



UNIVERSITATEA TEHNICĂ “GH ASACHI” IAȘI
FACULTATEA AUTOMATICĂ ȘI CALCULATOARE
SPECIALIZAREA CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
DISCIPLINA BAZE DE DATE PROIECT

GESTIUNEA UNUI CENTRU DE ANIMALE

Coordonator,

Prof. Cătălin Mironeanu

Student,

Pristanda Amalia-Maria

Iași, 2021

Titlu proiect: Gestiunea unui centru de animale

Analiza, proiectarea și implementarea unei baze de date și a aplicației aferente care să modeleze activitatea unui centru de animale cu privire la gestiunea procesului de adopție.

Descrierea cerințelor și modul de organizare al proiectului

Activitatea de gestiune al unui centru de animale implică un efort considerabil în ceea ce privește numeroase documente care țin de înregistrarea animalelor, a relațiilor în care se află acestea cu centrul (dacă au fost adoptate sau nu) și de monitorizare permanentă a stării de sănătate.

Informațiile de care avem nevoie sunt cele legate de:

- **animale**: datele referitoare la animale se vor stoca în 3 tabele. Astfel, într-o tabelă părinte vom stoca informațiile cu privire la tipul de animale; ne interesează să știm tipul animalului și denumirea sa. De aici va o relație one-to-many către Animale, care conține informații legate de data nașterii, data aducerii și data adopției (unde este cazul). De asemenea, va mai exista o tabelă, Detalii animale, în care vor fi păstrate mai multe informații (sex, nume, culoare, talie, eventual rasa) despre animalul respectiv. Această tabelă va avea o relație one-to-one către Animale.

- **client**: este considerată persoana care adoptă un animal de la centru. Pentru aceasta ne interesează CNP-ul (clienții trebuie să aibă măcar 18 ani sau să împlinească această vârstă pe parcursul anului curent), numele și adresa. Se consideră că animalele sunt adoptate pe viață.

- **fișa medicală**: fiecare animal este dus la un control anual (dacă nu apar alte probleme) cât timp este în grija centrului. În cazul unei probleme de sănătate se va înregistra atât denumirea bolii cât și tratamentul administrat. Altfel, se înregistrează doar data controlului și faptul că animalul este sănătos.

- **cușcă**: fiecărui animal îi este atribuită o cușcă de o anumită dimensiune la venirea în centru, aceasta putând fi schimbată ulterior. Pentru fiecare cușcă în parte trebuie să cunoaștem starea ei (ocupată sau liberă)

Descrierea funcțională a aplicației

Principalele funcții permise de aplicație:

- ✓ Evidența animalelor din centru
- ✓ Evidența adopțiilor

Descrierea detaliată a modelului logic

Entitățile din modelul logic sunt:

- ✓ Tipuri_animale;
- ✓ Cusca;
- ✓ Animale;
- ✓ Detalii_animale;
- ✓ Fisa_medicala;
- ✓ Client

În proiectarea acestei baze de date s-au identificat tipurile de relații 1:n, n:1 și 1:1.

Între entitățile **Tipuri_animale** și **Animale** se întâlnește o relație de tip **1:n** deoarece un tip cuprinde mai multe animale. Reciproca nu este valabilă, deoarece un animal nu poate fi de mai multe tipuri (spre exemplu, nu poate să fie și pisică și câine în același timp). Legătura dintre cele două tabele este realizată prin câmpul **id_tip**.

Între **Cusca** și **Animale** se stabilește o relație de tip **n:1** deoarece de-a lungul timpului într-o cușcă pot sta mai multe animale, dar doar un animal poate ocupa o cușcă la un moment dat. Legătura între cele două se realizează prin câmpul **id_cusca**.

Între **Animale** și **Detalii_animale** există o legătură **1:1**. Fiecărui animal îi corespunde un singur set de date și reciproc. Legătura dintre cele două entități este realizată prin câmpul **id_animal**.

Între entitățile **Fisa_medicala** și **Animale** este o legătură de tip **1:n** deoarece o fișă medicală îi corespunde unui singur animal, dar un animal poate avea mai multe fișe medicale. Legătura dintre cele două se realizează prin câmpul **id_animal**.

Între **Animale** și **Client** se întâlnește o legătură de tip **1:n** deoarece un animal poate fi adoptat de o singură persoană, dar o persoană poate adopta mai multe animale. Câmpul comun între aceste două entități este **cnp**.

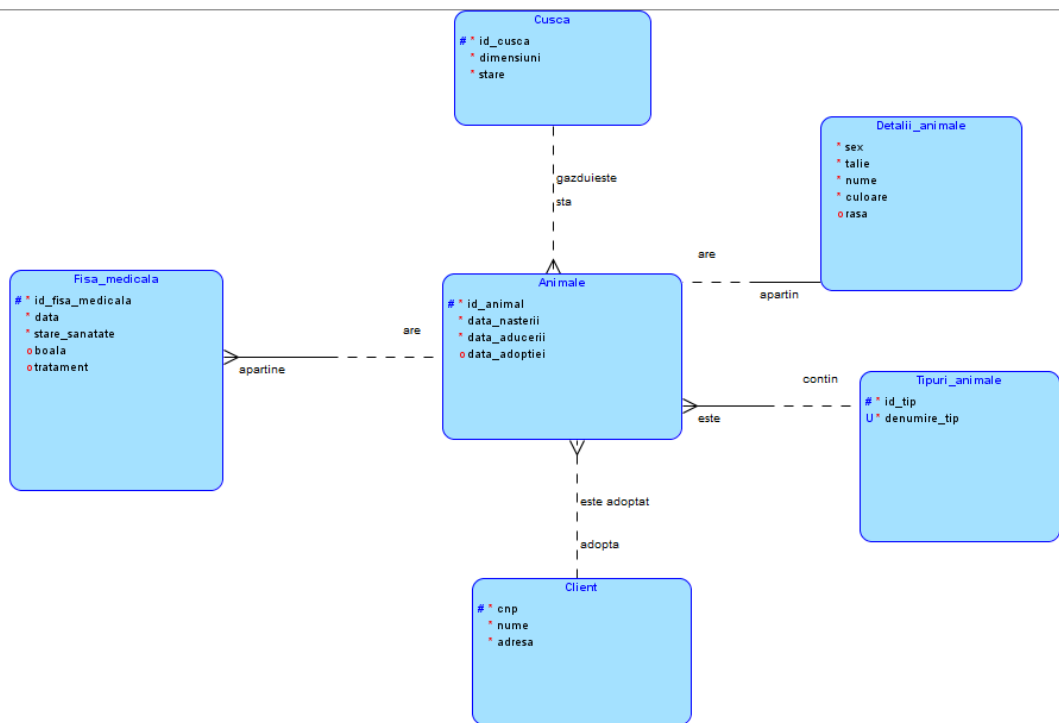


Fig1: Modelul logic

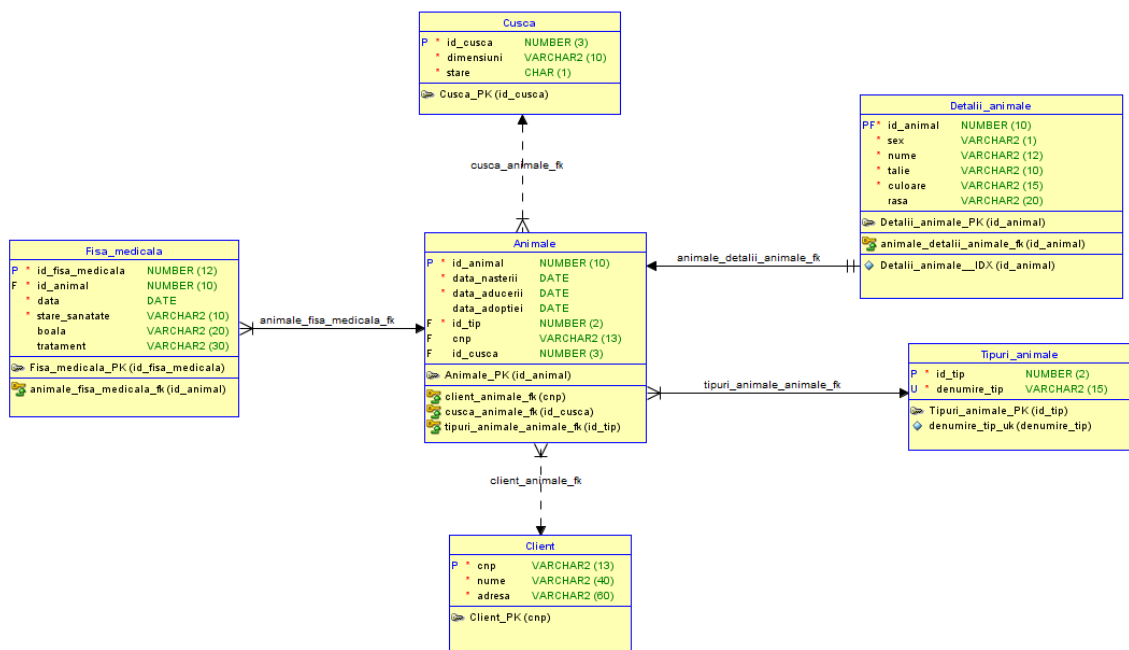


Fig2: Modelul fizic

Descrierea detaliată a modelului fizic

Tabelele din modelul fizic sunt:

- ✓ Tipuri_animale;
- ✓ Cusca;
- ✓ Animale;
- ✓ Detalii_animale;
- ✓ Fisa_medicala;
- ✓ Client

Relațiile sunt aceleași ca și în modelul logic. În schimb, există unele completări.

Tabela **Animale** are un atribut, id_animale care este generat prin autoincrement. De asemenea, există 3 trigger-uri:

- *Trg_animale_data_aducerii* si *Trg_Animale_data_nasterii* – verifică ca data aducerii, respectiv data nașterii unui animal sa nu fie mai mari decât data actuală;
- *Trg_animale_cusca* – verifică dacă este liberă cușca pe care dorim să o asignăm animalului, fie când ajunge în centru, fie în urma unei mutări ulterioare. Dacă da, atunci verifică ca și dimensiunea să fie corespunzătoare. Dacă ambele sunt corecte atunci marchează cușca ca fiind ocupată și o eliberează pe cea veche dacă este cazul. În plus, verifică dacă este introdusă data adopției. Dacă da, atunci verifică dacă animalul a fost adoptat deja. Dacă nu, cușca ocupată este eliberată și poate fi folosită din nou de alt animal

Tabela **Fisa_medicala** are la rândul ei un atribut care este generat automat, și anume id_fisa_medicala.

Există și aici un trigger, *Trg_fisa_medicala_data*, care se asigură ca data unei fișe medicale să nu fie mai mare decât data curentă.

Tabela **Client** conține un trigger, *Trg_client_cnp*, care verifică ca un client să aibă împliniți 18 ani sau să împlinească această vârstă în decursul anului curent.

Atât atributul id_tip din tabela **Tipuri_animale**, cât și atributul id_cusca din tabela **Cusca** sunt generate automat prin autoincrement.

Constrângeri utilizate

În tabela **Animale**:

- Primary key: id_animal - folosit pentru a identifica fiecare animal
- Not Null:
 - data_nasterii – se presupune că fiecare animal are una (poate fi și aproximată). Clienții pot dori să adopte un animal mai tânăr sau mai bătrân
 - data_aducerii – pentru a vedea de cât timp este un animal în centru
 - id_tip – fiecare animal este de un anumit tip
- Check:
 - Animale_data_aducerii_ck – data aducerii unui animal la centru trebuie să fie cel puțin egală cu data nașterii
 - Animale_data_adopției_ck – data adopției unui animal trebuie să fie mai mare decât data aducerii la centru a acestuia
- Foreign Key:
 - cnp – poate fi NULL dacă animalul nu a fost adoptat. Altfel, este folosit pentru a identifica persoana ce a adoptat animalul respectiv
 - id_tip – pentru a vedea care este tipul animalului
 - id_cusca – folosit pentru a vedea în ce cușcă este ținut animalul

În tabela **Client**:

- Primary key: cnp – folosit pentru a identifica fiecare client
- Not Null: nume, adresa – informații necesare pentru evidența clienților
- Check:
 - Client_cnp_ck – folosește un regex pentru a verifica dacă cnp-ul este unul valid
 - Client_nume_ck – numele clientului trebuie să aibă o lungime de măcar 3 caractere și să conțină un spațiu
 - Client_adresa_ck – adresa trebuie să conțină măcar orașul, strada și codul poștal

În tabela **Cusca**:

- Primary key: id_cusca - folosit pentru a identifica fiecare cușcă

- Not Null: dimensiuni, stare – informații necesare pentru coerența datelor
- Check:
 - Cusca_stare_ck – starea poate fi doar 0(libera) sau 1(ocupat)
 - Cusca_dimensiuni_ck – cuștile au dimensiuni prestabilite: 1x1m,1.5x1.5m,2x2m

În tabela **Detalii_animale**:

- Primary key: id_animal
- Not Null:
 - sex, talie, culoare – unii clienți pot dori să adopte un animal în funcție de aceste criterii
 - nume – fiecare animal trebuie să răspundă la un nume
- Check:
 - Detalii_animale_sex_ck – sexul poate fi doar F(feminin) si M(masculin)
 - Detalii_animale_nume_ck – numele animalelor trebuie să aibă măcar 3 caractere
 - Detalii_animale_talie_ck – talia poate lua valori doar din lista (mica,medie,mare)
- Foreign Key:
 - id_animal – folosit pentru a vedea cărui animal îi aparțin detaliile respective

În tabela **Fisa_medicala**:

- Primary key: id_fisa_medicala
- Not Null:
 - data, stare_sanatate, id_animal – informații necesare unei fișe medicale
- Check:
 - Fisa_medicala_stare_sanatate – starea poate fi doar ‘bolnav’ sau ‘sanatos’
- Foreign Key:
 - id_animal – pentru a vedea cărui animal îi corespunde o anumită fișă medicală

În tabela **Tipuri_animale**:

- Primary key: id_tip
- Check:
 - Tip_anim_denumire_tip – în centru pot fi primiți doar câini, pisici, iepuri, chinchilla și papagali
- Unique: denumire_tip_uk – denumirea tipului animalelor trebuie să fie unică pentru a nu avea la tipuri două câmpuri ‘caine’ spre exemplu

