV');
16
17 END trigger_cerinta_12;
18 /

TRIGGER created.

Commit complete.
```

Poza 2/2:

```
1044
1045     e_stergere_tabel_vehicule EXCEPTION;
1046
1047 BEGIN
1048
1049     IF UPPER(DICTIONARY_OBJ_NAME) = 'VEHICULE' THEN
1050         RAISE e_stergere_tabel_vehicule;
1051     END IF;
1052
1053 EXCEPTION
1054 WHEN e_stergere_tabel_vehicule THEN
1055     RAISE_APPLICATION_ERROR(-20999, 'TABELUL <<VEHICULE>> ESTE UN TABEL DEFINITIV');
1056
1057 END trigger_cerinta_12;
1058
1059 /
1060
1061 DROP TRIGGER trigger_cerinta_12;
1062
1063 SELECT * FROM VEHICLE;
1064
1065 DROP TABLE VEHICLE CASCADE CONSTRAINTS;
1066
1067 SELECT * FROM VEHICLE;
```

SQL> SELECT \* FROM VEHICLE;

ID_VEHICUL	ID_MOTOR	ID_SHOWROOM	ID_CLIENT	MARCA	MODEL
1 1	1	1		Dacia	Duster
2 2	2	1		Dacia	Sandero
3 3	3	2		Dacia	Logan
4 4	4	3		Dacia	Loggy
5 5	5	4		Dacia	Dokker
6 6	6	5		Dacia	Spring
7 7	7	1		Dacia	Duster
8 8	8	2		Dacia	Sandero
9 9	9	3		Dacia	Logan
10 10	10	4		Dacia	Loggy

```
SQL> DROP TABLE VEHICLE CASCADE CONSTRAINTS;

DROP TABLE VEHICLE CASCADE CONSTRAINTS
*
ERROR at line 1064:
ORA-04088: error during execution of trigger 'AN08E14.TRIGGER_CERINTA_12'
ORA-04088: error occurred at recursive SQL level 1
ORA-20999: TABELUL <<VEHICULE>> ESTE UN TABEL DEFINITIV
ORA-04088: #1 line 13
https://docs.oracle.com/error-help/db/ora-04088/
```