Centrul de adopție

pentru animalele de companie "Happy Puppy"

Tache Daria Elena

Grupa 234

|  |  |  |
| --- | --- | --- |
|  |  |  |

# *~ Cuprins ~*

1. [Descrierea modelului real, a utilității acestuia și a regulilor de funcționare](#_~_Modelul_real)
2. [Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului](#_~_Prezentarea_constrângerilor)
3. [Descrierea entităților, incluzând precizarea cheii primare.](#_~_Descrierea_entităților)
4. [Descrierea relațiilor, incluzând precizarea cardinalității acestora](#_~_Descrierea_relațiilor)
5. [Descrierea atributelor, incluzând tipul de date si eventualele constrângeri, valori implicite, valori posibile ale tuturor atributelor](#_~_Descrierea_atributelor)
6. [Diagrama entitate-relație](#_~Diagrama_Entitate_–)
7. [Diagrama conceptuală](#_~Diagrama_Conceptuală_~)
8. [Enumerarea schemelor relaționale corespunzătoare diagramei entitate-relație proiectate la punctul 7](#_~Enumerarea_schemelor_relaționale~)
9. [Normalizarea până la forma normală 3 (FN1- FN3)](#_~_Normalizarea_FN1-)
10. [Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).](#_~_Crearea_unei)
11. [Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative).](#_~_Crearea_tabelelor)
12. Formulați în limbaj natural și [implementați 5 cereri SQL complexe](#_~_Cerința_12) ce vor utiliza, în ansamblul lor, următoarele elemente:
    * + subcereri sincronizate în care intervin cel puțin 3 tabele
      + subcereri nesincronizate în clauza FROM
      + grupări de date, funcții grup, filtrare la nivel de grupuri cu subcereri nesincronizate (în clauza de HAVING) in care intervin cel puțin 3 tabele (în cadrul aceleiași cereri)
      + ordonări si utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri)
      + utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE
      + utilizarea a cel puțin 1 bloc de cerere (clauza WITH)

[13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.](#Marcaj în document2)

# *~ Modelul real ~*

Proiectul are ca scop crearea unei baze de date pentru fundația de adopție al animăluțelor de companie “Happy Puppy”, cu cabinete în întreaga țară. În acest fel, suntem mai aproape de persoanele ce vor să ofere micilor prieteni șansa la o viață alături de o familie iubitoare.

Centrele noastre vă oferă mai multe opțiuni: puteți adopta un animăluț, puteți dona pentru micii amici pufoși sau puteți veni cu copiii în baza unei programări la joacă!

Procesul de adopție este mai ușor ca niciodată prin contractul de adopție, iar colegii noștri vă vor fi alături la fiecare pas. La final veți primi carnetul medical al noului vostru animăluț de companie, cu vaccinurile la zi!

# *~ Prezentarea constrângerilor ~*

Fiecare animal de companie pus spre adopție are o fișă unică asociată, care conține date despre acesta, și carnetul medical corespunzător.

Carnetul medical conține date despre starea de sănătate a animalului, și mereu va exista o fișă la care să fie anexat.

Prin contractul de adopție se face legătura dintre un animal pus spre adopție și un client dispus să adopte.

La fiecare centru sunt înscrise un număr maxim de 30 de animale de companie.

Considerăm client o persoană care face cel puțin una dintre acțiunile următoare: adoptă un animăluț, donează către oricare din centrele noastre, sau face o programare pentru o vizită una unul din centre.

Donațiile sunt anonime și sunt făcute în contul unui singur centru. În cazul în care se dorește a se dona mai multor centre, se fac mai multe donații, fiecare în contul centrului dorit.

Înainte de normalizare:

Vizitele se pot face la oricare dintre centrele noastre de Luni până Vineri. Cu o programare puteți alege să vizitați unul sau mai multe centre pe care le specificați când faceți programarea, pe tot parcursul zilei. Programarea se putea face pe unul sau mai multe centre pentru a favoriza adoptarea animăluțelor (Daca un client nu adoptă din primul centru, va avea posibilitatea să meargă și la restul centrelor pentru care a făcut programarea și să adopte de acolo).

După normalizare:

Vizitele se pot face la oricare dintre centrele noastre de Luni până Vineri fie în cadrul unui TUR care are următoarele intervale orare: 1(08:00-13:00) sau 2(17:00- 20:00). Programarea poate fi făcută la centru, prin platforma online, sau la telefon.

Un centru va avea cel puțin doi angajați.

# *~ Descrierea entităților ~*

ANIMAL\_COMPANIE are cheie primară #*petID,* cheile externe *#id\_fișă* (care conține informații despre fiecare animal de companie în parte, alături de fișa medicală corespunzătoare), *#id\_centru* (care reține datele centrului în care a fost repartizat animăluțul) și atributul *dată\_preluare* unde este specificată data la care acesta a fost luat în centru.

FIȘA\_AC are drept cheie primară *#id\_fișă,* atributele *specie, rasă, sex* și este superentitate pentru *CARNET\_MEDICAL.*

CARNET\_MEDICAL are cheia primară moștenită de la *FIȘA\_AC (#id\_fișă)* întrucât este în strânsă relație cu aceasta, și nu este folosită în mod individual în baza noastră de date. Moștenește atributele *specie, rasă și sex,* însă are și atributele sale proprii cum ar fi: *vârsta, vaccinuri, sterilizat, deparazitat* (iar pentru ultimele două se poate alege doar ‘DA’ sau ‘NU’) și *mențiuni.*

CLIENT este tabelul ce stochează date despre clienții noștri. Acesta are cheia primară *id\_client*, și atributele *nume, prenume, telefon, adresa* și *email.*

CENTRU este tabelul care ține evidența locațiilor noastre. Are cheia primară *#id\_centru,* și reține date precum: *adresa, nr\_tel, program* și *nr\_animale\_companie.*

CONTRACT\_ADOPȚIE face legătura dintre un client doritor să adopte și un animăluț dintr-un centru. Are drept cheie primară *#id\_contract*, și chei externe precum *#id\_pet* și *#id\_client.* Singurul său atribut este data\_adopție.

DONAȚIE reține date despre donațiile făcute de către clienți. Are cheia primară *#id\_donație,*  face legătura cu centrul care va beneficia de donație prin cheia externă *#id\_centru* și clientul care a făcut donația prin cheia externă *#id\_client.* Are și atributele *data\_donație* și *suma.*

PROGRAMARE\_VIZITA are drept cheie primară *#id\_prog.* Cheile externe fac legătura cu clientul (*#id\_client)* care a inițiat programarea și turul *(#id\_tur*) la care s-a înscris,iar atributele de sine stătătoare sunt *nr\_pers, data\_prog,* și *interval\_orar.*

TUR este entitatea care face legătura între o programare făcută de un client si centrul la care se ține acel tur. Are ca scop să evite conflictele în baza de date. Cheia primară este *#id\_tur,* iar cea externă este *#id\_centru.* Mai ține minte și atributul *detalii.*

ANGAJAT ține evidența angajaților din fiecare centru. Are drept cheie primară #*id\_angajat*, iar cheia externă provine de la centrul în care lucrează fiecare angajat *(#id\_centru)*. Atributele sale sunt: *nume, prenume, telefon, data\_angajare, ore\_săptămână* și *salariu.*

# *~ Descrierea relațiilor ~*

*CARNET\_MEDICAL* se află în relație de 1 la 1 cu *FIȘA\_AC*, și depinde de aceasta. O *fișă* trebuie să aibă un *carnet medical* asociat, iar un *carnet medical* trebuie să aibă o singură *fișă* căreia să i se poată atașa. ( 1 – 1)

*FIȘA\_AC* este dependentă de *ANIMAL\_COMPANIE* având în vedere că o *fișă* trebuie să fie atașată unui singur animal de companie și nu poate exista altfel, iar pentru fiecare *animal de companie* trebuie făcută o *fișă.* (1 – 1)

*ANIMAL\_COMPANIE* și *CONTRACT\_ADOPȚIE* sunt legate printr-o relație de one-to-one deoarece pentru un *animal de companie* poate fi creat un *contract de adopție*, iar un *contract de adopție* trebuie să fie creat în cazul unei adopții. ( 1(0) - 1 ).

Un *CLIENT* poate să facă unul sau mai multe *contracte de adopție*, în timp ce un *CONTRACT\_ADOPȚIE* trebuie să fie făcut de un singur *client.* ( 1 – M(0) )

Un *CLIENT* poate să facă unul sau mai multe *donații*, în timp ce o *DONAȚIE* trebuie să fie făcut de un singur *client.* ( 1 – M(0) ).

Un *CLIENT* poate face una sau mai multe PROGRAMARI\_VIZITA*,* daro singură PROGRAMARE\_VIZITA poate fi făcută de un singur *client.* A nu se confunda cu faptul ca toți clienții au acest drept. ( M(0) – 1) ).

Se poate face o *PROGRAMARE\_VIZITA* la unul sau mai multe CENTRE iar un CENTRU poate găzdui una sau mai multe PROGRAMARI\_VIZITA. ( M(0) - M(0) ).

Acest lucru se va rezolva prin adăugarea entității *TUR.* Așadar, un *client* va face o *programare\_vizita* pentru un anumit *tur* organizat de la un anume *centru*.

Așadar, relațiile devin:

O *programare\_vizita* este destinată unui singur TURdar pentru un TUR se pot face una sau mai multe PROGRAMARI\_VIZITA*.* (1 – M(0) ).

Un *TUR* se va ține la un singur *CENTRU,* iar un *centru* poate avea unul sau mai multe tururi. (1 – M(0) ).

Un *ANGAJAT* va lucra la un singur *CENTRU,* iar un *centru* va avea mai mulți angajați. (1 – M(1)).

# *~ Descrierea atributelor ~*

FISA\_AC

* id\_fisa: INT, PRIMARY KEY - identificator unic pentru fișa animalului.
* specie: VARCHAR2(50) - specia animalului.
* rasa: VARCHAR2(100) - rasa animalului.
* sex: CHAR(1), CHECK (sex IN ('M', 'F')) - sexul animalului ('M' sau 'F').

CARNET\_MEDICAL

* id\_fisa: INT, PRIMARY KEY - referință către FISA\_AC(id\_fisa) - CARNET MEDICAL este subentitate pentru FISA\_AC și pentru că sunt în relație de 1-1 nu au nevoie de două chei diferite.
* varsta: INT - vârsta animalului în ani.
* vaccin: VARCHAR2(500) - lista de vaccinuri administrate.
* sterilizat: CHAR(2), CHECK (sterilizat IN ('DA', 'NU')) - dacă animalul este sterilizat ('DA' sau 'NU').
* deparazitat: CHAR(2), CHECK (deparazitat IN ('DA', 'NU')) - dacă animalul este deparazitat ('DA' sau 'NU').
* mentiuni: VARCHAR2(500) - alte mențiuni medicale, un șir de caractere de lungime maxima 500.
* FOREIGN KEY(id\_fisa) REFERENCES FISA\_AC(id\_fisa) ON DELETE CASCADE – pentru că nu poate exista un CARNET\_MEDICAL fără o FISA\_MEDICALA, în cazul în care ștergem o intrare a unui carnet, vrem să stergem și fișa medicală asociată.

CENTRU

* id\_centru: INT, PRIMARY KEY - identificator unic pentru centrul de adopție.
* adresa: VARCHAR2(200) - adresa centrului.
* nr\_tel: VARCHAR2(20) - numărul de telefon al centrului.
* program: VARCHAR2(200) - programul de lucru al centrului.
* nr\_animale\_companie: INT - numărul de animale de companie în centru.

ANIMAL\_COMPANIE

* id\_pet: INT, PRIMARY KEY - identificator unic pentru animalul de companie.
* id\_centru: INT, FOREIGN KEY - referință către CENTRU(id\_centru).
* id\_fisa: INT, FOREIGN KEY - referință către FISA\_AC(id\_fisa), ON DELETE CASCADE.
* data\_preluare: DATE - data la care animalul a fost preluat în centru.

CLIENT

* id\_client: INT, PRIMARY KEY, AUTO\_INCREMENT - identificator unic pentru client.
* nume: VARCHAR2(20) - numele clientului.
* prenume: VARCHAR2(50) - prenumele clientului.
* telefon: VARCHAR2(20) - numărul de telefon al clientului.
* adresa: VARCHAR2(200) - adresa clientului.
* email: VARCHAR2(50) - adresa de email a clientului.

TUR

* id\_tur: INT, PRIMARY KEY
* id\_centru: INT, FOREIGN KEY - referință către CENTRU(id\_centru).
* detalii: VARCHAR2(50) - detalii despre turul prestabilit.

PROGRAMARE\_VIZITA

* id\_prog: INT, PRIMARY KEY - identificator unic pentru programarea vizitei.
* id\_client: INT, FOREIGN KEY - referință către CLIENT(id\_centru).
* id\_tur: INT, FOREIGN KEY - referință către TUR(id\_tur).
* nr\_pers: INT - numărul de persoane care participă la vizită.
* data\_prog: DATE - data programării.
* interval\_orar: CHAR(1) - intervalul pentru turul la care a fost făcută programarea.

DONATIE

* id\_donatie: INT, PRIMARY KEY - identificator unic pentru donație.
* id\_centru: INT, FOREIGN KEY - referință către CENTRU(id\_centru).
* id\_client: INT, FOREIGN KEY - referință către CLIENT(id\_client).
* data\_donatie: DATE - data donației.
* suma: INT - suma donată.

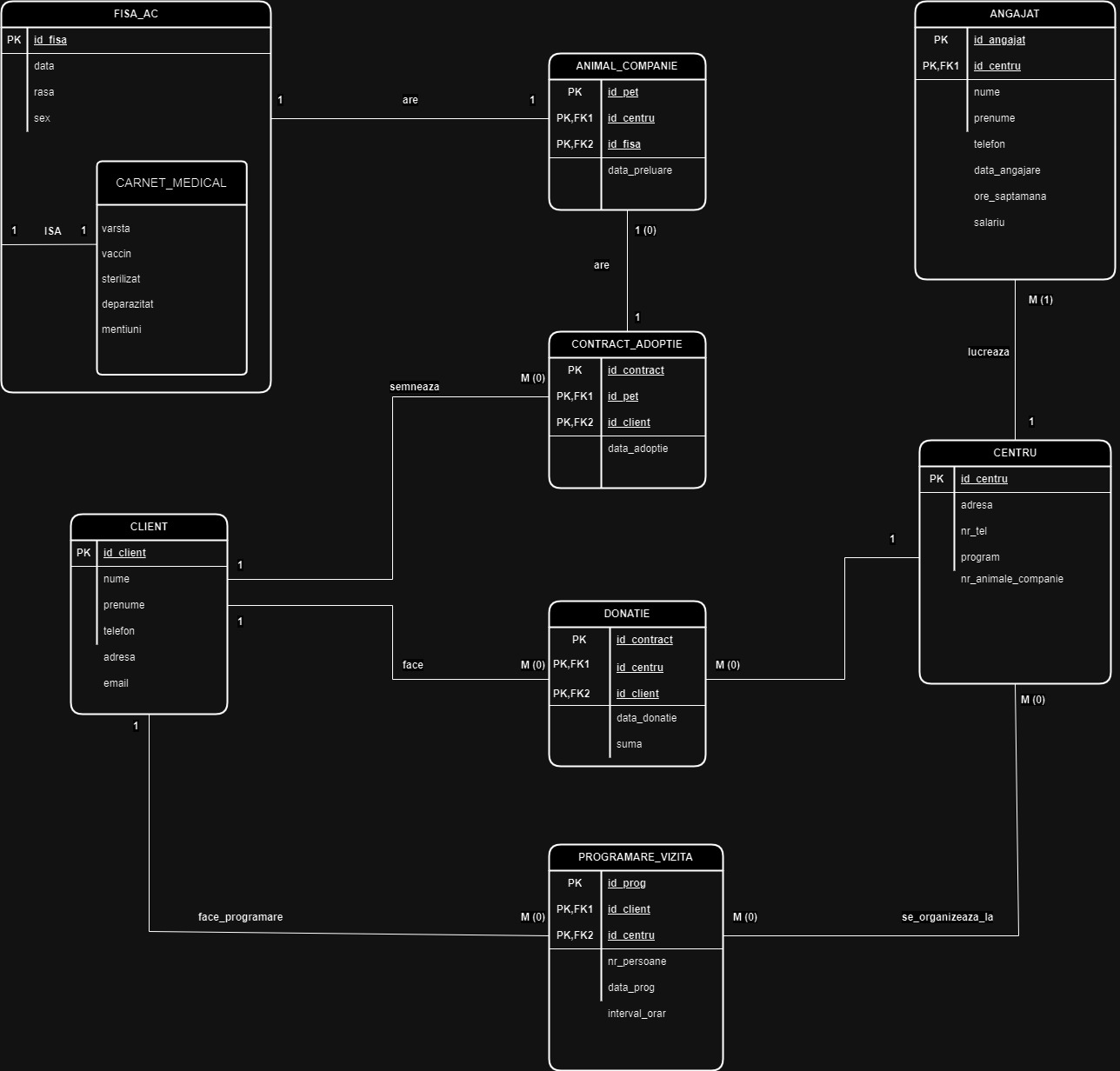
CONTRACT\_ADOPTIE

* id\_contract: INT, PRIMARY KEY - identificator unic pentru contractul de adopție.
* id\_pet: INT, FOREIGN KEY - referință către ANIMAL\_COMPANIE(id\_pet).
* id\_client: INT, FOREIGN KEY - referință către CLIENT(id\_client).
* data\_adoptie: DATE - data adopției.

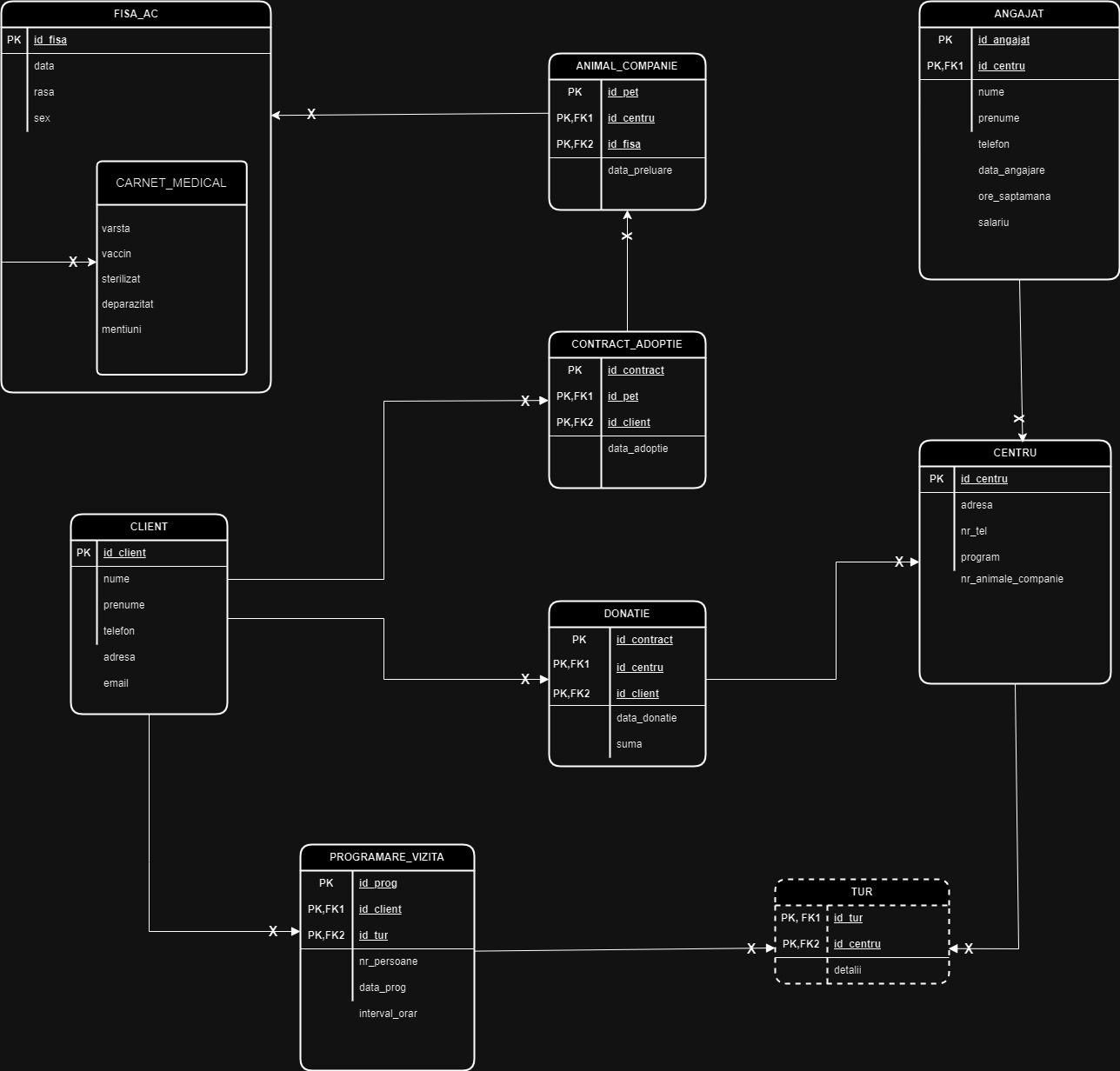
ANGAJAT

* id\_angajat: INT, PRIMARY KEY - identificator unic pentru angajat.
* id\_centru: INT, FOREIGN KEY - referință către CENTRU(id\_centru).
* nume: VARCHAR2(20) - numele angajatului.
* prenume: VARCHAR2(50) - prenumele angajatului.
* telefon: VARCHAR2(20) - numărul de telefon al angajatului.
* data\_angajare: DATE - data angajării.
* ore\_saptamana: INT - numărul de ore lucrate pe săptămână.
* salariu: INT - salariul angajatului.

# *~Diagrama Entitate – Relație ~*



# *~Diagrama Conceptuală ~*



# *~Enumerarea schemelor relaționale~*

FISA\_AC(id\_fisa#, specie, rasa, sex)

CARNET\_MEDICAL(id\_fisa#, varsta, vaccin, sterilizat, deparazitat, mentiuni)

CENTRU(id\_centru#, adresa, nr\_tel, program, nr\_animale\_companie)

ANIMAL\_COMPANIE(id\_pet#, id\_centru#, id\_fisa#, data\_preluare)

CLIENT(id\_client#, nume, prenume, telefon, adresa, email)

TUR(id\_tur#, id\_centru#, detalii)

PROGRAMARE\_VIZITA(id\_prog#, id\_client#, id\_tur#, nr\_pers, data\_prog, interval\_orar)

DONATIE(id\_donatie#, id\_centru#, id\_client#, data\_donatie, suma)

CONTRACT\_ADOPTIE(id\_contract#, id\_pet#, id\_client#, data\_adoptie)

ANGAJAT(id\_angajat#, id\_centru#, nume, prenume, telefon, data\_angajare, ore\_saptamana, salariu)

# *~ Normalizarea FN1- FN3 ~*

Schema inițială îndeplinește FN1 prin faptul că toate coloanele au valori atomice și fiecare tabel are o cheie primară unică.

Forma normala 2 este îndeplinită datorită faptului că toate coloanele non-cheie depind de întreaga cheie primară.

Pentru a fi în FN3, tabelele nu trebuie să aibă dependențe tranzitive. În această schemă toate atributele non-cheie sunt dependente direct de cheia primară, așadar respectă FN3.

Exemplu non-FN1:

Presupunem că am fi stocat in tabela CLIENT un singur string care să rețină toate informațiile pentru un client (nume, prenume, telefon, adresa, email). Acest exemplu încalcă FN1 deoarece valorile nu ar mai fi atomice

Exemplu non-FN2:

În cazul în care am fi avut o cheie primară compusă pentru tabelul DONATIE formată din (‘id\_donatie’, ‘id\_centru’) și am fi reținut și ‘nume\_client’ în acest tabel, ar exista o dependență parțială de ‘id\_client’, dar nu de întreaga cheie primară compusă, încalcând FN2.

Exemplu non-FN3:

Un exemplu bun de încălcare a integrității formei normale 3 ar putea fi următorul: adăugăm ‘adresa\_centru’ în tabelul ANGAJAT. În acest caz, ‘adresa\_centru’ depinde de ‘id\_centru’, care la randul său depinde de cheia primară ‘id\_angajat’.

# *~ Crearea unei secvențe utilizată la inserări în tabele ~*

CREATE SEQUENCE seq\_fisa\_ac START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;

# *~ Crearea tabelelor in SQL și inserarea de date coerente ~*

CREATE TABLE FISA\_AC(

id\_fisa INT PRIMARY KEY,

specie VARCHAR2(50),

rasa VARCHAR2(100),

sex CHAR(1)

);

CREATE TABLE CARNET\_MEDICAL(

id\_fisa INT PRIMARY KEY,

varsta INT,

vaccin VARCHAR2(500),

sterilizat CHAR(2) CHECK (sterilizat IN ('DA', 'NU')),

deparazitat CHAR(2) CHECK (deparazitat IN ('DA', 'NU')),

mentiuni VARCHAR2(500),

FOREIGN KEY(id\_fisa) REFERENCES FISA\_AC(id\_fisa) ON DELETE CASCADE

);

CREATE TABLE CENTRU(

id\_centru INT PRIMARY KEY,

adresa VARCHAR2(200),

nr\_tel VARCHAR2(20),

program VARCHAR2(200),

nr\_animale\_companie INT

);

CREATE TABLE ANIMAL\_COMPANIE(

id\_pet INT PRIMARY KEY,

id\_centru INT,

id\_fisa INT,

data\_preluare DATE,

FOREIGN KEY (id\_centru) REFERENCES CENTRU(id\_centru),

FOREIGN KEY (id\_fisa) REFERENCES FISA\_AC(id\_fisa) ON DELETE CASCADE

);

CREATE TABLE CLIENT(

id\_client INT PRIMARY KEY,

nume VARCHAR2(20),

prenume VARCHAR2(50),

telefon VARCHAR2(20),

adresa VARCHAR2(200),

email VARCHAR2(50)

);

CREATE TABLE TUR(

id\_tur INT PRIMARY KEY,

id\_centru INT,

detalii VARCHAR2(50),

FOREIGN KEY (id\_centru) REFERENCES CENTRU(id\_centru)

);

CREATE TABLE PROGRAMARE\_VIZITA(

id\_prog INT PRIMARY KEY,

id\_client INT,

id\_tur INT,

nr\_pers INT,

data\_prog DATE,

interval\_orar CHAR(1) CHECK (interval\_orar IN ('1', '2')),

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

FOREIGN KEY (id\_tur) REFERENCES TUR(id\_tur)

);

CREATE TABLE DONATIE(

id\_donatie INT PRIMARY KEY,

id\_centru INT,

id\_client INT,

data\_donatie DATE,

suma INT,

FOREIGN KEY(id\_centru) REFERENCES CENTRU(id\_centru),

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

);

CREATE TABLE CONTRACT\_ADOPTIE(

id\_contract INT PRIMARY KEY,

id\_pet INT,

id\_client INT,

data\_adoptie DATE,

FOREIGN KEY (id\_pet) REFERENCES ANIMAL\_COMPANIE(id\_pet),

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

);

CREATE TABLE ANGAJAT(

id\_angajat INT PRIMARY KEY,

id\_centru INT,

nume VARCHAR2(20),

prenume VARCHAR2(50),

telefon VARCHAR2(20),

data\_angajare DATE,

ore\_saptamana INT,

salariu INT,

FOREIGN KEY (id\_centru) REFERENCES CENTRU(id\_centru)

);

---------------------------------------- SEQUENCE SI INSERARI CENTRU ----------------------------------------

CREATE SEQUENCE seq\_centru START WITH 100 INCREMENT BY 1 NOCACHE NOCYCLE;

INSERT INTO CENTRU (id\_centru, adresa, nr\_tel, program, nr\_animale\_companie) VALUES (seq\_centru.NEXTVAL, 'Strada Florilor, Nr. 5', '0712345678', 'Luni-Vineri 9-17', 0);

INSERT INTO CENTRU (id\_centru, adresa, nr\_tel, program, nr\_animale\_companie) VALUES (seq\_centru.NEXTVAL, 'Strada Marilor, Nr. 8', '0722345678', 'Luni-Vineri 9-18', 0);

INSERT INTO CENTRU (id\_centru, adresa, nr\_tel, program, nr\_animale\_companie) VALUES (seq\_centru.NEXTVAL, 'Strada Trandafirilor, Nr. 10', '0732345678', 'Luni-Vineri 8-16', 0);

INSERT INTO CENTRU (id\_centru, adresa, nr\_tel, program, nr\_animale\_companie) VALUES (seq\_centru.NEXTVAL, 'Strada Lalelelor, Nr. 12', '0742345678', 'Luni-Vineri 10-18', 0);

INSERT INTO CENTRU (id\_centru, adresa, nr\_tel, program, nr\_animale\_companie) VALUES (seq\_centru.NEXTVAL, 'Strada Violetei, Nr. 15', '0752345678', 'Luni-Vineri 9-17', 0);

INSERT INTO CENTRU (id\_centru, adresa, nr\_tel, program, nr\_animale\_companie) VALUES (seq\_centru.NEXTVAL, 'Strada Zambilei, Nr. 20', '0762345678', 'Luni-Vineri 8-15', 0);

------------------------------------- SEQUENCE SI INSERARI FISA SI CARNET\_AC ----------------------------------------

CREATE SEQUENCE seq\_fisa\_ac START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;

INSERT INTO FISA\_AC (id\_fisa, specie, rasa, sex) VALUES (seq\_fisa\_ac.NEXTVAL, 'Caine', 'Labrador', 'M');

INSERT INTO FISA\_AC (id\_fisa, specie, rasa, sex) VALUES (seq\_fisa\_ac.NEXTVAL, 'Pisica', 'Siameza', 'F');

INSERT INTO FISA\_AC (id\_fisa, specie, rasa, sex) VALUES (seq\_fisa\_ac.NEXTVAL, 'Caine', 'Beagle', 'M');

INSERT INTO FISA\_AC (id\_fisa, specie, rasa, sex) VALUES (seq\_fisa\_ac.NEXTVAL, 'Pisica', 'Persana', 'F');

INSERT INTO FISA\_AC (id\_fisa, specie, rasa, sex) VALUES (seq\_fisa\_ac.NEXTVAL, 'Caine', 'Bulldog', 'M');

INSERT INTO FISA\_AC (id\_fisa, specie, rasa, sex) VALUES (seq\_fisa\_ac.NEXTVAL, 'Pisica', 'Maine Coon', 'F');

INSERT INTO FISA\_AC (id\_fisa, specie, rasa, sex) VALUES (seq\_fisa\_ac.NEXTVAL, 'Caine', 'Golden Retriever', 'M');

INSERT INTO FISA\_AC (id\_fisa, specie, rasa, sex) VALUES (seq\_fisa\_ac.NEXTVAL, 'Pisica', 'Bengaleza', 'F');

INSERT INTO FISA\_AC (id\_fisa, specie, rasa, sex) VALUES (seq\_fisa\_ac.NEXTVAL, 'Caine', 'Micut', 'F');

INSERT INTO FISA\_AC (id\_fisa, specie, rasa, sex) VALUES (seq\_fisa\_ac.NEXTVAL, 'Pisica', 'Siameza', 'M');

INSERT INTO FISA\_AC (id\_fisa, specie, rasa, sex) VALUES (seq\_fisa\_ac.NEXTVAL, 'Caine', 'Labrador', 'F');

INSERT INTO FISA\_AC (id\_fisa, specie, rasa, sex) VALUES (seq\_fisa\_ac.NEXTVAL, 'Pisica', 'Persana', 'M');

INSERT INTO CARNET\_MEDICAL (id\_fisa, varsta, vaccin, sterilizat, deparazitat, mentiuni) VALUES (1 , 2, 'Rabie', 'DA', 'NU', 'None');

INSERT INTO CARNET\_MEDICAL (id\_fisa, varsta, vaccin, sterilizat, deparazitat, mentiuni) VALUES (2 , 3, 'Parvo', 'NU', 'DA', 'Checkup needed');

INSERT INTO CARNET\_MEDICAL (id\_fisa, varsta, vaccin, sterilizat, deparazitat, mentiuni) VALUES (3 , 1, 'Distemper', 'DA', 'DA', 'Healthy');

INSERT INTO CARNET\_MEDICAL (id\_fisa, varsta, vaccin, sterilizat, deparazitat, mentiuni) VALUES (4 , 4, 'Leptospirosis', 'NU', 'NU', 'Needs vaccination');

INSERT INTO CARNET\_MEDICAL (id\_fisa, varsta, vaccin, sterilizat, deparazitat, mentiuni) VALUES (5 , 2, 'Bordetella', 'DA', 'DA', 'None');

INSERT INTO CARNET\_MEDICAL (id\_fisa, varsta, vaccin, sterilizat, deparazitat, mentiuni) VALUES (6 , 3, 'Rabie', 'NU', 'NU', 'Slightly underweight');

INSERT INTO CARNET\_MEDICAL (id\_fisa, varsta, vaccin, sterilizat, deparazitat, mentiuni) VALUES (7 , 1, 'Parvo', 'DA', 'DA', 'Recovered from illness');

INSERT INTO CARNET\_MEDICAL (id\_fisa, varsta, vaccin, sterilizat, deparazitat, mentiuni) VALUES (8 , 4, 'Distemper', 'DA', 'NU', 'None');

INSERT INTO CARNET\_MEDICAL (id\_fisa, varsta, vaccin, sterilizat, deparazitat, mentiuni) VALUES (9 , 1, 'Distemper', 'DA', 'NU', 'Healthy');

INSERT INTO CARNET\_MEDICAL (id\_fisa, varsta, vaccin, sterilizat, deparazitat, mentiuni) VALUES (10 , 2, 'Leptospirosis', 'NU', 'NU', 'Needs vaccination');

INSERT INTO CARNET\_MEDICAL (id\_fisa, varsta, vaccin, sterilizat, deparazitat, mentiuni) VALUES (11 , 2, 'Bordetella', 'NU', 'DA', 'None');

INSERT INTO CARNET\_MEDICAL (id\_fisa, varsta, vaccin, sterilizat, deparazitat, mentiuni) VALUES (12 , 5, 'Rabie', 'NU', 'NU', 'None');

------------------------------------------SEQUENCE SI INSERARI ANIMAL\_COMPANIE----------------------------------------------

CREATE SEQUENCE seq\_animal\_companie START WITH 1400 INCREMENT BY 1 NOCACHE NOCYCLE;

INSERT INTO ANIMAL\_COMPANIE (id\_pet, id\_centru, id\_fisa, data\_preluare) VALUES (seq\_animal\_companie.NEXTVAL, 100, 1, DATE '2023-01-01');

INSERT INTO ANIMAL\_COMPANIE (id\_pet, id\_centru, id\_fisa, data\_preluare) VALUES (seq\_animal\_companie.NEXTVAL, 101, 2, DATE '2023-02-15');

INSERT INTO ANIMAL\_COMPANIE (id\_pet, id\_centru, id\_fisa, data\_preluare) VALUES (seq\_animal\_companie.NEXTVAL, 100, 3, DATE '2023-03-10');

INSERT INTO ANIMAL\_COMPANIE (id\_pet, id\_centru, id\_fisa, data\_preluare) VALUES (seq\_animal\_companie.NEXTVAL, 103, 4, DATE '2023-04-20');

INSERT INTO ANIMAL\_COMPANIE (id\_pet, id\_centru, id\_fisa, data\_preluare) VALUES (seq\_animal\_companie.NEXTVAL, 104, 5, DATE '2023-05-05');

INSERT INTO ANIMAL\_COMPANIE (id\_pet, id\_centru, id\_fisa, data\_preluare) VALUES (seq\_animal\_companie.NEXTVAL, 100, 6, DATE '2023-06-18');

INSERT INTO ANIMAL\_COMPANIE (id\_pet, id\_centru, id\_fisa, data\_preluare) VALUES (seq\_animal\_companie.NEXTVAL, 105, 7, DATE '2023-07-22');

INSERT INTO ANIMAL\_COMPANIE (id\_pet, id\_centru, id\_fisa, data\_preluare) VALUES (seq\_animal\_companie.NEXTVAL, 100, 8, DATE '2023-08-30');

INSERT INTO ANIMAL\_COMPANIE (id\_pet, id\_centru, id\_fisa, data\_preluare) VALUES (seq\_animal\_companie.NEXTVAL, 102, 9, DATE '2023-05-05');

INSERT INTO ANIMAL\_COMPANIE (id\_pet, id\_centru, id\_fisa, data\_preluare) VALUES (seq\_animal\_companie.NEXTVAL, 103, 10, DATE '2023-06-18');

INSERT INTO ANIMAL\_COMPANIE (id\_pet, id\_centru, id\_fisa, data\_preluare) VALUES (seq\_animal\_companie.NEXTVAL, 104, 11, DATE '2023-07-22');

INSERT INTO ANIMAL\_COMPANIE (id\_pet, id\_centru, id\_fisa, data\_preluare) VALUES (seq\_animal\_companie.NEXTVAL, 101, 12, DATE '2023-08-30');

------------------------------------------ SEQUENCE SI INSERARI CLIENT --------------------------------------------------------------

CREATE SEQUENCE seq\_client START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;

INSERT INTO CLIENT (id\_client, nume, prenume, telefon, adresa, email) VALUES (seq\_client.NEXTVAL, 'Popescu', 'Ion', '0799123456', 'Strada Verde, Nr. 10', '[ion.popescu@example.com](mailto:ion.popescu@example.com)');

INSERT INTO CLIENT (id\_client, nume, prenume, telefon, adresa, email) VALUES (seq\_client.NEXTVAL, 'Ionescu', 'Maria', '0799234567', 'Strada Alba, Nr. 12', '[maria.ionescu@example.com](mailto:maria.ionescu@example.com)');

INSERT INTO CLIENT (id\_client, nume, prenume, telefon, adresa, email) VALUES (seq\_client.NEXTVAL, 'Georgescu', 'Ana', '0799345678', 'Strada Rosu, Nr. 15', '[ana.georgescu@example.com](mailto:ana.georgescu@example.com)');

INSERT INTO CLIENT (id\_client, nume, prenume, telefon, adresa, email) VALUES (seq\_client.NEXTVAL, 'Vasilescu', 'Andrei', '0799456789', 'Strada Galbena, Nr. 18', '[andrei.vasilescu@example.com](mailto:andrei.vasilescu@example.com)');

INSERT INTO CLIENT (id\_client, nume, prenume, telefon, adresa, email) VALUES (seq\_client.NEXTVAL, 'Dumitrescu', 'Elena', '0799567890', 'Strada Alba, Nr. 4' [,'elena.dumitrescu@example.com](mailto:,'elena.dumitrescu@example.com)');

INSERT INTO CLIENT (id\_client, nume, prenume, telefon, adresa, email) VALUES (seq\_client.NEXTVAL, 'Marinescu', 'Cristina', '0799678901', 'Strada Albastra, Nr. 25', '[cristina.marinescu@example.com](mailto:cristina.marinescu@example.com)');

INSERT INTO CLIENT (id\_client, nume, prenume, telefon, adresa, email) VALUES (seq\_client.NEXTVAL, 'Stoicescu', 'Dan', '0799789012', 'Strada Mov, Nr. 30', '[dan.stoicescu@example.com](mailto:dan.stoicescu@example.com)');

INSERT INTO CLIENT (id\_client, nume, prenume, telefon, adresa, email) VALUES (seq\_client.NEXTVAL, 'Nica', 'Gabriela', '0799890123', 'Strada Roz, Nr. 35', '[gabriela.nica@example.com](mailto:gabriela.nica@example.com)');

----------------------------------------------------- SEQUENCE SI INSERARI TUR ------------------------------------------------------------

CREATE SEQUENCE seq\_tur START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;

INSERT INTO TUR (id\_tur, id\_centru) VALUES (seq\_tur.NEXTVAL, 100);

INSERT INTO TUR (id\_tur, id\_centru, detalii) VALUES (seq\_tur.NEXTVAL, 101, 'Detalii');

INSERT INTO TUR (id\_tur, id\_centru) VALUES (seq\_tur.NEXTVAL, 102);

INSERT INTO TUR (id\_tur, id\_centru, detalii) VALUES (seq\_tur.NEXTVAL, 103, 'Joaca');

INSERT INTO TUR (id\_tur, id\_centru) VALUES (seq\_tur.NEXTVAL, 104);

INSERT INTO TUR (id\_tur, id\_centru) VALUES (seq\_tur.NEXTVAL, 105);

INSERT INTO TUR (id\_tur, id\_centru, detalii) VALUES (seq\_tur.NEXTVAL, 102, 'Vizualizare');

INSERT INTO TUR (id\_tur, id\_centru) VALUES (seq\_tur.NEXTVAL, 100);

--------------------------------------------- SEQUENCE SI INSERARI PROGRAMARE\_VIZITA ---------------------------------------------------

CREATE SEQUENCE seq\_programare\_vizita START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;

INSERT INTO PROGRAMARE\_VIZITA (id\_prog, id\_client, id\_tur, nr\_pers, data\_prog, interval\_orar) VALUES (seq\_programare\_vizita.NEXTVAL, 1, 1, 2, DATE '2023-01-05', '1');

INSERT INTO PROGRAMARE\_VIZITA (id\_prog, id\_client, id\_tur, nr\_pers, data\_prog, interval\_orar) VALUES (seq\_programare\_vizita.NEXTVAL, 2, 1, 3, DATE '2023-02-20', '2');

INSERT INTO PROGRAMARE\_VIZITA (id\_prog, id\_client, id\_tur, nr\_pers, data\_prog, interval\_orar) VALUES (seq\_programare\_vizita.NEXTVAL, 3, 4, 1, DATE '2023-03-15', '1');

INSERT INTO PROGRAMARE\_VIZITA (id\_prog, id\_client, id\_tur, nr\_pers, data\_prog, interval\_orar) VALUES (seq\_programare\_vizita.NEXTVAL, 4, 5, 4, DATE '2023-04-25', '2');

INSERT INTO PROGRAMARE\_VIZITA (id\_prog, id\_client, id\_tur, nr\_pers, data\_prog, interval\_orar) VALUES (seq\_programare\_vizita.NEXTVAL, 5, 6, 2, DATE '2023-05-10', '1');

INSERT INTO PROGRAMARE\_VIZITA (id\_prog, id\_client, id\_tur, nr\_pers, data\_prog, interval\_orar) VALUES (seq\_programare\_vizita.NEXTVAL, 6, 7, 1, DATE '2023-06-20', '2');

INSERT INTO PROGRAMARE\_VIZITA (id\_prog, id\_client, id\_tur, nr\_pers, data\_prog, interval\_orar) VALUES (seq\_programare\_vizita.NEXTVAL, 7, 2, 3, DATE '2023-07-25', '1');

INSERT INTO PROGRAMARE\_VIZITA (id\_prog, id\_client, id\_tur, nr\_pers, data\_prog, interval\_orar) VALUES (seq\_programare\_vizita.NEXTVAL, 8, 1, 2, DATE '2022-02-15', '1');

INSERT INTO PROGRAMARE\_VIZITA (id\_prog, id\_client, id\_tur, nr\_pers, data\_prog, interval\_orar) VALUES (seq\_programare\_vizita.NEXTVAL, 4, 1, 3, DATE '2023-08-05', '2');

INSERT INTO PROGRAMARE\_VIZITA (id\_prog, id\_client, id\_tur, nr\_pers, data\_prog, interval\_orar) VALUES (seq\_programare\_vizita.NEXTVAL, 3, 8, 2, DATE '2023-02-21', '2');

INSERT INTO PROGRAMARE\_VIZITA (id\_prog, id\_client, id\_tur, nr\_pers, data\_prog, interval\_orar) VALUES (seq\_programare\_vizita.NEXTVAL, 2, 3, 5, DATE '2023-03-11', '1');

INSERT INTO PROGRAMARE\_VIZITA (id\_prog, id\_client, id\_tur, nr\_pers, data\_prog, interval\_orar) VALUES (seq\_programare\_vizita.NEXTVAL, 6, 2, 1, DATE '2023-11-05', '2');

------------------------------------------ SEQUENCE SI INSERARI DONATIE -------------------------------------------------

CREATE SEQUENCE seq\_donatie START WITH 400 INCREMENT BY 1 NOCACHE NOCYCLE;

INSERT INTO DONATIE (id\_donatie, id\_centru, id\_client, data\_donatie, suma) VALUES (seq\_donatie.NEXTVAL, 100, 1, DATE '2023-01-10', 500);

INSERT INTO DONATIE (id\_donatie, id\_centru, id\_client, data\_donatie, suma) VALUES (seq\_donatie.NEXTVAL, 101, 2, DATE '2023-02-25', 300);

INSERT INTO DONATIE (id\_donatie, id\_centru, id\_client, data\_donatie, suma) VALUES (seq\_donatie.NEXTVAL, 102, 3, DATE '2023-03-20', 200);

INSERT INTO DONATIE (id\_donatie, id\_centru, id\_client, data\_donatie, suma) VALUES (seq\_donatie.NEXTVAL, 100, 4, DATE '2023-04-30', 400);

INSERT INTO DONATIE (id\_donatie, id\_centru, id\_client, data\_donatie, suma) VALUES (seq\_donatie.NEXTVAL, 104, 5, DATE '2023-05-15', 150);

INSERT INTO DONATIE (id\_donatie, id\_centru, id\_client, data\_donatie, suma) VALUES (seq\_donatie.NEXTVAL, 100, 6, DATE '2023-06-25', 350);

INSERT INTO DONATIE (id\_donatie, id\_centru, id\_client, data\_donatie, suma) VALUES (seq\_donatie.NEXTVAL, 104, 5, DATE '2022-03-08', 100);

------------------------------------------ SEQUENCE SI INSERARI CONTRACT\_ADOPTIE ------------------------------------------

CREATE SEQUENCE seq\_contract\_adoptie START WITH 1 INCREMENT BY 1 NOCACHE NOCYCLE;

INSERT INTO CONTRACT\_ADOPTIE (id\_contract, id\_pet, id\_client, data\_adoptie) VALUES (seq\_contract\_adoptie.NEXTVAL, 1400, 1, DATE '2023-01-15');

INSERT INTO CONTRACT\_ADOPTIE (id\_contract, id\_pet, id\_client, data\_adoptie) VALUES (seq\_contract\_adoptie.NEXTVAL, 1402, 2, DATE '2023-02-28');

INSERT INTO CONTRACT\_ADOPTIE (id\_contract, id\_pet, id\_client, data\_adoptie) VALUES (seq\_contract\_adoptie.NEXTVAL, 1403, 3, DATE '2023-03-25');

INSERT INTO CONTRACT\_ADOPTIE (id\_contract, id\_pet, id\_client, data\_adoptie) VALUES (seq\_contract\_adoptie.NEXTVAL, 1404, 4, DATE '2023-04-30');

INSERT INTO CONTRACT\_ADOPTIE (id\_contract, id\_pet, id\_client, data\_adoptie) VALUES (seq\_contract\_adoptie.NEXTVAL, 1405, 5, DATE '2023-05-20');

------------------------------------------ SEQUENCE SI INSERARI ANGAJAT -----------------------------------------------

CREATE SEQUENCE seq\_angajat START WITH 250 INCREMENT BY 1 NOCACHE NOCYCLE;

INSERT INTO ANGAJAT (id\_angajat, id\_centru, nume, prenume, telefon, data\_angajare, ore\_saptamana, salariu) VALUES (seq\_angajat.NEXTVAL, 100, 'Popa', 'Cristian', '0711111111', DATE '2022-01-01', 40, 3000);

INSERT INTO ANGAJAT (id\_angajat, id\_centru, nume, prenume, telefon, data\_angajare, ore\_saptamana, salariu) VALUES (seq\_angajat.NEXTVAL, 101, 'Ionescu', 'Simona', '0722222222', DATE '2022-02-01', 35, 2800);

INSERT INTO ANGAJAT (id\_angajat, id\_centru, nume, prenume, telefon, data\_angajare, ore\_saptamana, salariu) VALUES (seq\_angajat.NEXTVAL, 102, 'Vasilescu', 'Alexandru', '0733333333', DATE '2022-03-01', 40, 3100);

INSERT INTO ANGAJAT (id\_angajat, id\_centru, nume, prenume, telefon, data\_angajare, ore\_saptamana, salariu) VALUES (seq\_angajat.NEXTVAL, 103, 'Marin', 'Adriana', '0744444444', DATE '2022-04-01', 30, 2700);

INSERT INTO ANGAJAT (id\_angajat, id\_centru, nume, prenume, telefon, data\_angajare, ore\_saptamana, salariu) VALUES (seq\_angajat.NEXTVAL, 104, 'Dumitru', 'Stefan', '0755555555', DATE '2022-05-01', 40, 3200);

INSERT INTO ANGAJAT (id\_angajat, id\_centru, nume, prenume, telefon, data\_angajare, ore\_saptamana, salariu) VALUES (seq\_angajat.NEXTVAL, 105, 'Nistor', 'Larisa', '0766666666', DATE '2022-06-01', 35, 2900);

INSERT INTO ANGAJAT (id\_angajat, id\_centru, nume, prenume, telefon, data\_angajare, ore\_saptamana, salariu) VALUES (seq\_angajat.NEXTVAL, 100, 'Toma', 'Bogdan', '0777777777', DATE '2022-07-01', 40, 3300);

INSERT INTO ANGAJAT (id\_angajat, id\_centru, nume, prenume, telefon, data\_angajare, ore\_saptamana, salariu) VALUES (seq\_angajat.NEXTVAL, 102, 'Radu', 'Elena', '0788888888', DATE '2022-08-01', 30, 2600);

INSERT INTO ANGAJAT (id\_angajat, id\_centru, nume, prenume, telefon, data\_angajare, ore\_saptamana, salariu) VALUES (seq\_angajat.NEXTVAL, 104, 'Petrescu', 'Andreea', '0799999999', DATE '2022-09-01', 38, 2950);

INSERT INTO ANGAJAT (id\_angajat, id\_centru, nume, prenume, telefon, data\_angajare, ore\_saptamana, salariu) VALUES (seq\_angajat.NEXTVAL, 103, 'Luca', 'George', '0700000000', DATE '2022-10-01', 37, 2850);

# *~ Cerința 12 ~*

-- Selecteaza numele si prenumele clientilor care au facut donatii

-- mai mari de 300 si care au adoptat animale din centrul cu adresa 'Strada Florilor, Nr. 5'

-- (subcereri sincronizate cu cel putin 3 tabele)

SELECT c.id\_client, c.nume, c.prenume

FROM CLIENT c

JOIN DONATIE d ON c.id\_client = d.id\_client

WHERE d.suma >= 300 AND EXISTS (

SELECT 1

FROM CONTRACT\_ADOPTIE ca

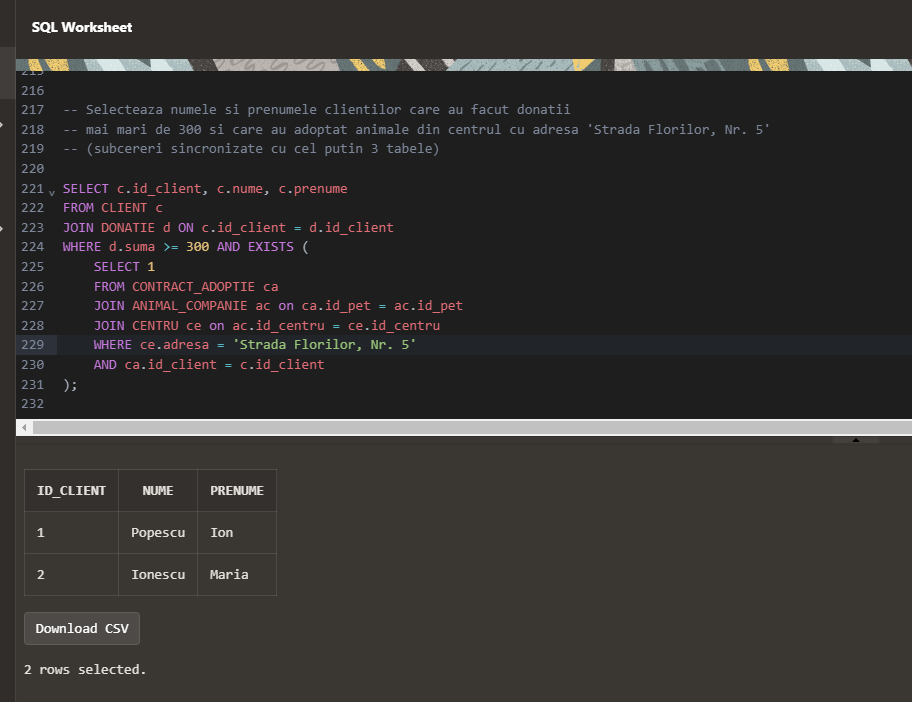
JOIN ANIMAL\_COMPANIE ac on ca.id\_pet = ac.id\_pet

JOIN CENTRU ce on ac.id\_centru = ce.id\_centru

WHERE ce.adresa = 'Strada Florilor, Nr. 5'

AND ca.id\_client = c.id\_client

);



Rezultate ok. Doar Popescu Ion si Ionescu Maria îndeplinesc criteriile

-- Selecteaza numele, prenumele, numarul de telefon si numarul

-- de ore/saptamana impreuna cu numarul total de donatii pentru fiecare

-- angajat care lucreaza in centrele care au cel putin 2 donatii

-- (subcereri nesincronizate in clauza FROM)

SELECT a.nume, a.prenume, a.telefon, a.ore\_saptamana, COUNT(d.id\_donatie) AS numar\_donatii

FROM (

SELECT c.id\_centru

FROM CENTRU c

JOIN DONATIE d ON d.id\_centru = c.id\_centru

GROUP BY c.id\_centru

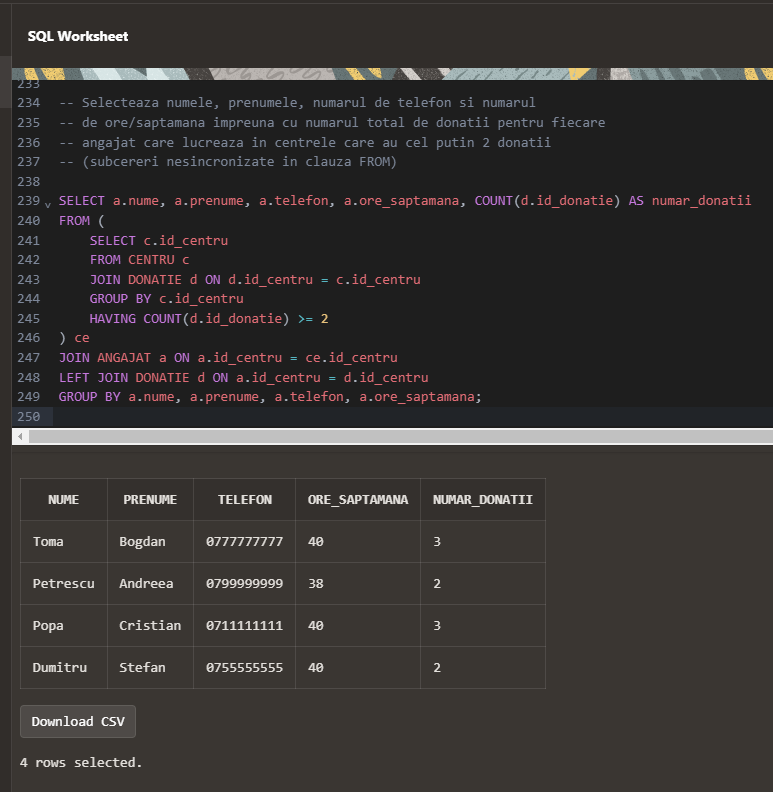
HAVING COUNT(d.id\_donatie) >= 2

) ce

JOIN ANGAJAT a ON a.id\_centru = ce.id\_centru

LEFT JOIN DONATIE d ON a.id\_centru = d.id\_centru

GROUP BY a.nume, a.prenume, a.telefon, a.ore\_saptamana;



Rezultate ok.

-- Selectează numele, prenumele, adresa și numărul de telefon al clienților

-- împreună cu suma persoanelor din toate programările făcute de aceștia,

-- grupati in functie de datele clientilor si verificati

-- pentru fiecare client care daca făcut cel puțin o programare în intervalul

-- orar '2' și cel puțin o programare pentru centrul cu id\_centru = 100.

-- (Grupari de date cu subcereri nesincronizate in clauza

-- HAVING cu cel putin 3 tabele)

SELECT c.nume, c.prenume, c.adresa, c.telefon, SUM(pv.nr\_pers) AS suma\_persoanelor

FROM CLIENT c

JOIN PROGRAMARE\_VIZITA pv ON c.id\_client = pv.id\_client

JOIN TUR t ON pv.id\_tur = t.id\_tur

JOIN CENTRU ce ON t.id\_centru = ce.id\_centru

WHERE c.id\_client IN (

SELECT DISTINCT pv1.id\_client

FROM PROGRAMARE\_VIZITA pv1

JOIN TUR t1 ON pv1.id\_tur = t1.id\_tur

WHERE pv1.interval\_orar = '2'

)

AND c.id\_client IN (

SELECT DISTINCT pv2.id\_client

FROM PROGRAMARE\_VIZITA pv2

JOIN TUR t2 ON pv2.id\_tur = t2.id\_tur

WHERE t2.id\_centru = 100

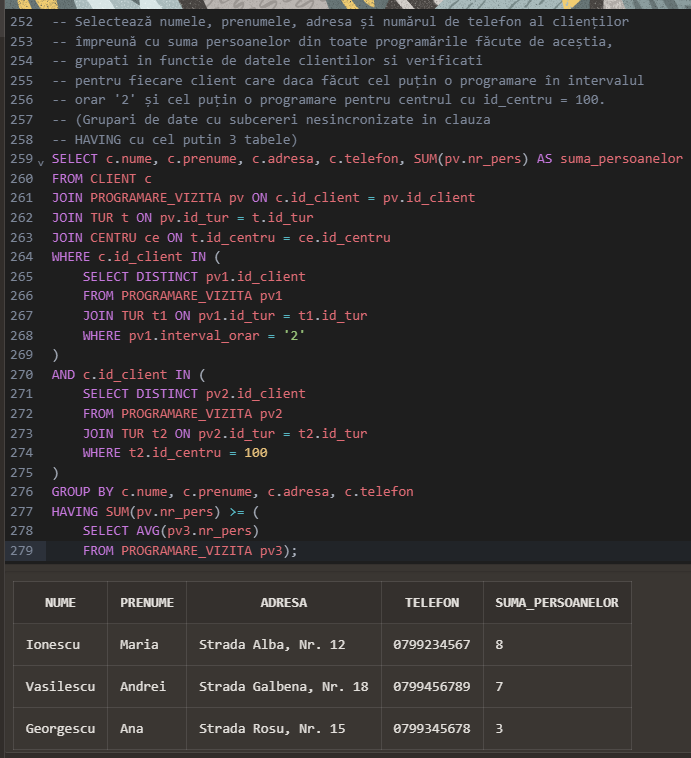
)

GROUP BY c.nume, c.prenume, c.adresa, c.telefon

HAVING SUM(pv.nr\_pers) >= (

SELECT AVG(pv3.nr\_pers)

FROM PROGRAMARE\_VIZITA pv3);



-- Selectează numele clienților, suma totală donată și un status al donației

-- ('High' dacă suma totală este mai mare de 500, altfel 'Low'), ordonat descrescător

-- după suma totală donată.

-- ( ordonare, folosirea NVL si DECODE )

SELECT c.nume, c.prenume, NVL(SUM(d.suma), 0) AS suma\_totala,

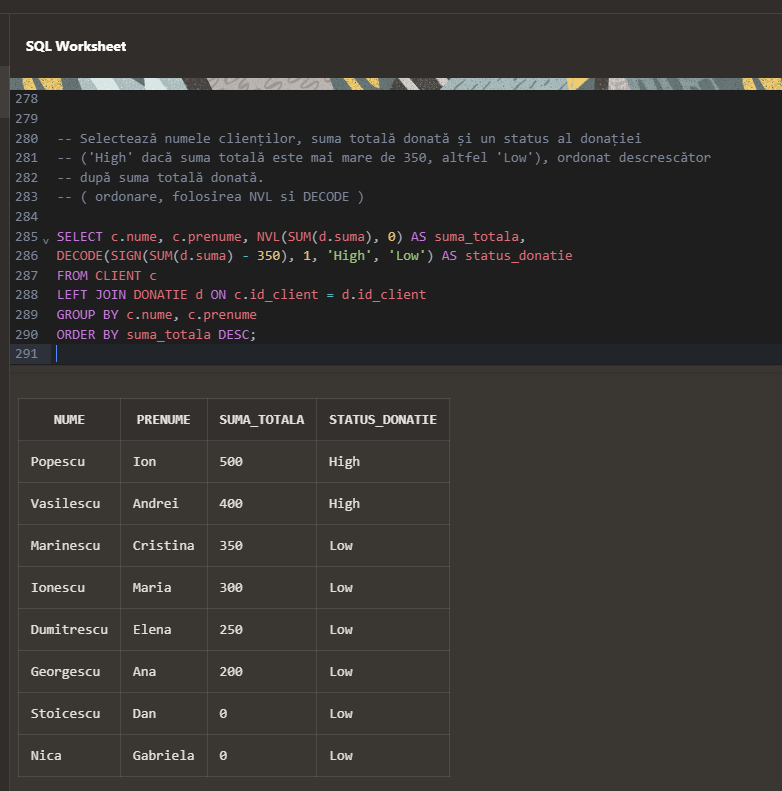
DECODE(SIGN(SUM(d.suma) - 500), 1, 'High', 'Low') AS status\_donatie

FROM CLIENT c

LEFT JOIN DONATIE d ON c.id\_client = d.id\_client

GROUP BY c.nume, c.prenume

ORDER BY suma\_totala DESC;



-- Selectează numele centrelor, data ultimei donații, numărul total de animale și

-- primele 10 caractere ale adresei centrului. Utilizează clauza WITH

-- pentru a obține datele despre donații și animalele din centru.

-- (functii pe siruri de caractere, date calendaristice, CASE si WITH)

WITH CentruInfo AS (

SELECT c.id\_centru, COUNT(ac.id\_pet) AS nr\_animale, MAX(d.data\_donatie) AS ultima\_donatie

FROM CENTRU c

LEFT JOIN ANIMAL\_COMPANIE ac ON c.id\_centru = ac.id\_centru

LEFT JOIN DONATIE d ON c.id\_centru = d.id\_centru

GROUP BY c.id\_centru

)

SELECT c.adresa, SUBSTR(c.adresa, 1, 10) AS adresa\_truncata, ci.nr\_animale,

TO\_CHAR(ci.ultima\_donatie, 'DD-MON-YYYY') AS data\_ultima\_donatie,

CASE

WHEN ci.nr\_animale > 5 THEN 'Multe'

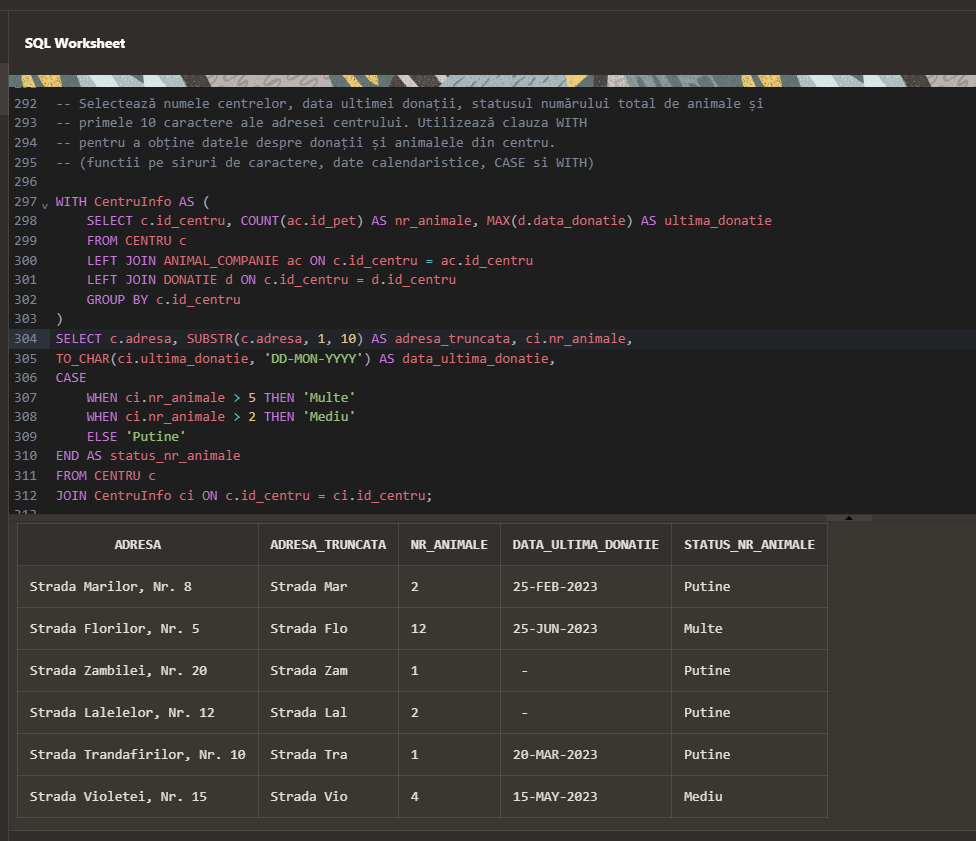
WHEN ci.nr\_animale > 2 THEN 'Mediu'

ELSE 'Putine'

END AS status\_nr\_animale

FROM CENTRU c

JOIN CentruInfo ci ON c.id\_centru = ci.id\_centru;



*~ Cerința 13 ~*

-- 1.Setam pentru fiecare centru numarul de animale de companie

-- care au fost inserate in tabelul ANIMAL\_COMPANIE

UPDATE CENTRU c

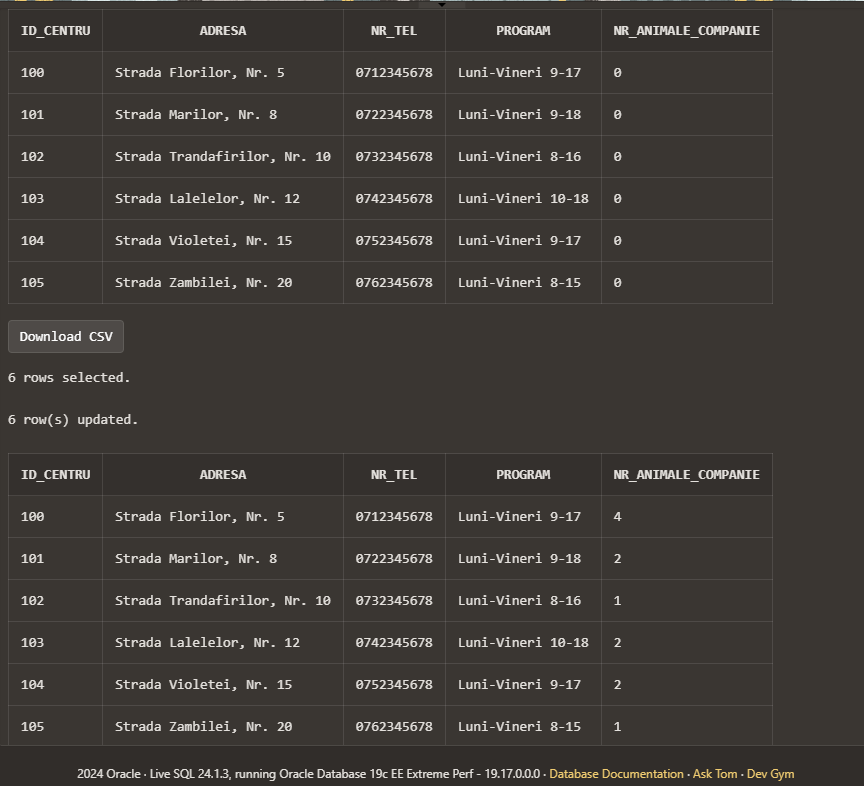
SET c.nr\_animale\_companie = (

SELECT COUNT(\*)

FROM ANIMAL\_COMPANIE ac

WHERE ac.id\_centru = c.id\_centru

);



-- 2. Adaug o noua coloana in tabelul CLIENT care va tine evidenta

-- numarului total de programari pe care le-a facut fiecare client

ALTER TABLE CLIENT ADD total\_programari INT;

UPDATE CLIENT c

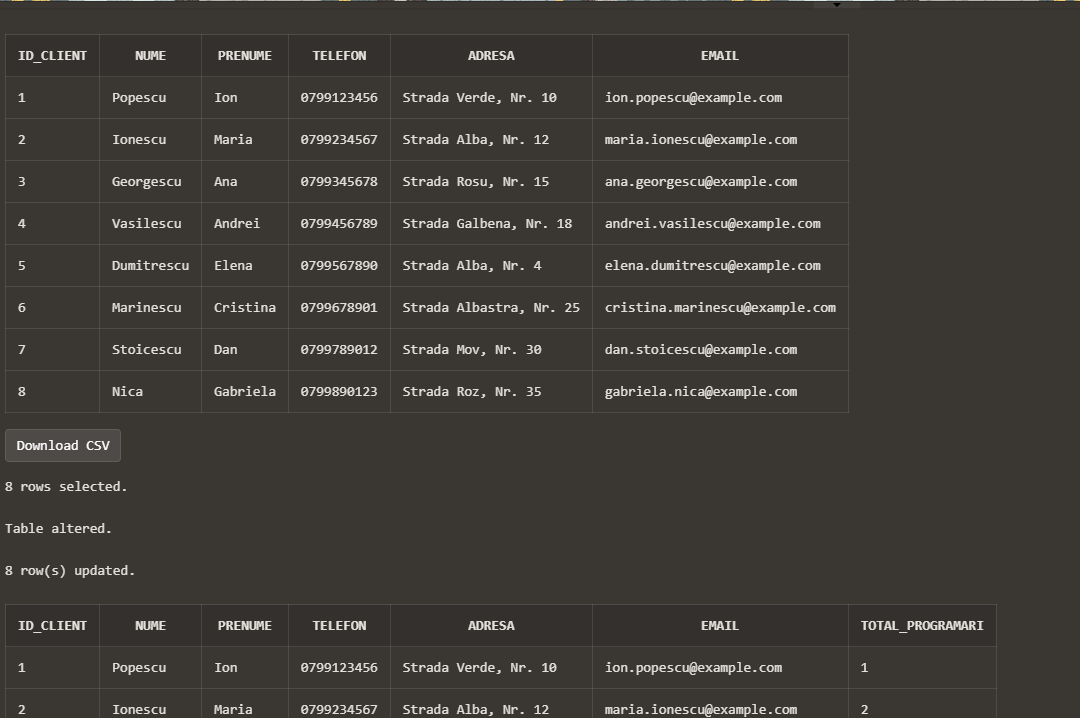
SET c.total\_programari = (

SELECT COUNT(\*)

FROM PROGRAMARE\_VIZITA pv

WHERE pv.id\_client = c.id\_client

);





-- 3. Sterg toate programarile care au fost facute inainte de

-- o anumita data. Exemplu: 2024-01-01

DELETE FROM PROGRAMARE\_VIZITA

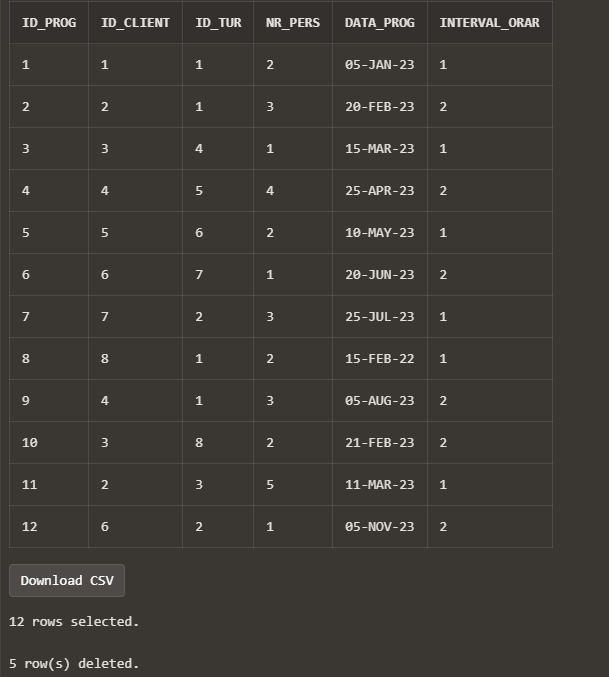
WHERE id\_tur IN (

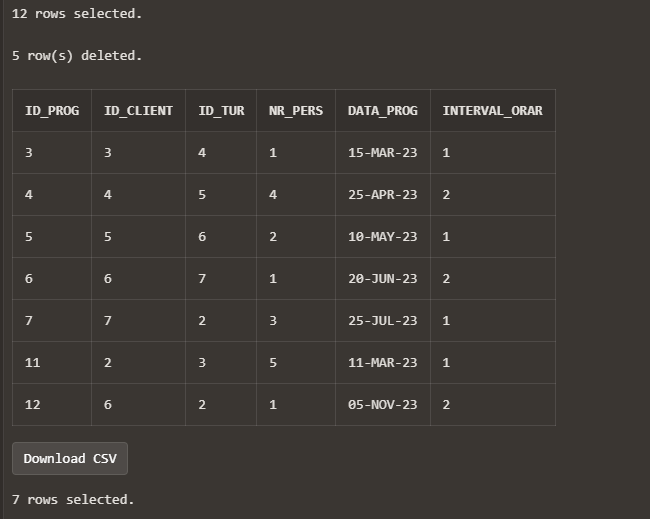
SELECT id\_tur

FROM TUR

WHERE data\_tur < TO\_DATE('2024-01-01', 'YYYY-MM-DD')

);





# *~ Printscreen-uri tabele și cod ~*

