**Ministerul Educaţiei şi Cercetării**

**al Republicii Moldova**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

RAPORT

la lucrarea de laborator nr. 10

La disciplina: Baze de Date

Tema: “Crearea si actualizarea tabelelor in SQL “

A efectuat:     Buza Cătălin TI-214

A verificat:    Cebotar Gabriela

­­

Chișinău 2023

Cerinte catre BD proprie:

-Pentru definirea atributelor se vor utiliza tipuri de date cat mai diverse (numerice diverse, caracteriale, temporale, logice), sa nu se limiteze doar la INTEGER si VARCHAR.

DROP TABLE IF EXISTS Hrana

IF OBJECT\_ID('Hrana') IS NULL

CREATE TABLE Hrana (

Id\_hrana INT IDENTITY(1, 1) NOT NULL,

Tip\_hrana VARCHAR(50) NOT NULL,

Pret FLOAT NOT NULL

);

DROP TABLE IF EXISTS Donatori

IF OBJECT\_ID('Donatori') IS NULL

CREATE TABLE Donatori (

Id\_donator INT IDENTITY(1, 1) NOT NULL,

Idnp\_donator CHAR(13) NOT NULL UNIQUE,

Nume VARCHAR(50) NOT NULL,

Prenume VARCHAR(50) NOT NULL,

Adresa VARCHAR(100) NOT NULL,

Oras VARCHAR(50) NOT NULL,

Email VARCHAR(50) NOT NULL CHECK (Email LIKE '%@%') UNIQUE,

Telefon VARCHAR(20) NOT NULL CHECK (Telefon LIKE '+373[0-9]%') UNIQUE,

Gen CHAR(1) NOT NULL CHECK (gen IN ('M', 'F')),

Data\_nastere DATE NOT NULL,

CONSTRAINT Pk\_Id\_donator PRIMARY KEY(Id\_donator, Idnp\_donator)

);

În scriptul furnizat, este creată o tabelă numită "Donatori" și ’’Hrana’’ pentru a gestiona informații despre donatori. Iată o explicație detaliată a fiecărui atribut al acestor tabele:

Tabelul Hrana:

1. **Id\_hrana (INT IDENTITY(1, 1) NOT NULL):**
   * Este un identificator unic al tipului de hrană.
   * Este de tip întreg (INTEGER) și este incrementat automat cu 1 pentru fiecare înregistrare nouă, datorită specificației "IDENTITY(1, 1)".
   * Nu poate fi nul, conform specificației "NOT NULL".
2. **Tip\_hrana (VARCHAR(50) NOT NULL):**
   * Reprezintă tipul de hrană.
   * Este de tip caracter variabil (VARCHAR) cu o lungime maximă de 50 de caractere.
   * Nu poate fi nul, conform specificației "NOT NULL".
3. **Pret (FLOAT NOT NULL):**
   * Reprezintă prețul asociat tipului de hrană.
   * Este de tip float și poate conține valori cu zecimale.
   * Nu poate fi nul, conform specificației "NOT NULL".

Tabelul Donator:

1. **Id\_donator (INT IDENTITY(1, 1) NOT NULL):**
   * Este un identificator unic al donatorului.
   * Este de tip întreg (INTEGER) și este incrementat automat cu 1 pentru fiecare înregistrare nouă datorită specificației "IDENTITY(1, 1)".
2. **Idnp\_donator (CHAR(13) NOT NULL UNIQUE):**
   * Reprezintă Codul Numeric Personal al donatorului.
   * Este de tip caracter (CHAR) cu o lungime fixă de 13 caractere.
   * Nu poate fi nul și trebuie să fie unic, conform specificației "NOT NULL UNIQUE".
3. **Nume (VARCHAR(50) NOT NULL):**
   * Reprezintă numele donatorului.
   * Este de tip caracter variabil (VARCHAR) cu o lungime maximă de 50 de caractere.
   * Nu poate fi nul, conform specificației "NOT NULL".
4. **Prenume (VARCHAR(50) NOT NULL):**
   * Reprezintă prenumele donatorului.
   * Este de tip caracter variabil (VARCHAR) cu o lungime maximă de 50 de caractere.
   * Nu poate fi nul, conform specificației "NOT NULL".
5. **Adresa (VARCHAR(100) NOT NULL):**
   * Reprezintă adresa donatorului.
   * Este de tip caracter variabil (VARCHAR) cu o lungime maximă de 100 de caractere.
   * Nu poate fi nul, conform specificației "NOT NULL".
6. **Oras (VARCHAR(50) NOT NULL):**
   * Reprezintă orașul în care locuiește donatorul.
   * Este de tip caracter variabil (VARCHAR) cu o lungime maximă de 50 de caractere.
   * Nu poate fi nul, conform specificației "NOT NULL".
7. **Email (VARCHAR(50) NOT NULL CHECK (Email LIKE '%@%') UNIQUE):**
   * Reprezintă adresa de e-mail a donatorului.
   * Este de tip caracter variabil (VARCHAR) cu o lungime maximă de 50 de caractere.
   * Trebuie să conțină simbolul "@" în adresa de e-mail.
   * Nu poate fi nul și trebuie să fie unic, conform specificațiilor "NOT NULL CHECK" și "UNIQUE".
8. **Telefon (VARCHAR(20) NOT NULL CHECK (Telefon LIKE '+373[0-9]%') UNIQUE):**
   * Reprezintă numărul de telefon al donatorului.
   * Este de tip caracter variabil (VARCHAR) cu o lungime maximă de 20 de caractere.
   * Trebuie să înceapă cu "+373" și să urmeze de cel puțin un caracter numeric.
   * Nu poate fi nul și trebuie să fie unic, conform specificațiilor "NOT NULL CHECK" și "UNIQUE".
9. **Gen (CHAR(1) NOT NULL CHECK (gen IN ('M', 'F'))):**
   * Reprezintă genul donatorului.
   * Este de tip caracter (CHAR) cu o lungime fixă de 1 caracter.
   * Nu poate fi nul și poate fi doar 'M' (masculin) sau 'F' (feminin), conform specificațiilor "NOT NULL CHECK".
10. **Data\_nastere (DATE NOT NULL):**
    * Reprezintă data nașterii a donatorului.
    * Este de tip dată (DATE).
    * Nu poate fi nul, conform specificației "NOT NULL".
11. **CONSTRAINT Pk\_Id\_donator PRIMARY KEY(Id\_donator, Idnp\_donator):**
    * Defineste o cheie primară compusă din Id\_donator și Idnp\_donator. Aceasta asigură unicitatea combinației acestor două câmpuri în tabel.

Sarcina data a fost îndeplinită deoarece am folosit mai multe tipuri de date nu doar INT și VARCHAR spre exemplu : CHAR , DATE , iar în tabela Hrana am folosit și FLOAT.

-Printre tipurile utilizate trebuie sa se regaseasca si IDENTITY (autoincrementare), si COMPUTED COLUMN (campuri calculate).

DROP TABLE IF EXISTS Adoptanti

IF OBJECT\_ID('Adoptanti') IS NULL

CREATE TABLE Adoptanti (

Id\_adoptant INT IDENTITY(1, 1) NOT NULL,

Idnp\_adoptant CHAR(13) NOT NULL,

Nume VARCHAR(50) NOT NULL,

Prenume VARCHAR(50) NOT NULL,

Adresa VARCHAR(100) NOT NULL,

Oras VARCHAR(50) NOT NULL DEFAULT 'Chisinau',

Email VARCHAR(50) NOT NULL CHECK (Email LIKE '%@%')UNIQUE,

Telefon VARCHAR(20) NOT NULL CHECK (Telefon LIKE '+373[0-9]%')UNIQUE,

Data\_adoptiei DATE NOT NULL,

Data\_nastere DATE NOT NULL,

Varsta AS DATEDIFF(YEAR, Data\_nastere, GETDATE()),

CONSTRAINT Pk\_Id\_adoptanti PRIMARY KEY(Id\_adoptant,Idnp\_adoptant),

CONSTRAINT UC\_adoptant UNIQUE (Idnp\_adoptant)

);

În scriptul SQL furnizat pentru crearea tabelului "Adoptanti", sunt utilizate tipuri diverse de coloane, inclusiv IDENTITY pentru un câmp de autoincrementare și o coloană COMPUTED pentru calculul vârstei. Iată o explicație detaliată a fiecărui atribut :

1. **Id\_adoptant (INT IDENTITY(1, 1) NOT NULL):**
   * Este un identificator unic al adoptantului.
   * Este de tip întreg (INTEGER) și este configurat pentru autoincrementare cu 1 pentru fiecare înregistrare nouă, datorită specificației "IDENTITY(1, 1)".
   * Nu poate fi nul, conform specificației "NOT NULL".
2. **Idnp\_adoptant (CHAR(13) NOT NULL):**
   * Reprezintă Codul Numeric Personal al adoptantului.
   * Este de tip caracter (CHAR) cu o lungime fixă de 13 caractere.
   * Nu poate fi nul, conform specificației "NOT NULL".
3. **Nume (VARCHAR(50) NOT NULL):**
   * Reprezintă numele adoptantului.
   * Este de tip caracter variabil (VARCHAR) cu o lungime maximă de 50 de caractere.
   * Nu poate fi nul, conform specificației "NOT NULL".
4. **Prenume (VARCHAR(50) NOT NULL):**
   * Reprezintă prenumele adoptantului.
   * Este de tip caracter variabil (VARCHAR) cu o lungime maximă de 50 de caractere.
   * Nu poate fi nul, conform specificației "NOT NULL".
5. **Adresa (VARCHAR(100) NOT NULL):**
   * Reprezintă adresa adoptantului.
   * Este de tip caracter variabil (VARCHAR) cu o lungime maximă de 100 de caractere.
   * Nu poate fi nul, conform specificației "NOT NULL".
6. **Oras (VARCHAR(50) NOT NULL DEFAULT 'Chisinau'):**
   * Reprezintă orașul în care locuiește adoptantul.
   * Este de tip caracter variabil (VARCHAR) cu o lungime maximă de 50 de caractere.
   * Nu poate fi nul, conform specificației "NOT NULL", iar valoarea implicită este 'Chisinau' atunci când nu este furnizată o valoare.
7. **Email (VARCHAR(50) NOT NULL CHECK (Email LIKE '%@%') UNIQUE):**
   * Reprezintă adresa de e-mail a adoptantului.
   * Este de tip caracter variabil (VARCHAR) cu o lungime maximă de 50 de caractere.
   * Trebuie să conțină simbolul "@" în adresa de e-mail.
   * Nu poate fi nul și trebuie să fie unic, conform specificațiilor "NOT NULL CHECK" și "UNIQUE".
8. **Telefon (VARCHAR(20) NOT NULL CHECK (Telefon LIKE '+373[0-9]%') UNIQUE):**
   * Reprezintă numărul de telefon al adoptantului.
   * Este de tip caracter variabil (VARCHAR) cu o lungime maximă de 20 de caractere.
   * Trebuie să înceapă cu "+373" și să urmeze de cel puțin un caracter numeric.
   * Nu poate fi nul și trebuie să fie unic, conform specificațiilor "NOT NULL CHECK" și "UNIQUE".
9. **Data\_adoptiei (DATE NOT NULL):**
   * Reprezintă data la care a avut loc adoptarea.
   * Este de tip dată (DATE).
   * Nu poate fi nul, conform specificației "NOT NULL".
10. **Data\_nastere (DATE NOT NULL):**
    * Reprezintă data nașterii adoptantului.
    * Este de tip dată (DATE).
    * Nu poate fi nul, conform specificației "NOT NULL".
11. **Varsta AS DATEDIFF(YEAR, Data\_nastere, GETDATE()):**
    * Este o coloană COMPUTED care calculează vârsta adoptantului în funcție de diferența dintre data curentă (GETDATE()) și data nașterii.
    * Aceasta este o modalitate de a calcula automat vârsta în funcție de data nașterii.
12. **CONSTRAINT Pk\_Id\_adoptanti PRIMARY KEY(Id\_adoptant,Idnp\_adoptant):**
    * Defineste o cheie primară compusă din Id\_adoptant și Idnp\_adoptant. Aceasta asigură unicitatea combinației acestor două câmpuri în tabel.
13. **CONSTRAINT UC\_adoptant UNIQUE (Idnp\_adoptant):**
    * Asigură că Codul Numeric Personal (Idnp\_adoptant) este unic în tabel, impunând o restricție de unicitate.

-In BD trebuie sa se regaseasca toate tipurile de constrangeri enumerate (la nivel de atribut si la nivel de relație):

PRIMARY KEY, UNIQUE, NOT NULL, CHECK, DEFAULT, FOREIGN KEY ... REFERENCES [ON DELETE ... ON UPDATE]

DROP TABLE IF EXISTS Animale

IF OBJECT\_ID('Animale') IS NULL

CREATE TABLE Animale (

Id\_animal INT IDENTITY(1, 1) NOT NULL PRIMARY KEY,

Nume\_animal VARCHAR(100) NOT NULL,

Tip\_animal VARCHAR(50) NOT NULL,

Greutate FLOAT NOT NULL,

Varsta INT NOT NULL,

Gen CHAR(1) NOT NULL CHECK (gen IN ('M', 'F')),

Id\_hrana INT NOT NULL,

Id\_donator INT NOT NULL,

Idnp\_donator CHAR(13) NOT NULL,

CONSTRAINT FK\_hrana FOREIGN KEY (Id\_hrana) REFERENCES Hrana (Id\_hrana) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT FK\_Donator FOREIGN KEY (Id\_donator,Idnp\_donator) REFERENCES Donatori (Id\_donator,Idnp\_donator) ON DELETE CASCADE ON UPDATE CASCADE

);

DROP TABLE IF EXISTS Adoptanti

IF OBJECT\_ID('Adoptanti') IS NULL

CREATE TABLE Adoptanti (

Id\_adoptant INT IDENTITY(1, 1) NOT NULL,

Idnp\_adoptant CHAR(13) NOT NULL,

Nume VARCHAR(50) NOT NULL,

Prenume VARCHAR(50) NOT NULL,

Adresa VARCHAR(100) NOT NULL,

Oras VARCHAR(50) NOT NULL DEFAULT 'Chisinau',

Email VARCHAR(50) NOT NULL CHECK (Email LIKE '%@%')UNIQUE,

Telefon VARCHAR(20) NOT NULL CHECK (Telefon LIKE '+373[0-9]%')UNIQUE,

Data\_adoptiei DATE NOT NULL,

Data\_nastere DATE NOT NULL,

Varsta AS DATEDIFF(YEAR, Data\_nastere, GETDATE()),

CONSTRAINT Pk\_Id\_adoptanti PRIMARY KEY(Id\_adoptant,Idnp\_adoptant),

CONSTRAINT UC\_adoptant UNIQUE (Idnp\_adoptant)

);

În acest script SQL se observă că se ragăsesc toate tipurile de constrângeri :

* PRIMARY KEY este utilizat în ambele tabele pentru atributele **Id\_animal** și  **Id\_adoptant,Idnp\_adoptant;**
* FOREIGN KEY este utilizat în tabelul ,,Animale’’ pentru atributele **Id\_donator,Idnp\_donator , Id\_hrana ;**
* NOT NULL este utilizat de toate atributele din ambele tabele;
* CHECK este folosit în ambele tabele pentru atributele **Gen, Email și Telefon**;
* DEFAULT este utilizat în tabelul Adoptanti pentru atributul **Oras**;
* FOREIGN KEY ... REFERENCES [ON DELETE ... ON UPDATE] este utilizat în tabelul Animale pentru constrângerile **FK\_hrana , FK\_Donator**

-Dupa posibilitate, in dependenta de caz, in BD ar trebui sa se regaseasca nu doar chei simple, dar si chei compuse , precum si chei externe compuse.

Chei simple din tabelele de mai sus : **Id\_animal, Id\_hrana**;

Chei compuse : **Pk\_Id\_adoptanti**;

Chei compuse externe: **FK\_Donator**;

-In schema trebuie sa fie prezente obligatoriu toate tipurile de asocieri (1:1, 1:n, n:m).

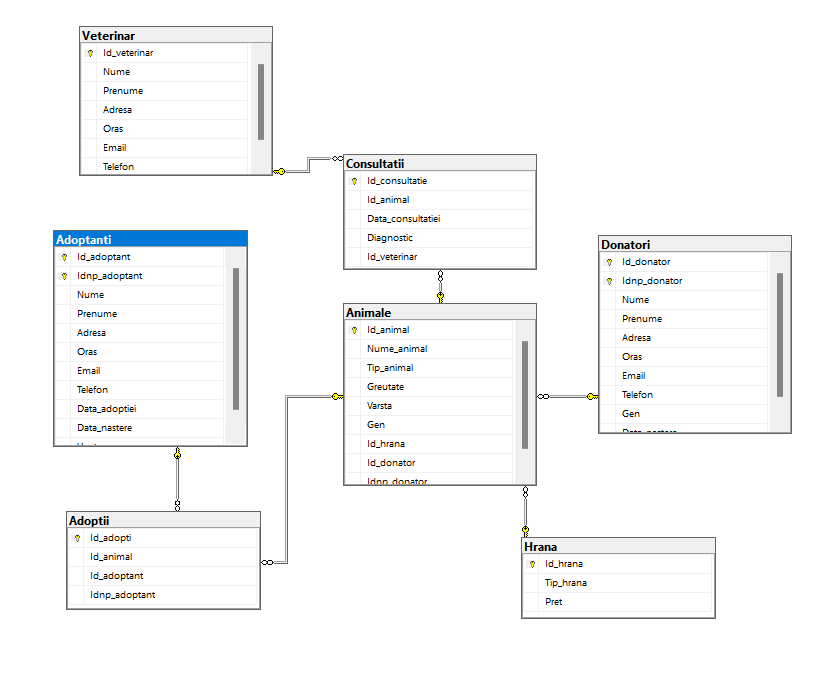


Figura 1 – Schema bazei de date

Tipurile de asocieri dintre tabele:

1:1- Animale-Hrana;

1:n- Donatori-Animale, Animale-Consultati, Veterinar-Consultati;

n:m – Adoptanti-Animale prin tabelul Adopți.

-Baza de date va fi populata cu tupluri doar prin comenzi INSERT in SQL (15 tupluri sau mai multe in fiecare relatie, in dependenta de caz).

-Nu se accepta date dubioase, introduse de mantuiala (de ex. Nume Prenume sa fie reale, Adrese reale, denumiri de produse si marci comerciale reale).

Baza de date a fost populate folosind doar comenizi INSERT datele introduce nu sunt de puse de muntuiala cum se vede mai jos :

-- Adaugare 15 valori in tabela Hrana

INSERT INTO Hrana (Tip\_hrana, Pret) VALUES

('Hrana pentru broaste testoase', 25.98),

('Hrana pentru iepuri', 18.50),

('Hrana uscata pentru ponei', 26.49),

('Hrana pentru serpi', 15.49),

('Hrana uscata pentru caini', 33.25),

('Hrana umeda pentru pesti', 19.99),

('Hrana uscata pentru cai', 28.75),

('Hrana pentru vrabii', 14.49),

('Hrana umeda pentru hamsteri', 10.99),

('Hrana pentru pesti exotici', 12.99),

('Biscuiti pentru pisici', 13.49),

('Hrana pentru reptile', 16.75),

('Hrana pentru gaini', 18.99),

('Hrana uscata pentru papagali', 15.99),

('Hrana uscata pentru porcusori de Guinea', 22.25);

-- Adaugare 15 valori în tabela Donatori

INSERT INTO Donatori (Idnp\_donator, Nume, Prenume, Adresa, Oras, Email, Telefon, Gen, Data\_nastere)

VALUES

('1234567890123', 'Popescu', 'Ion', 'Str. Primaverii, Nr. 1', 'Calarasi', 'popescu1.ion@example.com', '+37361212345', 'M', '1980-05-15'),

('9876543210987', 'Ionescu', 'Ana', 'Str. Florilor, Nr. 10','Straseni','ionesc2u.ana@example.com', '+373304512789', 'F', '1995-08-22'),

('1111222533344', 'Vasilescu', 'Elena', 'Str. Republicii, Nr. 7', 'Ialoveni', 'vasilescu.elena@example.com', '+37367821234', 'F', '1982-02-10'),

('5555646677788', 'Marinescu', 'Mihai', 'Str. Victoriei, Nr. 20', 'Balti', 'marinescu.mihai@example.com', '+37364567840', 'M', '1990-07-18'),

('9999888377666', 'Radulescu', 'Cristina', 'Str. Independentei, Nr. 15', 'Cahul', 'radulescu.cristina@example.com', '+37362127454', 'F', '1985-11-30'),

('1212121512121', 'Iancu', 'Alexandru', 'Str. Cuza Voda, Nr. 7', 'Ungheni', 'iancu.alex@example.com', '+37367738899', 'M', '1975-09-05'),

('3434343434943', 'Gheorghe', 'Maria', 'Str. Stefan cel Mare, Nr. 12', 'Soroca', 'gheorghe.maria@example.com', '+37366754433', 'F', '1998-04-25'),

('7676767676767', 'Dumitru', 'Mihai', 'Str. Republicii, Nr. 3', 'Chisinau', 'dumitru.mihai@example.com', '+37360234567', 'M', '1988-12-15'),

('1234567823123', 'Popov', 'Svetlana', 'Str. Rozelor, Nr. 15', 'Chisinau', 'popov.svetlana@example.com', '+37360987154', 'F', '1998-03-21'),

('9876233210987', 'Ivanov', 'Dmitrii', 'Str. Mihai Eminescu, Nr. 8', 'Chisinau', 'ivanov.dmitrii@example.com', '+37361789456', 'M', '1982-07-19'),

('1111222233355', 'Stoica', 'Elena', 'Str. Traian, Nr. 25', 'Chisinau', 'stoica.elena@example.com', '+37360121098', 'F', '1999-11-12'),

('5555666677799', 'Sorin', 'Iuliana', 'Str. Alba Iulia, Nr. 17', 'Chisinau', 'sorin.iuliana@example.com', '+37360178901', 'M', '1987-06-30'),

('9999888777677', 'Nicolae', 'Vladimir', 'Str. Calea Iesilor, Nr. 13', 'Chisinau', 'nicolae.vladimir@example.com', '+37360451789', 'M', '1993-09-08'),

('1212121212222', 'Ifrim', 'Roxana', 'Str. Dacia, Nr. 4', 'Chisinau', 'ifrim.roxana@example.com', '+37360123416', 'F', '1985-01-14'),

('3434343434433', 'Munteanu', 'Adrian', 'Str. Mitropolit Dosoftei, Nr. 11', 'Chisinau', 'munteanu.adrian@example.com', '+37361987654', 'M', '1978-04-07');

-- Adaugare 15 valori în tabela Animale

INSERT INTO Animale (Nume\_animal, Tip\_animal, Greutate, Varsta, Gen, Id\_hrana, Id\_donator,Idnp\_donator)

VALUES

('Rex', 'Broasca Testoasa', 15.5, 3, 'M', 1, 1,'1234567890123'),

('Whiskers', 'Iepure', 5.2, 2, 'F', 2, 1,'1234567890123'),

('Buddy', 'Ponel', 220.3, 4, 'M', 3, 2, '9876543210987'),

('Fluffy', 'Sarpe', 3.7, 1, 'F', 4, 2, '9876543210987'),

('Max', 'Caine', 8.9, 2, 'M', 5, 2, '9876543210987'),

('Luna', 'Peste', 0.25, 3, 'F', 6, 3, '1111222533344'),

('Rocky', 'Cal', 340.2, 5, 'M', 7, 3, '1111222533344'),

('Misty', 'Vrabie', 0.8, 2, 'F', 8, 3, '1111222533344'),

('Charlie', 'Hamster', 0.7, 3, 'M', 9, 4, '5555646677788'),

('Lucy', 'Pesti exotici', 0.30, 4, 'F', 10, 4, '5555646677788'),

('Teddy', 'Pisica', 2.1, 1, 'M', 11, 4, '5555646677788'),

('Oreo', 'Reptila', 2.3, 2, 'F', 12, 5, '9999888377666'),

('Coco', 'Gaina', 2.7, 3, 'M', 13, 5, '9999888377666'),

('Bella', 'Papagal', 3.4, 1, 'F', 14, 5, '9999888377666'),

('Simba', 'Porcusor de guinea', 1.2, 4, 'M', 15, 5, '9999888377666');

-- Adaugare 15 valori în tabela Veterinar

INSERT INTO Veterinar (Nume, Prenume, Adresa, Oras, Email, Telefon)

VALUES

('Ivanov', 'Dmitrii', 'Str. Mihai Eminescu, Nr. 8', 'Chisinau', 'ivanov.dmitrii\_vet@example.com', '+37361789456'),

('Popescu', 'Ion', 'Str. Independentei, Nr. 15', 'Cahul', 'popescu.ion\_vet@example.com', '+37362127454'),

('Vasilescu', 'Elena', 'Str. Republicii, Nr. 7', 'Ialoveni', 'vasilescu.elena\_vet@example.com', '+37367821234'),

('Marinescu', 'Mihai', 'Str. Victoriei, Nr. 20', 'Balti', 'marinescu.mihai\_vet@example.com', '+37364567840'),

('Radulescu', 'Cristina', 'Str. Cuza Voda, Nr. 7', 'Ungheni', 'radulescu.cristina\_vet@example.com', '+37367738899'),

('Iancu', 'Alexandru', 'Str. Stefan cel Mare, Nr. 12', 'Soroca', 'iancu.alex\_vet@example.com', '+37366754433'),

( 'Gheorghe', 'Maria', 'Str. Republicii, Nr. 3', 'Chisinau', 'gheorghe.maria\_vet@example.com', '+37360234567'),

('Dumitru', 'Mihai', 'Str. Rozelor, Nr. 15', 'Chisinau', 'dumitru.mihai\_vet@example.com', '+37360987154'),

('Popov', 'Svetlana', 'Str. Mihai Eminescu, Nr. 8', 'Chisinau', 'popov.svetlana\_vet@example.com', '+37361789426'),

('Ionescu', 'Ana', 'Str. Traian, Nr. 25', 'Chisinau', 'ionescu.ana\_vet@example.com', '+37360121028'),

('Stoica', 'Elena', 'Str. Alba Iulia, Nr. 17', 'Chisinau', 'stoica.elena\_vet@example.com', '+37362178901'),

('Sorin', 'Iuliana', 'Str. Calea Iesilor, Nr. 13', 'Chisinau', 'sorin.iuliana\_vet@example.com', '+37360251789'),

('Nicolae', 'Vladimir', 'Str. Dacia, Nr. 4', 'Chisinau', 'nicolae.vladimir\_vet@example.com', '+37360123216'),

('Ifrimos', 'Dima', 'Str. Mitropolit Dosoftei IVAN, Nr. 01', 'Chisinau', 'ifrimos.dima\_vet@example.com', '+37321984554'),

('Ifrim', 'Roxana', 'Str. Mitropolit Dosoftei, Nr. 11', 'Chisinau', 'ifrim.roxana\_vet@example.com', '+37321987654');

-- Adaugare 15 valori în tabela Consultatii

INSERT INTO Consultatii (Id\_animal, Data\_consultatiei, Diagnostic, Id\_veterinar)

VALUES

(1, '2023-01-05', 'Raceli', 1),

(2, '2023-02-10', 'Febra', 2),

(3, '2023-03-15', 'Indigestie', 3),

(4, '2023-04-20', 'Fractura', 4),

(5, '2023-05-25', 'Gastrita', 5),

(6, '2023-06-30', 'Alergie', 6),

(7, '2023-07-05', 'Paraziti intestinali', 7),

(8, '2023-08-10', 'Otita', 8),

(9, '2023-09-15', 'Raceala', 9),

(10, '2023-10-20', 'Deshidratare', 10),

(11, '2023-11-25', 'Colici', 11),

(12, '2023-12-30', 'Diaree', 12),

(13, '2024-01-05', 'Probleme dentare', 13),

(14, '2024-02-10', 'Afectiuni cardiace', 14),

(6, '2024-03-15', 'Anemie', 7);

-- Adaugare 15 valori în tabela Adoptanti

INSERT INTO Adoptanti (Idnp\_adoptant, Nume, Prenume, Adresa, Oras, Email, Telefon, Data\_adoptiei, Data\_nastere)

VALUES

('1234567890120', 'Popescu', 'Ana', 'Str. Libertatii, Nr. 5', 'Chisinau', 'popescu.ana@example.com', '+37360123456', '2022-01-15', '1990-03-20'),

('9876543210981', 'Ionescu', 'Mihai', 'Str. Unirii, Nr. 12', 'Chisinau', 'ionescu.mihai@example.com', '+37360234567', '2022-02-20', '1985-05-10'),

('1111222533342', 'Vasilescu', 'Elena', 'Str. Stefan cel Mare, Nr. 8', 'Chisinau', 'vasilescu.elena@example.com', '+37360345678', '2022-03-25', '1988-09-15'),

('5555646677783', 'Marinescu', 'Mihai', 'Str. Victoriei, Nr. 15', 'Chisinau', 'marinescu.mihai@example.com', '+37360456789', '2022-04-30', '1975-12-05'),

('9999888377664', 'Radulescu', 'Cristina', 'Str. Independentei, Nr. 10', 'Chisinau', 'radulescu.cristina@example.com', '+37360567890', '2022-05-05', '1995-08-22'),

('1212121512125', 'Iancu', 'Alexandru', 'Str. Cuza Voda, Nr. 7', 'Chisinau', 'iancu.alex@example.com', '+37360678901', '2022-06-10', '1982-02-10'),

('3434343434946', 'Gheorghe', 'Maria', 'Str. Stefan cel Mare, Nr. 12', 'Chisinau', 'gheorghe.maria@example.com', '+37360789012', '2022-07-15', '1990-07-18'),

('7676767676767', 'Dumitru', 'Mihai', 'Str. Republicii, Nr. 3', 'Chisinau', 'dumitru.mihai@example.com', '+37360890123', '2022-08-20', '1985-11-30'),

('1234567823128', 'Popov', 'Svetlana', 'Str. Rozelor, Nr. 15', 'Chisinau', 'popov.svetlana@example.com', '+37360901234', '2022-09-25', '1998-03-21'),

('9876233210999', 'Ivanov', 'Dmitrii', 'Str. Mihai Eminescu, Nr. 8', 'Chisinau', 'ivanov.dmitrii@example.com', '+37361012345', '2022-10-30', '1982-07-19'),

('1111222233350', 'Stoica', 'Elena', 'Str. Traian, Nr. 25', 'Chisinau', 'stoica.elena@example.com', '+37361123456', '2022-11-04', '1999-11-12'),

('5555666677791', 'Sorin', 'Iuliana', 'Str. Alba Iulia, Nr. 17', 'Chisinau', 'sorin.iuliana@example.com', '+37361234567', '2022-12-09', '1987-06-30'),

('9999888777672', 'Nicolae', 'Vladimir', 'Str. Calea Iesilor, Nr. 13', 'Chisinau', 'nicolae.vladimir@example.com', '+37361345678', '2023-01-14', '1993-09-08'),

('1212121212223', 'Ifrim', 'Roxana', 'Str. Dacia, Nr. 4', 'Chisinau', 'ifrim.roxana@example.com', '+37361456789', '2023-02-19', '1985-01-14'),

('3434343434434', 'Munteanu', 'Adrian', 'Str. Mitropolit Dosoftei, Nr. 11', 'Chisinau', 'munteanu.adrian@example.com', '+37361567890', '2023-03-26', '1978-04-07');

-- Adaugare 15 valori în tabela Adoptii

INSERT INTO Adoptii (Id\_animal, Id\_adoptant, Idnp\_adoptant)

VALUES

(1, 1, '1234567890120'),

(2, 2, '9876543210981'),

(3, 3, '1111222533342'),

(4, 4, '5555646677783'),

(5, 5, '9999888377664'),

(6, 6, '1212121512125'),

(7, 7, '3434343434946'),

(8, 8, '7676767676767'),

(9, 9, '1234567823128'),

(10, 10, '9876233210999'),

(11, 11, '1111222233350'),

(12, 12, '5555666677791'),

(13, 13, '9999888777672'),

(14, 14, '1212121212223'),

(15, 15, '3434343434434');

-Oricare interogare/comanda va avea un comentariu scris deasupra, alaturi de codul propriu-zis.

-- Creează Hrana dacă nu există deja

DROP TABLE IF EXISTS Hrana

IF OBJECT\_ID('Hrana') IS NULL

CREATE TABLE Hrana (

Id\_hrana INT IDENTITY(1, 1) NOT NULL,

Tip\_hrana VARCHAR(50) NOT NULL,

Pret FLOAT NOT NULL

);

ALTER TABLE Hrana

ADD CONSTRAINT PK\_hrana PRIMARY KEY (Id\_hrana);

--Introdcuerea datelor de proba

INSERT INTO Hrana (Tip\_hrana, Pret) VALUES

('Hrana pentru broaste testoase', 21.99),

('Hrana pentru iepuri', 18.50);

-- Stergea datelor din tabel dar tabelul ramane

TRUNCATE TABLE Hrana;

După cum se vede în scriptul SQL furnizat de mine fiecare comanda și interogare are comentariu scris deasupra.

-Tot codul se va scrie intr-un singur script sql (cu comentarii), care sa fie executabil multiplu, fara erori. Pentru asta, prin cod trebuie mai intai stearsa BD existenta, apoi creata aceeasi BD din nou, cu inserturi etc.

-- Verifică dacă baza de date există deja

IF EXISTS (SELECT name FROM sys.databases WHERE name = 'AdapostAnimale')

BEGIN

-- Schimbă contextul în baza de date master și închide conexiunile active

USE master;

ALTER DATABASE AdapostAnimale SET SINGLE\_USER WITH ROLLBACK IMMEDIATE;

-- Șterge baza de date existentă

DROP DATABASE AdapostAnimale;

END;

-- Creează baza de date

CREATE DATABASE AdapostAnimale;

-- Utilizează baza de date

USE AdapostAnimale;

-- Creează Hrana dacă nu există deja

DROP TABLE IF EXISTS Hrana

IF OBJECT\_ID('Hrana') IS NULL

CREATE TABLE Hrana (

Id\_hrana INT IDENTITY(1, 1) NOT NULL,

Tip\_hrana VARCHAR(50) NOT NULL,

Pret FLOAT NOT NULL

);

Cum se cerea în condiție am creat scriptul ca să verifice daca baza de date este deja create sau nu . Daca exista ea se va sterge si se vac rea din nou daca nu exista ea doar va fi create. După care se verifică existența tabelelor . Dacă tabelul exista el se va sterge daca el nu exista se va crea.

**Concluzie**

În cadrul acestei lucrări de laborator, am dobândit cunoștințe semnificative privind proiectarea și implementarea unei baze de date complexe. Am creat o structură de bază pentru un adapost de animale, abordând diverse aspecte ale gestionării informațiilor, cum ar fi hrană, donatori, animale, veterinari, consultatii, adoptanți și adoptii.

Am utilizat diverse tipuri de date și constrângeri pentru a asigura o reprezentare realistă și coerentă a informațiilor. Implementarea cheilor primare, cheilor externe, precum și a diverselor constrângeri de integritate a datelor, a contribuit la stabilitatea și coerența bazei de date.

De asemenea, am învățat să populez baza de date cu date de test relevante și realiste, respectând condițiile specifice ale anumitor atribute. Scriptul final este conceput pentru a fi executat repetat, începând cu ștergerea bazei de date existente și recrearea acesteia, asigurând astfel o funcționalitate și reproducibilitate adecvată.

Prin această experiență, am dezvoltat abilități esențiale în proiectarea, implementarea și administrarea bazelor de date, pregătindu-ne pentru gestionarea eficientă a informațiilor în proiecte viitoare.