

Diaconu Andreea-Teodora

Grupa 332AB

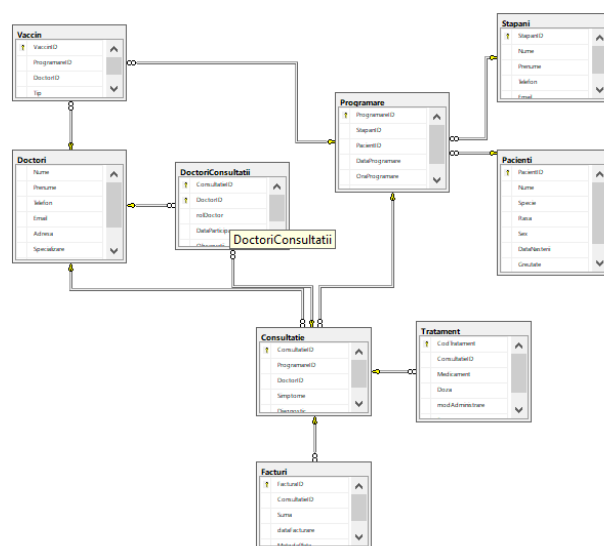
Proiect BD

Proiect Cabinet Veteinar

Proiectul „Cabinet Veterinar” urmărește să creeze o soluție software modernă pentru gestionarea activităților unei clinici veterinare. Scopul principal este de a eficientiza procesul de administrare a datelor despre pacienți, consultații, programări și tratamente. Aplicația este proiectată pentru a răspunde atât cerințelor funcționale, precum și nevoii de integritate și securitate a datelor. În cadrul acestui proiect, am acordat o atenție deosebită organizării informațiilor, printr-o bază de date bine structurată, care să permită relații clare între tabele și o interacțiune eficientă cu utilizatorii.

Pentru a îndeplini aceste obiective, proiectul include o etapă detaliată de proiectare, definirea relațiilor și constrângerilor dintre entități, precum și implementarea unor interogări SQL care să ofere rapoarte și statistici relevante pentru utilizatori. Funcționalitatea aplicației este construită astfel încât să îmbunătățească fluxurile de lucru dintr-un cabinet veterinar, de la gestionarea pacienților la administrarea programărilor și consultațiilor.

Descrierea tabelelor si a relatiilor



1. Tabelul Doctori

- Coloane:
 - DoctorID (PK) - identificator unic pentru fiecare doctor.
 - Nume, Prenume, Telefon, Email, Adresa, Specializare - detalii personale și profesionale ale doctorilor.
- Relații:
 - Este conectat la tabelul DoctorConsultatii prin coloana DoctorID.
 - Este conectat la Consultatie prin DoctorID.
- Constrângeri:
 - DoctorID este cheie primară și nu poate fi NULL.
 - Email este unic.

2. Tabelul Pacienti

- Coloane:
 - PacientID (PK) - identificator unic pentru fiecare pacient.
 - Nume, Specie, Rasa, Sex, DataNasterii, Greutate - detalii despre fiecare pacient.
- Relații:
 - Este conectat la tabelul Programare prin coloana PacientID.
- Constrângeri:
 - PacientID este cheie primară și nu poate fi NULL.

3. Tabelul Stapani

- Coloane:
 - StapanID (PK) - identificator unic pentru fiecare stăpân.
 - Nume, Prenume, Telefon, Email - detalii de contact pentru stăpâni.
- Relații:
 - Este conectat la Programare prin coloana StapanID.
- Constrângeri:
 - StapanID este cheie primară și nu poate fi NULL.

4. Tabelul Programare

- Coloane:
 - ProgramareID (PK) - identificator unic pentru fiecare programare.
 - StapanID (FK), PacientID (FK), DataProgramare, OraProgramare, Motiv - detalii ale programării.
- Relații:
 - Conectat la Stapani prin StapanID.
 - Conectat la Pacienti prin PacientID.
 - Conectat la Consultatie prin ProgramareID.
- Constrângeri:
 - ProgramareID este cheie primară și nu poate fi NULL.
 - StapanID și PacientID sunt chei externe și trebuie să existe în tabelele respective.

5. Tabelul Consultatie

- Coloane:
 - ConsultatieID (PK) - identificator unic pentru fiecare consultație.
 - ProgramareID (FK), DoctorID (FK), Simptome - detalii despre consultație.
- Relații:
 - Conectat la Programare prin ProgramareID.
 - Conectat la Doctori prin DoctorID.
 - Conectat la Facturi și Tratament.
- Constrângeri:
 - ConsultatieID este cheie primară și nu poate fi NULL.
 - ProgramareID și DoctorID sunt chei externe și trebuie să existe în tabelele respective.

6. Tabelul DoctorConsultatii

- Coloane:
 - ConsultatieID (FK), DoctorID (FK), DataParticipare, RolDoctor.
- Relații:
 - Conectat la Doctori și Consultatie.

- Constrângeri:
 - ConsultatieID și DoctorID sunt chei externe.

7. Tabelul Tratament

- Coloane:
 - CodTratament (PK), ConsultatieID (FK), Medicament, Doza, ModAdministrare, Durata.
- Relații:
 - Conectat la Consultatie prin ConsultatieID.
- Constrângeri:
 - CodTratament este cheie primară și nu poate fi NULL.
 - ConsultatieID este cheie externă.

8. Tabelul Facturi

- Coloane:
 - FacturaID (PK), ConsultatieID (FK), Suma, DataFacturare, MetodaPlata.
- Relații:
 - Conectat la Consultatie prin ConsultatieID.
- Constrângeri:
 - FacturaID este cheie primară.
 - ConsultatieID este cheie externă.

9. Tabelul Vaccin

- Coloane:
 - VaccinID (PK), ProgramareID (FK), DoctorID (FK), Tip.
- Relații:
 - Conectat la Programare și Doctori.
- Constrângeri:
 - VaccinID este cheie primară.
 - ProgramareID și DoctorID sunt chei externe.

Constrângeri de integritate impuse

- Fiecare tabel are o cheie primară (PK) care identifică în mod unic fiecare rând.

- Relațiile dintre tabele sunt definite prin chei externe (FK), asigurând consistența datelor.
- Coloanele care nu pot accepta valori NULL asigură existența obligatorie a datelor esențiale.
- Integritatea referențială este menținută prin constrângeri de tip FOREIGN KEY.
- Datele precum Email (în Doctori și Stapani) sunt unice pentru a preveni duplicatele.

Interogari

```
SELECT
    PR.DataProgramare,
    PR.OraProgramare,
    PR.Motiv,
    P.Nume AS PatientName
FROM
    Programare PR
INNER JOIN
    Pacienti P ON PR.PacientID = P.PacientID
WHERE
    PR.StapanID = @DoctorID
ORDER BY
    PR.DataProgramare, PR.OraProgramare";
```

Adaugă Programare

ID Pacient

Data Programării

Ora Programării

Motiv

Prima interogare cu join pe care am realizat-o este regasita pe pagina de “Login”, odata ce un doctor s-a autentificat, pentru a adauga o programare noua. Această interogare selectează datele programărilor (**DataProgramare**, **OraProgramare**, **Motiv**) și numele pacientului (**PatientName**) din tabelul **Programare**, alăturându-le cu tabelul **Pacienti** pe baza ID-ului pacientului (**PacientID**). Filtrează doar programările asociate unui anumit doctor (**StapanID** = **@DoctorID**) și ordonează rezultatele în funcție de dată și oră.

```
SELECT
    C.ConsultatieID,
    D.Nume AS NumeDoctor,
    P.Nume AS NumePacient,
    C.Simptome
FROM Consultatie C
INNER JOIN Doctori D ON C.DoctorID = D.DoctorID
INNER JOIN Programare PR ON C.ProgramareID = PR.ProgramareID
INNER JOIN Pacienti P ON PR.PacientID = P.PacientID
WHERE D.Nume = @NumeDoctor;";
```

```

SELECT
    D.Nume AS NumeDoctor,
    COUNT(C.ConsultatieID) AS NumarConsultatii
FROM Doctori D
LEFT JOIN Consultatie C ON D.DoctorID = C.DoctorID
WHERE D.Nume = @NumeDoctor
GROUP BY D.Nume;";

```

Căutare Consultații și Statistici

Introduceți Numele Doctorului:

Caută

Rezultatele Consultațiilor

Nu există rezultate pentru numele introdus.

Statistici per Doctor

Nu există statistici pentru numele introdus.

Prima dintre acestea doua returnează informații despre consultații (**ConsultatieID**, simptomele consultației și numele doctorului și al pacientului) din mai multe tabele. Alături tabelele **Consultatie**, **Doctori**, **Programare**, și **Pacienti** pe baza relațiilor lor logice, și filtrează doar consultațiile efectuate de doctorul al cărui nume corespunde parametrului **@NumeDoctor**, iar a doua calculează numărul de consultații efectuate de un anumit doctor. Folosește un **LEFT JOIN** între tabelele **Doctori** și **Consultatie** pentru a include toți doctorii, chiar dacă unii nu au efectuat consultații. Numără consultațiile asociate doctorului specificat de parametrul **@NumeDoctor**, grupând rezultatele pe numele doctorului.

```

SELECT
    DoctorID,
    (SELECT Nume
     FROM Doctori
     WHERE Doctori.DoctorID = Vaccin.DoctorID) AS NumeDoctor,
    COUNT(VaccinID) AS NumarVaccinuri,
    (SELECT COUNT(DISTINCT ProgramareID)
     FROM Vaccin V2
     WHERE V2.DoctorID = Vaccin.DoctorID) AS ProgramariUnice
FROM Vaccin
WHERE Tip = @TipVaccinCautat
GROUP BY DoctorID
ORDER BY NumarVaccinuri DESC;";

```

```

SELECT
    P.PacientID,
    P.Nume AS NumePacient,
    V.Tip AS TipVaccin
FROM Vaccin V
INNER JOIN Programare PR ON V.ProgramareID = PR.ProgramareID
INNER JOIN Pacienti P ON PR.PacientID = P.PacientID
WHERE V.Tip = @TipVaccinCautat
ORDER BY P.Nume;";

```

Detalii Vaccinare

Introduceți tipul vaccinului:

În cazul paginii de vaccin, prima calculează statistici despre vaccinurile administrate de doctori pentru un anumit tip de vaccin specificat de parametrul @TipVaccinCautat. Returnează **DoctorID**, numele doctorului (determinat printr-o subinterogare), numărul total de vaccinuri administrate și numărul de programări unice asociate fiecărui doctor. Datele sunt grupate pe **DoctorID** și sortate descrescător după numărul de vaccinuri, iar a doua returnează o listă cu pacienții care au primit un anumit tip de vaccin specificat de parametrul @TipVaccinCautat. Se alătură tabelele **Vaccin**, **Programare** și **Pacienti**, filtrând pe baza tipului de vaccin, și ordonează rezultatele alfabetic după numele pacienților. Rezultatul include **PacientID**, numele pacientului și tipul de vaccin primit.

```
SELECT
    D.Nume AS NumeDoctor,
    D.Specializare AS Specializare,
    T.Medicament AS MedicamentPrescis,
    T.Doza AS DozaRecomandata,
    T.modAdministrare AS ModAdministrare
FROM Tratament T
INNER JOIN Consultatie C ON T.ConsultatieID = C.ConsultatieID
INNER JOIN Doctori D ON C.DoctorID = D.DoctorID
WHERE C.ProgramareID IN (
    SELECT PR.ProgramareID
    FROM Programare PR
    INNER JOIN Pacienti P ON PR.PacientID = P.PacientID
    WHERE P.Nume LIKE '%' + @NumePacientCautat + '%'
)
ORDER BY D.Nume, T.Medicament;";
```

```
SELECT
    P.Nume AS NumePacient,
    P.Rasa AS RasaPacient,
    (SELECT COUNT(*)
     FROM Tratament T
     WHERE T.ConsultatieID IN (
         SELECT C.ConsultatieID
         FROM Consultatie C
         INNER JOIN Programare PR ON C.ProgramareID = PR.ProgramareID
         WHERE PR.PacientID = P.PacientID
     )) AS NumarTratamente,
    (SELECT STUFF(
        (SELECT ', ' + T.Medicament
         FROM Tratament T
         WHERE T.ConsultatieID IN (
             SELECT C.ConsultatieID
             FROM Consultatie C
             INNER JOIN Programare PR ON C.ProgramareID = PR.ProgramareID
             WHERE PR.PacientID = P.PacientID
         ))
        FOR XML PATH('')), 1, 2, '')
    ) AS MedicamentePrescise
FROM Pacienti P
WHERE P.Nume LIKE '%' + @NumePacientCautat + '%'
ORDER BY NumarTratamente DESC;";
```

Caută Tratamente Prescise

Introduceți numele pacientului:

În ceea ce privește tratamentul, prima obține informații despre tratamentele prescrise de doctori pentru un anumit pacient al cărui nume conține șirul specificat prin parametrul @NumePacientCautat. Folosind un **INNER JOIN** între tabelele **Tratament**, **Consultatie**, și **Doctori**, se identifică medicamentele prescrise de doctori în cadrul consultațiilor pentru programările asociate pacientului. Subinterogarea din clauza WHERE caută programările pentru pacientul dorit pe baza numelui. Rezultatele sunt sortate după numele doctorilor și medicamentele prescrise, iar a doua oferă statistici despre pacienții al căror nume conține șirul specificat prin parametrul @NumePacientCautat. Pentru fiecare pacient, returnează numele, rasa, numărul total de tratamente și o listă concatenată cu medicamentele prescrise. Subinterogările calculează, respectiv:

1. Numărul de tratamente pe baza consultațiilor asociate pacientului.
2. Lista medicamentelor prescrise folosind FOR XML PATH (pentru concatenare).
Rezultatele sunt sortate descrescător după numărul de tratamente.

```
SELECT
    D.DoctorID,
    D.Nume,
    D.Prenume,
    D.Email,
    D.Specializare
FROM Doctori D
INNER JOIN Programare P ON D.DoctorID = P.StapanID
WHERE D.Specializare = @Specializare
GROUP BY D.DoctorID, D.Nume, D.Prenume, D.Email, D.Specializare
ORDER BY D.Nume;
```

Căutare Doctori

Selectați Specializarea

Oncologie Veterinară

Această interogare obține informații despre doctorii care au specializarea specificată prin parametrul @Specializare. Folosind un **INNER JOIN** între tabelele **Doctori** și **Programare**, se identifică doctorii care sunt asociați cu programări (prin legătura dintre DoctorID și StapanID)


```
SELECT DISTINCT
    P.PacientID,
    P.Nume AS NumePacient,
    P.Rasa AS RasaPacient,
    PR.DataProgramare,
    PR.Motiv
FROM Pacienti P
INNER JOIN Programare PR ON P.PacientID = PR.PacientID
WHERE PR.ProgramareID IN (
    SELECT C.ProgramareID
    FROM Consultatie C
    WHERE C.DoctorID = (
        SELECT DoctorID
        FROM Doctori
        WHERE Nume LIKE '%' + @DoctorName + '%'
    )
)
ORDER BY P.Nume;"
```

Caută Pacienți după Doctor

Introduceți numele doctorului:

Această interogare obține informații despre pacienții care sunt asociați cu programări, având consultații realizate de un doctor al cărui nume conține un anumit șir de caractere specificat prin parametrul @DoctorName

Functionarea aplicatiei

Designul este orientat către utilizator, asigurând accesibilitate și ușurință în utilizare pentru toate categoriile de utilizatori, de la administratori la personal medical.

Un aspect central al aplicației este gestionarea informațiilor despre doctori. Sistemul permite înregistrarea detaliilor relevante despre fiecare doctor, precum numele, specializarea și datele de contact. Această evidență facilitează organizarea activităților și asigură acces rapid la informațiile necesare pentru coordonarea programărilor și consultațiilor.

Aplicația include, de asemenea, un modul dedicat gestionării pacienților și a stăpânilor acestora. Fiecare pacient este înregistrat cu detalii precum specia, rasa, sexul, greutatea și data nașterii, iar stăpânii sunt identificați prin informații personale și de contact. Această organizare permite asocierea fiecărui pacient cu stăpânul său și oferă un istoric medical complet și ușor de accesat.

Un alt punct forte al aplicației este funcționalitatea de planificare a programărilor. Utilizatorii pot crea programări pentru pacienți, specificând data, ora și motivul vizitei. Programările sunt gestionate eficient și integrate cu modulele de consultații și tratamente, asigurând un flux de lucru coerent și bine structurat.

Consultațiile reprezintă un element esențial al aplicației, acestea fiind înregistrate cu detalii despre simptomele pacientului, diagnosticul oferit de doctor și tratamentele recomandate. Modulul de tratamente asigură o evidență clară a medicamentelor prescrise, a dozelor și a modului de administrare, contribuind astfel la îmbunătățirea calității serviciilor medicale oferite.

Procesul de facturare este automatizat, generând facturi pe baza consultațiilor realizate. Aceste facturi includ informații relevante despre sume, metode de plată și detalii ale consultațiilor, garantând transparență și organizare în gestionarea financiară a cabinetului.

Pe lângă funcționalitățile operaționale, aplicația oferă și capabilități de analiză și raportare. Statistici precum numărul de consultații realizate, distribuția pacienților pe specii sau cele mai frecvente tratamente sunt generate prin interogări complexe. Aceste informații sunt prezentate într-un format ușor de înțeles și contribuie la procesul decizional, ajutând la optimizarea serviciilor și la identificarea tendințelor.

Prin designul său intuitiv și funcționalitățile robuste, aplicația răspunde cerințelor unui cabinet veterinar modern, oferind suport atât pentru activitățile administrative, cât și pentru cele medicale. Integrarea tuturor modulelor într-un singur sistem facilitează gestionarea eficientă a resurselor și contribuie la îmbunătățirea experienței clienților și a personalului.