Centrul de adopție pentru animalele de companie "Happy Puppy"

Tache Daria Elena

Grupa 134

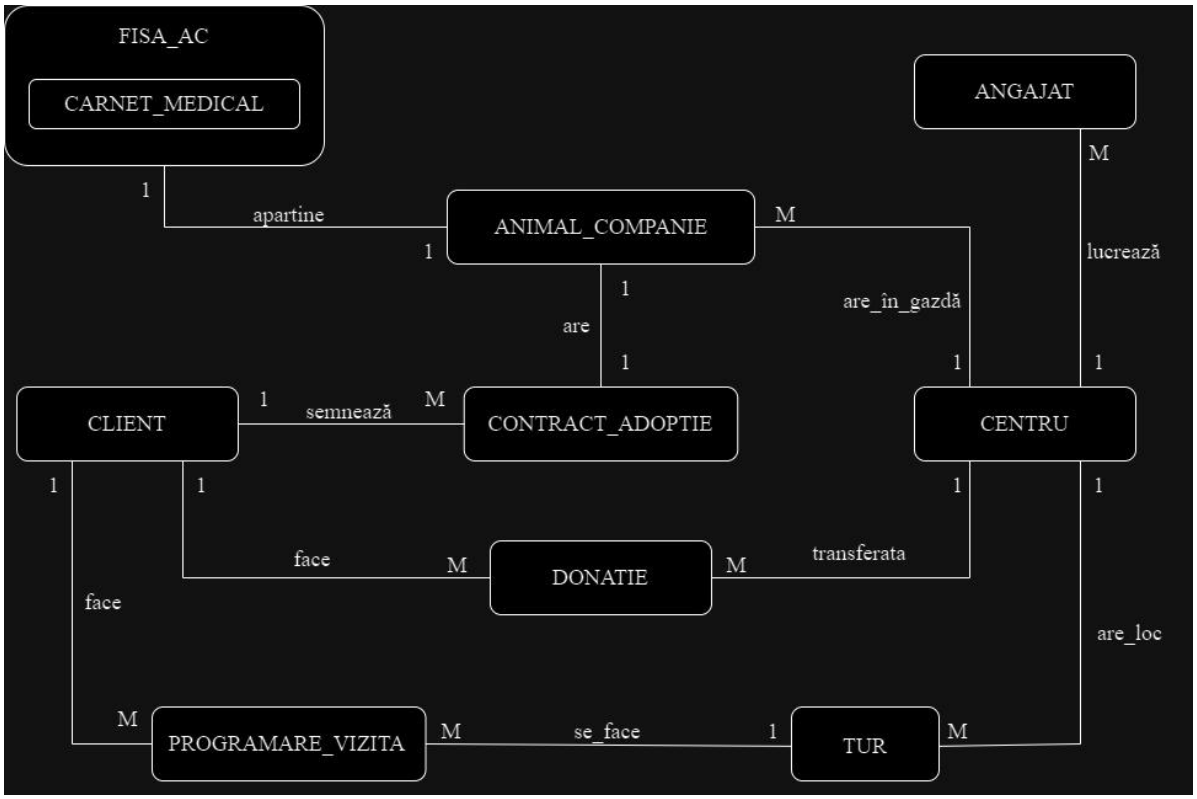
1. Prezentați pe scurt baza de date (utilitatea ei).

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.

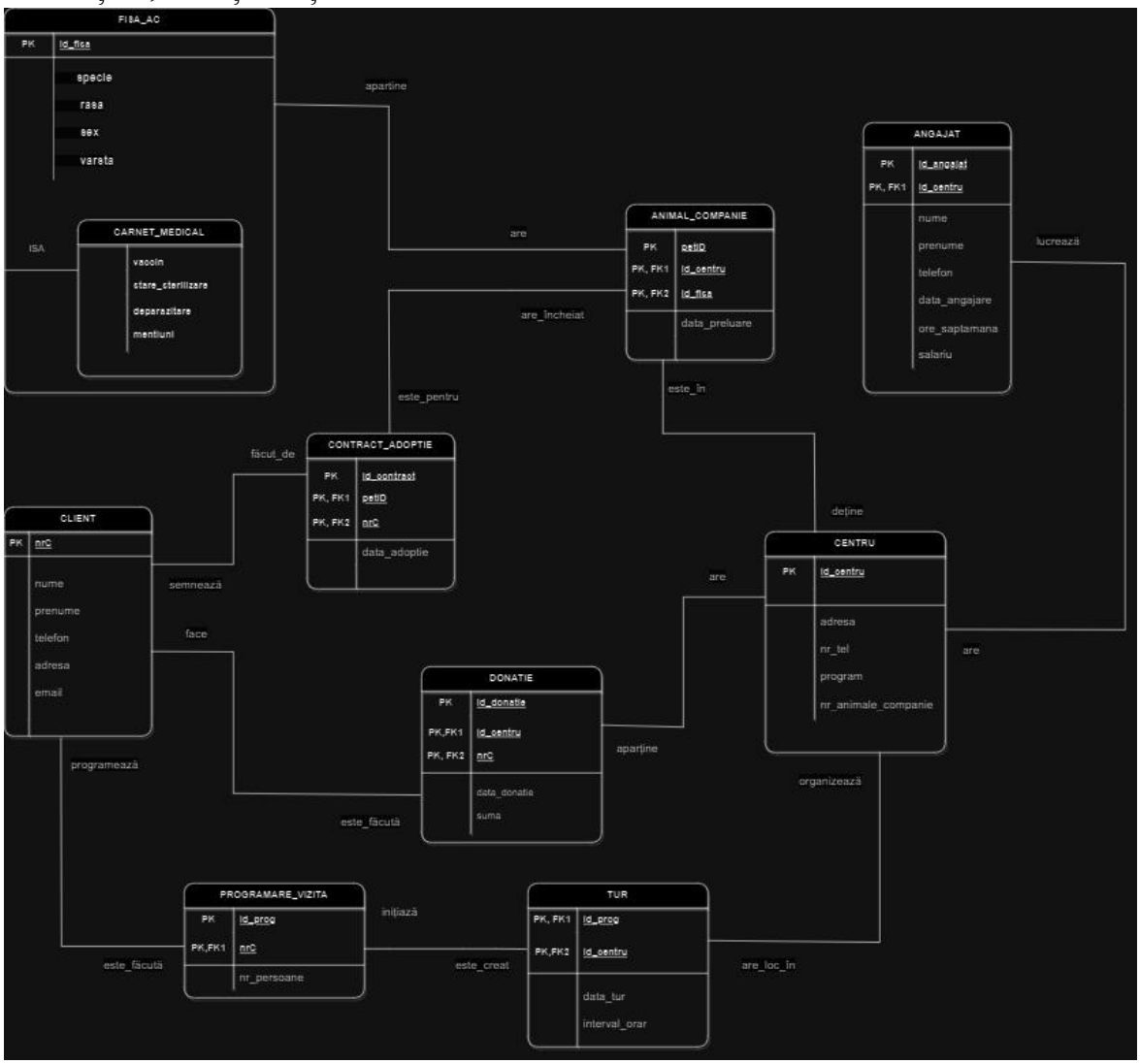
Centrele noastre vă oferă mai multe opțiuni: puteți adopta un animăluț, puteți dona pentru micii amici pufoși sau puteți veni cu copiii în baza unei programări la joacă!

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!

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



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



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

```
CREATE TABLE FISA_AC (
    id_fisa NUMBER PRIMARY KEY,
    specie VARCHAR2(50),
    rasa VARCHAR2(100),
```

```
    sex CHAR(1),
    varsta NUMBER
);

CREATE TABLE CARNET_MEDICAL(
    id_fisa NUMBER,
    vaccin VARCHAR2(500),
    stare_sterilizare CHAR(3) CHECK (UPPER(stare_sterilizare) IN
('DA', 'NU')),
    deparazitare CHAR(4) CHECK (UPPER(deparazitare) IN ('DA',
'NU')),
    mentiuni VARCHAR2(500),
    PRIMARY KEY(id_fisa),
    FOREIGN KEY(id_fisa) REFERENCES FISA_AC(id_fisa)
);

CREATE TABLE CENTRU (
    id_centru NUMBER PRIMARY KEY,
    adresa VARCHAR2(200),
    nr_tel VARCHAR2(20),
    program VARCHAR2(200),
    nr_animale_companie NUMBER
);

CREATE TABLE ANIMAL_COMPANIE (
    petID NUMBER PRIMARY KEY,
    id_centru NUMBER,
```

```
    id_fisa NUMBER,
    data_preluare DATE,
    adoptat CHAR(4) CHECK (UPPER(adoptat) IN ('DA', 'NU')),
    FOREIGN KEY (id_centru) REFERENCES CENTRU(id_centru),
    FOREIGN KEY (id_fisa) REFERENCES FISA_AC(id_fisa)
);

CREATE TABLE CLIENT(
    nrC NUMBER PRIMARY KEY,
    nume VARCHAR2(20),
    prenume VARCHAR2(50),
    telefon VARCHAR2(20),
    adresa VARCHAR2(200),
    email VARCHAR2(50)
);

CREATE TABLE PROGRAMARE_VIZITA(
    id_prog NUMBER PRIMARY KEY,
    nrC NUMBER NOT NULL,
    nr_pers NUMBER,
    FOREIGN KEY (nrC) REFERENCES CLIENT (nrC)
);

CREATE TABLE TUR(
    id_prog NUMBER,
    id_centru NUMBER,
```

```
    data_tur DATE,
    interval_orar VARCHAR2(50),
    PRIMARY KEY(id_prog, id_centru),
    FOREIGN KEY(id_prog) REFERENCES
PROGRAMARE_VIZITA(id_prog),
    FOREIGN KEY (id_centru) REFERENCES CENTRU(id_centru)
);

CREATE TABLE DONATIE(
    id_donatie NUMBER PRIMARY KEY,
    id_centru NUMBER,
    nrC NUMBER,
    data_donatie DATE,
    suma NUMBER,
    FOREIGN KEY(id_centru) REFERENCES CENTRU(id_centru),
    FOREIGN KEY(nrC) REFERENCES CLIENT(nrC)
);

CREATE TABLE CONTRACT_ADOPTIE(
    id_contract NUMBER PRIMARY KEY,
    petID NUMBER,
    nrC NUMBER,
    data_adoptie DATE,
    FOREIGN KEY (petID) REFERENCES
ANIMAL_COMPANIE(petID),
    FOREIGN KEY (nrC) REFERENCES CLIENT(nrC)
);
```

```
CREATE TABLE ANGAJAT(
    id_angajat NUMBER PRIMARY KEY,
    id_centru NUMBER,
    nume VARCHAR2(20),
    prenume VARCHAR2(50),
    telefon VARCHAR2(20),
    data_angajare DATE,
    ore_saptamana NUMBER,
    salariu NUMBER,
    FOREIGN KEY (id_centru) REFERENCES CENTRU(id_centru)
);
```

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

Cateva exemple:

```
CREATE SEQUENCE seq_centreID
    INCREMENT BY 1
    START WITH 100
    MINVALUE 1
    MAXVALUE 5000
    NOCYCLE;
```

```
CREATE SEQUENCE seq_petID
INCREMENT BY 1
START WITH 1400
MINVALUE 1
MAXVALUE 250000
NOCYCLE;


CREATE SEQUENCE seq_id_fisa
    START WITH 1
    INCREMENT BY 1
    NOCYCLE
    CACHE 40;


INSERT INTO CENTRU
    (id_centru, adresa, nr_tel, program, nr_animale_companie)
VALUES
    (seq_centreID.NEXTVAL, 'Str Alexandru cel Mare, 21',
'0726248310', '08:00 - 18:00', 0 );

INSERT INTO CENTRU
    (id_centru, adresa, nr_tel, program, nr_animale_companie)
VALUES
    (seq_centreID.NEXTVAL, 'Strada Victoriei, 10', '0723123456',
'09:00 - 19:00', 0);
```

```
INSERT INTO CENTRU
    (id_centru, adresa, nr_tel, program, nr_animale_companie)
VALUES
    (seq_centreID.NEXTVAL, 'Bulevardul Independenței, 5',
'0732112233', '10:00 - 20:00', 0);

INSERT INTO CENTRU
    (id_centru, adresa, nr_tel, program, nr_animale_companie)
VALUES
    (seq_centreID.NEXTVAL, 'Strada Aviatorilor, 15', '0712345678',
'08:30 - 18:30', 0);

INSERT INTO CENTRU
    (id_centru, adresa, nr_tel, program, nr_animale_companie)
VALUES
    (seq_centreID.NEXTVAL, 'Bulevardul Dacia, 8', '0755555555',
'10:30 - 20:30', 0);


DECLARE

    TYPE SpeciiType IS TABLE OF VARCHAR2(20);
    TYPE VarstaType IS TABLE OF NUMBER;
    TYPE RaseType IS TABLE OF VARCHAR2(50);
    TYPE CheckType IS TABLE OF VARCHAR2(3);
    TYPE MentiuniType IS TABLE OF VARCHAR2(20);
    TYPE GenderType IS TABLE OF VARCHAR2(3);
```

```
    TYPE VaccinType IS TABLE OF VARCHAR(20);
    TYPE DatePreluareType IS TABLE OF DATE INDEX BY
PLS_INTEGER;



    v_specii SpeciiType := SpeciiType('Caine', 'Pisica', 'Iepure',
'Hamster', 'Papagal');
    v_varsta VarstaType := VarstaType(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,
12);
    v_rase_caine RaseType := RaseType('Labrador', 'Ciobanesc
German', 'Bullog');
    v_rase_pisica RaseType := RaseType('Siameza', 'Albastru de
Rusia', 'British Shorthair');
    v_rase_iepure RaseType := RaseType('Pitic', 'Belier');
    v_rase_hamster RaseType := RaseType('Pitic Roborovsky');
    v_check CheckType := CheckType('DA', 'NU');
    v_mentiuni MentiuniType := MentiuniType('Nicio mentiune', 'Are
alergii', 'Iubitor');
    v_gender GenderType := GenderType('F', 'M');
    v_vaccin VaccinType := VaccinType('Vaccin A', 'Vaccin B',
'Vaccin C', 'Vaccin D');
    v_date_preluare DatePreluareType;


    v_nr_specii CONSTANT NUMBER := v_specii.COUNT;
    v_nr_rase_caine CONSTANT NUMBER := v_rase_caine.COUNT;
    v_nr_rase_pisica CONSTANT NUMBER :=
v_rase_pisica.COUNT;
    v_nr_rase_iepure CONSTANT NUMBER :=
v_rase_iepure.COUNT;
```

```
    v_nr_rase_hamster CONSTANT NUMBER :=
v_rase_hamster.COUNT;
    v_nr_check CONSTANT NUMBER := v_check.COUNT;
    v_nr_mentiuni CONSTANT NUMBER := v_mentiuni.COUNT;
    v_nr_gender CONSTANT NUMBER := v_gender.COUNT;
    v_nr_vaccin CONSTANT NUMBER := v_vaccin.COUNT;

    v_random_specie VARCHAR2(20);
    v_random_varsta NUMBER;
    v_random_rasa VARCHAR2(50);
    v_random_check1 VARCHAR2(3);
    v_random_check2 VARCHAR2(3);
    v_random_mentiune VARCHAR2(20);
    v_random_gender VARCHAR(3);
    v_random_vaccin VARCHAR(20);

    v_random_data DATE;

    v_counter NUMBER := 0;

    v_id_fisa NUMBER;
    v_id_centru NUMBER;
    v_petID NUMBER;
    v_nr_centre NUMBER;
    v_centru_curent NUMBER;
BEGIN
```

```
    -- Introduc date in tabel:
    v_date_preluare(1) := TO_DATE('2023-01-01', 'YYYY-MM-DD');
    v_date_preluare(2) := TO_DATE('2023-02-15', 'YYYY-MM-DD');
    v_date_preluare(3) := TO_DATE('2023-03-10', 'YYYY-MM-DD');
    v_date_preluare(4) := TO_DATE('2023-04-22', 'YYYY-MM-DD');
    v_date_preluare(5) := TO_DATE('2023-05-05', 'YYYY-MM-DD');
    v_date_preluare(6) := TO_DATE('2023-06-18', 'YYYY-MM-DD');
    v_date_preluare(7) := TO_DATE('2023-07-03', 'YYYY-MM-DD');
    v_date_preluare(8) := TO_DATE('2023-08-14', 'YYYY-MM-DD');
    v_date_preluare(9) := TO_DATE('2023-09-27', 'YYYY-MM-DD');
    v_date_preluare(10) := TO_DATE('2023-10-09', 'YYYY-MM-
DD');

    SELECT COUNT(*) INTO v_nr_centre FROM CENTRU;



    WHILE v_counter < 16 LOOP

        -- Se obtine id-ul unui centru random
        SELECT id_centru INTO v_centru_curent FROM CENTRU
WHERE ROWNUM = 1 ORDER BY DBMS_RANDOM.VALUE;

        -- Se obține id-ul unei fise disponibile
        v_id_fisa := seq_id_fisa.NEXTVAL;

        -- Se obține un nou id pentru animal
        v_petID := seq_petID.NEXTVAL;
```

```
        v_random_specie := v_specii(DBMS_RANDOM.VALUE(1,
v_nr_specii));


        -- Alegere aleatoare pentru rasa în funcție de specie
        CASE v_random_specie
            WHEN 'Caine' THEN v_random_rasa :=
v_rase_caine(DBMS_RANDOM.VALUE(1, v_nr_rase_caine));
            WHEN 'Pisica' THEN v_random_rasa :=
v_rase_pisica(DBMS_RANDOM.VALUE(1, v_nr_rase_pisica));
            WHEN 'Iepure' THEN v_random_rasa :=
v_rase_iepure(DBMS_RANDOM.VALUE(1, v_nr_rase_iepure));
            WHEN 'Hamster' THEN v_random_rasa :=
v_rase_hamster(DBMS_RANDOM.VALUE(1, v_nr_rase_hamster));
            ELSE v_random_rasa := NULL;
        END CASE;


        -- Alegere aleatoare pentru Check, Varsta, Gender, Vaccin,
Mentiuni si Data
        v_random_check1 := v_check(DBMS_RANDOM.VALUE(1,
v_nr_check));
        v_random_check2 := v_check(DBMS_RANDOM.VALUE(1,
v_nr_check));
        v_random_varsta :=
v_varsta(DBMS_RANDOM.VALUE(1,12));
        v_random_gender := v_gender(DBMS_RANDOM.VALUE(1,
v_nr_gender));
        v_random_vaccin := v_vaccin(DBMS_RANDOM.VALUE(1,
v_nr_vaccin));
```

```
        v_random_mentiune :=
v_mentiuni(DBMS_RANDOM.VALUE(1, v_nr_mentiuni));
        v_random_data :=
v_date_preluare(DBMS_RANDOM.VALUE(1,10));

        INSERT INTO FISA_AC
            (id_fisa, specie, rasa, sex, varsta)
        VALUES
            (v_id_fisa, v_random_specie, v_random_rasa,
v_random_gender, v_random_varsta);

        INSERT INTO CARNET_MEDICAL
            (id_fisa, vaccin, stare_sterilizare, deparazitare, mentiuni)
        VALUES
            (v_id_fisa, v_random_vaccin, v_random_check1,
v_random_check2, v_random_mentiune);

        INSERT INTO ANIMAL_COMPANIE
            (petID, id_centru, id_fisa, data_preluare, adoptat)
        VALUES
            (v_petID, v_centru_curent, v_id_fisa, v_random_data, 'NU');

        -- După inserarea în ANIMAL_COMPANIE
        UPDATE CENTRU
        SET nr_animale_companie = nr_animale_companie + 1
        WHERE id_centru = v_centru_curent;

        v_counter := v_counter + 1;
```

```
    END LOOP;
END;
/
```

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

```
/*
Clientii "Star" sunt clientii care au donat, au planificat o vizita si au adoptat.
Retinand in 3 tipuri diferite de colectii de date id-urile clientilor din tabela de donatie,
planificare_vizita si contract_adoptie, voi afisa pe ecran cati clienti "Star" avem
    Raspuns: 1
*/

CREATE OR REPLACE PROCEDURE Exercitiul6 IS
    TYPE DonatieType IS TABLE OF NUMBER INDEX BY PLS_INTEGER;
    TYPE VizitaType IS TABLE OF NUMBER;
    TYPE AdoptieType IS VARRAY(50) OF NUMBER;

TYPE DetaliiType IS RECORD (
        nume CLIENT.nume%TYPE,
        prenume CLIENT.prenume%TYPE
```

```
    );

    v_detalii DetaliiType;

    v_donatii DonatieType;
    v_vizite VizitaType := VizitaType();
    v_adoptii AdoptieType := AdoptieType();

v_nrC NUMBER;

    v_star_clients NUMBER := 0;
BEGIN

    SELECT DISTINCT nrC
    BULK COLLECT INTO v_donatii
    FROM DONATIE;

    FOR i IN (SELECT DISTINCT nrC FROM
PROGRAMARE_VIZITA) LOOP
        v_vizite.EXTEND;
        v_vizite(v_vizite.LAST) := i.nrC;
    END LOOP;

 FOR i IN (SELECT DISTINCT nrC FROM
CONTRACT_ADOPTIE) LOOP
        v_adoptii.EXTEND;
        v_adoptii(v_adoptii.LAST) := i.nrC;
```

```
    END LOOP;

   FOR i IN (SELECT DISTINCT nrC FROM CLIENT) LOOP
      v_nrC := i.nrC;

      -- Vedem dacă nrC există în toate cele trei colecții
      IF v_donatii.EXISTS(v_nrC) AND v_vizite.EXISTS(v_nrC)
AND v_adoptii.EXISTS(v_nrC) THEN

         SELECT nume, prenume INTO v_detalii.nume,
v_detalii.prenume
FROM CLIENT
WHERE nrC = v_nrC;

DBMS_OUTPUT.PUT_LINE('Client ' || v_nrC || ', ' || v_detalii.nume
|| ' ' || v_detalii.prenume ||' este un Star Client!');

         v_star_clients := v_star_clients + 1;
      END IF;
   END LOOP;

   DBMS_OUTPUT.PUT_LINE('Numar total de Star Clients: ' ||
v_star_clients);
END Exercitiul6;
/


-- Apelul subprogramului:
```

```
BEGIN
Exercitiul6;
END;
/
```

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

```
CREATE OR REPLACE PROCEDURE CalculSumaDonatii IS
    -- Cursorul implicit pentru Centre
    CURSOR centru_cursor IS
        SELECT id_centru, adresa FROM CENTRU;

    -- Cursorul pentru Donații
    CURSOR donatie_cursor (p_id_centru_donatie NUMBER) IS
        SELECT suma FROM DONATIE
        WHERE id_centru = p_id_centru_donatie;

    v_id_centru CENTRU.id_centru%TYPE;
    v_adresa_centru CENTRU.adresa%TYPE;
    v_suma_totala NUMBER := 0;
    v_suma_donatie NUMBER;
BEGIN
    OPEN centru_cursor;
```

```
    LOOP
        FETCH centru_cursor INTO v_id_centru, v_adresa_centru;
        EXIT WHEN centru_cursor%NOTFOUND;

        OPEN donatie_cursor(v_id_centru);
        LOOP
            FETCH donatie_cursor INTO v_suma_donatie;
            EXIT WHEN donatie_cursor%NOTFOUND;

            v_suma_totala := v_suma_totala + v_suma_donatie;
        END LOOP;
        CLOSE donatie_cursor;

        DBMS_OUTPUT.PUT_LINE('Centrul cu adresa: ' ||
v_adresa_centru);
        DBMS_OUTPUT.PUT_LINE('Suma totală donată: ' ||
v_suma_totala);

        v_suma_totala := 0;
    END LOOP;
    CLOSE centru_cursor;
END CalculSumaDonatii;
/

select * from donatie;

BEGIN
```

```
    CalculSumaDonatii();
END;
/
```

Exercitiul 8:

```
/

-- Apel cu client nou, fara eroare:
BEGIN
    ProgramareTur('Toma', 'Alex', '0777227722', 102, 3, 11, 13);
END;
/
```

```
Procedure created.

Statement processed.
Programare pentru tur realizată cu succes!
```

```
-- Apel cu client nou, cu eroare la id_centru:
BEGIN
    ProgramareTur('Toma', 'Alex', '0777227722', 202, 3, 11, 13);
END;
/
```

```
Statement processed.
Id_centru NU există în tabela CENTRU.
```

```
-- Apel cu client nou, cu eroare la durata turului:
BEGIN
    ProgramareTur('Ion', 'Mihnea', '0733333332', 104, 3, 8, 13);
END;
/
```

```
Statement processed.
Eroare: Durata turului nu poate depăși 3 ore.
```

```
1006
1007  -- Apel cu client nou, cu eroare la orele de inceput si final ale turului:
1008  BEGIN
1009      ProgramareTur('Calin', 'Anca', '0735533332', 104, 2, 10, 8);
1010  END;
1011  /
1012
```

```
Statement processed.
Eroare: Ora de final nu poate fi precedentă orei de început.
```

Exercitiul 10:

```
992   ---------- Declansare Trigger -----------
993
994   DECLARE
995       v_id_donatie NUMBER;
996       v_id_centru NUMBER;
997       v_nrC NUMBER;
998   BEGIN
999       SELECT id_centru INTO v_id_centru FROM CENTRU WHERE ROWNUM = 1 ORDER BY
1000      SELECT nrC INTO v_nrC FROM CLIENT WHERE ROWNUM = 1 ORDER BY DBMS_RANDOM
1001      v_id_donatie := seq_id_donatie.NEXTVAL;
1002
1003      INSERT INTO DONATIE
1004          (id_donatie, id_centru, nrC, data_donatie, suma)
1005      VALUES
1006          (seq_id_donatie.NEXTVAL, v_id_centru, v_nrC, TO_DATE('2022-09-30',
1007      END;
1008  /
```

```
Statement processed.
Operație INSERT asupra tabelului donatie
```

Exercitiul 11:

```
1023      ------- Declansare Trigger --------
1024  INSERT INTO DONATIE (id_donatie, id_centru, nrC, suma)
1025  VALUES (404, 101, 17, 1000);
1026
1027  select * from donatie;
1028
1029
```

408	103	3	31-AUG-23	120
410	101	4	02-SEP-23	150
412	100	1	30-SEP-22	45
404	101	17	12-JAN-24	1000

Exercitiul 12:

```
1034  CREATE OR REPLACE TRIGGER trg_LDD_Donatie
1035  BEFORE DELETE ON DONATIE
1036  FOR EACH ROW
1037  BEGIN
1038      DBMS_OUTPUT.PUT_LINE('Ștergere donație cu id ' || :OLD.id_donatie || ' înainte de ștergere.');
1039
1040  END;
1041  /
1042
1043  DELETE FROM DONATIE WHERE id_donatie = 404;
1044
1045
```

```
1 row(s) deleted.
Ștergere donație cu id 404 înainte de ștergere.
```