**Universitatea din București**

**Facultatea de Matematică și Informatică**

**Sisteme de gestiune a bazelor de date - Proiect anul II**

**BAZA DE DATE**

**A UNEI FARMACII VETERINARE**

**Anul universitar 2025-2026**

**Student:**

**Chirițoiu Andra-Florentina**

**Seria 23, grupa 234**

**CUPRINS**

[**INTRODUCERE 3**](#_mnwta4q9li6v)

[Tema proiectului 3](#_r3nalv70hjh5)

[Infrastructura utilizată 3](#_ue9pdqorboie)

[**1. Descrierea bazei de date 4**](#_89heefxjudgf)

[**2. Diagrama Entitate - Relație 6**](#_g4lav66qgb58)

[**3. Diagramă Conceptuală 7**](#_ebeiqptptkw1)

[**4. Implementarea diagramei conceptuale 8**](#_x8rp3nrc1oup)

[**5. Introducere de date 18**](#_sl7z1349lphx)

[**6. Subprogram ce utilizează trei tipuri de colecții 46**](#_v5yjt2fv2pah)

[**7. Subprogram ce utilizează două tipuri de cursoare 54**](#_xo9acbx80qi7)

[**8. Funcție care utilizează trei tabele și tratează toate excepțiile 60**](#_ris43xsaqy4o)

[**9. Procedura care utilizează cinci tabele și are două excepții personalizate 63**](#_mjwlf6xlm9cy)

# 

# INTRODUCERE

## Tema proiectului

Am ales ca temă pentru proiectul de la disciplina *Sisteme de Gestiune a Bazelor de Date* proiectarea și implementarea unei baze de date relaționale destinate **gestiunii unei farmacii veterinare**. Baza de date este concepută pentru a reflecta fidel activitatea reală a unei astfel de unități, integrând atât **componenta medicală** – consultații, rețete și campanii veterinare – cât și **componenta comercială**, care include gestiunea stocurilor pe loturi, vânzarea produselor către clienți și aprovizionarea de la furnizori.

Modelul de date urmărește relațiile dintre **clienți, animale, personal medical (medici veterinari și farmaciști), medicamente, stocuri, comenzi și facturi**, asigurând o organizare coerentă a informațiilor. Structura bazei de date permite realizarea de **interogări complexe** și **automatizarea unor procese specifice** prin utilizarea subprogramelor PL/SQL și a trigger-elor, contribuind astfel la eficientizarea activității farmaciei veterinare.

## **Infrastructura utilizată**

Baza de date aferentă acestui proiect a fost implementată utilizând **Oracle Database 19c**, pe un sistem de operare **Microsoft Windows 11 x64**. Pentru dezvoltarea și administrarea obiectelor bazei de date a fost folosit mediul **Oracle SQL Developer**, care a permis rularea și testarea scripturilor SQL și PL/SQL.

Instalarea Oracle Database a fost realizată în mod **non-container**, baza de date fiind accesată local prin intermediul identificatorului **SID HOMEDB1**, pe portul implicit **1521**. În vederea realizării proiectului individual, a fost creat un **utilizator dedicat**, căruia i-au fost atribuite drepturile necesare prin intermediul unui rol specific, asigurând astfel separarea schemelor și gestionarea corectă a obiectelor bazei de date.

# Descrierea bazei de date

**Ex.1 - cerință:**

Prezentați pe scurt baza de date.

**Ex.1 – rezolvare:**

Baza de date ***Farmacie Veterinară*** a fost proiectată pentru gestionarea integrată a activității unei farmacii veterinare, acoperind atât componenta medicală (consultații și rețete), cât și componenta comercială (gestiunea stocurilor, vânzări și aprovizionare).

**Clienții** sunt înregistrați cu datele lor de identificare și contact, iar un client poate deține unul sau mai multe **animale**. Fiecare animal este asociat unui singur proprietar, ceea ce permite evidențierea clară a istoricului medical și a interacțiunilor cu farmacia.

Animalele pot beneficia de **consultații** realizate de **medici veterinari**. O consultație este asociată unui singur animal și unui singur medic, dar atât un animal, cât și un medic pot avea mai multe consultații de-a lungul timpului. În urma unei consultații se poate emite o **rețetă**, iar rețeta poate conține unul sau mai multe **medicamente**. Relația dintre rețetă și medicamente este una de tip many-to-many și este implementată printr-o tabelă asociativă, în care se pot stoca și informații specifice prescrierii (de exemplu cantitatea și modul de administrare).

Farmacia poate desfășura **campanii veterinare** (de exemplu vaccinări sau deparazitări), iar participarea este înregistrată printr-o relație care leagă **campania**, **animalul** și **personalul medical** implicat, astfel încât să se poată urmări cine a intervenit, când și în cadrul cărei campanii.

Pe partea comercială, medicamentele sunt definite la nivel de produs (denumire, producător, substanță activă), iar evidența se face pe **loturi** în entitatea **STOC**. Astfel, un medicament poate exista în mai multe înregistrări de stoc, corespunzătoare aprovizionărilor succesive, fiecare lot având **dată de expirare**, **cantități** și **prețuri** proprii.

Comenzile sunt centralizate în entitatea **COMANDĂ** și sunt de două tipuri: **comenzi ale clienților** și **comenzi ale farmaciei către furnizori**. Pentru comenzile clienților, medicamentele sunt vândute din loturi concrete, motiv pentru care există o tabelă asociativă între **COMANDA\_CLIENT** și **STOC**, unde se înregistrează cantitatea vândută și prețul final. Pentru aprovizionare, farmacia plasează comenzi către **furnizori**, iar produsele comandate sunt evidențiate printr-o tabelă asociativă între **COMANDA\_FARMACIE** și **MEDICAMENT**, deoarece o comandă poate include mai multe medicamente, iar același medicament poate apărea în mai multe comenzi.

În final, fiecare comandă poate avea asociată o **factură**, care reține informații precum data emiterii și suma totală. Prin această structură, baza de date permite urmărirea completă a proceselor unei farmacii veterinare, de la consultații și rețete până la gestionarea stocurilor, vânzare și aprovizionare.

# Diagrama Entitate - Relație

**Ex.2 - cerință:**

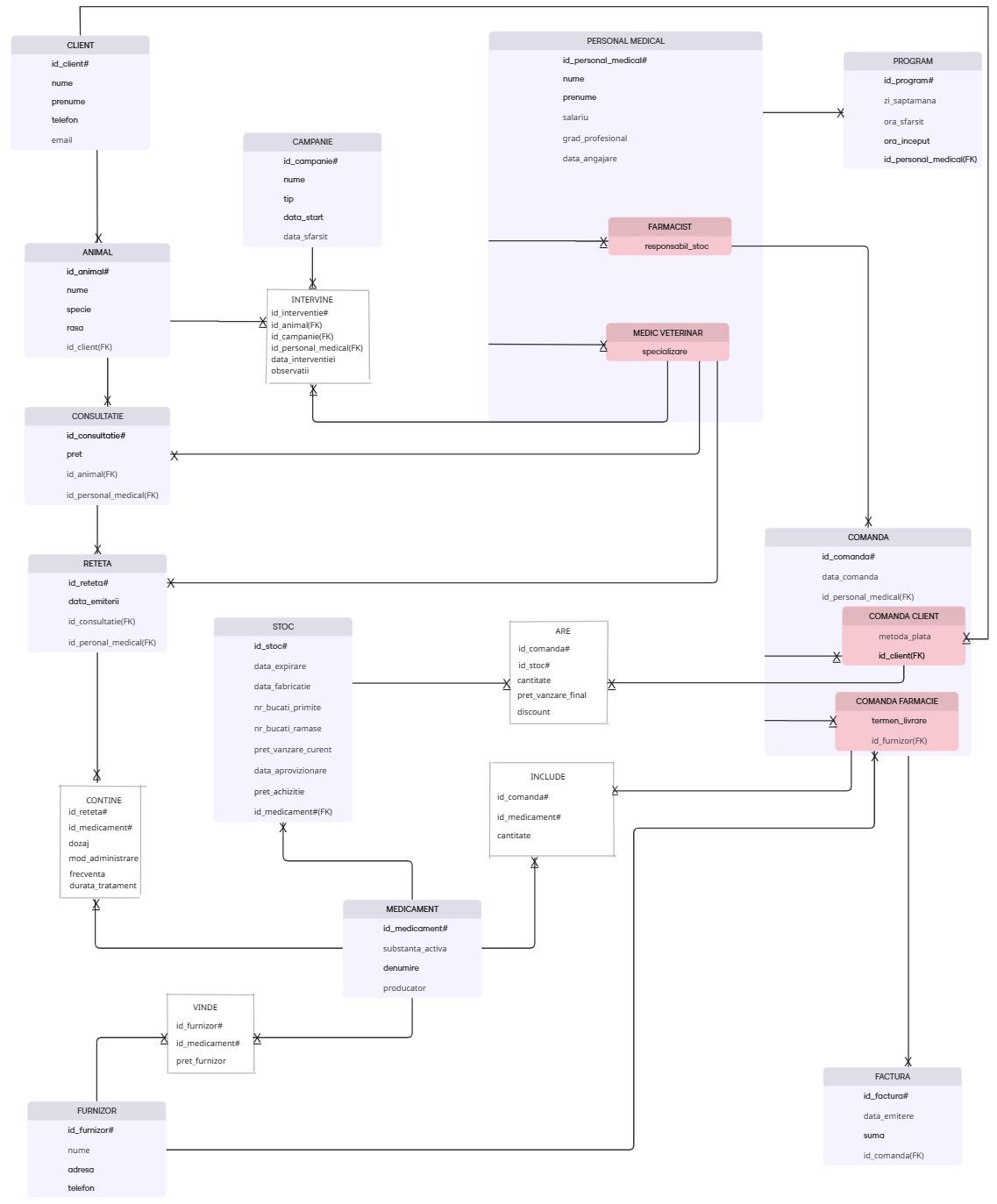
Realizați diagrama entitate-relație (ERD): entitățile, relațiile și atributele trebuie definite în limba română (vezi curs SGBD, model de diagramă entitate-relație; nu se va accepta alt format).

**Ex.2 – rezolvare:** 

# Diagramă Conceptuală

**Ex.3 - cerință:**

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.

**Ex.3 – rezolvare:**

# Implementarea diagramei conceptuale

**Ex.4 - cerință:**

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

**Ex.4 – rezolvare:**

Crearea secvențelor ce vor fi utilizate în inserarea înregistrărilor în tabele.

CREATE SEQUENCE seq\_client START WITH 1;

CREATE SEQUENCE seq\_animal START WITH 1;

CREATE SEQUENCE seq\_personal\_medical START WITH 1;

CREATE SEQUENCE seq\_campanie START WITH 1;

CREATE SEQUENCE seq\_consultatie START WITH 1;

CREATE SEQUENCE seq\_reteta START WITH 1;

CREATE SEQUENCE seq\_program START WITH 1;

CREATE SEQUENCE seq\_medicament START WITH 1;

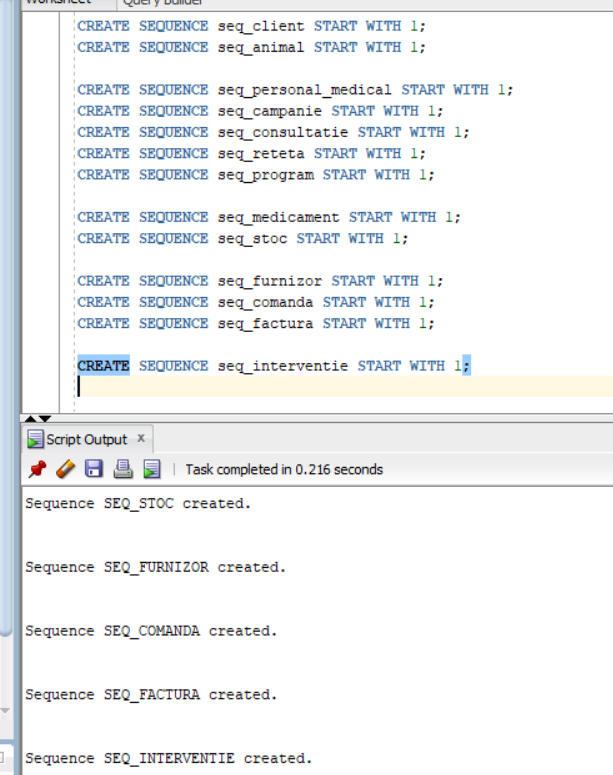
CREATE SEQUENCE seq\_stoc START WITH 1;

CREATE SEQUENCE seq\_furnizor START WITH 1;

CREATE SEQUENCE seq\_comanda START WITH 1;

CREATE SEQUENCE seq\_factura START WITH 1;

CREATE SEQUENCE seq\_interventie START WITH 1;



Crearea tabelelor.

CREATE TABLE CLIENT (

id\_client INT DEFAULT seq\_client.NEXTVAL PRIMARY KEY,

nume VARCHAR2(50) NOT NULL,

prenume VARCHAR2(50) NOT NULL,

telefon VARCHAR2(10) NOT NULL CHECK (REGEXP\_LIKE(telefon, '^[0-9]{10}$')),

email VARCHAR2(50)

);

CREATE TABLE PERSONAL\_MEDICAL (

id\_personal\_medical INT DEFAULT seq\_personal\_medical.NEXTVAL PRIMARY KEY,

nume VARCHAR2(50) NOT NULL,

prenume VARCHAR2(50) NOT NULL,

salariu NUMBER(10,2) NOT NULL CHECK (salariu > 0),

grad\_profesional VARCHAR2(50),

data\_angajare DATE NOT NULL

);

CREATE TABLE FARMACIST (

id\_personal\_medical INT PRIMARY KEY,

responsabil\_stoc NUMBER(1) DEFAULT 0 NOT NULL CHECK (responsabil\_stoc IN (0,1)),

FOREIGN KEY (id\_personal\_medical) REFERENCES PERSONAL\_MEDICAL(id\_personal\_medical)

);

CREATE TABLE MEDIC\_VETERINAR (

id\_personal\_medical INT PRIMARY KEY,

specializare VARCHAR2(50),

FOREIGN KEY (id\_personal\_medical) REFERENCES PERSONAL\_MEDICAL(id\_personal\_medical)

);

CREATE TABLE PROGRAM (

id\_program INT DEFAULT seq\_program.NEXTVAL PRIMARY KEY,

zi\_saptamana VARCHAR2(50) NOT NULL,

ora\_inceput VARCHAR2(5) NOT NULL CHECK (REGEXP\_LIKE(ora\_inceput,'^[0-2][0-9]:[0-5][0-9]$')),

ora\_sfarsit VARCHAR2(5) NOT NULL CHECK (REGEXP\_LIKE(ora\_sfarsit,'^[0-2][0-9]:[0-5][0-9]$')),

id\_personal\_medical INT NOT NULL,

UNIQUE (id\_personal\_medical, zi\_saptamana),

FOREIGN KEY (id\_personal\_medical) REFERENCES PERSONAL\_MEDICAL(id\_personal\_medical)

);

CREATE TABLE ANIMAL (

id\_animal INT DEFAULT seq\_animal.NEXTVAL PRIMARY KEY,

nume VARCHAR2(50) NOT NULL,

specie VARCHAR2(50) NOT NULL,

rasa VARCHAR2(50),

id\_client INT NOT NULL,

FOREIGN KEY (id\_client) REFERENCES CLIENT(id\_client)

);

CREATE TABLE CAMPANIE (

id\_campanie INT DEFAULT seq\_campanie.NEXTVAL PRIMARY KEY,

nume VARCHAR2(50) NOT NULL,

tip VARCHAR2(50) NOT NULL,

data\_start DATE NOT NULL,

data\_sfarsit DATE NOT NULL,

CHECK (data\_sfarsit >= data\_start)

);

CREATE TABLE CONSULTATIE (

id\_consultatie INT DEFAULT seq\_consultatie.NEXTVAL PRIMARY KEY,

pret NUMBER(10,2) NOT NULL CHECK (pret > 0),

id\_animal INT NOT NULL,

id\_personal\_medical INT NOT NULL,

FOREIGN KEY (id\_animal) REFERENCES ANIMAL(id\_animal),

FOREIGN KEY (id\_personal\_medical) REFERENCES MEDIC\_VETERINAR(id\_personal\_medical)

);

CREATE TABLE RETETA (

id\_reteta INT DEFAULT seq\_reteta.NEXTVAL PRIMARY KEY,

data\_emitere DATE DEFAULT SYSDATE NOT NULL,

id\_consultatie INT NOT NULL UNIQUE,

id\_personal\_medical INT NOT NULL,

FOREIGN KEY (id\_consultatie) REFERENCES CONSULTATIE(id\_consultatie),

FOREIGN KEY (id\_personal\_medical) REFERENCES MEDIC\_VETERINAR(id\_personal\_medical)

);

CREATE TABLE MEDICAMENT (

id\_medicament INT DEFAULT seq\_medicament.NEXTVAL PRIMARY KEY,

substanta\_activa VARCHAR2(50) NOT NULL,

denumire VARCHAR2(50) NOT NULL,

producator VARCHAR2(50) NOT NULL

);

CREATE TABLE STOC (

id\_stoc INT DEFAULT seq\_stoc.NEXTVAL PRIMARY KEY,

data\_expirare DATE NOT NULL,

data\_fabricatie DATE,

nr\_bucati\_primite INT NOT NULL CHECK (nr\_bucati\_primite > 0),

nr\_bucati\_ramase INT NOT NULL CHECK (nr\_bucati\_ramase >= 0),

pret\_vanzare\_curent NUMBER(10,2) NOT NULL CHECK (pret\_vanzare\_curent > 0),

data\_aprovizionare DATE DEFAULT SYSDATE NOT NULL,

pret\_achizitie NUMBER(10,2) NOT NULL CHECK (pret\_achizitie > 0),

id\_medicament INT NOT NULL,

CHECK (nr\_bucati\_ramase <= nr\_bucati\_primite),

FOREIGN KEY (id\_medicament) REFERENCES MEDICAMENT(id\_medicament)

);

CREATE TABLE FURNIZOR (

id\_furnizor INT DEFAULT seq\_furnizor.NEXTVAL PRIMARY KEY,

nume VARCHAR2(50) NOT NULL,

adresa VARCHAR2(50),

telefon VARCHAR2(10) NOT NULL CHECK (REGEXP\_LIKE(telefon, '^[0-9]{10}$'))

);

CREATE TABLE COMANDA (

id\_comanda INT DEFAULT seq\_comanda.NEXTVAL PRIMARY KEY,

data\_comanda DATE DEFAULT SYSDATE NOT NULL,

id\_personal\_medical INT NOT NULL,

FOREIGN KEY (id\_personal\_medical) REFERENCES FARMACIST(id\_personal\_medical)

);

CREATE TABLE COMANDA\_CLIENT (

id\_comanda INT PRIMARY KEY,

metoda\_plata VARCHAR2(20) NOT NULL CHECK (LOWER(metoda\_plata) IN ('cash','card')),

id\_client INT NOT NULL,

FOREIGN KEY (id\_comanda) REFERENCES COMANDA(id\_comanda),

FOREIGN KEY (id\_client) REFERENCES CLIENT(id\_client)

);

CREATE TABLE COMANDA\_FARMACIE (

id\_comanda INT PRIMARY KEY,

termen\_livrare DATE,

id\_furnizor INT NOT NULL,

FOREIGN KEY (id\_comanda) REFERENCES COMANDA(id\_comanda),

FOREIGN KEY (id\_furnizor) REFERENCES FURNIZOR(id\_furnizor)

);

CREATE TABLE FACTURA (

id\_factura INT DEFAULT seq\_factura.NEXTVAL PRIMARY KEY,

data\_emitere DATE DEFAULT SYSDATE NOT NULL,

suma NUMBER(12,2) NOT NULL CHECK (suma > 0),

id\_comanda INT NOT NULL UNIQUE,

FOREIGN KEY (id\_comanda) REFERENCES COMANDA(id\_comanda)

);

CREATE TABLE VINDE (

id\_furnizor INT NOT NULL,

id\_medicament INT NOT NULL,

pret\_furnizor NUMBER(10,2) NOT NULL CHECK (pret\_furnizor > 0),

PRIMARY KEY (id\_furnizor, id\_medicament),

FOREIGN KEY (id\_furnizor) REFERENCES FURNIZOR(id\_furnizor),

FOREIGN KEY (id\_medicament) REFERENCES MEDICAMENT(id\_medicament)

);

CREATE TABLE INCLUDE (

id\_comanda INT NOT NULL,

id\_medicament INT NOT NULL,

cantitate INT NOT NULL CHECK (cantitate > 0),

PRIMARY KEY (id\_comanda, id\_medicament),

FOREIGN KEY (id\_comanda) REFERENCES COMANDA\_FARMACIE(id\_comanda),

FOREIGN KEY (id\_medicament) REFERENCES MEDICAMENT(id\_medicament)

);

CREATE TABLE ARE (

id\_comanda INT NOT NULL,

id\_stoc INT NOT NULL,

cantitate INT NOT NULL CHECK (cantitate > 0),

pret\_vanzare\_final NUMBER(10,2) NOT NULL CHECK (pret\_vanzare\_final > 0),

discount NUMBER(5,2) DEFAULT 0 NOT NULL CHECK (discount >= 0 AND discount <= 100),

PRIMARY KEY (id\_comanda, id\_stoc),

FOREIGN KEY (id\_comanda) REFERENCES COMANDA\_CLIENT(id\_comanda),

FOREIGN KEY (id\_stoc) REFERENCES STOC(id\_stoc)

);

CREATE TABLE CONTINE (

id\_reteta INT NOT NULL,

id\_medicament INT NOT NULL,

dozaj VARCHAR2(50),

mod\_administrare VARCHAR2(50),

frecventa VARCHAR2(50),

durata\_tratament VARCHAR2(50),

PRIMARY KEY (id\_reteta, id\_medicament),

FOREIGN KEY (id\_reteta) REFERENCES RETETA(id\_reteta),

FOREIGN KEY (id\_medicament) REFERENCES MEDICAMENT(id\_medicament)

);

CREATE TABLE INTERVINE (

id\_interventie INT DEFAULT seq\_interventie.NEXTVAL PRIMARY KEY,

id\_animal INT NOT NULL,

id\_campanie INT NOT NULL,

id\_personal\_medical INT NOT NULL,

data\_interventiei DATE DEFAULT SYSDATE NOT NULL,

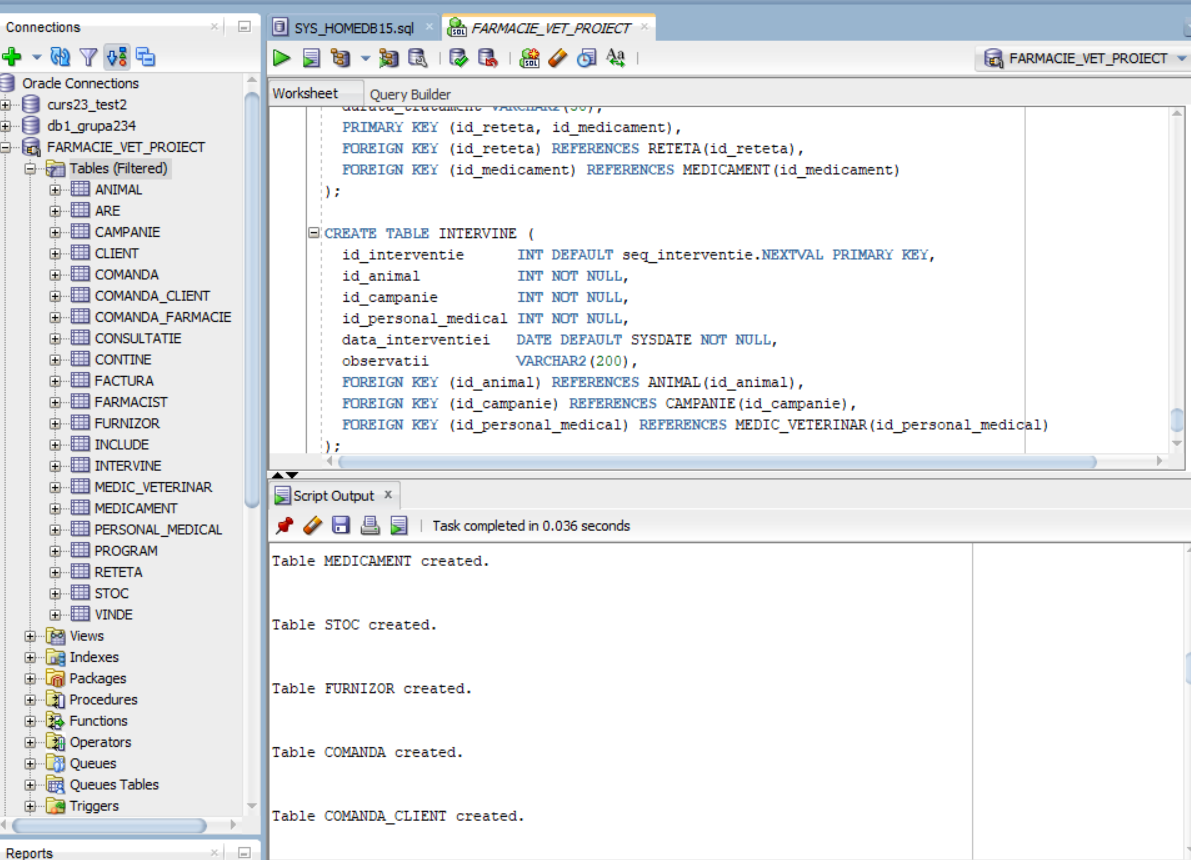
observatii VARCHAR2(200),

FOREIGN KEY (id\_animal) REFERENCES ANIMAL(id\_animal),

FOREIGN KEY (id\_campanie) REFERENCES CAMPANIE(id\_campanie),

FOREIGN KEY (id\_personal\_medical) REFERENCES MEDIC\_VETERINAR(id\_personal\_medical)

);



# Introducere de date

**Ex.5 - cerință:**

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

**Ex.5 – rezolvare:**

INSERT INTO CLIENT(nume, prenume, telefon, email) VALUES ('Popescu','Ana','0712345678','ana.popescu@gmail.com');

INSERT INTO CLIENT(nume, prenume, telefon, email) VALUES ('Ionescu','Vlad','0723456789','vlad.ionescu@gmail.com');

INSERT INTO CLIENT(nume, prenume, telefon, email) VALUES ('Dumitru','Andreea','0734567890','andreea.dumitru@yahoo.com');

INSERT INTO CLIENT(nume, prenume, telefon, email) VALUES ('Marin','Mihai','0745678901','mihai.marin@gmail.com');

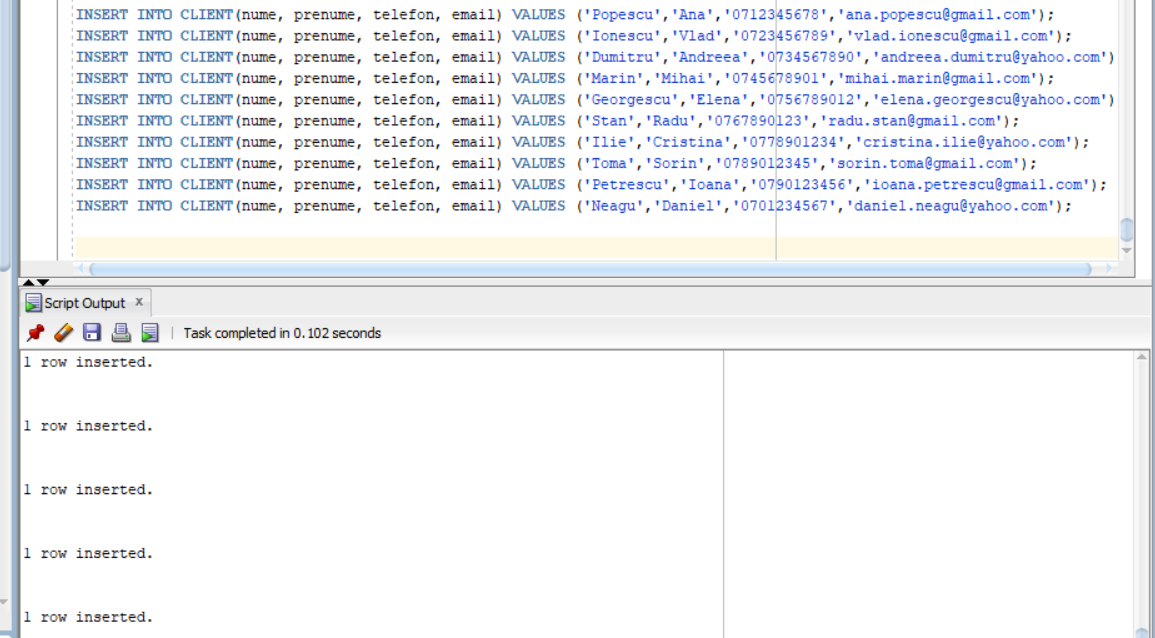
INSERT INTO CLIENT(nume, prenume, telefon, email) VALUES ('Georgescu','Elena','0756789012','elena.georgescu@yahoo.com');

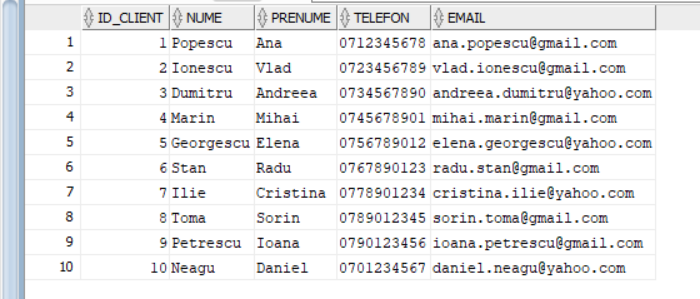
INSERT INTO CLIENT(nume, prenume, telefon, email) VALUES ('Stan','Radu','0767890123','radu.stan@gmail.com');

INSERT INTO CLIENT(nume, prenume, telefon, email) VALUES ('Ilie','Cristina','0778901234','cristina.ilie@yahoo.com');

INSERT INTO CLIENT(nume, prenume, telefon, email) VALUES ('Toma','Sorin','0789012345','sorin.toma@gmail.com');

INSERT INTO CLIENT(nume, prenume, telefon, email) VALUES ('Petrescu','Ioana','0790123456','ioana.petrescu@gmail.com');

INSERT INTO CLIENT(nume, prenume, telefon, email) VALUES ('Neagu','Daniel','0701234567','[daniel.neagu@yahoo.com](mailto:daniel.neagu@yahoo.com)');

****

INSERT INTO PERSONAL\_MEDICAL(nume, prenume, salariu, grad\_profesional, data\_angajare)

VALUES ('Vasilescu','Dan',8000,'Dr.',DATE '2020-01-15');

INSERT INTO PERSONAL\_MEDICAL(nume, prenume, salariu, grad\_profesional, data\_angajare)

VALUES ('Marinescu','Laura',7500,'Dr.',DATE '2019-03-10');

INSERT INTO PERSONAL\_MEDICAL(nume, prenume, salariu, grad\_profesional, data\_angajare)

VALUES ('Stoica','Radu',6000,'Farm.',DATE '2021-05-20');

INSERT INTO PERSONAL\_MEDICAL(nume, prenume, salariu, grad\_profesional, data\_angajare)

VALUES ('Popa','Carmen',5500,'Farm.',DATE '2022-02-14');

INSERT INTO PERSONAL\_MEDICAL(nume, prenume, salariu, grad\_profesional, data\_angajare)

VALUES ('Tudor','Bogdan',7800,'Dr.',DATE '2018-09-05');

INSERT INTO PERSONAL\_MEDICAL(nume, prenume, salariu, grad\_profesional, data\_angajare)

VALUES ('Munteanu','Diana',5800,'Farm.',DATE '2023-01-10');

INSERT INTO PERSONAL\_MEDICAL(nume, prenume, salariu, grad\_profesional, data\_angajare)

VALUES ('Ionescu','Adrian',7200,'Dr.',DATE '2021-11-08');

INSERT INTO PERSONAL\_MEDICAL(nume, prenume, salariu, grad\_profesional, data\_angajare)

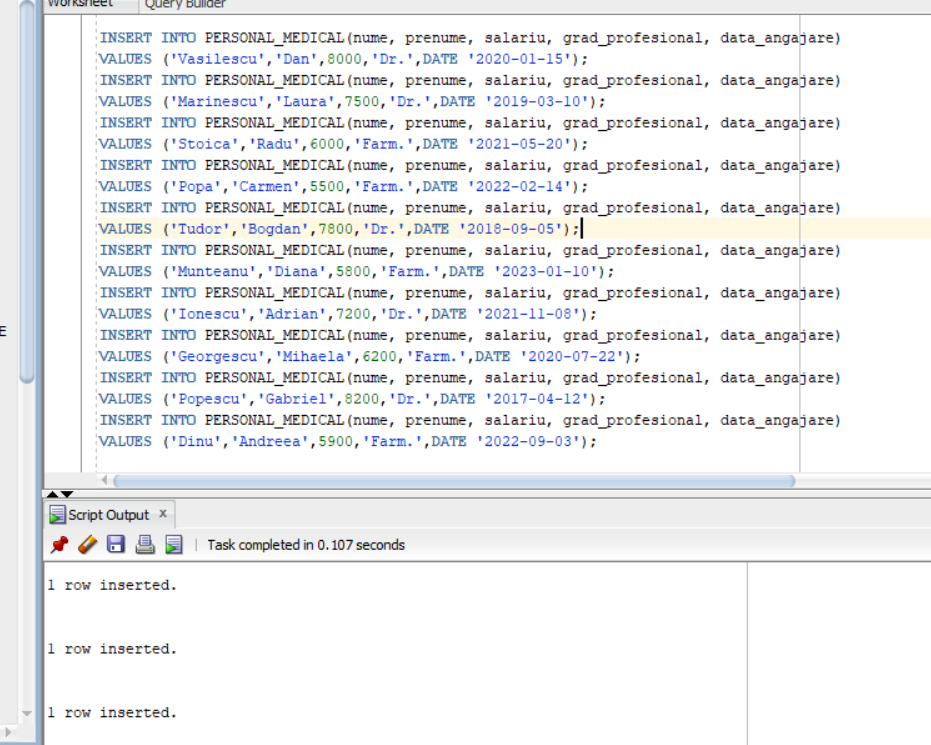
VALUES ('Georgescu','Mihaela',6200,'Farm.',DATE '2020-07-22');

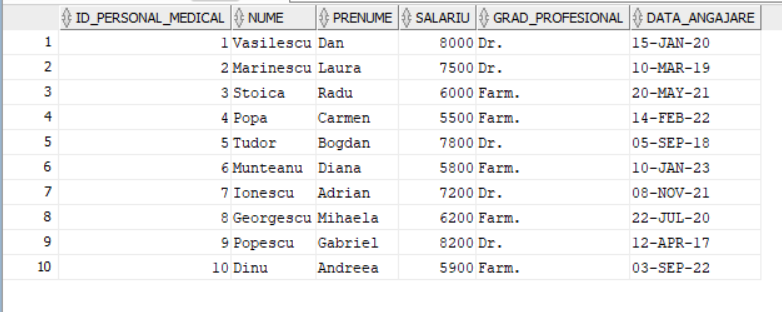
INSERT INTO PERSONAL\_MEDICAL(nume, prenume, salariu, grad\_profesional, data\_angajare)

VALUES ('Popescu','Gabriel',8200,'Dr.',DATE '2017-04-12');

INSERT INTO PERSONAL\_MEDICAL(nume, prenume, salariu, grad\_profesional, data\_angajare)

VALUES ('Dinu','Andreea',5900,'Farm.',DATE '2022-09-03');





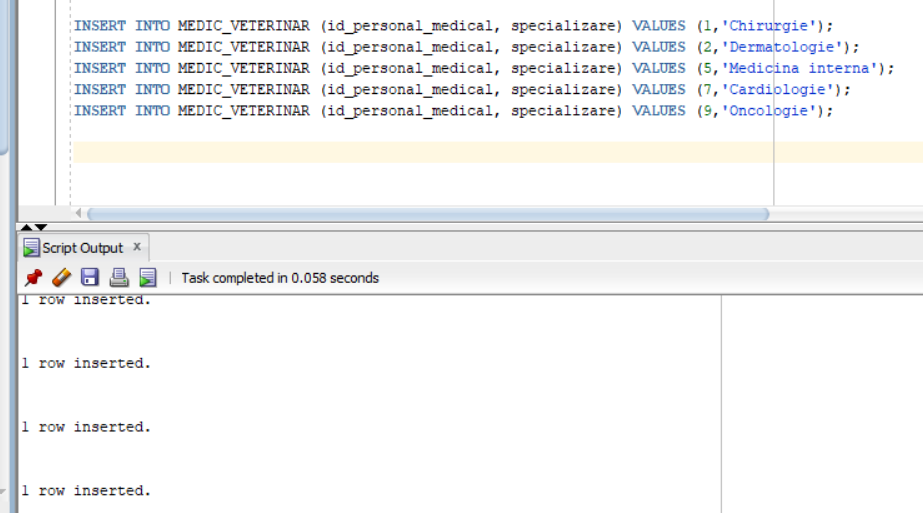
INSERT INTO MEDIC\_VETERINAR (id\_personal\_medical, specializare) VALUES (1,'Chirurgie');

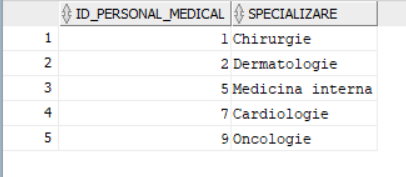
INSERT INTO MEDIC\_VETERINAR (id\_personal\_medical, specializare) VALUES (2,'Dermatologie');

INSERT INTO MEDIC\_VETERINAR (id\_personal\_medical, specializare) VALUES (5,'Medicina interna');

INSERT INTO MEDIC\_VETERINAR (id\_personal\_medical, specializare) VALUES (7,'Cardiologie');

INSERT INTO MEDIC\_VETERINAR (id\_personal\_medical, specializare) VALUES (9,'Oncologie');





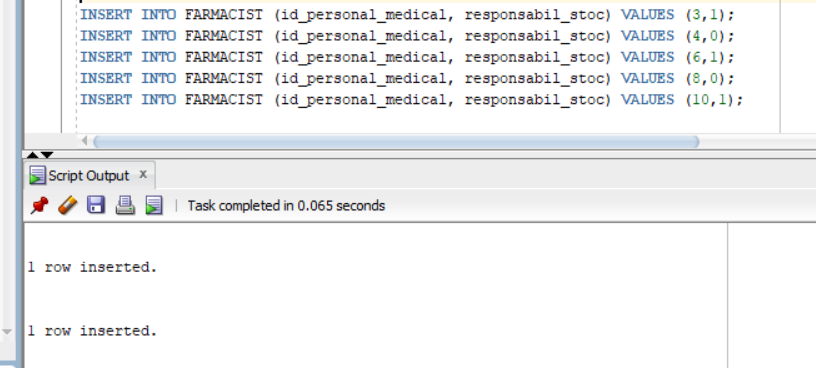
INSERT INTO FARMACIST (id\_personal\_medical, responsabil\_stoc) VALUES (3,1);

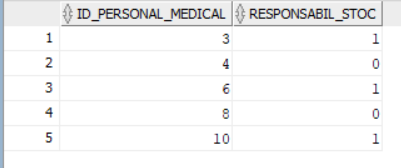
INSERT INTO FARMACIST (id\_personal\_medical, responsabil\_stoc) VALUES (4,0);

INSERT INTO FARMACIST (id\_personal\_medical, responsabil\_stoc) VALUES (6,1);

INSERT INTO FARMACIST (id\_personal\_medical, responsabil\_stoc) VALUES (8,0);

INSERT INTO FARMACIST (id\_personal\_medical, responsabil\_stoc) VALUES (10,1);





INSERT INTO PROGRAM (zi\_saptamana, ora\_inceput, ora\_sfarsit, id\_personal\_medical) VALUES ('Luni','08:00','16:00',1);

INSERT INTO PROGRAM (zi\_saptamana, ora\_inceput, ora\_sfarsit, id\_personal\_medical) VALUES ('Marti','08:00','16:00',1);

INSERT INTO PROGRAM (zi\_saptamana, ora\_inceput, ora\_sfarsit, id\_personal\_medical) VALUES ('Marti','09:00','17:00',2);

INSERT INTO PROGRAM (zi\_saptamana, ora\_inceput, ora\_sfarsit, id\_personal\_medical) VALUES ('Miercuri','09:00','17:00',2);

INSERT INTO PROGRAM (zi\_saptamana, ora\_inceput, ora\_sfarsit, id\_personal\_medical) VALUES ('Miercuri','10:00','18:00',3);

INSERT INTO PROGRAM (zi\_saptamana, ora\_inceput, ora\_sfarsit, id\_personal\_medical) VALUES ('Joi','10:00','18:00',4);

INSERT INTO PROGRAM (zi\_saptamana, ora\_inceput, ora\_sfarsit, id\_personal\_medical) VALUES ('Vineri','10:00','18:00',4);

INSERT INTO PROGRAM (zi\_saptamana, ora\_inceput, ora\_sfarsit, id\_personal\_medical) VALUES ('Vineri','08:00','16:00',5);

INSERT INTO PROGRAM (zi\_saptamana, ora\_inceput, ora\_sfarsit, id\_personal\_medical) VALUES ('Sambata','08:00','12:00',5);

INSERT INTO PROGRAM (zi\_saptamana, ora\_inceput, ora\_sfarsit, id\_personal\_medical) VALUES ('Sambata','10:00','14:00',6);

INSERT INTO PROGRAM (zi\_saptamana, ora\_inceput, ora\_sfarsit, id\_personal\_medical) VALUES ('Vineri','10:00','14:00',6);

INSERT INTO PROGRAM (zi\_saptamana, ora\_inceput, ora\_sfarsit, id\_personal\_medical) VALUES ('Luni','12:00','20:00',7);

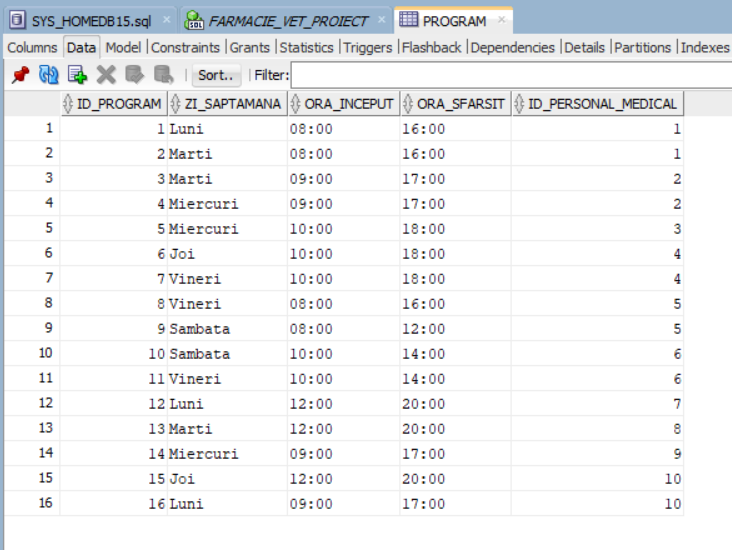
INSERT INTO PROGRAM (zi\_saptamana, ora\_inceput, ora\_sfarsit, id\_personal\_medical) VALUES ('Marti','12:00','20:00',8);

INSERT INTO PROGRAM (zi\_saptamana, ora\_inceput, ora\_sfarsit, id\_personal\_medical) VALUES ('Miercuri','09:00','17:00',9);

INSERT INTO PROGRAM (zi\_saptamana, ora\_inceput, ora\_sfarsit, id\_personal\_medical) VALUES ('Joi','12:00','20:00',10);

INSERT INTO PROGRAM (zi\_saptamana, ora\_inceput, ora\_sfarsit, id\_personal\_medical) VALUES ('Luni','09:00','17:00',10);





INSERT INTO ANIMAL (nume, specie, rasa, id\_client) VALUES ('Rex','Caine','Labrador',1);

INSERT INTO ANIMAL (nume, specie, rasa, id\_client) VALUES ('Mia','Pisica','Persana',2);

INSERT INTO ANIMAL (nume, specie, rasa, id\_client) VALUES ('Max','Caine','Ciobanesc german',3);

INSERT INTO ANIMAL (nume, specie, rasa, id\_client) VALUES ('Luna','Pisica','Siameza',4);

INSERT INTO ANIMAL (nume, specie, rasa, id\_client) VALUES ('Buddy','Caine','Golden Retriever',5);

INSERT INTO ANIMAL (nume, specie, rasa, id\_client) VALUES ('Kira','Caine','Beagle',6);

INSERT INTO ANIMAL (nume, specie, rasa, id\_client) VALUES ('Oscar','Pisica','Europeana',7);

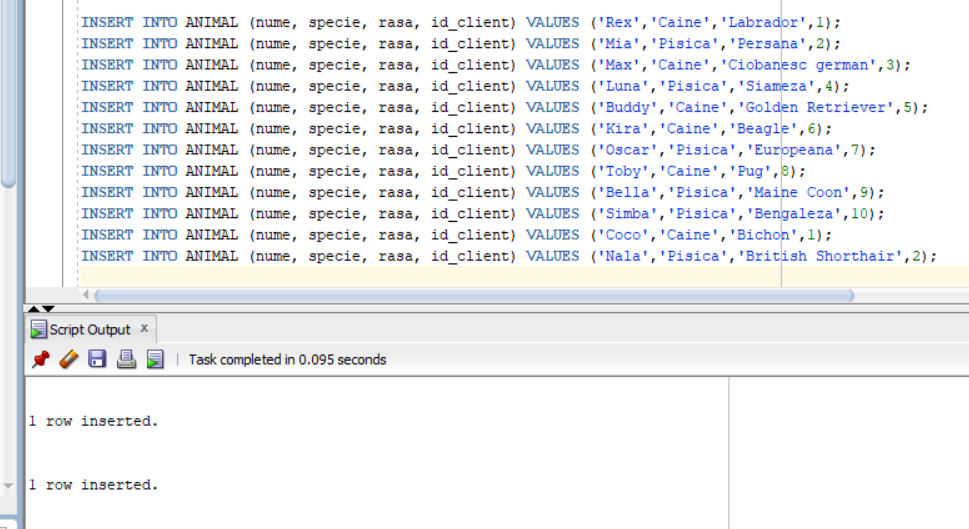
INSERT INTO ANIMAL (nume, specie, rasa, id\_client) VALUES ('Toby','Caine','Pug',8);

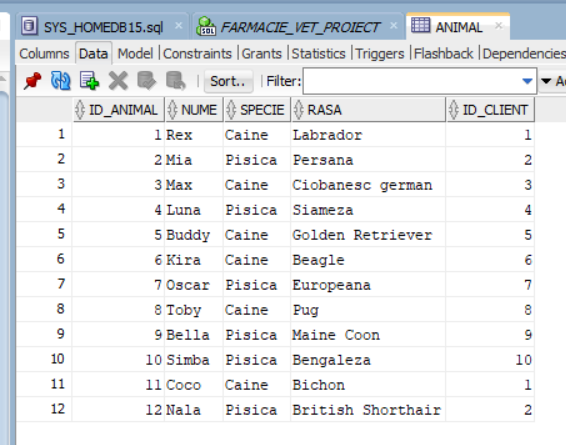
INSERT INTO ANIMAL (nume, specie, rasa, id\_client) VALUES ('Bella','Pisica','Maine Coon',9);

INSERT INTO ANIMAL (nume, specie, rasa, id\_client) VALUES ('Simba','Pisica','Bengaleza',10);

INSERT INTO ANIMAL (nume, specie, rasa, id\_client) VALUES ('Coco','Caine','Bichon',1);

INSERT INTO ANIMAL (nume, specie, rasa, id\_client) VALUES ('Nala','Pisica','British Shorthair',2);





INSERT INTO CAMPANIE (nume, tip, data\_start, data\_sfarsit)

VALUES ('Vaccinare antirabica','Vaccinare',DATE '2025-01-10',DATE '2025-02-10');

INSERT INTO CAMPANIE (nume, tip, data\_start, data\_sfarsit)

VALUES ('Deparazitare gratuita','Preventie',DATE '2025-03-01',DATE '2025-03-31');

INSERT INTO CAMPANIE (nume, tip, data\_start, data\_sfarsit)

VALUES ('Sterilizare','Chirurgie',DATE '2025-04-01',DATE '2025-04-30');

INSERT INTO CAMPANIE (nume, tip, data\_start, data\_sfarsit)

VALUES ('Control cardiologic','Diagnostic',DATE '2025-05-01',DATE '2025-05-31');

INSERT INTO CAMPANIE (nume, tip, data\_start, data\_sfarsit)

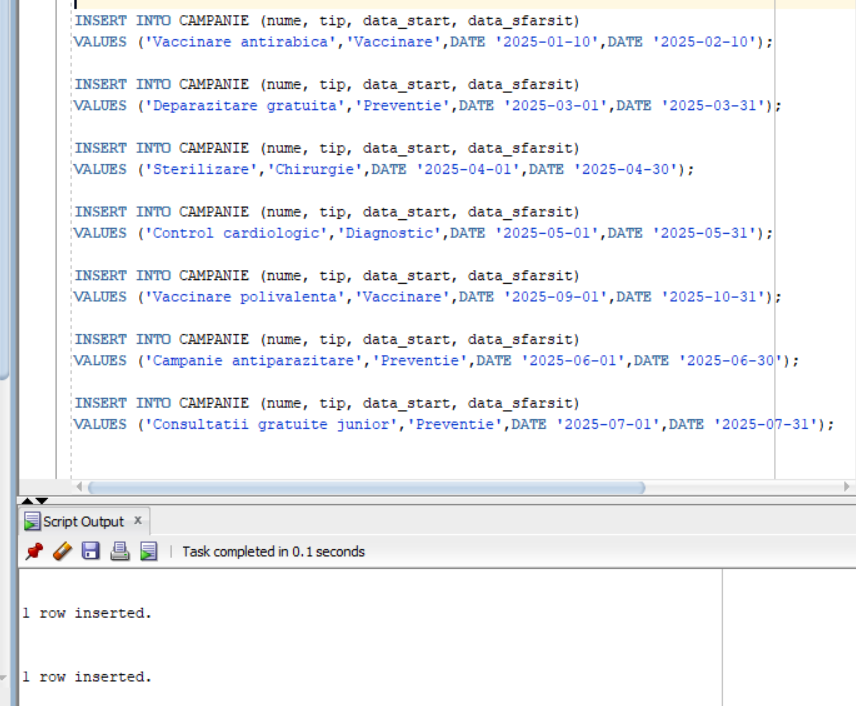
VALUES ('Vaccinare polivalenta','Vaccinare',DATE '2025-09-01',DATE '2025-10-31');

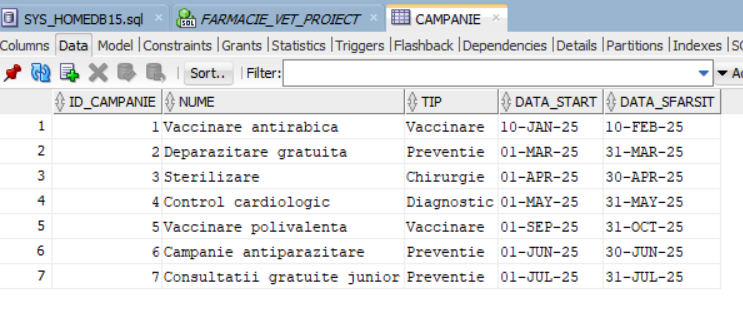
INSERT INTO CAMPANIE (nume, tip, data\_start, data\_sfarsit)

VALUES ('Campanie antiparazitare','Preventie',DATE '2025-06-01',DATE '2025-06-30');

INSERT INTO CAMPANIE (nume, tip, data\_start, data\_sfarsit)

VALUES ('Consultatii gratuite junior','Preventie',DATE '2025-07-01',DATE '2025-07-31');





INSERT INTO MEDICAMENT (substanta\_activa, denumire, producator) VALUES ('Amoxicilina','Amoxivet','VetPharma');

INSERT INTO MEDICAMENT (substanta\_activa, denumire, producator) VALUES ('Meloxicam','Meloxivet','AnimalHealth');

INSERT INTO MEDICAMENT (substanta\_activa, denumire, producator) VALUES ('Fipronil','Frontline','Boehringer');

INSERT INTO MEDICAMENT (substanta\_activa, denumire, producator) VALUES ('Ivermectina','Ivomec','Merial');

INSERT INTO MEDICAMENT (substanta\_activa, denumire, producator) VALUES ('Dexametazona','Dexavet','VetMedica');

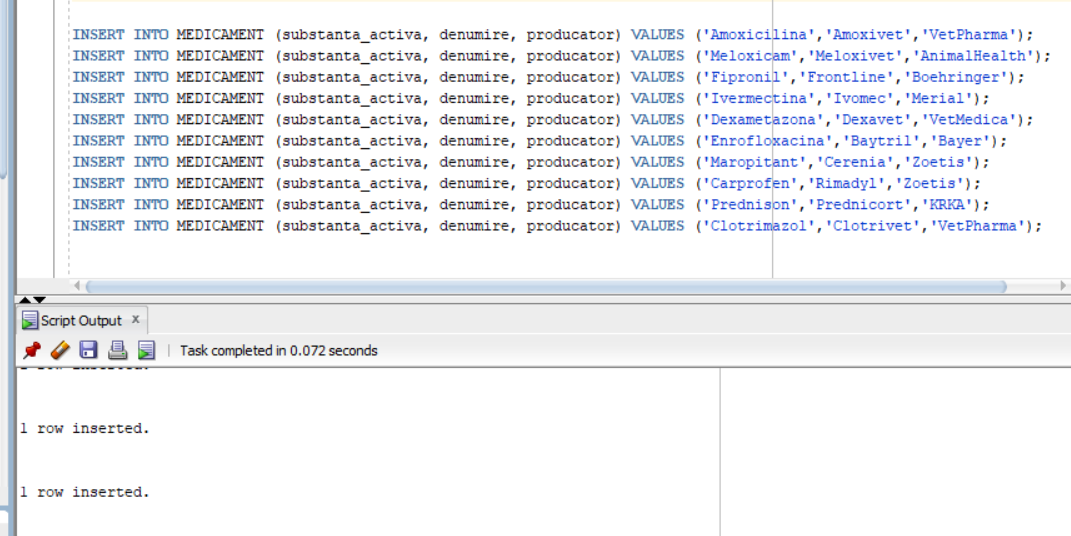
INSERT INTO MEDICAMENT (substanta\_activa, denumire, producator) VALUES ('Enrofloxacina','Baytril','Bayer');

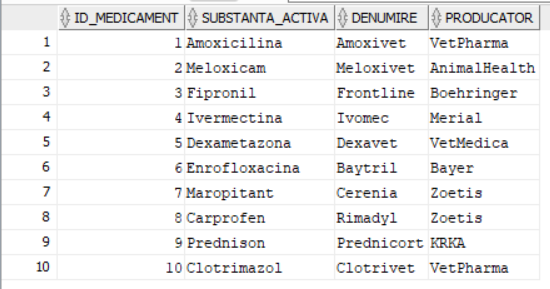
INSERT INTO MEDICAMENT (substanta\_activa, denumire, producator) VALUES ('Maropitant','Cerenia','Zoetis');

INSERT INTO MEDICAMENT (substanta\_activa, denumire, producator) VALUES ('Carprofen','Rimadyl','Zoetis');

INSERT INTO MEDICAMENT (substanta\_activa, denumire, producator) VALUES ('Prednison','Prednicort','KRKA');

INSERT INTO MEDICAMENT (substanta\_activa, denumire, producator) VALUES ('Clotrimazol','Clotrivet','VetPharma');





INSERT INTO FURNIZOR (nume, adresa, telefon) VALUES ('VetPharma Distribution','Bucuresti','0213456789');

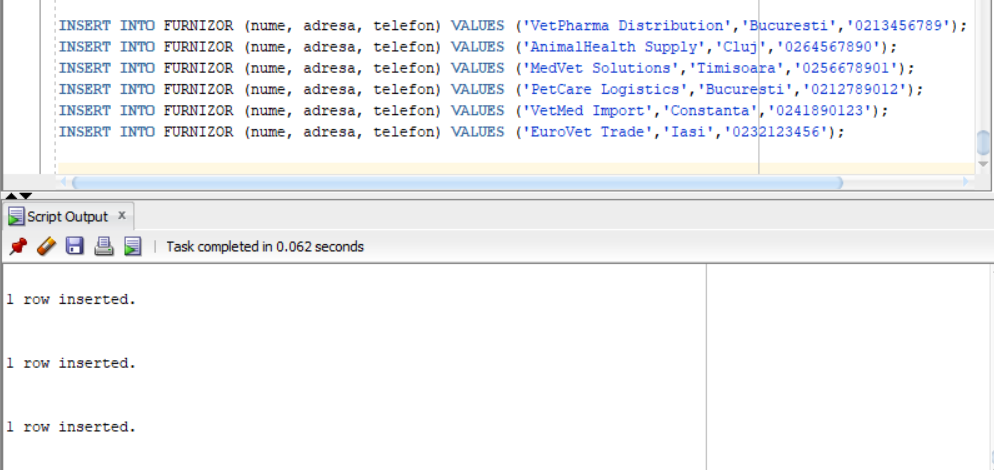
INSERT INTO FURNIZOR (nume, adresa, telefon) VALUES ('AnimalHealth Supply','Cluj','0264567890');

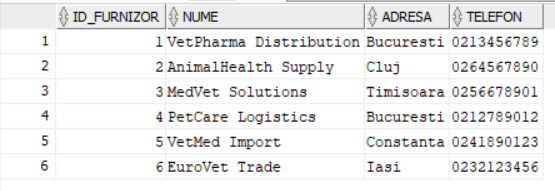
INSERT INTO FURNIZOR (nume, adresa, telefon) VALUES ('MedVet Solutions','Timisoara','0256678901');

INSERT INTO FURNIZOR (nume, adresa, telefon) VALUES ('PetCare Logistics','Bucuresti','0212789012');

INSERT INTO FURNIZOR (nume, adresa, telefon) VALUES ('VetMed Import','Constanta','0241890123');

INSERT INTO FURNIZOR (nume, adresa, telefon) VALUES ('EuroVet Trade','Iasi','0232123456');





INSERT INTO STOC (data\_expirare, data\_fabricatie, nr\_bucati\_primite, nr\_bucati\_ramase, pret\_vanzare\_curent, data\_aprovizionare, pret\_achizitie, id\_medicament)

VALUES (DATE '2026-12-31', DATE '2025-01-10', 100, 80, 30, DATE '2025-02-01', 20, 1);

INSERT INTO STOC (data\_expirare, data\_fabricatie, nr\_bucati\_primite, nr\_bucati\_ramase, pret\_vanzare\_curent, data\_aprovizionare, pret\_achizitie, id\_medicament)

VALUES (DATE '2027-01-31', DATE '2025-06-01', 120, 120, 35, DATE '2025-06-10', 22, 1);

INSERT INTO STOC (data\_expirare, data\_fabricatie, nr\_bucati\_primite, nr\_bucati\_ramase, pret\_vanzare\_curent, data\_aprovizionare, pret\_achizitie, id\_medicament)

VALUES (DATE '2026-06-30', DATE '2025-02-10', 50, 35, 55, DATE '2025-02-15', 40, 2);

INSERT INTO STOC (data\_expirare, data\_fabricatie, nr\_bucati\_primite, nr\_bucati\_ramase, pret\_vanzare\_curent, data\_aprovizionare, pret\_achizitie, id\_medicament)

VALUES (DATE '2025-08-31', DATE '2025-01-05', 80, 60, 45, DATE '2025-02-20', 30, 3);

INSERT INTO STOC (data\_expirare, data\_fabricatie, nr\_bucati\_primite, nr\_bucati\_ramase, pret\_vanzare\_curent, data\_aprovizionare, pret\_achizitie, id\_medicament)

VALUES (DATE '2025-11-30', DATE '2025-03-01', 60, 40, 65, DATE '2025-03-05', 50, 4);

INSERT INTO STOC (data\_expirare, data\_fabricatie, nr\_bucati\_primite, nr\_bucati\_ramase, pret\_vanzare\_curent, data\_aprovizionare, pret\_achizitie, id\_medicament)

VALUES (DATE '2026-01-31', DATE '2025-02-20', 40, 25, 40, DATE '2025-03-10', 27, 5);

INSERT INTO STOC (data\_expirare, data\_fabricatie, nr\_bucati\_primite, nr\_bucati\_ramase, pret\_vanzare\_curent, data\_aprovizionare, pret\_achizitie, id\_medicament)

VALUES (DATE '2026-10-31', DATE '2025-04-10', 70, 55, 60, DATE '2025-04-20', 45, 6);

INSERT INTO STOC (data\_expirare, data\_fabricatie, nr\_bucati\_primite, nr\_bucati\_ramase, pret\_vanzare\_curent, data\_aprovizionare, pret\_achizitie, id\_medicament)

VALUES (DATE '2026-09-30', DATE '2025-03-15', 30, 22, 95, DATE '2025-04-01', 75, 7);

INSERT INTO STOC (data\_expirare, data\_fabricatie, nr\_bucati\_primite, nr\_bucati\_ramase, pret\_vanzare\_curent, data\_aprovizionare, pret\_achizitie, id\_medicament)

VALUES (DATE '2027-03-31', DATE '2025-05-05', 90, 90, 85, DATE '2025-05-10', 62, 8);

INSERT INTO STOC (data\_expirare, data\_fabricatie, nr\_bucati\_primite, nr\_bucati\_ramase, pret\_vanzare\_curent, data\_aprovizionare, pret\_achizitie, id\_medicament)

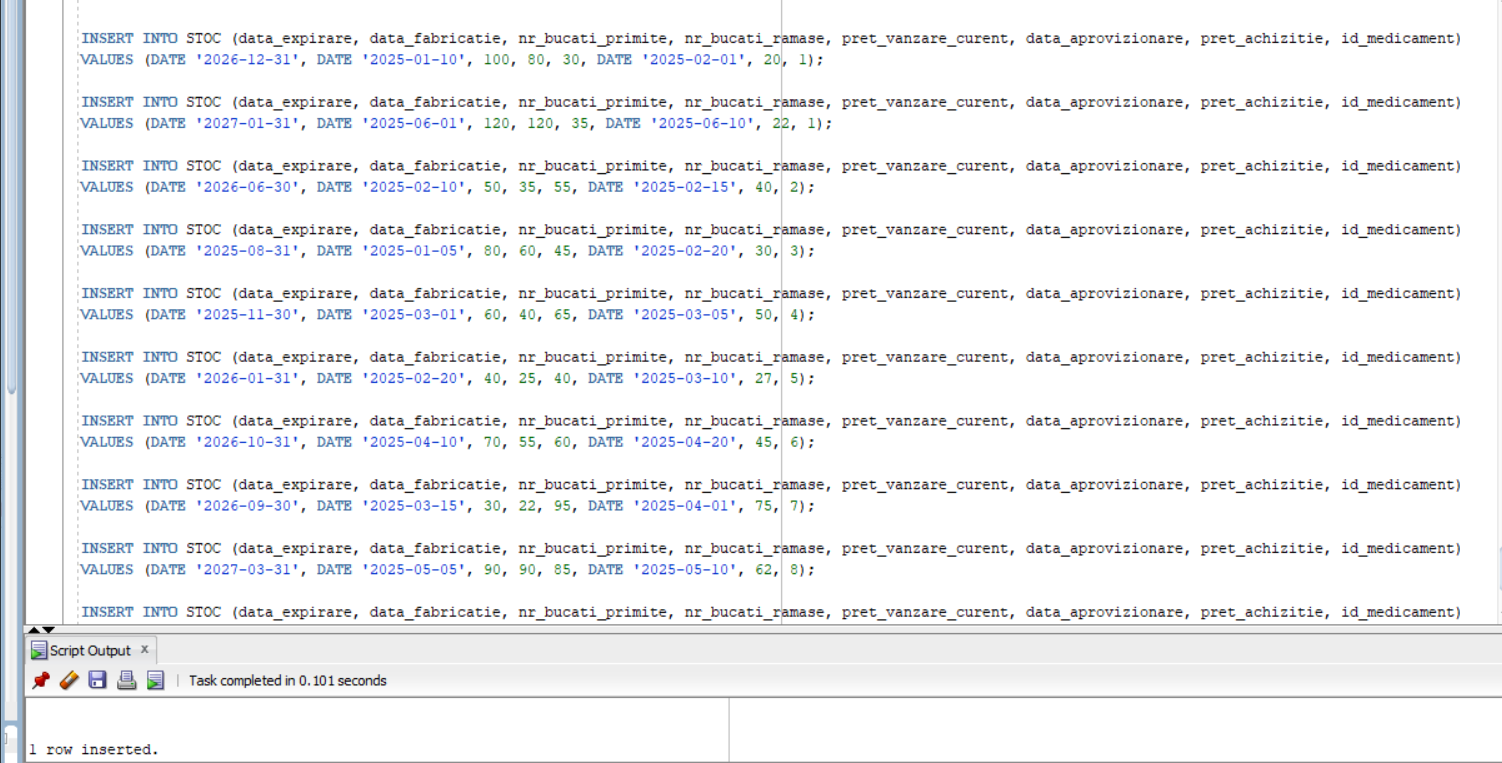
VALUES (DATE '2026-04-30', DATE '2025-02-01', 60, 48, 25, DATE '2025-02-12', 15, 9);

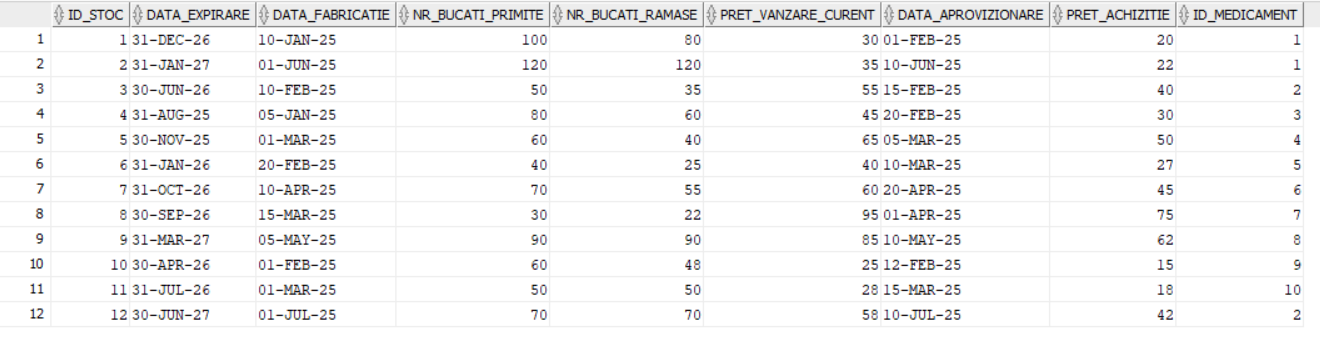
INSERT INTO STOC (data\_expirare, data\_fabricatie, nr\_bucati\_primite, nr\_bucati\_ramase, pret\_vanzare\_curent, data\_aprovizionare, pret\_achizitie, id\_medicament)

VALUES (DATE '2026-07-31', DATE '2025-03-01', 50, 50, 28, DATE '2025-03-15', 18, 10);

INSERT INTO STOC (data\_expirare, data\_fabricatie, nr\_bucati\_primite, nr\_bucati\_ramase, pret\_vanzare\_curent, data\_aprovizionare, pret\_achizitie, id\_medicament)

VALUES (DATE '2027-06-30', DATE '2025-07-01', 70, 70, 58, DATE '2025-07-10', 42, 2);





INSERT INTO CONSULTATIE (pret, id\_animal, id\_personal\_medical) VALUES (150,1,1);

INSERT INTO CONSULTATIE (pret, id\_animal, id\_personal\_medical) VALUES (200,2,2);

INSERT INTO CONSULTATIE (pret, id\_animal, id\_personal\_medical) VALUES (180,3,1);

INSERT INTO CONSULTATIE (pret, id\_animal, id\_personal\_medical) VALUES (220,4,5);

INSERT INTO CONSULTATIE (pret, id\_animal, id\_personal\_medical) VALUES (170,5,2);

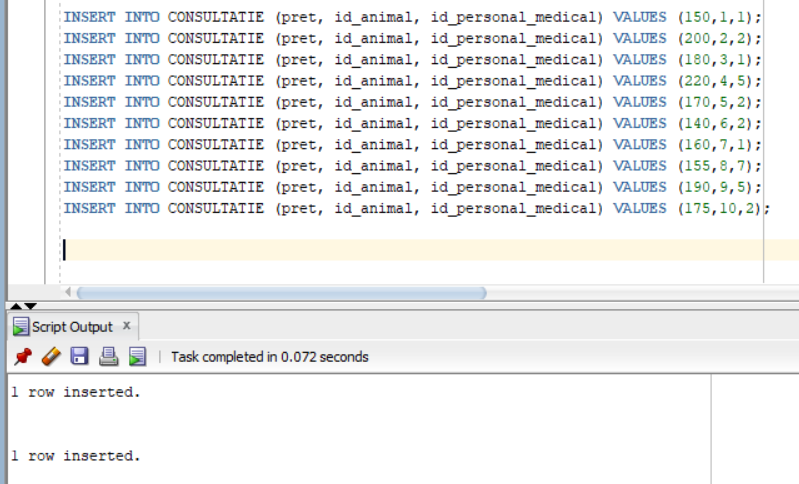
INSERT INTO CONSULTATIE (pret, id\_animal, id\_personal\_medical) VALUES (140,6,2);

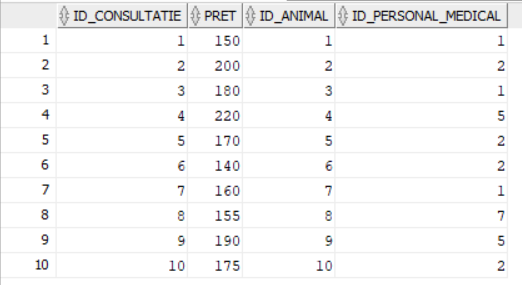
INSERT INTO CONSULTATIE (pret, id\_animal, id\_personal\_medical) VALUES (160,7,1);

INSERT INTO CONSULTATIE (pret, id\_animal, id\_personal\_medical) VALUES (155,8,7);

INSERT INTO CONSULTATIE (pret, id\_animal, id\_personal\_medical) VALUES (190,9,5);

INSERT INTO CONSULTATIE (pret, id\_animal, id\_personal\_medical) VALUES (175,10,2);





INSERT INTO RETETA (data\_emitere, id\_consultatie, id\_personal\_medical) VALUES (DATE '2025-03-10',1,1);

INSERT INTO RETETA (data\_emitere, id\_consultatie, id\_personal\_medical) VALUES (DATE '2025-03-12',2,2);

INSERT INTO RETETA (data\_emitere, id\_consultatie, id\_personal\_medical) VALUES (DATE '2025-03-15',3,1);

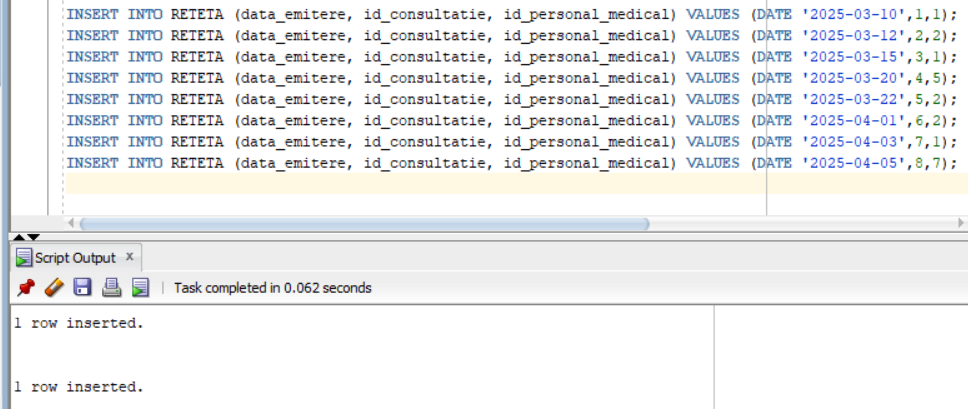
INSERT INTO RETETA (data\_emitere, id\_consultatie, id\_personal\_medical) VALUES (DATE '2025-03-20',4,5);

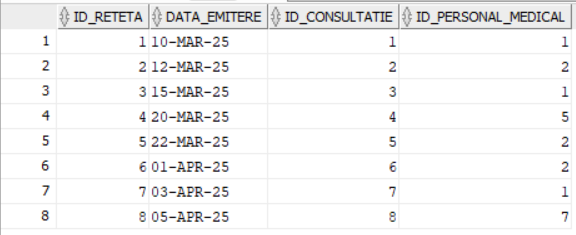
INSERT INTO RETETA (data\_emitere, id\_consultatie, id\_personal\_medical) VALUES (DATE '2025-03-22',5,2);

INSERT INTO RETETA (data\_emitere, id\_consultatie, id\_personal\_medical) VALUES (DATE '2025-04-01',6,2);

INSERT INTO RETETA (data\_emitere, id\_consultatie, id\_personal\_medical) VALUES (DATE '2025-04-03',7,1);

INSERT INTO RETETA (data\_emitere, id\_consultatie, id\_personal\_medical) VALUES (DATE '2025-04-05',8,7);





INSERT INTO COMANDA (data\_comanda, id\_personal\_medical) VALUES (DATE '2025-04-01',3);

INSERT INTO COMANDA (data\_comanda, id\_personal\_medical) VALUES (DATE '2025-04-02',4);

INSERT INTO COMANDA (data\_comanda, id\_personal\_medical) VALUES (DATE '2025-04-03',6);

INSERT INTO COMANDA (data\_comanda, id\_personal\_medical) VALUES (DATE '2025-04-04',3);

INSERT INTO COMANDA (data\_comanda, id\_personal\_medical) VALUES (DATE '2025-04-05',10);

INSERT INTO COMANDA (data\_comanda, id\_personal\_medical) VALUES (DATE '2025-04-06',4);

INSERT INTO COMANDA (data\_comanda, id\_personal\_medical) VALUES (DATE '2025-04-07',6);

INSERT INTO COMANDA (data\_comanda, id\_personal\_medical) VALUES (DATE '2025-04-08',3);

INSERT INTO COMANDA (data\_comanda, id\_personal\_medical) VALUES (DATE '2025-04-09',10);

INSERT INTO COMANDA (data\_comanda, id\_personal\_medical) VALUES (DATE '2025-04-10',4);

INSERT INTO COMANDA (data\_comanda, id\_personal\_medical) VALUES (DATE '2025-04-11',6);

INSERT INTO COMANDA (data\_comanda, id\_personal\_medical) VALUES (DATE '2025-04-12',3);

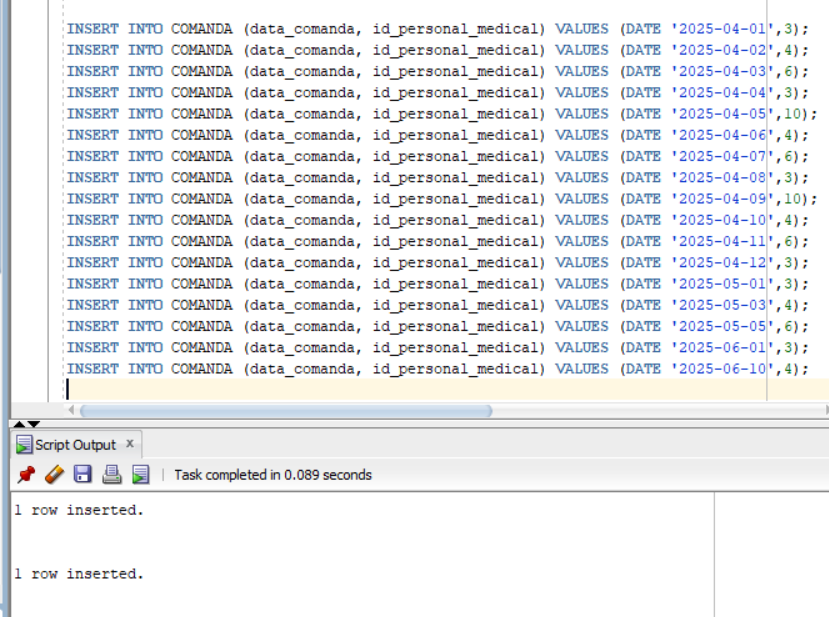
INSERT INTO COMANDA (data\_comanda, id\_personal\_medical) VALUES (DATE '2025-05-01',3);

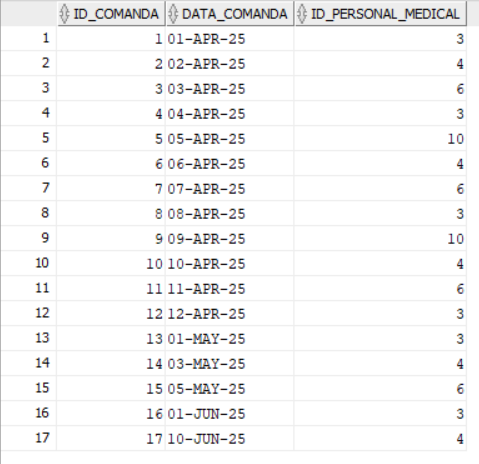
INSERT INTO COMANDA (data\_comanda, id\_personal\_medical) VALUES (DATE '2025-05-03',4);

INSERT INTO COMANDA (data\_comanda, id\_personal\_medical) VALUES (DATE '2025-05-05',6);

INSERT INTO COMANDA (data\_comanda, id\_personal\_medical) VALUES (DATE '2025-06-01',3);

INSERT INTO COMANDA (data\_comanda, id\_personal\_medical) VALUES (DATE '2025-06-10',4);





INSERT INTO COMANDA\_CLIENT VALUES (1,'card',1);

INSERT INTO COMANDA\_CLIENT VALUES (2,'cash',2);

INSERT INTO COMANDA\_CLIENT VALUES (3,'card',3);

INSERT INTO COMANDA\_CLIENT VALUES (4,'cash',4);

INSERT INTO COMANDA\_CLIENT VALUES (5,'card',5);

INSERT INTO COMANDA\_CLIENT VALUES (6,'cash',6);

INSERT INTO COMANDA\_CLIENT VALUES (7,'card',7);

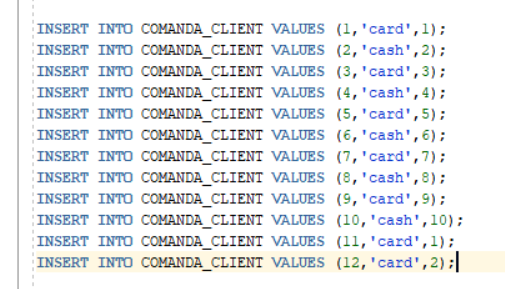
INSERT INTO COMANDA\_CLIENT VALUES (8,'cash',8);

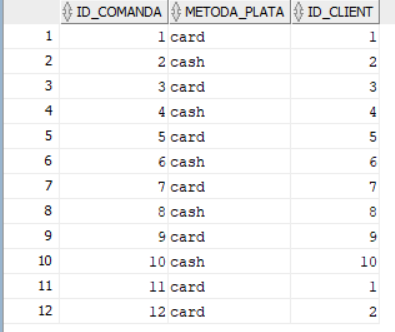
INSERT INTO COMANDA\_CLIENT VALUES (9,'card',9);

INSERT INTO COMANDA\_CLIENT VALUES (10,'cash',10);

INSERT INTO COMANDA\_CLIENT VALUES (11,'card',1);

INSERT INTO COMANDA\_CLIENT VALUES (12,'card',2);





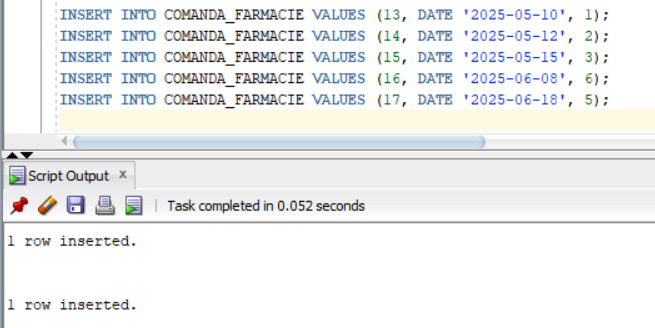
INSERT INTO COMANDA\_FARMACIE VALUES (13, DATE '2025-05-10', 1);

INSERT INTO COMANDA\_FARMACIE VALUES (14, DATE '2025-05-12', 2);

INSERT INTO COMANDA\_FARMACIE VALUES (15, DATE '2025-05-15', 3);

INSERT INTO COMANDA\_FARMACIE VALUES (16, DATE '2025-06-08', 6);

INSERT INTO COMANDA\_FARMACIE VALUES (17, DATE '2025-06-18', 5);





INSERT INTO FACTURA (data\_emitere, suma, id\_comanda) VALUES (DATE '2025-04-01',120,1);

INSERT INTO FACTURA (data\_emitere, suma, id\_comanda) VALUES (DATE '2025-04-02', 90,2);

INSERT INTO FACTURA (data\_emitere, suma, id\_comanda) VALUES (DATE '2025-04-03',140,3);

INSERT INTO FACTURA (data\_emitere, suma, id\_comanda) VALUES (DATE '2025-04-04',160,4);

INSERT INTO FACTURA (data\_emitere, suma, id\_comanda) VALUES (DATE '2025-04-05',130,5);

INSERT INTO FACTURA (data\_emitere, suma, id\_comanda) VALUES (DATE '2025-04-06',100,6);

INSERT INTO FACTURA (data\_emitere, suma, id\_comanda) VALUES (DATE '2025-04-07',115,7);

INSERT INTO FACTURA (data\_emitere, suma, id\_comanda) VALUES (DATE '2025-04-08',125,8);

INSERT INTO FACTURA (data\_emitere, suma, id\_comanda) VALUES (DATE '2025-04-09',140,9);

INSERT INTO FACTURA (data\_emitere, suma, id\_comanda) VALUES (DATE '2025-04-10',110,10);

INSERT INTO FACTURA (data\_emitere, suma, id\_comanda) VALUES (DATE '2025-04-11', 85,11);

INSERT INTO FACTURA (data\_emitere, suma, id\_comanda) VALUES (DATE '2025-04-12', 95,12);

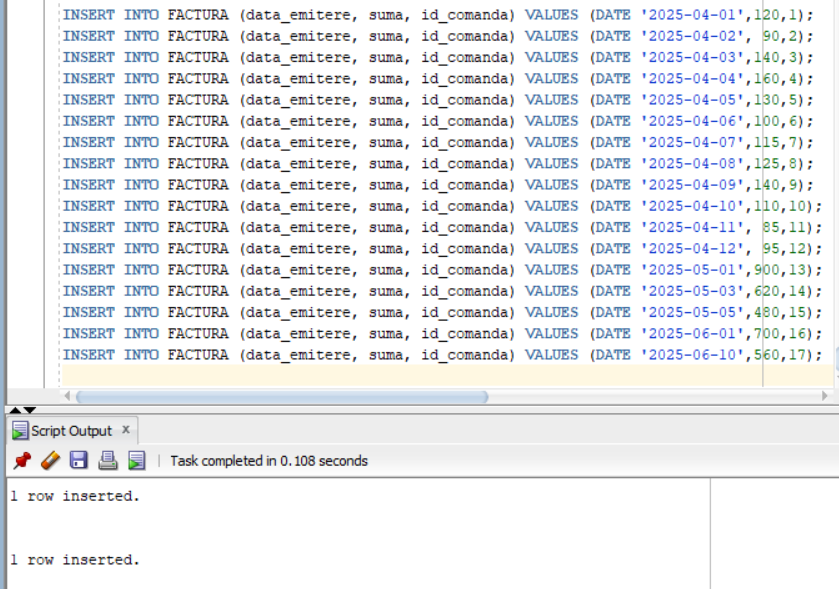
INSERT INTO FACTURA (data\_emitere, suma, id\_comanda) VALUES (DATE '2025-05-01',900,13);

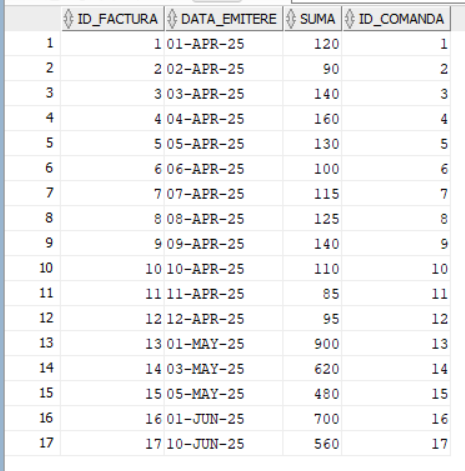
INSERT INTO FACTURA (data\_emitere, suma, id\_comanda) VALUES (DATE '2025-05-03',620,14);

INSERT INTO FACTURA (data\_emitere, suma, id\_comanda) VALUES (DATE '2025-05-05',480,15);

INSERT INTO FACTURA (data\_emitere, suma, id\_comanda) VALUES (DATE '2025-06-01',700,16);

INSERT INTO FACTURA (data\_emitere, suma, id\_comanda) VALUES (DATE '2025-06-10',560,17);





**TABELELE ASOCIATIVE**

INSERT INTO VINDE VALUES (1,1,20);

INSERT INTO VINDE VALUES (1,10,18);

INSERT INTO VINDE VALUES (1,5,27);

INSERT INTO VINDE VALUES (2,2,40);

INSERT INTO VINDE VALUES (2,8,62);

INSERT INTO VINDE VALUES (2,9,15);

INSERT INTO VINDE VALUES (3,6,45);

INSERT INTO VINDE VALUES (3,7,75);

INSERT INTO VINDE VALUES (3,4,50);

INSERT INTO VINDE VALUES (4,3,30);

INSERT INTO VINDE VALUES (4,2,42);

INSERT INTO VINDE VALUES (4,1,22);

INSERT INTO VINDE VALUES (5,4,48);

INSERT INTO VINDE VALUES (5,3,33);

INSERT INTO VINDE VALUES (5,6,44);

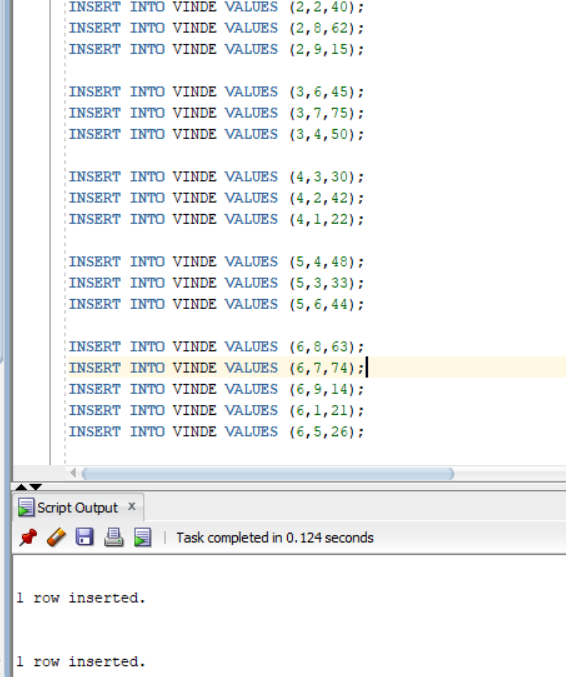
INSERT INTO VINDE VALUES (6,8,63);

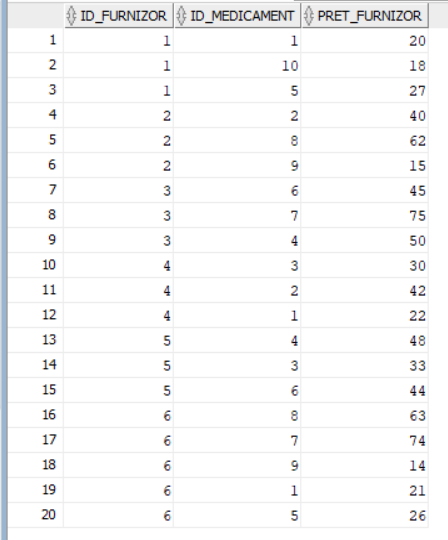
INSERT INTO VINDE VALUES (6,7,74);

INSERT INTO VINDE VALUES (6,9,14);

INSERT INTO VINDE VALUES (6,1,21);

INSERT INTO VINDE VALUES (6,5,26);





INSERT INTO CONTINE (id\_reteta, id\_medicament, dozaj, mod\_administrare, frecventa, durata\_tratament)

VALUES (1, 1, '1 cp', 'oral', '2/zi', '7 zile');

INSERT INTO CONTINE (id\_reteta, id\_medicament, dozaj, mod\_administrare, frecventa, durata\_tratament)

VALUES (1, 2, '1 cp', 'oral', '1/zi', '5 zile');

INSERT INTO CONTINE (id\_reteta, id\_medicament, dozaj, mod\_administrare, frecventa, durata\_tratament)

VALUES (2, 3, '1 pipeta', 'topic', 'o data', '1 zi');

INSERT INTO CONTINE (id\_reteta, id\_medicament, dozaj, mod\_administrare, frecventa, durata\_tratament)

VALUES (2, 10, 'aplicare locala', 'topic', '2/zi', '7 zile');

INSERT INTO CONTINE (id\_reteta, id\_medicament, dozaj, mod\_administrare, frecventa, durata\_tratament)

VALUES (2, 9, '1 cp', 'oral', '1/zi', '5 zile');

INSERT INTO CONTINE (id\_reteta, id\_medicament, dozaj, mod\_administrare, frecventa, durata\_tratament)

VALUES (3, 6, '1 cp', 'oral', '2/zi', '7 zile');

INSERT INTO CONTINE (id\_reteta, id\_medicament, dozaj, mod\_administrare, frecventa, durata\_tratament)

VALUES (3, 5, '0.5 ml', 'injectabil', 'o data', '1 zi');

INSERT INTO CONTINE (id\_reteta, id\_medicament, dozaj, mod\_administrare, frecventa, durata\_tratament)

VALUES (4, 7, '1 cp', 'oral', '1/zi', '3 zile');

INSERT INTO CONTINE (id\_reteta, id\_medicament, dozaj, mod\_administrare, frecventa, durata\_tratament)

VALUES (4, 2, '1 cp', 'oral', '1/zi', '5 zile');

INSERT INTO CONTINE (id\_reteta, id\_medicament, dozaj, mod\_administrare, frecventa, durata\_tratament)

VALUES (5, 8, '1 cp', 'oral', '2/zi', '7 zile');

INSERT INTO CONTINE (id\_reteta, id\_medicament, dozaj, mod\_administrare, frecventa, durata\_tratament)

VALUES (5, 1, '1 cp', 'oral', '2/zi', '5 zile');

INSERT INTO CONTINE (id\_reteta, id\_medicament, dozaj, mod\_administrare, frecventa, durata\_tratament)

VALUES (6, 4, '1 ml', 'injectabil', 'o data', '1 zi');

INSERT INTO CONTINE (id\_reteta, id\_medicament, dozaj, mod\_administrare, frecventa, durata\_tratament)

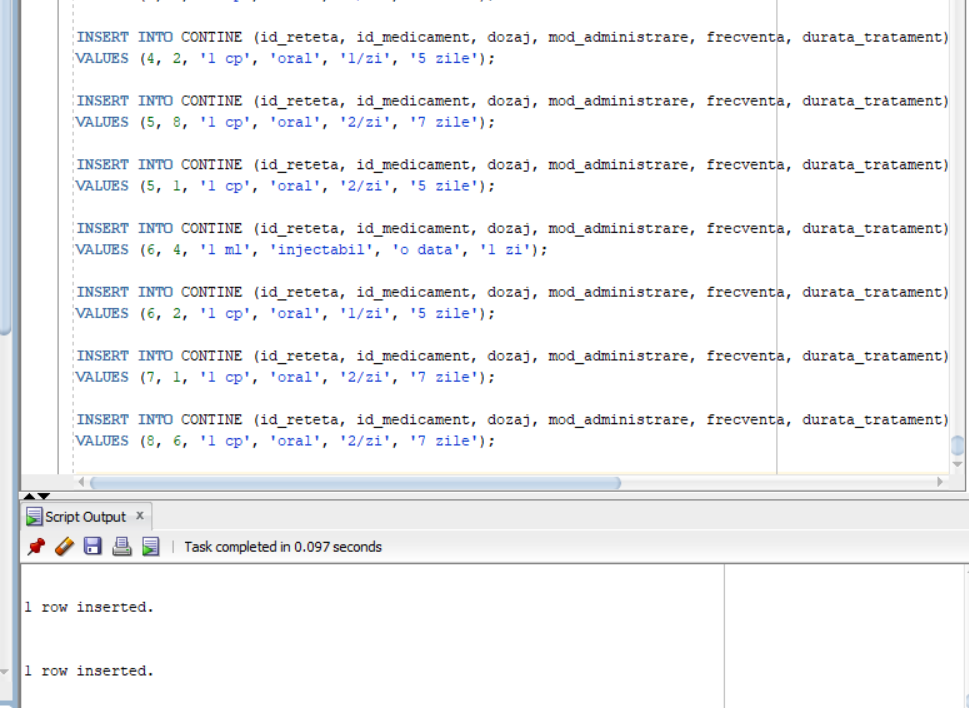
VALUES (6, 2, '1 cp', 'oral', '1/zi', '5 zile');

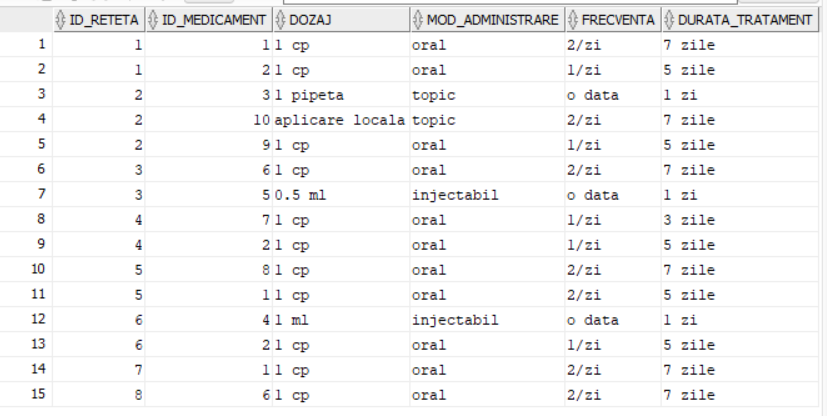
INSERT INTO CONTINE (id\_reteta, id\_medicament, dozaj, mod\_administrare, frecventa, durata\_tratament)

VALUES (7, 1, '1 cp', 'oral', '2/zi', '7 zile');

INSERT INTO CONTINE (id\_reteta, id\_medicament, dozaj, mod\_administrare, frecventa, durata\_tratament)

VALUES (8, 6, '1 cp', 'oral', '2/zi', '7 zile');





INSERT INTO ARE (id\_comanda, id\_stoc, cantitate, pret\_vanzare\_final, discount)

VALUES (1, 1, 2, 30, 0);

INSERT INTO ARE (id\_comanda, id\_stoc, cantitate, pret\_vanzare\_final, discount)

VALUES (2, 4, 1, 45, 0);

INSERT INTO ARE (id\_comanda, id\_stoc, cantitate, pret\_vanzare\_final, discount)

VALUES (2, 6, 1, 40, 5);

INSERT INTO ARE (id\_comanda, id\_stoc, cantitate, pret\_vanzare\_final, discount)

VALUES (3, 7, 1, 60, 0);

INSERT INTO ARE (id\_comanda, id\_stoc, cantitate, pret\_vanzare\_final, discount)

VALUES (3, 10, 2, 25, 10);

INSERT INTO ARE (id\_comanda, id\_stoc, cantitate, pret\_vanzare\_final, discount)

VALUES (3, 3, 1, 55, 0);

INSERT INTO ARE (id\_comanda, id\_stoc, cantitate, pret\_vanzare\_final, discount)

VALUES (4, 8, 1, 95, 0);

INSERT INTO ARE (id\_comanda, id\_stoc, cantitate, pret\_vanzare\_final, discount)

VALUES (4, 11, 1, 28, 0);

INSERT INTO ARE (id\_comanda, id\_stoc, cantitate, pret\_vanzare\_final, discount)

VALUES (4, 2, 1, 35, 5);

INSERT INTO ARE (id\_comanda, id\_stoc, cantitate, pret\_vanzare\_final, discount)

VALUES (4, 5, 1, 65, 0);

INSERT INTO ARE (id\_comanda, id\_stoc, cantitate, pret\_vanzare\_final, discount)

VALUES (5, 9, 1, 85, 0);

INSERT INTO ARE (id\_comanda, id\_stoc, cantitate, pret\_vanzare\_final, discount)

VALUES (5, 1, 1, 30, 0);

INSERT INTO ARE (id\_comanda, id\_stoc, cantitate, pret\_vanzare\_final, discount)

VALUES (6, 12, 1, 58, 0);

INSERT INTO ARE (id\_comanda, id\_stoc, cantitate, pret\_vanzare\_final, discount)

VALUES (7, 4, 1, 45, 0);

INSERT INTO ARE (id\_comanda, id\_stoc, cantitate, pret\_vanzare\_final, discount)

VALUES (7, 10, 1, 25, 0);

INSERT INTO ARE (id\_comanda, id\_stoc, cantitate, pret\_vanzare\_final, discount)

VALUES (7, 6, 1, 40, 5);

INSERT INTO ARE (id\_comanda, id\_stoc, cantitate, pret\_vanzare\_final, discount)

VALUES (8, 3, 1, 55, 0);

INSERT INTO ARE (id\_comanda, id\_stoc, cantitate, pret\_vanzare\_final, discount)

VALUES (9, 7, 1, 60, 0);

INSERT INTO ARE (id\_comanda, id\_stoc, cantitate, pret\_vanzare\_final, discount)

VALUES (9, 6, 1, 40, 0);

INSERT INTO ARE (id\_comanda, id\_stoc, cantitate, pret\_vanzare\_final, discount)

VALUES (10, 11, 1, 28, 0);

INSERT INTO ARE (id\_comanda, id\_stoc, cantitate, pret\_vanzare\_final, discount)

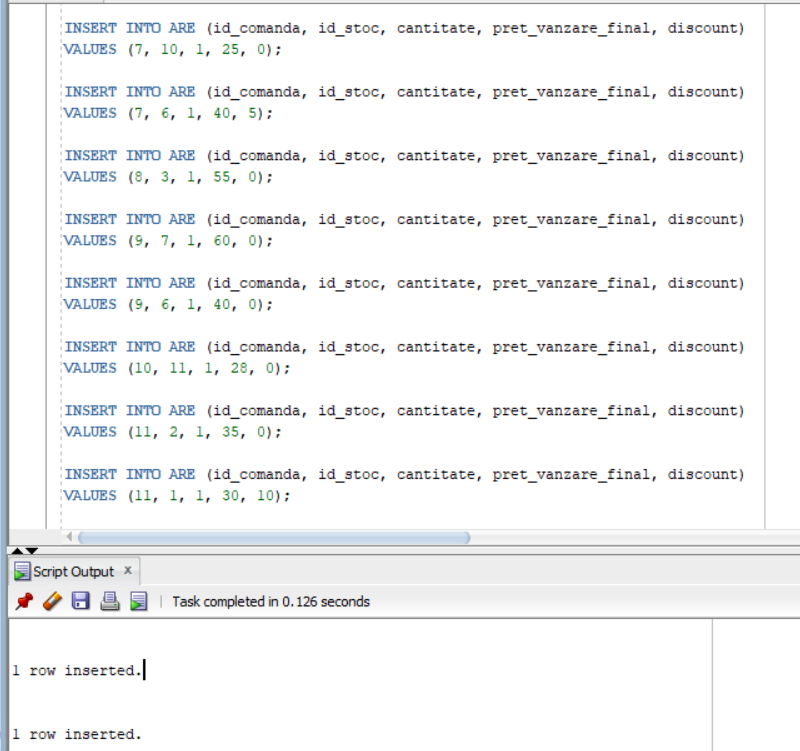
VALUES (11, 2, 1, 35, 0);

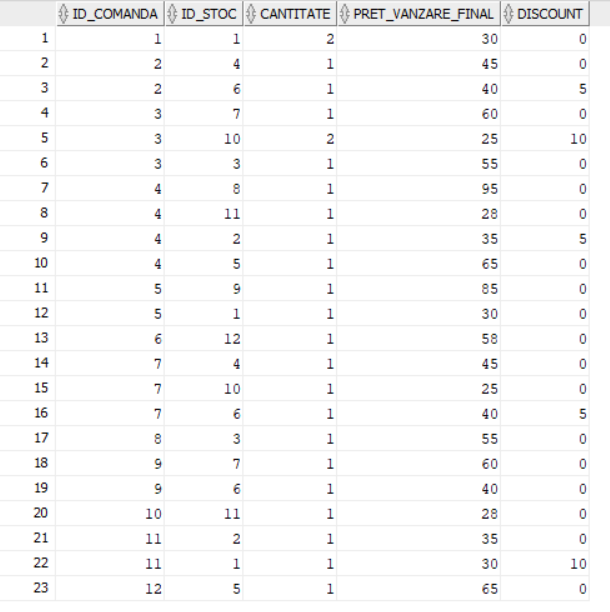
INSERT INTO ARE (id\_comanda, id\_stoc, cantitate, pret\_vanzare\_final, discount)

VALUES (11, 1, 1, 30, 10);

INSERT INTO ARE (id\_comanda, id\_stoc, cantitate, pret\_vanzare\_final, discount)

VALUES (12, 5, 1, 65, 0);





INSERT INTO INTERVINE (id\_animal, id\_campanie, id\_personal\_medical, data\_interventiei, observatii)

VALUES (1, 1, 1, DATE '2025-01-20', 'Vaccinare efectuata');

INSERT INTO INTERVINE (id\_animal, id\_campanie, id\_personal\_medical, data\_interventiei, observatii)

VALUES (2, 1, 2, DATE '2025-01-22', 'Vaccinare efectuata');

INSERT INTO INTERVINE (id\_animal, id\_campanie, id\_personal\_medical, data\_interventiei, observatii)

VALUES (3, 2, 1, DATE '2025-03-10', 'Deparazitare interna');

INSERT INTO INTERVINE (id\_animal, id\_campanie, id\_personal\_medical, data\_interventiei, observatii)

VALUES (4, 2, 2, DATE '2025-03-12', 'Deparazitare externa');

INSERT INTO INTERVINE (id\_animal, id\_campanie, id\_personal\_medical, data\_interventiei, observatii)

VALUES (5, 3, 1, DATE '2025-04-05', 'Programare sterilizare');

INSERT INTO INTERVINE (id\_animal, id\_campanie, id\_personal\_medical, data\_interventiei, observatii)

VALUES (6, 4, 5, DATE '2025-05-08', 'Control efectuat');

INSERT INTO INTERVINE (id\_animal, id\_campanie, id\_personal\_medical, data\_interventiei, observatii)

VALUES (7, 4, 5, DATE '2025-05-10', 'Control efectuat');

INSERT INTO INTERVINE (id\_animal, id\_campanie, id\_personal\_medical, data\_interventiei, observatii)

VALUES (8, 5, 2, DATE '2025-09-15', 'Vaccinare polivalenta');

INSERT INTO INTERVINE (id\_animal, id\_campanie, id\_personal\_medical, data\_interventiei, observatii)

VALUES (9, 5, 1, DATE '2025-09-20', 'Vaccinare polivalenta');

INSERT INTO INTERVINE (id\_animal, id\_campanie, id\_personal\_medical, data\_interventiei, observatii)

VALUES (10, 6, 7, DATE '2025-06-12', 'Tratament antiparazitar');

INSERT INTO INTERVINE (id\_animal, id\_campanie, id\_personal\_medical, data\_interventiei, observatii)

VALUES (11, 6, 7, DATE '2025-06-18', 'Preventie');

INSERT INTO INTERVINE (id\_animal, id\_campanie, id\_personal\_medical, data\_interventiei, observatii)

VALUES (12, 7, 2, DATE '2025-07-05', 'Consultatie junior');

INSERT INTO INTERVINE (id\_animal, id\_campanie, id\_personal\_medical, data\_interventiei, observatii)

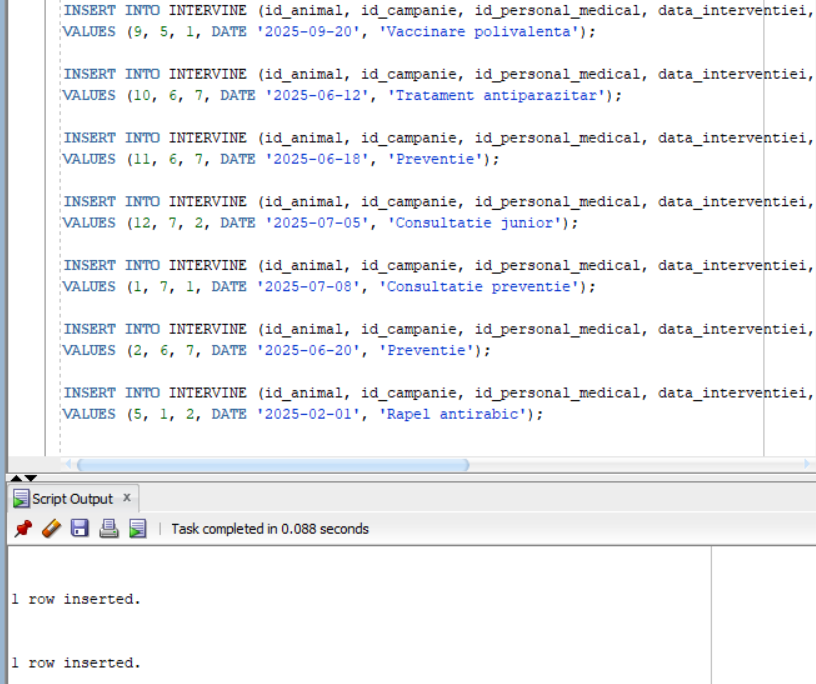
VALUES (1, 7, 1, DATE '2025-07-08', 'Consultatie preventie');

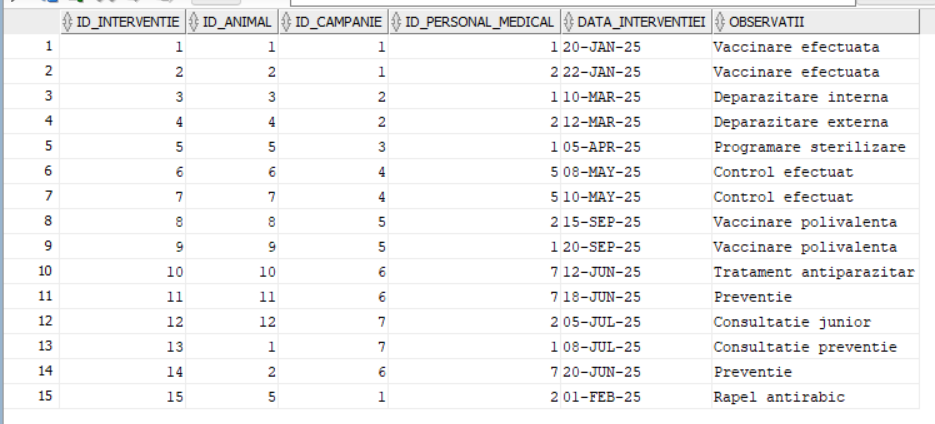
INSERT INTO INTERVINE (id\_animal, id\_campanie, id\_personal\_medical, data\_interventiei, observatii)

VALUES (2, 6, 7, DATE '2025-06-20', 'Preventie');

INSERT INTO INTERVINE (id\_animal, id\_campanie, id\_personal\_medical, data\_interventiei, observatii)

VALUES (5, 1, 2, DATE '2025-02-01', 'Rapel antirabic');





INSERT INTO INCLUDE VALUES (13,1,60);

INSERT INTO INCLUDE VALUES (13,10,40);

INSERT INTO INCLUDE VALUES (13,5,30);

INSERT INTO INCLUDE VALUES (14,2,50);

INSERT INTO INCLUDE VALUES (14,8,25);

INSERT INTO INCLUDE VALUES (14,9,80);

INSERT INTO INCLUDE VALUES (15,6,40);

INSERT INTO INCLUDE VALUES (15,7,20);

INSERT INTO INCLUDE VALUES (15,4,35);

INSERT INTO INCLUDE VALUES (16,8,30);

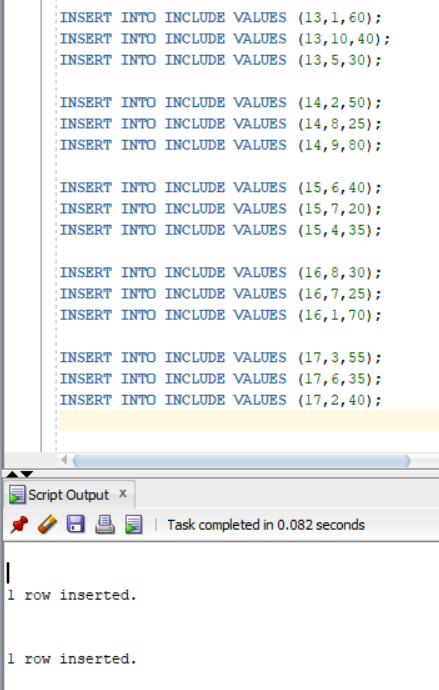
INSERT INTO INCLUDE VALUES (16,7,25);

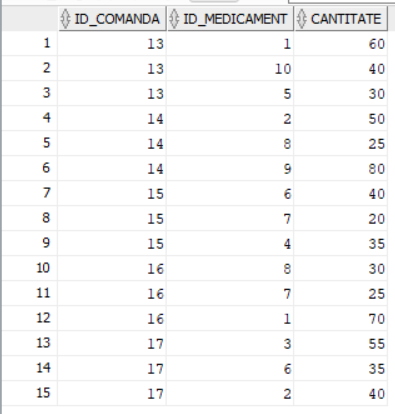
INSERT INTO INCLUDE VALUES (16,1,70);

INSERT INTO INCLUDE VALUES (17,3,55);

INSERT INTO INCLUDE VALUES (17,6,35);

INSERT INTO INCLUDE VALUES (17,2,40);





# Subprogram ce utilizează trei tipuri de colecții

**Ex.6 - cerință:**

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.

**Ex.6 – rezolvare:**

Pentru un tip de campanie primit ca **parametru** (de exemplu „Vaccinare” sau „Preventie”), se vor determina toate campaniile din baza de date care au acest tip și pentru care **există cel puțin o intervenție înregistrată**. Pentru fiecare campanie determinată, se vor afișa **denumirea campaniei** și **informații detaliate despre animalele participante**, respectiv **lista animalelor** pentru care s-au efectuat intervenții în cadrul campaniei. Pentru fiecare animal din listă, se vor afișa **intervențiile realizate** în cadrul campaniei respective, incluzând data intervenției, numele personalului medical (medic veterinar) care a intervenit și observațiile asociate intervenției.

În final, se va afișa pentru **fiecare zi a lunii** (de la 1 la 31) **numărul total de intervenții realizate**, pe baza zilei extrase din data intervenției, afișându-se numai zilele pentru care există cel puțin o intervenție.

Rezolvarea va fi implementată printr-un **subprogram stocat independent (procedură)** care utilizează toate cele trei tipuri de colecții studiate: un **VARRAY** de dimensiune 31 pentru contorizarea intervențiilor pe zile, un **NESTED TABLE** pentru memorarea listei de intervenții aferente fiecărui animal și un **tablou asociativ (INDEX-BY table)** pentru maparea campanie → animale → intervenții.

SET SERVEROUTPUT ON;

CREATE OR REPLACE PROCEDURE ex6\_tip\_campanie(

prm\_tip IN CAMPANIE.tip%TYPE

)

IS

--record

TYPE rec\_interventie IS RECORD(

data\_int INTERVINE.data\_interventiei%TYPE,

medic VARCHAR2(120),

obs INTERVINE.observatii%TYPE,

zi NUMBER(2)

);

--nested table

TYPE t\_interventii IS TABLE OF rec\_interventie;

v\_interventii t\_interventii;

--index by table

TYPE t\_animale IS TABLE OF t\_interventii

INDEX BY VARCHAR2(200);

v\_animale t\_animale;

TYPE t\_campanii IS TABLE OF t\_animale

INDEX BY VARCHAR2(200);

v\_campanii t\_campanii;

--varray

TYPE v\_zile IS VARRAY(31) OF NUMBER;

contor\_zile v\_zile := v\_zile();

gasit BOOLEAN := FALSE;

k\_campanie VARCHAR2(200);

k\_animal VARCHAR2(200);

BEGIN

FOR i IN 1..31 LOOP

contor\_zile.EXTEND;

contor\_zile(i) := 0;

END LOOP;

--campaniile de tipul cerut care au interventii

FOR camp IN(

SELECT DISTINCT c.id\_campanie, c.nume, c.tip, c.data\_start, c.data\_sfarsit

FROM CAMPANIE c

JOIN INTERVINE i ON i.id\_campanie=c.id\_campanie

WHERE LOWER(c.tip) = LOWER(prm\_tip)

ORDER BY c.id\_campanie

)LOOP

gasit := TRUE;

--cheie campanie

k\_campanie := camp.nume || ' [' || camp.tip || '] (' ||

TO\_CHAR(camp.data\_start,'YYYY-MM-DD') || ' - ' ||

TO\_CHAR(camp.data\_sfarsit,'YYYY-MM-DD') || ')';

v\_animale.DELETE;

FOR animl IN(

SELECT DISTINCT a.id\_animal, a.nume, a.specie, a.rasa

FROM ANIMAL a

JOIN INTERVINE i ON i.id\_animal = a.id\_animal

WHERE i.id\_campanie = camp.id\_campanie

ORDER BY a.id\_animal

)LOOP

k\_animal := animl.nume || ' (' || animl.specie || ', ' || animl.rasa || ', ID=' || animl.id\_animal || ')';

--interventiile fiecarui animal

SELECT i.data\_interventiei, pm.prenume || ' ' || pm.nume, i.observatii, EXTRACT(DAY FROM i.data\_interventiei)

BULK COLLECT INTO v\_interventii

FROM INTERVINE i

JOIN PERSONAL\_MEDICAL pm ON pm.id\_personal\_medical = i.id\_personal\_medical

WHERE i.id\_campanie = camp.id\_campanie AND i.id\_animal = animl.id\_animal

ORDER BY i.data\_interventiei;

v\_animale(k\_animal) := v\_interventii;

--actualizam controul de zile

FOR j IN 1..v\_interventii.COUNT LOOP

IF v\_interventii(j).zi BETWEEN 1 AND 31 THEN

contor\_zile(v\_interventii(j).zi) := contor\_zile(v\_interventii(j).zi)+1;

END IF;

END LOOP;

END LOOP;

v\_campanii(k\_campanie) := v\_animale;

END LOOP;

IF NOT gasit THEN

DBMS\_OUTPUT.PUT\_LINE('Nu exista campanii de tipul: ' || prm\_tip || ' sau nu exista interventii pentru acestea.');

RETURN;

END IF;

--afisare

k\_campanie := v\_campanii.FIRST;

WHILE k\_campanie IS NOT NULL LOOP

DBMS\_OUTPUT.PUT\_LINE('Campanie: ' || k\_campanie);

v\_animale := v\_campanii(k\_campanie);

k\_animal := v\_animale.FIRST;

WHILE k\_animal IS NOT NULL LOOP

DBMS\_OUTPUT.PUT\_LINE(' Animal: ' || k\_animal);

v\_interventii := v\_animale(k\_animal);

FOR j IN 1..v\_interventii.COUNT LOOP

DBMS\_OUTPUT.PUT\_LINE(

' - ' || TO\_CHAR(v\_interventii(j).data\_int,'YYYY-MM-DD') ||

' | ' || v\_interventii(j).medic ||

' | ' || NVL(v\_interventii(j).obs,'(fara observatii)')

);

END LOOP;

k\_animal := v\_animale.NEXT(k\_animal);

END LOOP;

DBMS\_OUTPUT.NEW\_LINE;

k\_campanie := v\_campanii.NEXT(k\_campanie);

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('- Total interventii pe zile (1..31) -');

FOR i IN 1..31 LOOP

IF contor\_zile(i)=1 THEN

DBMS\_OUTPUT.PUT\_LINE('Ziua ' || i || ': o interventie.');

ELSIF contor\_zile(i)>1 THEN

DBMS\_OUTPUT.PUT\_LINE('Ziua ' || i || ': ' || contor\_zile(i) || ' interventii.');

END IF;

END LOOP;

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Eroare: ' || SQLERRM);

END ex6\_tip\_campanie;

/

**--APEL**

BEGIN

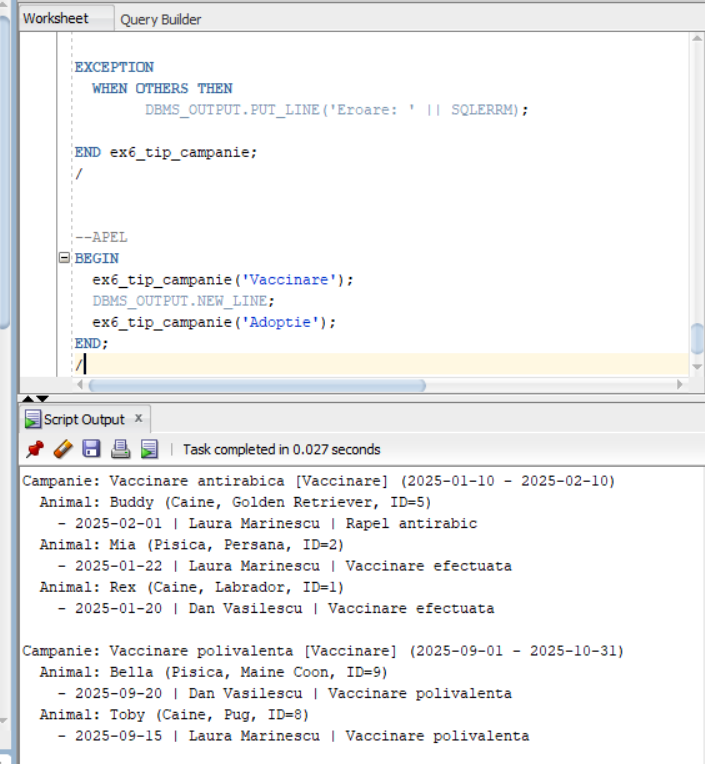
ex6\_tip\_campanie('Vaccinare');

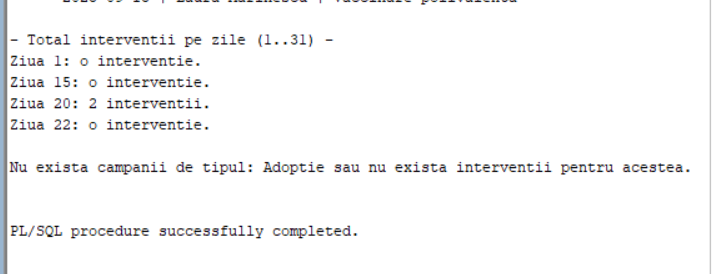
DBMS\_OUTPUT.NEW\_LINE;

ex6\_tip\_campanie('Adoptie');

END;

/





# Subprogram ce utilizează două tipuri de cursoare

**Ex.7 - cerință:**

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.

**Ex.7 – rezolvare:**

Pentru un **interval de date** primit ca parametru, se vor afișa toate **comenzile clienților** înregistrate în acel interval. Pentru fiecare comandă se vor afișa **identificatorul comenzii, data comenzii, numele complet al clientului și metoda de plată utilizată**. În continuare, pentru fiecare comandă se va afișa **lista medicamentelor cumpărate** în cadrul acesteia, incluzând **denumirea medicamentului, cantitatea, prețul final și discount-ul aplicat**. Lista medicamentelor aferente fiecărei comenzi va fi determinată utilizând un cursor parametrizat dependent de cursorul principal al comenzilor.

CREATE OR REPLACE PROCEDURE ex7\_comenzi(

p\_data\_start IN DATE,

p\_data\_sfarsit IN DATE

)

IS

--exceptie personalizata

ex\_fara\_comenzi EXCEPTION;

nr\_comenzi NUMBER := 0;

--cursor clasic(explicit)

CURSOR c\_detalii(p\_id\_comanda COMANDA.id\_comanda%TYPE) IS

SELECT m.denumire, a.cantitate, a.pret\_vanzare\_final, a.discount

FROM ARE a

JOIN STOC s ON s.id\_stoc = a.id\_stoc

JOIN MEDICAMENT m ON m.id\_medicament = s.id\_medicament

WHERE a.id\_comanda = p\_id\_comanda

ORDER BY m.denumire;

detalii\_comanda c\_detalii%ROWTYPE;

BEGIN

--ciclu cursor cu subcerere

FOR comanda IN (

SELECT c.id\_comanda, c.data\_comanda, cl.prenume || ' ' || cl.nume AS client, cc.metoda\_plata

FROM COMANDA c

JOIN COMANDA\_CLIENT cc ON cc.id\_comanda = c.id\_comanda

JOIN CLIENT cl ON cl.id\_client = cc.id\_client

WHERE c.data\_comanda BETWEEN p\_data\_start AND p\_data\_sfarsit

ORDER BY c.id\_comanda

)LOOP

nr\_comenzi := nr\_comenzi+1;

DBMS\_OUTPUT.PUT\_LINE('Comanda #' || comanda.id\_comanda || ' (' || TO\_CHAR(comanda.data\_comanda,'YYYY-MM-DD') || ')');

DBMS\_OUTPUT.PUT\_LINE('Client: ' || comanda.client);

DBMS\_OUTPUT.PUT\_LINE('Metoda plata: ' || comanda.metoda\_plata);

DBMS\_OUTPUT.PUT\_LINE('Produse (medicamente):');

OPEN c\_detalii(comanda.id\_comanda);

FETCH c\_detalii INTO detalii\_comanda;

IF c\_detalii%FOUND THEN

LOOP

DBMS\_OUTPUT.PUT\_LINE(

' - ' || detalii\_comanda.denumire ||

' | cant=' || detalii\_comanda.cantitate ||

' | pret\_final=' || detalii\_comanda.pret\_vanzare\_final ||

' | disc=' || detalii\_comanda.discount || '%'

);

FETCH c\_detalii INTO detalii\_comanda;

EXIT WHEN c\_detalii%NOTFOUND;

END LOOP;

ELSE

DBMS\_OUTPUT.PUT\_LINE(' (nu exista medicamente asociate acestei comenzi)');

END IF;

CLOSE c\_detalii;

DBMS\_OUTPUT.NEW\_LINE;

END LOOP;

IF nr\_comenzi = 0 THEN

RAISE ex\_fara\_comenzi;

ELSIF nr\_comenzi = 1 THEN

DBMS\_OUTPUT.PUT\_LINE('Au fost afisate informatii despre o singura comanda.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Au fost afisate informatii despre ' || nr\_comenzi || ' comenzi.');

END IF;

EXCEPTION

WHEN ex\_fara\_comenzi THEN

DBMS\_OUTPUT.PUT\_LINE('Nu au fost gasite comenzi in intervalul specificat.');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Eroare: ' || SQLERRM);

END ex7\_comenzi;

/

**--APEL**

BEGIN

--nu sunt comenzi

ex7\_comenzi(DATE '2024-01-01', DATE '2024-01-31');

DBMS\_OUTPUT.NEW\_LINE;

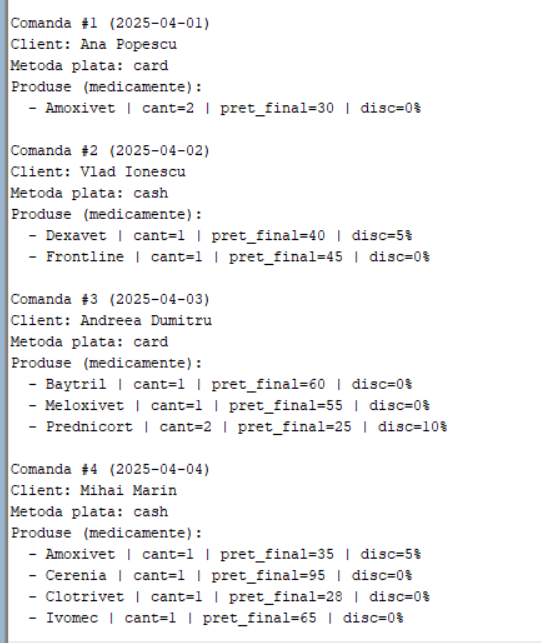
--sunt comenzi

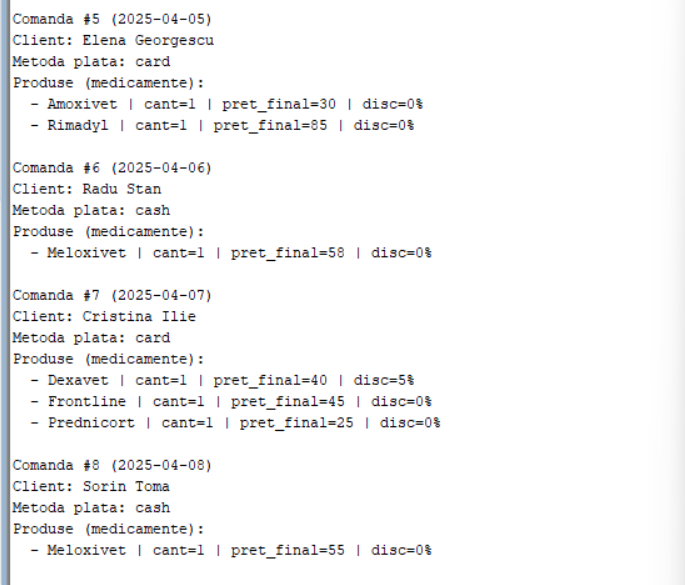
ex7\_comenzi(DATE '2025-04-01', DATE '2025-04-30');

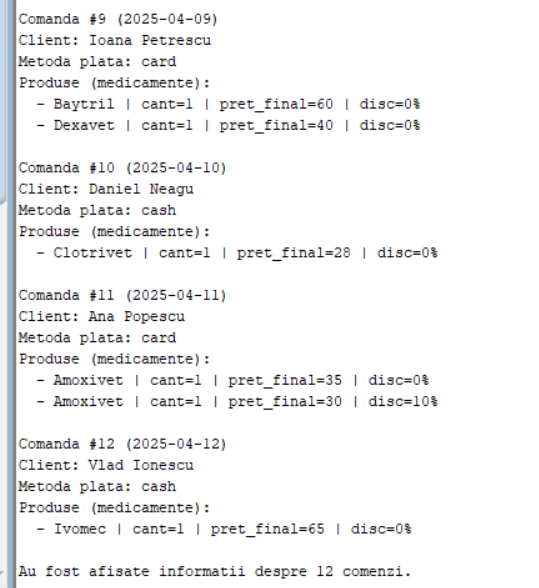
END;

/









# Funcție care utilizează trei tabele și tratează toate excepțiile

**Ex.8 - cerință:**

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.

**Ex.8 – rezolvare:**

Pentru un client identificat prin **nume și prenume**, primite ca parametri, se va determina **metoda de plată asociată comenzilor** acestuia, precum și **informații suplimentare despre activitatea clientului**. Se va implementa un subprogram stocat independent de tip funcție care utilizează, într-o singură comandă SQL, trei dintre tabelele create în baza de date: **CLIENT, COMANDA\_CLIENT și COMANDA.** Funcția va returna **metoda de plată utilizată de client, numărul total de comenzi efectuate și data ultimei comenzi**, doar în situația în care **metoda de plată este unică**.

În cazul în care clientul nu există sau nu are comenzi înregistrate, funcția va trata excepția predefinită **NO\_DATA\_FOUND**. În cazul în care clientul are comenzi cu metode de plată diferite, instrucțiunea SQL va genera excepția **TOO\_MANY\_ROWS**, care va fi tratată corespunzător. Orice altă eroare apărută în timpul execuției va fi tratată în secțiunea **WHEN OTHERS**.

CREATE OR REPLACE FUNCTION ex8\_metoda\_plata(

p\_nume IN CLIENT.nume%TYPE,

p\_prenume IN CLIENT.prenume%TYPE

)

RETURN VARCHAR2

IS

v\_metoda COMANDA\_CLIENT.metoda\_plata%type;

v\_nr\_comenzi NUMBER;

v\_ultima\_data DATE;

BEGIN

SELECT cc.metoda\_plata,

COUNT(DISTINCT co.id\_comanda) AS nr\_comenzi,

MAX(co.data\_comanda) AS ultima\_data

INTO v\_metoda, v\_nr\_comenzi, v\_ultima\_data

FROM CLIENT c

JOIN COMANDA\_CLIENT cc ON cc.id\_client = c.id\_client

JOIN COMANDA co ON co.id\_comanda = cc.id\_comanda

WHERE LOWER(c.nume) = LOWER(p\_nume) AND

LOWER(c.prenume) = LOWER(p\_prenume)

GROUP BY cc.metoda\_plata;

RETURN 'Metoda=' || v\_metoda

|| ', comenzi=' || v\_nr\_comenzi

|| ', ultima=' || TO\_CHAR(v\_ultima\_data, 'YYYY-MM-DD');

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 'NO\_DATA\_FOUND: client inexistent sau fara comenzi';

WHEN TOO\_MANY\_ROWS THEN

RETURN 'TOO\_MANY\_ROWS: clientul are comenzi cu metode de plata diferite';

WHEN OTHERS THEN

RETURN 'ALTA EROARE: ' || SQLERRM;

END;

/

**--APEL**

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Caz normal: ' || ex8\_metoda\_plata('Popescu', 'Ana'));

DBMS\_OUTPUT.NEW\_LINE;

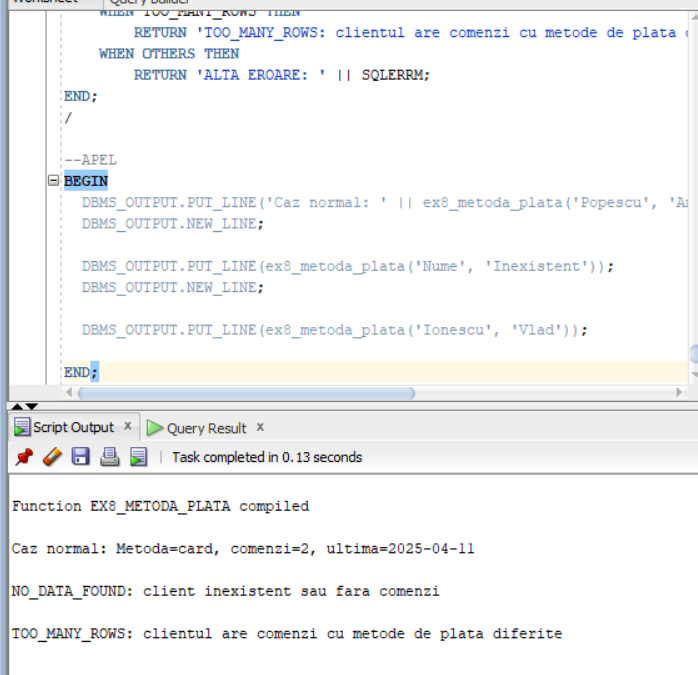
DBMS\_OUTPUT.PUT\_LINE(ex8\_metoda\_plata('Nume', 'Inexistent'));

DBMS\_OUTPUT.NEW\_LINE;

DBMS\_OUTPUT.PUT\_LINE(ex8\_metoda\_plata('Ionescu', 'Vlad'));

END;

/



# Procedura care utilizează cinci tabele și are două excepții personalizate

# 

**Ex.9 - cerință:**

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.

**Ex.9 – rezolvare:**