FACULTATEA DE MATEMATICA SI INFORMATICA

UNIVERSITATEA BUCURESTI

REALIZAT DE OPRAN ANDREI  
GRUPA 142

bAZA DE DATE

DEALER AUTO

CUPRINS

[**1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.** 5](#_Toc167628155)

[ Descrierea modelului real 5](#_Toc167628156)

[ Utilitatea 5](#_Toc167628157)

[ Reguli de funcționare 5](#_Toc167628158)

[**2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.** 6](#_Toc167628159)

[ Tabel SHOWROOM 6](#_Toc167628160)

[ Tabel SHOWROOM\_NUMERE\_TELEFON 6](#_Toc167628161)

[ Tabel ADRESA 6](#_Toc167628162)

[ Tabel ANGAJAT\_SHOWROOM 6](#_Toc167628163)

[ Tabel CARBURANT 7](#_Toc167628164)

[ Tabel CAROSERIE 7](#_Toc167628165)

[ Tabel CLIENT 7](#_Toc167628166)

[ Tabel CLIENT\_NUMERE\_TELEFON 7](#_Toc167628167)

[ Tabel MECANIC 7](#_Toc167628168)

[ Tabel MOTOR 8](#_Toc167628169)

[ Tabel MOTOR\_CARBURANT 8](#_Toc167628170)

[ Tabel MOTOR\_ELECTRIC 8](#_Toc167628171)

[ Tabel PLATA 8](#_Toc167628172)

[ Tabel SERVICE 9](#_Toc167628173)

[ Tabel TRANZACTIE 9](#_Toc167628174)

[ Tabel VANZATOR 9](#_Toc167628175)

[ Tabel VEHICUL 9](#_Toc167628176)

[ Tabel VEHICULE\_DETINUTE\_CLIENTI 10](#_Toc167628177)

[**3. Descrierea entităților, incluzând precizarea cheii primare** 10](#_Toc167628178)

[ Entitatea SHOWROOM 10](#_Toc167628179)

[ Entitatea SHOWROOM\_NUMERE\_TELEFON 10](#_Toc167628180)

[ Entitatea ADRESA 11](#_Toc167628181)

[ Entitatea ANGAJAT\_SHOWROOM 11](#_Toc167628182)

[ Entitatea CARBURANT 11](#_Toc167628183)

[ Entitatea CAROSERIE 12](#_Toc167628184)

[ Entitatea CLIENT 12](#_Toc167628185)

[ Entitatea CLIENT\_NUMERE\_TELEFON 12](#_Toc167628186)

[ Entitatea MECANIC 13](#_Toc167628187)

[ Entitatea MOTOR 13](#_Toc167628188)

[ Entitatea MOTOR\_CARBURANT 13](#_Toc167628189)

[ Entitatea MOTOR\_ELECTRIC 13](#_Toc167628190)

[ Entitatea PLATA 14](#_Toc167628191)

[ Entitatea SERVICE 14](#_Toc167628192)

[ Entitatea TRANZACTIE 14](#_Toc167628193)

[ Entitatea VANZATOR 15](#_Toc167628194)

[ Entitatea VEHICUL 15](#_Toc167628195)

[ Entitatea VEHICUL\_DETINUT\_CLIENT 15](#_Toc167628196)

[**4. Descrierea relațiilor, incluzând precizarea cardinalității acestora** 16](#_Toc167628197)

[ MOTOR\_CARBURANT (ID\_MOTOR) - MOTOR (ID\_MOTOR) 16](#_Toc167628198)

[ MOTOR\_CARBURANT (ID\_CARBURANT) - CARBURANT (ID\_CARBURANT) 16](#_Toc167628199)

[ MOTOR\_ELECTRIC (ID\_MOTOR) - MOTOR (ID\_MOTOR) 17](#_Toc167628200)

[ SHOWROOM (ID\_ADRESA) - ADRESA (ID\_ADRESA) 17](#_Toc167628201)

[ SHOWROOM\_NR\_TELEFON (ID\_SHOWROOM\_NR\_TEL) - SHOWROOM (ID\_SHOWROOM) 17](#_Toc167628202)

[ VEHICUL (ID\_SHOWROOM) - SHOWROOM (ID\_SHOWROOM) 17](#_Toc167628203)

[ VEHICUL (ID\_CAROSERIE) - CAROSERIE (ID\_CAROSERIE) 17](#_Toc167628204)

[ VEHICUL (ID\_MOTOR) - MOTOR (ID\_MOTOR) 18](#_Toc167628205)

[ ANGAJAT\_SHOWROOM (ID\_SHOWROOM) - SHOWROOM (ID\_SHOWROOM) 18](#_Toc167628206)

[ MECANIC (ID\_ANGAJAT) - ANGAJAT\_SHOWROOM (ID\_ANGAJAT) 18](#_Toc167628207)

[ MECANIC (ID\_SPECIALIZARE) - MOTOR (ID\_MOTOR) 18](#_Toc167628208)

[ VANZATOR (ID\_ANGAJAT) - ANGAJAT\_SHOWROOM (ID\_ANGAJAT) 18](#_Toc167628209)

[ VANZATOR (ID\_SPECIALIZARE) - CAROSERIE (ID\_CAROSERIE) 19](#_Toc167628210)

[ CLIENT\_NUMERE\_TELEFON (ID\_CLIENT\_NR\_TELEFON) - CLIENT (ID\_CLIENT) 19](#_Toc167628211)

[ VEHICUL\_DETINUT\_CLIENT (ID\_CLIENT) - CLIENT (ID\_CLIENT) 19](#_Toc167628212)

[ VEHICUL\_DETINUT\_CLIENT (ID\_CAROSERIE) - CAROSERIE (ID\_CAROSERIE) 19](#_Toc167628213)

[ VEHICUL\_DETINUT\_CLIENT (ID\_MOTOR) - MOTOR (ID\_MOTOR) 19](#_Toc167628214)

[ PLATA (ID\_CLIENT) - CLIENT (ID\_CLIENT) 19](#_Toc167628215)

[ TRANZACTIE (ID\_CLIENT) - CLIENT (ID\_CLIENT) 20](#_Toc167628216)

[ TRANZACTIE (ID\_PLATA) - PLATA (ID\_PLATA) 20](#_Toc167628217)

[ TRANZACTIE (ID\_VEHICUL) - VEHICUL\_DETINUT\_CLIENT (ID\_VEHICUL) 20](#_Toc167628218)

[ TRANZACTIE (ID\_ANGAJAT) - VANZATOR (ID\_ANGAJAT) 20](#_Toc167628219)

[ SERVICE (ID\_ANGAJAT) - MECANIC (ID\_ANGAJAT) 20](#_Toc167628220)

[ SERVICE (ID\_VEHICUL) - VEHICUL\_DETINUT\_CLIENT (ID\_VEHICUL) 21](#_Toc167628221)

[ SERVICE (ID\_PLATA) - PLATA (ID\_PLATA) 21](#_Toc167628222)

[**5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.** 21](#_Toc167628223)

[ Tabel MOTOR 21](#_Toc167628224)

[ Tabel CARBURANT 21](#_Toc167628225)

[ Tabel MOTOR\_CARBURANT 22](#_Toc167628226)

[ Tabel MOTOR\_ELECTRIC 22](#_Toc167628227)

[ Tabel CAROSERIE 23](#_Toc167628228)

[ Tabel ADRESA 23](#_Toc167628229)

[ Tabel SHOWROOM 23](#_Toc167628230)

[ Tabel SHOWROOM\_NUMERE\_TELEFON 24](#_Toc167628231)

[ Tabel VEHICUL 24](#_Toc167628232)

[ Tabel ANGAJAT\_SHOWROOM 24](#_Toc167628233)

[ Tabel MECANIC 25](#_Toc167628234)

[ Tabel VANZATOR 25](#_Toc167628235)

[ Tabel CLIENT 25](#_Toc167628236)

[ Tabel CLIENT\_NUMERE\_TELEFON 25](#_Toc167628237)

[ Tabel VEHICUL\_DETINUT\_CLIENT 26](#_Toc167628238)

[ Tabel PLATA 26](#_Toc167628239)

[ Tabel TRANZACTIE 26](#_Toc167628240)

[ Tabel SERVICE 27](#_Toc167628241)

[**6. Realizarea diagramei entitate-relație.** 28](#_Toc167628242)

[**7. Realizarea diagramei conceptuale.** 29](#_Toc167628243)

[**8**. **Enumerarea schemelor relaționale.** 31](#_Toc167628244)

[**9. Realizarea normalizării până la forma normală 3 (FN1-FN3).** 32](#_Toc167628245)

[**10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).** 35](#_Toc167628246)

[**11. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea.** 37](#_Toc167628247)

[ Tabel MOTOR 37](#_Toc167628248)

[ Tabel CARBURANT 40](#_Toc167628249)

[ Tabel MOTOR\_CARBURANT 42](#_Toc167628250)

[ Tabel MOTOR\_ELECTRIC 44](#_Toc167628251)

[ Tabel CAROSERIE 46](#_Toc167628252)

[ Tabel ADRESA 48](#_Toc167628253)

[ Tabel SHOWROOM 50](#_Toc167628254)

[ Tabel SHOWROOM\_NUMERE\_TELEFON 52](#_Toc167628255)

[ Tabel VEHICUL 54](#_Toc167628256)

[ Tabel ANGAJAT\_SHOWROOM 57](#_Toc167628257)

[ Tabel MECANIC 60](#_Toc167628258)

[ Tabel VANZATOR 62](#_Toc167628259)

[ Tabel CLIENT 64](#_Toc167628260)

[ Tabel CLIENT\_NUMERE\_TELEFON 66](#_Toc167628261)

[ Tabel VEHICUL\_DETINUT\_CLIENT 68](#_Toc167628262)

[ Tabel PLATA 70](#_Toc167628263)

[ Tabel TRANZACTIE 73](#_Toc167628264)

[ Tabel SERVICE 75](#_Toc167628265)

[**12. Formulați în limbaj natural și implementați 5 cereri SQL complexe** 77](#_Toc167628266)

[ Cererea 1 78](#_Toc167628267)

[ Cererea 2 79](#_Toc167628268)

[ Cererea 3 80](#_Toc167628269)

[ Cererea 4 81](#_Toc167628270)

[ Cererea 5 82](#_Toc167628271)

[**13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.** 84](#_Toc167628272)

[ Operația 1 84](#_Toc167628273)

[ Operația 2 86](#_Toc167628274)

[ Operația 3 88](#_Toc167628275)

# **1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.**

## Descrierea modelului real

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

## Utilitatea

Baza de date poate stoca informații despre showroom-uri si angajații acestora (vânzători sau mecanici), vehiculele dintr-un anumit showroom cu specificații tehnice detaliate legate de caroserie si motor, cel din urma putând fi de tipul carburant sau electric.

De asemenea, se pot stoca informații despre clienți, vehiculele clienților, plățile clienților, tranzacțiile de vânzare ale unui vehicul si fise de service.

## Reguli de funcționare

Baza de date poate fi văzută ca o compunere dintre partea de clienți si partea de showroom, legătura făcând-se prin tabelele TRANZACTIE si SERVICE. O operație consta in mutarea unui vehicul din tabelul VEHICUL in VEHICUL\_DETINUT\_CLIENT, generând in același timp înregistrări corespunzătoare pentru TRANZACTIE si PLATA..

Pentru păstrarea corecta a datelor, la atribuirea unui vehicul in tabelul VEHICUL\_DETINUT\_CLIENT, se impune ca o noua înregistrare corecta sa fie atribuita tabelelor TRANZACTIE, PLATA si sa se scoată înregistrarea vehiculului respectiv din tabelul VEHICUL.

In plus, la adăugarea unui vehicul in tabelul VEHICUL\_DETINUT\_CLIENT, in înregistrarea din tabelul TRANZACTIE, ID\_ANGAJAT trebuie sa corespunda unui VANZATOR care are un ID\_SPECIALIZARE egal cu ID\_CAROSERIE al mașinii adăugate.

# **2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.**

## Tabel SHOWROOM

* NUME = un sir de maxim 30 de caractere.
* ID\_ADRESA = un număr unic la nivel de tabel, cu rol de cheie străină pentru tabelul ADRESA.
* ID\_SHOWROOM = un număr unic la nivel de tabel, cu rolul de cheie primara.

## Tabel SHOWROOM\_NUMERE\_TELEFON

* ID\_NUMAR = un număr unic la nivel de tabel, cu rolul de cheie primară.
* ID\_SHOWROOM\_NUMAR\_TELEFON = un număr unic la nivel de tabel, cu rol de cheie străină pentru tabelul SHOWROOM.
* NR\_TELEFON = un șir de caractere care respectă formatul unui număr de telefon (începe cu 07, urmat de 2 cifre, un spațiu, 3 cifre, un spațiu, și încă 3 cifre).

## Tabel ADRESA

* STRADA = un sir de maxim 30 de caractere.
* NR = un număr strict pozitiv , numărul străzii.
* ORAS = un sir de maxim 30 de caractere .
* ID\_ADRESA = un număr unic la nivel de tabel, cu rolul de cheie primara, face legătura cu tabelul SHOWROOM.

## Tabel ANGAJAT\_SHOWROOM

* ID\_SHOWROOM = un număr unic la nivel de tabel, cu rol de cheie străină pentru tabelul SHOWROOM.
* NUME = un sir de maxim 30 de caractere.
* DATA\_ANGAJARE = valoare de tipul data.
* SALARIU = un număr de maxim 10 cifre si cu 2 zecimale, strict pozitiv.
* ID\_ANGAJAT = = un număr unic la nivel de tabel, cu rolul de cheie primara, face legătura cu tabelele VANZATOR si MECANIC.

## Tabel CARBURANT

* ID\_CARBURANT = un număr unic la nivel de tabel, cu rolul de cheie primară.
* TIP = un șir de caractere care poate fi fie Benzina, fie Motorina.
* VARIANTA = un șir de caractere care descrie varianta specifică a tipului de carburant.
* PRET\_LITRU = un număr care reprezintă prețul pe litru al carburantului. Acesta trebuie să fie mai mare decât 0.

## Tabel CAROSERIE

* ID\_CAROSERIE = un număr unic la nivel de tabel, cu rolul de cheie primară.
* NUME = un șir de maxim 30 de caractere care descrie numele caroseriei.
* NR\_USI = un număr care reprezintă numărul de uși al caroseriei. Acesta trebuie să fie mai mare decât 0.
* NR\_LOCURI = un număr care reprezintă numărul de locuri al caroseriei. Acesta trebuie să fie mai mare decât 0.
* VOLUM\_PORTBAGAJ\_LITRI = un număr care reprezintă volumul portbagajului în litri. Acesta trebuie să fie mai mare decât 0.

## Tabel CLIENT

* ID\_CLIENT = un număr unic la nivel de tabel, cu rolul de cheie primară.
* NUME = un șir de maxim 30 de caractere care reprezintă numele clientului.

## Tabel CLIENT\_NUMERE\_TELEFON

* ID\_NUMAR = un număr unic la nivel de tabel, cu rolul de cheie primară.
* ID\_CLIENT \_NR\_TELEFON = un număr unic la nivel de tabel, cu rol de cheie străină pentru tabelul CLIENT.
* NR\_TELEFON = un șir de caractere care respectă formatul unui număr de telefon (începe cu 07, urmat de 2 cifre, un spațiu, 3 cifre, un spațiu, și încă 3 cifre).

## Tabel MECANIC

* ID\_ANGAJAT = un număr unic la nivel de tabel, cu rolul de cheie primară. Acesta face referire la ID-ul unui angajat din tabelul ANGAJAT\_SHOWROOM.
* ID\_SPECIALIZARE = un număr care face referire la ID-ul unui motor din tabelul MOTOR. Acesta reprezintă specializarea mecanicului.

## Tabel MOTOR

* ID\_MOTOR = un număr unic la nivel de tabel, cu rolul de cheie primară.
* MARCA = un șir de maxim 30 de caractere care reprezintă marca motorului.
* MODEL = un șir de maxim 30 de caractere care reprezintă modelul motorului.

## Tabel MOTOR\_CARBURANT

* ID\_MOTOR = un număr unic la nivel de tabel, cu rolul de cheie primară. Acesta face referire la ID-ul unui motor din tabelul MOTOR.
* ID\_CARBURANT = un număr care face referire la ID-ul unui tip de carburant din tabelul CARBURANT.
* CAI\_PUTERE = numărul de cai putere ai motorului. Valoarea trebuie să fie mai mare decât 0.
* CAPACITATE\_CILINDRICA\_LITRI = capacitatea cilindrică a motorului, exprimată în litri. Valoarea trebuie să fie mai mare decât 0.
* CONSUM\_MEDIU\_LITRI = consumul mediu de carburant al motorului, exprimat în litri. Valoarea trebuie să fie mai mare decât 0.
* ASPIRATIE = tipul de aspirație al motorului. Poate fi fie Aspirat, fie Turbina.

## Tabel MOTOR\_ELECTRIC

* ID\_MOTOR = un număr unic la nivel de tabel, cu rolul de cheie primară. Acesta face referire la ID-ul unui motor din tabelul MOTOR.
* CAI\_PUTERE = numărul de cai putere ai motorului electric. Valoarea trebuie să fie mai mare decât 0.
* TIMP\_INCARCARE\_ORE = timpul necesar pentru încărcarea completă a motorului electric, exprimat în ore. Valoarea trebuie să fie mai mare decât 0.
* AUTONOMIE\_KM = autonomia motorului electric, exprimată în kilometri. Valoarea trebuie să fie mai mare decât 0.

## Tabel PLATA

* ID\_PLATA = un număr unic la nivel de tabel, cu rolul de cheie primară.
* ID\_CLIENT = un număr care face referire la ID-ul unui client din tabelul CLIENT.
* SUMA = suma de bani implicată într-o tranzacție. Valoarea trebuie să fie mai mare decât 0.

## Tabel SERVICE

* ID\_ANGAJAT = un număr care face referire la ID-ul unui angajat din tabelul MECANIC.
* ID\_VEHICUL = un număr care face referire la ID-ul unui vehicul din tabelul VEHICUL\_DETINUT\_CLIENT.
* ID\_PLATA = un număr care face referire la ID-ul unei plăți din tabelul PLATA.
* SUMA = suma de bani implicată într-o tranzacție de service.

## Tabel TRANZACTIE

* ID\_TRANZACTIE = un număr unic la nivel de tabel, cu rolul de cheie primară.
* ID\_CLIENT = un număr care face referire la ID-ul unui client din tabelul CLIENT.
* ID\_PLATA = un număr care face referire la ID-ul unei plăți din tabelul PLATA.
* ID\_VEHICUL = un număr care face referire la ID-ul unui vehicul din tabelul VEHICUL\_DETINUT\_CLIENT.
* ID\_ANGAJAT = un număr care face referire la ID-ul unui angajat din tabelul VANZATOR.
* DATA\_TRANZACTIE = data la care a avut loc tranzacția. Dacă nu este furnizată, va fi folosită data curentă.
* SUMA = suma de bani implicată în tranzacție. Valoarea trebuie să fie mai mare decât 0.

## Tabel VANZATOR

* ID\_ANGAJAT = un număr unic la nivel de tabel, cu rolul de cheie primară. Acesta face referire la ID-ul unui angajat din tabelul ANGAJAT\_SHOWROOM.
* ID\_SPECIALIZARE = un număr care face referire la ID-ul unei specializări din tabelul CAROSERIE.

## Tabel VEHICUL

* ID\_VEHICUL = un număr unic la nivel de tabel, cu rolul de cheie primară.
* ID\_SHOWROOM = un număr care face referire la ID-ul unui showroom din tabelul SHOWROOM.
* ID\_CAROSERIE = un număr care face referire la ID-ul unei caroserii din tabelul CAROSERIE.
* ID\_MOTOR = un număr care face referire la ID-ul unui motor din tabelul MOTOR.
* MARCA = marca vehiculului.
* MODEL = modelul vehiculului.
* AN\_FABRICATIE = anul în care a fost fabricat vehiculul. Valoarea trebuie să fie mai mare decât 0.
* PRET = prețul vehiculului. Valoarea trebuie să fie mai mare decât 0.

## Tabel VEHICULE\_DETINUTE\_CLIENTI

* ID\_VEHICUL = un număr unic la nivel de tabel, cu rolul de cheie primară.
* ID\_CLIENT = un număr care face referire la ID-ul unui client din tabelul CLIENT.
* ID\_CAROSERIE = un număr care face referire la ID-ul unei caroserii din tabelul CAROSERIE.
* ID\_MOTOR = un număr care face referire la ID-ul unui motor din tabelul MOTOR.
* MARCA = marca vehiculului.
* MODEL = modelul vehiculului.
* AN\_FABRICATIE = anul în care a fost fabricat vehiculul. Valoarea trebuie să fie mai mare decât 0.
* NUMAR\_INMATRICULARE = numărul de înmatriculare al vehiculului. Acesta trebuie să fie unic și să se potrivească cu formatul de număr de înmatriculare din Romania.

# **3. Descrierea entităților, incluzând precizarea cheii primare**

## Entitatea SHOWROOM

Tabelul SHOWROOM reprezintă o entitate din baza de date care stochează informații despre diverse showroom-uri. Acest tabel are trei coloane: ID\_SHOWROOM, care este cheia primară și identifică în mod unic fiecare showroom; NUME, care conține denumirea showroom-ului și are o limită de 30 de caractere; și ID\_ADRESA, care este o cheie externă referitoare la tabela ADRESA și indică adresa showroom-ului. Această structură asigură că fiecare showroom are o identificare unică, un nume și o legătură către o adresă specifică, garantând astfel integritatea datelor și relația corectă între showroom-uri și adresele lor.

## Entitatea SHOWROOM\_NUMERE\_TELEFON

Tabelul SHOWROOM\_NUMERE\_TELEFON reprezintă o entitate din baza de date care stochează informații despre numerele de telefon asociate cu diverse showroom-uri. Acest tabel are patru coloane: ID\_NUMAR, care este cheia primară și identifică în mod unic fiecare înregistrare; ID\_SHOWROOM\_NUMAR\_TELEFON, care este o cheie externă referitoare la tabela SHOWROOM și indică showroom-ul căruia îi aparține numărul de telefon; NR\_TELEFON, care conține numărul de telefon în formatul specificat; și o constrângere CHECK care asigură că numerele de telefon respectă formatul românesc standard, început cu 07 urmat de două cifre, un spațiu, trei cifre, un alt spațiu și alte trei cifre. Această structură asigură integritatea datelor și corelația corectă între numerele de telefon și showroom-uri.

## Entitatea ADRESA

Tabelul ADRESA reprezintă o entitate din baza de date care stochează informații despre adrese. Acest tabel are patru coloane: ID\_ADRESA, care este cheia primară și identifică în mod unic fiecare adresă; STRADA, care conține numele străzii și are o limită de 30 de caractere; NR, care conține numărul clădirii și trebuie să fie un număr pozitiv, verificat printr-o constrângere CHECK; și ORAS, care conține numele orașului și are o limită de 30 de caractere. Această structură asigură că fiecare adresă are o identificare unică, o stradă, un număr de clădire valid și un oraș, garantând astfel integritatea și corectitudinea datelor referitoare la adrese.

## Entitatea ANGAJAT\_SHOWROOM

Tabelul ANGAJAT\_SHOWROOM reprezintă o entitate din baza de date care stochează informații despre angajații showroom-urilor. Acest tabel are cinci coloane: ID\_ANGAJAT, care este cheia primară și identifică în mod unic fiecare angajat; ID\_SHOWROOM, care este o cheie externă referitoare la tabela SHOWROOM și indică showroom-ul unde lucrează angajatul; NUME, care conține numele angajatului și are o limită de 30 de caractere; DATA\_ANGAJARE, care înregistrează data angajării și are o valoare implicită setată la data curentă (SYSDATE); și SALARIU, care reprezintă salariul angajatului și este un număr cu până la 10 cifre și două zecimale, verificat printr-o constrângere CHECK pentru a asigura că este un număr pozitiv. Această structură asigură integritatea datelor și corelația corectă între angajați și showroom-urile în care aceștia lucrează.

## Entitatea CARBURANT

Tabelul CARBURANT reprezintă o entitate din baza de date care stochează informații despre tipurile de combustibil. Acest tabel are patru coloane: ID\_CARBURANT, care este cheia primară și identifică în mod unic fiecare tip de combustibil; TIP, care conține tipul de combustibil și are o limită de 30 de caractere, verificat printr-o constrângere CHECK pentru a asigura că valorile sunt fie Benzina, fie Motorina; VARIANTA, care conține varianta specifică a combustibilului și are o limită de 30 de caractere; și PRET\_LITRU, care reprezintă prețul pe litru al combustibilului, verificat printr-o constrângere CHECK pentru a asigura că este un număr pozitiv cu până la 3 cifre și 2 zecimale. Această structură asigură integritatea datelor și corectitudinea informațiilor despre tipurile de combustibil disponibile.

## Entitatea CAROSERIE

Tabelul CAROSERIE reprezintă o entitate din baza de date care stochează informații despre tipurile de caroserii ale vehiculelor. Acest tabel are cinci coloane: ID\_CAROSERIE, care este cheia primară și identifică în mod unic fiecare tip de caroserie; NUME, care conține denumirea caroseriei și are o limită de 30 de caractere; NR\_USI, care specifică numărul de uși și este verificat printr-o constrângere CHECK pentru a asigura că valoarea este pozitivă; NR\_LOCURI, care indică numărul de locuri și este verificat printr-o constrângere CHECK pentru a asigura că valoarea este pozitivă; și VOLUM\_PORTBAGAJ\_LITRI, care reprezintă volumul portbagajului în litri, verificat printr-o constrângere CHECK pentru a asigura că este un număr pozitiv. Această structură asigură integritatea datelor și corectitudinea informațiilor despre caracteristicile caroseriei vehiculelor.

## Entitatea CLIENT

Tabelul CLIENT reprezintă o entitate din baza de date care stochează informații despre clienți. Acest tabel are două coloane: ID\_CLIENT, care este cheia primară și identifică în mod unic fiecare client; și NUME, care conține numele clientului și are o limită de 30 de caractere. Această structură permite înregistrarea și gestionarea datelor despre clienți în mod eficient și asigură integritatea informațiilor despre aceștia.

## Entitatea CLIENT\_NUMERE\_TELEFON

Tabelul CLIENT\_NUMERE\_TELEFON reprezintă o entitate din baza de date care stochează informații despre numerele de telefon ale clienților. Acest tabel are trei coloane: ID\_NUMAR, care este cheia primară și identifică în mod unic fiecare înregistrare de număr de telefon; ID\_CLIENT\_NR\_TELEFON, care este o cheie externă referitoare la tabela CLIENT și indică clientul căruia îi aparține numărul de telefon; și NR\_TELEFON, care conține numărul de telefon al clientului și trebuie să respecte formatul standard românesc, verificat printr-o constrângere CHECK. Această structură permite înregistrarea și gestionarea informațiilor despre numerele de telefon ale clienților, asigurând integritatea și corectitudinea datelor.

## Entitatea MECANIC

Tabelul MECANIC reprezintă o entitate din baza de date care stochează informații despre mecanicii angajați în showroom-uri. Acest tabel are trei coloane: ID\_ANGAJAT, care este cheia primară și identifică în mod unic fiecare mecanic, fiind totodată și o cheie externă care face referire la tabelul ANGAJAT\_SHOWROOM; ID\_SPECIALIZARE, care este o cheie externă referitoare la tabelul MOTOR și indică specializarea mecanicului. Această structură permite asocierea între mecanici și showroom-urile unde lucrează, precum și specializările lor în domeniul motoarelor.

## Entitatea MOTOR

Tabelul MOTOR reprezintă o entitate din baza de date care stochează informații despre motoare. Acest tabel are trei coloane: ID\_MOTOR, care este cheia primară și identifică în mod unic fiecare motor; MARCA, care conține marca motorului și are o limită de 30 de caractere; și MODEL, care conține modelul motorului și are o limită de 30 de caractere. Această structură permite înregistrarea și gestionarea datelor despre diversele tipuri de motoare într-un mod organizat și eficient.

## Entitatea MOTOR\_CARBURANT

Tabelul MOTOR\_CARBURANT reprezintă o entitate din baza de date care stochează informații despre motoarele și tipurile de combustibil asociate. Acest tabel are șase coloane: ID\_MOTOR, care este cheia primară și identifică în mod unic fiecare motor; ID\_CARBURANT, care este o cheie externă referitoare la tabela CARBURANT și indică tipul de combustibil folosit de motor; CAI\_PUTERE, care reprezintă puterea motorului în cai putere și trebuie să fie un număr pozitiv, verificat printr-o constrângere CHECK; CAPACITATE\_CILINDRICA\_LITRI, care indică capacitatea cilindrică a motorului în litri și trebuie să fie un număr pozitiv, verificat printr-o constrângere CHECK; CONSUM\_MEDIU\_LITRI, care reprezintă consumul mediu de combustibil al motorului în litri și trebuie să fie un număr pozitiv, verificat printr-o constrângere CHECK; și ASPIRATIE, care specifică tipul de aspirație al motorului și poate fi fie "Aspirat", fie "Turbina". Această structură permite asocierea între motoare și tipurile de combustibil, precum și înregistrarea caracteristicilor și specificațiilor tehnice ale acestora.

## Entitatea MOTOR\_ELECTRIC

Tabelul MOTOR\_ELECTRIC reprezintă o entitate din baza de date care stochează informații despre motoarele electrice. Acest tabel are patru coloane: ID\_MOTOR, care este cheia primară și identifică în mod unic fiecare motor electric; CAI\_PUTERE, care reprezintă puterea motorului în cai putere și trebuie să fie un număr pozitiv, verificat printr-o constrângere CHECK; TIMP\_INCARCARE\_ORE, care indică timpul necesar pentru încărcarea motorului în ore și trebuie să fie un număr pozitiv, verificat printr-o constrângere CHECK; și AUTONOMIE\_KM, care reprezintă autonomia motorului electric în kilometri și trebuie să fie un număr pozitiv, verificat printr-o constrângere CHECK. Această structură permite înregistrarea și gestionarea datelor despre motoarele electrice, inclusiv specificațiile lor tehnice.

## Entitatea PLATA

Tabelul PLATA reprezintă o entitate din baza de date care stochează informații despre plățile efectuate de către clienți. Acest tabel are trei coloane: ID\_PLATA, care este cheia primară și identifică în mod unic fiecare plată; ID\_CLIENT, care este o cheie externă referitoare la tabela CLIENT și indică clientul care efectuează plata; SUMA, care reprezintă suma plătită și trebuie să fie un număr pozitiv, verificat printr-o constrângere CHECK. Această structură permite înregistrarea și gestionarea plăților efectuate de către clienți în cadrul sistemului.

## Entitatea SERVICE

Tabelul SERVICE reprezintă o entitate din baza de date, tabel intermediar in relația „many-to-many” dintre MECANIC si VEHICUL\_DETINUT\_CLIENT, care stochează informații despre serviciile efectuate asupra vehiculelor. Acest tabel are patru coloane: ID\_ANGAJAT, care este o cheie externă referitoare la tabela MECANIC și indică mecanicul care a efectuat serviciul; ID\_VEHICUL, care este o cheie externă referitoare la tabela VEHICUL\_DETINUT\_CLIENT și indică vehiculul asupra căruia s-a efectuat serviciul; ID\_PLATA, care este o cheie externă referitoare la tabela PLATA și indică plata asociată cu serviciul; și SUMA, care reprezintă suma plătită pentru serviciu și trebuie să fie un număr pozitiv, verificat printr-o constrângere CHECK. Această structură permite înregistrarea și gestionarea serviciilor efectuate asupra vehiculelor, incluzând informații despre mecanici, vehicule și plăți asociate.

## Entitatea TRANZACTIE

Tabelul TRANZACTIE reprezintă o entitate din baza de date care stochează informații despre tranzacțiile efectuate. Acest tabel are șapte coloane: ID\_TRANZACTIE, care este cheia primară și identifică în mod unic fiecare tranzacție; ID\_CLIENT, care este o cheie externă referitoare la tabela CLIENT și indică clientul implicat în tranzacție; ID\_PLATA, care este o cheie externă referitoare la tabela PLATA și indică plata asociată cu tranzacția; ID\_VEHICUL, care este o cheie externă referitoare la tabela VEHICUL\_DETINUT\_CLIENT și indică vehiculul implicat în tranzacție; ID\_ANGAJAT, care este o cheie externă referitoare la tabela VANZATOR și indică angajatul care a efectuat tranzacția; DATA\_TRANZACTIE, care înregistrează data tranzacției și are o valoare implicită setată la data curentă (SYSDATE); și SUMA, care reprezintă suma totală a tranzacției și trebuie să fie un număr pozitiv, verificat printr-o constrângere CHECK. Această structură permite înregistrarea și gestionarea informațiilor despre tranzacțiile efectuate între clienți, vehicule și angajați, precum și detalierea plăților asociate.

## Entitatea VANZATOR

Tabelul VANZATOR reprezintă o entitate din baza de date care stochează informații despre vânzătorii din showroom-uri. Acest tabel are două coloane: ID\_ANGAJAT, care este cheia primară și identifică în mod unic fiecare vânzător, fiind totodată și o cheie externă care face referire la tabela ANGAJAT\_SHOWROOM; și ID\_SPECIALIZARE, care este o cheie externă referitoare la tabela CAROSERIE și indică specializarea vânzătorului. Această structură permite asocierea între vânzători și showroom-uri, precum și specializările lor în domeniul caroseriilor.

## Entitatea VEHICUL

Tabelul VEHICUL reprezintă o entitate din baza de date care stochează informații despre vehicule. Acest tabel are opt coloane: ID\_VEHICUL, care este cheia primară și identifică în mod unic fiecare vehicul; ID\_SHOWROOM, care este o cheie externă referitoare la tabela SHOWROOM și indică showroom-ul în care este expus vehiculul; ID\_CAROSERIE, care este o cheie externă referitoare la tabela CAROSERIE și indică tipul de caroserie al vehiculului; ID\_MOTOR, care este o cheie externă referitoare la tabela MOTOR și indică motorul montat pe vehicul; MARCA, care conține marca vehiculului și are o limită de 30 de caractere; MODEL, care conține modelul vehiculului și are o limită de 30 de caractere; AN\_FABRICATIE, care indică anul fabricației vehiculului și trebuie să fie un număr pozitiv, verificat printr-o constrângere CHECK; și PRET, care reprezintă prețul vehiculului și trebuie să fie un număr pozitiv, verificat printr-o constrângere CHECK. Această structură permite înregistrarea și gestionarea informațiilor despre diversele vehicule disponibile în showroom-uri, inclusiv detalii despre marca, modelul, anul fabricației și prețul lor.

## Entitatea VEHICUL\_DETINUT\_CLIENT

Tabelul VEHICUL\_DETINUT\_CLIENT reprezintă o entitate din baza de date care stochează informații despre vehiculele deținute de către clienți. Acest tabel are opt coloane: ID\_VEHICUL, care este cheia primară și identifică în mod unic fiecare vehicul; ID\_CLIENT, care este o cheie externă referitoare la tabela CLIENT și indică clientul care deține vehiculul; ID\_CAROSERIE, care este o cheie externă referitoare la tabela CAROSERIE și indică tipul de caroserie al vehiculului; ID\_MOTOR, care este o cheie externă referitoare la tabela MOTOR și indică motorul montat pe vehicul; MARCA, care conține marca vehiculului și are o limită de 30 de caractere; MODEL, care conține modelul vehiculului și are o limită de 30 de caractere; AN\_FABRICATIE, care indică anul fabricației vehiculului și trebuie să fie un număr pozitiv, verificat printr-o constrângere CHECK; și NUMAR\_INMATRICULARE, care reprezintă numărul de înmatriculare al vehiculului și trebuie să respecte un format specific românesc, verificat printr-o constrângere CHECK. Această structură permite înregistrarea și gestionarea informațiilor despre vehiculele deținute de către clienți, inclusiv detalii despre marca, modelul, anul fabricației și numărul de înmatriculare.

# **4. Descrierea relațiilor, incluzând precizarea cardinalității acestora**

## MOTOR\_CARBURANT (ID\_MOTOR) - MOTOR (ID\_MOTOR)

One-to-One: Fiecare motor din tabelul MOTOR\_CARBURANT este asociat cu un singur motor din tabelul MOTOR. Această relație indică faptul că specificațiile de carburant sunt unice pentru fiecare motor, astfel un motor nu poate avea multiple tipuri de carburant asociate, 1(0):1.

## MOTOR\_CARBURANT (ID\_CARBURANT) - CARBURANT (ID\_CARBURANT)

One-to-Many: Un tip de carburant din tabelul CARBURANT poate fi utilizat de mai multe motoare din tabelul MOTOR\_CARBURANT. Acest lucru reflectă faptul că un tip de carburant, cum ar fi Benzina sau Motorina, poate fi compatibil cu multiple modele de motoare, permițând utilizarea sa în diverse aplicații, cardinalitatea fiind M(0):1.

## MOTOR\_ELECTRIC (ID\_MOTOR) - MOTOR (ID\_MOTOR)

One-to-One: Fiecare motor electric din tabelul MOTOR\_ELECTRIC corespunde unui singur motor din tabelul MOTOR. Acest aranjament subliniază că fiecare motor electric are specificații unice care sunt înregistrate în tabelul MOTOR, eliminând posibilitatea duplicării acestor specificații, cardinalitatea fiind 1(0):1.

## SHOWROOM (ID\_ADRESA) - ADRESA (ID\_ADRESA)

One-to-One: Fiecare showroom din tabelul SHOWROOM este asociat cu o singură adresă din tabelul ADRESA. Aceasta reflectă faptul că fiecare showroom are o locație fizică unică, necesară pentru identificarea exactă și logistică, cardinalitatea fiind 1(0):1.

## SHOWROOM\_NUMERE\_TELEFON (ID\_SHOWROOM\_NUMAR\_TELEFON) - SHOWROOM (ID\_SHOWROOM)

One-to-Many: Un showroom din tabelul SHOWROOM poate avea mai multe numere de telefon înregistrate în tabelul SHOWROOM\_NUMERE\_TELEFON. Aceasta permite unui showroom să gestioneze mai multe linii telefonice pentru diferite scopuri, cum ar fi departamentele de vânzări, servicii clienți sau suport tehnic, cardinalitatea fiind M(1):1(0).

## VEHICUL (ID\_SHOWROOM) - SHOWROOM (ID\_SHOWROOM)

One-to-Many: Un showroom din tabelul SHOWROOM poate avea mai multe vehicule listate în tabelul VEHICUL. Această relație indică faptul că showroom-urile pot expune și vinde o gamă largă de vehicule, fiecare showroom având un inventar propriu de vehicule, cardinalitatea fiind M(1):1(0).

## VEHICUL (ID\_CAROSERIE) - CAROSERIE (ID\_CAROSERIE)

One-to-Many: O caroserie din tabelul CAROSERIE poate fi utilizată de mai multe vehicule înregistrate în tabelul VEHICUL. Acest lucru reflectă realitatea că un anumit tip de caroserie poate fi folosit în producția mai multor modele de vehicule, permițând diversitatea designului vehiculelor bazate pe aceleași specificații de caroserie, cardinalitatea fiind M(0):1.

## VEHICUL (ID\_MOTOR) - MOTOR (ID\_MOTOR)

One-to-Many: Un motor din tabelul MOTOR poate fi utilizat de mai multe vehicule înregistrate în tabelul VEHICUL. Această relație arată că un anumit tip de motor poate echipa diverse modele de vehicule, subliniind modularitatea și reutilizarea motoarelor în designul vehiculelor, cardinalitatea fiind M(0):1.

## ANGAJAT\_SHOWROOM (ID\_SHOWROOM) - SHOWROOM (ID\_SHOWROOM)

One-to-Many: Un showroom din tabelul SHOWROOM poate avea mai mulți angajați înregistrați în tabelul ANGAJAT\_SHOWROOM. Aceasta indică faptul că showroom-urile angajează personal divers pentru a îndeplini diferite roluri și responsabilități, de la vânzări pana la service, cardinalitatea fiind M(1):1.

## MECANIC (ID\_ANGAJAT) - ANGAJAT\_SHOWROOM (ID\_ANGAJAT)

One-to-One: Fiecare mecanic din tabelul MECANIC este un angajat unic înregistrat în tabelul ANGAJAT\_SHOWROOM. Aceasta subliniază faptul că fiecare mecanic are o identitate și un set de competențe unice, fiind recunoscut individual în cadrul organizației, cardinalitatea fiind 1(0):1.

## MECANIC (ID\_SPECIALIZARE) - MOTOR (ID\_MOTOR)

One-to-Many: Un motor din tabelul MOTOR poate avea mai mulți mecanici specializați în tabelul MECANIC. Aceasta reflectă faptul că multiple persoane pot fi specializate în întreținerea și repararea aceluiași tip de motor, asigurând o expertiză variată și disponibilitate pentru clienți, cardinalitatea fiind M(0):1(0).

## VANZATOR (ID\_ANGAJAT) - ANGAJAT\_SHOWROOM (ID\_ANGAJAT)

One-to-One: Fiecare vânzător din tabelul VANZATOR este un angajat unic înregistrat în tabelul ANGAJAT\_SHOWROOM. Aceasta indică faptul că fiecare vânzător are o identitate individuală și este responsabil pentru propriile vânzări și performanțe, cardinalitatea fiind 1(0):1.

## VANZATOR (ID\_SPECIALIZARE) - CAROSERIE (ID\_CAROSERIE)

One-to-Many: O caroserie din tabelul CAROSERIE poate fi specializarea mai multor vânzători din tabelul VANZATOR. Aceasta arată că diferiți vânzători pot avea expertiză în vânzarea vehiculelor cu aceeași tip de caroserie, facilitând abordări diverse în marketing și vânzări, cardinalitatea fiind M(0):1(0).

## CLIENT\_NUMERE\_TELEFON (ID\_CLIENT\_NR\_TELEFON) - CLIENT (ID\_CLIENT)

One-to-Many: Un client din tabelul CLIENT poate avea mai multe numere de telefon înregistrate în tabelul CLIENT\_NUMERE\_TELEFON. Aceasta permite unui client să fie contactat prin multiple linii telefonice, cardinalitatea fiind M(1):1.

## VEHICUL\_DETINUT\_CLIENT (ID\_CLIENT) - CLIENT (ID\_CLIENT)

One-to-Many: Un client din tabelul CLIENT poate deține mai multe vehicule înregistrate în tabelul VEHICUL\_DETINUT\_CLIENT. Aceasta indică faptul că un client poate avea în proprietate mai multe vehicule, cardinalitatea fiind M(1):1.

## VEHICUL\_DETINUT\_CLIENT (ID\_CAROSERIE) - CAROSERIE (ID\_CAROSERIE)

One-to-Many: O caroserie din tabelul CAROSERIE poate fi utilizată de mai multe vehicule deținute de clienți în tabelul VEHICUL\_DETINUT\_CLIENT. Aceasta subliniază că un anumit tip de caroserie poate fi comun pentru mai multe vehicule, reflectând popularitatea și versatilitatea sa, cardinalitatea fiind M(0):1.

## VEHICUL\_DETINUT\_CLIENT (ID\_MOTOR) - MOTOR (ID\_MOTOR)

One-to-Many: Un motor din tabelul MOTOR poate fi utilizat în mai multe vehicule deținute de clienți în tabelul VEHICUL\_DETINUT\_CLIENT. Aceasta arată că același tip de motor poate fi instalat în diferite vehicule, demonstrând fiabilitatea și adaptabilitatea sa, cardinalitatea fiind M(0):1.

## PLATA (ID\_CLIENT) - CLIENT (ID\_CLIENT)

One-to-Many: Un client din tabelul CLIENT poate efectua mai multe plăți înregistrate în tabelul PLATA. Aceasta reflectă faptul că un client poate face mai multe achiziții sau poate plăti pentru diverse servicii, având multiple tranzacții financiare, cardinalitatea fiind M(1):1.

## TRANZACTIE (ID\_CLIENT) - CLIENT (ID\_CLIENT)

One-to-Many: Un client din tabelul CLIENT poate avea mai multe tranzacții înregistrate în tabelul TRANZACTIE. Aceasta subliniază că un client poate participa la mai multe tranzacții de cumpărare sau vânzare de vehicule sau servicii, demonstrând activitatea sa economică diversificată, cardinalitatea fiind M(1):1.

## TRANZACTIE (ID\_PLATA) - PLATA (ID\_PLATA)

One-to-One: Fiecare tranzacție din tabelul TRANZACTIE are asociată o singură plată în tabelul PLATA. Aceasta indică faptul că fiecare tranzacție este legată de o plată specifică, asigurând trasabilitatea și claritatea tranzacțiilor financiare, cardinalitatea fiind 1(0):1.

## TRANZACTIE (ID\_VEHICUL) - VEHICUL\_DETINUT\_CLIENT (ID\_VEHICUL)

One-to-One: Fiecare tranzacție din tabelul TRANZACTIE este asociată cu un singur vehicul din tabelul VEHICUL\_DETINUT\_CLIENT. Aceasta arată că fiecare tranzacție se referă la un vehicul specific, facilitând identificarea și managementul vehiculelor implicate în tranzacții, cardinalitatea fiind 1:1.

## TRANZACTIE (ID\_ANGAJAT) - VANZATOR (ID\_ANGAJAT)

One-to-One: Fiecare tranzacție din tabelul TRANZACTIE este gestionată de un singur vânzător din tabelul VANZATOR. Aceasta subliniază că fiecare tranzacție este responsabilitatea unui vânzător specific, permițând monitorizarea performanței și evaluarea individuală a vânzătorilor, cardinalitatea fiind M(0):1.

## SERVICE (ID\_ANGAJAT) - MECANIC (ID\_ANGAJAT)

One-to-Many: Un mecanic din tabelul MECANIC poate efectua mai multe servicii înregistrate în tabelul SERVICE. Aceasta indică faptul că un mecanic poate oferi reparații și întreținere pentru multiple vehicule, reflectând experiența și capacitatea sa de a gestiona diverse sarcini tehnice, cardinalitatea fiind M(0):1.

## SERVICE (ID\_VEHICUL) - VEHICUL\_DETINUT\_CLIENT (ID\_VEHICUL)

One-to-Many: Un vehicul din tabelul VEHICUL\_DETINUT\_CLIENT poate necesita mai multe servicii înregistrate în tabelul SERVICE. Aceasta subliniază că un vehicul poate necesita mai multe intervenții de service pe durata utilizării sale, asigurând funcționarea optimă și longevitatea vehiculului, cardinalitatea fiind M(0):1.

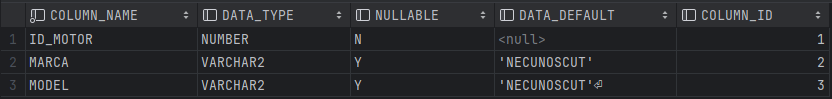
## SERVICE (ID\_PLATA) - PLATA (ID\_PLATA)

One-to-One: Fiecare serviciu din tabelul SERVICE are asociată o singură plată în tabelul PLATA. Aceasta arată că fiecare serviciu de reparație sau întreținere implică o plată specifică, facilitând gestionarea costurilor și trasabilitatea financiară a serviciilor oferite, cardinalitatea fiind 1(0):1.

# **5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.**

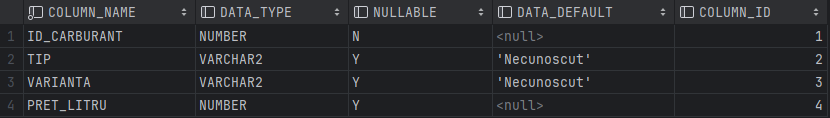
## Tabel MOTOR

* ID\_MOTOR: INT, PRIMARY KEY. Identificatorul unic al fiecărui motor.
* MARCA: VARCHAR(30), DEFAULT 'NECUNOSCUT'. Marca motorului. Valoare implicită este 'NECUNOSCUT'.
* MODEL: VARCHAR(30), DEFAULT 'NECUNOSCUT'. Modelul motorului. Valoare implicită este 'NECUNOSCUT'.



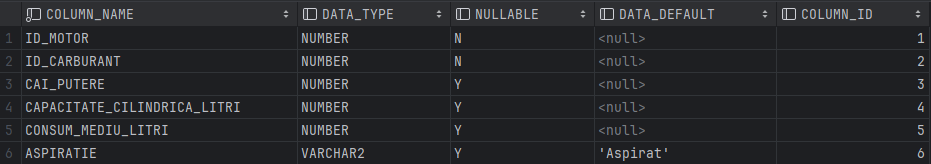
## Tabel CARBURANT

* ID\_CARBURANT: INT, PRIMARY KEY. Identificatorul unic al tipului de carburant.
* TIP: VARCHAR(30), DEFAULT 'Necunoscut'. Tipul carburantului (ex: Benzina, Motorina). Valoare implicită este 'Necunoscut'. Constrângere: TIP poate avea valori doar 'Benzina' sau 'Motorina'.
* VARIANTA: VARCHAR(30), DEFAULT 'Necunoscut'. Varianta carburantului. Valoare implicită este 'Necunoscut'.
* PRET\_LITRU: NUMBER(3,2). Prețul per litru al carburantului. Trebuie să fie o valoare pozitivă.



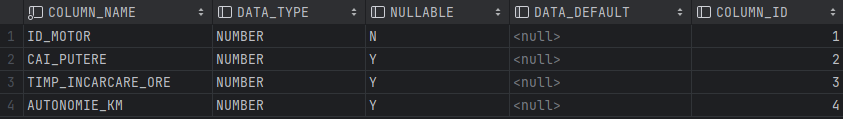
## Tabel MOTOR\_CARBURANT

* ID\_MOTOR: INT, PRIMARY KEY. Identificatorul motorului, legat de tabelul MOTOR.
* ID\_CARBURANT: INT. Identificatorul tipului de carburant, legat de tabelul CARBURANT.
* CAI\_PUTERE: INT. Puterea motorului în cai putere. Trebuie să fie o valoare pozitivă.
* CAPACITATE\_CILINDRICA\_LITRI: NUMBER(3,1). Capacitatea cilindrică a motorului în litri. Trebuie să fie o valoare pozitivă.
* CONSUM\_MEDIU\_LITRI: NUMBER(4,2). Consumul mediu de carburant în litri la 100 km. Trebuie să fie o valoare pozitivă.
* ASPIRATIE: VARCHAR(30), DEFAULT 'Aspirat'. Tipul de aspirație al motorului. Valoare implicită este 'Aspirat'. Constrângere: ASPIRATIE poate avea valori doar 'Aspirat' sau 'Turbina'.



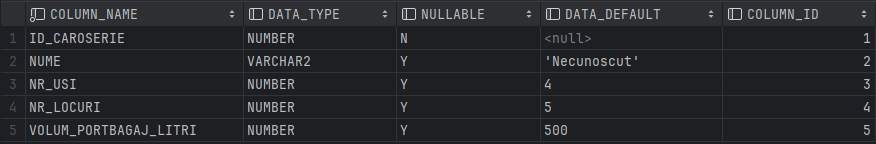
## Tabel MOTOR\_ELECTRIC

* ID\_MOTOR: INT, PRIMARY KEY. Identificatorul motorului electric, legat de tabelul MOTOR.
* CAI\_PUTERE: INT. Puterea motorului electric în cai putere. Trebuie să fie o valoare pozitivă.
* TIMP\_INCARCARE\_ORE: NUMBER(4,3). Timpul de încărcare al bateriei în ore. Trebuie să fie o valoare pozitivă.
* AUTONOMIE\_KM: INT. Autonomia motorului electric în kilometri. Trebuie să fie o valoare pozitivă.



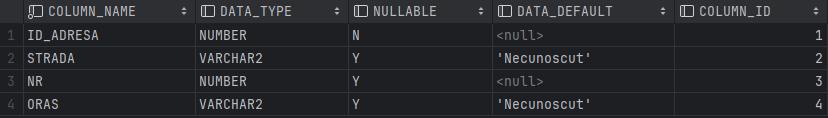
## Tabel CAROSERIE

* ID\_CAROSERIE: INT, PRIMARY KEY. Identificatorul unic al caroseriei.
* NUME: VARCHAR(30), DEFAULT 'Necunoscut'. Numele caroseriei. Valoare implicită este 'Necunoscut'.
* NR\_USI: INT, DEFAULT 4. Numărul de uși al caroseriei. Valoare implicită este 4. Trebuie să fie o valoare pozitivă.
* NR\_LOCURI: INT, DEFAULT 5. Numărul de locuri al caroseriei. Valoare implicită este 5. Trebuie să fie o valoare pozitivă.
* VOLUM\_PORTBAGAJ\_LITRI: INT, DEFAULT 500. Volumul portbagajului în litri. Valoare implicită este 500. Trebuie să fie o valoare pozitivă.



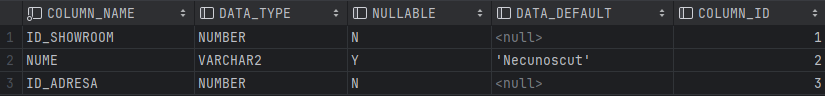
## Tabel ADRESA

* ID\_ADRESA: INT, PRIMARY KEY. Identificatorul unic al adresei.
* STRADA: VARCHAR(30), DEFAULT 'Necunoscut'. Numele străzii. Valoare implicită este 'Necunoscut'.
* NR: INT. Numărul străzii. Trebuie să fie o valoare pozitivă.
* ORAS: VARCHAR(30), DEFAULT 'Necunoscut'. Numele orașului. Valoare implicită este 'Necunoscut'.



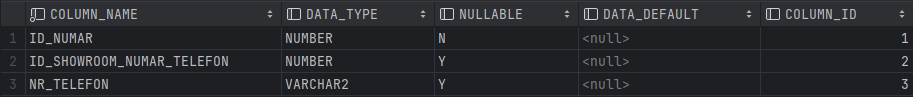
## Tabel SHOWROOM

* ID\_SHOWROOM: INT, PRIMARY KEY. Identificatorul unic al showroom-ului.
* NUME: VARCHAR(30), DEFAULT 'Necunoscut'. Numele showroom-ului. Valoare implicită este 'Necunoscut'.
* ID\_ADRESA: INT. Identificatorul adresei showroom-ului, legat de tabelul ADRESA.



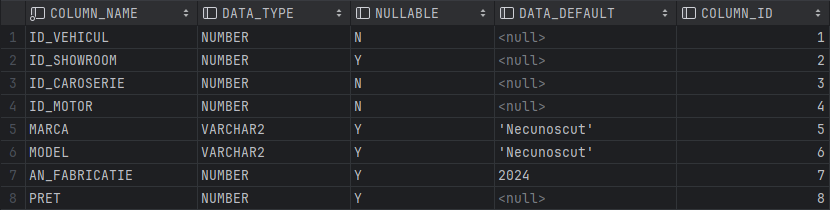
## Tabel SHOWROOM\_NUMERE\_TELEFON

* ID\_NUMAR: INT, PRIMARY KEY. Identificatorul unic al numărului de telefon.
* ID\_SHOWROOM\_NUMAR\_TELEFON: INT. Identificatorul showroom-ului, legat de tabelul SHOWROOM.
* NR\_TELEFON: VARCHAR(30). Numărul de telefon al showroom-ului. Trebuie să respecte formatul '07CC CCC CCC', unde C este o cifra de la 0 la 9.



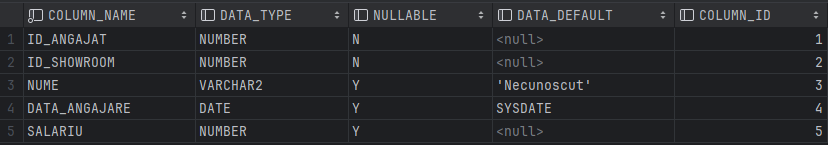
## Tabel VEHICUL

* ID\_VEHICUL: INT, PRIMARY KEY. Identificatorul unic al vehiculului.
* ID\_SHOWROOM: INT. Identificatorul showroom-ului unde se află vehiculul, legat de tabelul SHOWROOM.
* ID\_CAROSERIE: INT. Identificatorul caroseriei vehiculului, legat de tabelul CAROSERIE.
* ID\_MOTOR: INT. Identificatorul motorului vehiculului, legat de tabelul MOTOR.
* MARCA: VARCHAR(30), DEFAULT 'Necunoscut'. Marca vehiculului. Valoare implicită este 'Necunoscut'.
* MODEL: VARCHAR(30), DEFAULT 'Necunoscut'. Modelul vehiculului. Valoare implicită este 'Necunoscut'.
* AN\_FABRICATIE: INT, DEFAULT 2024. Anul fabricației vehiculului. Valoare implicită este 2024. Trebuie să fie o valoare pozitivă.
* PRET: NUMBER(10,2). Prețul vehiculului. Trebuie să fie o valoare pozitivă.



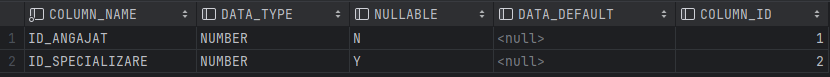
## Tabel ANGAJAT\_SHOWROOM

* ID\_ANGAJAT: INT, PRIMARY KEY. Identificatorul unic al angajatului.
* ID\_SHOWROOM: INT. Identificatorul showroom-ului unde lucrează angajatul, legat de tabelul SHOWROOM.
* NUME: VARCHAR(30), DEFAULT 'Necunoscut'. Numele angajatului. Valoare implicită este 'Necunoscut'.
* DATA\_ANGAJARE: DATE, DEFAULT SYSDATE. Data angajării angajatului. Valoare implicită este data curentă.
* SALARIU: NUMBER(10,2). Salariul angajatului. Trebuie să fie o valoare pozitivă.



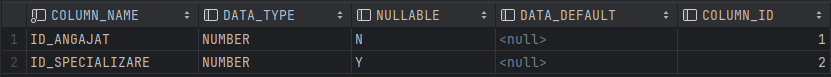
## Tabel MECANIC

* ID\_ANGAJAT: INT, PRIMARY KEY. Identificatorul unic al mecanicului.
* ID\_SPECIALIZARE: INT. Identificatorul specializării, legat de tabelul MOTOR.



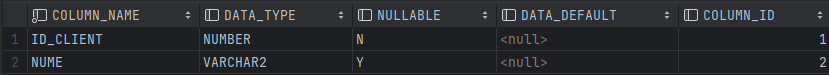
## Tabel VANZATOR

* ID\_ANGAJAT: INT, PRIMARY KEY. Identificatorul unic al vânzătorului.
* ID\_SPECIALIZARE: INT. Identificatorul specializării, legat de tabelul CAROSERIE.



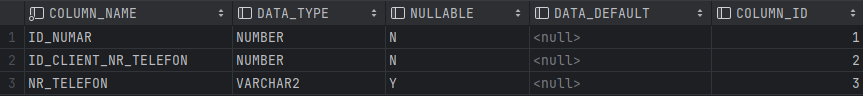
## Tabel CLIENT

* ID\_CLIENT: INT, PRIMARY KEY. Identificatorul unic al clientului.
* NUME: VARCHAR(30). Numele clientului.



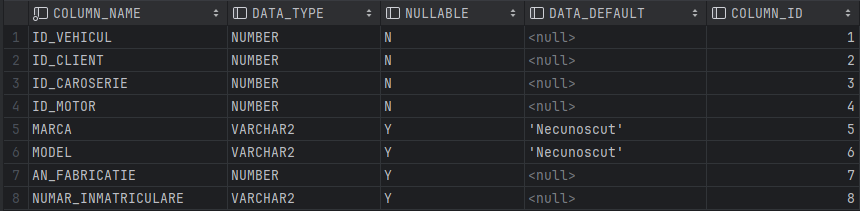
## Tabel CLIENT\_NUMERE\_TELEFON

* ID\_NUMAR: INT, PRIMARY KEY. Identificatorul unic al numărului de telefon al clientului.
* ID\_CLIENT\_NR\_TELEFON: INT. Identificatorul clientului, legat de tabelul CLIENT.
* NR\_TELEFON: VARCHAR(30). Numărul de telefon al clientului. Trebuie să respecte formatul '07CC CCC CCC’, unde C este o cifra de la 0 la 9.



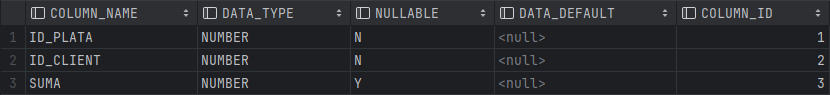
## Tabel VEHICUL\_DETINUT\_CLIENT

* ID\_VEHICUL: INT, PRIMARY KEY. Identificatorul unic al vehiculului deținut de client.
* ID\_CLIENT: INT. Identificatorul clientului, legat de tabelul CLIENT.
* ID\_CAROSERIE: INT. Identificatorul caroseriei vehiculului, legat de tabelul CAROSERIE.
* ID\_MOTOR: INT. Identificatorul motorului vehiculului, legat de tabelul MOTOR.
* MARCA: VARCHAR(30), DEFAULT 'Necunoscut'. Marca vehiculului. Valoare implicită este 'Necunoscut'.
* MODEL: VARCHAR(30), DEFAULT 'Necunoscut'. Modelul vehiculului. Valoare implicită este 'Necunoscut'.
* AN\_FABRICATIE: INT. Anul fabricației vehiculului. Trebuie să fie o valoare pozitivă.
* NUMAR\_INMATRICULARE: VARCHAR(30), UNIQUE. Numărul de înmatriculare al vehiculului. Trebuie să respecte formatul specificat pentru numerele de înmatriculare românești.



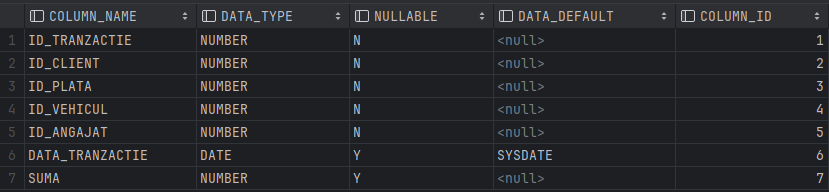
## Tabel PLATA

* ID\_PLATA: INT, PRIMARY KEY. Identificatorul unic al plății.
* ID\_CLIENT: INT. Identificatorul clientului care face plata, legat de tabelul CLIENT.
* SUMA: NUMBER(10,2). Suma plătită. Trebuie să fie o valoare pozitivă.



## Tabel TRANZACTIE

* ID\_TRANZACTIE: INT, PRIMARY KEY. Identificatorul unic al tranzacției.
* ID\_CLIENT: INT. Identificatorul clientului implicat în tranzacție, legat de tabelul CLIENT.
* ID\_PLATA: INT. Identificatorul plății asociate tranzacției, legat de tabelul PLATA.
* ID\_VEHICUL: INT. Identificatorul vehiculului implicat în tranzacție, legat de tabelul VEHICUL\_DETINUT\_CLIENT.
* ID\_ANGAJAT: INT. Identificatorul angajatului care a efectuat tranzacția, legat de tabelul VANZATOR.
* DATA\_TRANZACTIE: DATE, DEFAULT SYSDATE. Data tranzacției. Valoare implicită este data curentă.
* SUMA: NUMBER(10,2). Suma tranzacției. Trebuie să fie o valoare pozitivă.

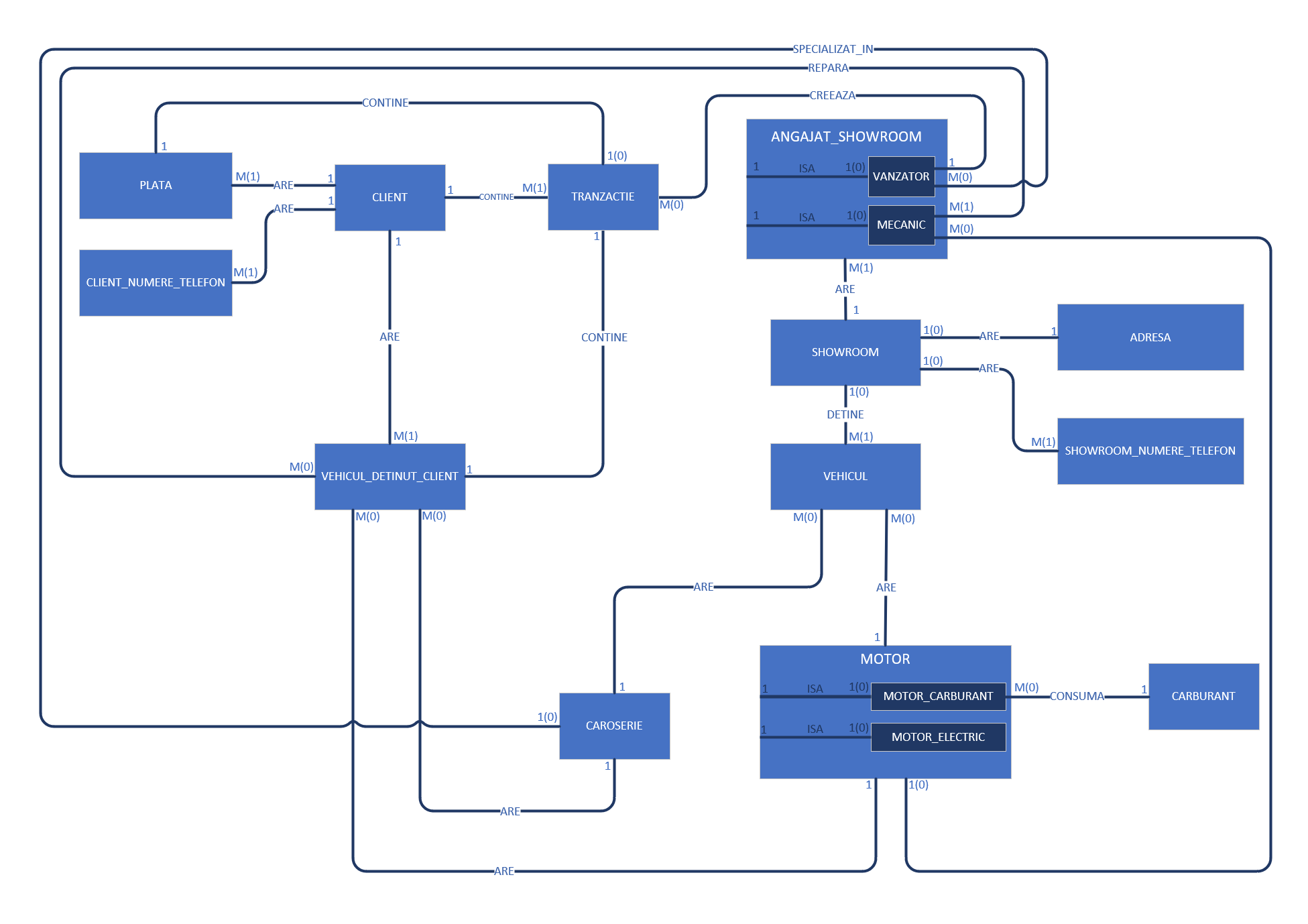


## Tabel SERVICE

* ID\_ANGAJAT: INT. Identificatorul mecanicului care a efectuat serviciul, legat de tabelul MECANIC.
* ID\_VEHICUL: INT. Identificatorul vehiculului care a primit serviciul, legat de tabelul VEHICUL\_DETINUT\_CLIENT.
* ID\_PLATA: INT. Identificatorul plății asociate serviciului, legat de tabelul PLATA.
* SUMA: NUMBER(10,2). Suma serviciului. Trebuie să fie o valoare pozitivă.

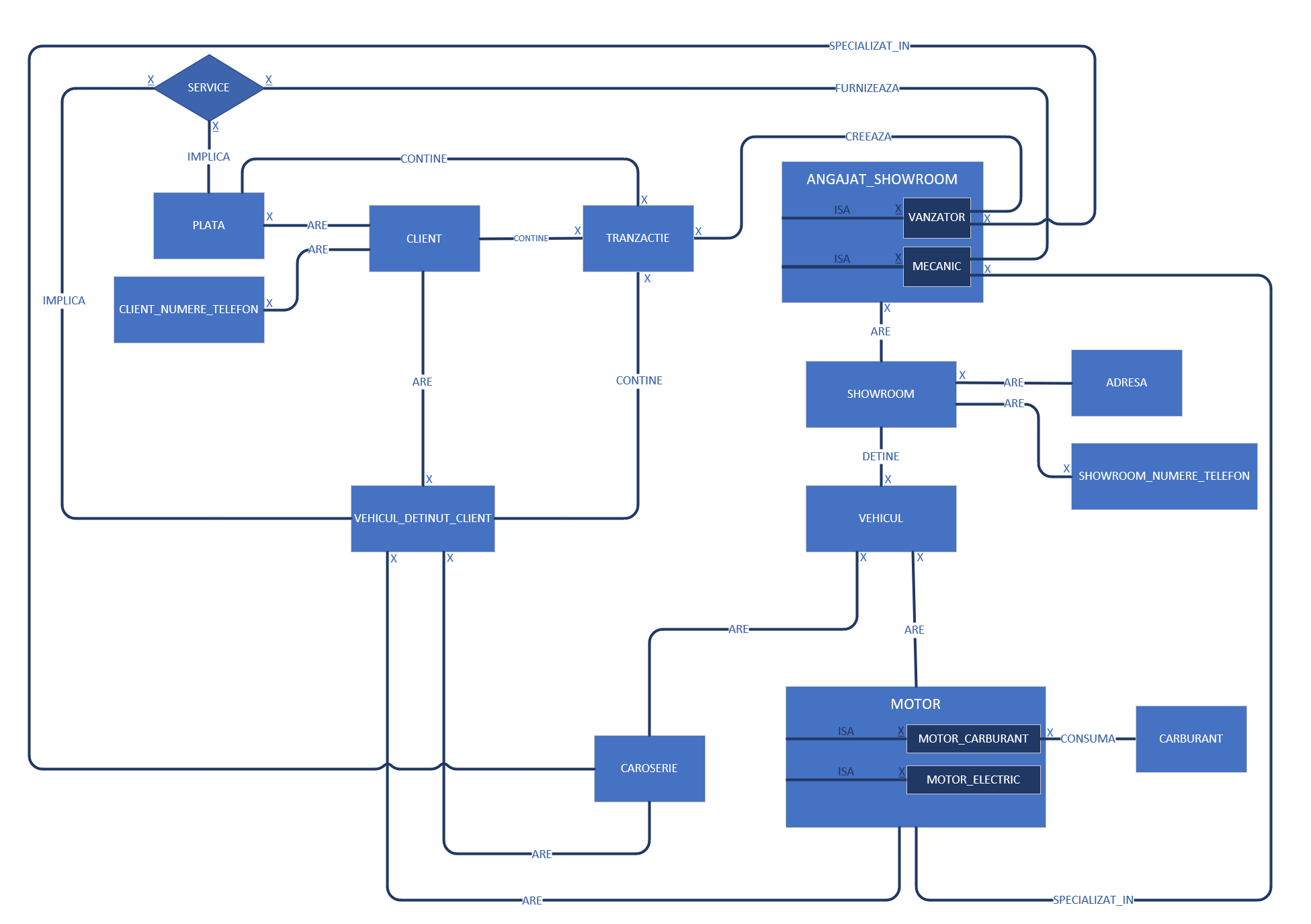


# **6. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.**

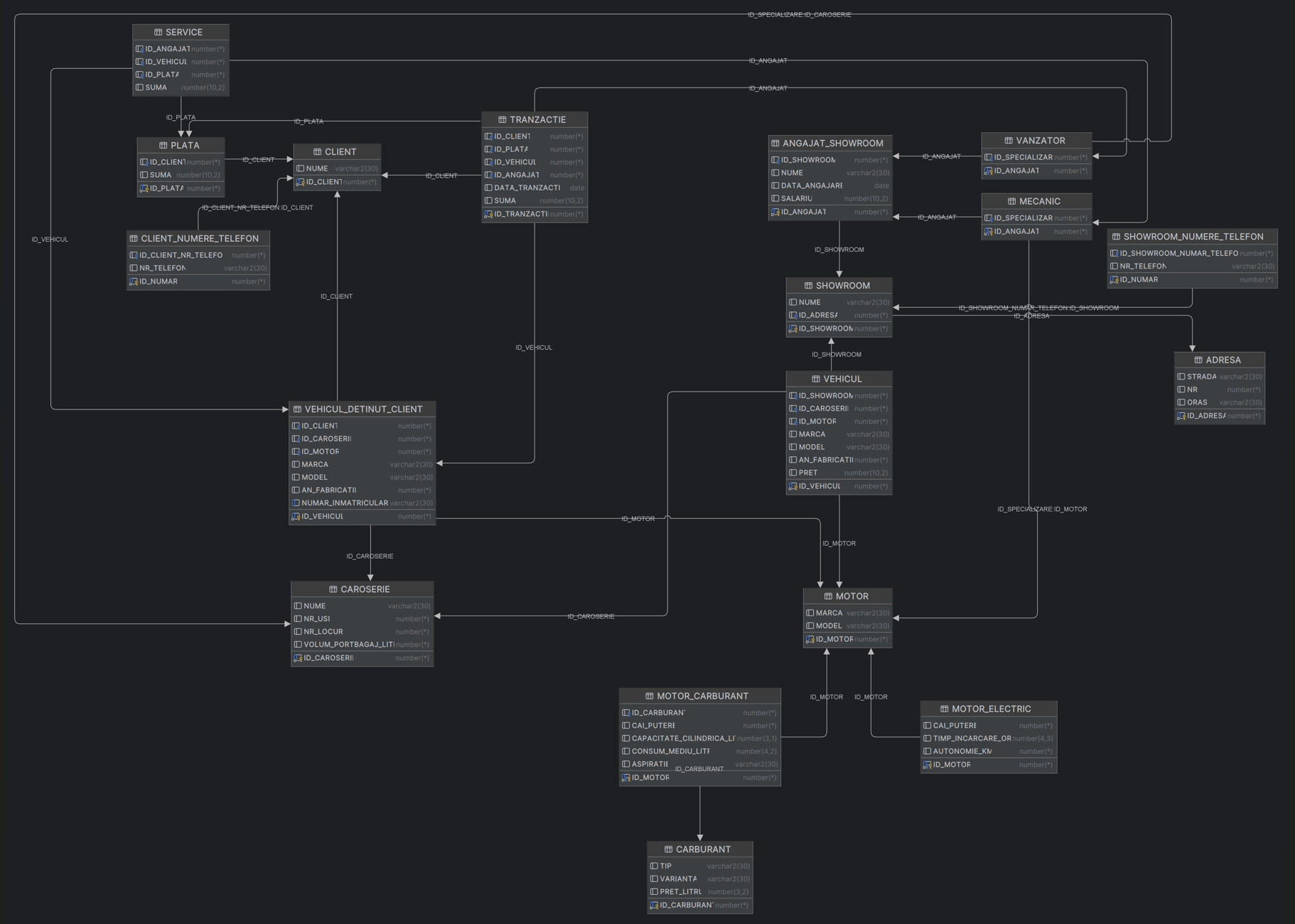


# **7. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6.**

* Diagrama conceptuală obținută trebuie să conțină minimum 7 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ.



* In plus, voi adaugă si diagrama conceptuala generata automat de DataGrip, unde se pot vedea si coloanele fiecărui tabel si cheile primare/străine.



# **8**. **Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.**

* **ADRESA** (ID\_ADRESA#, STRADA, NR, ORAS)
* **ANGAJAT\_SHOWROOM** (ID\_ANGAJAT#, ID\_SHOWROOM, NUME, DATA\_ANAGAJARE, SALARIU)
* **CARBURANT** (ID\_CARBURANT#, TIP, VARIANTA, PRET\_LITRU)
* **CAROSERIE** (ID\_CAROSERIE#, NUME, NR\_USI, NR\_LOCURI, VOLUM\_PORTBAGAJ\_LITRI)
* **CLIENT** (ID\_CLIENT#, NUME)
* **CLIENT\_NUMERE\_TELEFON** (ID\_NUMAR#, ID\_CLIENT\_NR\_TELEFON, NR\_TELEFON)
* **MECANIC** (ID\_ANGAJAT#, ID\_SPECIALIZARE)
* **MOTOR** (ID\_MOTOR#, MARCA, MODEL)
* **MOTOR\_CARBURANT** (ID\_MOTOR#, ID\_CARBURANT, CAI\_PUTERE, CAPACITATE\_CILINDRICA\_LITRI, CONSUM\_MEDIU\_LITRI, ASPIRATIE)
* **MOTOR\_ELECTRIC** (ID\_MOTOR#, CAI\_PUTERE, TIMP\_INCARCARE\_ORE, AUTONOMIE\_KM)
* **PLATA** (ID\_PLATA#, ID\_CLIENT, SUMA)
* **SERVICE** (ID\_ANGAJAT#, ID\_VEHICUL#, ID\_PLATA#, SUMA)
* **SHOWROOM** (ID\_SHOWROOM#, NUME, ID\_ADRESA)
* **SHOWROOM** (NUMERE\_TELEFON (ID\_NUMAR#, ID\_SHOWROOM\_NUMAR\_TELEFON, NR\_TELEFON)
* **TRANZACTIE** (ID\_TRANZACTIE#, ID\_CLIENT, ID\_PLATA, ID\_VEHICUL, ID\_ANGAJAT, DATA\_TRANZACTIE, SUMA)
* **VANZATOR** (ID\_ANGAJAT#, ID\_SPECIALIZARE)
* **VEHICUL** (ID\_VEHICUL#, ID\_SHOWROOM, ID\_CAROSERIE, ID\_MOTOR, MARCA, MODEL, AN\_FABRICATIE, PRET)
* **VEHICUL\_DETINUT\_CLIENT** (ID\_VEHICUL#, ID\_CLIENT, ID\_CAROSERIE, ID\_MOTOR, MARCA, MODEL, AN\_FABRICATIE, NUMAR\_INMATRICULARE)

# **9. Realizarea normalizării până la forma normală 3 (FN1-FN3).**

**EXEMPLU NON-FN1 -> FN1:**

**NON-FN1:**

TABEL SHOWROOM

|  |  |
| --- | --- |
| ID\_SHOWROOM | NUMAR\_TELEFON |
| 1 | 0700 000 001, 0700 000 002 |
| 2 | 0700 000 003, 0700 000 004, 0700 000 005 |

**SOLUTIA:** introducerea unui nou tabel SHOWROOM\_NUMERE\_TELEFON

**FN1:**

TABEL SHOWROOM

|  |  |
| --- | --- |
| ID\_SHOWROOM | ... |
| 1 | ... |
| 2 | ... |

TABEL SHOWROOM\_NUMERE\_TELEFON

|  |  |  |
| --- | --- | --- |
| ID\_NUMAR | ID\_SHOWROOM\_NR\_TELEFON | NR\_TELEFON |
| 1 | 1 | 0700 000 001 |
| 2 | 1 | 0700 000 002 |
| 3 | 2 | 0700 000 003 |
| 4 | 2 | 0700 000 004 |
| 5 | 2 | 0700 000 005 |

**EXEMPLU FN1 -> FN2:**

Fiecare atribut non-cheie trebuie sa fie complet dependent de cheia primară.

**FN1:**

TABEL VEHICUL\_CLIENT

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID\_VEHICUL | MARCA | MODEL | ID\_CLIENT | NUME\_CLIENT | NR\_TEL |
| 1 | DACIA | LOGAN | 1 | ION | 0770 000 001 |
| 2 | DACIA | SANDERO | 2 | POP | 0770 000 002 |

**SOLUTIA:** spargere in 2 tabele

**FN2:**

TABEL CLIENT

|  |  |  |
| --- | --- | --- |
| ID\_CLIENT | NUME\_CLIENT | NUMAR\_TELEFON |
| 1 | ION | 0770 000 001 |
| 2 | POP | 0770 000 002 |

TABEL VEHICUL\_DETINUT\_CLIENT

|  |  |  |  |
| --- | --- | --- | --- |
| ID\_VEHICUL | MARCA | MODEL | ID\_CLIENT |
| 1 | DACIA | LOGAN | 1 |
| 2 | DACIA | SANDERO | 2 |

**EXEMPLU FN2 -> FN3:**

Nu trebuie să existe dependențe tranzitive unde un atribut non-cheie depinde de alt atribut non-cheie care la rândul său depinde de cheia primară.

**FN2:**

TABEL\_VEHICUL

|  |  |  |
| --- | --- | --- |
| ID\_VEHICUL | ID\_SHOWROOM | ID\_ADRESA |
| 1 | 1 | 1 |
| 2 | 1 | 1 |
| 3 | 2 | 2 |

**SOLUTIA:** adăugarea unui nou tabel pentru adrese SHOWROOM

**FN3:**

TABEL\_VEHICUL

|  |  |
| --- | --- |
| ID\_VEHICUL | ID\_SHOWROOM |
| 1 | 1 |
| 2 | 1 |
| 3 | 2 |

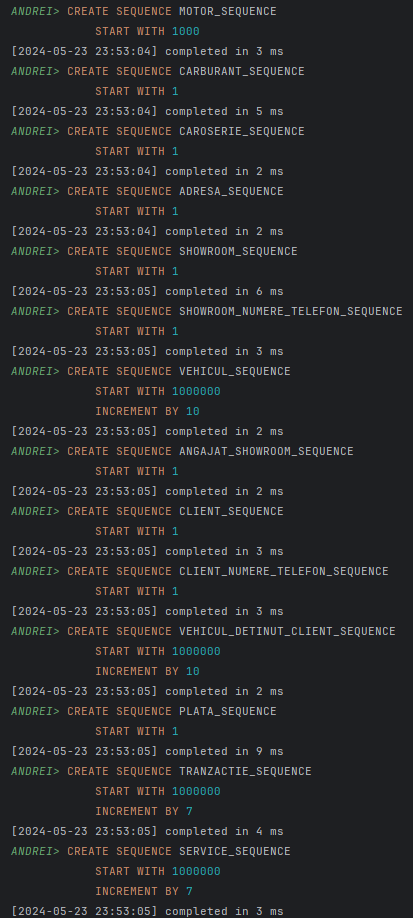
TABEL\_ADRESA

|  |  |
| --- | --- |
| ID\_ADRESA | ID\_SHOWROOM |
| 1 | 1 |
| 2 | 2 |

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

**COD:**

CREATE SEQUENCE MOTOR\_SEQUENCE  
 START WITH 1000;  
CREATE SEQUENCE CARBURANT\_SEQUENCE  
 START WITH 1;  
CREATE SEQUENCE CAROSERIE\_SEQUENCE  
 START WITH 1;  
CREATE SEQUENCE ADRESA\_SEQUENCE  
 START WITH 1;  
CREATE SEQUENCE SHOWROOM\_SEQUENCE  
 START WITH 1;  
CREATE SEQUENCE SHOWROOM\_NUMERE\_TELEFON\_SEQUENCE  
 START WITH 1;  
CREATE SEQUENCE VEHICUL\_SEQUENCE  
 START WITH 1000000  
 INCREMENT BY 10;  
CREATE SEQUENCE ANGAJAT\_SHOWROOM\_SEQUENCE  
 START WITH 1;  
CREATE SEQUENCE CLIENT\_SEQUENCE  
 START WITH 1;  
CREATE SEQUENCE CLIENT\_NUMERE\_TELEFON\_SEQUENCE  
 START WITH 1;  
CREATE SEQUENCE VEHICUL\_DETINUT\_CLIENT\_SEQUENCE  
 START WITH 1000000  
 INCREMENT BY 10;  
CREATE SEQUENCE PLATA\_SEQUENCE  
 START WITH 1;  
CREATE SEQUENCE TRANZACTIE\_SEQUENCE  
 START WITH 1000000  
 INCREMENT BY 7;  
CREATE SEQUENCE SERVICE\_SEQUENCE  
 START WITH 1000000  
 INCREMENT BY 7;

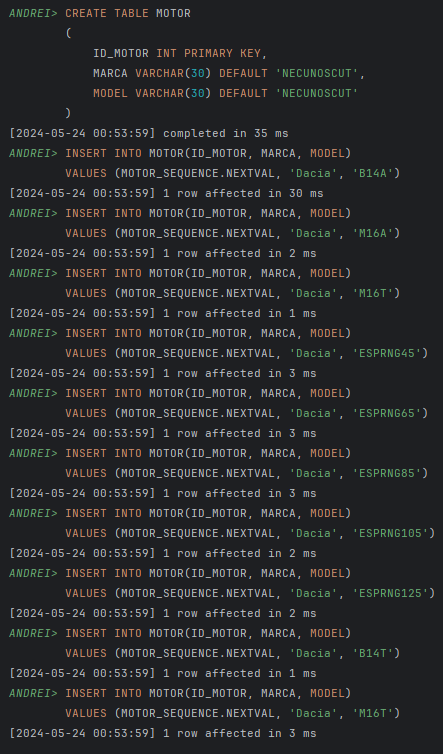
**CONSOLA:**

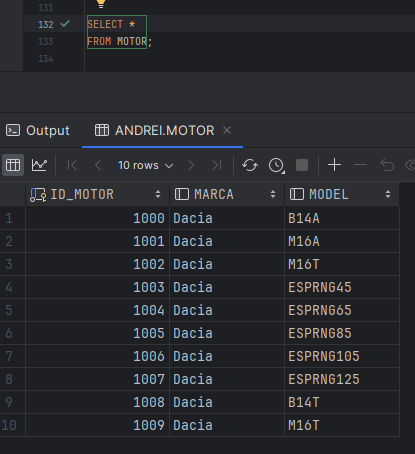
# **11. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea.**

(minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative; maxim 30 de înregistrări în fiecare tabel).

## Tabel MOTOR

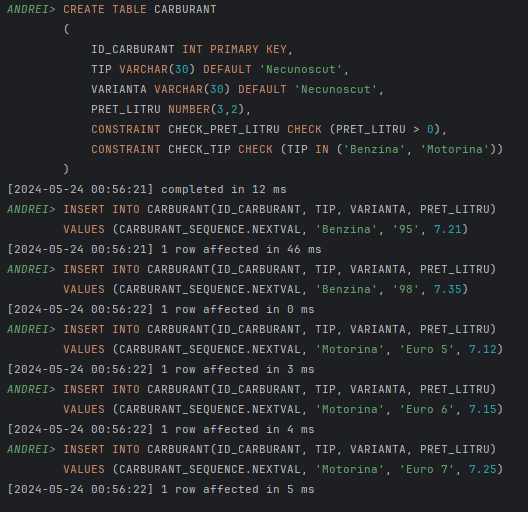
-- CREARE TABEL MOTOR  
CREATE TABLE MOTOR  
(  
 ID\_MOTOR INT PRIMARY KEY,  
 MARCA VARCHAR(30) DEFAULT 'NECUNOSCUT',  
 MODEL VARCHAR(30) DEFAULT 'NECUNOSCUT'  
);  
  
-- INSERARE INREGISTRARI IN TABEL MOTOR  
  
INSERT INTO MOTOR(ID\_MOTOR, MARCA, MODEL)  
VALUES (MOTOR\_SEQUENCE.NEXTVAL, 'Dacia', 'B14A');  
  
INSERT INTO MOTOR(ID\_MOTOR, MARCA, MODEL)  
VALUES (MOTOR\_SEQUENCE.NEXTVAL, 'Dacia', 'M16A');  
  
INSERT INTO MOTOR(ID\_MOTOR, MARCA, MODEL)  
VALUES (MOTOR\_SEQUENCE.NEXTVAL, 'Dacia', 'M16T');  
  
INSERT INTO MOTOR(ID\_MOTOR, MARCA, MODEL)  
VALUES (MOTOR\_SEQUENCE.NEXTVAL, 'Dacia', 'ESPRNG45');  
  
INSERT INTO MOTOR(ID\_MOTOR, MARCA, MODEL)  
VALUES (MOTOR\_SEQUENCE.NEXTVAL, 'Dacia', 'ESPRNG65');  
  
INSERT INTO MOTOR(ID\_MOTOR, MARCA, MODEL)  
VALUES (MOTOR\_SEQUENCE.NEXTVAL, 'Dacia', 'ESPRNG85');  
  
INSERT INTO MOTOR(ID\_MOTOR, MARCA, MODEL)  
VALUES (MOTOR\_SEQUENCE.NEXTVAL, 'Dacia', 'ESPRNG105');  
  
INSERT INTO MOTOR(ID\_MOTOR, MARCA, MODEL)  
VALUES (MOTOR\_SEQUENCE.NEXTVAL, 'Dacia', 'ESPRNG125');  
  
INSERT INTO MOTOR(ID\_MOTOR, MARCA, MODEL)  
VALUES (MOTOR\_SEQUENCE.NEXTVAL, 'Dacia', 'B14T');  
  
INSERT INTO MOTOR(ID\_MOTOR, MARCA, MODEL)  
VALUES (MOTOR\_SEQUENCE.NEXTVAL, 'Dacia', 'M16T');

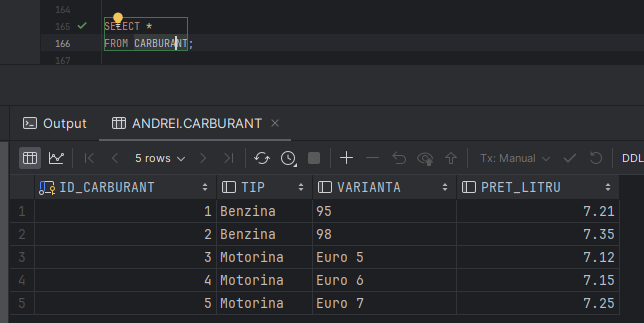




## Tabel CARBURANT

-- TABEL CARBURANT  
  
-- CREARE TABEL CARBURANT  
CREATE TABLE CARBURANT  
(  
 ID\_CARBURANT INT PRIMARY KEY,  
 TIP VARCHAR(30) DEFAULT 'Necunoscut',  
 VARIANTA VARCHAR(30) DEFAULT 'Necunoscut',  
 PRET\_LITRU NUMBER(3,2),  
 CONSTRAINT CHECK\_PRET\_LITRU CHECK (PRET\_LITRU > 0),  
 CONSTRAINT CHECK\_TIP CHECK (TIP IN ('Benzina', 'Motorina'))  
);  
  
-- INSERARE INREGISTRARI IN TABEL CARBURANT  
  
INSERT INTO CARBURANT(ID\_CARBURANT, TIP, VARIANTA, PRET\_LITRU)  
VALUES (CARBURANT\_SEQUENCE.NEXTVAL, 'Benzina', '95', 7.21);  
  
INSERT INTO CARBURANT(ID\_CARBURANT, TIP, VARIANTA, PRET\_LITRU)  
VALUES (CARBURANT\_SEQUENCE.NEXTVAL, 'Benzina', '98', 7.35);  
  
INSERT INTO CARBURANT(ID\_CARBURANT, TIP, VARIANTA, PRET\_LITRU)  
VALUES (CARBURANT\_SEQUENCE.NEXTVAL, 'Motorina', 'Euro 5', 7.12);  
  
INSERT INTO CARBURANT(ID\_CARBURANT, TIP, VARIANTA, PRET\_LITRU)  
VALUES (CARBURANT\_SEQUENCE.NEXTVAL, 'Motorina', 'Euro 6', 7.15);  
  
INSERT INTO CARBURANT(ID\_CARBURANT, TIP, VARIANTA, PRET\_LITRU)  
VALUES (CARBURANT\_SEQUENCE.NEXTVAL, 'Motorina', 'Euro 7', 7.25);

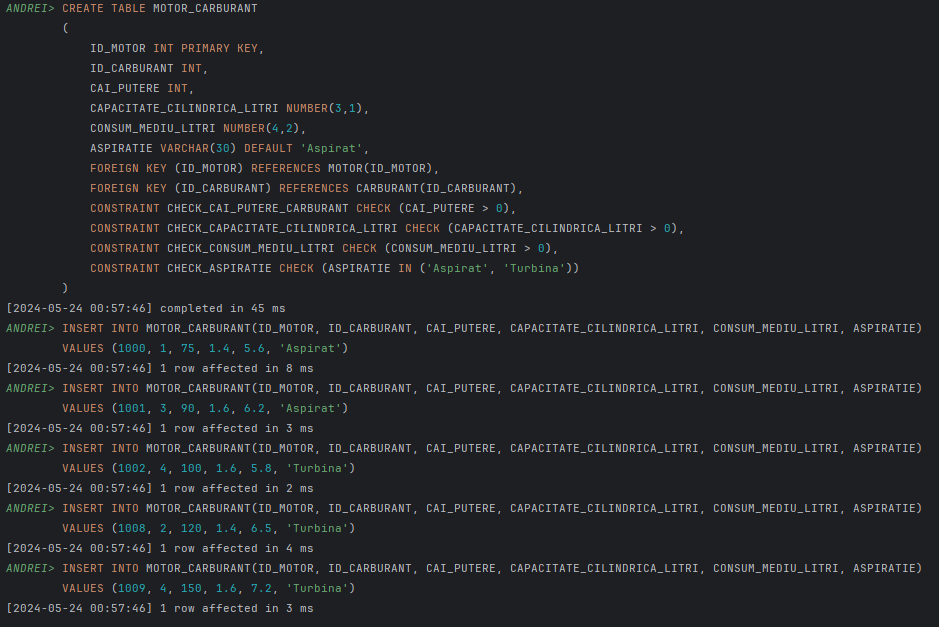


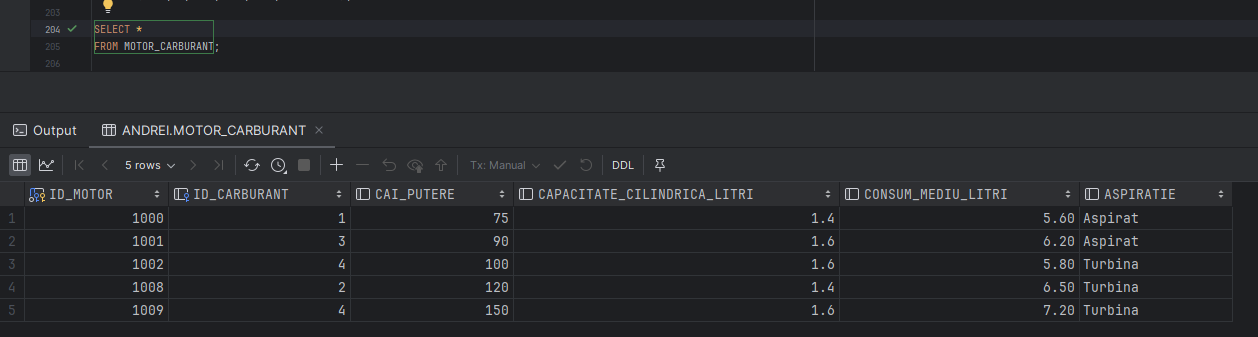


## Tabel MOTOR\_CARBURANT

-- CREARE TABEL MOTOR\_CARBURANT

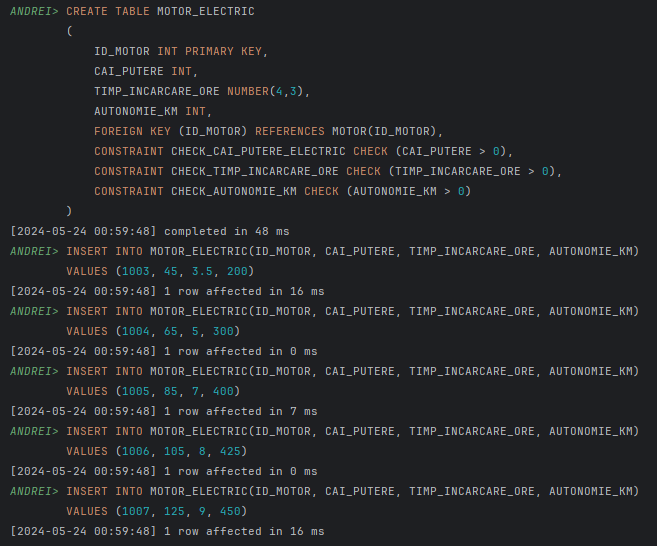
CREATE TABLE MOTOR\_CARBURANT  
(  
 ID\_MOTOR INT PRIMARY KEY,  
 ID\_CARBURANT INT,  
 CAI\_PUTERE INT,  
 CAPACITATE\_CILINDRICA\_LITRI NUMBER(3,1),  
 CONSUM\_MEDIU\_LITRI NUMBER(4,2),  
 ASPIRATIE VARCHAR(30) DEFAULT 'Aspirat',  
 FOREIGN KEY (ID\_MOTOR) REFERENCES MOTOR(ID\_MOTOR),  
 FOREIGN KEY (ID\_CARBURANT) REFERENCES CARBURANT(ID\_CARBURANT),  
 CONSTRAINT CHECK\_CAI\_PUTERE\_CARBURANT CHECK (CAI\_PUTERE > 0),  
 CONSTRAINT CHECK\_CAPACITATE\_CILINDRICA\_LITRI CHECK (CAPACITATE\_CILINDRICA\_LITRI > 0),  
 CONSTRAINT CHECK\_CONSUM\_MEDIU\_LITRI CHECK (CONSUM\_MEDIU\_LITRI > 0),  
 CONSTRAINT CHECK\_ASPIRATIE CHECK (ASPIRATIE IN ('Aspirat', 'Turbina'))  
);  
  
-- INSERARE INREGISTRARI IN TABEL MOTOR\_CARBURANT  
  
INSERT INTO MOTOR\_CARBURANT(ID\_MOTOR, ID\_CARBURANT, CAI\_PUTERE, CAPACITATE\_CILINDRICA\_LITRI, CONSUM\_MEDIU\_LITRI, ASPIRATIE)  
VALUES (1000, 1, 75, 1.4, 5.6, 'Aspirat');  
  
INSERT INTO MOTOR\_CARBURANT(ID\_MOTOR, ID\_CARBURANT, CAI\_PUTERE, CAPACITATE\_CILINDRICA\_LITRI, CONSUM\_MEDIU\_LITRI, ASPIRATIE)  
VALUES (1001, 3, 90, 1.6, 6.2, 'Aspirat');  
  
INSERT INTO MOTOR\_CARBURANT(ID\_MOTOR, ID\_CARBURANT, CAI\_PUTERE, CAPACITATE\_CILINDRICA\_LITRI, CONSUM\_MEDIU\_LITRI, ASPIRATIE)  
VALUES (1002, 4, 100, 1.6, 5.8, 'Turbina');  
  
INSERT INTO MOTOR\_CARBURANT(ID\_MOTOR, ID\_CARBURANT, CAI\_PUTERE, CAPACITATE\_CILINDRICA\_LITRI, CONSUM\_MEDIU\_LITRI, ASPIRATIE)  
VALUES (1008, 2, 120, 1.4, 6.5, 'Turbina');  
  
INSERT INTO MOTOR\_CARBURANT(ID\_MOTOR, ID\_CARBURANT, CAI\_PUTERE, CAPACITATE\_CILINDRICA\_LITRI, CONSUM\_MEDIU\_LITRI, ASPIRATIE)  
VALUES (1009, 4, 150, 1.6, 7.2, 'Turbina');

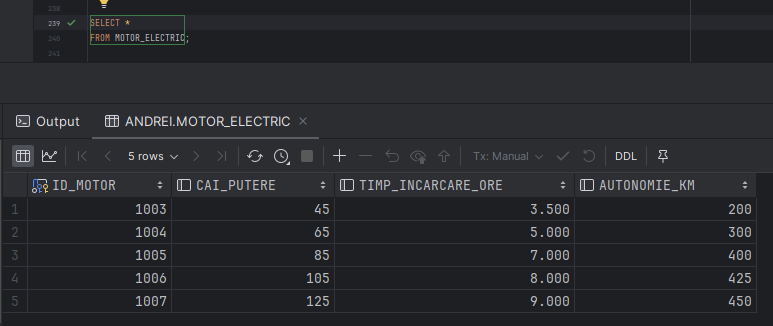




## Tabel MOTOR\_ELECTRIC

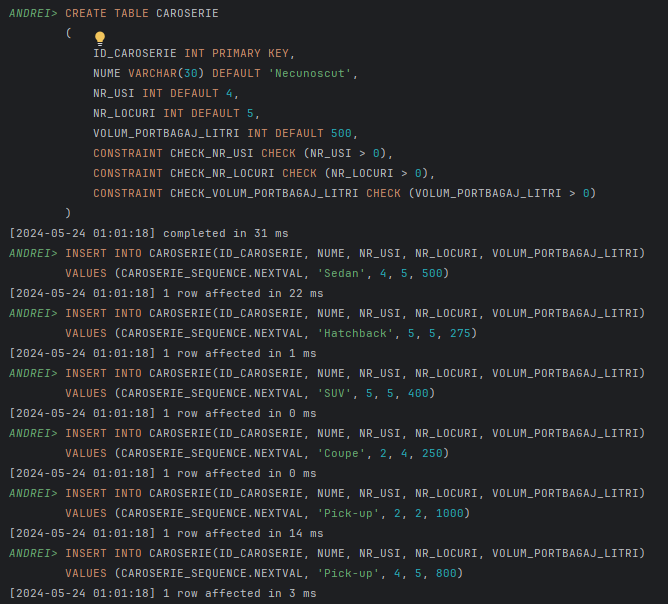
-- CREARE TABEL MOTOR\_ELECTRIC  
CREATE TABLE MOTOR\_ELECTRIC  
(  
 ID\_MOTOR INT PRIMARY KEY,  
 CAI\_PUTERE INT,  
 TIMP\_INCARCARE\_ORE NUMBER(4,3),  
 AUTONOMIE\_KM INT,  
 FOREIGN KEY (ID\_MOTOR) REFERENCES MOTOR(ID\_MOTOR),  
 CONSTRAINT CHECK\_CAI\_PUTERE\_ELECTRIC CHECK (CAI\_PUTERE > 0),  
 CONSTRAINT CHECK\_TIMP\_INCARCARE\_ORE CHECK (TIMP\_INCARCARE\_ORE > 0),  
 CONSTRAINT CHECK\_AUTONOMIE\_KM CHECK (AUTONOMIE\_KM > 0)  
);  
  
-- INSERARE INREGISTRARI IN TABEL MOTOR\_ELECTRIC  
  
INSERT INTO MOTOR\_ELECTRIC(ID\_MOTOR, CAI\_PUTERE, TIMP\_INCARCARE\_ORE, AUTONOMIE\_KM)  
VALUES (1003, 45, 3.5, 200);  
  
INSERT INTO MOTOR\_ELECTRIC(ID\_MOTOR, CAI\_PUTERE, TIMP\_INCARCARE\_ORE, AUTONOMIE\_KM)  
VALUES (1004, 65, 5, 300);  
  
INSERT INTO MOTOR\_ELECTRIC(ID\_MOTOR, CAI\_PUTERE, TIMP\_INCARCARE\_ORE, AUTONOMIE\_KM)  
VALUES (1005, 85, 7, 400);  
  
INSERT INTO MOTOR\_ELECTRIC(ID\_MOTOR, CAI\_PUTERE, TIMP\_INCARCARE\_ORE, AUTONOMIE\_KM)  
VALUES (1006, 105, 8, 425);  
  
INSERT INTO MOTOR\_ELECTRIC(ID\_MOTOR, CAI\_PUTERE, TIMP\_INCARCARE\_ORE, AUTONOMIE\_KM)  
VALUES (1007, 125, 9, 450);

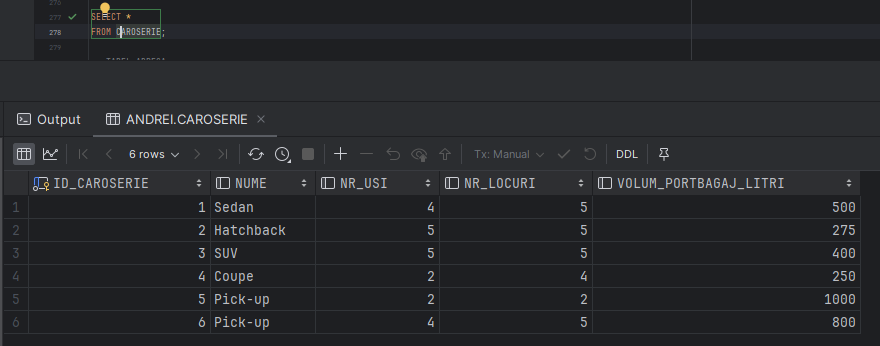




## Tabel CAROSERIE

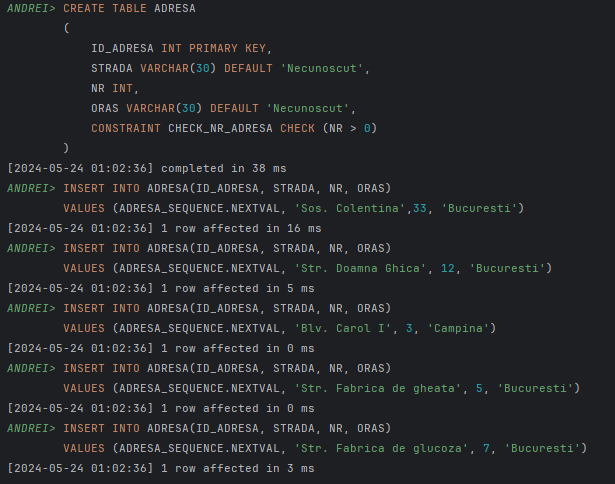
-- CREARE TABEL CAROSERIE  
CREATE TABLE CAROSERIE  
(  
 ID\_CAROSERIE INT PRIMARY KEY,  
 NUME VARCHAR(30) DEFAULT 'Necunoscut',  
 NR\_USI INT DEFAULT 4,  
 NR\_LOCURI INT DEFAULT 5,  
 VOLUM\_PORTBAGAJ\_LITRI INT DEFAULT 500,  
 CONSTRAINT CHECK\_NR\_USI CHECK (NR\_USI > 0),  
 CONSTRAINT CHECK\_NR\_LOCURI CHECK (NR\_LOCURI > 0),  
 CONSTRAINT CHECK\_VOLUM\_PORTBAGAJ\_LITRI CHECK (VOLUM\_PORTBAGAJ\_LITRI > 0)  
);  
  
-- INSERARE TABEL CAROSERIE  
  
INSERT INTO CAROSERIE(ID\_CAROSERIE, NUME, NR\_USI, NR\_LOCURI, VOLUM\_PORTBAGAJ\_LITRI)  
VALUES (CAROSERIE\_SEQUENCE.NEXTVAL, 'Sedan', 4, 5, 500);  
  
INSERT INTO CAROSERIE(ID\_CAROSERIE, NUME, NR\_USI, NR\_LOCURI, VOLUM\_PORTBAGAJ\_LITRI)  
VALUES (CAROSERIE\_SEQUENCE.NEXTVAL, 'Hatchback', 5, 5, 275);  
  
INSERT INTO CAROSERIE(ID\_CAROSERIE, NUME, NR\_USI, NR\_LOCURI, VOLUM\_PORTBAGAJ\_LITRI)  
VALUES (CAROSERIE\_SEQUENCE.NEXTVAL, 'SUV', 5, 5, 400);  
  
INSERT INTO CAROSERIE(ID\_CAROSERIE, NUME, NR\_USI, NR\_LOCURI, VOLUM\_PORTBAGAJ\_LITRI)  
VALUES (CAROSERIE\_SEQUENCE.NEXTVAL, 'Coupe', 2, 4, 250);  
  
INSERT INTO CAROSERIE(ID\_CAROSERIE, NUME, NR\_USI, NR\_LOCURI, VOLUM\_PORTBAGAJ\_LITRI)  
VALUES (CAROSERIE\_SEQUENCE.NEXTVAL, 'Pick-up', 2, 2, 1000);  
  
INSERT INTO CAROSERIE(ID\_CAROSERIE, NUME, NR\_USI, NR\_LOCURI, VOLUM\_PORTBAGAJ\_LITRI)  
VALUES (CAROSERIE\_SEQUENCE.NEXTVAL, 'Pick-up', 4, 5, 800);

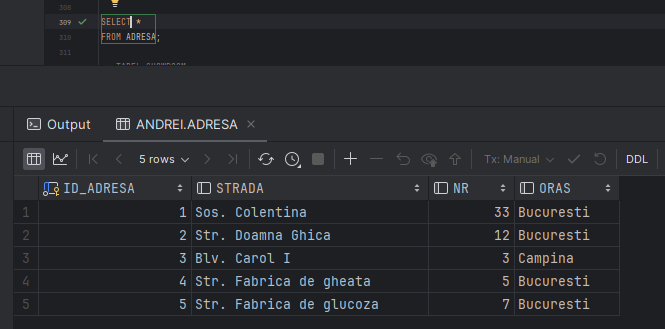




## Tabel ADRESA

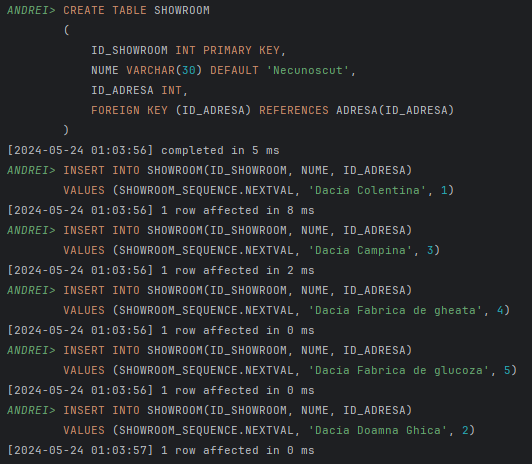
-- CREARE TABEL ADRESA  
CREATE TABLE ADRESA  
(  
 ID\_ADRESA INT PRIMARY KEY,  
 STRADA VARCHAR(30) DEFAULT 'Necunoscut',  
 NR INT,  
 ORAS VARCHAR(30) DEFAULT 'Necunoscut',  
 CONSTRAINT CHECK\_NR\_ADRESA CHECK (NR > 0)  
);  
  
-- INSERARE INREGISTRARI IN TABEL ADRESA  
  
INSERT INTO ADRESA(ID\_ADRESA, STRADA, NR, ORAS)  
VALUES (ADRESA\_SEQUENCE.NEXTVAL, 'Sos. Colentina',33, 'Bucuresti');  
  
INSERT INTO ADRESA(ID\_ADRESA, STRADA, NR, ORAS)  
VALUES (ADRESA\_SEQUENCE.NEXTVAL, 'Str. Doamna Ghica', 12, 'Bucuresti');  
  
INSERT INTO ADRESA(ID\_ADRESA, STRADA, NR, ORAS)  
VALUES (ADRESA\_SEQUENCE.NEXTVAL, 'Blvd. Carol I', 3, 'Campina');  
  
INSERT INTO ADRESA(ID\_ADRESA, STRADA, NR, ORAS)  
VALUES (ADRESA\_SEQUENCE.NEXTVAL, 'Str. Fabrica de gheata', 5, 'Bucuresti');  
  
INSERT INTO ADRESA(ID\_ADRESA, STRADA, NR, ORAS)  
VALUES (ADRESA\_SEQUENCE.NEXTVAL, 'Str. Fabrica de glucoza', 7, 'Bucuresti');

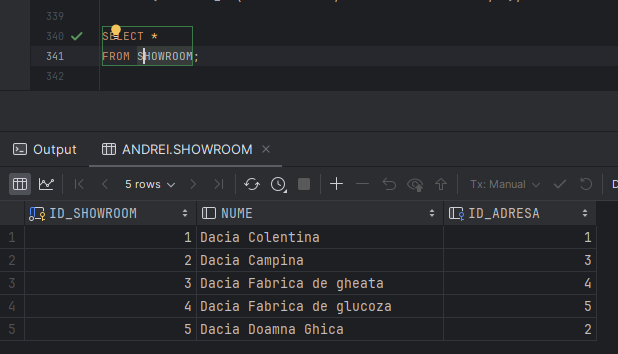




## Tabel SHOWROOM

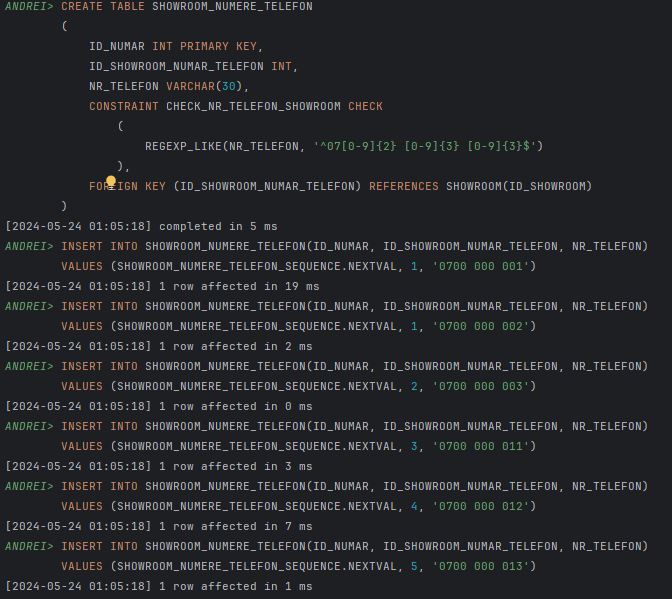
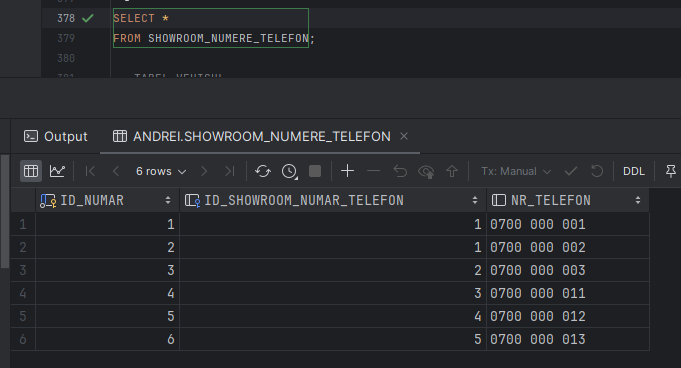
-- CREARE TABEL SHOWROOM  
CREATE TABLE SHOWROOM  
(  
 ID\_SHOWROOM INT PRIMARY KEY,  
 NUME VARCHAR(30) DEFAULT 'Necunoscut',  
 ID\_ADRESA INT,  
 FOREIGN KEY (ID\_ADRESA) REFERENCES ADRESA(ID\_ADRESA)  
);  
  
-- INSERARE TABEL SHOWROOM  
  
INSERT INTO SHOWROOM(ID\_SHOWROOM, NUME, ID\_ADRESA)  
VALUES (SHOWROOM\_SEQUENCE.NEXTVAL, 'Dacia Colentina', 1);  
  
INSERT INTO SHOWROOM(ID\_SHOWROOM, NUME, ID\_ADRESA)  
VALUES (SHOWROOM\_SEQUENCE.NEXTVAL, 'Dacia Campina', 3);  
  
INSERT INTO SHOWROOM(ID\_SHOWROOM, NUME, ID\_ADRESA)  
VALUES (SHOWROOM\_SEQUENCE.NEXTVAL, 'Dacia Fabrica de gheata', 4);  
  
INSERT INTO SHOWROOM(ID\_SHOWROOM, NUME, ID\_ADRESA)  
VALUES (SHOWROOM\_SEQUENCE.NEXTVAL, 'Dacia Fabrica de glucoza', 5);  
  
INSERT INTO SHOWROOM(ID\_SHOWROOM, NUME, ID\_ADRESA)  
VALUES (SHOWROOM\_SEQUENCE.NEXTVAL, 'Dacia Doamna Ghica', 2);





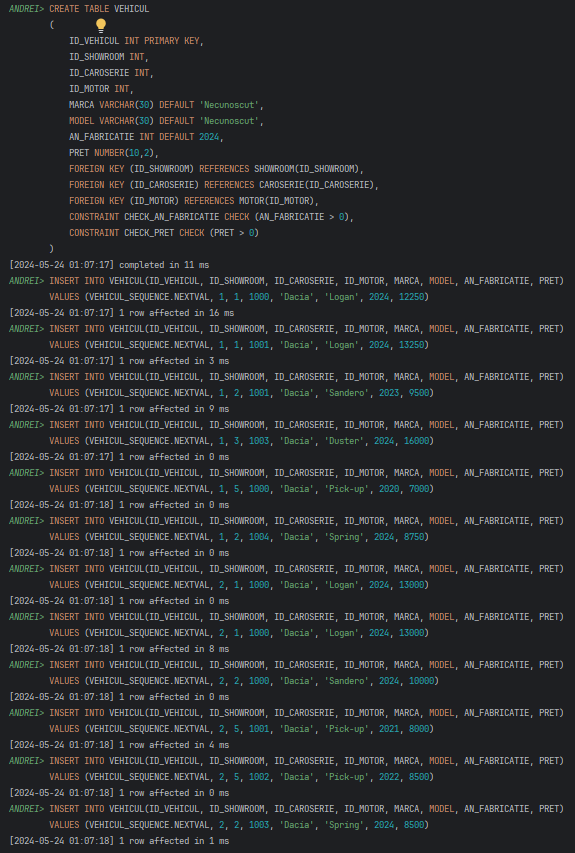
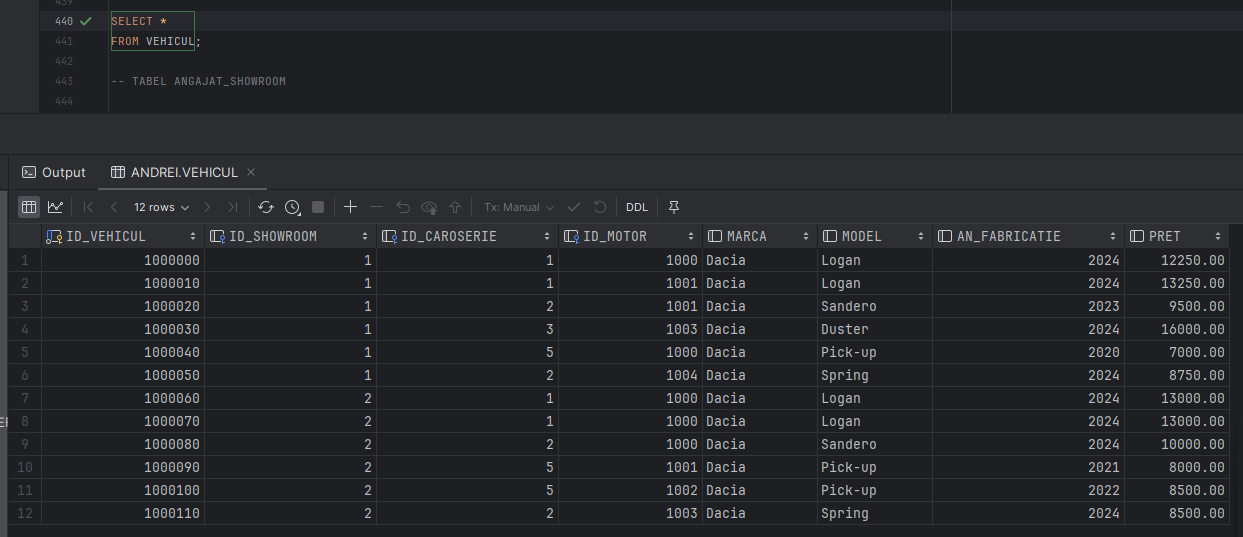
## Tabel SHOWROOM\_NUMERE\_TELEFON

-- CREARE TABEL SHOWROOM\_NUMERE\_TELEFON  
CREATE TABLE SHOWROOM\_NUMERE\_TELEFON  
(  
 ID\_NUMAR INT PRIMARY KEY,  
 ID\_SHOWROOM\_NUMAR\_TELEFON INT,  
 NR\_TELEFON VARCHAR(30),  
 CONSTRAINT CHECK\_NR\_TELEFON\_SHOWROOM CHECK  
 (  
 *REGEXP\_LIKE*(NR\_TELEFON, '^07[0-9]{2} [0-9]{3} [0-9]{3}$')  
 ),  
 FOREIGN KEY (ID\_SHOWROOM\_NUMAR\_TELEFON) REFERENCES SHOWROOM(ID\_SHOWROOM)  
);  
  
-- INSERARE TABEL SHOWROOM\_NUMERE\_TELEFON  
  
INSERT INTO SHOWROOM\_NUMERE\_TELEFON(ID\_NUMAR, ID\_SHOWROOM\_NUMAR\_TELEFON, NR\_TELEFON)  
VALUES (SHOWROOM\_NUMERE\_TELEFON\_SEQUENCE.NEXTVAL, 1, '0700 000 001');  
  
INSERT INTO SHOWROOM\_NUMERE\_TELEFON(ID\_NUMAR, ID\_SHOWROOM\_NUMAR\_TELEFON, NR\_TELEFON)  
VALUES (SHOWROOM\_NUMERE\_TELEFON\_SEQUENCE.NEXTVAL, 1, '0700 000 002');  
  
INSERT INTO SHOWROOM\_NUMERE\_TELEFON(ID\_NUMAR, ID\_SHOWROOM\_NUMAR\_TELEFON, NR\_TELEFON)  
VALUES (SHOWROOM\_NUMERE\_TELEFON\_SEQUENCE.NEXTVAL, 2, '0700 000 003');  
  
INSERT INTO SHOWROOM\_NUMERE\_TELEFON(ID\_NUMAR, ID\_SHOWROOM\_NUMAR\_TELEFON, NR\_TELEFON)  
VALUES (SHOWROOM\_NUMERE\_TELEFON\_SEQUENCE.NEXTVAL, 3, '0700 000 011');  
  
INSERT INTO SHOWROOM\_NUMERE\_TELEFON(ID\_NUMAR, ID\_SHOWROOM\_NUMAR\_TELEFON, NR\_TELEFON)  
VALUES (SHOWROOM\_NUMERE\_TELEFON\_SEQUENCE.NEXTVAL, 4, '0700 000 012');  
  
INSERT INTO SHOWROOM\_NUMERE\_TELEFON(ID\_NUMAR, ID\_SHOWROOM\_NUMAR\_TELEFON, NR\_TELEFON)  
VALUES (SHOWROOM\_NUMERE\_TELEFON\_SEQUENCE.NEXTVAL, 5, '0700 000 013');

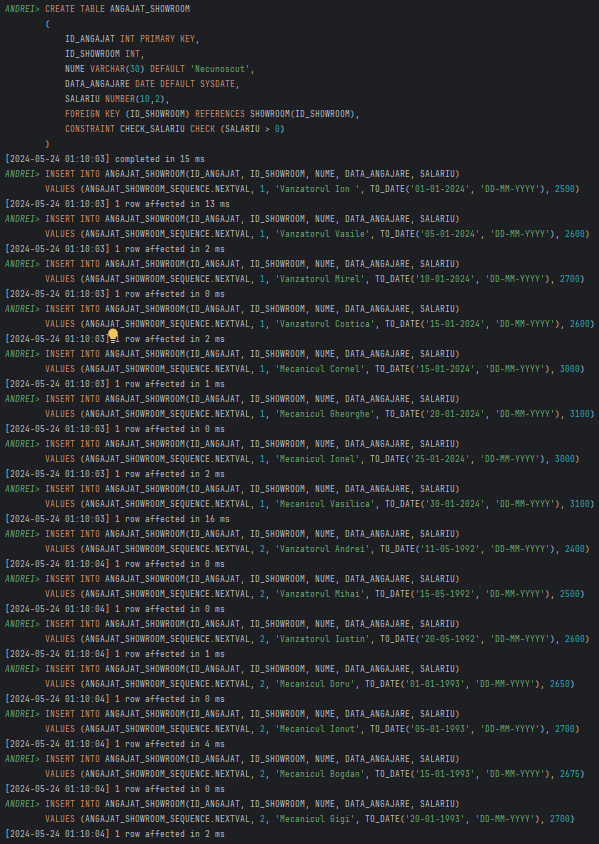
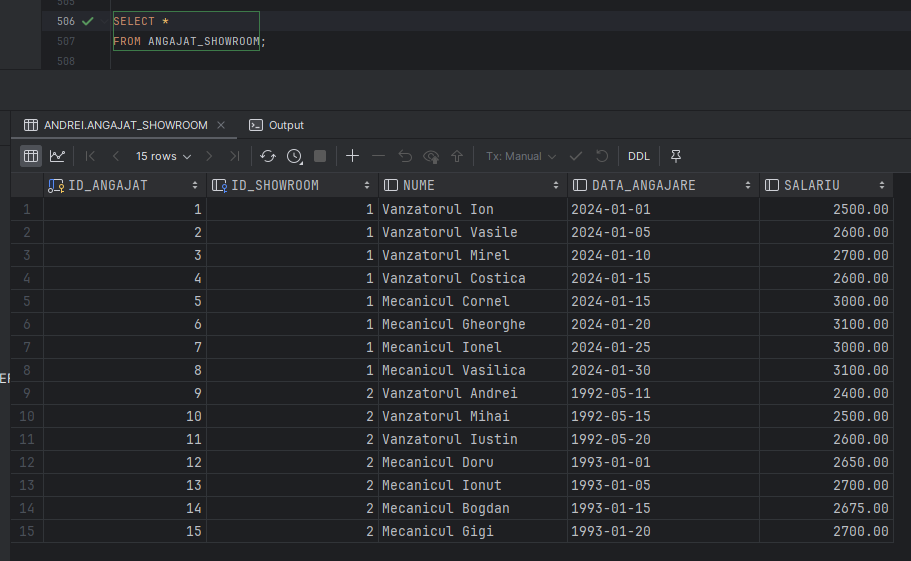
## Tabel VEHICUL

-- CREARE TABEL VEHICUL  
CREATE TABLE VEHICUL  
(  
 ID\_VEHICUL INT PRIMARY KEY,  
 ID\_SHOWROOM INT,  
 ID\_CAROSERIE INT,  
 ID\_MOTOR INT,  
 MARCA VARCHAR(30) DEFAULT 'Necunoscut',  
 MODEL VARCHAR(30) DEFAULT 'Necunoscut',  
 AN\_FABRICATIE INT DEFAULT 2024,  
 PRET NUMBER(10,2),  
 FOREIGN KEY (ID\_SHOWROOM) REFERENCES SHOWROOM(ID\_SHOWROOM),  
 FOREIGN KEY (ID\_CAROSERIE) REFERENCES CAROSERIE(ID\_CAROSERIE),  
 FOREIGN KEY (ID\_MOTOR) REFERENCES MOTOR(ID\_MOTOR),  
 CONSTRAINT CHECK\_AN\_FABRICATIE CHECK (AN\_FABRICATIE > 0),  
 CONSTRAINT CHECK\_PRET CHECK (PRET > 0)  
);  
  
-- INSERARE TABEL VEHICUL  
  
INSERT INTO VEHICUL(ID\_VEHICUL, ID\_SHOWROOM, ID\_CAROSERIE, ID\_MOTOR, MARCA, MODEL, AN\_FABRICATIE, PRET)  
VALUES (VEHICUL\_SEQUENCE.NEXTVAL, 1, 1, 1000, 'Dacia', 'Logan', 2024, 12250);  
  
INSERT INTO VEHICUL(ID\_VEHICUL, ID\_SHOWROOM, ID\_CAROSERIE, ID\_MOTOR, MARCA, MODEL, AN\_FABRICATIE, PRET)  
VALUES (VEHICUL\_SEQUENCE.NEXTVAL, 1, 1, 1001, 'Dacia', 'Logan', 2024, 13250);  
  
INSERT INTO VEHICUL(ID\_VEHICUL, ID\_SHOWROOM, ID\_CAROSERIE, ID\_MOTOR, MARCA, MODEL, AN\_FABRICATIE, PRET)  
VALUES (VEHICUL\_SEQUENCE.NEXTVAL, 1, 2, 1001, 'Dacia', 'Sandero', 2023, 9500);  
  
INSERT INTO VEHICUL(ID\_VEHICUL, ID\_SHOWROOM, ID\_CAROSERIE, ID\_MOTOR, MARCA, MODEL, AN\_FABRICATIE, PRET)  
VALUES (VEHICUL\_SEQUENCE.NEXTVAL, 1, 3, 1003, 'Dacia', 'Duster', 2024, 16000);  
  
INSERT INTO VEHICUL(ID\_VEHICUL, ID\_SHOWROOM, ID\_CAROSERIE, ID\_MOTOR, MARCA, MODEL, AN\_FABRICATIE, PRET)  
VALUES (VEHICUL\_SEQUENCE.NEXTVAL, 1, 5, 1000, 'Dacia', 'Pick-up', 2020, 7000);  
  
INSERT INTO VEHICUL(ID\_VEHICUL, ID\_SHOWROOM, ID\_CAROSERIE, ID\_MOTOR, MARCA, MODEL, AN\_FABRICATIE, PRET)  
VALUES (VEHICUL\_SEQUENCE.NEXTVAL, 1, 2, 1004, 'Dacia', 'Spring', 2024, 8750);  
  
  
INSERT INTO VEHICUL(ID\_VEHICUL, ID\_SHOWROOM, ID\_CAROSERIE, ID\_MOTOR, MARCA, MODEL, AN\_FABRICATIE, PRET)  
VALUES (VEHICUL\_SEQUENCE.NEXTVAL, 2, 1, 1000, 'Dacia', 'Logan', 2024, 13000);  
  
INSERT INTO VEHICUL(ID\_VEHICUL, ID\_SHOWROOM, ID\_CAROSERIE, ID\_MOTOR, MARCA, MODEL, AN\_FABRICATIE, PRET)  
VALUES (VEHICUL\_SEQUENCE.NEXTVAL, 2, 1, 1000, 'Dacia', 'Logan', 2024, 13000);  
  
INSERT INTO VEHICUL(ID\_VEHICUL, ID\_SHOWROOM, ID\_CAROSERIE, ID\_MOTOR, MARCA, MODEL, AN\_FABRICATIE, PRET)  
VALUES (VEHICUL\_SEQUENCE.NEXTVAL, 2, 2, 1000, 'Dacia', 'Sandero', 2024, 10000);  
  
INSERT INTO VEHICUL(ID\_VEHICUL, ID\_SHOWROOM, ID\_CAROSERIE, ID\_MOTOR, MARCA, MODEL, AN\_FABRICATIE, PRET)  
VALUES (VEHICUL\_SEQUENCE.NEXTVAL, 2, 5, 1001, 'Dacia', 'Pick-up', 2021, 8000);  
  
INSERT INTO VEHICUL(ID\_VEHICUL, ID\_SHOWROOM, ID\_CAROSERIE, ID\_MOTOR, MARCA, MODEL, AN\_FABRICATIE, PRET)  
VALUES (VEHICUL\_SEQUENCE.NEXTVAL, 2, 5, 1002, 'Dacia', 'Pick-up', 2022, 8500);  
  
INSERT INTO VEHICUL(ID\_VEHICUL, ID\_SHOWROOM, ID\_CAROSERIE, ID\_MOTOR, MARCA, MODEL, AN\_FABRICATIE, PRET)  
VALUES (VEHICUL\_SEQUENCE.NEXTVAL, 2, 2, 1003, 'Dacia', 'Spring', 2024, 8500);

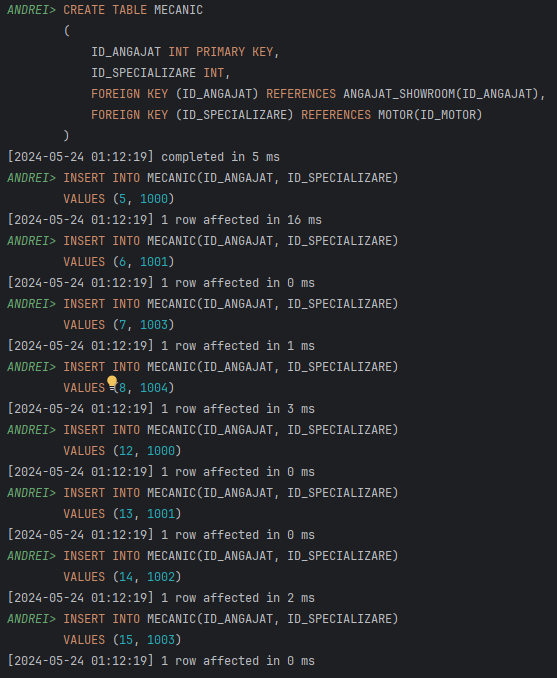
## Tabel ANGAJAT\_SHOWROOM

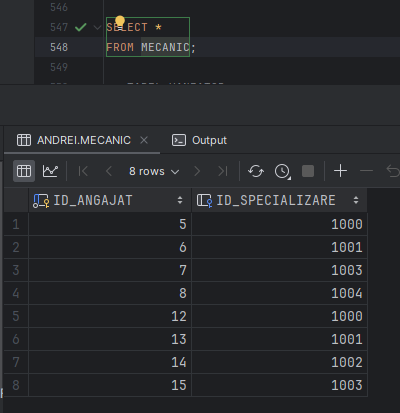
-- CREARE TABEL ANGAJAT\_SHOWROOM  
CREATE TABLE ANGAJAT\_SHOWROOM  
(  
 ID\_ANGAJAT INT PRIMARY KEY,  
 ID\_SHOWROOM INT,  
 NUME VARCHAR(30) DEFAULT 'Necunoscut',  
 DATA\_ANGAJARE DATE DEFAULT *SYSDATE*,  
 SALARIU NUMBER(10,2),  
 FOREIGN KEY (ID\_SHOWROOM) REFERENCES SHOWROOM(ID\_SHOWROOM),  
 CONSTRAINT CHECK\_SALARIU CHECK (SALARIU > 0)  
);  
  
-- INSERARE INREGISTRARI IN TABEL ANGAJAT\_SHOWROOM  
  
--SHOWROOM 1  
INSERT INTO ANGAJAT\_SHOWROOM(ID\_ANGAJAT, ID\_SHOWROOM, NUME, DATA\_ANGAJARE, SALARIU)  
VALUES (ANGAJAT\_SHOWROOM\_SEQUENCE.NEXTVAL, 1, 'Vanzatorul Ion', *TO\_DATE*('01-01-2024', 'DD-MM-YYYY'), 2500);  
  
INSERT INTO ANGAJAT\_SHOWROOM(ID\_ANGAJAT, ID\_SHOWROOM, NUME, DATA\_ANGAJARE, SALARIU)  
VALUES (ANGAJAT\_SHOWROOM\_SEQUENCE.NEXTVAL, 1, 'Vanzatorul Vasile', *TO\_DATE*('05-01-2024', 'DD-MM-YYYY'), 2600);  
  
INSERT INTO ANGAJAT\_SHOWROOM(ID\_ANGAJAT, ID\_SHOWROOM, NUME, DATA\_ANGAJARE, SALARIU)  
VALUES (ANGAJAT\_SHOWROOM\_SEQUENCE.NEXTVAL, 1, 'Vanzatorul Mirel', *TO\_DATE*('10-01-2024', 'DD-MM-YYYY'), 2700);  
  
INSERT INTO ANGAJAT\_SHOWROOM(ID\_ANGAJAT, ID\_SHOWROOM, NUME, DATA\_ANGAJARE, SALARIU)  
VALUES (ANGAJAT\_SHOWROOM\_SEQUENCE.NEXTVAL, 1, 'Vanzatorul Costica', *TO\_DATE*('15-01-2024', 'DD-MM-YYYY'), 2600);  
  
INSERT INTO ANGAJAT\_SHOWROOM(ID\_ANGAJAT, ID\_SHOWROOM, NUME, DATA\_ANGAJARE, SALARIU)  
VALUES (ANGAJAT\_SHOWROOM\_SEQUENCE.NEXTVAL, 1, 'Mecanicul Cornel', *TO\_DATE*('15-01-2024', 'DD-MM-YYYY'), 3000);  
  
INSERT INTO ANGAJAT\_SHOWROOM(ID\_ANGAJAT, ID\_SHOWROOM, NUME, DATA\_ANGAJARE, SALARIU)  
VALUES (ANGAJAT\_SHOWROOM\_SEQUENCE.NEXTVAL, 1, 'Mecanicul Gheorghe', *TO\_DATE*('20-01-2024', 'DD-MM-YYYY'), 3100);  
  
INSERT INTO ANGAJAT\_SHOWROOM(ID\_ANGAJAT, ID\_SHOWROOM, NUME, DATA\_ANGAJARE, SALARIU)  
VALUES (ANGAJAT\_SHOWROOM\_SEQUENCE.NEXTVAL, 1, 'Mecanicul Ionel', *TO\_DATE*('25-01-2024', 'DD-MM-YYYY'), 3000);  
  
INSERT INTO ANGAJAT\_SHOWROOM(ID\_ANGAJAT, ID\_SHOWROOM, NUME, DATA\_ANGAJARE, SALARIU)  
VALUES (ANGAJAT\_SHOWROOM\_SEQUENCE.NEXTVAL, 1, 'Mecanicul Vasilica', *TO\_DATE*('30-01-2024', 'DD-MM-YYYY'), 3100);  
  
--SHOWROOM 2  
INSERT INTO ANGAJAT\_SHOWROOM(ID\_ANGAJAT, ID\_SHOWROOM, NUME, DATA\_ANGAJARE, SALARIU)  
VALUES (ANGAJAT\_SHOWROOM\_SEQUENCE.NEXTVAL, 2, 'Vanzatorul Andrei', *TO\_DATE*('11-05-1992', 'DD-MM-YYYY'), 2400);  
  
INSERT INTO ANGAJAT\_SHOWROOM(ID\_ANGAJAT, ID\_SHOWROOM, NUME, DATA\_ANGAJARE, SALARIU)  
VALUES (ANGAJAT\_SHOWROOM\_SEQUENCE.NEXTVAL, 2, 'Vanzatorul Mihai', *TO\_DATE*('15-05-1992', 'DD-MM-YYYY'), 2500);  
  
INSERT INTO ANGAJAT\_SHOWROOM(ID\_ANGAJAT, ID\_SHOWROOM, NUME, DATA\_ANGAJARE, SALARIU)  
VALUES (ANGAJAT\_SHOWROOM\_SEQUENCE.NEXTVAL, 2, 'Vanzatorul Iustin', *TO\_DATE*('20-05-1992', 'DD-MM-YYYY'), 2600);  
  
INSERT INTO ANGAJAT\_SHOWROOM(ID\_ANGAJAT, ID\_SHOWROOM, NUME, DATA\_ANGAJARE, SALARIU)  
VALUES (ANGAJAT\_SHOWROOM\_SEQUENCE.NEXTVAL, 2, 'Mecanicul Doru', *TO\_DATE*('01-01-1993', 'DD-MM-YYYY'), 2650);  
  
INSERT INTO ANGAJAT\_SHOWROOM(ID\_ANGAJAT, ID\_SHOWROOM, NUME, DATA\_ANGAJARE, SALARIU)  
VALUES (ANGAJAT\_SHOWROOM\_SEQUENCE.NEXTVAL, 2, 'Mecanicul Ionut', *TO\_DATE*('05-01-1993', 'DD-MM-YYYY'), 2700);  
  
INSERT INTO ANGAJAT\_SHOWROOM(ID\_ANGAJAT, ID\_SHOWROOM, NUME, DATA\_ANGAJARE, SALARIU)  
VALUES (ANGAJAT\_SHOWROOM\_SEQUENCE.NEXTVAL, 2, 'Mecanicul Bogdan', *TO\_DATE*('15-01-1993', 'DD-MM-YYYY'), 2675);  
  
INSERT INTO ANGAJAT\_SHOWROOM(ID\_ANGAJAT, ID\_SHOWROOM, NUME, DATA\_ANGAJARE, SALARIU)  
VALUES (ANGAJAT\_SHOWROOM\_SEQUENCE.NEXTVAL, 2, 'Mecanicul Gigi', *TO\_DATE*('20-01-1993', 'DD-MM-YYYY'), 2700);

## Tabel MECANIC

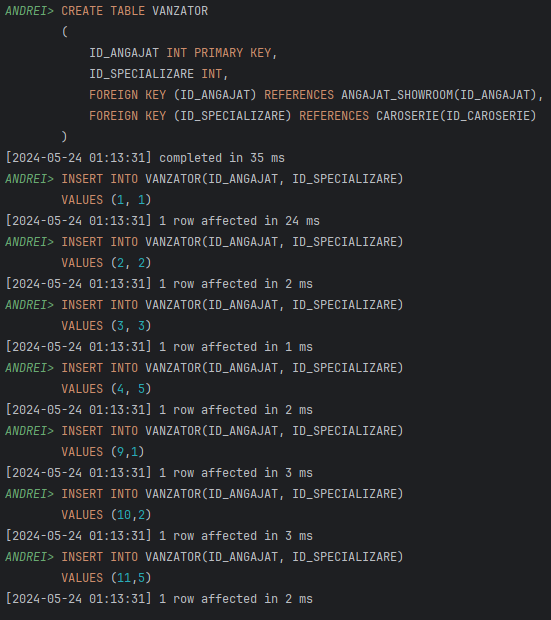
-- CREARE TABEL MECANIC  
CREATE TABLE MECANIC  
(  
 ID\_ANGAJAT INT PRIMARY KEY,  
 ID\_SPECIALIZARE INT,  
 FOREIGN KEY (ID\_ANGAJAT) REFERENCES ANGAJAT\_SHOWROOM(ID\_ANGAJAT),  
 FOREIGN KEY (ID\_SPECIALIZARE) REFERENCES MOTOR(ID\_MOTOR)  
);  
  
-- INSERARE INREGISTRARI IN TABEL MECANIC  
  
INSERT INTO MECANIC(ID\_ANGAJAT, ID\_SPECIALIZARE)  
VALUES (5, 1000);  
  
INSERT INTO MECANIC(ID\_ANGAJAT, ID\_SPECIALIZARE)  
VALUES (6, 1001);  
  
INSERT INTO MECANIC(ID\_ANGAJAT, ID\_SPECIALIZARE)  
VALUES (7, 1003);  
  
INSERT INTO MECANIC(ID\_ANGAJAT, ID\_SPECIALIZARE)  
VALUES (8, 1004);  
  
  
INSERT INTO MECANIC(ID\_ANGAJAT, ID\_SPECIALIZARE)  
VALUES (12, 1000);  
  
INSERT INTO MECANIC(ID\_ANGAJAT, ID\_SPECIALIZARE)  
VALUES (13, 1001);  
  
INSERT INTO MECANIC(ID\_ANGAJAT, ID\_SPECIALIZARE)  
VALUES (14, 1002);  
  
INSERT INTO MECANIC(ID\_ANGAJAT, ID\_SPECIALIZARE)  
VALUES (15, 1003);

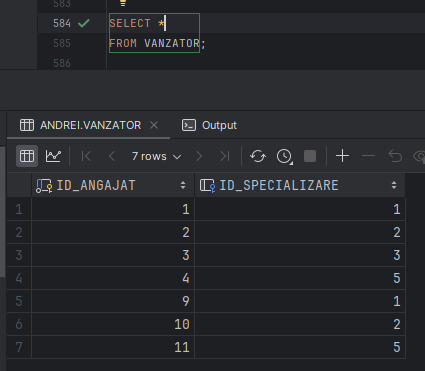




## Tabel VANZATOR

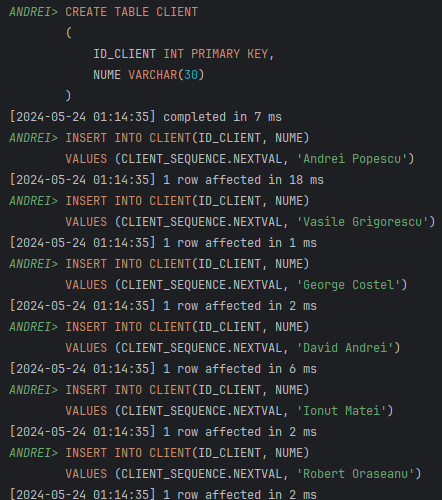
-- CREARE TABEL VANZATOR  
CREATE TABLE VANZATOR  
(  
 ID\_ANGAJAT INT PRIMARY KEY,  
 ID\_SPECIALIZARE INT,  
 FOREIGN KEY (ID\_ANGAJAT) REFERENCES ANGAJAT\_SHOWROOM(ID\_ANGAJAT),  
 FOREIGN KEY (ID\_SPECIALIZARE) REFERENCES CAROSERIE(ID\_CAROSERIE)  
);  
  
-- INSERARE INREGISTRARI IN TABEL VANZATOR  
  
INSERT INTO VANZATOR(ID\_ANGAJAT, ID\_SPECIALIZARE)  
VALUES (1, 1);  
  
INSERT INTO VANZATOR(ID\_ANGAJAT, ID\_SPECIALIZARE)  
VALUES (2, 2);  
  
INSERT INTO VANZATOR(ID\_ANGAJAT, ID\_SPECIALIZARE)  
VALUES (3, 3);  
  
INSERT INTO VANZATOR(ID\_ANGAJAT, ID\_SPECIALIZARE)  
VALUES (4, 5);  
  
INSERT INTO VANZATOR(ID\_ANGAJAT, ID\_SPECIALIZARE)  
VALUES (9,1);  
  
INSERT INTO VANZATOR(ID\_ANGAJAT, ID\_SPECIALIZARE)  
VALUES (10,2);  
  
INSERT INTO VANZATOR(ID\_ANGAJAT, ID\_SPECIALIZARE)  
VALUES (11,5);

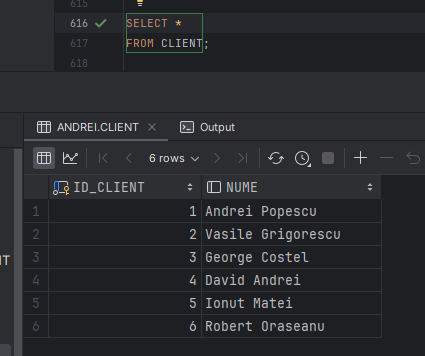




## Tabel CLIENT

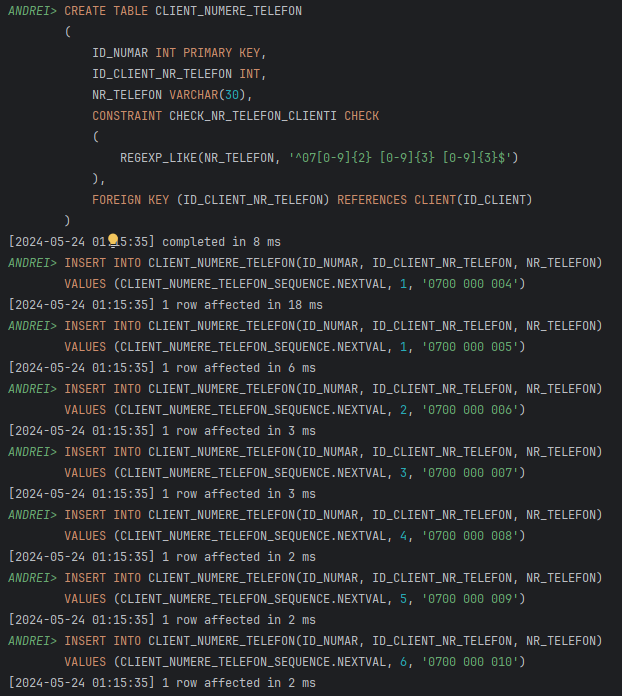
-- CREARE TABEL CLIENT  
CREATE TABLE CLIENT  
(  
 ID\_CLIENT INT PRIMARY KEY,  
 NUME VARCHAR(30)  
);  
  
-- INSERARE INREGISTRARI IN TABEL CLIENT  
  
INSERT INTO CLIENT(ID\_CLIENT, NUME)  
VALUES (CLIENT\_SEQUENCE.NEXTVAL, 'Andrei Popescu');  
  
INSERT INTO CLIENT(ID\_CLIENT, NUME)  
VALUES (CLIENT\_SEQUENCE.NEXTVAL, 'Vasile Grigorescu');  
  
INSERT INTO CLIENT(ID\_CLIENT, NUME)  
VALUES (CLIENT\_SEQUENCE.NEXTVAL, 'George Costel');  
  
INSERT INTO CLIENT(ID\_CLIENT, NUME)  
VALUES (CLIENT\_SEQUENCE.NEXTVAL, 'David Andrei');  
  
INSERT INTO CLIENT(ID\_CLIENT, NUME)  
VALUES (CLIENT\_SEQUENCE.NEXTVAL, 'Ionut Matei');  
  
INSERT INTO CLIENT(ID\_CLIENT, NUME)  
VALUES (CLIENT\_SEQUENCE.NEXTVAL, 'Robert Oraseanu');

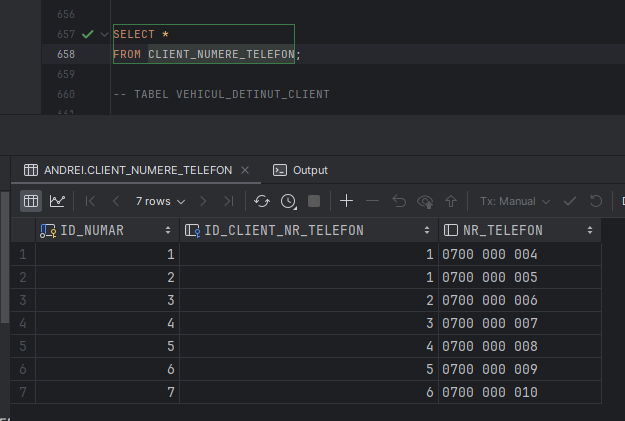




## Tabel CLIENT\_NUMERE\_TELEFON

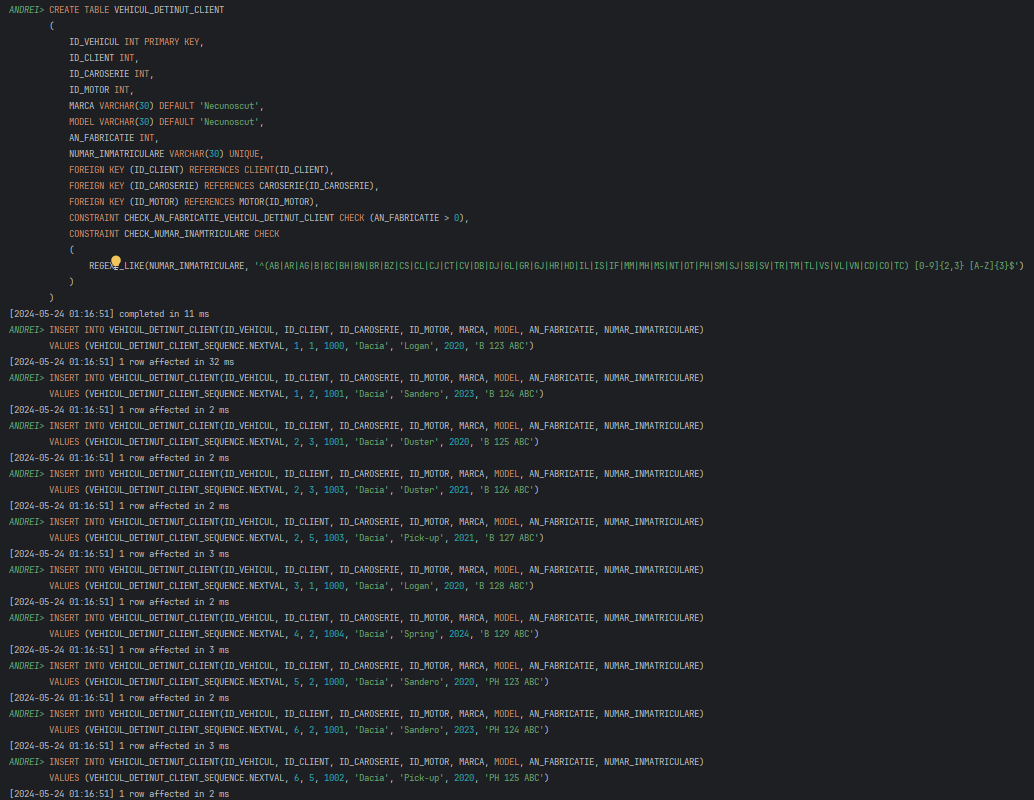
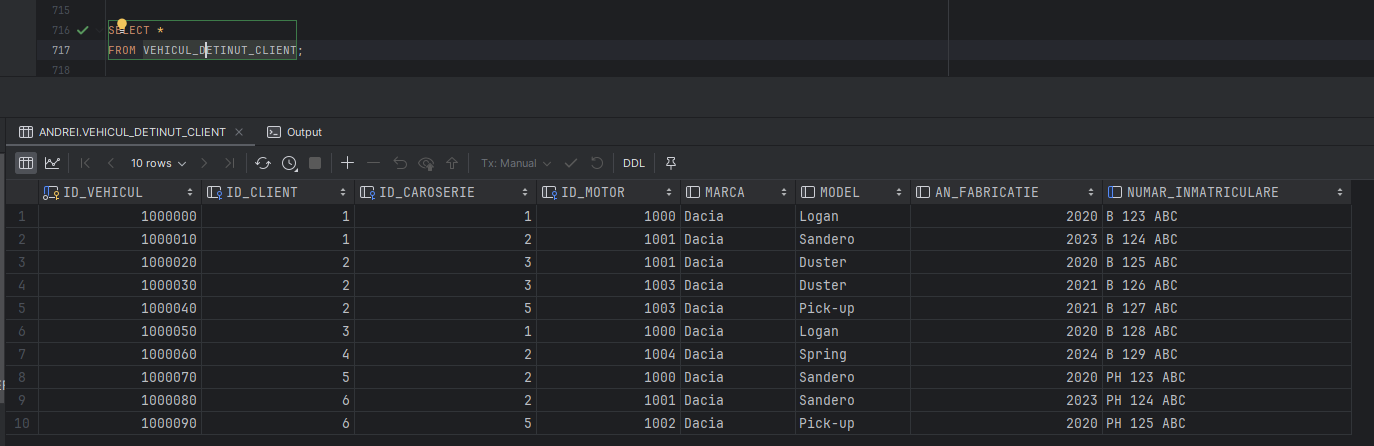
-- CREARE TABEL CLIENT\_NUMERE\_TELEFON  
CREATE TABLE CLIENT\_NUMERE\_TELEFON  
(  
 ID\_NUMAR INT PRIMARY KEY,  
 ID\_CLIENT\_NR\_TELEFON INT,  
 NR\_TELEFON VARCHAR(30),  
 CONSTRAINT CHECK\_NR\_TELEFON\_CLIENTI CHECK  
 (  
 *REGEXP\_LIKE*(NR\_TELEFON, '^07[0-9]{2} [0-9]{3} [0-9]{3}$')  
 ),  
 FOREIGN KEY (ID\_CLIENT\_NR\_TELEFON) REFERENCES CLIENT(ID\_CLIENT)  
);  
  
-- INSERARE INREGISTRARI IN TABEL CLIENT\_NUMERE\_TELEFON  
  
INSERT INTO CLIENT\_NUMERE\_TELEFON(ID\_NUMAR, ID\_CLIENT\_NR\_TELEFON, NR\_TELEFON)  
VALUES (CLIENT\_NUMERE\_TELEFON\_SEQUENCE.NEXTVAL, 1, '0700 000 004');  
  
INSERT INTO CLIENT\_NUMERE\_TELEFON(ID\_NUMAR, ID\_CLIENT\_NR\_TELEFON, NR\_TELEFON)  
VALUES (CLIENT\_NUMERE\_TELEFON\_SEQUENCE.NEXTVAL, 1, '0700 000 005');  
  
INSERT INTO CLIENT\_NUMERE\_TELEFON(ID\_NUMAR, ID\_CLIENT\_NR\_TELEFON, NR\_TELEFON)  
VALUES (CLIENT\_NUMERE\_TELEFON\_SEQUENCE.NEXTVAL, 2, '0700 000 006');  
  
INSERT INTO CLIENT\_NUMERE\_TELEFON(ID\_NUMAR, ID\_CLIENT\_NR\_TELEFON, NR\_TELEFON)  
VALUES (CLIENT\_NUMERE\_TELEFON\_SEQUENCE.NEXTVAL, 3, '0700 000 007');  
  
INSERT INTO CLIENT\_NUMERE\_TELEFON(ID\_NUMAR, ID\_CLIENT\_NR\_TELEFON, NR\_TELEFON)  
VALUES (CLIENT\_NUMERE\_TELEFON\_SEQUENCE.NEXTVAL, 4, '0700 000 008');  
  
INSERT INTO CLIENT\_NUMERE\_TELEFON(ID\_NUMAR, ID\_CLIENT\_NR\_TELEFON, NR\_TELEFON)  
VALUES (CLIENT\_NUMERE\_TELEFON\_SEQUENCE.NEXTVAL, 5, '0700 000 009');  
  
INSERT INTO CLIENT\_NUMERE\_TELEFON(ID\_NUMAR, ID\_CLIENT\_NR\_TELEFON, NR\_TELEFON)  
VALUES (CLIENT\_NUMERE\_TELEFON\_SEQUENCE.NEXTVAL, 6, '0700 000 010');





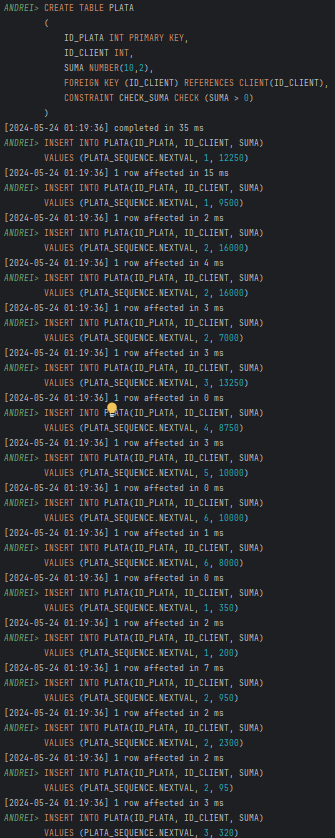
* Tabel VEHICUL\_DETINUT\_CLIENT

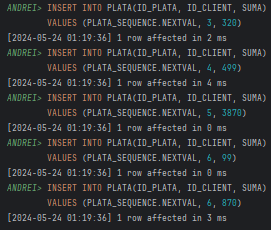
-- CREARE TABEL VEHICUL\_DETINUT\_CLIENT  
CREATE TABLE VEHICUL\_DETINUT\_CLIENT  
(  
 ID\_VEHICUL INT PRIMARY KEY,  
 ID\_CLIENT INT,  
 ID\_CAROSERIE INT,  
 ID\_MOTOR INT,  
 MARCA VARCHAR(30) DEFAULT 'Necunoscut',  
 MODEL VARCHAR(30) DEFAULT 'Necunoscut',  
 AN\_FABRICATIE INT,  
 NUMAR\_INMATRICULARE VARCHAR(30) UNIQUE,  
 FOREIGN KEY (ID\_CLIENT) REFERENCES CLIENT(ID\_CLIENT),  
 FOREIGN KEY (ID\_CAROSERIE) REFERENCES CAROSERIE(ID\_CAROSERIE),  
 FOREIGN KEY (ID\_MOTOR) REFERENCES MOTOR(ID\_MOTOR),  
 CONSTRAINT CHECK\_AN\_FABRICATIE\_VEHICUL\_DETINUT\_CLIENT CHECK (AN\_FABRICATIE > 0),  
 CONSTRAINT CHECK\_NUMAR\_INAMTRICULARE CHECK  
 (  
 *REGEXP\_LIKE*(NUMAR\_INMATRICULARE, '^(AB|AR|AG|B|BC|BH|BN|BR|BZ|CS|CL|CJ|CT|CV|DB|DJ|GL|GR|GJ|HR|HD|IL|IS|IF|MM|MH|MS|NT|OT|PH|SM|SJ|SB|SV|TR|TM|TL|VS|VL|VN|CD|CO|TC) [0-9]{2,3} [A-Z]{3}$')  
 )  
);  
  
-- INSERARE INREGISTRARI IN TABEL VEHICUL\_DETINUT\_CLIENT  
  
INSERT INTO VEHICUL\_DETINUT\_CLIENT(ID\_VEHICUL, ID\_CLIENT, ID\_CAROSERIE, ID\_MOTOR, MARCA, MODEL, AN\_FABRICATIE, NUMAR\_INMATRICULARE)  
VALUES (VEHICUL\_DETINUT\_CLIENT\_SEQUENCE.NEXTVAL, 1, 1, 1000, 'Dacia', 'Logan', 2020, 'B 123 ABC');  
  
INSERT INTO VEHICUL\_DETINUT\_CLIENT(ID\_VEHICUL, ID\_CLIENT, ID\_CAROSERIE, ID\_MOTOR, MARCA, MODEL, AN\_FABRICATIE, NUMAR\_INMATRICULARE)  
VALUES (VEHICUL\_DETINUT\_CLIENT\_SEQUENCE.NEXTVAL, 1, 2, 1001, 'Dacia', 'Sandero', 2023, 'B 124 ABC');  
  
INSERT INTO VEHICUL\_DETINUT\_CLIENT(ID\_VEHICUL, ID\_CLIENT, ID\_CAROSERIE, ID\_MOTOR, MARCA, MODEL, AN\_FABRICATIE, NUMAR\_INMATRICULARE)  
VALUES (VEHICUL\_DETINUT\_CLIENT\_SEQUENCE.NEXTVAL, 2, 3, 1001, 'Dacia', 'Duster', 2020, 'B 125 ABC');  
  
INSERT INTO VEHICUL\_DETINUT\_CLIENT(ID\_VEHICUL, ID\_CLIENT, ID\_CAROSERIE, ID\_MOTOR, MARCA, MODEL, AN\_FABRICATIE, NUMAR\_INMATRICULARE)  
VALUES (VEHICUL\_DETINUT\_CLIENT\_SEQUENCE.NEXTVAL, 2, 3, 1003, 'Dacia', 'Duster', 2021, 'B 126 ABC');  
  
INSERT INTO VEHICUL\_DETINUT\_CLIENT(ID\_VEHICUL, ID\_CLIENT, ID\_CAROSERIE, ID\_MOTOR, MARCA, MODEL, AN\_FABRICATIE, NUMAR\_INMATRICULARE)  
VALUES (VEHICUL\_DETINUT\_CLIENT\_SEQUENCE.NEXTVAL, 2, 5, 1003, 'Dacia', 'Pick-up', 2021, 'B 127 ABC');  
  
INSERT INTO VEHICUL\_DETINUT\_CLIENT(ID\_VEHICUL, ID\_CLIENT, ID\_CAROSERIE, ID\_MOTOR, MARCA, MODEL, AN\_FABRICATIE, NUMAR\_INMATRICULARE)  
VALUES (VEHICUL\_DETINUT\_CLIENT\_SEQUENCE.NEXTVAL, 3, 1, 1000, 'Dacia', 'Logan', 2020, 'B 128 ABC');  
  
INSERT INTO VEHICUL\_DETINUT\_CLIENT(ID\_VEHICUL, ID\_CLIENT, ID\_CAROSERIE, ID\_MOTOR, MARCA, MODEL, AN\_FABRICATIE, NUMAR\_INMATRICULARE)  
VALUES (VEHICUL\_DETINUT\_CLIENT\_SEQUENCE.NEXTVAL, 4, 2, 1004, 'Dacia', 'Spring', 2024, 'B 129 ABC');  
  
  
INSERT INTO VEHICUL\_DETINUT\_CLIENT(ID\_VEHICUL, ID\_CLIENT, ID\_CAROSERIE, ID\_MOTOR, MARCA, MODEL, AN\_FABRICATIE, NUMAR\_INMATRICULARE)  
VALUES (VEHICUL\_DETINUT\_CLIENT\_SEQUENCE.NEXTVAL, 5, 2, 1000, 'Dacia', 'Sandero', 2020, 'PH 123 ABC');  
  
INSERT INTO VEHICUL\_DETINUT\_CLIENT(ID\_VEHICUL, ID\_CLIENT, ID\_CAROSERIE, ID\_MOTOR, MARCA, MODEL, AN\_FABRICATIE, NUMAR\_INMATRICULARE)  
VALUES (VEHICUL\_DETINUT\_CLIENT\_SEQUENCE.NEXTVAL, 6, 2, 1001, 'Dacia', 'Sandero', 2023, 'PH 124 ABC');  
  
INSERT INTO VEHICUL\_DETINUT\_CLIENT(ID\_VEHICUL, ID\_CLIENT, ID\_CAROSERIE, ID\_MOTOR, MARCA, MODEL, AN\_FABRICATIE, NUMAR\_INMATRICULARE)  
VALUES (VEHICUL\_DETINUT\_CLIENT\_SEQUENCE.NEXTVAL, 6, 5, 1002, 'Dacia', 'Pick-up', 2020, 'PH 125 ABC');

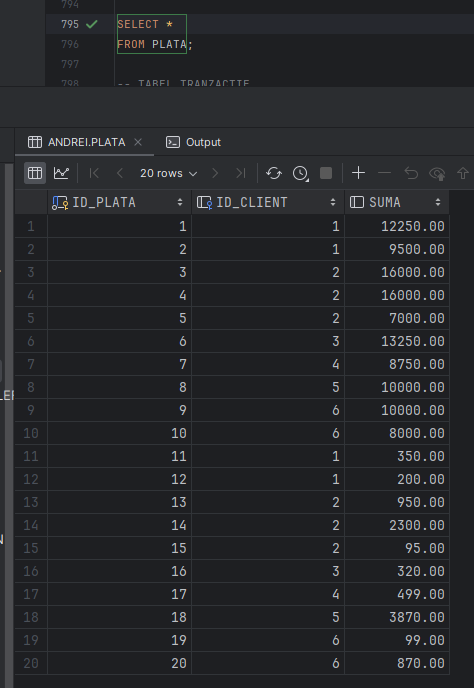
  
  


## Tabel PLATA

-- CREARE TABEL PLATA  
CREATE TABLE PLATA  
(  
 ID\_PLATA INT PRIMARY KEY,  
 ID\_CLIENT INT,  
 SUMA NUMBER(10,2),  
 FOREIGN KEY (ID\_CLIENT) REFERENCES CLIENT(ID\_CLIENT),  
 CONSTRAINT CHECK\_SUMA CHECK (SUMA > 0)  
);  
  
-- INSERARE INREGISTRARI IN TABEL PLATA  
  
-- PLATI ACHIZITIONAR VEHICUL  
INSERT INTO PLATA(ID\_PLATA, ID\_CLIENT, SUMA)  
VALUES (PLATA\_SEQUENCE.NEXTVAL, 1, 12250);  
  
INSERT INTO PLATA(ID\_PLATA, ID\_CLIENT, SUMA)  
VALUES (PLATA\_SEQUENCE.NEXTVAL, 1, 9500);  
  
INSERT INTO PLATA(ID\_PLATA, ID\_CLIENT, SUMA)  
VALUES (PLATA\_SEQUENCE.NEXTVAL, 2, 16000);  
  
INSERT INTO PLATA(ID\_PLATA, ID\_CLIENT, SUMA)  
VALUES (PLATA\_SEQUENCE.NEXTVAL, 2, 16000);  
  
INSERT INTO PLATA(ID\_PLATA, ID\_CLIENT, SUMA)  
VALUES (PLATA\_SEQUENCE.NEXTVAL, 2, 7000);  
  
INSERT INTO PLATA(ID\_PLATA, ID\_CLIENT, SUMA)  
VALUES (PLATA\_SEQUENCE.NEXTVAL, 3, 13250);  
  
INSERT INTO PLATA(ID\_PLATA, ID\_CLIENT, SUMA)  
VALUES (PLATA\_SEQUENCE.NEXTVAL, 4, 8750);  
  
INSERT INTO PLATA(ID\_PLATA, ID\_CLIENT, SUMA)  
VALUES (PLATA\_SEQUENCE.NEXTVAL, 5, 10000);  
  
INSERT INTO PLATA(ID\_PLATA, ID\_CLIENT, SUMA)  
VALUES (PLATA\_SEQUENCE.NEXTVAL, 6, 10000);  
  
INSERT INTO PLATA(ID\_PLATA, ID\_CLIENT, SUMA)  
VALUES (PLATA\_SEQUENCE.NEXTVAL, 6, 8000);  
  
-- PLATI SERVICE  
INSERT INTO PLATA(ID\_PLATA, ID\_CLIENT, SUMA)  
VALUES (PLATA\_SEQUENCE.NEXTVAL, 1, 350);  
  
INSERT INTO PLATA(ID\_PLATA, ID\_CLIENT, SUMA)  
VALUES (PLATA\_SEQUENCE.NEXTVAL, 1, 200);  
  
INSERT INTO PLATA(ID\_PLATA, ID\_CLIENT, SUMA)  
VALUES (PLATA\_SEQUENCE.NEXTVAL, 2, 950);  
  
INSERT INTO PLATA(ID\_PLATA, ID\_CLIENT, SUMA)  
VALUES (PLATA\_SEQUENCE.NEXTVAL, 2, 2300);  
  
INSERT INTO PLATA(ID\_PLATA, ID\_CLIENT, SUMA)  
VALUES (PLATA\_SEQUENCE.NEXTVAL, 2, 95);  
  
INSERT INTO PLATA(ID\_PLATA, ID\_CLIENT, SUMA)  
VALUES (PLATA\_SEQUENCE.NEXTVAL, 3, 320);  
  
INSERT INTO PLATA(ID\_PLATA, ID\_CLIENT, SUMA)  
VALUES (PLATA\_SEQUENCE.NEXTVAL, 4, 499);  
  
INSERT INTO PLATA(ID\_PLATA, ID\_CLIENT, SUMA)  
VALUES (PLATA\_SEQUENCE.NEXTVAL, 5, 3870);  
  
INSERT INTO PLATA(ID\_PLATA, ID\_CLIENT, SUMA)  
VALUES (PLATA\_SEQUENCE.NEXTVAL, 6, 99);  
  
INSERT INTO PLATA(ID\_PLATA, ID\_CLIENT, SUMA)  
VALUES (PLATA\_SEQUENCE.NEXTVAL, 6, 870);

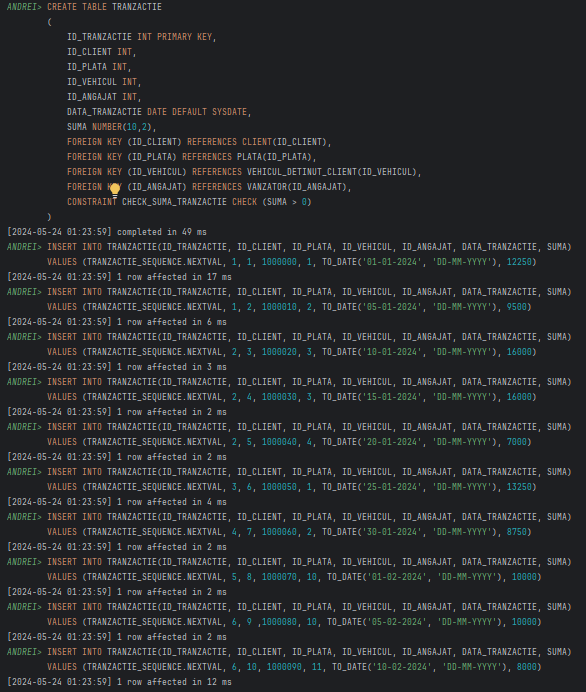


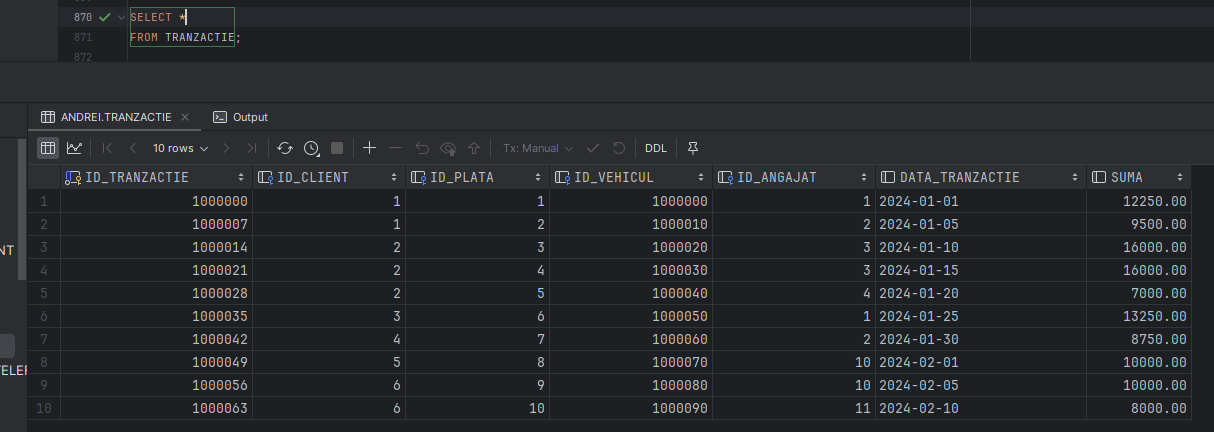




## Tabel TRANZACTIE

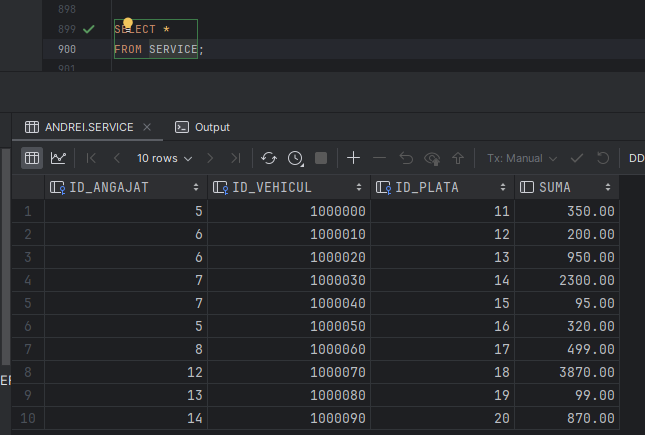
-- CREARE TABEL TRANZACTIE  
CREATE TABLE TRANZACTIE  
(  
 ID\_TRANZACTIE INT PRIMARY KEY,  
 ID\_CLIENT INT,  
 ID\_PLATA INT,  
 ID\_VEHICUL INT,  
 ID\_ANGAJAT INT,  
 DATA\_TRANZACTIE DATE DEFAULT *SYSDATE*,  
 SUMA NUMBER(10,2),  
 FOREIGN KEY (ID\_CLIENT) REFERENCES CLIENT(ID\_CLIENT),  
 FOREIGN KEY (ID\_PLATA) REFERENCES PLATA(ID\_PLATA),  
 FOREIGN KEY (ID\_VEHICUL) REFERENCES VEHICUL\_DETINUT\_CLIENT(ID\_VEHICUL),  
 FOREIGN KEY (ID\_ANGAJAT) REFERENCES VANZATOR(ID\_ANGAJAT),  
 CONSTRAINT CHECK\_SUMA\_TRANZACTIE CHECK (SUMA > 0)  
);  
  
-- INSERARE INREGISTRARI IN TABEL TRANZACTIE  
  
INSERT INTO TRANZACTIE(ID\_TRANZACTIE, ID\_CLIENT, ID\_PLATA, ID\_VEHICUL, ID\_ANGAJAT, DATA\_TRANZACTIE, SUMA)  
VALUES (TRANZACTIE\_SEQUENCE.NEXTVAL, 1, 1, 1000000, 1, *TO\_DATE*('01-01-2024', 'DD-MM-YYYY'), 12250);  
  
INSERT INTO TRANZACTIE(ID\_TRANZACTIE, ID\_CLIENT, ID\_PLATA, ID\_VEHICUL, ID\_ANGAJAT, DATA\_TRANZACTIE, SUMA)  
VALUES (TRANZACTIE\_SEQUENCE.NEXTVAL, 1, 2, 1000010, 2, *TO\_DATE*('05-01-2024', 'DD-MM-YYYY'), 9500);  
  
INSERT INTO TRANZACTIE(ID\_TRANZACTIE, ID\_CLIENT, ID\_PLATA, ID\_VEHICUL, ID\_ANGAJAT, DATA\_TRANZACTIE, SUMA)  
VALUES (TRANZACTIE\_SEQUENCE.NEXTVAL, 2, 3, 1000020, 3, *TO\_DATE*('10-01-2024', 'DD-MM-YYYY'), 16000);  
  
INSERT INTO TRANZACTIE(ID\_TRANZACTIE, ID\_CLIENT, ID\_PLATA, ID\_VEHICUL, ID\_ANGAJAT, DATA\_TRANZACTIE, SUMA)  
VALUES (TRANZACTIE\_SEQUENCE.NEXTVAL, 2, 4, 1000030, 3, *TO\_DATE*('15-01-2024', 'DD-MM-YYYY'), 16000);  
  
INSERT INTO TRANZACTIE(ID\_TRANZACTIE, ID\_CLIENT, ID\_PLATA, ID\_VEHICUL, ID\_ANGAJAT, DATA\_TRANZACTIE, SUMA)  
VALUES (TRANZACTIE\_SEQUENCE.NEXTVAL, 2, 5, 1000040, 4, *TO\_DATE*('20-01-2024', 'DD-MM-YYYY'), 7000);  
  
INSERT INTO TRANZACTIE(ID\_TRANZACTIE, ID\_CLIENT, ID\_PLATA, ID\_VEHICUL, ID\_ANGAJAT, DATA\_TRANZACTIE, SUMA)  
VALUES (TRANZACTIE\_SEQUENCE.NEXTVAL, 3, 6, 1000050, 1, *TO\_DATE*('25-01-2024', 'DD-MM-YYYY'), 13250);  
  
INSERT INTO TRANZACTIE(ID\_TRANZACTIE, ID\_CLIENT, ID\_PLATA, ID\_VEHICUL, ID\_ANGAJAT, DATA\_TRANZACTIE, SUMA)  
VALUES (TRANZACTIE\_SEQUENCE.NEXTVAL, 4, 7, 1000060, 2, *TO\_DATE*('30-01-2024', 'DD-MM-YYYY'), 8750);  
  
  
INSERT INTO TRANZACTIE(ID\_TRANZACTIE, ID\_CLIENT, ID\_PLATA, ID\_VEHICUL, ID\_ANGAJAT, DATA\_TRANZACTIE, SUMA)  
VALUES (TRANZACTIE\_SEQUENCE.NEXTVAL, 5, 8, 1000070, 10, *TO\_DATE*('01-02-2024', 'DD-MM-YYYY'), 10000);  
  
INSERT INTO TRANZACTIE(ID\_TRANZACTIE, ID\_CLIENT, ID\_PLATA, ID\_VEHICUL, ID\_ANGAJAT, DATA\_TRANZACTIE, SUMA)  
VALUES (TRANZACTIE\_SEQUENCE.NEXTVAL, 6, 9 ,1000080, 10, *TO\_DATE*('05-02-2024', 'DD-MM-YYYY'), 10000);  
  
INSERT INTO TRANZACTIE(ID\_TRANZACTIE, ID\_CLIENT, ID\_PLATA, ID\_VEHICUL, ID\_ANGAJAT, DATA\_TRANZACTIE, SUMA)  
VALUES (TRANZACTIE\_SEQUENCE.NEXTVAL, 6, 10, 1000090, 11, *TO\_DATE*('10-02-2024', 'DD-MM-YYYY'), 8000);





## Tabel SERVICE

SELECT \*  
FROM TRANZACTIE;  
  
-- TABEL SERVICE  
  
-- CREARE TABEL SERVICE  
CREATE TABLE SERVICE  
(  
 ID\_ANGAJAT INT,  
 ID\_VEHICUL INT,  
 ID\_PLATA INT,  
 SUMA NUMBER(10,2),  
 FOREIGN KEY (ID\_ANGAJAT) REFERENCES MECANIC(ID\_ANGAJAT),  
 FOREIGN KEY (ID\_VEHICUL) REFERENCES VEHICUL\_DETINUT\_CLIENT(ID\_VEHICUL),  
 FOREIGN KEY (ID\_PLATA) REFERENCES PLATA(ID\_PLATA)  
);  
  
-- INSERARE INREGISTRARI IN TABEL SERVICE  
  
INSERT INTO SERVICE(ID\_ANGAJAT, ID\_VEHICUL, ID\_PLATA, SUMA)  
VALUES (5, 1000000, 11, 350);  
  
INSERT INTO SERVICE(ID\_ANGAJAT, ID\_VEHICUL, ID\_PLATA, SUMA)  
VALUES (6, 1000010, 12, 200);  
  
INSERT INTO SERVICE(ID\_ANGAJAT, ID\_VEHICUL, ID\_PLATA, SUMA)  
VALUES (6, 1000020, 13, 950);  
  
INSERT INTO SERVICE(ID\_ANGAJAT, ID\_VEHICUL, ID\_PLATA, SUMA)  
VALUES (7, 1000030, 14, 2300);  
  
INSERT INTO SERVICE(ID\_ANGAJAT, ID\_VEHICUL, ID\_PLATA, SUMA)  
VALUES (7, 1000040, 15, 95);  
  
INSERT INTO SERVICE(ID\_ANGAJAT, ID\_VEHICUL, ID\_PLATA, SUMA)  
VALUES (5, 1000050, 16, 320);  
  
INSERT INTO SERVICE(ID\_ANGAJAT, ID\_VEHICUL, ID\_PLATA, SUMA)  
VALUES (8, 1000060, 17, 499);  
  
INSERT INTO SERVICE(ID\_ANGAJAT, ID\_VEHICUL, ID\_PLATA, SUMA)  
VALUES (12, 1000070, 18, 3870);  
  
INSERT INTO SERVICE(ID\_ANGAJAT, ID\_VEHICUL, ID\_PLATA, SUMA)  
VALUES (13, 1000080, 19, 99);  
  
INSERT INTO SERVICE(ID\_ANGAJAT, ID\_VEHICUL, ID\_PLATA, SUMA)  
VALUES (14, 1000090, 20, 870);

# **12. Formulați în limbaj natural și implementați 5 cereri SQL complexe**

Cererile vor utiliza următoarele elemente:

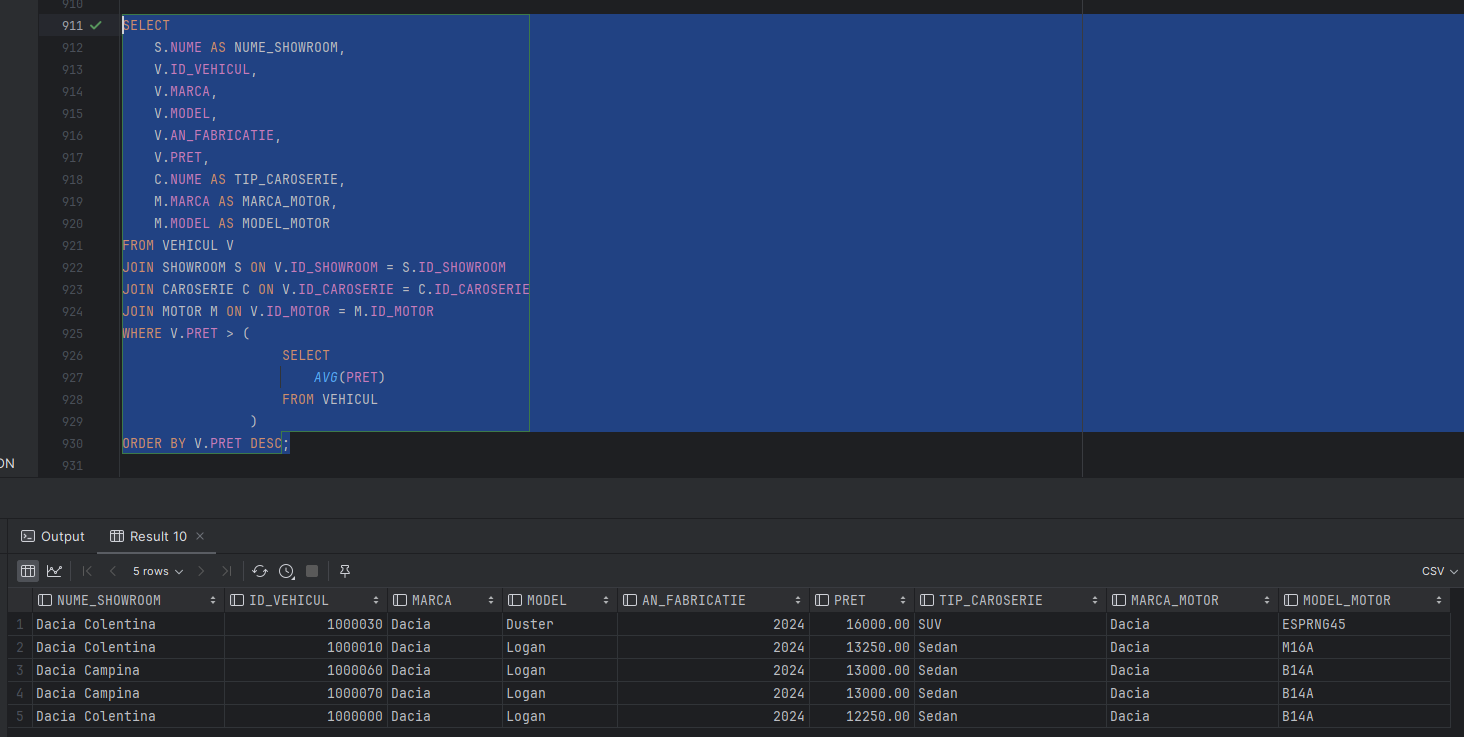
1. subcereri sincronizate în care intervin cel puțin 3 tabele
2. subcereri nesincronizate în clauza FROM
3. grupări de date, funcții grup, filtrare la nivel de grupuri cu subcereri nesincronizate (în clauza de HAVING) în care intervin cel puțin 3 tabele (in cadrul aceleiași cereri)
4. ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri)
5. utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE
6. utilizarea a cel puțin 1 bloc de cerere (clauza WITH)

**Observație**: Într-o cerere se vor regăsi mai multe elemente dintre cele enumerate mai sus, astfel încât cele 5 cereri să le cuprindă pe toate.

## Cererea 1

* AFISEAZA VEHICULELE DIN SHOWROOM-URI CARE AU PRETUL PESTE MEDIE (DE AFISAT: NUME\_SHOWROOM, ID\_VEHICUL, MARCA, MODEL, AN\_FABRICATIE, PRET, TIP\_CAROSERIE, MARCA\_MOTOR, MODEL\_MOTOR).
* a) subcereri sincronizate in care intervin cel puțin 3 tabele

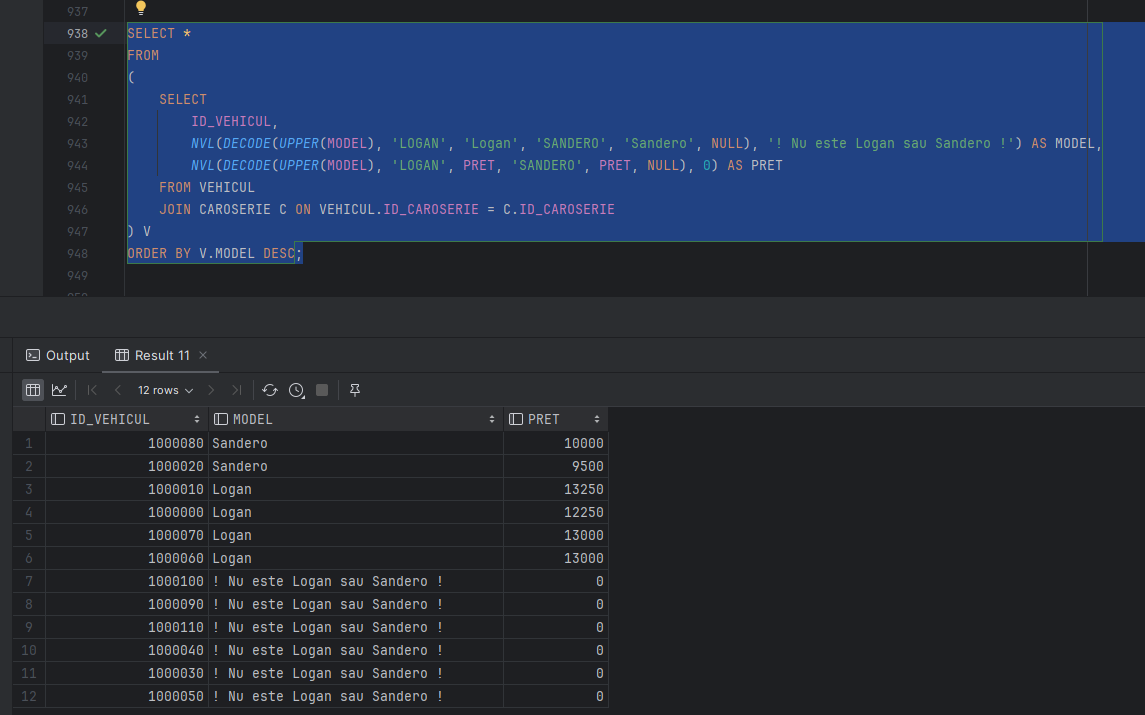
SELECT  
 S.NUME AS NUME\_SHOWROOM,  
 V.ID\_VEHICUL,  
 V.MARCA,  
 V.MODEL,  
 V.AN\_FABRICATIE,  
 V.PRET,  
 C.NUME AS TIP\_CAROSERIE,  
 M.MARCA AS MARCA\_MOTOR,  
 M.MODEL AS MODEL\_MOTOR  
FROM VEHICUL V  
JOIN SHOWROOM S ON V.ID\_SHOWROOM = S.ID\_SHOWROOM  
JOIN CAROSERIE C ON V.ID\_CAROSERIE = C.ID\_CAROSERIE  
JOIN MOTOR M ON V.ID\_MOTOR = M.ID\_MOTOR  
WHERE V.PRET > (  
 SELECT  
 *AVG*(PRET)  
 FROM VEHICUL  
 )  
ORDER BY V.PRET DESC;



## Cererea 2

* AFISATI ID\_UL, MODELUL SI PRETUL VEHICULELOR CARE AU MODELUL LOGAN SAU SANDERO, IAR IN CAZ CONTRAR AFISATI ID\_UL VEHICULULUI, MESAJUL 'Nu este Logan sau Sandero' IN CADRUL COLOANEI MODEL, SI PRETUL EGAL CU 0. ORDONATI LEXICOGRAFIC DESCRESCATOR DUPA MODEL.
* d) ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri)

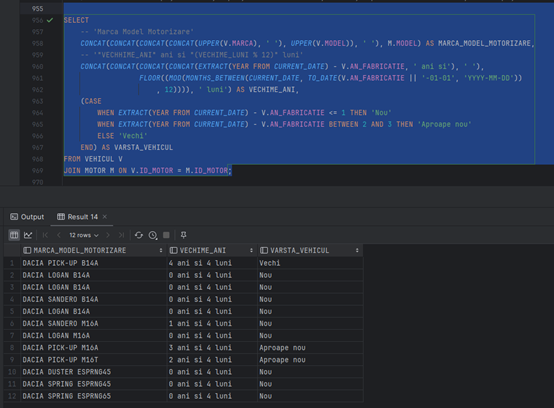
SELECT \*  
FROM  
(  
 SELECT  
 ID\_VEHICUL,  
 *NVL*(*DECODE*(*UPPER*(MODEL), 'LOGAN', 'Logan', 'SANDERO', 'Sandero', NULL)  
 , '! Nu este Logan sau Sandero !') AS MODEL,  
 *NVL*(*DECODE*(*UPPER*(MODEL), 'LOGAN', PRET, 'SANDERO', PRET, NULL), 0) AS PRET  
 FROM VEHICUL  
 JOIN CAROSERIE C ON VEHICUL.ID\_CAROSERIE = C.ID\_CAROSERIE  
) V  
ORDER BY V.MODEL DESC;



## Cererea 3

* AFISATI PE O COLOANA MARCA MODELUL SI MOTORIZAREA SEPARATE PRINTR-UN SPATIU ('Marca Model Motorizare'), VECHIMEA VEHICULULUI IN ANI SI LUNI FORMATATE ('"VECHHIME\_ANI" ani si "(VECHIME\_LUNI % 12)" luni'), SI VARSTA VEHICULULUI ('Nou' < 1 AN, 'Aproape nou' APARTINE [2,3] ANI, 'Vechi' > 3 ANI).
* e) utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE.

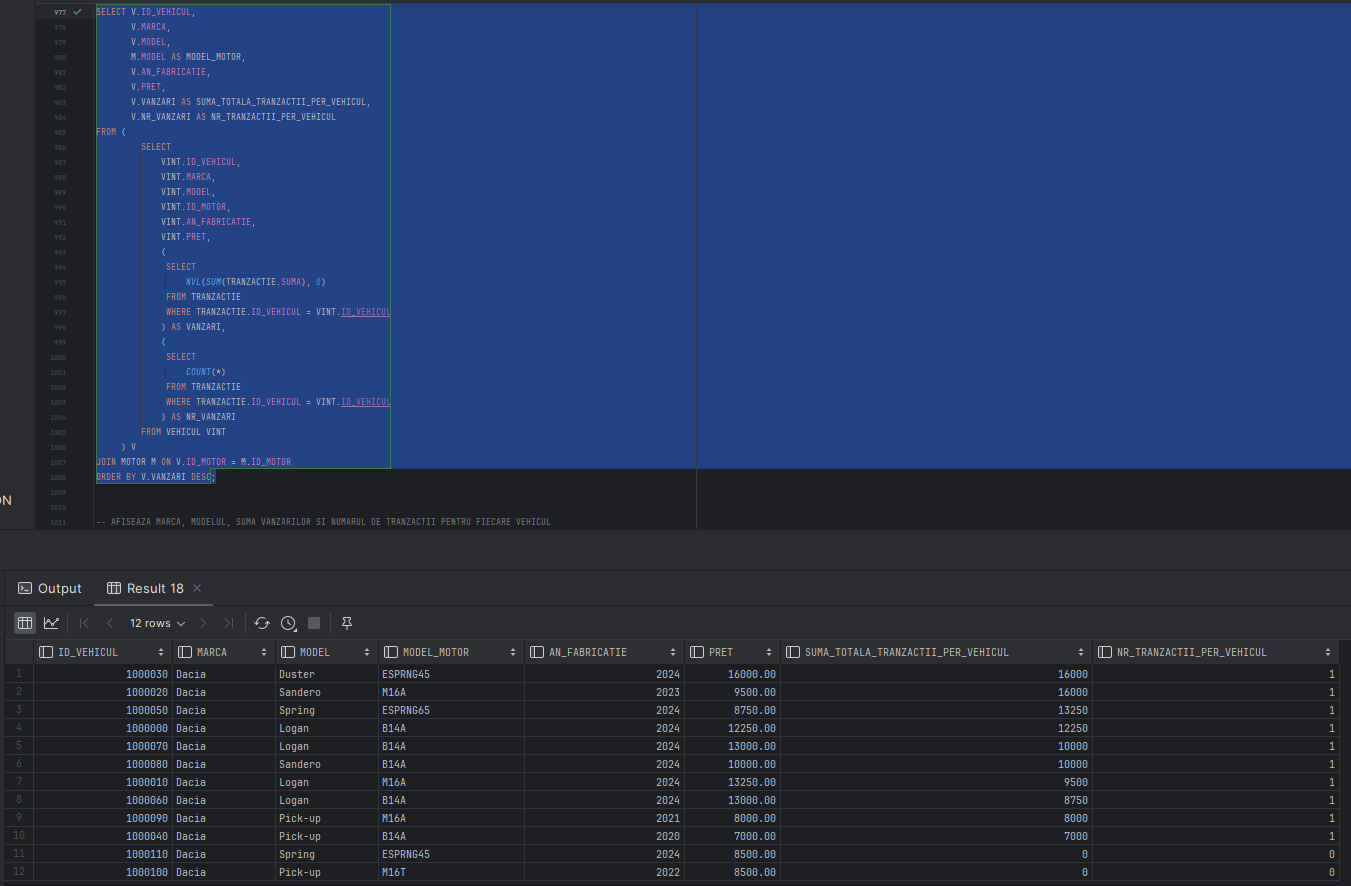
SELECT  
 -- 'Marca Model Motorizare'  
 *CONCAT*(*CONCAT*(*CONCAT*(*CONCAT*(*UPPER*(V.MARCA), ' '), *UPPER*(V.MODEL)), ' '), M.MODEL) AS MARCA\_MODEL\_MOTORIZARE,  
 -- '"VECHHIME\_ANI" ani si "(VECHIME\_LUNI % 12)" luni'  
 *CONCAT*(*CONCAT*(*CONCAT*(*CONCAT*(*EXTRACT*(YEAR FROM *CURRENT\_DATE*) - V.AN\_FABRICATIE, ' ani si'), ' '),   
 *FLOOR*((*MOD*(*MONTHS\_BETWEEN*(*CURRENT\_DATE*, *TO\_DATE*(V.AN\_FABRICATIE || '-01-01', 'YYYY-MM-DD'))  
 , 12)))), ' luni') AS VECHIME\_ANI,  
 (CASE  
 WHEN *EXTRACT*(YEAR FROM *CURRENT\_DATE*) - V.AN\_FABRICATIE <= 1 THEN 'Nou'  
 WHEN *EXTRACT*(YEAR FROM *CURRENT\_DATE*) - V.AN\_FABRICATIE BETWEEN 2 AND 3 THEN 'Aproape nou'  
 ELSE 'Vechi'  
 END) AS VARSTA\_VEHICUL  
FROM VEHICUL V  
JOIN MOTOR M ON V.ID\_MOTOR = M.ID\_MOTOR;



## Cererea 4

* SA SE AFISEZE ID\_VEHICUL, MARCA, MODEL, MODELUL MOTORULUI, ANUL DE FABRICATIE, PRETUL SI SUMA TOTALA A TRANZACTIILOR CE CONTIN VEHICULUL RESPECTIV, IMPREUNA CU NUMARUL DE TRANZACTII CE CONTIN VEHICULUL RESPECTIV (DE CATE ORI A FOST VANDUT ACEST MODEL)
* b) subcereri nesincronizate în clauza FROM

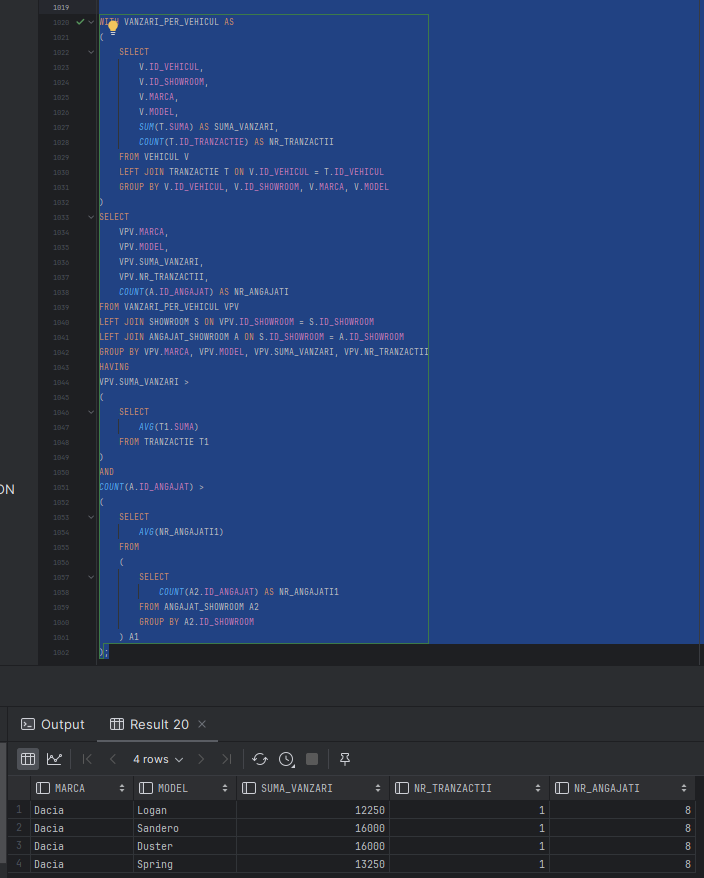
SELECT V.ID\_VEHICUL,  
 V.MARCA,  
 V.MODEL,  
 M.MODEL AS MODEL\_MOTOR,  
 V.AN\_FABRICATIE,  
 V.PRET,  
 V.VANZARI AS SUMA\_TOTALA\_TRANZACTII\_PER\_VEHICUL,  
 V.NR\_VANZARI AS NR\_TRANZACTII\_PER\_VEHICUL  
FROM (  
 SELECT  
 VINT.ID\_VEHICUL,  
 VINT.MARCA,  
 VINT.MODEL,  
 VINT.ID\_MOTOR,  
 VINT.AN\_FABRICATIE,  
 VINT.PRET,  
 (  
 SELECT  
 *NVL*(*SUM*(TRANZACTIE.SUMA), 0)  
 FROM TRANZACTIE  
 WHERE TRANZACTIE.ID\_VEHICUL = VINT.ID\_VEHICUL  
 ) AS VANZARI,  
 (  
 SELECT  
 *COUNT*(\*)  
 FROM TRANZACTIE  
 WHERE TRANZACTIE.ID\_VEHICUL = VINT.ID\_VEHICUL  
 ) AS NR\_VANZARI  
 FROM VEHICUL VINT  
 ) V  
JOIN MOTOR M ON V.ID\_MOTOR = M.ID\_MOTOR  
ORDER BY V.VANZARI DESC;



## Cererea 5

* AFISEAZA MARCA, MODELUL, SUMA VANZARILOR SI NUMARUL DE TRANZACTII PENTRU FIECARE VEHICUL CARE A GENERAT TRANZACTII CU SUMA MAI MARE DECAT MEDIA SUMELOR TUTUROR TRANZACTIILOR, SI CARE A FOST VANDUT DE CATRE UN SHOWROOM CU UN NUMAR DE ANGAJATI MAI MARE DECAT MEDIA NUMARULUI DE ANGAJATI DIN TOATE SHOWROOM-URILE.
* c) grupări de date, funcții grup, filtrare la nivel de grupuri cu subcereri nesincronizate (in clauza de HAVING) in care intervin cel puțin 3 tabele (in cadrul aceleiași cereri)
* utilizarea a cel puțin 1 bloc de cerere (clauza WITH)

WITH VANZARI\_PER\_VEHICUL AS  
(  
 SELECT  
 V.ID\_VEHICUL,  
 V.ID\_SHOWROOM,  
 V.MARCA,  
 V.MODEL,  
 *SUM*(T.SUMA) AS SUMA\_VANZARI,  
 *COUNT*(T.ID\_TRANZACTIE) AS NR\_TRANZACTII  
 FROM VEHICUL V  
 LEFT JOIN TRANZACTIE T ON V.ID\_VEHICUL = T.ID\_VEHICUL  
 GROUP BY V.ID\_VEHICUL, V.ID\_SHOWROOM, V.MARCA, V.MODEL  
)  
SELECT  
 VPV.MARCA,  
 VPV.MODEL,  
 VPV.SUMA\_VANZARI,  
 VPV.NR\_TRANZACTII,  
 *COUNT*(A.ID\_ANGAJAT) AS NR\_ANGAJATI  
FROM VANZARI\_PER\_VEHICUL VPV  
LEFT JOIN SHOWROOM S ON VPV.ID\_SHOWROOM = S.ID\_SHOWROOM  
LEFT JOIN ANGAJAT\_SHOWROOM A ON S.ID\_SHOWROOM = A.ID\_SHOWROOM  
GROUP BY VPV.MARCA, VPV.MODEL, VPV.SUMA\_VANZARI, VPV.NR\_TRANZACTII  
HAVING  
VPV.SUMA\_VANZARI >  
(  
 SELECT  
 *AVG*(T1.SUMA)  
 FROM TRANZACTIE T1  
)  
AND  
*COUNT*(A.ID\_ANGAJAT) >  
(  
 SELECT  
 *AVG*(NR\_ANGAJATI1)  
 FROM  
 (  
 SELECT  
 *COUNT*(A2.ID\_ANGAJAT) AS NR\_ANGAJATI1  
 FROM ANGAJAT\_SHOWROOM A2  
 GROUP BY A2.ID\_SHOWROOM  
 ) A1  
);



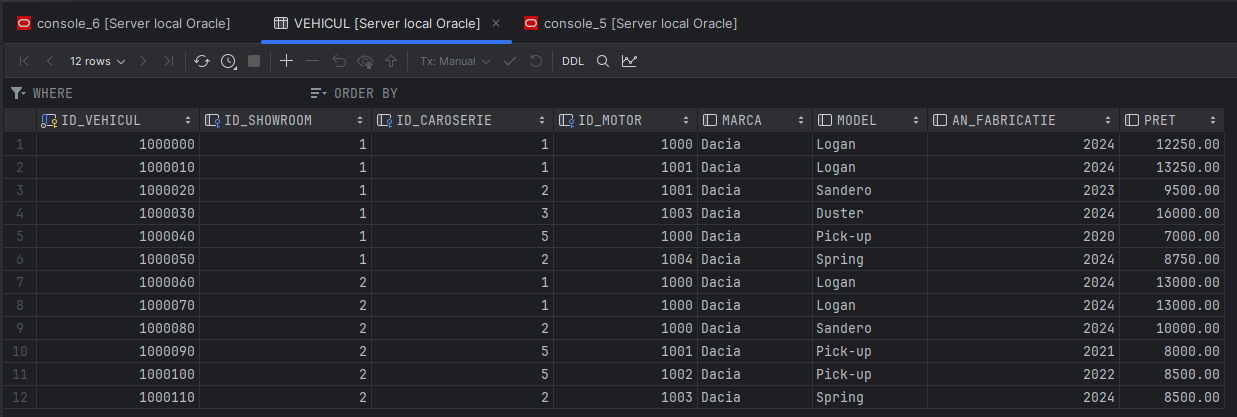
# **13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.**

## Operația 1

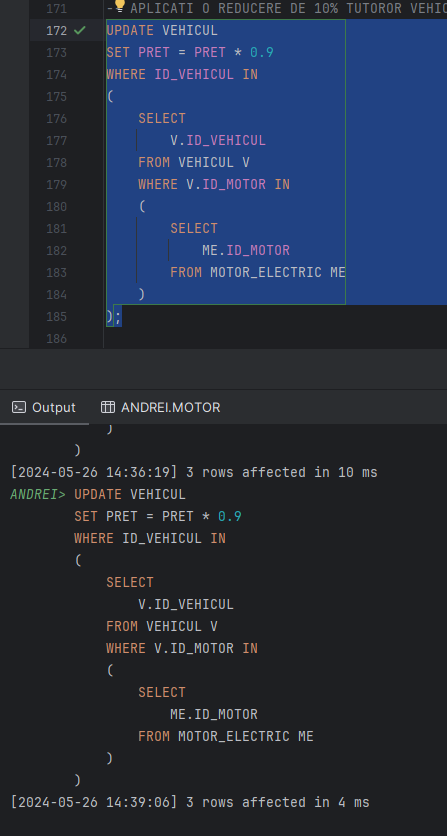
APLICATI O REDUCERE DE 10% TUTOROR VEHICULELOR CARE AU MOTOR ELECTRIC. (MODIFICARILE SE VOR FACE ASUPRA VEHICULELOR CU ID\_VEHICUL IN (1000030, 1000050, 1000110)).

UPDATE VEHICUL  
SET PRET = PRET \* 0.9  
WHERE ID\_VEHICUL IN  
(  
 SELECT  
 V.ID\_VEHICUL  
 FROM VEHICUL V  
 WHERE V.ID\_MOTOR IN  
 (  
 SELECT  
 ME.ID\_MOTOR  
 FROM MOTOR\_ELECTRIC ME  
 )  
);

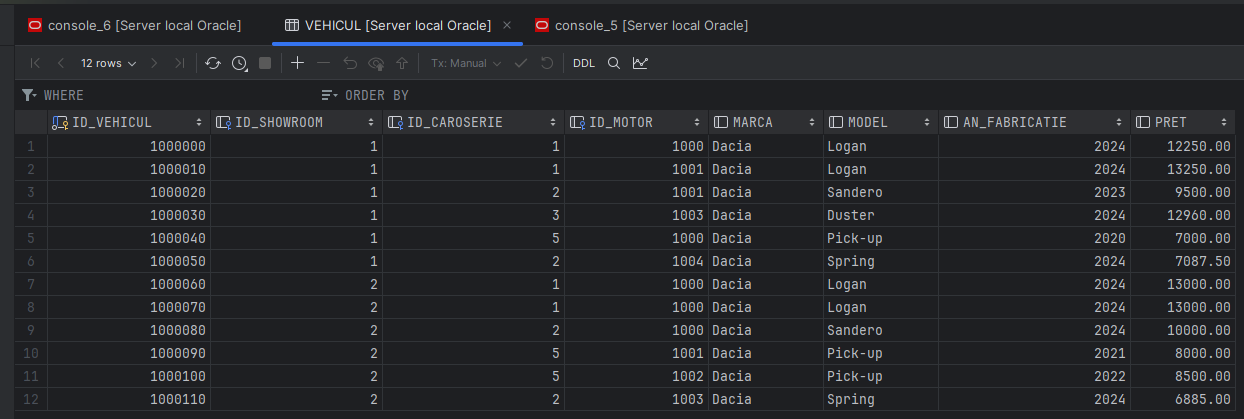
BEFORE:



RULARE:



AFTER:

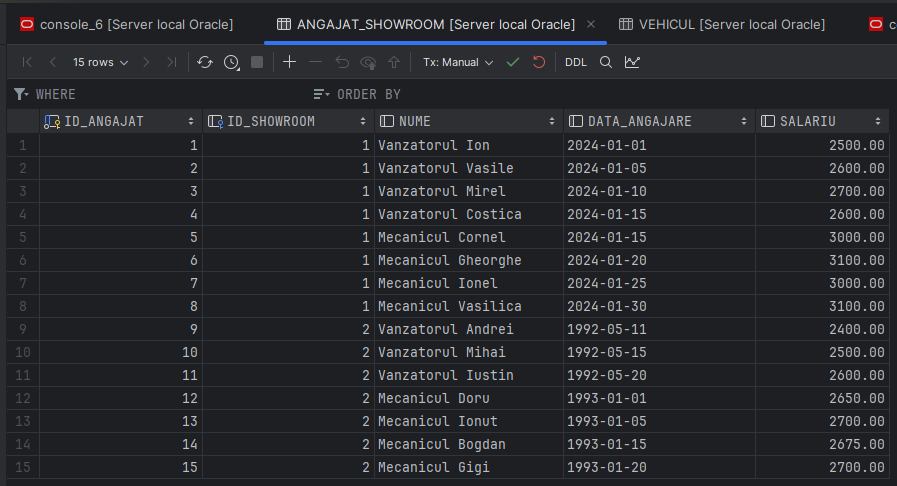


## Operația 2

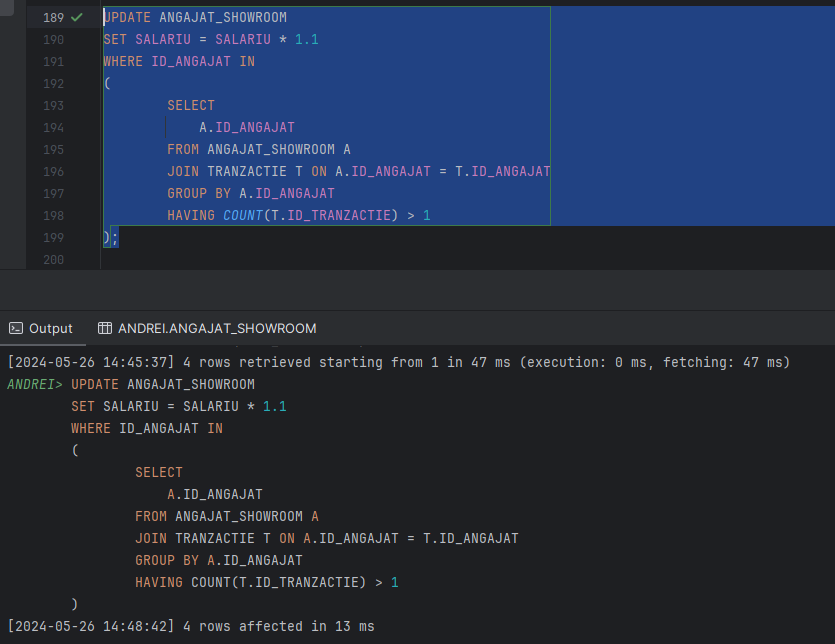
APLICATI O MARIRE DE SALARIU DE 10% ANGAJATILOR CARE AU EFECTUAT MAI MULTE DE O TRANZACTIE. (MODIFICARILE SE VOR FACE ASUPRA ANGAJATILOR CU ID\_ANGAJAT IN (1, 2, 3, 10)

UPDATE ANGAJAT\_SHOWROOM  
SET SALARIU = SALARIU \* 1.1  
WHERE ID\_ANGAJAT IN  
(  
 SELECT  
 A.ID\_ANGAJAT  
 FROM ANGAJAT\_SHOWROOM A  
 JOIN TRANZACTIE T ON A.ID\_ANGAJAT = T.ID\_ANGAJAT  
 GROUP BY A.ID\_ANGAJAT  
 HAVING *COUNT*(T.ID\_TRANZACTIE) > 1  
);

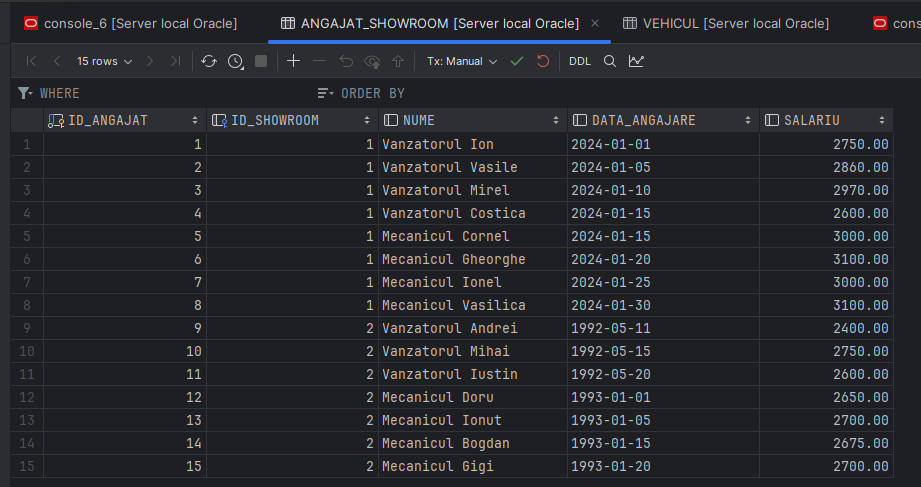
BEFORE:



RULARE:



AFTER:

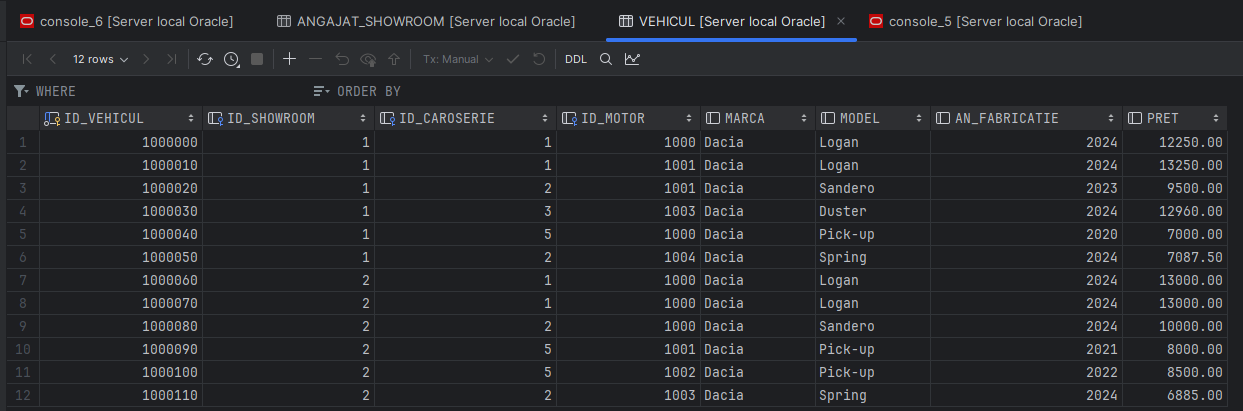


## Operația 3

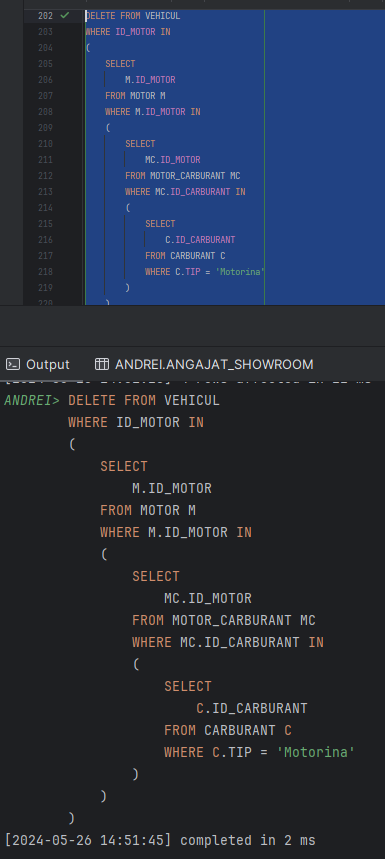
STERGETI TOATE VEHICULELE DIN SHOWROOM-URI CARE FOLOSESC MOTORINA.

DELETE FROM VEHICUL  
WHERE ID\_MOTOR IN  
(  
 SELECT  
 M.ID\_MOTOR  
 FROM MOTOR M  
 WHERE M.ID\_MOTOR IN  
 (  
 SELECT  
 MC.ID\_MOTOR  
 FROM MOTOR\_CARBURANT MC  
 WHERE MC.ID\_CARBURANT IN  
 (  
 SELECT  
 C.ID\_CARBURANT  
 FROM CARBURANT C  
 WHERE C.TIP = 'Motorina'  
 )  
 )  
);

BEFORE:



RULARE:



AFTER:

