**PROIECT BAZE DE DATE**

**Adapost de Animale**

**MOCANU ALEXANDRA**

**Grupa 132**

**Cuprins**

Codul intreg………………………………………………………………………………………………3

Descriere model real…………………………………………………………………………………15

Constrangeri……………………………………………………………………………………………..15

Entitati………………………………………………………………………………………………………15

Relatii si cardinalitate………………………………………………………………………………..17

Atribute…………………………………………………………………………………………………….18

Diagrama entitate-relatie………………………………………………………………………….21

Diagrama conceptuala………………………………………………………………………………22

Scheme relationale……………………………………………………………………………………22

Normalizare……………………………………………………………………………………………….23

Secventa…………………………………………………………………………………………………….23

Creare si inserare……………………………………………………………………………………….23

Cereri SQL………………………………………………………………………………………………….24

Operatii……………………………………………………………………………………………………...26

Cereri (division,top-n,outer-join)…………………………………………………………………27

Optimizare…………………………………………………………………………………………………..28

BCNF, FN4, FN5…………………………………………………………………………………………….28

**Codul intreg:**

CREATE TABLE CONTRACT (

NUMAR INT PRIMARY KEY,

DATA\_EMITERE DATE,

DURATA INT,

TIP VARCHAR(255),

id\_magazin INT,

id\_farmacie INT,

id\_clinica INT,

FOREIGN KEY (id\_magazin) REFERENCES MAGAZIN (id\_magazin),

FOREIGN KEY (id\_farmacie) REFERENCES FARMACIE (id\_farmacie),

FOREIGN KEY (id\_clinica) REFERENCES CLINICA\_VETERINARA (ID\_CLINICA)

);

CREATE TABLE ANGAJAT (

ID\_ANGAJAT INT PRIMARY KEY,

NUME VARCHAR(255),

PRENUME VARCHAR(255),

CNP VARCHAR(13),

ADRESA VARCHAR(255),

TELEFON VARCHAR(255),

EMAIL VARCHAR(255),

ID\_CONTRACT INT,

FOREIGN KEY (ID\_CONTRACT) REFERENCES CONTRACT (NUMAR)

);

CREATE TABLE DIRECTOR (

ID\_DIRECTOR INT PRIMARY KEY,

ID\_CONTRACT INT,

ID\_ANGAJAT INT,

FOREIGN KEY (ID\_CONTRACT) REFERENCES CONTRACT (NUMAR),

FOREIGN KEY (ID\_ANGAJAT) REFERENCES ANGAJAT (ID\_ANGAJAT)

);

CREATE TABLE SECTOR (

ID\_SECTOR INT PRIMARY KEY,

TIP\_NEVOIE VARCHAR(255),

NUMAR\_LOCURI INT

);

CREATE TABLE SECTIE\_SECTOR (

ID\_SECTIE INT,

ID\_SECTOR INT,

PRIMARY KEY (ID\_SECTIE, ID\_SECTOR),

FOREIGN KEY (ID\_SECTIE) REFERENCES SECTIE (ID\_SECTIE),

FOREIGN KEY (ID\_SECTOR) REFERENCES SECTOR (ID\_SECTOR)

);

CREATE TABLE SECTIE (

ID\_SECTIE INT PRIMARY KEY,

NUME VARCHAR(255),

ID\_CUSCA INT,

FOREIGN KEY (ID\_CUSCA) REFERENCES CUSCA (ID\_CUSCA)

);

CREATE TABLE FARMACIE (

ID\_FARMACIE INT PRIMARY KEY,

NUME VARCHAR(255),

TELEFON VARCHAR(255),

PROGRAM VARCHAR(255),

NUMAR INT,

FOREIGN KEY (NUMAR) REFERENCES MEDICAMENT (NUMAR)

);

CREATE TABLE MEDICAMENT (

NUMAR INT PRIMARY KEY,

DENUMIRE\_PRODUS VARCHAR(255),

CANTITATE INT

);

CREATE TABLE VIZITATOR (

CNP VARCHAR(13) PRIMARY KEY,

NUME VARCHAR(255),

PRENUME VARCHAR(255),

TELEFON VARCHAR(255),

EMAIL VARCHAR(255)

);

CREATE TABLE CERERE\_ADOPTIE (

SERIE VARCHAR(255),

ID\_ANIMAL INT,

ID\_VIZITATOR VARCHAR(13),

DATA DATE,

--APROBATA\_DE\_DIRECTOR BOOLEAN DEFAULT FALSE,

PRIMARY KEY (SERIE, ID\_ANIMAL, ID\_VIZITATOR),

FOREIGN KEY (ID\_ANIMAL) REFERENCES ANIMAL (ID\_ANIMAL),

FOREIGN KEY (ID\_VIZITATOR) REFERENCES VIZITATOR (CNP)

);

CREATE TABLE ANIMAL (

ID\_ANIMAL INT PRIMARY KEY,

NUME VARCHAR(255),

RASA VARCHAR(255),

VARSTA INT,

SEX VARCHAR(255),

DATA\_SOSIRE DATE,

ID\_CUSCA INT,

ID\_VETERINAR VARCHAR(13),

FOREIGN KEY (ID\_CUSCA) REFERENCES CUSCA (ID\_CUSCA),

FOREIGN KEY (ID\_VETERINAR) REFERENCES VETERINAR (CNP)

);

CREATE TABLE CUSCA (

ID\_CUSCA INT PRIMARY KEY,

NUMAR\_LOCURI INT

);

CREATE TABLE VETERINAR (

CNP VARCHAR(13) PRIMARY KEY,

NUME VARCHAR(255),

PRENUME VARCHAR(255),

ID\_ANIMAL INT,

ID\_CONSULTATIE,

FOREIGN KEY (ID\_ANIMAL) REFERENCES ANIMAL (ID\_ANIMAL),

FOREIGN KEY (ID\_CONSLTATIE) REFERENCES CONSULTATIE (ID\_CONSULTATIE)

);

CREATE TABLE MAGAZIN (

ID\_MAGAZIN INT PRIMARY KEY,

ID\_CONTRACT INT,

NUME VARCHAR(255),

TELEFON VARCHAR(255),

PROGRAM VARCHAR(255),

FOREIGN KEY (ID\_MAGAZIN) REFERENCES MAGAZIN (ID\_MAGAZIN)

);

CREATE TABLE HRANA (

NUMAR INT PRIMARY KEY,

DENUMIRE\_PRODUS VARCHAR(255),

CANTITATE INT,

ID\_MAGAZIN INT,

FOREIGN KEY (ID\_MAGAZIN) REFERENCES MAGAZIN (ID\_MAGAZIN)

);

CREATE TABLE CARTE\_DE\_SANATATE (

ID\_ANIMAL INT PRIMARY KEY,

VACCINURI VARCHAR(255),

DATA\_NASTERE DATE,

ANTECEDENTE\_MEDICALE VARCHAR(255),

ID\_VETERINAR VARCHAR(13),

FOREIGN KEY (ID\_VETERINAR) REFERENCES VETERINAR (CNP)

);

CREATE TABLE CONSULTATIE (

ID\_CONSULTATIE INT PRIMARY KEY,

ID\_ANIMAL INT,

DATA DATE,

ID\_VETERINAR VARCHAR(13),

FOREIGN KEY (ID\_VETERINAR) REFERENCES VETERINAR (CNP)

);

CREATE TABLE CLINICA\_VETERINARA (

ID\_CLINICA INT PRIMARY KEY,

DENUMIRE VARCHAR(255),

ADRESA VARCHAR(255),

TELEFON VARCHAR(255),

EMAIL VARCHAR(255),

ID\_VETERINAR VARCHAR(13),

FOREIGN KEY (ID\_VETERINAR) REFERENCES VETERINAR (CNP)

);

-- Tabela CONTRACT

INSERT INTO CONTRACT (NUMAR, DATA\_EMITERE, DURATA, TIP, id\_magazin, id\_farmacie, id\_clinica)

VALUES (1, TO\_DATE('2023-01-01', 'YYYY-MM-DD'), 12, 'Standard', 1, 1, 1),

VALUES (2, TO\_DATE('2023-02-01', 'YYYY-MM-DD'), 6, 'Premium', 2, 2, 2),

VALUES (3, TO\_DATE('2023-01-01', 'YYYY-MM-DD'), 3, 'Standard', 3, 3, 3),

VALUES (4, TO\_DATE('2023-01-01', 'YYYY-MM-DD'), 10, 'Premium', 4, 4, 4),

VALUES (5, TO\_DATE('2023-01-01', 'YYYY-MM-DD'), 12, 'Standard', 5, 5, 5);

-- Tabela ANGAJAT

INSERT INTO ANGAJAT (ID\_ANGAJAT, NUME, PRENUME, CNP, ADRESA, TELEFON, EMAIL, ID\_CONTRACT)

VALUES (1, 'Popescu', 'Ion', '1234567890123', 'Str. Muncii, Nr. 1', '123456789', 'popescu.ion@example.com', 1),

(2, 'Ionescu', 'Maria', '9876543210987', 'Str. Victoriei, Nr. 2', '987654321', 'ionescu.maria@example.com', 2),

(3, 'Dumitrescu', 'Ana', '4567890123456', 'Str. Libert??ii, Nr. 3', '456789012', 'dumitrescu.ana@example.com', 3),

(4, 'Constantinescu', 'Mihai', '7890123456789', 'Str. Independen?ei, Nr. 4', '789012345', 'constantinescu.mihai@example.com', 4),

(5, 'Georgescu', 'Elena', '2345678901234', 'Str. Revolu?iei, Nr. 5', '234567890', 'georgescu.elena@example.com', 5);

-- Tabela DIRECTOR

INSERT INTO DIRECTOR (ID\_DIRECTOR, ID\_CONTRACT, ID\_ANGAJAT)

VALUES (1, 1, 1),

VALUES (2, 2, 2),

VALUES (3, 3, 3),

VALUES (4, 4, 4),

VALUES (5, 5, 5);

-- Tabela SECTOR --

INSERT INTO SECTOR (ID\_SECTOR, TIP\_NEVOIE, NUMAR\_LOCURI)

VALUES (1, 'Tip 1', 10),

VALUES (2, 'Tip 2', 5),

VALUES (3, 'Tip 3', 8),

VALUES (4, 'Tip 4', 11),

VALUES (5, 'Tip 5', 15);

-- Tabela SECTIE\_SECTOR --

INSERT INTO SECTIE\_SECTOR (ID\_SECTIE, ID\_SECTOR)

VALUES (1, 1),

VALUES (2, 2),

VALUES (3, 3),

VALUES (4, 4),

VALUES (5, 5);

-- Tabela SECTIE --

INSERT INTO SECTIE (ID\_SECTIE, NUME, ID\_CUSCA)

VALUES (1, 'Sectie 1', 1),

VALUES (2, 'Sectie 2', 2),

VALUES (3, 'Sectie 3', 3),

VALUES (4, 'Sectie 4', 4),

VALUES (5, 'Sectie 5', 5);

-- Tabela FARMACIE --

INSERT INTO FARMACIE (ID\_FARMACIE, NUME, TELEFON, PROGRAM, NUMAR)

VALUES (1, 'Pharmacy 1', '123-456-7890', '9 AM - 6 PM', 1);

INSERT INTO FARMACIE (ID\_FARMACIE, NUME, TELEFON, PROGRAM, NUMAR)

VALUES (2, 'Pharmacy 2', '123-456-7899', '9 AM - 6 PM', 1);

INSERT INTO FARMACIE (ID\_FARMACIE, NUME, TELEFON, PROGRAM, NUMAR)

VALUES (3, 'Pharmacy 3', '123-456-3451', '9 AM - 10 PM', 1),

VALUES (4, 'Pharmacy 4', '123-456-2469', '7 AM - 3 PM', 1),

VALUES (5, 'Pharmacy 5', '123-456-9452', '9 AM - 6 PM', 1);

-- Tabela MEDICAMENT --

INSERT INTO MEDICAMENT (NUMAR, DENUMIRE\_PRODUS, CANTITATE)

VALUES (1, 'Medicament 1', 50),

VALUES (2, 'Medicament 2', 100),

VALUES (3, 'Medicament 3', 20),

VALUES (4, 'Medicament 4', 80),

VALUES (5, 'Medicament 5', 150);

-- Tabela VIZITATOR --

INSERT INTO VIZITATOR (CNP, NUME, PRENUME, TELEFON, EMAIL)

VALUES ('1234567890123', 'Jane', 'Doe', '987-654-3210', 'jane@example.com');

INSERT INTO VIZITATOR (CNP, NUME, PRENUME, TELEFON, EMAIL)

VALUES ('1234678390124', 'Jany', 'Denice', '947-967-3215', 'jany@gmail.com'),

VALUES ('5434564573465', 'Dan', 'Brice', '987-654-4662', 'brice@gmail.com'),

VALUES ('1234903767384', 'Jacob', 'Marian', '246-654-3215', 'marian@gmail.com'),

VALUES ('6542952096781', 'Ghica', 'Suditu', '375-654-0697', 'suditu@gmail.com');

--Tabela CUSCA --

INSERT INTO CUSCA (ID\_CUSCA, NUMAR\_LOCURI)

VALUES (1, 5),

VALUES (2, 1),

VALUES (3, 4),

VALUES (4, 10),

VALUES (5, 2);

-- Tabela CERERE\_ADOPTIE --

INSERT INTO CERERE\_ADOPTIE (SERIE, ID\_ANIMAL, ID\_VIZITATOR, DATA)

VALUES ('ABC123', 1, '1', TO\_DATE('2023-01-01', 'YYYY-MM-DD')),

VALUES ('C354', 2, '2', TO\_DATE('2023-01-01', 'YYYY-MM-DD')),

VALUES ('Z782', 3, '3', TO\_DATE('2023-01-01', 'YYYY-MM-DD')),

VALUES ('A32', 4, '4', TO\_DATE('2023-01-01', 'YYYY-MM-DD')),

VALUES ('Q70', 5, '5', TO\_DATE('2023-01-01', 'YYYY-MM-DD'));

-- Tabela ANIMAL

INSERT INTO ANIMAL (ID\_ANIMAL, NUME, RASA, VARSTA, SEX, DATA\_SOSIRE, ID\_CUSCA, ID\_VETERINAR)

VALUES (1, 'Caine', 'bulldog', 3, 'Male', TO\_DATE('2023-01-01', 'YYYY-MM-DD'), 1, 1),

VALUES (2, 'Caine', 'pechinez', 13, 'Male', TO\_DATE('2023-01-01', 'YYYY-MM-DD'), 2, 2),

VALUES (3, 'Caine', 'shit-zu', 4, 'Male', TO\_DATE('2023-01-01', 'YYYY-MM-DD'), 3, 3),

VALUES (4, 'Caine', 'bichon', 9, 'Female', TO\_DATE('2023-01-01', 'YYYY-MM-DD'), 4, 4),

VALUES (5, 'Caine', 'labrador', 1, 'Female', TO\_DATE('2023-01-01', 'YYYY-MM-DD'), 5, 5);

--CERINTA 12

-- Cereri SQL complexe

-- Cererea 1: Subcereri sincronizate în care intervin cel putin 3 tabele

SELECT a.NUME, a.PRENUME, v.NUME AS NUME\_VETERINAR

FROM ANIMAL a

JOIN ANIMAL\_VETERINAR av ON a.ID\_ANIMAL = av.ID\_ANIMAL

JOIN VETERINAR v ON av.ID\_VETERINAR = v.CNP

WHERE a.RASA = 'Caine';

-- Cererea 2: Subcereri nesincronizate în clauza FROM

SELECT AVG(cantitate) AS CANTITATE\_MEDIE

FROM (SELECT CANTITATE FROM HRANA WHERE DENUMIRE\_PRODUS = 'Royal' UNION ALL

SELECT CANTITATE

FROM HRANA

WHERE DENUMIRE\_PRODUS = 'Conserva') AS sub;

-- Cererea 3: Grupari de date cu subcereri nesincronizate în care intervin cel putin 3 tabele.

SELECT s.NUME AS NUME\_SECTIE, COUNT(a.ID\_ANIMAL) AS NUMAR\_ANIMALE

FROM SECTIE\_SECTOR ss

JOIN SECTIE s ON ss.ID\_SECTIE = s.ID\_SECTIE

JOIN ANGAJAT a ON ss.ID\_ANGAJAT = a.ID\_ANGAJAT

GROUP BY s.NUME

HAVING COUNT(a.ID\_ANIMAL) > 2;

-- Cererea 4: Ordonare si utilizarea functiilor NVL si DECODE (în cadrul aceleiasi cereri)

SELECT NUME, PRENUME, NVL(TELEFON, 'Num?r de telefon necunoscut') AS TELEFON,

DECODE(SEX, 'M', 'Masculin', 'F', 'Feminin', 'Necunoscut') AS SEX

FROM VETERINAR

ORDER BY NUME ASC;

-- Cererea 5: Utilizarea a cel putin 2 functii pe siruri de caractere, 2 functii pe date calendaristice, a cel putin unei expresii CASE

SELECT

CONCAT(NUME, ' ', PRENUME) AS NUME\_COMPLET,

UPPER(RASA) AS RASA\_UPPER,

DATE\_DIFF(CURRENT\_DATE(), DATA\_SOSIRE) AS VARSTA\_ZILE,

CASE

WHEN VARSTA > 5 THEN 'Matur'

WHEN VARSTA <= 5 THEN 'Tanar'

ELSE 'Necunoscut'

END AS CATEGORIE\_VARSTA

FROM ANIMAL;

-- 13. Implementarea a 3 operatii de actualizare si de suprimare a datelor utilizând subcereri.

--Actualizarea unui angajat:

UPDATE ANGAJAT

SET NUME = 'Popescu', PRENUME = 'Ion'

WHERE ID\_ANGAJAT = (

SELECT ID\_ANGAJAT

FROM ANGAJAT

WHERE NUME = 'Ionescu' AND PRENUME = 'Maria'

LIMIT 1

);

--stergerea unui animal si a legaturilor sale:

DELETE FROM ANIMAL

WHERE ID\_ANIMAL = (

SELECT ID\_ANIMAL

FROM ANIMAL

WHERE NUME = 'Rex'

LIMIT 1

);

DELETE FROM ANIMAL\_VETERINAR

WHERE ID\_ANIMAL = (

SELECT ID\_ANIMAL

FROM ANIMAL

WHERE NUME = 'Rex'

LIMIT 1

);

DELETE FROM CARTE\_DE\_SANATATE

WHERE ID\_ANIMAL = (

SELECT ID\_ANIMAL

FROM ANIMAL

WHERE NUME = 'Rex'

LIMIT 1

);

--Actualizarea datelor unui contract:

UPDATE CONTRACT

SET DURATA = 24

WHERE NUMAR = (

SELECT NUMAR

FROM CONTRACT

WHERE TIP = 'Standard'

LIMIT 1

);

-- Cererea 14: Crearea unei vizualizari complexe ?

CREATE VIEW v\_angajati\_sectii AS

SELECT an.NUME, an.PRENUME, an.ADRESA, s.NUME AS NUME\_SECTIE

FROM ANGAJAT an

LEFT JOIN CONTRACT c ON an.ID\_CONTRACT = c.NUMAR

LEFT JOIN DIRECTOR d ON d.ID\_CONTRACT = c.NUMAR

LEFT JOIN SECTIE\_SECTOR ss ON an.ID\_ANGAJAT = ss.ID\_ANGAJAT

LEFT JOIN SECTIE s ON ss.ID\_SECTIE = s.ID\_SECTIE;

-- Cererea 15: Cerere cu outer-join pe minim 4 tabele, cerere cu operatia de division si cerere cu analiza top-n

-- Exemplu de cerere cu outer-join pe minim 4 tabele

SELECT a.NUME, c.DENUMIRE\_PRODUS, h.CANTITATE

FROM ANIMAL a

LEFT JOIN CARTE\_DE\_SANATATE cs ON a.ID\_ANIMAL = cs.ID\_ANIMAL

LEFT JOIN CONSULTATIE co ON a.ID\_ANIMAL = co.ID\_ANIMAL

LEFT JOIN HRANA h ON co.DATA = h.DATA

LEFT JOIN CUSCA c ON a.ID\_CUSCA = c.ID\_CUSCA;

-- Exemplu de cerere cu operatia de division

SELECT a.NUME, a.RASA

FROM ANIMAL a

WHERE NOT EXISTS (

SELECT \*

FROM CARTE\_DE\_SANATATE cs

WHERE a.ID\_ANIMAL = cs.ID\_ANIMAL

AND cs.VACCINURI != 'Rabie'

);

-- Exemplu de cerere cu analiza top-n

SELECT NUME, VARSTA

FROM ANIMAL

ORDER BY VARSTA DESC

FETCH FIRST 5 ROWS ONLY;

-- Cererea 16: Optimizarea unei cereri

-- Cererea înainte de optimizare

SELECT a.NUME, a.PRENUME, s.NUME AS NUME\_SECTIE

FROM ANGAJAT a, SECTIE s, CONTRACT c

WHERE a.ID\_CONTRACT = c.NUMAR

AND c.ID\_CONTRACT = d.ID\_CONTRACT

AND a.ID\_ANGAJAT = ss.ID\_ANGAJAT

AND ss.ID\_SECTIE = s.ID\_SECTIE

AND a.PRENUME = 'John';

-- Cererea dupa optimizare

SELECT a.NUME, a.PRENUME, s.NUME AS NUME\_SECTIE

FROM ANGAJAT a

JOIN CONTRACT c ON a.ID\_CONTRACT = c.NUMAR

JOIN DIRECTOR d ON c.ID\_CONTRACT = d.ID\_CONTRACT

JOIN SECTIE\_SECTOR ss ON a.ID\_ANGAJAT = ss.ID\_ANGAJAT

JOIN SECTIE s ON ss.ID\_SECTIE = s.ID\_SECTIE

WHERE a.PRENUME = 'John';

-- Cererea 17a: Realizarea normalizarii BCNF, FN4, FN5

-- Cererea 17b: Aplicarea denormalizarii, justificând necesitatea acesteia

-- Exemplu de denormalizare prin ad?ugarea unei coloane redundante în tabelul ANIMAL

ALTER TABLE ANIMAL ADD COLUMN NUME\_VETERINAR VARCHAR(255);

UPDATE ANIMAL a

SET NUME\_VETERINAR = (

SELECT v.NUME

FROM VETERINAR v

WHERE v.CNP = a.ID\_VETERINAR

);

**PROIECT BAZE DE DATE - ADAPOST DE ANIMALE**

1. **Descrierea modelului real, a utilității acestuiași a regulilor de funcționare.**

Acest proiect urmareste realizarea design-ului unei baze de date a unui adapost de animale. Utilitatea acestui design este aceea ca se poate tine cu usurinta evidenta angajatilor (veterinari, directori) din adapost, dar si a unor consumabile (medicamente, hrana), sau documente (cereri de adoptie).

In ceea ce priveste organizarea, adapostul este impartit in sectii, iar fiecare sectie exista sectoare in care se afla custile animalelor adapostite. Veterinarii pot face consultatii si noteaza in carnetul veterinary al fiecarui animal toate datele necesare.

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

Constrângerile din codul furnizat includ: chei primare, chei externe, restricții de referință, chei primare compuse, restricții de unicitate și restricții de integritate referențială. Acestea asigură unicitatea valorilor, relațiile între tabele, respectarea valorilor cheilor primare și integritatea datelor.

Cheie primară: O constrângere de cheie primară asigură că o coloană sau un set de coloane dintr-o tabelă au valori unice și identifică în mod unic fiecare înregistrare. De exemplu, o constrângere de cheie primară ar putea fi aplicată coloanei "id" dintr-o tabelă "Angajat", astfel încât fiecare angajatsă aibă un ID unic.

Cheie externă: O constrângere de cheie externă definește relații între tabele, asigurând integritatea referențială. De exemplu, o constrângere de cheie externă ar putea fi aplicată coloanei "id\_angajat" dintr-o tabelă "Contract", referindu-se la coloana "id" din tabela "Angajat", pentru a asigura că fiecare comandă este asociată cu un utilizator valid.

Restricție de unicitate: O restricție de unicitate asigură că o coloană sau un set de coloane dintr-o tabelă nu conține valori duplicate. De exemplu, o restricție de unicitate ar putea fi aplicată coloanei "CNP" dintr-o tabelă "Vizitator", astfel încât fiecare vizitator să aibe un CNP unic.

1. **Descrierea entităților, incluzând precizarea cheii primare.**

Tabela CONTRACT:

Cheia primară: NUMAR (de tip INT)

Tabela ANGAJAT:

Cheia primară: ID\_ANGAJAT (de tip INT)

Tabela DIRECTOR:

Cheia primară: ID\_DIRECTOR (de tip INT)

Tabela SECTIE:

Cheia primară: ID\_SECTIE (de tip INT)

Tabela SECTOR:

Cheia primară: ID\_SECTOR (de tip INT)

Tabela SECTIE\_SECTOR:

Cheile primare: ID\_SECTIE și ID\_SECTOR (ambele de tip INT)

Tabela FARMACIE:

Cheia primară: ID\_FARMACIE (de tip INT)

Tabela MEDICAMENT:

Cheia primară: NUMAR (de tip INT)

Tabela VIZITATOR:

Cheia primară: CNP (de tip VARCHAR(13))

Tabela CERERE\_ADOPTIE:

Cheia primară: SERIE, ID\_ANIMAL, ID\_VIZITATOR (toate de tip VARCHAR(255), INT, VARCHAR(13) respectiv)

Tabela ANIMAL:

Cheia primară: ID\_ANIMAL (de tip INT)

Tabela CUSCA:

Cheia primară: ID\_CUSCA (de tip INT)

Tabela VETERINAR:

Cheia primară: CNP (de tip VARCHAR(13))

Tabela MAGAZIN:

Cheia primară: ID\_MAGAZIN (de tip INT)

Tabela HRANA:

Cheia primară: NUMAR (de tip INT)

Tabela CARTE\_DE\_SANATATE:

Cheia primară: ID\_ANIMAL (de tip INT)

Tabela CONSULTATIE:

Cheia primară: ID\_CONSULTATIE (de tip INT)

Tabela CLINICA\_VETERINARA:

Cheia primară: ID\_CLINICA (de tip INT)

Vizualizarea v\_angajati\_sectii:

Această vizualizare nu are o cheie primară, deoarece este o vizualizare rezultată din interogarea datelor din mai multe tabele.

1. **Descrierea relațiilor, incluzând precizarea cardinalității acestora.**

Tabelul CONTRACT are o relație de tip 1 la N cu tabelul ANGAJAT. Un contract poate fi asociat unui singur angajat, iar un angajat poate avea asociat unul sau mai multe contracte.

Tabelul ANGAJAT are o relație de tip 1 la 1 cu tabelul CONTRACT. Fiecare angajat este asociat cu un singur contract.

Tabelul DIRECTOR are o relație de tip 1 la 1 cu tabelul CONTRACT. Fiecare director este asociat cu un singur contract.

Tabelul SECTIE are o relatie cu tabela Angajat si una cu tabela SECTIE\_SECTOR.

Tabelul SECTOR are relatie atat cu tablea SECTIE\_SECTOR cat si cu tabela CUSCA.

Tabelul SECTIE\_SECTOR are o relație de tip N la N cu tabelele SECTIE și SECTOR. O sectie poate fi asociata cu mai multe sectoare, iar un sector poate fi asociat cu mai multe sectii.

Tabelul FARMACIE are relațiE cu tabelele CONTRACT si FARMACIE.

Tabelul MEDICAMENT are o relație de tip 1 la N cu tabelul FARMACIE. Un medicament poate fi asociat unei singure farmacii, iar o farmacie poate avea asociate unul sau mai multe medicamente.

Tabelul VIZITATOR nu are relații cu alte tabele.

Tabelul CERERE\_ADOPTIE are o relație de tip N la N cu tabelele ANIMAL și VIZITATOR. O cerere de adoptie poate fi asociata cu mai multe animale si vizitatori, iar un animal si un vizitator pot fi asociati cu mai multe cereri de adoptie.

Tabelul ANIMAL are o relație de tip 1 la 1 cu tabelul CUSCA. Fiecare animal este asociat cu o singură cusca.

Tabelul CUSCA are relații cu tabelele SECTOR si ANIMAL.

Tabelul VETERINAR are o relație de tip 1 la N cu tabelul ANIMAL. Un veterinar poate fi asociat cu unul sau mai multe animale, iar un animal poate fi asociat cu un singur veterinar.

Tabelul MAGAZIN are relații cu tabelea HRANA si tabela CONTRACT.

Tabelul HRANA are o relație de tip 1 la N cu tabelul MAGAZIN. Un produs de hrana poate fi asociat unui singur magazin, iar un magazin poate avea asociate unul sau mai multe produse de hrana.

Tabelul CARTE\_DE\_SANATATE are o relație de tip 1 la 1 cu tabelul ANIMAL. Fiecare animal are asociată o singură carte de sănătate.

Tabelul CONSULTATIE are relatie cu tabela ANIMALl si cu tabela VETERINAR.

Tabelul CLINICA\_VETERINARA are relatie cu tabelele VETERINAR si tabela CONTRACT.

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

Tabelul CONTRACT:

Atribute: NUMAR (cheie primară), DATA\_EMITERE, DURATA, TIP, id\_angajat (cheie externă referențiază ANGAJAT.ID\_ANGAJAT).

Tabelul ANGAJAT:

Atribute: ID\_ANGAJAT (cheie primară), NUME, PRENUME, CNP, ADRESA, TELEFON, EMAIL, ID\_CONTRACT (cheie externă referențiază CONTRACT.NUMAR).

Tabelul DIRECTOR:

Atribute: ID\_DIRECTOR (cheie primară), ID\_CONTRACT (cheie externă referențiază CONTRACT.NUMAR).

Tabelul SECTIE:

Atribute: ID\_SECTIE (cheie primară), NUME.

Tabelul SECTOR:

Atribute: ID\_SECTOR (cheie primară), TIP\_NEVOIE, NUMAR\_LOCURI.

Tabelul SECTIE\_SECTOR:

Atribute: ID\_SECTIE, ID\_SECTOR (cheie primară compusă), ID\_SECTIE (cheie externă referențiază SECTIE.ID\_SECTIE), ID\_SECTOR (cheie externă referențiază SECTOR.ID\_SECTOR).

Tabelul FARMACIE:

Atribute: ID\_FARMACIE (cheie primară), NUME, TELEFON, PROGRAM.

Tabelul MEDICAMENT:

Atribute: NUMAR (cheie primară), DENUMIRE\_PRODUS, CANTITATE, ID\_FARMACIE (cheie externă referențiază FARMACIE.ID\_FARMACIE).

Tabelul VIZITATOR:

Atribute: CNP (cheie primară), NUME, PRENUME, TELEFON, EMAIL.

Tabelul CERERE\_ADOPTIE:

Atribute: SERIE, ID\_ANIMAL, ID\_VIZITATOR, DATA, APROBATA\_DE\_DIRECTOR, APROBATA\_DE\_ASOCIATIE (cheie primară compusă), ID\_ANIMAL (cheie externă referențiază ANIMAL.ID\_ANIMAL), ID\_VIZITATOR (cheie externă referențiază VIZITATOR.CNP).

Tabelul ANIMAL:

Atribute: ID\_ANIMAL (cheie primară), NUME, RASA, VARSTA, SEX, DATA\_SOSIRE, ID\_CUSCA (cheie externă referențiază CUSCA.ID\_CUSCA), ID\_VETERINAR (cheie externă referențiază VETERINAR.CNP).

Tabelul CUSCA:

Atribute: ID\_CUSCA (cheie primară), NUMAR\_LOCURI.

Tabelul VETERINAR:

Atribute: CNP (cheie primară), NUME, PRENUME, ID\_ANIMAL (cheie externă referențiază ANIMAL.ID\_ANIMAL).

Tabelul MAGAZIN:

Atribute: ID\_MAGAZIN (cheie primară), NUME, TELEFON, PROGRAM.

Tabelul HRANA:

Atribute: ID\_HRANA (cheie primară), DENUMIRE, CANTITATE, ID\_MAGAZIN (cheie externă referențiază MAGAZIN.ID\_MAGAZIN).

Tabelul COMANDA:

Atribute: ID\_COMANDA (cheie primară), DATA, SUMA\_TOTALA, ID\_MAGAZIN (cheie externă referențiază MAGAZIN.ID\_MAGAZIN).

Tabelul DETALII\_COMANDA:

Atribute: ID\_COMANDA, ID\_HRANA (cheie primară compusă), ID\_COMANDA (cheie externă referențiază COMANDA.ID\_COMANDA), ID\_HRANA (cheie externă referențiază HRANA.ID\_HRANA).

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

A computer screen shot of a computer screen

Description automatically generated with low confidence

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

A screenshot of a computer

Description automatically generated

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

* Contract (numar, data\_emitere, durata, id\_angajat)
* Angajat (id\_angajat, nume, prenume, cnp, adresa, telefon, email, id\_contract)
* Director (id\_director, id\_contract)
* Sectie (id\_sectie, nume)
* Sector (id\_sector, tip\_nevoie, numar\_locuri)
* Sectie\_sector (id\_sectie, id\_sector)
* Farmacie (id\_farmacie, nume, telefon, program)
* Medicament (numar, denumire\_produs, cantitate, id\_farmacie)
* Vizitator (cnp, nume, prenume, telefon, email)
* Cerere\_adoptie (serie, id\_animal, id\_vizitator, data, aprobata\_de\_director, aprobata\_de\_asociatie)
* Animal (id\_animal, nume, rasa, varsta, sex, data\_sosire, id\_cusca, id\_veterinar)
* Cusca (id\_cusca, numar\_locuri)
* Veterinar (cnp, nume, prenume, id\_animal, id\_veterinar)
* Magazin (id\_magazin, nume, telefon, program)
* Hrana (numar, denumire\_produs, cantitate, id\_magazin)

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

Forma normală de la 1 (FN1) implică eliminarea valorilor repetate și atomizarea atributelor într-o singură valoare. În codul tău, FN1 este respectată prin eliminarea duplicatelor și atomizarea datelor în tabelele tale. De exemplu, în tabela ANGAJAT, fiecare angajat are o valoare unică pentru ID\_ANGAJAT și nu există duplicări ale acestui atribut.

Forma normală de la 2 (FN2) implică eliminarea dependențelor funcționale parțiale și asigurarea dependențelor funcționale complete prin adăugarea cheilor primare în tabelele de referință. În codul tău, FN2 este respectată prin utilizarea cheilor primare și străine pentru a stabili relații între tabele. De exemplu, tabela ANGAJAT are o cheie străină ID\_CONTRACT care face referire la tabela CONTRACT, asigurând astfel dependența funcțională completă între aceste tabele.

Forma normală de la 3 (FN3) implică eliminarea dependențelor tranzitive și asigurarea dependențelor funcționale directe între toate atributele. În codul tău, FN3 pare să fie respectată, deoarece nu există dependențe tranzitive evidente și relațiile între tabelele tale par să fie bine definite.

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

Link spre inserarea in tabele:

[https://github.com/alexandramocanu1/Collag](https://github.com/alexandramocanu1/Collage-work/blob/main/Semestre 2/proiect_BD/inserari.sql)[e-work/blob/main/Semestre%202/proiect\_BD/inserari.sql](https://github.com/alexandramocanu1/Collage-work/blob/main/Semestre 2/proiect_BD/inserari.sql)

A picture containing text, software, computer icon, web page

Description automatically generated

**11. Crearea tabelelor in SQL si inserarea de date coerente in fiecare dintre acestea**

**Link spre codul de creare a tabelelor**: <https://github.com/alexandramocanu1/Collage-work/blob/main/Semestre%202/proiect_BD/adapostul_de_animale.sql>



**12. Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:**

* subcereri sincronizate în care intervin cel puțin 3 tabele subcereri nesincronizate în clauza FROM
* grupări de datecu subcereri nesincronizate in care intervin cel putin 3 tabele, funcții grup, filtrare la nivel degrupuri(in cadrul aceleiasi cereri)
* ordonărisi utilizarea funcțiilor NVL și DECODE (in cadrul aceleiasi 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)Observație:Într-o cerere se vor regăsi mai multe elemente dintre cele enumerate mai sus, astfel încât cele 5 cereri să le cuprindă pe toate.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

A picture containing text, screenshot, software, computer icon

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A computer screen shot of a computer

Description automatically generated with low confidence

**15. Formulați în limbaj natural și implementați în SQL.**

A screenshot of a computer

Description automatically generated

**16. Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile operatorilor algebrei relaționale. Cererea va fi exprimatăprin expresie algebrică, arbore algebric și limbaj (SQL), atât anterior cât și ulterior optimizării.**

A screenshot of a computer

Description automatically generated

**17.**

a. Realizarea normalizării BCNF, FN4, FN5.

b. Aplicarea denormalizării, justificând necesitatea acesteia.

A screenshot of a computer

Description automatically generated