**Proiect sisteme de gestiune a bazelor de date**

**Rețea națională de reprezentanțe auto**

Realizator: Anghel Răzvan-Alexandru

Universitatea din București

Facultatea de Matematică și Informatică

Seria 23, grupa 231

Profesor: dr. Mihai Gabriela

An universitar: 2024-2025

Cuprins

[Introducere 3](#_Toc187427769)

[1) Prezentarea bazei de date 4](#_Toc187427770)

[2) Diagrama entitate-relație. 5](#_Toc187427771)

[3) Diagrama conceptuală 6](#_Toc187427773)

[4) Implementarea în Oracle a diagramei conceptuale. 7](#_Toc187427775)

[5) Adăugarea de informații în tabelele create 14](#_Toc187427776)

[6) Exercițiul 6 - subprogram stocat independent care să utilizeze toate cele 3 tipuri de colecții studiate. 31](#_Toc187427777)

[7) Exercițiul 7 - un subprogram stocat independentcare să utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat, dependent de celălalt cursor.. 34](#_Toc187427778)

[8) Exercițiul 8 - un subprogram stocat independent de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele create 36](#_Toc187427779)

[9) Exercițiul 9 - un subprogram stocat independent de tip procedură care să aibă minim 2 parametri și să utilizeze într-o singură comandă SQL 5 dintre tabelele create 43](#_Toc187427780)

[10) Exercițiul 10 - Trigger de tip LMD la nivel de comandă 47](#_Toc187427781)

[11) Exercițiul 11 - Trigger de tip LMD la nivel de linie 49](#_Toc187427782)

[12) Exercițiul 12 - Trigger LDD 57](#_Toc187427783)

# Introducere

Am ales ca temă de proiect activitatea unei rețele naționale de reprezentanțe auto care doresc să își gestioneze activitatea sincron.

SGBD-ul folosit este Oracle Database 21c Express Edition Release 21.0.0.0.0 – Production.

Configurația hardware este: *sga\_target: 1610612736 bytes* ( aproximativ 1.5 GB ); *pga\_target: 536870912* ( aproximativ 512 MB ).

Configurația software este reprezentată de sistemul de operare Windows 11 Pro, versiunea 24H2, pe 64 de biți, rulat nativ, nu într-o mașină virtuală.

# Prezentați pe scurt baza de date (utilitatea ei).

Acest proiect a fost conceput pentru a descrie modalitatea de funcționare a unei rețele de reprezentanțe auto cu multiple sedii la nivel național. Reprezentanțele ( numite în continuare și dealeri pentru a evita repetițiile ) au ca activitate principală vânzarea de autovehicule noi și second hand către clienții lor. În oferta de automobile sunt cuprinse mașini fabricate de mai multe mărci, importate în România de către entități juridice numite importatori. Ele au rolul de a asigura legalitatea și conformitatea bunurilor vândute de dealeri. Reprezentanțele pun la dispoziția clienților și o rețea de service-uri, câte unul pentru fiecare dealer, unde clienții pot efectua revizii și reparații în cadrul garanției dar și după expirarea acesteia.

Fiecare reprezentanță oferă și servicii de buy-back, prin care un client poate renunța la mașina lui actuală și poate alege o mașină nouă pe care să o achiziționeze, din prețul acesteia fiind dedusă valoarea de piață a vehiculului adus de client. Dealer-ul le va vinde apoi cu un ados ca mașini second-hand certificate, ceea ce însemnă că o mașină poate fi vândută de mai multe ori de către același dealer, o dată ca mașină nouă, ulterior ca mașină second-hand. Drept urmare, entitatea *MAȘINĂ* va avea drept cheie primară un id generat automat și nu seria sa unică de șasiu.

Autovehiculele noi oferite de dealer pot fi de două tipuri: din stoc și la comandă. Cele aflate pe stoc pot fi cumpărate instant de client, în schimb, dacă acesta dorește o altă configurație pentru modelul ales ( spre exemplu preferând o altă culoare ), va trebui să opteze pentru un vehicul realizat la comandă. Prin intermediul dealer-ului, se configurează o mașină, apoi comanda se trimite la importatorul mărcii respective, care o transmite la cea mai apropiată fabrică, urmând a o livra către dealer în câteva luni. O reprezentață poate comanda mașini pentru a le avea pe stoc sau pentru uz propriu ( drive test pentru clienți, vehicule pentru angajați ).

Livrarea se realizează printr-o altă entitate juridică numită transportator. Acesta este contractat de importator și are scopul de a livra vehiculele de la locația unde sunt importate în țară ( de regulă port sau depou feroviar ) către dealer.

Utilitatea modelului descris constă în exemplificarea funționării unei rețele de reprezentanțe auto la nivel național, putând servi drept baza unei aplicații de gestiune a activității unor dealeri care doresc să își sincronizeze afacerile.

Un dealer poate comercializa mașini de la mai multe mărci, apelând la câte un importator diferit, în funcție de vehiculele comandate. Importatorii pot aduce în țară autovehicule doar de la un singur fabricant, adică o singură marcă.

Se consideră transportatori înregistrați în baza de date entitățile care au fost contracte de importatori. Similar, se consideră client, o entitate care a plasat cel puțin o comandă la una din reprezentanțe, pentru a evita redundanța.

Modelul nu include activitatea fabricilor, a procesului de fabricare, comunicarea importatorilor cu fabricile sau operațiunile interne din service-uri.

Am considerat că fiecare entitate contactabilă va avea un singur număr de telefon, un singur cont bancar și o singură adresă de e-mail.

Am considerat că importatorii au acces la comenzile inițiate de clienți și onorate de dealeri, astfel încât ei să poată comanda la rândul lor vehicule de la fabrici. De asemenea, am considerat că o comandă onorată de o reprezentanță ( adică o comandă înregistrată în baza de date ) este trimisă automat prin sistem la importatorii mărcilor din comandă. Astfel, o relație între importator și dealer nu este necesară.

Pentru simplificare, la adresele din municipiul București, am considerat județul ca fiind București, iar localitatea ca fiind sectorul unde este situată adresa.

# 2) Realizați diagrama entitate-relație (ERD): entitățile, relațiile și atributele trebuie definite în limba română.

# 

# 3) Pornind de la diagrama entitate-relație realizați diagrama conceptuală a modelului propus, integrând toate atributele necesare: entitățile, relațiile și atributele trebuie definite în limba română.

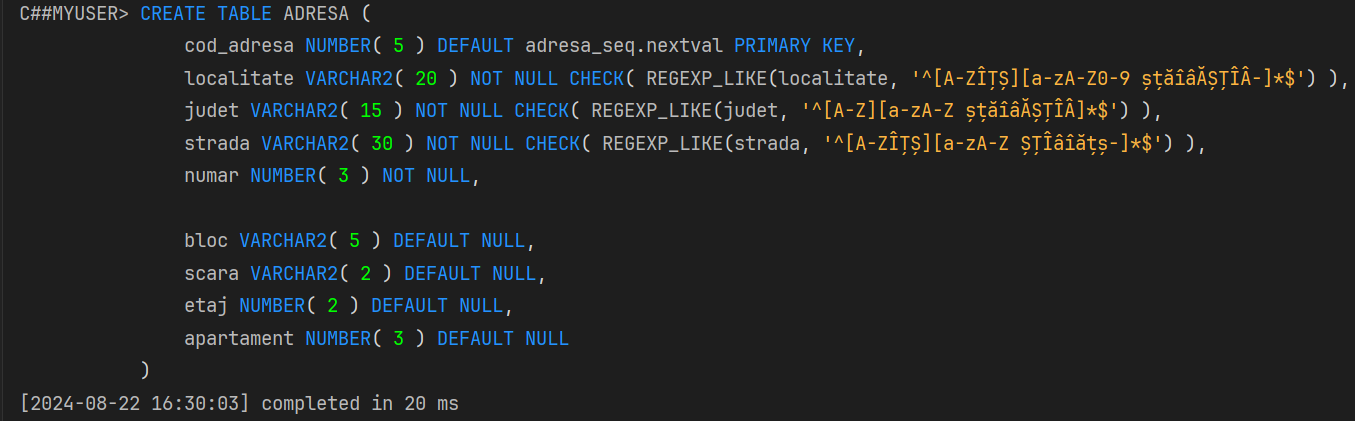
# 

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

Toate tabelele și secvențele folosite în acest proiect au fost create sub userul *C##MYUSER*, folosind comanda „*SYS> alter session set current\_schema = C##MYUSER*”.

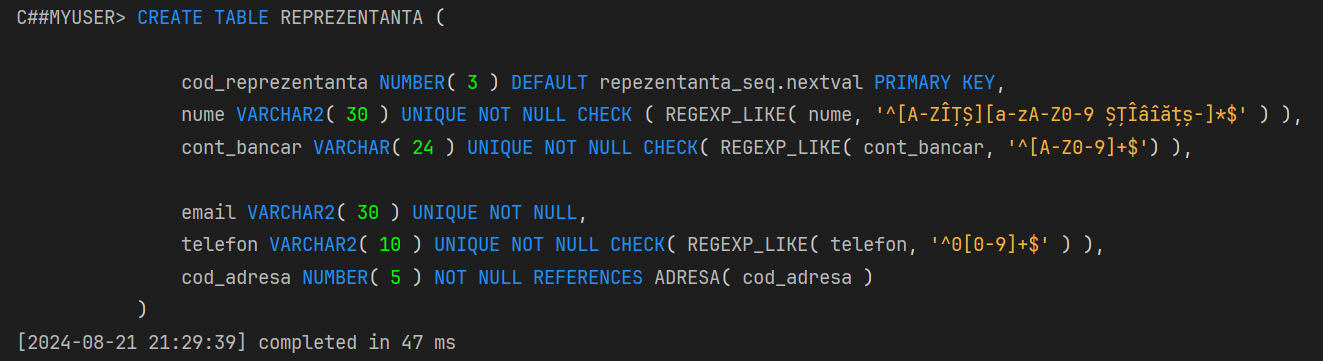
*-- SECVENȚE  
  
-- lungimea maximă a id-ului entității MASINĂ este 5*CREATE SEQUENCE masina\_seq  
START WITH 32456  
MAXVALUE 99999  
INCREMENT BY 1  
NOCYCLE  
NOCACHE;  
  
*-- lungimea maximă a id-ului entității COMANDĂ este 8*CREATE SEQUENCE comanda\_seq  
START WITH 1249455  
MAXVALUE 99999999  
INCREMENT BY 1  
NOCYCLE  
NOCACHE;  
  
*-- lungimea maximă a id-ului entității CLIENT este 5*CREATE SEQUENCE client\_seq  
START WITH 23215  
MAXVALUE 99999  
INCREMENT BY 1  
NOCYCLE  
NOCACHE;  
  
  
*-- lungimea maximă a id-ului entității ADRESĂ este 5*CREATE SEQUENCE adresa\_seq  
START WITH 12754  
MAXVALUE 99999  
INCREMENT BY 1  
NOCYCLE  
NOCACHE;  
  
*-- lungimea maximă a id-ului entității REPREZENTANȚĂ este 3*CREATE SEQUENCE repezentanta\_seq  
START WITH 485  
MAXVALUE 999  
INCREMENT BY 1  
NOCYCLE  
NOCACHE;  
  
*-- lungimea maximă a id-ului entității SERVICE este 3*CREATE SEQUENCE serivice\_seq  
START WITH 100  
MAXVALUE 999  
INCREMENT BY 1  
NOCYCLE  
NOCACHE;  
  
*-- lungimea maximă a id-ului entității IMPORTATOR este 4*CREATE SEQUENCE importator\_seq  
START WITH 5783  
MAXVALUE 9999  
INCREMENT BY 1  
NOCYCLE  
NOCACHE;  
  
*-- lungimea maximă a id-ului entității TRANSPORTATOR este 4*CREATE SEQUENCE transportator\_seq  
START WITH 4771  
MAXVALUE 9999  
INCREMENT BY 1  
NOCYCLE  
NOCACHE;  
  
*-- lungimea maximă a id-ului entității TRANSPORT este 4*CREATE SEQUENCE transport\_seq  
START WITH 3794  
MAXVALUE 9999  
INCREMENT BY 1  
NOCYCLE  
NOCACHE;

*-- TABELE*CREATE TABLE ADRESA (  
 cod\_adresa NUMBER( 5 ) DEFAULT adresa\_seq.nextval PRIMARY KEY,  
 localitate VARCHAR2( 20 ) NOT NULL CHECK( *REGEXP\_LIKE*(localitate, '^[A-ZÎȚȘ][a-zA-Z0-9 șțăîâĂȘȚÎÂ-]\*$') ),  
 judet VARCHAR2( 15 ) NOT NULL CHECK( *REGEXP\_LIKE*(judet, '^[A-Z][a-zA-Z șțăîâĂȘȚÎÂ]\*$') ),  
 strada VARCHAR2( 30 ) NOT NULL CHECK( *REGEXP\_LIKE*(strada, '^[A-ZÎȚȘ][a-zA-Z ȘȚÎâîățș-]\*$') ),  
 numar NUMBER( 3 ) NOT NULL,  
  
 bloc VARCHAR2( 5 ) DEFAULT NULL,  
 scara VARCHAR2( 2 ) DEFAULT NULL,  
 etaj NUMBER( 2 ) DEFAULT NULL,  
 apartament NUMBER( 3 ) DEFAULT NULL  
);



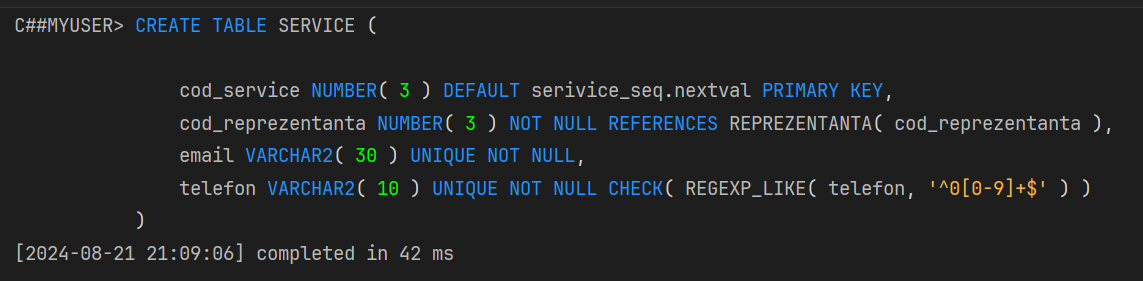
*Figura 4.1 – Rezultat creare tabel ADRESĂ*

CREATE TABLE REPREZENTANTA (  
  
 cod\_reprezentanta NUMBER( 3 ) DEFAULT repezentanta\_seq.nextval PRIMARY KEY,  
 nume VARCHAR2( 10 ) UNIQUE NOT NULL CHECK ( REGEXP\_LIKE( nume, '^[A-ZÎȚȘ][a-zA-Z0-9 ȘȚÎâîățș-]\*$' ) ),  
 cont\_bancar VARCHAR( 24 ) UNIQUE NOT NULL CHECK( REGEXP\_LIKE( cont\_bancar, '^[A-Z0-9]+$') ),  
  
 email VARCHAR2( 30 ) UNIQUE NOT NULL,  
 telefon VARCHAR2( 10 ) UNIQUE NOT NULL CHECK( REGEXP\_LIKE( telefon, '^0[0-9]+$' ) ),  
 cod\_adresa NUMBER( 5 ) NOT NULL REFERENCES ADRESA( cod\_adresa )  
);



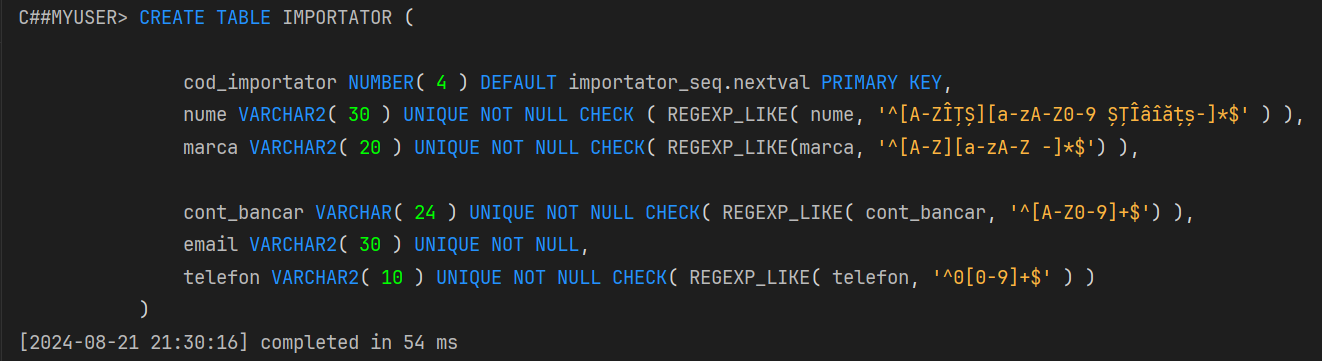
*Figura 4.2 – Rezultat creare tabel REPREZENTANȚĂ*

CREATE TABLE SERVICE (  
  
 cod\_service NUMBER( 3 ) DEFAULT serivice\_seq.nextval PRIMARY KEY,  
 cod\_reprezentanta NUMBER( 3 ) NOT NULL REFERENCES REPREZENTANTA( cod\_reprezentanta ),  
 email VARCHAR2( 30 ) UNIQUE NOT NULL,  
 telefon VARCHAR2( 10 ) UNIQUE NOT NULL CHECK( REGEXP\_LIKE( telefon, '^0[0-9]+$' ) )  
);



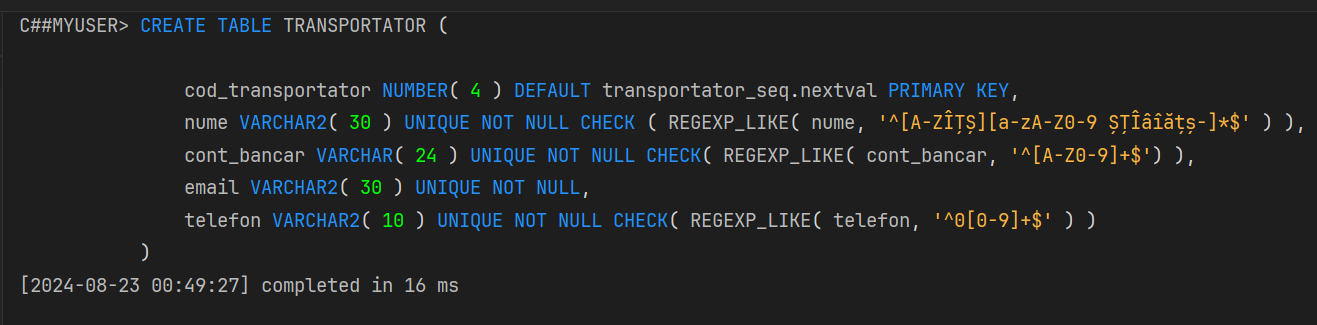
*Figura 4.3 – Rezultat creare tabel SERVICE*

CREATE TABLE IMPORTATOR (  
  
 cod\_importator NUMBER( 4 ) DEFAULT importator\_seq.nextval PRIMARY KEY,  
 nume VARCHAR2( 10 ) UNIQUE NOT NULL CHECK ( REGEXP\_LIKE( nume, '^[A-ZÎȚȘ][a-zA-Z0-9 ȘȚÎâîățș-]\*$' ) ),  
 marca VARCHAR2( 20 ) UNIQUE NOT NULL CHECK( REGEXP\_LIKE(marca, '^[A-Z][a-zA-Z -]\*$') ),  
  
 cont\_bancar VARCHAR( 24 ) UNIQUE NOT NULL CHECK( REGEXP\_LIKE( cont\_bancar, '^[A-Z0-9]+$') ),  
 email VARCHAR2( 30 ) UNIQUE NOT NULL,  
 telefon VARCHAR2( 10 ) UNIQUE NOT NULL CHECK( REGEXP\_LIKE( telefon, '^0[0-9]+$' ) )  
);



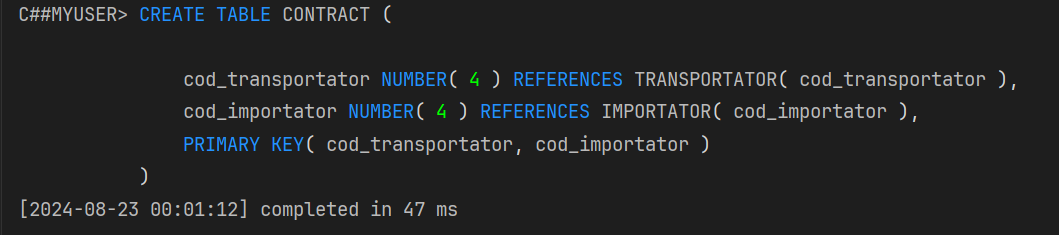
*Figura 4.4 – Rezultat creare tabel IMPORTATOR*

CREATE TABLE TRANSPORTATOR (  
  
 cod\_transportator NUMBER( 4 ) DEFAULT transportator\_seq.nextval PRIMARY KEY,  
 nume VARCHAR2( 30 ) UNIQUE NOT NULL CHECK ( REGEXP\_LIKE( nume, '^[A-ZÎȚȘ][a-zA-Z0-9 ȘȚÎâîățș-]\*$' ) ),  
 cont\_bancar VARCHAR( 24 ) UNIQUE NOT NULL CHECK( REGEXP\_LIKE( cont\_bancar, '^[A-Z0-9]+$') ),  
 email VARCHAR2( 30 ) UNIQUE NOT NULL,  
 telefon VARCHAR2( 10 ) UNIQUE NOT NULL CHECK( REGEXP\_LIKE( telefon, '^0[0-9]+$' ) )  
);



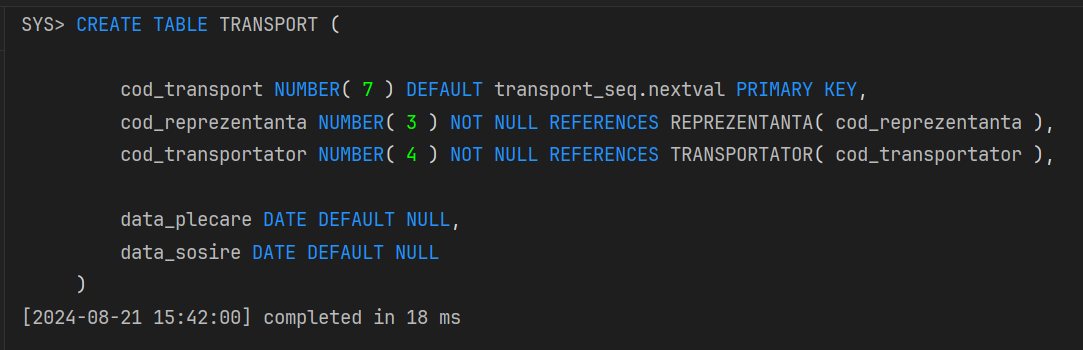
*Figura 4.5 – Rezultat creare tabel TRANSPORTATOR*

CREATE TABLE CONTRACT (  
  
 cod\_transportator NUMBER( 4 ) REFERENCES TRANSPORTATOR( cod\_transportator ),  
 cod\_importator NUMBER( 4 ) REFERENCES IMPORTATOR( cod\_importator ),  
 PRIMARY KEY( cod\_transportator, cod\_importator )  
);



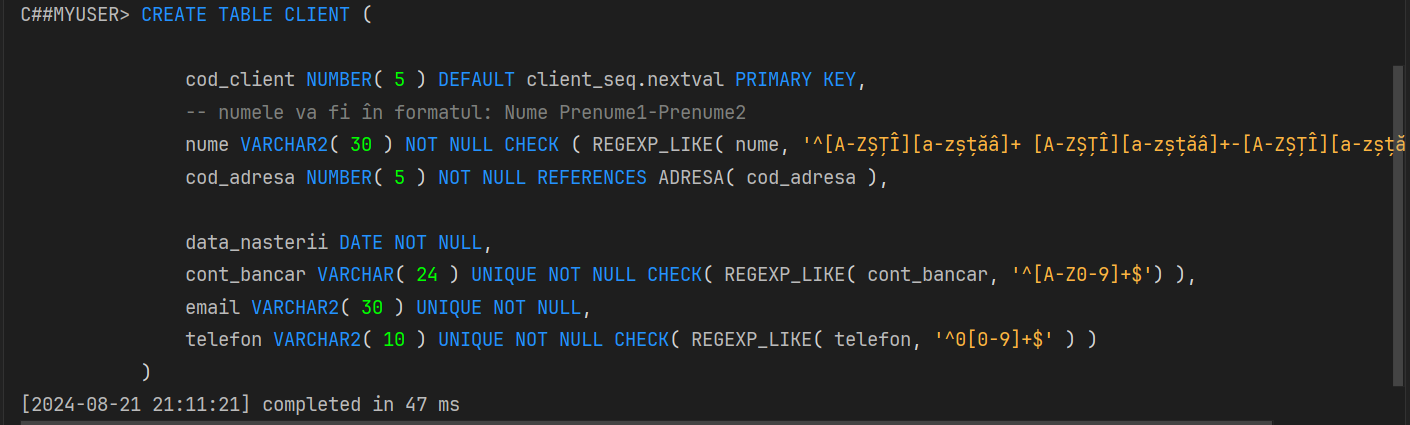
*Figura 4.6 – Rezultat creare tabel CONTRACT*

CREATE TABLE TRANSPORT (  
  
 cod\_transport NUMBER( 7 ) DEFAULT transport\_seq.nextval PRIMARY KEY,  
 cod\_reprezentanta NUMBER( 3 ) NOT NULL REFERENCES REPREZENTANTA( cod\_reprezentanta ),  
 cod\_transportator NUMBER( 4 ) NOT NULL REFERENCES TRANSPORTATOR( cod\_transportator ),  
  
 data\_plecare DATE DEFAULT NULL,  
 data\_sosire DATE DEFAULT NULL  
);

**

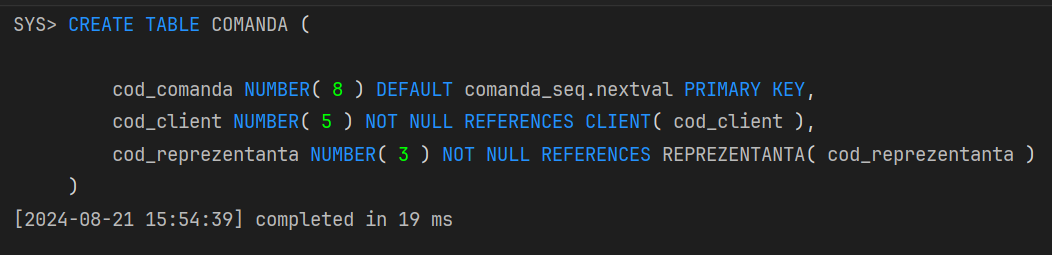
*Figura 4.7 – Rezultat creare tabel TRANSPORT*

CREATE TABLE CLIENT (  
  
 cod\_client NUMBER( 5 ) DEFAULT client\_seq.nextval PRIMARY KEY,  
 *-- numele va fi în formatul: Nume Prenume1-Prenume2* nume VARCHAR2( 30 ) NOT NULL,  
 cod\_adresa NUMBER( 5 ) NOT NULL REFERENCES ADRESA( cod\_adresa ),  
  
 data\_nasterii DATE NOT NULL,  
 cont\_bancar VARCHAR( 24 ) UNIQUE NOT NULL CHECK( REGEXP\_LIKE( cont\_bancar, '^[A-Z0-9]+$') ),  
 email VARCHAR2( 30 ) UNIQUE NOT NULL,  
 telefon VARCHAR2( 10 ) UNIQUE NOT NULL CHECK( REGEXP\_LIKE( telefon, '^0[0-9]+$' ) )  
);



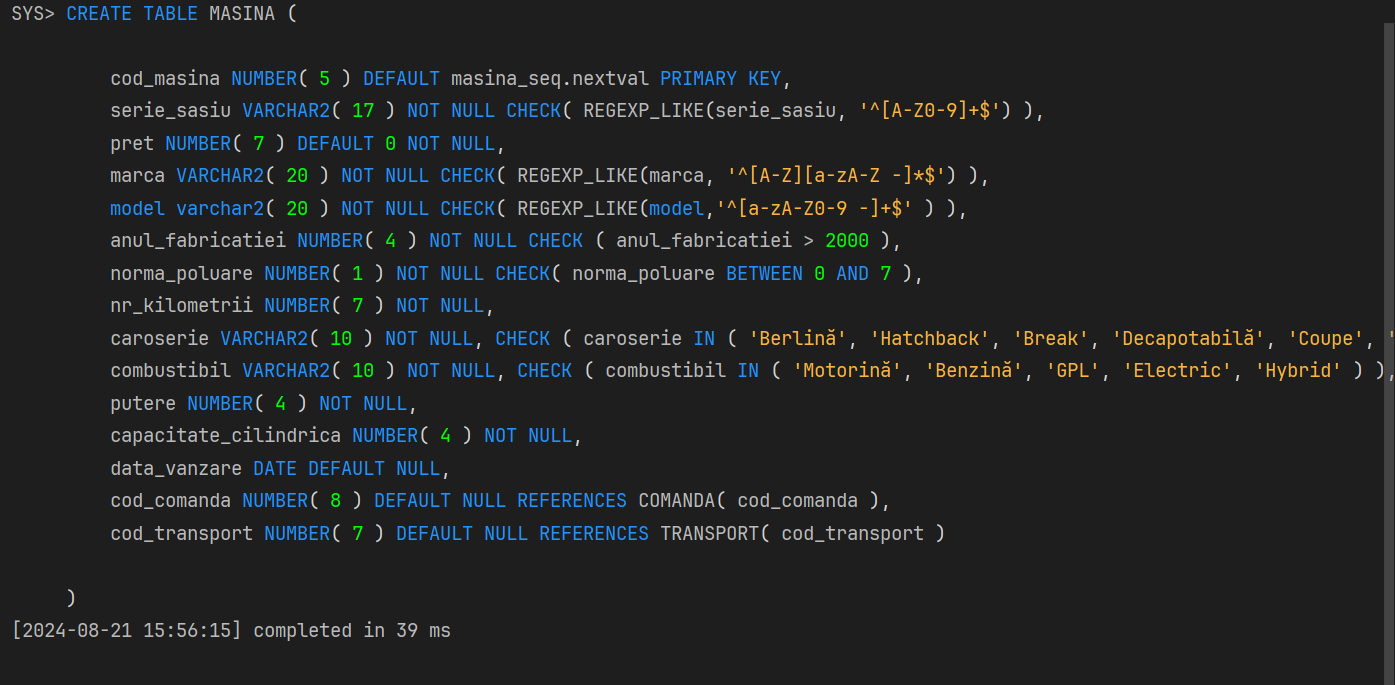
*Figura 4.8 – Rezultat creare tabel CLIENT*

CREATE TABLE COMANDA (  
  
 cod\_comanda NUMBER( 8 ) DEFAULT comanda\_seq.nextval PRIMARY KEY,  
 cod\_client NUMBER( 5 ) NOT NULL REFERENCES CLIENT( cod\_client ),  
 cod\_reprezentanta NUMBER( 3 ) NOT NULL REFERENCES REPREZENTANTA( cod\_reprezentanta )  
);

**

*Figura 4.9 – Rezultat creare tabel COMANDA*

CREATE TABLE MASINA (  
  
 cod\_masina NUMBER( 5 ) DEFAULT masina\_seq.nextval PRIMARY KEY,  
 serie\_sasiu VARCHAR2( 17 ) NOT NULL CHECK( REGEXP\_LIKE(serie\_sasiu, '^[A-Z0-9]+$') ),  
 pret NUMBER( 7 ) DEFAULT 0 NOT NULL,  
 marca VARCHAR2( 20 ) NOT NULL CHECK( REGEXP\_LIKE(marca, '^[A-Z][a-zA-Z -]\*$') ),  
 model varchar2( 20 ) NOT NULL CHECK( REGEXP\_LIKE(model,'^[a-zA-Z0-9 -]+$' ) ),  
  
 anul\_fabricatiei NUMBER( 4 ) NOT NULL CHECK ( anul\_fabricatiei > 2000 ),  
 norma\_poluare NUMBER( 1 ) NOT NULL CHECK( norma\_poluare BETWEEN 0 AND 7 ),  
 nr\_kilometrii NUMBER( 7 ) NOT NULL,  
 caroserie VARCHAR2( 15 ) NOT NULL, CHECK ( caroserie IN ( 'Berlină', 'Hatchback', 'Break', 'Decapotabilă', 'Coupe', 'SUV',  
 'Utilitară' ) ),  
 combustibil VARCHAR2( 10 ) NOT NULL, CHECK ( combustibil IN ( 'Motorină', 'Benzină', 'GPL', 'Electric', 'Hybrid' ) ),  
  
 putere NUMBER( 4 ) NOT NULL,  
 capacitate\_cilindrica NUMBER( 4 ) NOT NULL,  
 data\_vanzare DATE DEFAULT NULL,  
 cod\_comanda NUMBER( 8 ) DEFAULT NULL REFERENCES COMANDA( cod\_comanda ),  
 cod\_transport NUMBER( 7 ) DEFAULT NULL REFERENCES TRANSPORT( cod\_transport )  
  
);

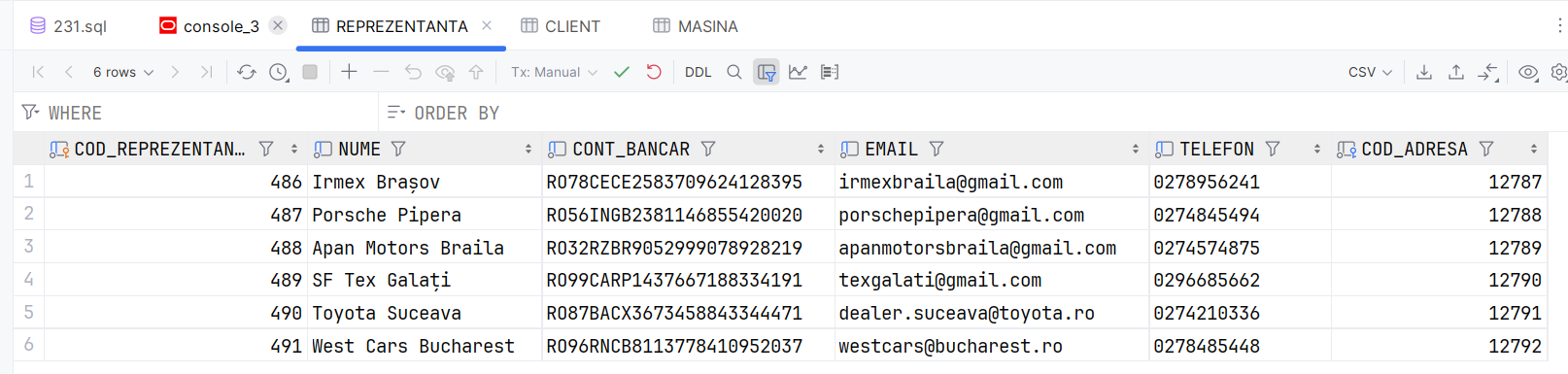
**

*Figura 4.10 – Rezultat creare tabel MASINA*

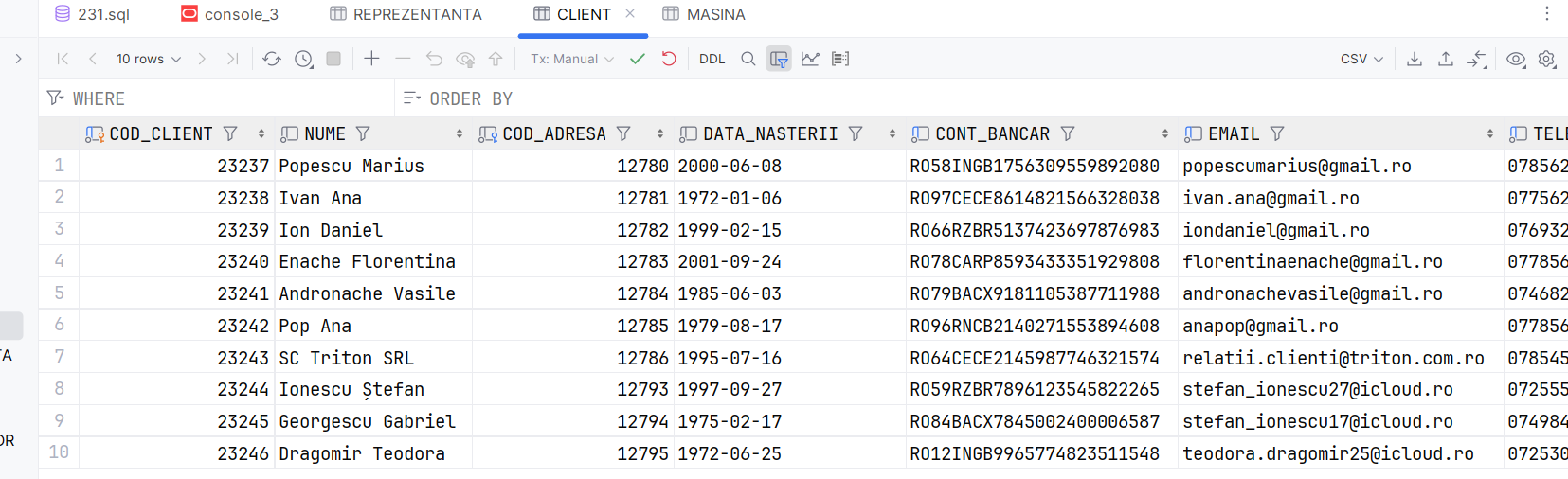
# 5) Adăugați informații coerente în tabelele create (minim 5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru fiecare tabelă asociativă).

*-- INSERARI  
-- Importatori*INSERT INTO IMPORTATOR( nume, marca, cont\_bancar, email, telefon ) *-- MG* VALUES( 'Quantum Auto Max', 'MG', 'RO89BTRL1064660423378420', 'quantumauto@gmail.ro', '0278456327' );  
  
INSERT INTO IMPORTATOR( nume, marca, cont\_bancar, email, telefon ) *-- Dacia* VALUES( 'Romcar Holdings SA', 'Dacia', 'RO64RNCB6779891571280995', 'romcarhds@dacia.ro', '0287943278' );  
  
INSERT INTO IMPORTATOR( nume, marca, cont\_bancar, email, telefon ) *-- BMW* VALUES( 'BMW Merger SRL', 'BMW', 'RO89BACX8202993140258288', 'mergerimport@bmw.ro', '0237379998' );  
  
INSERT INTO IMPORTATOR( nume, marca, cont\_bancar, email, telefon ) *-- Audi* VALUES ( 'Mazuma Imports SRL', 'Audi', 'RO95RZBR7096337779152766', 'mazumaimports@romania.ro', '0235985147' );  
  
INSERT INTO IMPORTATOR( nume, marca, cont\_bancar, email, telefon ) *-- Honda* VALUES ( 'Honda Trading', 'Honda', 'RO78CECE2030043488976256', 'hondatrading@honda.ro', '0256423056' );  
  
INSERT INTO IMPORTATOR( nume, marca, cont\_bancar, email, telefon ) *-- Subaru* VALUES ( 'Subaru Motors Trading ', 'Subaru', 'RO86INGB3771006962910725', 'subarumotorstrading@romania.ro', '0242067390' );  
  
INSERT INTO IMPORTATOR( nume, marca, cont\_bancar, email, telefon ) *-- Toyota* VALUES ( 'Toyota Romania', 'Toyota', 'RO62BACX3773216852419517', 'relatii.clienti@toyota.ro', '0285604304' );  
  
INSERT INTO IMPORTATOR( nume, marca, cont\_bancar, email, telefon ) *-- Skoda* VALUES ( 'Compexit Group', 'Skoda', 'RO42RNCB3352704571546788', 'complexitgroup@skoda.ro', '0274512063' );  
  
INSERT INTO IMPORTATOR( nume, marca, cont\_bancar, email, telefon ) *-- Porsche* VALUES ( 'Porsche Romania', 'Porsche', 'RO27INGB4508326272110825', 'porscheromania@porsche.ro', '0299756025' );  
  
INSERT INTO IMPORTATOR( nume, marca, cont\_bancar, email, telefon ) *-- Mercedes* VALUES ( 'Mercedes-Benz Romania SRL', 'Mercedes', 'RO96CECE2290238112594701', 'import.romania@mercedes.ro', '0285426018' );  
  
INSERT INTO IMPORTATOR( nume, marca, cont\_bancar, email, telefon ) *-- Hyundai* VALUES ( 'Țiriac Auto', 'Hyundai', 'RO45BACX0111255225821108', 'hyundai@romania.ro', '0202458234' );  
  
INSERT INTO IMPORTATOR( nume, marca, cont\_bancar, email, telefon ) *-- Ford* VALUES ( 'Cefin Auto SA', 'Ford', 'RO75INGB1531018827801352', 'cefinautosa@ford.ro', '0204562475' );  
  
INSERT INTO IMPORTATOR( nume, marca, cont\_bancar, email, telefon ) *-- Kia* VALUES ( 'Premium Auto', 'Kia', 'RO63RNCB1263599830910089', 'premiumauto@romania.ro', '0278956240' );  
  
INSERT INTO IMPORTATOR( nume, marca, cont\_bancar, email, telefon ) *-- Renault* VALUES ( 'Renault Commercial Roumanie', 'Renault', 'RO62CARP6567559359820225', 'renaultcommercial@romania.ro', '0255623410' );  
  
INSERT INTO IMPORTATOR( nume, marca, cont\_bancar, email, telefon ) *-- Mazda* VALUES ( 'Mazda Romania', 'Mazda', 'RO96CECE8158201480923475', 'mazdaromania@mazda.ro', '0203645219' );  
  
INSERT INTO IMPORTATOR( nume, marca, cont\_bancar, email, telefon ) *-- Volkswagen* VALUES ( 'German Motors SRL', 'Volkswagen', 'RO74INGB8743937258105468', 'germanmotors@volkswagen.ro', '0266953415' );  
  
  
*-- Adrese Clienti*INSERT INTO ADRESA( localitate, judet ,strada, numar, bloc, scara, etaj, apartament )  
 VALUES ( 'Turceni', 'Gorj', 'Eremia', 56, 'G', '4', 7, 130 );  
  
INSERT INTO ADRESA( localitate, judet ,strada, numar, bloc, scara, etaj, apartament )  
 VALUES ( 'Baia Mare', 'Maramureș', 'Georgescu', 201, 'B2', 'K', 4, 421 );  
  
INSERT INTO ADRESA( localitate, judet ,strada, numar, bloc, scara, etaj, apartament )  
 VALUES ( 'Deta', 'Timiș', 'Generalilor', 78, 'Turn', 8, 3, 14 );  
  
INSERT INTO ADRESA( localitate, judet ,strada, numar, bloc, scara, etaj, apartament )  
 VALUES ( 'Adjud', 'Vrancea', 'Ghioceilor', 21, 'C7', '10', 5, 29 );  
  
INSERT INTO ADRESA( localitate, judet ,strada, numar, bloc, scara, etaj, apartament )  
 VALUES ( 'Orșova', 'Mehedinți', 'Jiului', 45, 'H1', 'B', 8, 89 );  
  
INSERT INTO ADRESA( localitate, judet ,strada, numar )  
 VALUES ( 'Borsec', 'Harghita', 'Câmpului', 63 );  
  
INSERT INTO ADRESA( localitate, judet ,strada, numar, bloc, etaj ) *-- Adresa unei persoane juridice, într-o clădire de birouri* VALUES ( 'Brăila', 'Brăila', 'Calea Călărașilor', 56, 'Sky 2' ,4 );  
  
  
*-- Adrese Reprezentante*INSERT INTO ADRESA( localitate, judet ,strada, numar )  
 VALUES ( 'Brașov', 'Brașov', 'Dunărea', 154 );  
  
INSERT INTO ADRESA( localitate, judet ,strada, numar )  
 VALUES ( 'Sector 2', 'București', 'Șoseaua Pipera', 346 ); *--pipera*INSERT INTO ADRESA( localitate, judet ,strada, numar )  
 VALUES ( 'Brăila', 'Brăila', 'Șoseaua Dig Brăila-Galați', 8 );  
  
INSERT INTO ADRESA( localitate, judet ,strada, numar )  
 VALUES ( 'Galați', 'Galați', 'Verdun', 13 );  
  
INSERT INTO ADRESA( localitate, judet ,strada, numar )  
 VALUES( 'Suceava', 'Suceava', 'Siret', 79 );  
  
INSERT INTO ADRESA( localitate, judet, strada, numar)  
 VALUES( 'Sector 6', 'București', 'Preciziei', 14 );  
  
INSERT INTO ADRESA( localitate, judet, strada, numar, bloc, scara, etaj, apartament )  
 VALUES( 'Sector 3', 'București', 'Bulevardul Theodor Pallady', 375, 'B7', '3' ,4, 26 );  
  
INSERT INTO ADRESA( localitate, judet, strada, numar, bloc, scara, etaj, apartament )  
 VALUES( 'Sector 6', 'București', 'Bulevardul Iuliu Maniu', 185, 'E4', '7', 10, 98 );  
  
INSERT INTO ADRESA( localitate, judet, strada, numar, bloc, scara, etaj, apartament )  
 VALUES( 'Constanța', 'Constanța', 'Bulevardul Aurel Vlaicu', 280, 'A1', '9', 4, 11 );  
  
  
*-- Transportatori*INSERT INTO TRANSPORTATOR( nume, cont\_bancar, email, telefon )  
 VALUES( 'Transport Auto SRL', 'RO78CECE7919463267504484', 'transportauto@romania.ro', '0744756210');  
  
INSERT INTO TRANSPORTATOR( nume, cont\_bancar, email, telefon )  
 VALUES( 'Auto Plus', 'RO56INGB7866400372629299', 'autoplus@transport.ro', '0775621032');  
  
INSERT INTO TRANSPORTATOR( nume, cont\_bancar, email, telefon )  
 VALUES( 'Nonstop Transport', 'RO77RZBR5821134657564557', 'nonstop@transport.ro', '0755623201');  
  
INSERT INTO TRANSPORTATOR( nume, cont\_bancar, email, telefon )  
 VALUES( 'Transmarian', 'RO67CARP8294958279033040', 'transmarian@romania.ro', '0756923014');  
  
INSERT INTO TRANSPORTATOR( nume, cont\_bancar, email, telefon )  
 VALUES( 'Transauto', 'RO61BACX4786626493509428', 'transauto@transport.ro', '0759624107');  
  
INSERT INTO TRANSPORTATOR( nume, cont\_bancar, email, telefon )  
 VALUES( 'Autos Transport', 'RO36RNCB4865342064890901', 'autostransport@romania.ro', '0767634220');  
  
  
*-- Reprezentanțe*INSERT INTO REPREZENTANTA( nume, cont\_bancar, email, telefon, cod\_adresa )  
 VALUES( 'Irmex Brașov', 'RO78CECE2583709624128395', 'irmexbraila@gmail.com', '0278956241', 12787 );  
  
INSERT INTO REPREZENTANTA( nume, cont\_bancar, email, telefon, cod\_adresa )  
 VALUES( 'Porsche Pipera', 'RO56INGB2381146855420020', 'porschepipera@gmail.com', '0274845494', 12788 );  
  
INSERT INTO REPREZENTANTA( nume, cont\_bancar, email, telefon, cod\_adresa )  
 VALUES( 'Apan Motors Braila', 'RO32RZBR9052999078928219', 'apanmotorsbraila@gmail.com', '0274574875', 12789 );  
  
INSERT INTO REPREZENTANTA( nume, cont\_bancar, email, telefon, cod\_adresa )  
 VALUES( 'SF Tex Galați', 'RO99CARP1437667188334191', 'texgalati@gmail.com', '0296685662', 12790 );  
  
INSERT INTO REPREZENTANTA( nume, cont\_bancar, email, telefon, cod\_adresa )  
 VALUES( 'Toyota Suceava', 'RO87BACX3673458843344471', 'dealer.suceava@toyota.ro', '0274210336', 12791 );  
  
INSERT INTO REPREZENTANTA( nume, cont\_bancar, email, telefon, cod\_adresa )  
 VALUES( 'West Cars Bucharest', 'RO96RNCB8113778410952037', 'westcars@bucharest.ro', '0278485448', 12792 );  
  
  
*-- Service-uri*INSERT INTO SERVICE ( cod\_reprezentanta, email, telefon )  
 VALUES( 486, 'irmex\_brasov@service.ro', '0285963241' );  
  
INSERT INTO SERVICE ( cod\_reprezentanta, email, telefon )  
 VALUES( 487, 'porsche.pipera@service.ro', '0278956321' );  
  
INSERT INTO SERVICE ( cod\_reprezentanta, email, telefon )  
 VALUES( 488, 'service.braila@apan.ro', '0262539578' );  
  
INSERT INTO SERVICE ( cod\_reprezentanta, email, telefon )  
 VALUES( 489, 'service@sftex.ro', '0278956241' );  
  
INSERT INTO SERVICE ( cod\_reprezentanta, email, telefon )  
 VALUES( 490, 'service.suceava@toyota.ro', '0299658213' );  
  
INSERT INTO SERVICE ( cod\_reprezentanta, email, telefon )  
 VALUES( 491, 'service@westcars.ro', '0256958741' );  
  
  
*-- Clienti*INSERT INTO CLIENT( nume, cod\_adresa, data\_nasterii, cont\_bancar, email, telefon )  
 VALUES ( 'Popescu Marius', 12780, '8-JUN-2000', 'RO58INGB1756309559892080', 'popescumarius@gmail.ro', '0785624130' );  
  
INSERT INTO CLIENT( nume, cod\_adresa, data\_nasterii, cont\_bancar, email, telefon )  
 VALUES ( 'Ivan Ana', 12781, '6-JAN-1972', 'RO97CECE8614821566328038', 'ivan.ana@gmail.ro', '0775624130' );  
  
INSERT INTO CLIENT( nume, cod\_adresa, data\_nasterii, cont\_bancar, email, telefon )  
 VALUES ( 'Ion Daniel', 12782, '15-FEB-1999', 'RO66RZBR5137423697876983', 'iondaniel@gmail.ro', '0769324150' );  
  
INSERT INTO CLIENT( nume, cod\_adresa, data\_nasterii, cont\_bancar, email, telefon )  
 VALUES ( 'Enache Florentina', 12783, '24-SEP-2001', 'RO78CARP8593433351929808', 'florentinaenache@gmail.ro', '0778569240' );  
  
INSERT INTO CLIENT( nume, cod\_adresa, data\_nasterii, cont\_bancar, email, telefon )  
 VALUES ( 'Andronache Vasile', 12784, '3-JUN-1985', 'RO79BACX9181105387711988', 'andronachevasile@gmail.ro', '0746825012' );  
  
INSERT INTO CLIENT( nume, cod\_adresa, data\_nasterii, cont\_bancar, email, telefon )  
 VALUES ( 'Pop Ana', 12785, '17-AUG-1979', 'RO96RNCB2140271553894608', 'anapop@gmail.ro', '0778569241' );  
  
INSERT INTO CLIENT( nume, cod\_adresa, data\_nasterii, cont\_bancar, email, telefon )  
 VALUES( 'SC Triton SRL', 12786, '16-JUL-1995', 'RO64CECE2145987746321574', 'relatii.clienti@triton.com.ro', '0785452147' );  
  
INSERT INTO CLIENT( nume, cod\_adresa, data\_nasterii, cont\_bancar, email, telefon )  
 VALUES( 'Ionescu Ștefan', 12793, '27-SEP-1997', 'RO59RZBR7896123545822265', 'stefan\_ionescu27@icloud.ro', '0725557485' );  
  
INSERT INTO CLIENT( nume, cod\_adresa, data\_nasterii, cont\_bancar, email, telefon )  
 VALUES( 'Georgescu Gabriel', 12794, '17-FEB-1975', 'RO84BACX7845002400006587', 'stefan\_ionescu17@icloud.ro', '0749845254' );  
  
INSERT INTO CLIENT( nume, cod\_adresa, data\_nasterii, cont\_bancar, email, telefon )  
 VALUES( 'Dragomir Teodora', 12795, '25-JUN-1972', 'RO12INGB9965774823511548', 'teodora.dragomir25@icloud.ro', '0725300848' );  
  
  
*-- Contracte*INSERT INTO CONTRACT( cod\_transportator, cod\_importator ) VALUES( 4771, 5814 ) ;  
INSERT INTO CONTRACT( cod\_transportator, cod\_importator ) VALUES( 4772, 5815 ) ;  
INSERT INTO CONTRACT( cod\_transportator, cod\_importator ) VALUES( 4773, 5816 ) ;  
INSERT INTO CONTRACT( cod\_transportator, cod\_importator ) VALUES( 4774, 5817 ) ;  
INSERT INTO CONTRACT( cod\_transportator, cod\_importator ) VALUES( 4775, 5818 ) ;  
INSERT INTO CONTRACT( cod\_transportator, cod\_importator ) VALUES( 4776, 5819 ) ;  
INSERT INTO CONTRACT( cod\_transportator, cod\_importator ) VALUES( 4771, 5820 ) ;  
INSERT INTO CONTRACT( cod\_transportator, cod\_importator ) VALUES( 4772, 5821 ) ;  
INSERT INTO CONTRACT( cod\_transportator, cod\_importator ) VALUES( 4773, 5822 ) ;  
INSERT INTO CONTRACT( cod\_transportator, cod\_importator ) VALUES( 4774, 5823 ) ;  
INSERT INTO CONTRACT( cod\_transportator, cod\_importator ) VALUES( 4775, 5824 ) ;  
INSERT INTO CONTRACT( cod\_transportator, cod\_importator ) VALUES( 4776, 5825 ) ;  
INSERT INTO CONTRACT( cod\_transportator, cod\_importator ) VALUES( 4771, 5826 ) ;  
INSERT INTO CONTRACT( cod\_transportator, cod\_importator ) VALUES( 4772, 5827 ) ;  
INSERT INTO CONTRACT( cod\_transportator, cod\_importator ) VALUES( 4773, 5828 ) ;  
INSERT INTO CONTRACT( cod\_transportator, cod\_importator ) VALUES( 4774, 5829 ) ;  
  
  
  
*-- Comenzi  
-- Popescu Marius comanda un BMW 520d la Apan Motors Brăila în 2016*INSERT INTO COMANDA( cod\_client, cod\_reprezentanta ) VALUES( 23237, 488 );  
  
*-- Popescu Marius comanda un Mercedes E400 la SF Tex Galați în 2018 și vinde BMW 520d*INSERT INTO COMANDA( cod\_client, cod\_reprezentanta ) VALUES( 23237, 489 );  
  
*-- Popescu Marius comanda o Toyota Camry la Toyota Suceava în 2021 și vinde Mercedes E400*INSERT INTO COMANDA( cod\_client, cod\_reprezentanta ) VALUES( 23237, 490 );  
  
*-- Ivan Ana cumpara de la Sf Tex Galați un BMW 520d în 2019 și un Volkswagen Crafter*INSERT INTO COMANDA( cod\_client, cod\_reprezentanta ) VALUES( 23238, 489 );  
  
*-- Ivan Ana cumpara de la Apan Motors Brăila un BMW X3 în 2021 și vinde BMW 520d*INSERT INTO COMANDA( cod\_client, cod\_reprezentanta ) VALUES( 23238, 488 );  
  
*-- Ion Daniel cumpara de la Irmex Brașov un Ford Mondeo în 2016*INSERT INTO COMANDA( cod\_client, cod\_reprezentanta ) VALUES( 23239, 486 );  
  
*-- Ion Daniel cumpara de la Toyota Suceava o Toyota Corolla și vinde un Ford Mondeo*INSERT INTO COMANDA( cod\_client, cod\_reprezentanta ) VALUES( 23239, 490 );  
  
*-- Andronache Vasile cumpara de la Porsche Pipera un Audi S5 in 2024*INSERT INTO COMANDA( cod\_client, cod\_reprezentanta ) VALUES ( 23241, 487 );  
  
*-- Enache Florentina cumpara un Ford Mondeo de la Toyota Suceava in 2023*INSERT INTO COMANDA( cod\_client, cod\_reprezentanta ) VALUES ( 23240, 490 );  
  
*-- Enache Florentina cumpara un Prsche 911 Cabrio de la Porsche Pipera in 2023*INSERT INTO COMANDA( cod\_client, cod\_reprezentanta ) VALUES ( 23240, 487 );  
  
*-- SC Triton SRL cumpără in 2018 3 Loganuri și un Ford Tranzit de la Irmex Brașov*INSERT INTO COMANDA( cod\_client, cod\_reprezentanta ) VALUES( 23243, 486 );  
  
*-- SC Triton SRL cumpără in 2024 2 Mazda 2 și vinde 2 Loganuri la Irmex Brașov*INSERT INTO COMANDA( cod\_client, cod\_reprezentanta ) VALUES( 23243, 486 );  
  
*-- Ionescu Ștefan cumpără un Mercedes E400 de la Toyota Suceava în 2022*INSERT INTO COMANDA( cod\_client, cod\_reprezentanta ) VALUES ( 23244, 490 );  
  
*-- Georgescu Gabriel cumpără un BMW 520d de la Apan Motors Brăila în 2022*INSERT INTO COMANDA( cod\_client, cod\_reprezentanta ) VALUES ( 23245, 488 );  
  
*-- Ivan Ana cumpără în 2024 un Logan de la Irmex Brașov*INSERT INTO COMANDA( cod\_client, cod\_reprezentanta ) VALUES ( 23238, 486 );  
  
*-- Dragomir Teodora cumpără de la Toyota Suceava în 2022 un Kia Stinger*INSERT INTO COMANDA( cod\_client, cod\_reprezentanta ) VALUES( 23246, 490 );  
  
*-- Dragomir Teodora cumpără de la Toyota Suceava în 2023 o Honda Civic și vinde un Kia Stinger*INSERT INTO COMANDA( cod\_client, cod\_reprezentanta ) VALUES( 23246, 490 );  
  
*-- Popescu Marius cumpără de la Toyota Suceava în 2024 un Kia Stinger*INSERT INTO COMANDA( cod\_client, cod\_reprezentanta ) VALUES( 23237, 490 );  
  
*-- Popescu Marius cumpără de la Toyota Suceava în 2024 un Renault Captur și vinde un Kia Stinger*INSERT INTO COMANDA( cod\_client, cod\_reprezentanta ) VALUES( 23237, 490 );  
  
  
  
*-- Transporturi*INSERT INTO TRANSPORT( cod\_reprezentanta, cod\_transportator, data\_plecare, data\_sosire )  
 VALUES ( 488, 4773, '13-JAN-2017', '23-JAN-2017' ); *-- BMW 520d 2016*INSERT INTO TRANSPORT( cod\_reprezentanta, cod\_transportator, data\_plecare, data\_sosire )  
 VALUES ( 489, 4774, '5-JAN-2019', '14-JAN-2019' ); *-- Mercedes E400 2018*INSERT INTO TRANSPORT( cod\_reprezentanta, cod\_transportator, data\_plecare, data\_sosire )  
 VALUES ( 490, 4771, '28-FEB-2022', '8-MAR-2022' ); *-- Toyota Camry 2021*INSERT INTO TRANSPORT( cod\_reprezentanta, cod\_transportator, data\_plecare, data\_sosire )  
 VALUES ( 489, 4774, '27-MAY-2019', '3-JUN-2019' ); *-- Volkswagen Crafter 2019*INSERT INTO TRANSPORT( cod\_reprezentanta, cod\_transportator, data\_plecare, data\_sosire )  
 VALUES ( 488, 4773, '13-MAR-2022', '14-MAR-2022' ); *-- BMW X3 2021*INSERT INTO TRANSPORT( cod\_reprezentanta, cod\_transportator, data\_plecare, data\_sosire )  
 VALUES ( 487, 4774, '18-DEC-2024', '21-DEC-2024' ); *-- Audi S5*INSERT INTO TRANSPORT( cod\_reprezentanta, cod\_transportator, data\_plecare, data\_sosire )  
 VALUES ( 486, 4776, '18-JUL-2016', '21-JUL-2016' ); *-- Ford Mondeo 2016*INSERT INTO TRANSPORT( cod\_reprezentanta, cod\_transportator, data\_plecare, data\_sosire )  
 VALUES ( 490, 4771, '7-JUL-2023', '14-JUL-2023' ); *-- Toyota Corolla 2023*INSERT INTO TRANSPORT( cod\_reprezentanta, cod\_transportator, data\_plecare, data\_sosire )  
 VALUES ( 486, 4772, '23-JUN-2018', '26-JUN-2018' ); *-- Dacia Logan 2018*INSERT INTO TRANSPORT( cod\_reprezentanta, cod\_transportator, data\_plecare, data\_sosire )  
 VALUES ( 486, 4776, '12-SEP-2018', '15-SEP-2018' ); *-- Ford Tranzit 2018*INSERT INTO TRANSPORT( cod\_reprezentanta, cod\_transportator, data\_plecare, data\_sosire )  
 VALUES ( 486, 4773, '2-DEC-2024', '5-DEC-2024' ); *-- Mazda 2 2024*INSERT INTO TRANSPORT( cod\_reprezentanta, cod\_transportator, data\_plecare )  
 VALUES( 491, 4772, '16-NOV-2024' ); *-- Skoda Octavia 2024*INSERT INTO TRANSPORT( cod\_reprezentanta, cod\_transportator )  
 VALUES( 487, 4773 ); *-- Porsche 911 Cabrio*INSERT INTO TRANSPORT( cod\_reprezentanta, cod\_transportator, data\_plecare, data\_sosire )  
 VALUES( 491, 4771, '23-MAR-2023', '24-MAR-2023' ); *-- Kia Stinger 2022*INSERT INTO TRANSPORT( cod\_reprezentanta, cod\_transportator, data\_plecare, data\_sosire )  
 VALUES( 490, 4775, '9-OCT-2024', '10-OCT-2024' ); *-- Honda Civic 2024  
  
  
-- Masini  
-- Popescu Marius comanda un BMW 520d la Apan Motors Brăila în 2016*INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica, data\_vanzare, cod\_comanda, cod\_transport )  
 VALUES( 'WBAKP9C50GD980586', 75000, 'BMW', '520d', 2016, 5, 100, 'Berlină', 'Motorină', 184, 1995, '25-OCT-2016',  
 1249455, 3794325 );  
  
*-- Popescu Marius comanda un Mercedes E400 la SF Tex Galați în 2018 și vinde BMW 520d*INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica, data\_vanzare, cod\_comanda, cod\_transport )  
 VALUES( '4T4BF3EK9BR182409', 75000, 'Mercedez-Benz', 'E400', 2018, 6, 100, 'Berlină', 'Benzină', 333, 3498, '12-DEC-2018',  
 1249456, 3794326 );  
  
*-- Popescu Marius comanda o Toyota Camry la Toyota Suceava în 2021 și vinde Mercedes E400*INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica, data\_vanzare, cod\_comanda, cod\_transport )  
 VALUES( '1C4RDJAG8EC511538', 35000, 'Toyota', 'Camry', 2021, 6, 100, 'Berlină', 'Hybrid', 218, 2487, '8-AUG-2021',  
 1249457, 3794327 );  
  
*-- Ionescu Ștefan cumpără un Mercedes E400 de la Toyota Suceava în 2022*INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica, data\_vanzare, cod\_comanda )  
 VALUES( '4T4BF3EK9BR182409', 25000, 'Mercedez-Benz', 'E400', 2018, 6, 155879, 'Berlină', 'Benzină', 333, 3498, '15-APR-2022',  
 1249467 );  
  
*-- Ivan Ana cumpara de la Sf Tex Galați un BMW 520d în 2019 și un Volkswagen Crafter*INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica, data\_vanzare, cod\_comanda )  
 VALUES( 'WBAKP9C50GD980586', 45000, 'BMW', '520d', 2016, 5, 80000, 'Berlină', 'Motorină', 184, 1995, '12-FEB-2019', 1249458 );  
  
INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica, data\_vanzare, cod\_comanda, cod\_transport )  
 VALUES( 'WVGJV3AXXEW549191', 30000, 'Volkswagen', 'Crafter', 2021, 6, 80, 'Utilitară', 'Motorină', 155, 2254, '12-FEB-2019',  
 1249458, 3794328 );  
  
*-- Ivan Ana cumpara de la Apan Motors Brăila un BMW X3 și vinde un BMW 520d*INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica, data\_vanzare, cod\_comanda, cod\_transport )  
 VALUES( '1GCJTCDE0A8114263', 55000, 'BMW', 'X3', 2021, 6, 100, 'SUV', 'Benzină', 245, 1998, '4-JUN-2021',  
 1249459, 3794329 );  
  
*-- Georgescu Gabriel cumpără un BMW 520d de la Apan Motors Brăila în 2022*INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica, data\_vanzare, cod\_comanda )  
 VALUES( 'WBAKP9C50GD980586', 25000, 'BMW', '520d', 2016, 6, 100, 'Berlină', 'Motorină', 184, 1995, '23-APR-2022' , 1249468 );  
  
*-- West Cars cumpara stoc. Masinile nu au numar de comanda*INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica )  
 VALUES( '4A3AC84843E031749', 20000, 'MG', 'ZS', 2024, 6, 250, 'SUV', 'Benzină', 106, 1498 );  
  
INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica )  
 VALUES( '5GTEN13L1980066EX', 23700, 'MG', '3', 2024, 6, 200, 'Hatchback', 'Hybrid', 194, 1490 );  
  
  
*-- Andronache Vasile cumpara de la Porsche Pipera un Audi S5 in 2024*INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica, data\_vanzare, cod\_comanda, cod\_transport )  
 VALUES( 'WAUGD5440LN008570', 73000, 'Audi', 'S5', 2024, 6, 60, 'Decapotabilă', 'Benzină', 354, 2995, '7-JUL-2024',  
 1249462, 3794330 );  
  
*-- Ion Daniel cumpără de la Irmex Brașov un Ford Mondeo în 2016*INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica, data\_vanzare, cod\_comanda, cod\_transport )  
 VALUES( '1FABP3798FW328823', 28350, 'Ford', 'Mondeo', 2016, 5, 80, 'Hatchback', 'Motorină', 153, 1998, '31-JAN-2016',  
 1249460, 3794331 );  
  
*-- Ion Daniel cumpără de la Toyota Suceava o Toyota Corolla și vinde un Ford Mondeo în 2023*INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica, data\_vanzare, cod\_comanda, cod\_transport )  
 VALUES( 'JT2EL46S9R0502556', 26850, 'Toyota', 'Corolla', 2023, 6, 120, 'Break', 'Hybrid', 196, 1987, '13-JAN-2023',  
 1249461, 3794332 );  
  
*-- Enache Florentina cumpără un Ford Mondeo de la Toyota Suceava in 2023*INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica, data\_vanzare, cod\_comanda )  
 VALUES( '1FABP3798FW328823', 10500, 'Ford', 'Mondeo', 2016, 5, 180878, 'Hatchback', 'Motorină', 153, 1998, '8-AUG-2023', 1249463 );  
  
  
*--14  
-- SC Triton SRL cumpără in 2018 3 Loganuri și un Ford Tranzit de la Irmex Brașov*INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica, data\_vanzare, cod\_comanda, cod\_transport )  
 VALUES( '3C4PFABB1DT246517', 10500, 'Dacia', 'Logan', 2018, 6, 120, 'Berlină', 'GPL', 100, 999, '7-MAY-2018',  
 1249465, 3794333 );  
  
INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica, data\_vanzare, cod\_comanda, cod\_transport )  
 VALUES( '3C3CFFJH3FT630338', 10500, 'Dacia', 'Logan', 2018, 6, 120, 'Berlină', 'GPL', 100, 999, '7-MAY-2018',  
 1249465, 3794333 );  
  
INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica, data\_vanzare, cod\_comanda, cod\_transport )  
 VALUES( 'ZARED33E0S6303319', 10500, 'Dacia', 'Logan', 2018, 6, 120, 'Berlină', 'GPL', 100, 999, '7-MAY-2018',  
 1249465, 3794333 );  
  
INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica, data\_vanzare, cod\_comanda, cod\_transport )  
 VALUES( '1F1SG65683H726608', 25500, 'Ford', 'Tranzit', 2018, 6, 120, 'Utilitară', 'Motorină', 130, 2164, '7-MAY-2018',  
 1249465, 3794334 );  
  
  
*-- SC Triton SRL cumpără in 2024 2 Mazda 2 și o Octavia și vinde 3 Loganuri la Irmex Brașov*INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica )  
 VALUES( '3C3CFFJH3FT630338', 6500, 'Dacia', 'Logan', 2018, 6, 127028, 'Berlină', 'GPL', 100, 999 );  
  
INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica )  
 VALUES( 'ZARED33E0S6303319', 6500, 'Dacia', 'Logan', 2018, 6, 147476, 'Berlină', 'GPL', 100, 999 );  
  
*-- un Logan se vinde, 2 raman pe stoc*INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica, data\_vanzare, cod\_comanda )  
 VALUES( '3C4PFABB1DT246517', 5500, 'Dacia', 'Logan', 2018, 6, 220875, 'Berlină', 'GPL', 100, 999, '23-SEP-2024', 1249469 );  
  
INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica, data\_vanzare, cod\_comanda, cod\_transport )  
 VALUES( '1YVGE31C4S5423092', 23500, 'Mazda', '2', 2024, 6, 75, 'Hatchback', 'Hybrid', 91, 1496, '17-APR-2024',  
 1249466, 3794335 );  
  
INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica, data\_vanzare, cod\_comanda, cod\_transport )  
 VALUES( 'JT4RN01P0N7057480', 23500, 'Mazda', '2', 2024, 6, 75, 'Hatchback', 'Hybrid', 91, 1496, '17-APR-2024',  
 1249466, 3794335 );  
  
INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica, data\_vanzare, cod\_comanda, cod\_transport )  
 VALUES( 'WVW7771J7YW248023', 27500, 'Skoda', 'Octavia', 2024, 6, 75, 'Hatchback', 'Motorină', 150, 1988, '17-APR-2024',  
 1249467, 3794336 );  
  
*-- Enache Florentina cumpara un Porsche 911 Cabrio de la Porsche Pipera in 2023*INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica, data\_vanzare, cod\_comanda, cod\_transport )  
 VALUES( 'WP0AB29935S175494', 225000, 'Porsche', '911 Cabriolet', 2020, 6, 75, 'Decapotabilă', 'Benzină', 450, 2981, '4-MAR-2024',  
 1249464, 3794337 );  
  
  
SELECT C.cod\_comanda, C.cod\_client, cl.nume, m.marca, m.model  
 FROM COMANDA C  
 join CLIENT cl on cl.cod\_client = C.cod\_client  
 join MASINA m on m.cod\_comanda = C.cod\_comanda  
 order by cod\_comanda;  
  
*-- 25  
-- Dragomir Teodora cumpără de la Toyota suceava în 2022 un Kia Stinger*INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica, data\_vanzare, cod\_comanda, cod\_transport )  
 VALUES( 'KNADN5A32C6736025', 45000, 'Kia', 'Stinger', 2022, 6, 300, 'Hatchback', 'Benzină', 304, 2496, '24-NOV-2022',  
 1249471, 3794338 );  
  
*-- Dragomir Teodora cumpără de la Toyota Suceava în 2023 o Honda Civic și vinde un Kia Stinger*INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica, data\_vanzare, cod\_comanda, cod\_transport )  
 VALUES( '1HGCM825X3A802211', 38230, 'Honda', 'Civic', 2023, 6, 70, 'Hatchback', 'Hybrid', 184, 1993, '29-NOV-2024',  
 1249472, 3794339 );  
  
*-- Popescu Marius cumpără de la Toyota Suceava în 2024 un Kia Stinger*INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica, cod\_comanda, data\_vanzare )  
 VALUES( 'KNADN5A32C6736025', 28000, 'Kia', 'Stinger', 2022, 6, 36547, 'Hatchback', 'Benzină', 304, 2496, 1249474, '23-APR-2024' );  
  
  
*-- Popescu Marius cumpără de la Toyota Suceava în 2024 un Renault Captur și vinde un Kia Stinger*INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica, data\_vanzare, cod\_comanda )  
 VALUES( 'VF1BZB1A6BR585200', 26190, 'Renault', 'Captur', 2024, 6, 45, 'SUV', 'Hybrid', 143, 1598, '20-OCT-2024', 1249473 );  
  
*-- Kia Stinger rămâne în inventar la Toyota Suceava*INSERT INTO MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare, nr\_kilometrii, caroserie, combustibil,  
 putere, capacitate\_cilindrica )  
 VALUES( 'KNADN5A32C6736025', 25000, 'Kia', 'Stinger', 2022, 6, 45875, 'Hatchback', 'Benzină', 304, 2496 );  
  
COMMIT;

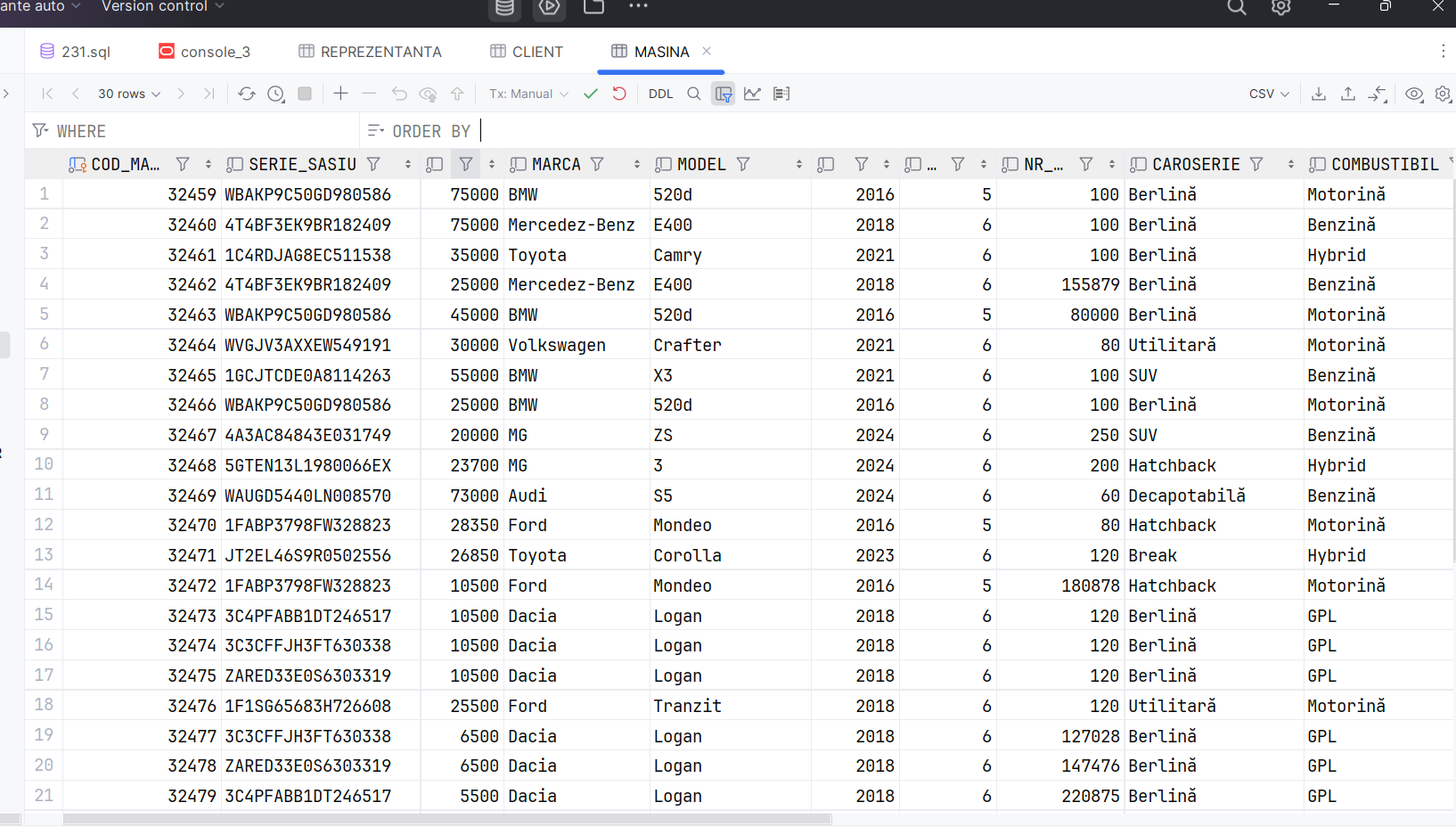
Rezultate inserări:



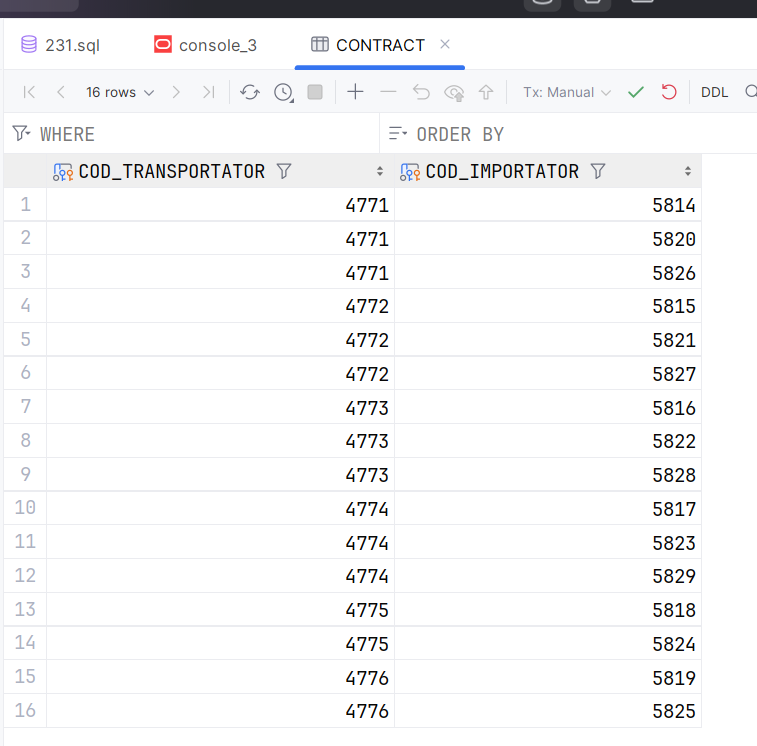
*Figura 5.1 – Rezultat inserare în tabelul REPREZENTANTA*



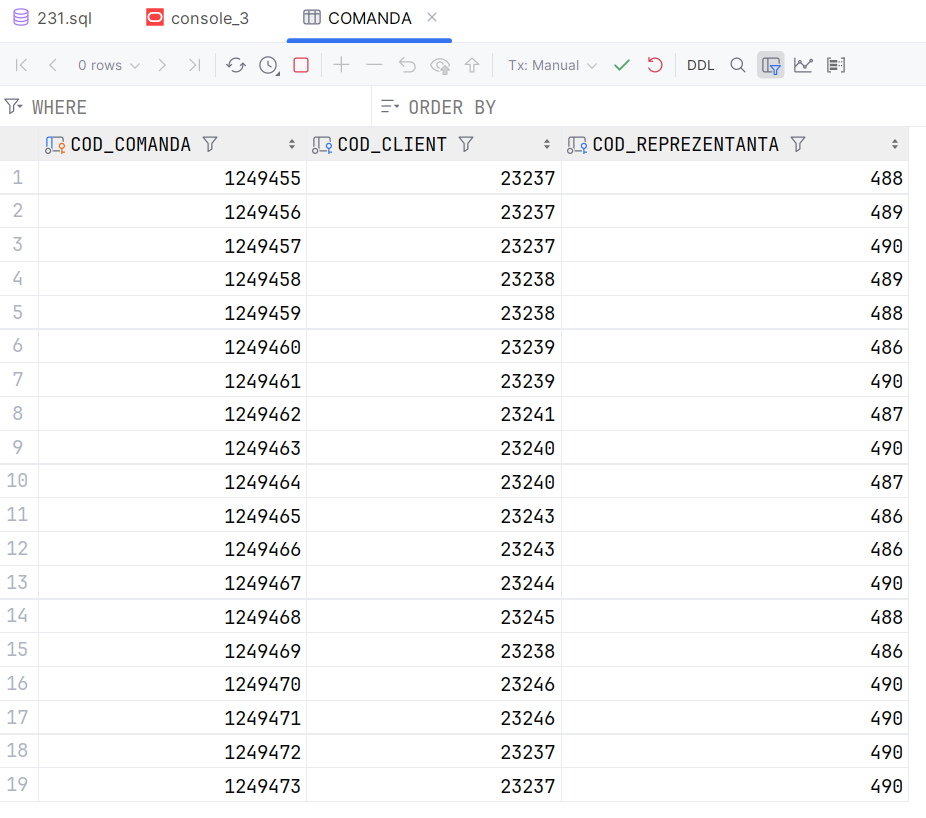
*Figura 5.2 – Rezultat inserare în tabelul CLIENT*



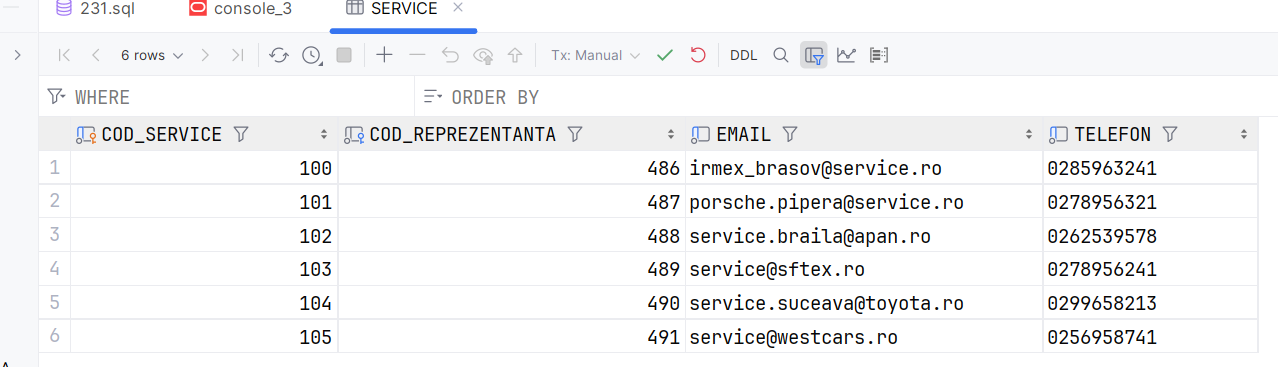
*Figura 5.3 – Rezultat inserare în tabelul MASINA*

**

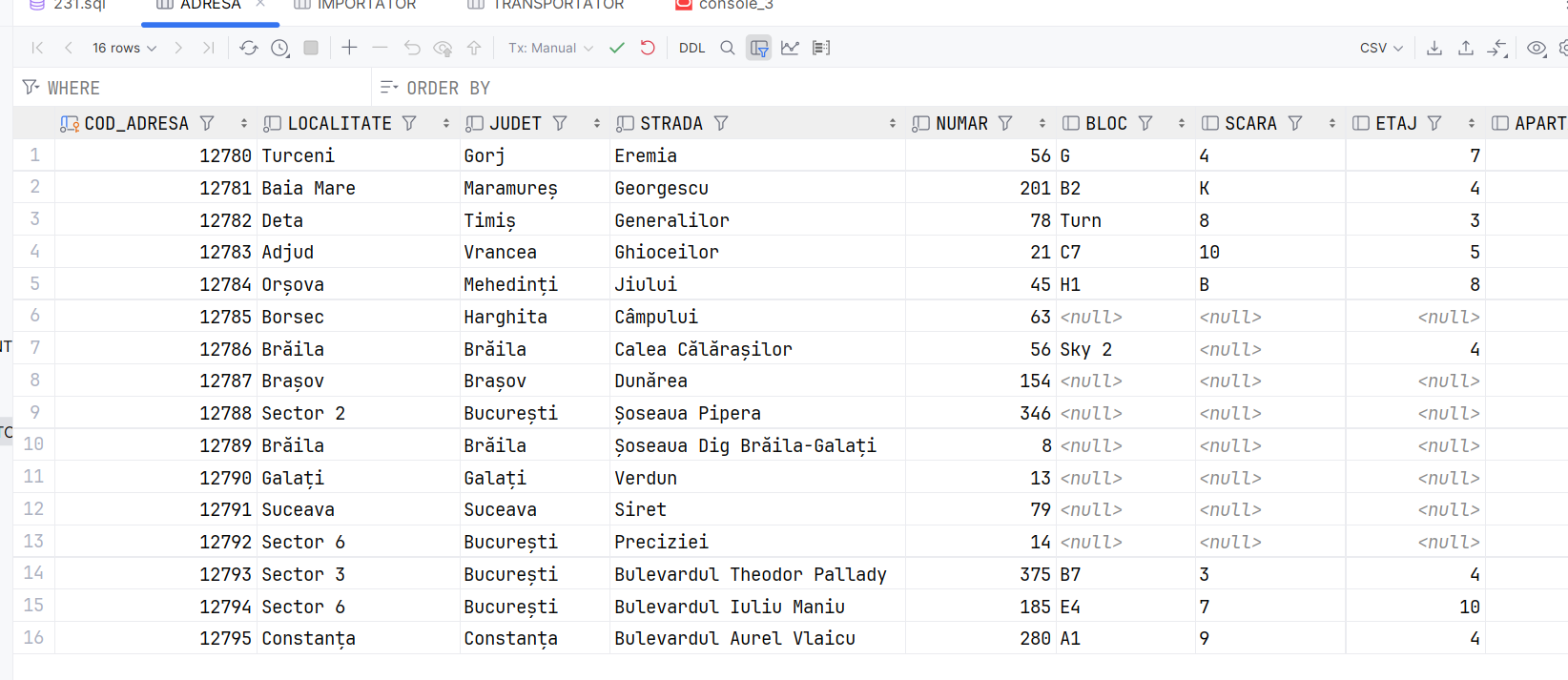
*Figura 5.4 – Rezultat inserare în tabelul CONTRACT*

**

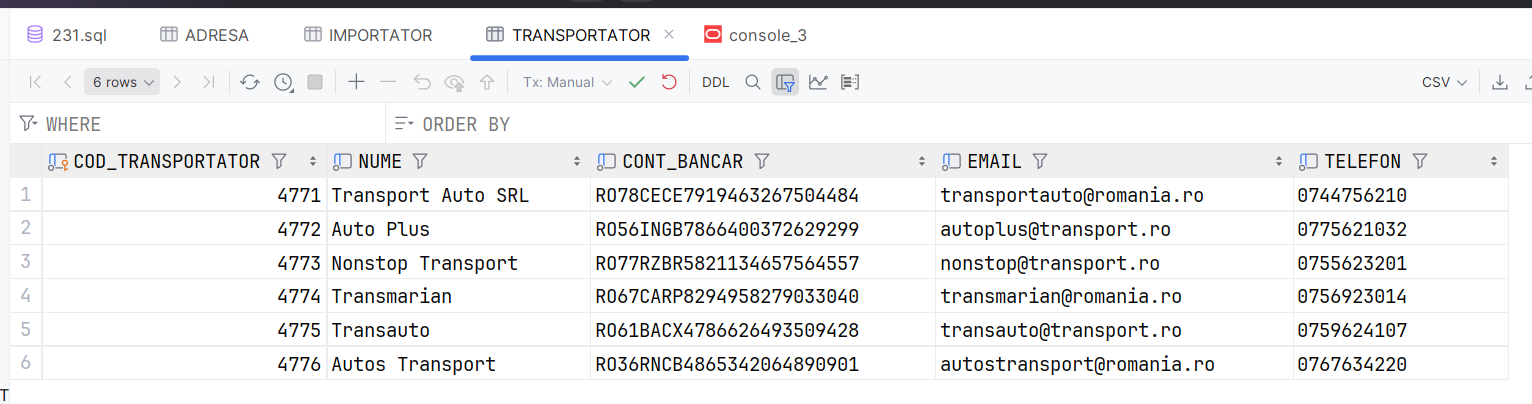
*Figura 5.5 – Rezultat inserare în tabelul COMANDA*

**

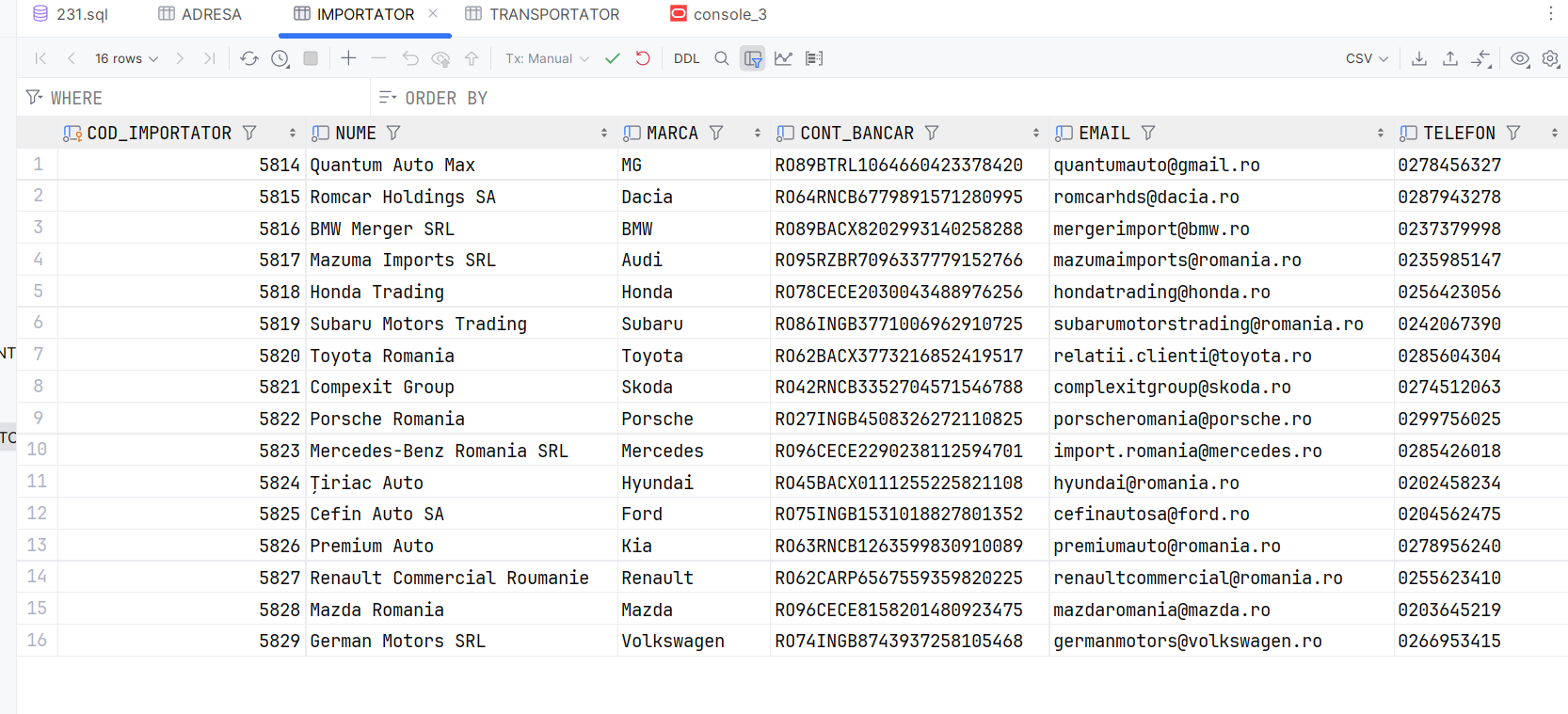
*Figura 5.6 – Rezultat inserare în tabelul SERVICE*



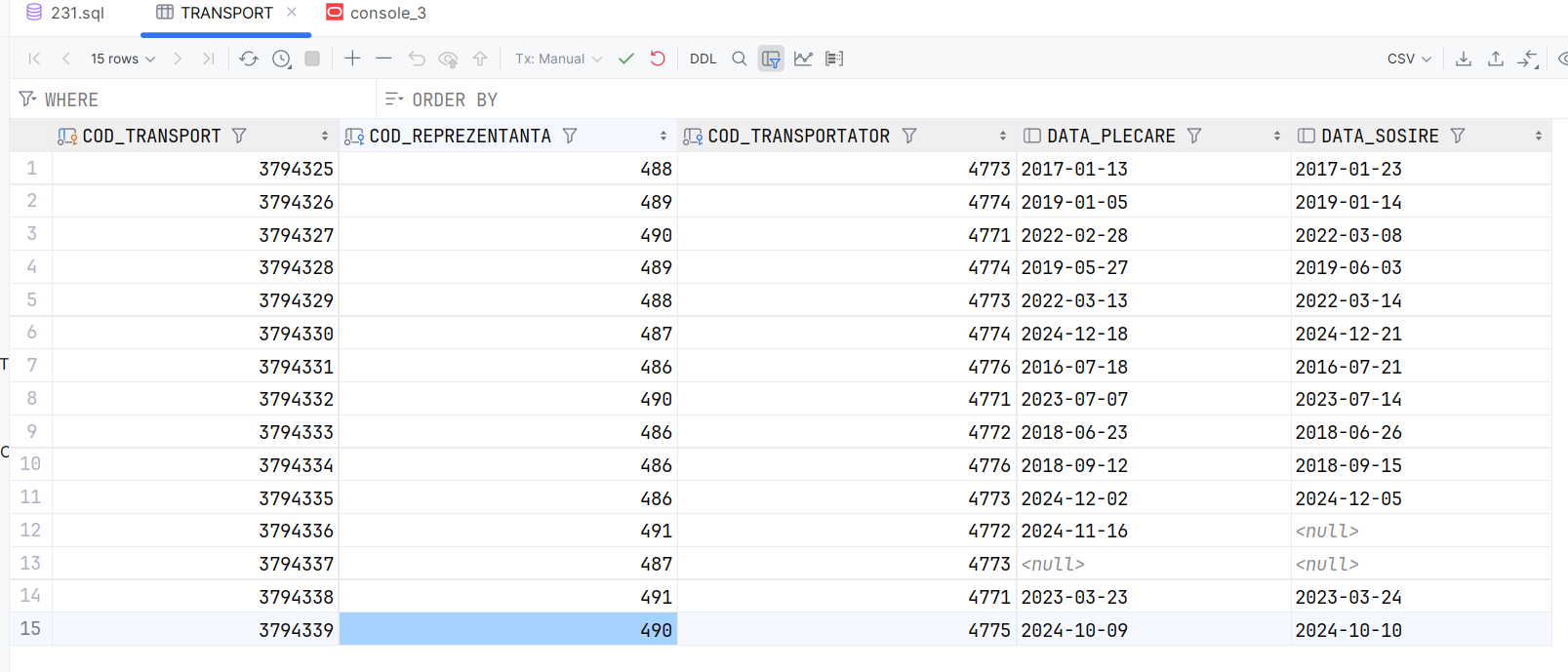
*Figura 5.7 – Rezultat inserare în tabelul ADRESA*



*Figura 5.8 – Rezultat inserare în tabelul TRANSPORTATOR*



*Figura 5.9 – Rezultat inserare în tabelul IMPORTATOR*

**

*Figura 5.10 – Rezultat inserare în tabelul TRANSPORT*

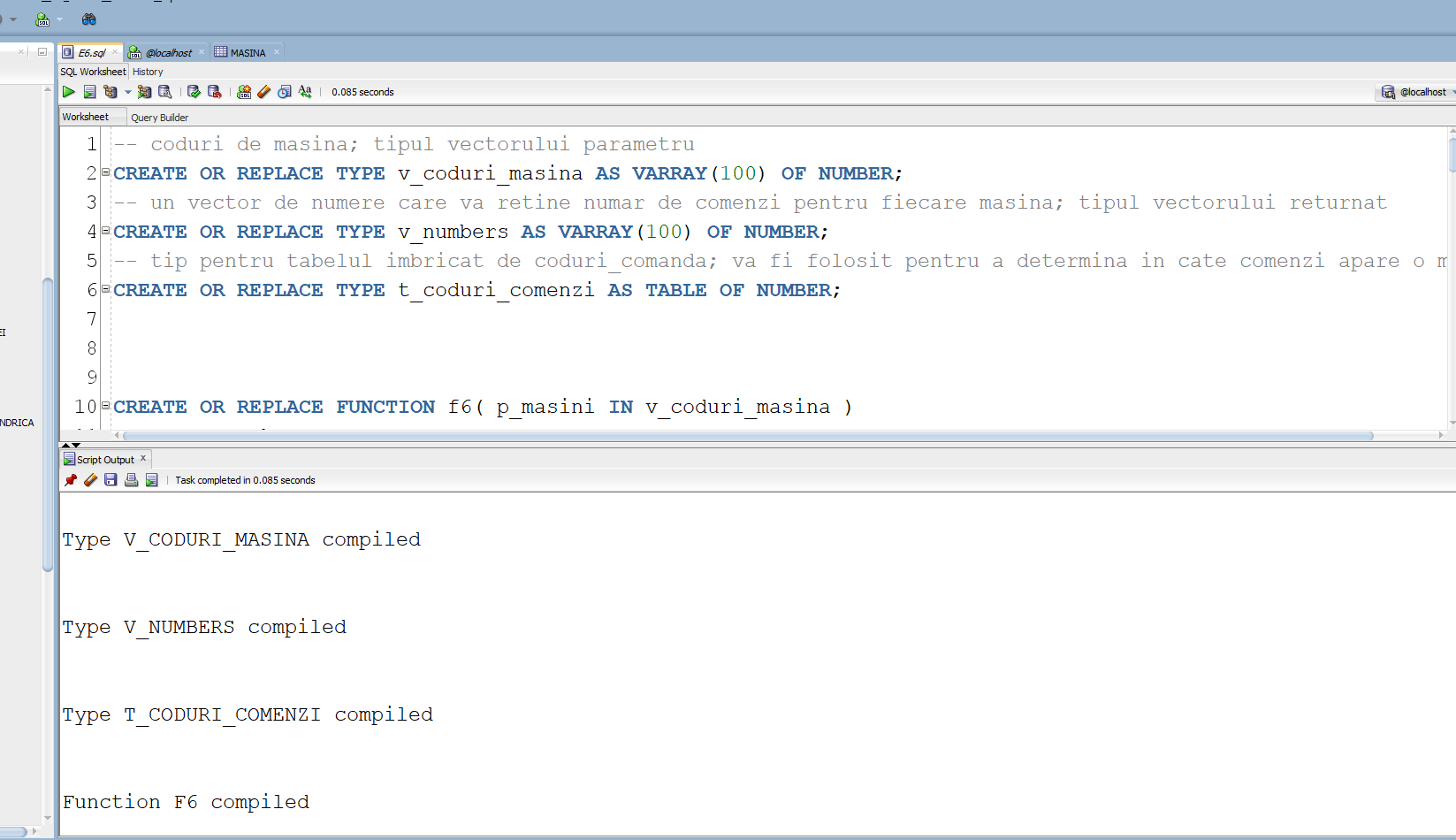
# 6) Formulați în limbaj natural o problemă pe care să o rezolvați folosind un **subprogram stocat independent** care să utilizeze toate cele 3 tipuri de colecții studiate. Apelați subprogramul.

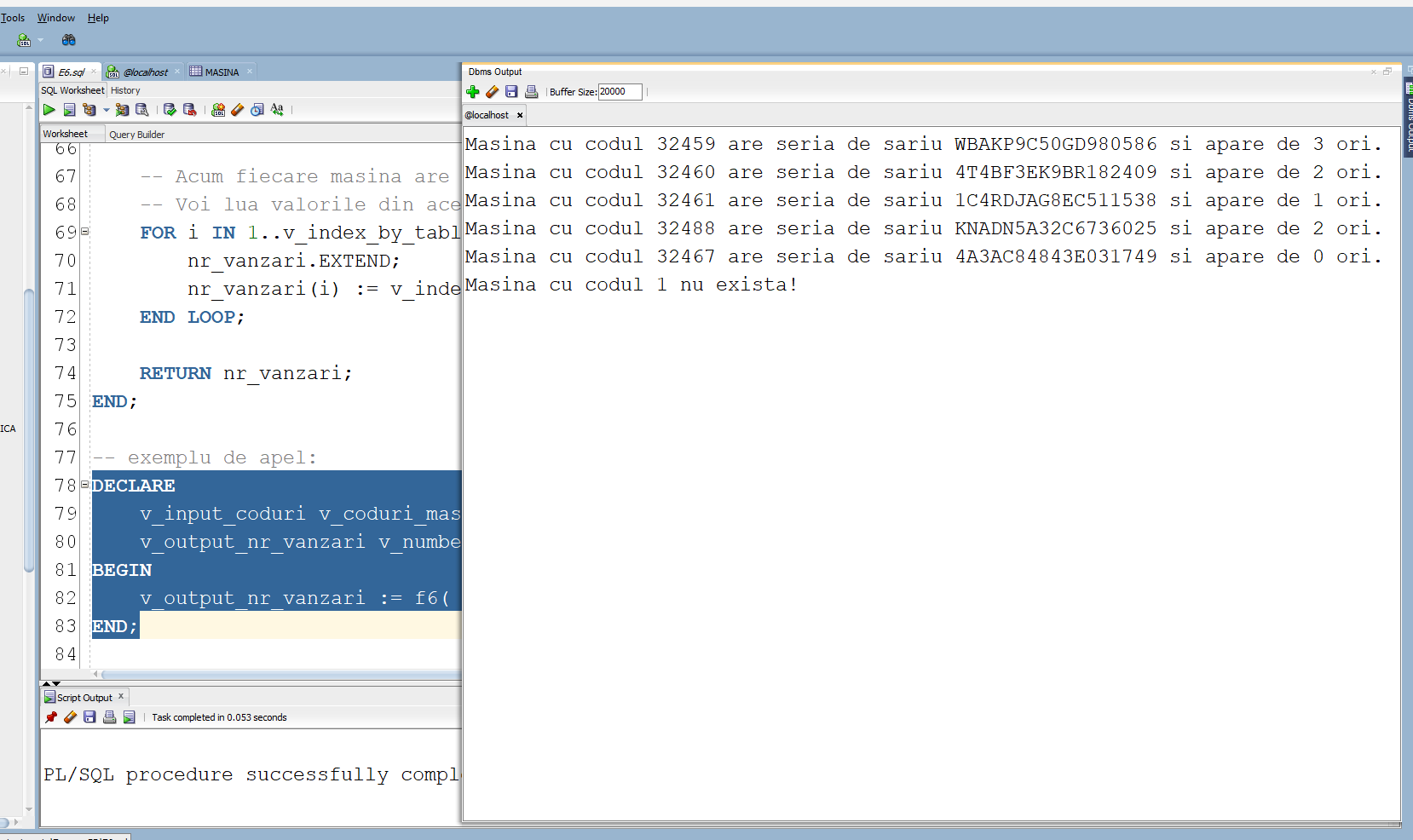
***Cerință***: Fiind dat un vector coduri de mașină, să se determine de câte ori a fost vândută fiecare mașină din vector, folosind un subprogram stocat independent. Rezultatul trebuie returnat într-un vector. Să se trateze cazul în care codul nu există în baza de date.

***Rezolvare***: Voi folosi doi vectori, unul dintre ei va fi parametrul ( coduri de mașini ), iar al doilea va fi rezultatul ( numere de vânzări pentru fiecare mașină ). De asemenea, voi folosi un tabel imbricat pentru a determina numărul de vânzări pentru fiecare mașină și un tabel indexat pentru a stoca perechi ( cod\_mașină, nr\_vânzări ). La final, voi insera valorile din tabloul indexat în vectorul ce va fi returnat ca rezultat.

*-- coduri de masina; tipul vectorului parametru*CREATE OR REPLACE TYPE v\_coduri\_masina AS VARRAY(100) OF NUMBER;  
*-- un vector de numere care va retine numar de comenzi pentru fiecare masina; tipul vectorului returnat*CREATE OR REPLACE TYPE v\_numbers AS VARRAY(100) OF NUMBER;   
*-- tip pentru tabelul imbricat de coduri\_comanda; va fi folosit pentru a determina in cate comenzi apare o masina*CREATE OR REPLACE TYPE t\_coduri\_comenzi AS TABLE OF NUMBER;   
  
  
  
CREATE OR REPLACE FUNCTION *f6*( p\_masini IN v\_coduri\_masina )  
RETURN v\_numbers AS  
  
 TYPE t\_cod\_vanzari IS TABLE OF NUMBER INDEX BY PLS\_INTEGER; *-- index by table ( cod\_masina, nr\_vanzari )* v\_index\_by\_table t\_cod\_vanzari;  
  
 nr\_vanzari v\_numbers := v\_numbers(); *-- de cate ori a fost vanduta fiecare masina* v\_comenzi t\_coduri\_comenzi; *-- vector cu codurile comenzilor in care apare o masina* v\_serie\_sasiu VARCHAR2( 17 );  
   
 no\_data\_found EXCEPTION;  
 v\_nr\_masini\_gasite NUMBER;  
   
BEGIN  
  
 *-- pentru fiecare cod\_masina din parametru, trebuie determinata seria de sasiu aferenta  
 -- apoi, pentru seria de sasiu trebuie rulata cererea  
 -- seria de sasiu apare de mai multe ori in tabel deoarece o masina poate fi vanduta de mai multe ori* FOR i IN 1 .. p\_masini.COUNT LOOP  
   
 BEGIN  
   
 SELECT *COUNT*(*\**)  
 INTO v\_nr\_masini\_gasite  
 FROM C##MYUSER.MASINA M  
 WHERE M.COD\_MASINA = p\_masini(i);  
  
 IF v\_nr\_masini\_gasite = 0 THEN  
 RAISE no\_data\_found;  
 END IF;  
  
 SELECT M.SERIE\_SASIU   
 INTO v\_serie\_sasiu  
 FROM C##MYUSER.MASINA M  
 WHERE M.COD\_MASINA = p\_masini(i);  
   
 EXCEPTION  
 WHEN no\_data\_found THEN  
 DBMS\_OUTPUT.*PUT\_LINE*('Masina cu codul ' || p\_masini(i) || ' nu exista!');  
 CONTINUE;  
 END;  
   
 *-- golesc tabelul imbricat pentru fiecare masina* v\_comenzi := t\_coduri\_comenzi();  
   
 SELECT *NVL*( M.COD\_COMANDA, 0 )  
 BULK COLLECT INTO v\_comenzi  
 FROM C##MYUSER.MASINA M  
 WHERE M.SERIE\_SASIU = v\_serie\_sasiu  
 AND M.COD\_COMANDA NOT LIKE 0;  
  
 v\_index\_by\_table(i) := v\_comenzi.COUNT;  
 DBMS\_OUTPUT.*PUT\_LINE*( 'Masina cu codul ' || p\_masini( i ) || ' are seria de sariu '   
 || v\_serie\_sasiu || ' si apare de ' || v\_index\_by\_table( i ) || ' ori.' );  
   
 END LOOP;  
  
 *-- Acum fiecare masina are asociat numarul sau de vanzari in v\_index\_by\_table  
 -- Voi lua valorile din aceasta colectie si le voi pune intr-un varray* FOR i IN 1..v\_index\_by\_table.COUNT LOOP  
 nr\_vanzari.EXTEND;  
 nr\_vanzari(i) := v\_index\_by\_table(i);  
 END LOOP;  
   
 RETURN nr\_vanzari;  
END;  
  
*-- exemplu de apel:*DECLARE  
 v\_input\_coduri v\_coduri\_masina := v\_coduri\_masina( 32459, 32460, 32461, 32488, 32467, 1 );  
 v\_output\_nr\_vanzari v\_numbers;  
BEGIN  
 v\_output\_nr\_vanzari := *f6*( v\_input\_coduri );  
END;

Rezultate rulare cod:





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

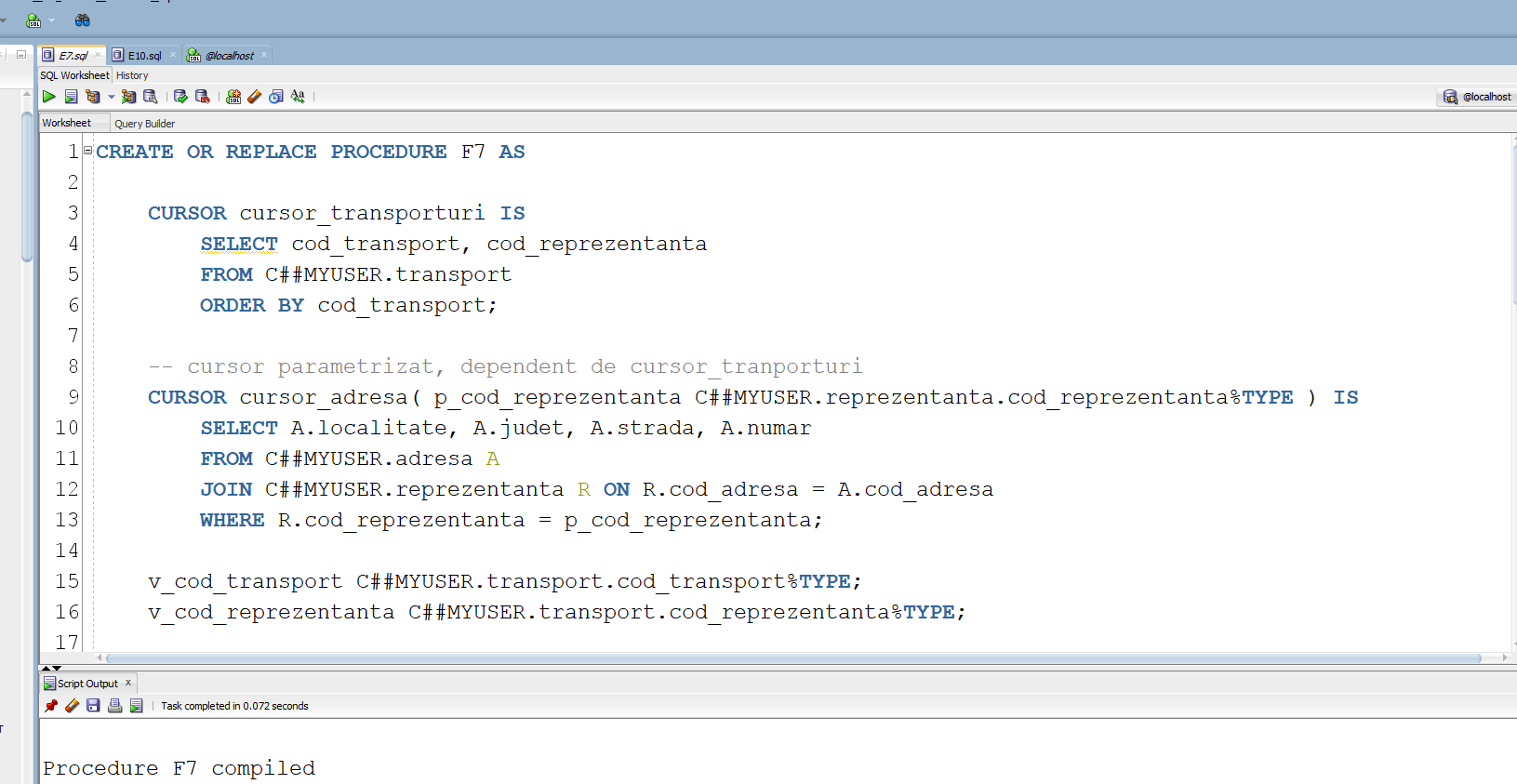
***Cerință***: Pentru toate transporturile din baza de date, să se afiseze adresa unde acestea trebuie să ajungă, adică adresa reprezentanței destinație.

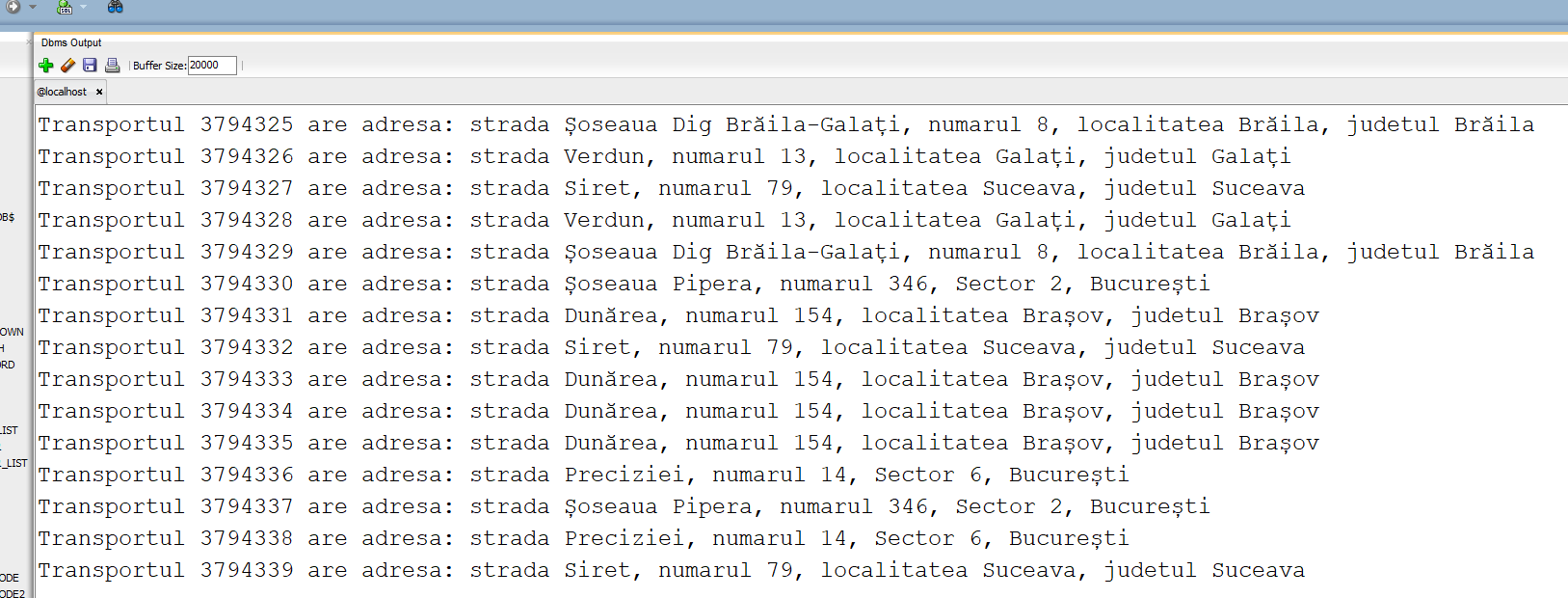
***Rezolvare***: Voi folosi două cursoare, un ciclu cursor care va itera prin toate transporturile din baza de date și un cursor clasic, parametrizat, dependent de celălalt cursor care va selecta adresa reprezentanței unde trebuie să ajungă transportul.

CREATE OR REPLACE PROCEDURE *F7* AS  
  
 CURSOR cursor\_transporturi IS  
 SELECT cod\_transport, cod\_reprezentanta  
 FROM C##MYUSER.transport  
 ORDER BY cod\_transport;  
   
 *-- cursor parametrizat, dependent de cursor\_tranporturi* CURSOR cursor\_adresa( p\_cod\_reprezentanta C##MYUSER.reprezentanta.cod\_reprezentanta%TYPE ) IS  
 SELECT A.localitate, A.judet, A.strada, A.numar  
 FROM C##MYUSER.adresa A  
 JOIN C##MYUSER.reprezentanta R ON R.cod\_adresa = A.cod\_adresa  
 WHERE R.cod\_reprezentanta = p\_cod\_reprezentanta;  
  
 v\_cod\_transport C##MYUSER.transport.cod\_transport%TYPE;  
 v\_cod\_reprezentanta C##MYUSER.transport.cod\_reprezentanta%TYPE;  
   
 v\_localitate C##MYUSER.adresa.localitate%TYPE;  
 v\_judet C##MYUSER.adresa.judet%TYPE;  
 v\_strada C##MYUSER.adresa.strada%TYPE;  
 v\_numar C##MYUSER.adresa.numar%TYPE;  
   
BEGIN  
  
 *-- Ciclu cursor pentru a parcurge toate transporturile* FOR transport IN cursor\_transporturi LOOP  
   
 v\_cod\_transport := transport.cod\_transport;  
 v\_cod\_reprezentanta := transport.cod\_reprezentanta;  
  
 *-- Cursor parametrizat clasic, dependent de celalalt pentru a determina adresa unui transport* OPEN cursor\_adresa( v\_cod\_reprezentanta );  
 FETCH cursor\_adresa INTO v\_localitate, v\_judet, v\_strada, v\_numar;  
  
 IF cursor\_adresa%FOUND THEN  
 IF *INSTR*( v\_localitate, 'Sector' ) = 0 THEN  
 DBMS\_OUTPUT.*PUT\_LINE*( 'Transportul ' || v\_cod\_transport || ' are adresa: strada ' || v\_strada   
 || ', numarul ' || v\_numar || ', localitatea ' || v\_localitate || ', judetul ' || v\_judet );  
 ELSE  
 DBMS\_OUTPUT.*PUT\_LINE*( 'Transportul ' || v\_cod\_transport || ' are adresa: strada ' || v\_strada   
 || ', numarul ' || v\_numar || ', ' || v\_localitate || ', ' || v\_judet );  
 END IF;  
 ELSE  
 DBMS\_OUTPUT.*PUT\_LINE*( 'Adresa transportului ' || v\_cod\_transport || ' nu a fost gasita!' );  
 END IF;  
   
 CLOSE cursor\_adresa;  
 END LOOP;  
END;  
/

BEGIN  
 *F7*();  
END;  
/

Rezultate rulare cod:





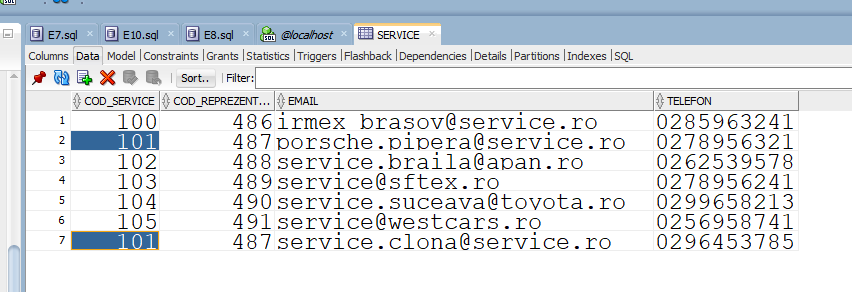
# 8) Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele create. Tratați toate excepțiile care pot apărea, incluzând excepțiile predefinite NO\_DATA\_FOUND și TOO\_MANY\_ROWS.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

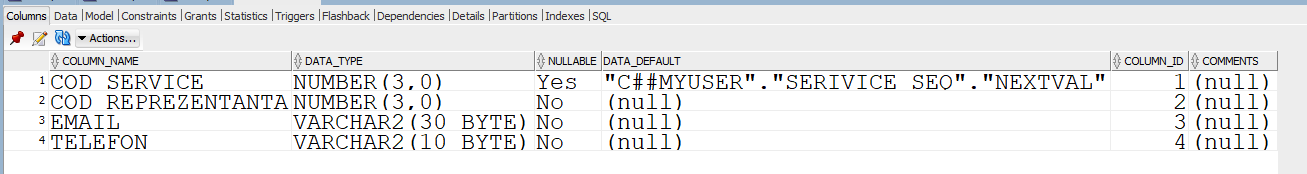
***Cerință***: Pentru un service dat ca parametru sub formă de cod, să se determine adresa sa în format VARCHAR2( 250 ), folosind un subprogram stocat independent de tip funcție.

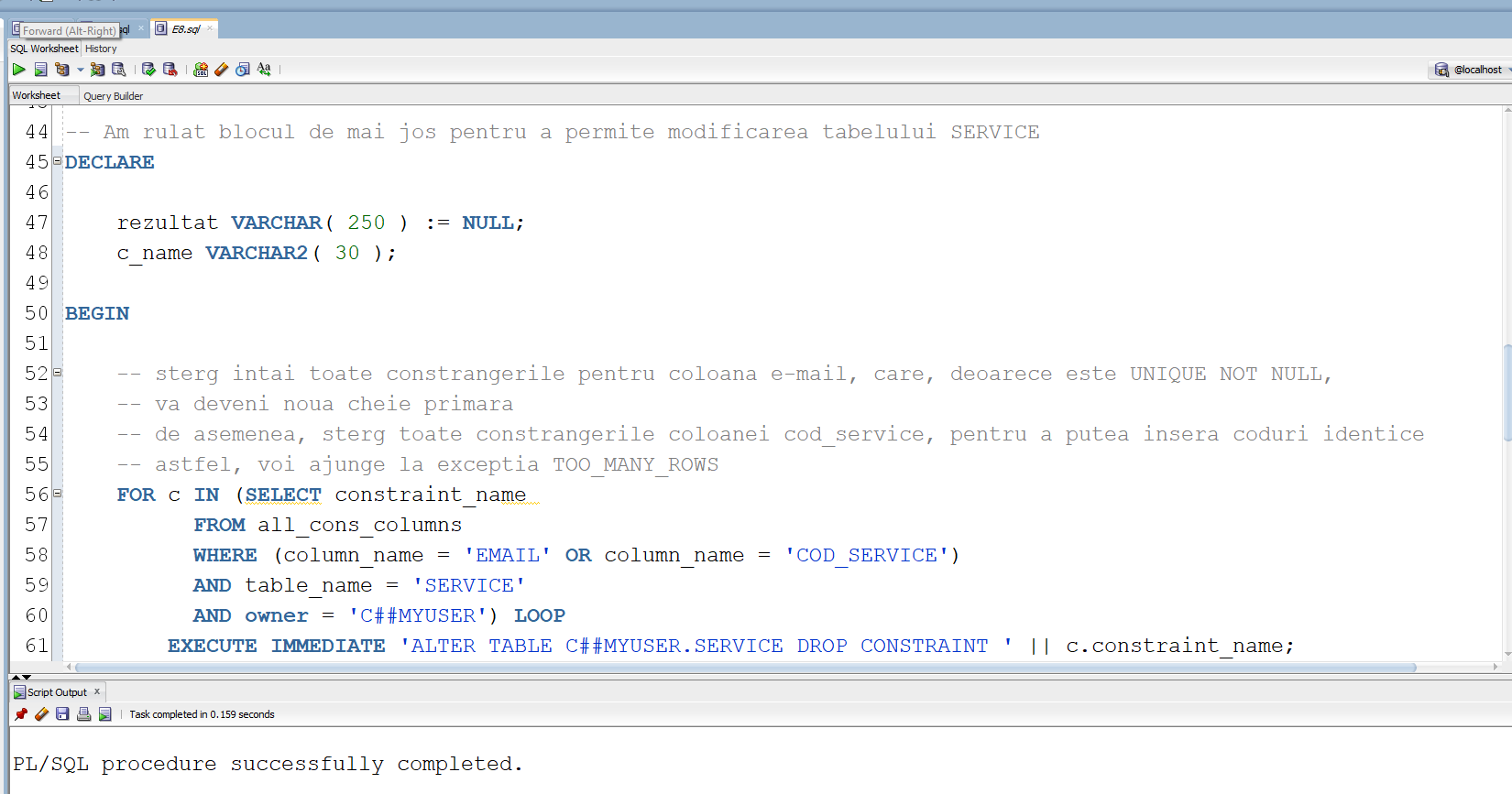
***Rezolvare***: Pentru excepția *TOO\_MANY\_ROWS*, am inserat în tabelul *SERVICE*, încă un service cu codul 101. Am selectat în variabile cu nume sugestiv detaliile despre adresa service-ului, apoi le-am afișat. Structura tabelului a fost modificata pentru acest exercițiu. Pentru a putea modifica structura, am rulat întâi:

BEGIN  
 *-- sterg intai toate constrangerile pentru coloana e-mail, care, deoarece este UNIQUE NOT NULL,   
 -- va deveni noua cheie primara  
 -- de asemenea, sterg toate constrangerile coloanei cod\_service, pentru a putea insera coduri identice  
 -- astfel, voi ajunge la exceptia TOO\_MANY\_ROWS* FOR c IN (SELECT constraint\_name   
 FROM all\_cons\_columns   
 WHERE (column\_name = 'EMAIL' OR column\_name = 'COD\_SERVICE')   
 AND table\_name = 'SERVICE'   
 AND owner = 'C##MYUSER') LOOP  
 EXECUTE IMMEDIATE 'ALTER TABLE C##MYUSER.SERVICE DROP CONSTRAINT ' || c.constraint\_name;  
 END LOOP;  
  
*-- execute immediate permite efectuarea comenzilor LDD in PL/SQL prin SQL dinamic ( se ruleaza la runtime )* EXECUTE IMMEDIATE 'ALTER TABLE C##MYUSER.SERVICE ADD CONSTRAINT email\_pk PRIMARY KEY (email)';  
   
 *-- insert into nu are nevoie de SQL dinamic* INSERT INTO C##MYUSER.SERVICE ( cod\_service, cod\_reprezentanta, email, telefon )  
 VALUES( 101, 487, 'service.clona@service.ro', '0296453785' );  
  
END;  
/

Astfel, au fost șterse constrângerile pentru coloana *email*, care va deveni noua cheie primară și cele pentru coloana cod\_service care era cheia primară. Apoi, am putut insera un service cu un cod deja prezent în baza de date, iar acum există două service-uri cu același cod. Modificările permit un rezultat de tip TOO\_MANY\_ROWS. Rezultatele sunt:





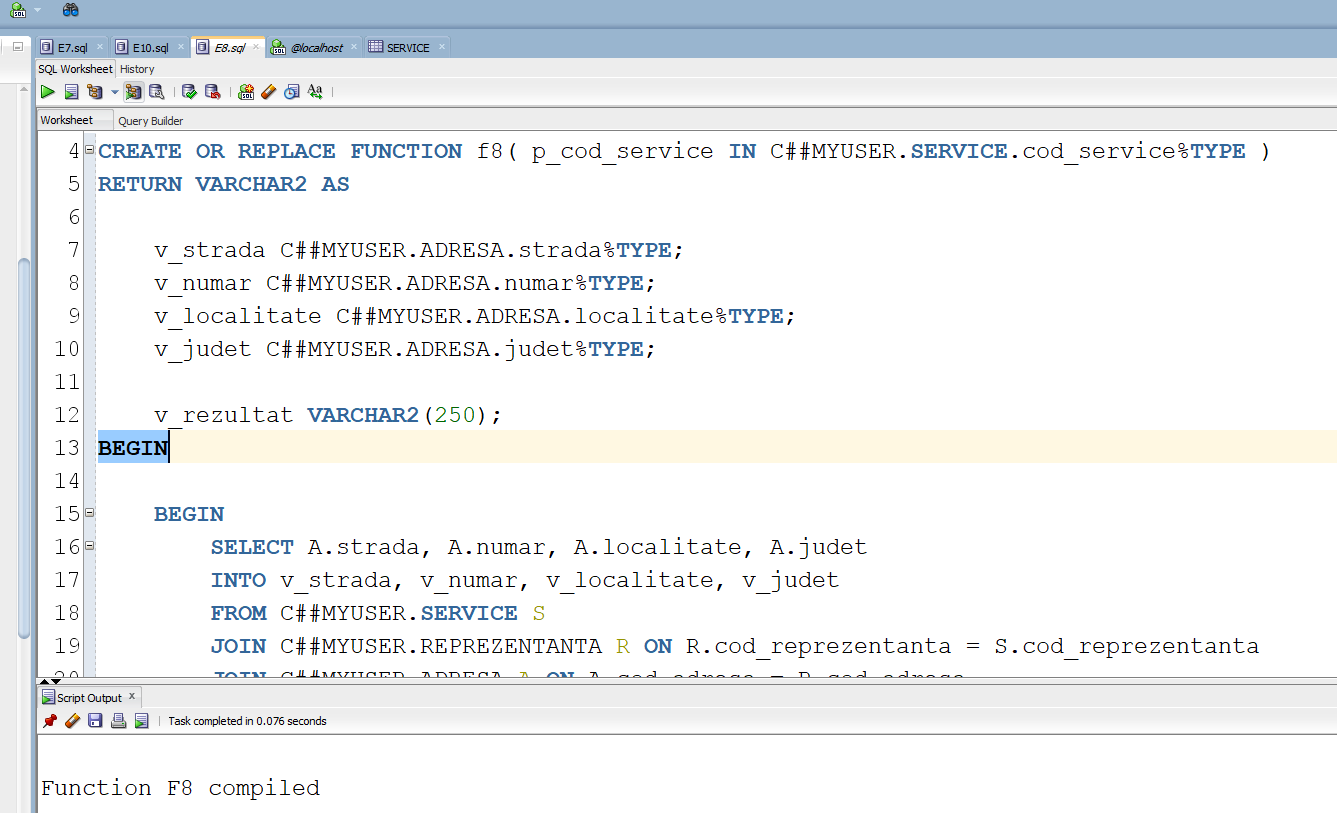


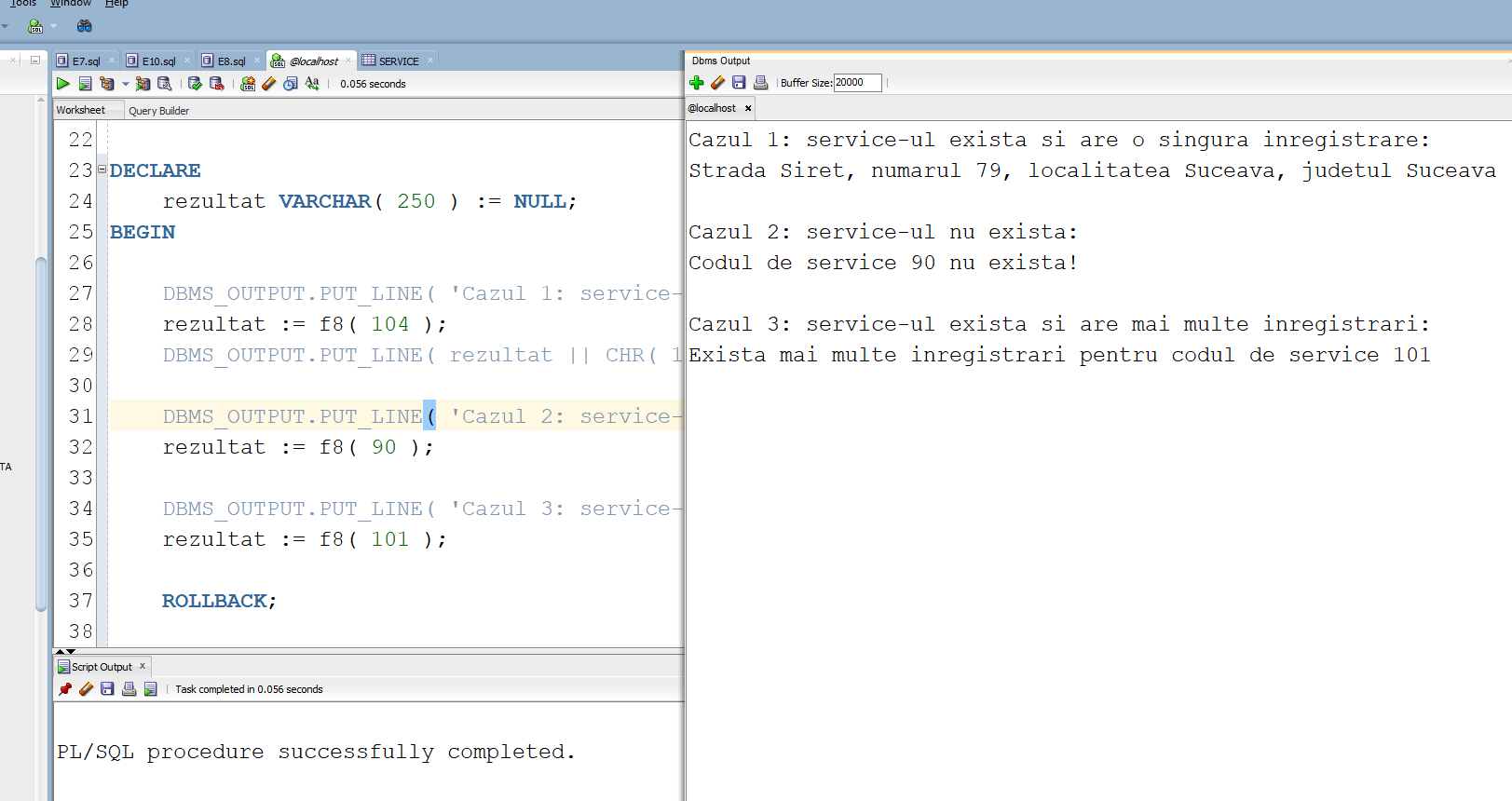
Cod funcție:

CREATE OR REPLACE FUNCTION *f8*( p\_cod\_service IN C##MYUSER.SERVICE.cod\_service%TYPE )  
RETURN VARCHAR2 AS  
  
 v\_strada C##MYUSER.ADRESA.strada%TYPE;  
 v\_numar C##MYUSER.ADRESA.numar%TYPE;  
 v\_localitate C##MYUSER.ADRESA.localitate%TYPE;  
 v\_judet C##MYUSER.ADRESA.judet%TYPE;  
   
 v\_rezultat VARCHAR2(250);  
BEGIN  
  
 BEGIN  
 SELECT A.strada, A.numar, A.localitate, A.judet  
 INTO v\_strada, v\_numar, v\_localitate, v\_judet  
 FROM C##MYUSER.SERVICE S  
 JOIN C##MYUSER.REPREZENTANTA R ON R.cod\_reprezentanta = S.cod\_reprezentanta  
 JOIN C##MYUSER.ADRESA A ON A.cod\_adresa = R.cod\_adresa  
 WHERE S.cod\_service = p\_cod\_service;  
  
 IF *INSTR*( v\_localitate, 'Sector' ) = 0 THEN  
 v\_rezultat := 'Strada ' || v\_strada || ', numarul ' || v\_numar || ', localitatea '   
 || v\_localitate || ', judetul ' || v\_judet;  
 ELSE  
 v\_rezultat := 'Strada ' || v\_strada || ', numarul ' || v\_numar || ', '   
 || v\_localitate || ', ' || v\_judet;  
 END IF;  
   
   
 EXCEPTION  
 WHEN NO\_DATA\_FOUND THEN  
 DBMS\_OUTPUT.*PUT\_LINE*( 'Codul de service ' || p\_cod\_service || ' nu exista!' || *CHR*(10) );  
 RETURN NULL;  
 WHEN TOO\_MANY\_ROWS THEN  
 DBMS\_OUTPUT.*PUT\_LINE*( 'Exista mai multe inregistrari pentru codul de service '   
 || p\_cod\_service || *CHR*(10) );  
 RETURN NULL;  
 WHEN OTHERS THEN  
 DBMS\_OUTPUT.*PUT\_LINE*( 'Alta eroare!' || *CHR*(10) );  
 RETURN NULL;  
 END;  
   
 RETURN v\_rezultat;  
   
END;

DECLARE  
 rezultat VARCHAR( 250 ) := NULL;  
BEGIN  
   
 DBMS\_OUTPUT.*PUT\_LINE*( 'Cazul 1: service-ul exista si are o singura inregistrare:' );  
 rezultat := *f8*( 104 );  
 DBMS\_OUTPUT.*PUT\_LINE*( rezultat || *CHR*( 10 ) );  
   
 DBMS\_OUTPUT.*PUT\_LINE*( 'Cazul 2: service-ul nu exista:' );  
 rezultat := *f8*( 90 );  
   
 DBMS\_OUTPUT.*PUT\_LINE*( 'Cazul 3: service-ul exista si are mai multe inregistrari:' );  
 rezultat := *f8*( 101 );  
   
 ROLLBACK;  
   
END;  
/

Rezultate rulare cod funcție:

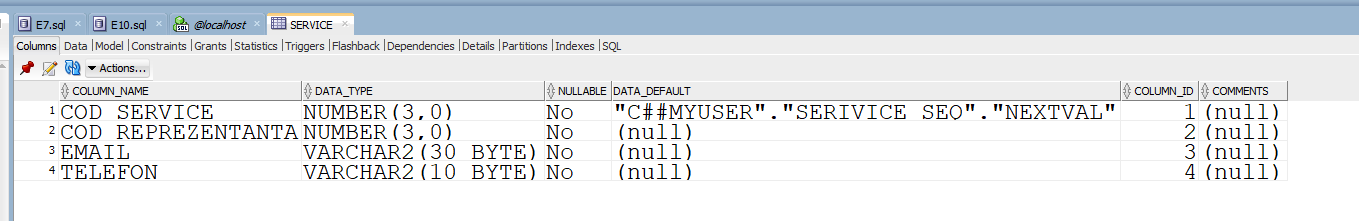




Codul pentru restaurarea structurii tabelului *SERVICE* este*:*

*-- bloc pentru restaurarea structurii tabelului SERVICE este:*DECLARE  
 c\_name VARCHAR2( 30 );  
 rezultat VARCHAR( 250 ) := NULL;  
BEGIN  
   
 DBMS\_OUTPUT.*PUT\_LINE*( 'Cazul 1: service-ul exista si are o singura inregistrare:' );  
 rezultat := *f8*( 104 );  
 DBMS\_OUTPUT.*PUT\_LINE*( rezultat || *CHR*(10) );  
   
 DBMS\_OUTPUT.*PUT\_LINE*( 'Cazul 2: service-ul nu exista:' );  
 rezultat := *f8*( 90 );  
   
 DBMS\_OUTPUT.*PUT\_LINE*( 'Cazul 3: service-ul exista si are mai multe inregistrari:' );  
 rezultat := *f8*( 101 );  
   
 *-- acum voi sterge din tabel service-ul inserat anterior* DELETE FROM C##MYUSER.SERVICE S  
 WHERE S.email LIKE 'service.clona@service.ro';  
   
 *-- Drop primary key constraint* SELECT constraint\_name INTO c\_name  
 FROM all\_constraints  
 WHERE table\_name = 'SERVICE' AND owner = 'C##MYUSER' AND constraint\_type = 'P';  
 EXECUTE IMMEDIATE 'ALTER TABLE C##MYUSER.SERVICE DROP CONSTRAINT ' || c\_name;  
   
 *-- Definesc din nou constrangerile initiale pentru tabela SERVICE* EXECUTE IMMEDIATE 'ALTER TABLE C##MYUSER.SERVICE MODIFY (email VARCHAR2(30) UNIQUE NOT NULL)';  
 EXECUTE IMMEDIATE 'ALTER TABLE C##MYUSER.SERVICE ADD CONSTRAINT service\_cod\_service\_pk PRIMARY KEY (cod\_service)';  
   
END;  
/

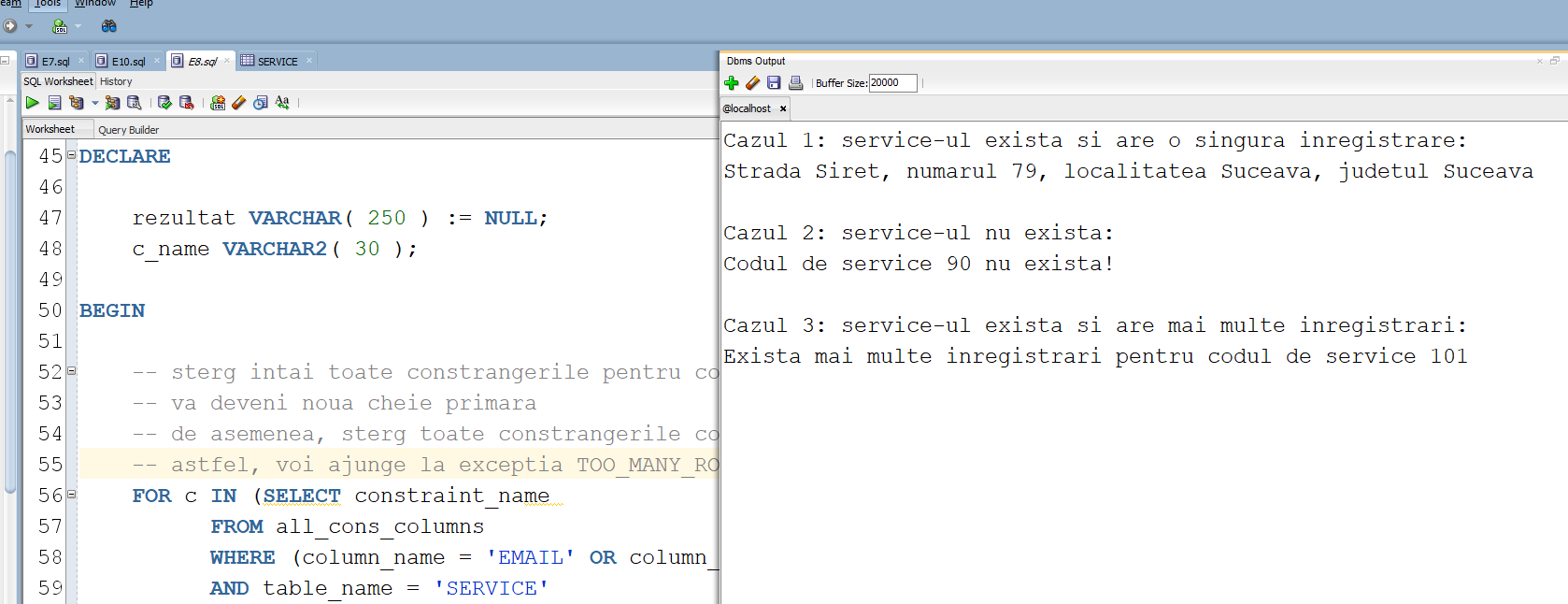
Astfel, tabelul își recapătă structura inițială:

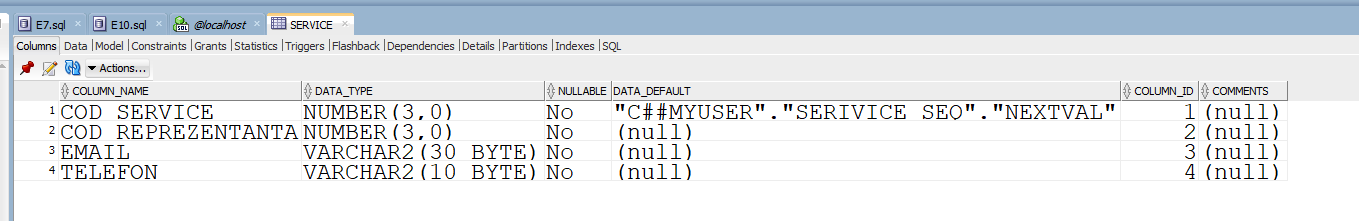


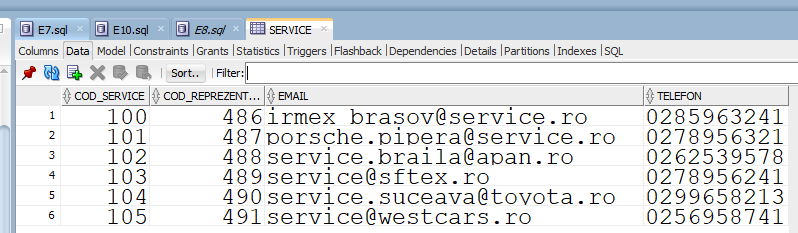
Un bloc care realizează apelul complet al funcției, fără a scinda transformările la nivel de tabel este:

*-- Am rulat blocul de mai jos pentru a permite modificarea tabelului SERVICE*DECLARE  
  
 rezultat VARCHAR( 250 ) := NULL;  
 c\_name VARCHAR2( 30 );  
   
BEGIN  
  
 *-- sterg intai toate constrangerile pentru coloana e-mail, care, deoarece este UNIQUE NOT NULL,   
 -- va deveni noua cheie primara  
 -- de asemenea, sterg toate constrangerile coloanei cod\_service, pentru a putea insera coduri identice  
 -- astfel, voi ajunge la exceptia TOO\_MANY\_ROWS* FOR c IN (SELECT constraint\_name   
 FROM all\_cons\_columns   
 WHERE (column\_name = 'EMAIL' OR column\_name = 'COD\_SERVICE')   
 AND table\_name = 'SERVICE'   
 AND owner = 'C##MYUSER') LOOP  
 EXECUTE IMMEDIATE 'ALTER TABLE C##MYUSER.SERVICE DROP CONSTRAINT ' || c.constraint\_name;  
 END LOOP;  
  
 *-- execute immediate permite efectuarea comenzilor LDD in PL/SQL prin SQL dinamic ( se ruleaza la runtime )* EXECUTE IMMEDIATE 'ALTER TABLE C##MYUSER.SERVICE ADD CONSTRAINT email\_pk PRIMARY KEY (email)';  
   
 *-- insert into nu are nevoie de SQL dinamic* INSERT INTO C##MYUSER.SERVICE ( cod\_service, cod\_reprezentanta, email, telefon )  
 VALUES( 101, 487, 'service.clona@service.ro', '0296453785' );  
  
 DBMS\_OUTPUT.*PUT\_LINE*( 'Cazul 1: service-ul exista si are o singura inregistrare:' );  
 rezultat := *f8*( 104 );  
 DBMS\_OUTPUT.*PUT\_LINE*( rezultat || *CHR*(10) );  
   
 DBMS\_OUTPUT.*PUT\_LINE*( 'Cazul 2: service-ul nu exista:' );  
 rezultat := *f8*( 90 );  
   
 DBMS\_OUTPUT.*PUT\_LINE*( 'Cazul 3: service-ul exista si are mai multe inregistrari:' );  
 rezultat := *f8*( 101 );  
   
 *-- acum voi sterge din tabel service-ul inserat anterior* DELETE FROM C##MYUSER.SERVICE S  
 WHERE S.email LIKE 'service.clona@service.ro';  
   
 *-- Drop primary key constraint* SELECT constraint\_name INTO c\_name  
 FROM all\_constraints  
 WHERE table\_name = 'SERVICE' AND owner = 'C##MYUSER' AND constraint\_type = 'P';  
 EXECUTE IMMEDIATE 'ALTER TABLE C##MYUSER.SERVICE DROP CONSTRAINT ' || c\_name;  
   
 *-- Definesc din nou constrangerile initiale pentru tabela SERVICE* EXECUTE IMMEDIATE 'ALTER TABLE C##MYUSER.SERVICE MODIFY (email VARCHAR2(30) UNIQUE NOT NULL)';  
 EXECUTE IMMEDIATE 'ALTER TABLE C##MYUSER.SERVICE ADD CONSTRAINT service\_cod\_service\_pk PRIMARY KEY (cod\_service)';  
   
END;  
/

Rularea lui duce la același rezultat:







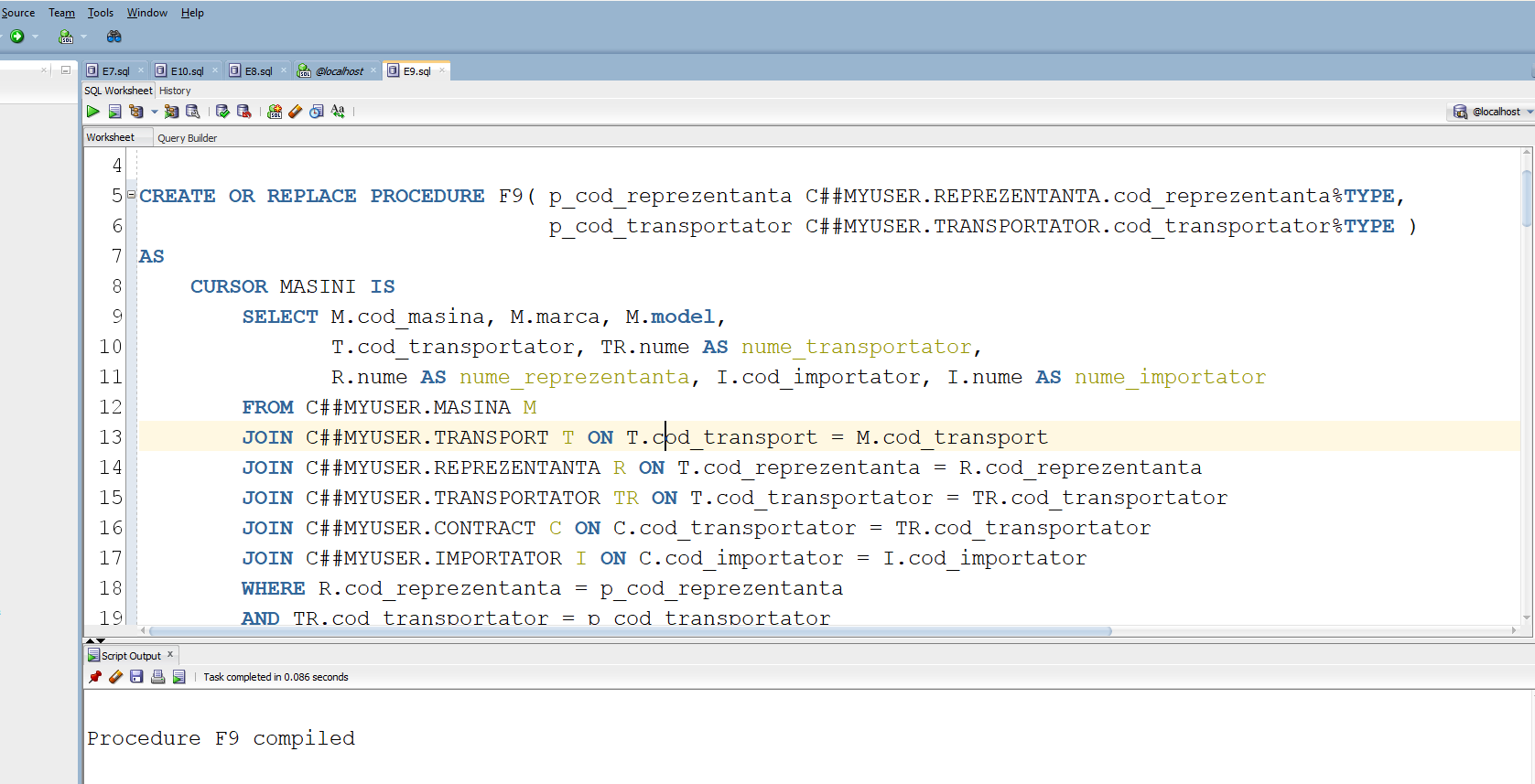
# 9) Formulați în limbaj natural o problemă pe care să o rezolvați folosind un **subprogram stocat independent de tip procedură** care să aibă minim 2 parametri și să utilizeze **într-o singură comandă SQL** 5 dintre tabelele create. Definiți minim 2 excepții proprii, altele decât cele predefinite la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

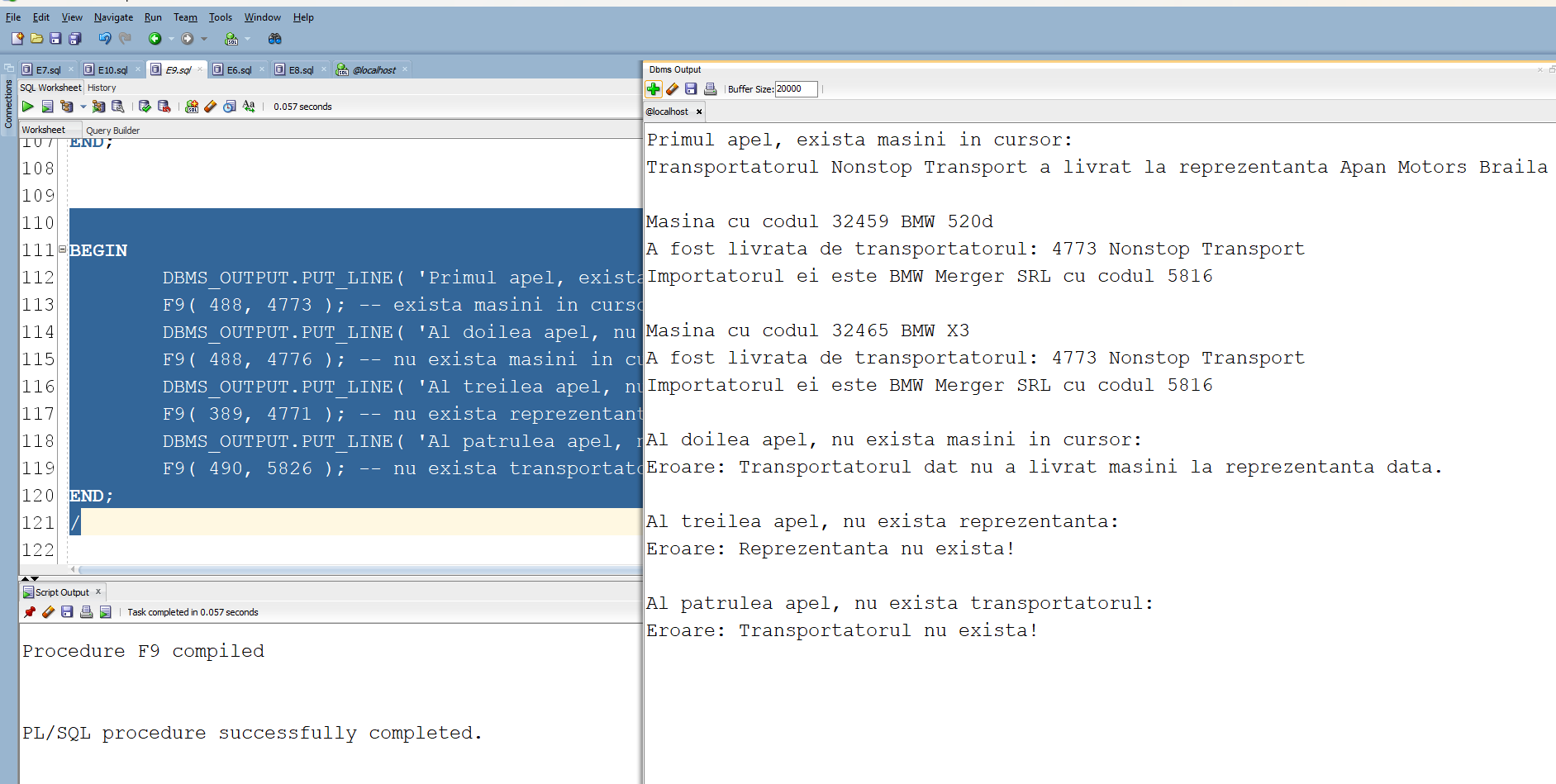
***Cerință***: Se dau un transportator și o reprezentanță ca parametrii, sub formă de coduri. Să se afișeze pentru fiecare mașina livrată de acel transportator la acea reprezentanță importatorul care a adus-o în țară.

***Rezolvare***: Am folosit un cursor prin care am aflat toate informațiile necesare pentru mașini, transportatori, reprezentanțe și importatori. O mașină este adusă în țară de un importator dacă cele două entități au aceeași valoare în coloana *marcă.* După ce am verificat dacă reprezentanța și transportatorul există, am deschis cursorul și am verificat dacă există date ( mașini ) în acesta. În caz afirmativ, l-am închis și l-am deschis din nou pentru a-l parcurge.

*--9) Se dau un transportator si o reprezentanta. Sa se afiseze pentru fiecare masina livrata de acel   
--transportator la acea reprezentanta importatorul care a adus-o in tara.*CREATE OR REPLACE PROCEDURE *F9*( p\_cod\_reprezentanta C##MYUSER.REPREZENTANTA.cod\_reprezentanta%TYPE,  
 p\_cod\_transportator C##MYUSER.TRANSPORTATOR.cod\_transportator%TYPE )  
AS  
 CURSOR MASINI IS  
 SELECT M.cod\_masina, M.marca, M.model,  
 T.cod\_transportator, TR.nume AS nume\_transportator,   
 R.nume AS nume\_reprezentanta, I.cod\_importator, I.nume AS nume\_importator  
 FROM C##MYUSER.MASINA M  
 JOIN C##MYUSER.TRANSPORT T ON T.cod\_transport = M.cod\_transport  
 JOIN C##MYUSER.REPREZENTANTA R ON T.cod\_reprezentanta = R.cod\_reprezentanta  
 JOIN C##MYUSER.TRANSPORTATOR TR ON T.cod\_transportator = TR.cod\_transportator  
 JOIN C##MYUSER.CONTRACT C ON C.cod\_transportator = TR.cod\_transportator  
 JOIN C##MYUSER.IMPORTATOR I ON C.cod\_importator = I.cod\_importator  
 WHERE R.cod\_reprezentanta = p\_cod\_reprezentanta  
 AND TR.cod\_transportator = p\_cod\_transportator  
 AND *UPPER*( *TRIM*( M.marca ) ) = *UPPER*( *TRIM*( I.marca ) );  
  
  
 *-- Variabile pentru a verifica daca reprezentanta si transportatorul exista* v\_nume\_reprezentanta VARCHAR2(50);  
 v\_nume\_transportator VARCHAR2(50);  
  
  
 *-- variabile pentru a verifica daca cursorul gaseste date, voi folosi prefixul ts pentru test* ts\_cod\_masina C##MYUSER.MASINA.cod\_masina%TYPE;  
 ts\_marca C##MYUSER.MASINA.marca%TYPE;  
 ts\_model C##MYUSER.MASINA.model%TYPE;  
 ts\_cod\_transportator C##MYUSER.TRANSPORTATOR.cod\_transportator%TYPE;  
 ts\_nume\_transportator C##MYUSER.TRANSPORTATOR.nume%TYPE;  
 ts\_nume\_reprezentanta C##MYUSER.REPREZENTANTA.nume%TYPE;  
 ts\_cod\_importator C##MYUSER.IMPORTATOR.cod\_importator%TYPE;  
 ts\_nume\_importator C##MYUSER.IMPORTATOR.nume%TYPE;  
  
  
 *-- Exceptii* e\_nu\_exista\_reprezentanta EXCEPTION;  
 e\_nu\_exista\_transportator EXCEPTION;  
 e\_nu\_exista\_masini EXCEPTION;  
   
BEGIN  
  
 BEGIN  
   
 SELECT R.nume  
 INTO v\_nume\_reprezentanta  
 FROM C##MYUSER.REPREZENTANTA R  
 WHERE R.cod\_reprezentanta = p\_cod\_reprezentanta;  
 EXCEPTION  
 WHEN NO\_DATA\_FOUND THEN  
 RAISE e\_nu\_exista\_reprezentanta;  
 END;  
  
  
 BEGIN  
   
 SELECT T.nume  
 INTO v\_nume\_transportator  
 FROM C##MYUSER.TRANSPORTATOR T  
 WHERE T.cod\_transportator = p\_cod\_transportator;  
 EXCEPTION  
 WHEN NO\_DATA\_FOUND THEN  
 RAISE e\_nu\_exista\_transportator;  
 END;  
  
  
 OPEN MASINI;  
 LOOP  
 FETCH MASINI INTO ts\_cod\_masina, ts\_marca, ts\_model,  
 ts\_cod\_transportator, ts\_nume\_transportator,   
 ts\_nume\_reprezentanta, ts\_cod\_importator, ts\_nume\_importator;  
 IF MASINI%FOUND THEN  
 DBMS\_OUTPUT.*PUT\_LINE*('Transportatorul ' || v\_nume\_transportator   
 || ' a livrat la reprezentanta ' || v\_nume\_reprezentanta || *CHR*( 10 ) );  
   
 CLOSE MASINI;  
   
 FOR v\_masina IN MASINI LOOP  
 DBMS\_OUTPUT.*PUT\_LINE*( 'Masina cu codul ' || v\_masina.cod\_masina || ' ' || v\_masina.marca || ' ' || v\_masina.model);  
 DBMS\_OUTPUT.*PUT\_LINE*( 'A fost livrata de transportatorul: ' || v\_masina.cod\_transportator || ' ' || v\_masina.nume\_transportator);  
 DBMS\_OUTPUT.*PUT\_LINE*( 'Importatorul ei este ' || v\_masina.nume\_importator || ' cu codul ' || v\_masina.cod\_importator || *CHR*( 10 ) );  
 END LOOP;  
   
 EXIT;  
   
 ELSE   
 RAISE e\_nu\_exista\_masini;  
 CLOSE MASINI;  
 EXIT;  
 END IF;  
   
 END LOOP;  
  
  
EXCEPTION  
 WHEN e\_nu\_exista\_reprezentanta THEN  
 DBMS\_OUTPUT.*PUT\_LINE*( 'Eroare: Reprezentanta nu exista!' || *CHR*( 10 ) );  
 WHEN e\_nu\_exista\_transportator THEN  
 DBMS\_OUTPUT.*PUT\_LINE*( 'Eroare: Transportatorul nu exista!' || *CHR*( 10 ) );  
 WHEN e\_nu\_exista\_masini THEN  
 DBMS\_OUTPUT.*PUT\_LINE*( 'Eroare: Transportatorul dat nu a livrat masini la reprezentanta data.' || *CHR*( 10 ) );  
 WHEN OTHERS THEN  
 DBMS\_OUTPUT.*PUT\_LINE*( 'Alta eroare!' || *CHR*( 10 ) );  
END;  
  
  
  
BEGIN  
 DBMS\_OUTPUT.*PUT\_LINE*( 'Primul apel, exista masini in cursor:');  
 *F9*( 488, 4773 ); *-- exista masini in cursor* DBMS\_OUTPUT.*PUT\_LINE*( 'Al doilea apel, nu exista masini in cursor:');  
 *F9*( 488, 4776 ); *-- nu exista masini in cursor* DBMS\_OUTPUT.*PUT\_LINE*( 'Al treilea apel, nu exista reprezentanta:');  
 *F9*( 389, 4771 ); *-- nu exista reprezentanta* DBMS\_OUTPUT.*PUT\_LINE*( 'Al patrulea apel, nu exista transportatorul:');  
 *F9*( 490, 5826 ); *-- nu exista transportatorul*END;  
/

Rezultate rulare cod:



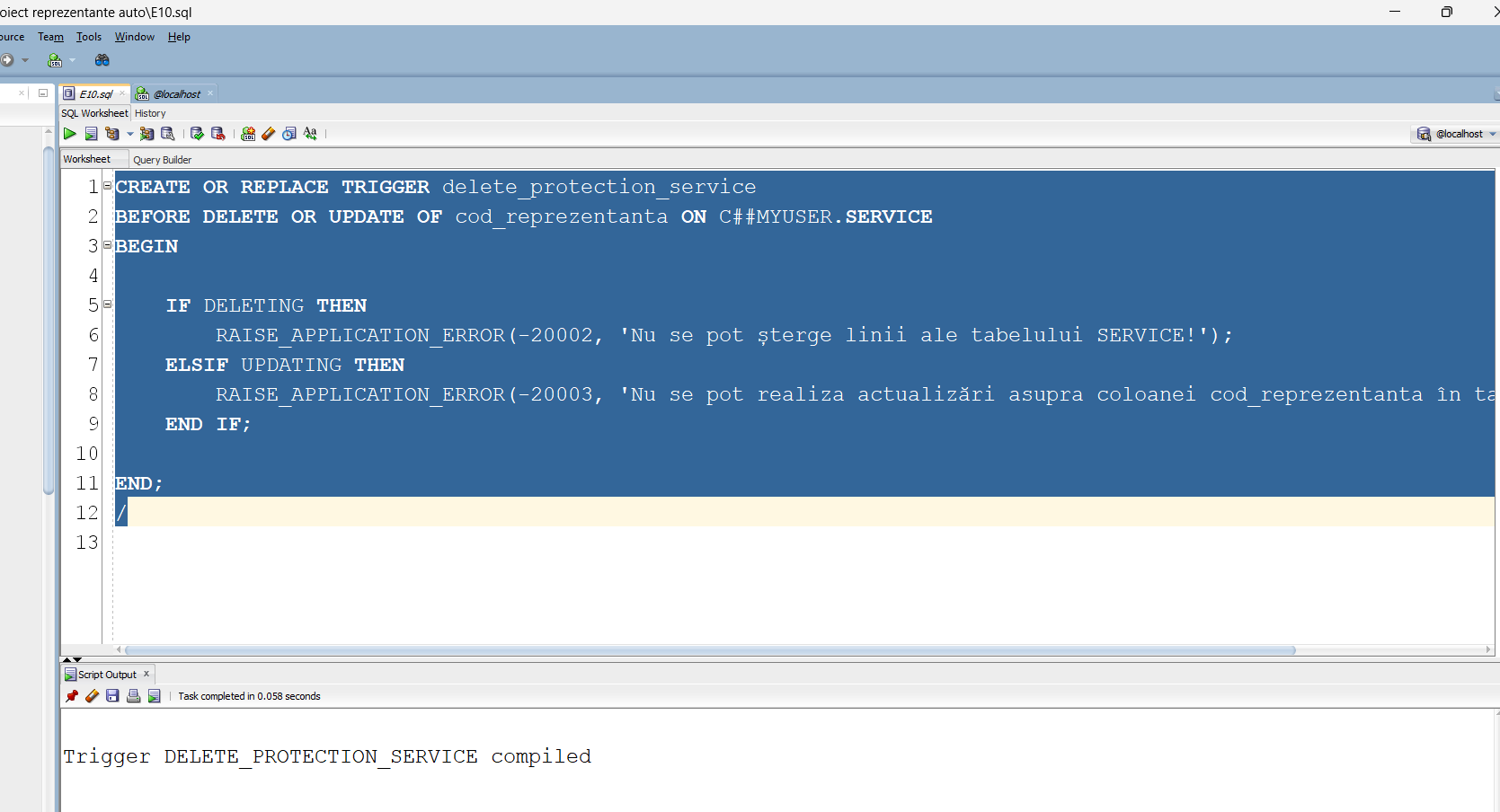


# 10) Definiți un *trigger* de tip LMD la nivel de comandă. Declanșați *trigger*-ul.

Voi defini un trigger care nu permite efectuarea operatiilor de delete și update asupra coloanei cod\_reprezentanta a tabelului *SERVICE*.

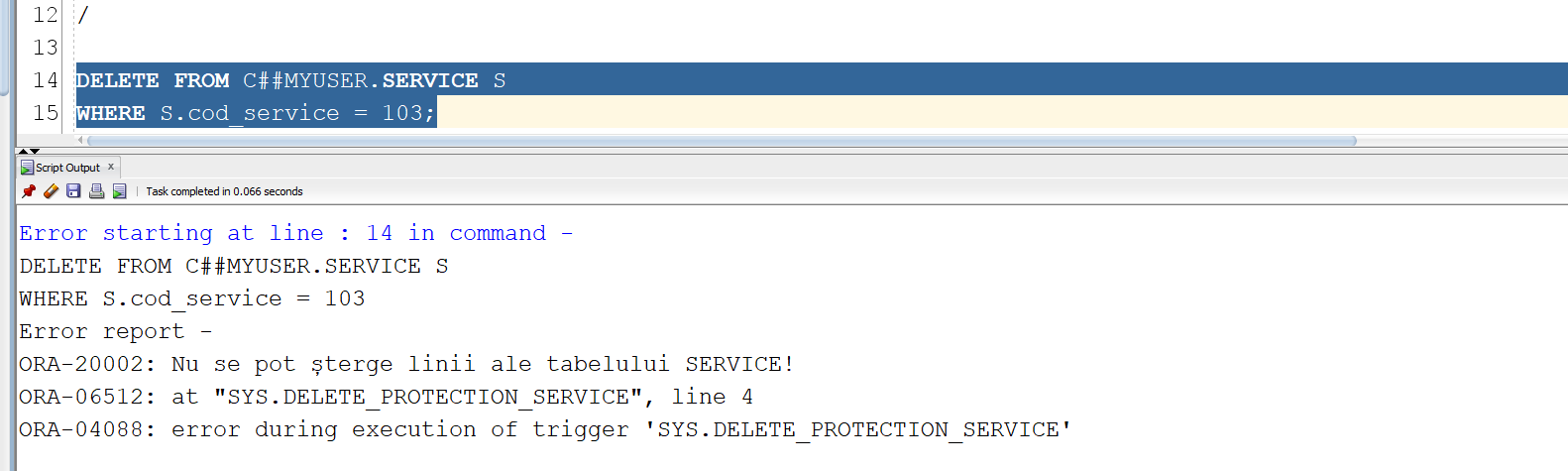
CREATE OR REPLACE TRIGGER delete\_protection  
BEFORE DELETE OR UPDATE OF cod\_reprezentanta ON C##MYUSER.SERVICE  
BEGIN  
  
 IF DELETING THEN  
 *RAISE\_APPLICATION\_ERROR*(-20002, 'Nu se pot șterge linii ale tabelului SERVICE!');  
 ELSIF UPDATING THEN  
 *RAISE\_APPLICATION\_ERROR*(-20003, 'Nu se pot realiza actualizări asupra coloanei cod\_reprezentanta în tabelul SERVICE!');  
 END IF;  
  
END;  
/

Rezultate rulare cod:



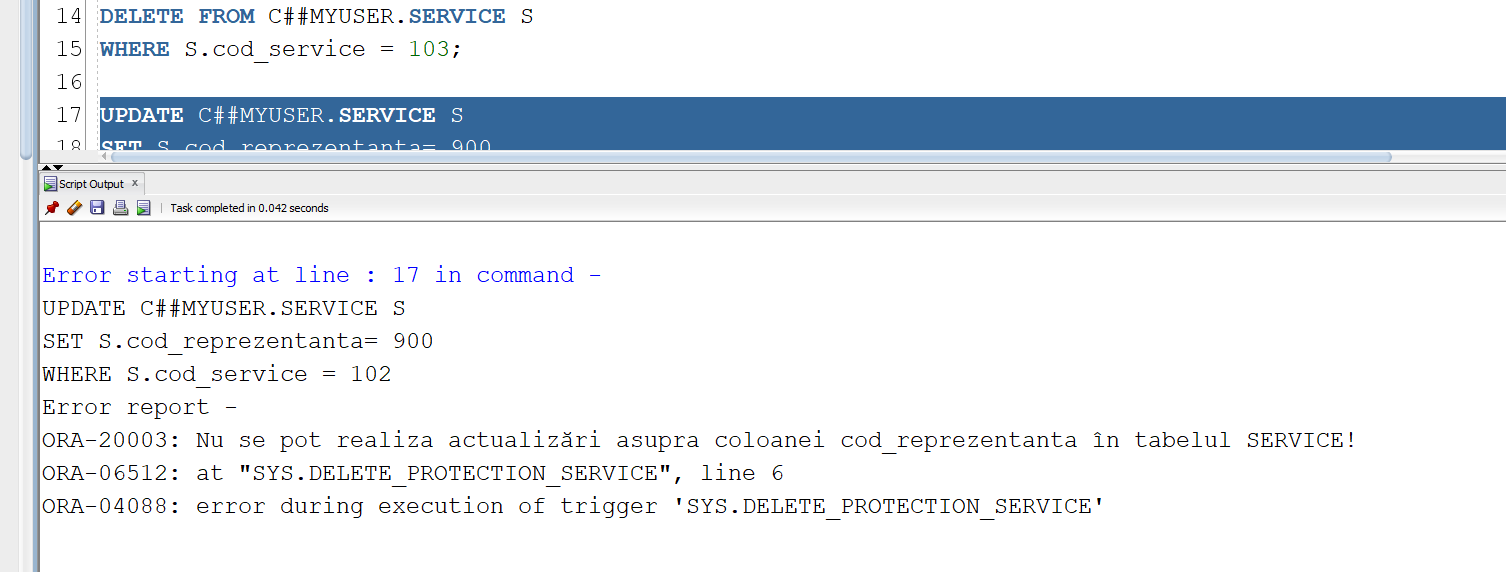
DELETE FROM C##MYUSER.SERVICE S  
WHERE S.cod\_service = 103;

Rezultat ștergere:



UPDATE C##MYUSER.SERVICE S  
SET S.cod\_reprezentanta = 900  
WHERE S.cod\_service = 102;

Rezultat modificare:

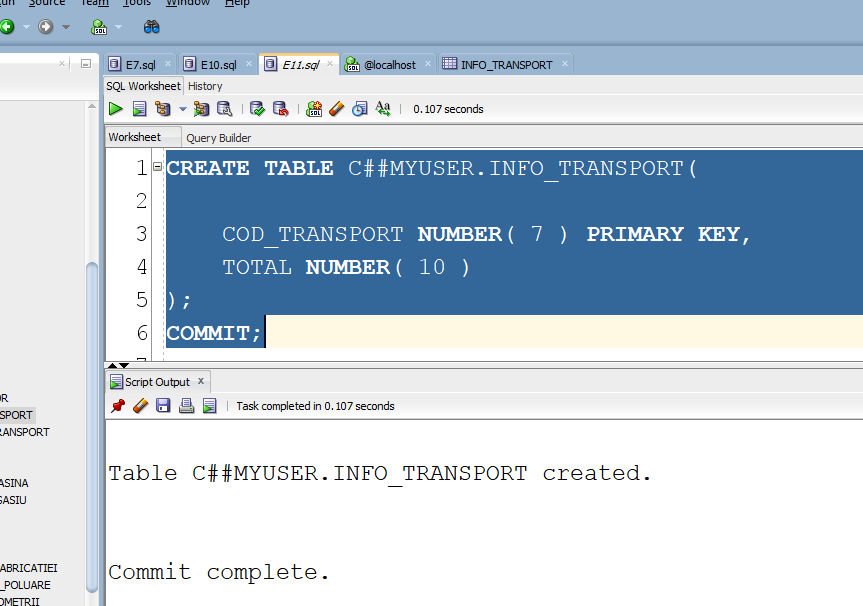


# 11) Definiți un *trigger* de tip LMD la nivel de linie. Declanșați *trigger*-ul.

***Rezolvare***: Voi crea un tabel *INFO\_TRANSPORT,* în care voi calcula valoarea unui transport. Triggerul se va declanșa pentru a calcula valorile din acest tabel.

CREATE TABLE C##MYUSER.INFO\_TRANSPORT(  
  
 COD\_TRANSPORT NUMBER( 7 ) PRIMARY KEY,  
 TOTAL NUMBER( 10 )  
);  
COMMIT;

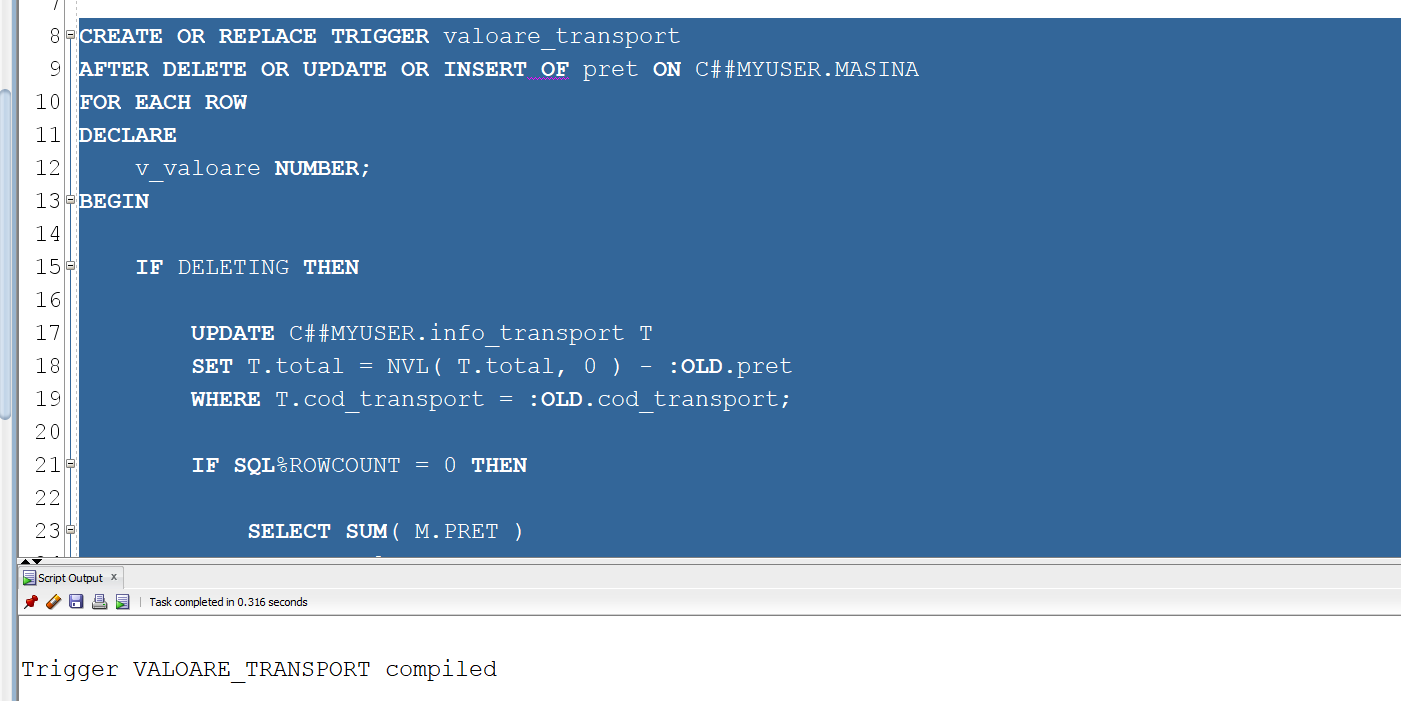
Rezultate creare tabel:



Cod trigger:

CREATE OR REPLACE TRIGGER valoare\_transport  
AFTER DELETE OR UPDATE OR INSERT OF pret ON C##MYUSER.MASINA  
FOR EACH ROW  
DECLARE   
 v\_valoare NUMBER;  
BEGIN  
  
 IF DELETING THEN  
  
 UPDATE C##MYUSER.info\_transport T  
 SET T.total = *NVL*( T.total, 0 ) - :OLD.pret  
 WHERE T.cod\_transport = :OLD.cod\_transport;  
   
 IF SQL%ROWCOUNT = 0 THEN  
   
 SELECT *SUM*( M.PRET )  
 INTO v\_valoare  
 FROM C##MYUSER.MASINA M  
 WHERE M.COD\_TRANSPORT = :OLD.**cod\_transport**;  
  
 INSERT INTO C##MYUSER.info\_transport( cod\_transport, total )  
 VALUES( :OLD.cod\_transport, v\_valoare );  
   
 END IF;  
  
 ELSIF UPDATING THEN  
  
 UPDATE C##MYUSER.info\_transport T  
 SET T.total = *NVL*( T.total, 0 ) + :NEW.pret - :OLD.pret  
 WHERE T.cod\_transport = :OLD.cod\_transport;  
   
 IF SQL%ROWCOUNT = 0 THEN  
   
 SELECT *SUM*( M.PRET )  
 INTO v\_valoare  
 FROM C##MYUSER.MASINA M  
 WHERE M.COD\_TRANSPORT = :NEW.**cod\_transport**;  
  
 INSERT INTO C##MYUSER.info\_transport( cod\_transport, total )  
 VALUES( :NEW.cod\_transport, v\_valoare );  
 END IF;  
  
 ELSIF INSERTING THEN  
  
 UPDATE C##MYUSER.info\_transport T  
 SET T.total = *NVL*( T.total, 0 ) + :NEW.pret  
 WHERE T.cod\_transport = :NEW.cod\_transport;  
   
 IF SQL%ROWCOUNT = 0 THEN  
   
 SELECT *SUM*( M.PRET )  
 INTO v\_valoare  
 FROM C##MYUSER.MASINA M  
 WHERE M.COD\_TRANSPORT = :NEW.**cod\_transport**;  
   
 INSERT INTO C##MYUSER.info\_transport( cod\_transport, total )  
 VALUES( :NEW.cod\_transport, v\_valoare );  
 END IF;  
   
 END IF;  
   
END;  
/

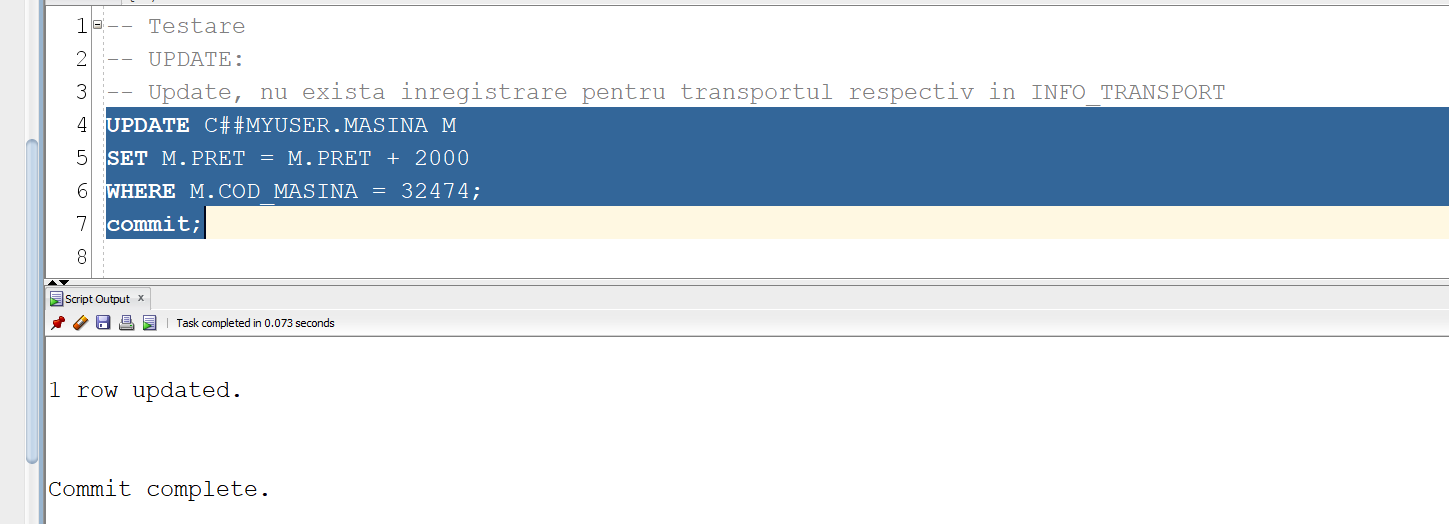
Rezultat creare trigger:



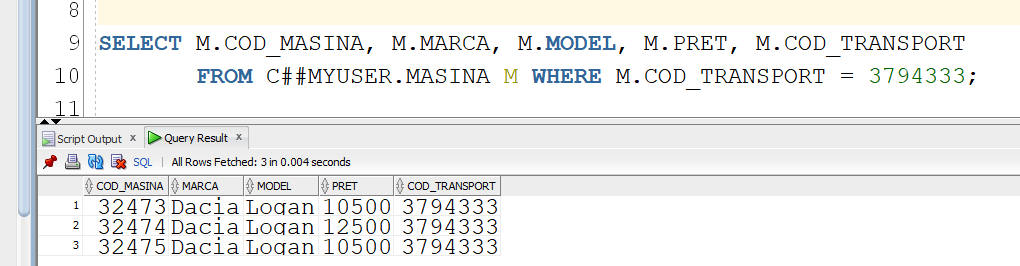
***Testare:***

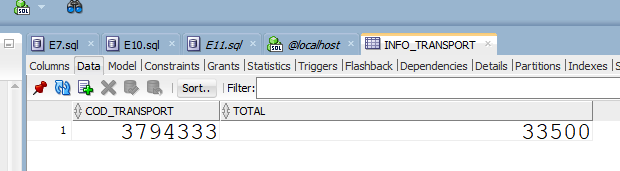
*-- Testare  
-- UPDATE:  
-- Update, nu exista inregistrare pentru transportul respectiv in INFO\_TRANSPORT*UPDATE C##MYUSER.MASINA M   
SET M.PRET = M.PRET + 2000  
WHERE M.COD\_MASINA = 32474;  
commit;  
  
SELECT M.COD\_MASINA, M.MARCA, M.MODEL, M.PRET, M.COD\_TRANSPORT  
 FROM C##MYUSER.MASINA M WHERE M.COD\_TRANSPORT = 3794333;

Rezultate rulare cod:



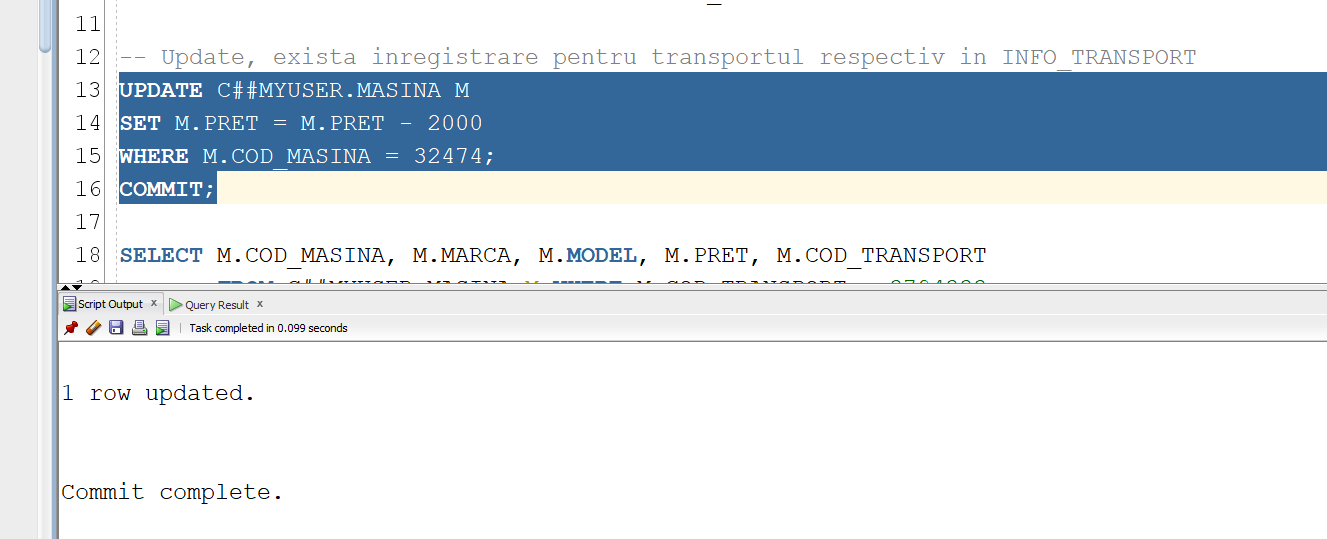
Se poate observa că există 3 mașini în transportul în care este inclusă mașina 32474, cu o valoare totală de 33 500 de euro, după reactualizare.

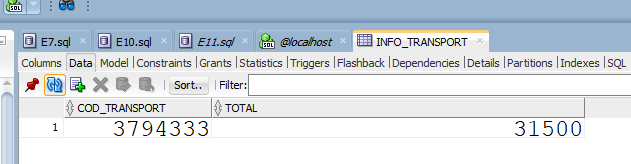




*-- Update, exista inregistrare pentru transportul respectiv in INFO\_TRANSPORT*UPDATE C##MYUSER.MASINA M   
SET M.PRET = M.PRET - 2000  
WHERE M.COD\_MASINA = 32474;  
commit;  
  
SELECT M.COD\_MASINA, M.MARCA, M.MODEL, M.PRET, M.COD\_TRANSPORT  
 FROM C##MYUSER.MASINA M WHERE M.COD\_TRANSPORT = 3794333;

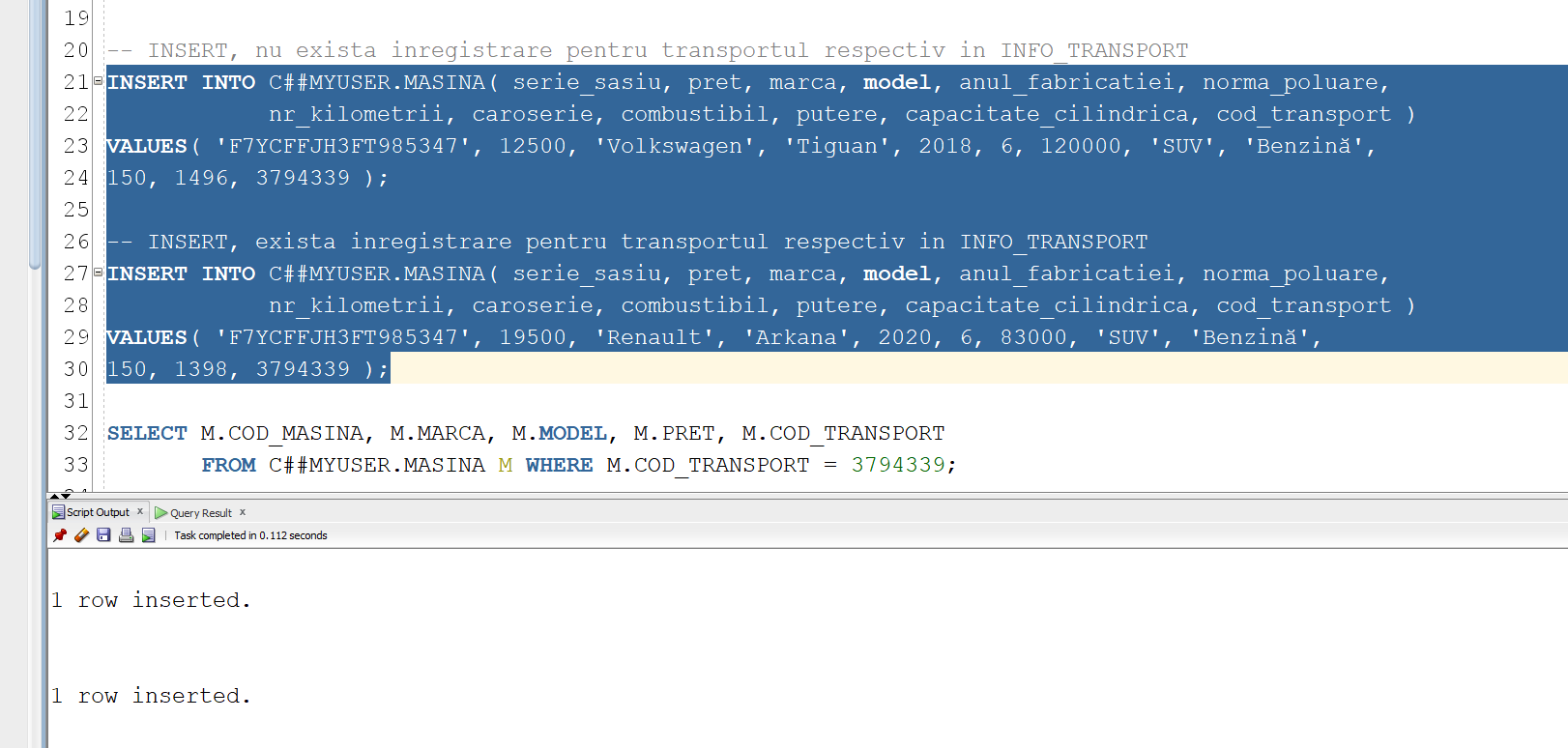
Rezultate rulare cod:

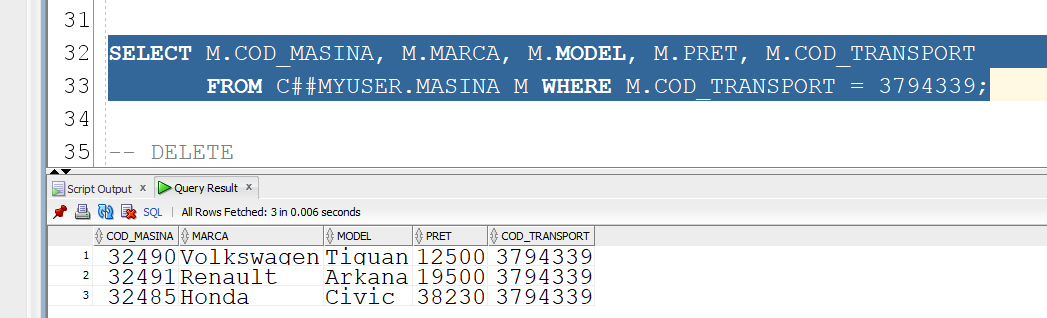




*-- INSERT, nu exista inregistrare pentru transportul respectiv in INFO\_TRANSPORT*INSERT INTO C##MYUSER.MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare,  
 nr\_kilometrii, caroserie, combustibil, putere, capacitate\_cilindrica, cod\_transport )  
VALUES( 'F7YCFFJH3FT985347', 12500, 'Volkswagen', 'Tiguan', 2018, 6, 120000, 'SUV', 'Benzină',  
150, 1496, 3794339 );  
  
*-- INSERT, exista inregistrare pentru transportul respectiv in INFO\_TRANSPORT*INSERT INTO C##MYUSER.MASINA( serie\_sasiu, pret, marca, model, anul\_fabricatiei, norma\_poluare,  
 nr\_kilometrii, caroserie, combustibil, putere, capacitate\_cilindrica, cod\_transport )  
VALUES( 'F7YCFFJH3FT985347', 19500, 'Renault', 'Arkana', 2020, 6, 83000, 'SUV', 'Benzină',  
150, 1398, 3794339 );  
  
SELECT M.COD\_MASINA, M.MARCA, M.MODEL, M.PRET, M.COD\_TRANSPORT  
 FROM C##MYUSER.MASINA M WHERE M.COD\_TRANSPORT = 3794339;

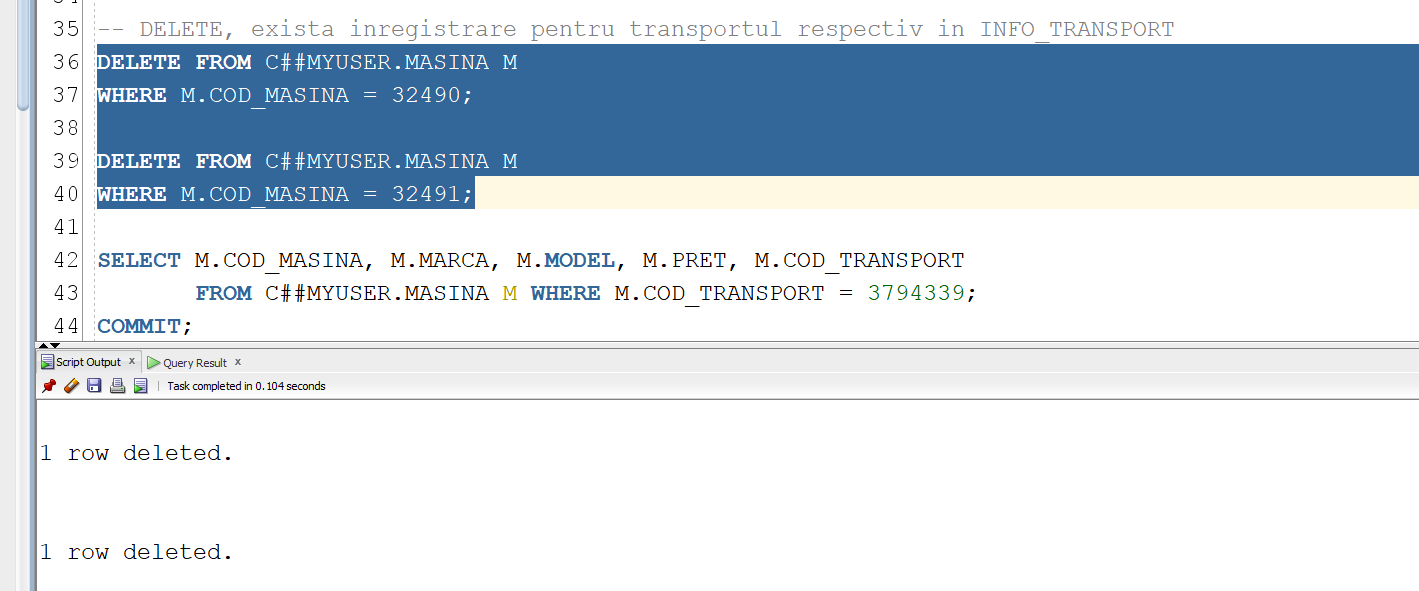
Rezultate rulare cod:

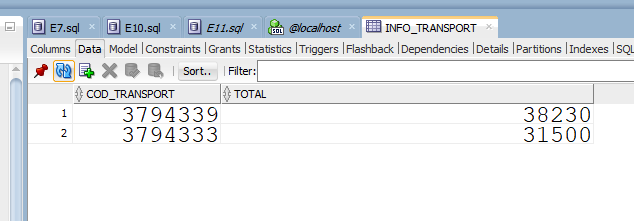


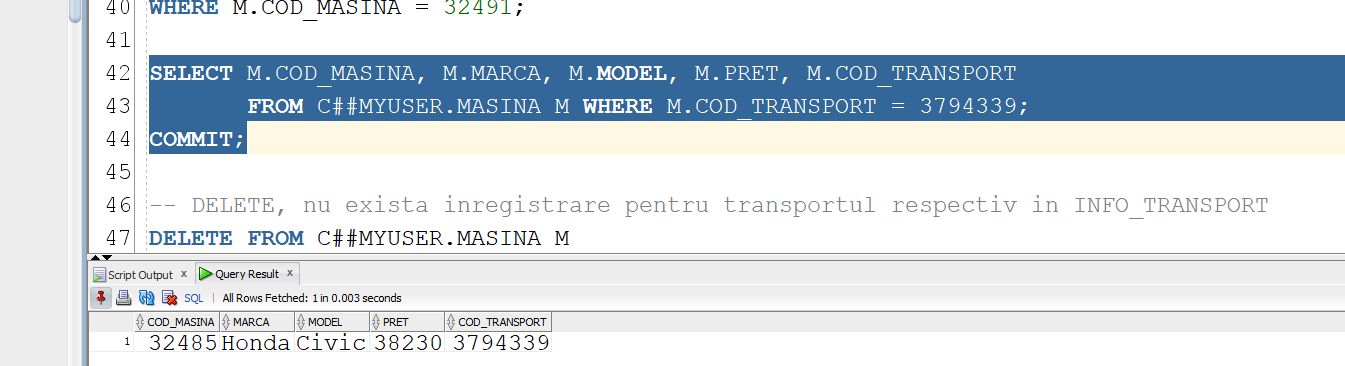


*-- DELETE, exista inregistrare pentru transportul respectiv in INFO\_TRANSPORT*DELETE FROM C##MYUSER.MASINA M  
WHERE M.COD\_MASINA = 32490;  
  
DELETE FROM C##MYUSER.MASINA M  
WHERE M.COD\_MASINA = 32491;  
  
SELECT M.COD\_MASINA, M.MARCA, M.MODEL, M.PRET, M.COD\_TRANSPORT  
 FROM C##MYUSER.MASINA M WHERE M.COD\_TRANSPORT = 3794339;  
COMMIT;

Rezultate rulare cod:

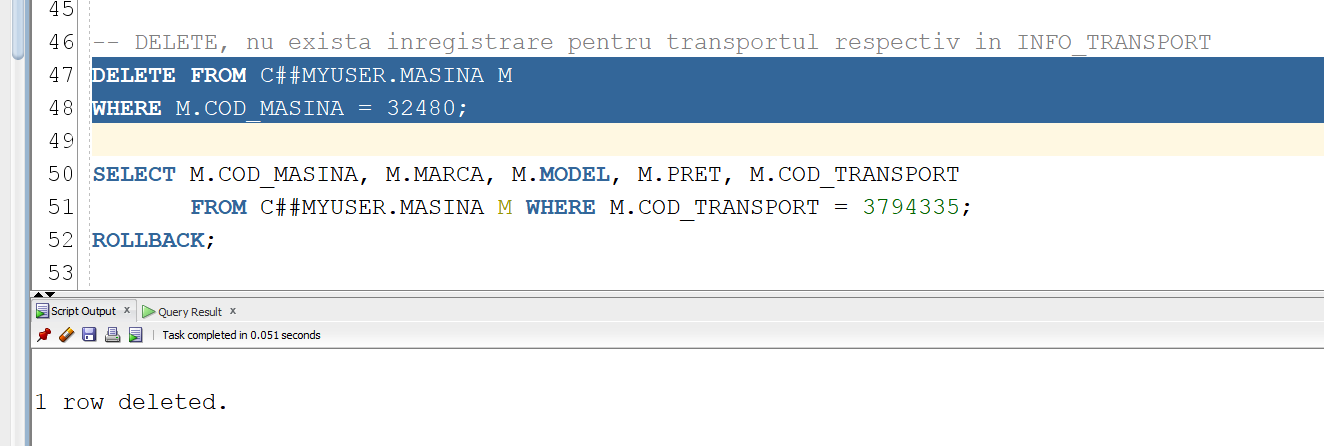


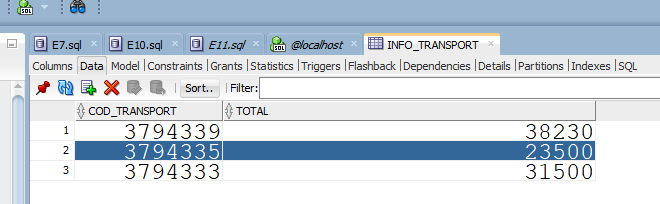


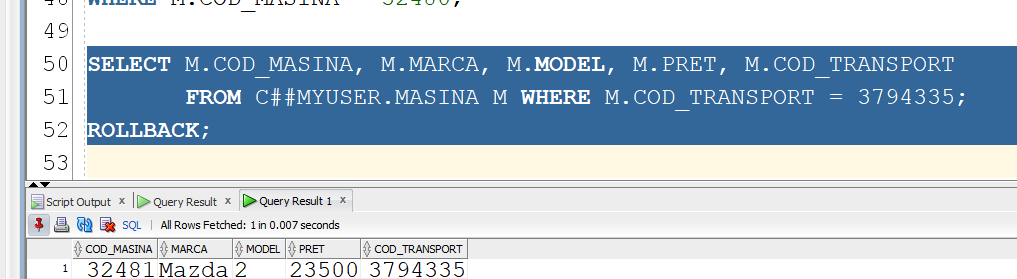


*-- DELETE, nu exista inregistrare pentru transportul respectiv in INFO\_TRANSPORT*DELETE FROM C##MYUSER.MASINA M  
WHERE M.COD\_MASINA = 32480;  
  
SELECT M.COD\_MASINA, M.MARCA, M.MODEL, M.PRET, M.COD\_TRANSPORT  
 FROM C##MYUSER.MASINA M WHERE M.COD\_TRANSPORT = 3794335;  
ROLLBACK;

Din acest transport au făcut parte 2 mașini identice, cu prețul de 23 500. Am șters una din ele. Rezultate rulare cod:





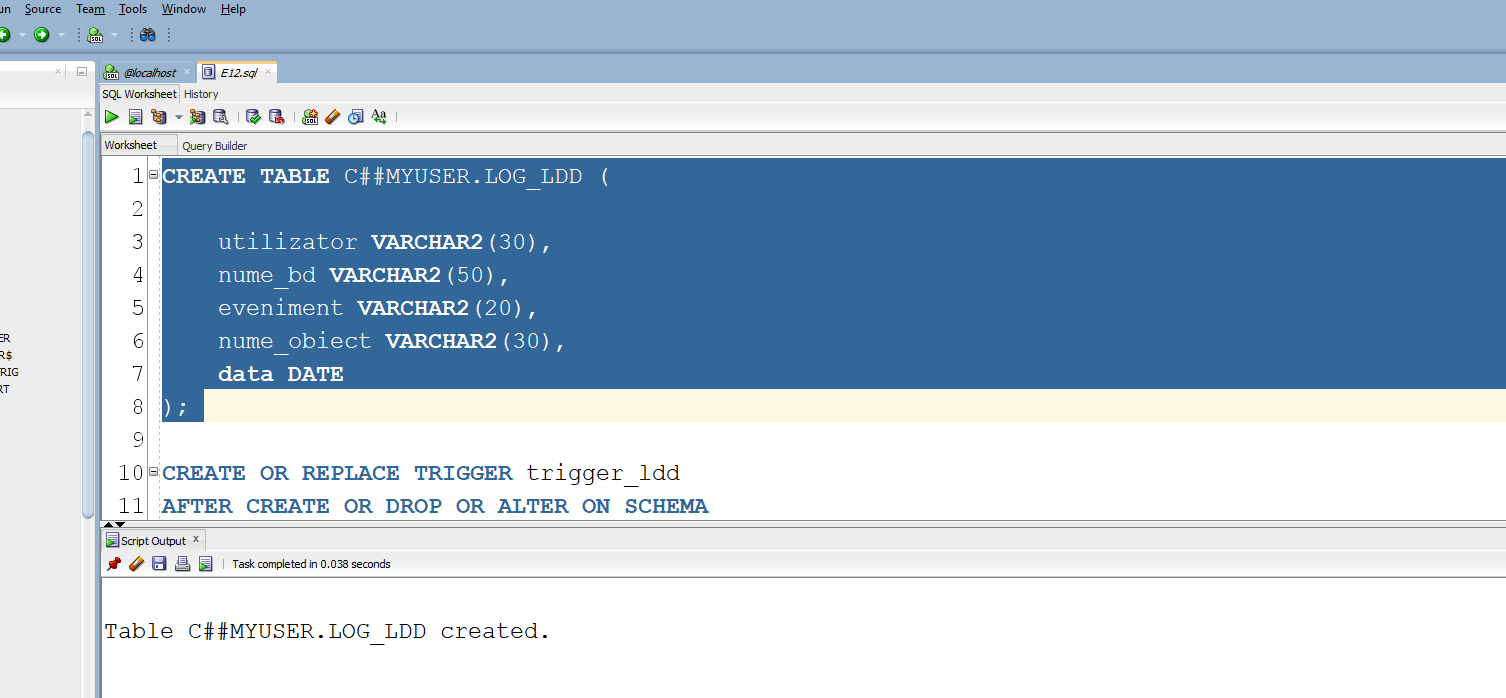


# 12) Definiți un trigger de tip LDD. Declanșați trigger-ul.

Voi crea o tabelă de log-uri, în funcție de evenimentul LDD întâlnit.

CREATE TABLE log\_ldd (  
  
 utilizator VARCHAR2(30),   
 nume\_bd VARCHAR2(50),   
 eveniment VARCHAR2(20),   
 nume\_obiect VARCHAR2(30),   
 data DATE  
);   
  
CREATE OR REPLACE TRIGGER trigger\_ldd   
AFTER CREATE OR DROP OR ALTER ON SCHEMA   
BEGIN   
  
 INSERT INTO log\_ldd   
 VALUES (SYS.*LOGIN\_USER*, SYS.*DATABASE\_NAME*, SYS.*SYSEVENT*, SYS.*DICTIONARY\_OBJ\_NAME*, *SYSDATE*);   
   
 DBMS\_OUTPUT.*PUT\_LINE*( 'A avut loc o operatie LDD de tip ' || SYS.*SYSEVENT* || ' pe obiectul '   
 || SYS.DICTIONATY\_OBJ\_NAME );  
  
END;   
/

Rezultate rulare cod:



CREATE TABLE C##MYUSER.TEST (  
   
 DATA DATE DEFAULT SYSDATE  
);  
DROP TABLE C##MYUSER.TEST;

Rezultate rulare cod:

