

Angheluță Vlad Matei

Grupa 333AB

Proiect Cabinet Veterinar

Introducere

Aplicația prezentată se bazează pe baza de date a unui cabinet veterinar, iar principalele funcționalități sunt afișarea de statistici legate de veterinarii, clienții, animalele tratate, serviciile oferite etc. ale cabinetului.

Baza de date – Tabele

Baza de date este compusă din 9 tabele în total.

- a. Veterinari:
Include informații despre veterinarii angajați în cabinetul medical anume: ID-ul, nume, prenume, CNP, Email, număr de telefon, adresă, orașul în care locuiesc, data nașterii precum și specializarea acestora.
- b. Clienți:
Include informații despre clienții care au apelat la serviciile cabinetului medical anume: ID-ul, nume, prenume, CNP, Email, număr de telefon, adresă, orașul în care locuiesc, data nașterii.
- c. Animale:
Conține detalii despre animalele tratate sau diagnosticate în cadrul cabinetului, include ID-ul numele, ID-ul clientului care îl deține, vârsta, rasa și specia.
- d. Afecțiuni medicale:
Oferă informații despre posibilele afecțiuni medicale care pot fi diagnosticate în cadrul cabinetului precum ID-ul, denumirea, o descriere a simptomelor și recomandări pentru tratare.
- e. Animale cu afecțiuni:
Acesta este un tabel de legătură care prezintă animalele diagnosticate cu diverse afecțiuni. Tabela conține ID-ul animalului, al afecțiunii, data diagnosticului precum și starea actuală a animalului(vindecat sau nevindecat).
- f. Categoriile servicii:
Tabela este compusă din categoriile de servicii oferite de cabinet, conține ID-ul, denumirea categoriei și detalii despre tipul de proceduri.
- g. Servicii:
Cuprinde toate serviciile puse la dispoziție de cabinetul medical. Câmpurile sunt ID-ul categoriei, al serviciului, denumirea, prețul și o scurtă descriere a procedurii.

h. Consultații:

Conține istoricul consultațiilor efectuate în cabinet, câmpurile prezente sunt: ID-ul consultației, al animalului tratat, al serviciului, al veterinarului care a prestat serviciul și data în care a avut loc.

i. Reviews:

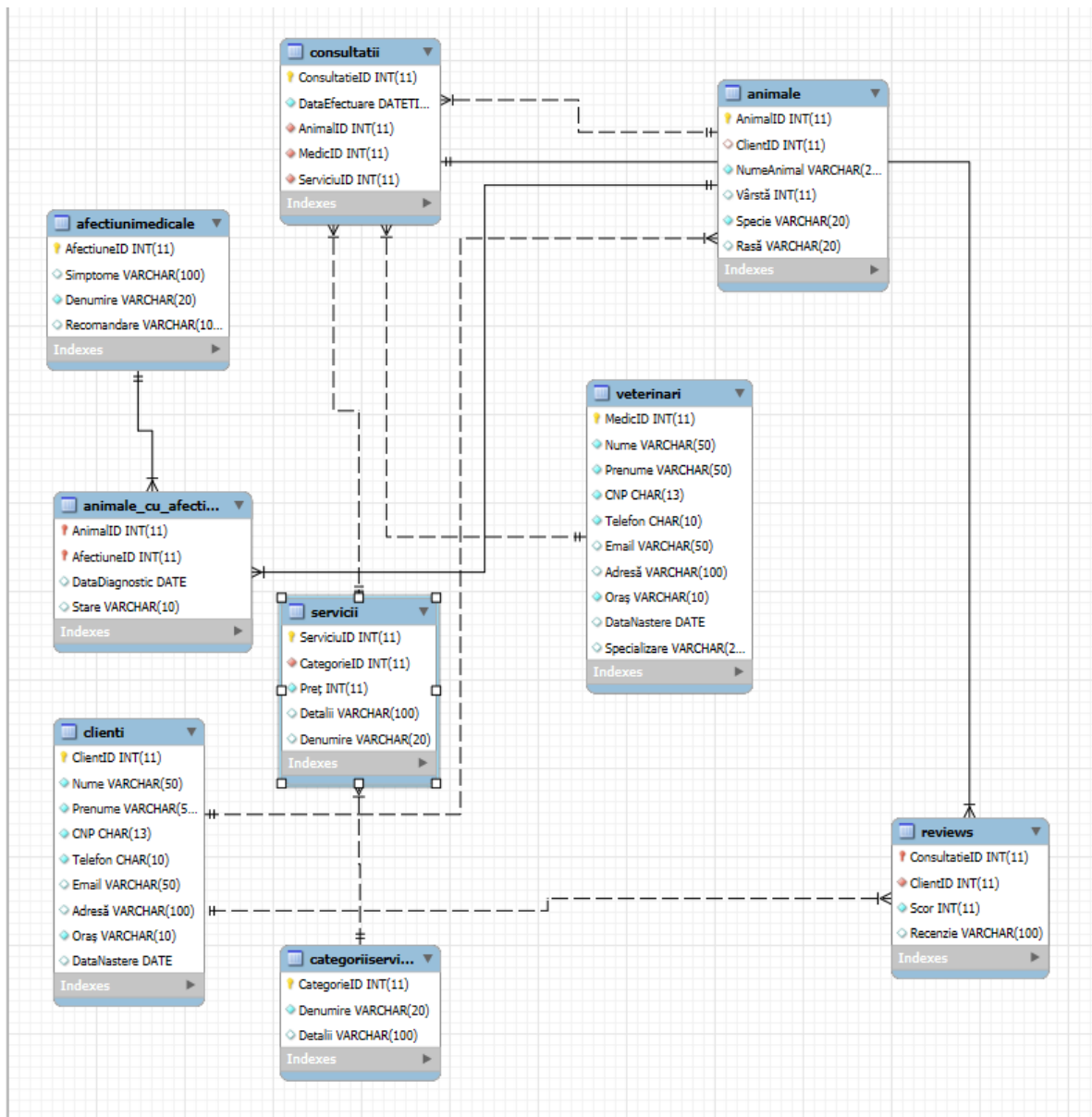
Este compusă din toate recenziile acordate serviciilor cabinetului. Pentru aceasta, tabela conține: ID-ul consultației, al clientului care a scris-o, scorul și recenzia scrisă.

Baza de date – Relații între tabele

Apar următoarele relații:

- Relație 1:N între categoriiservicii și servicii, o categorie poate avea mai multe servicii, un serviciu aparține unei singure categorii
- Relație 1:N între animale și clienți, un client poate avea mai multe animale, un animal aparține unui singur client
- Relație N:N între animale și afecțiuni medicale, o afecțiune poate să apară la mai multe animale, un animal poate avea mai multe afecțiuni, se realizează prin intermediul tabelului de legătură animale_cu_afecțiuni
- Relații N:N între animale, servicii, veterinari, toate cele 3 categorii pot să apară în mai multe consultații în combinații diferite, de aceea se implementează prin tabelul de legătură consultatii
- Relație 1:1 între consultatii și reviews, o consultație poate avea maxim 1 review și reciproc
- Relație 1:N între clienti și reviews, un client poate să dea mai multe reviews, iar un review este scris de un singur client.

Diagrama cu tabele și relațiile dintre ele



Constrângeri pentru funcționalitatea aplicației

- Toate ID-urile chei principale ale tabelor sunt nenule și se auto-incrementează
- Toate câmpurile din tabelele clienti și veterinari sunt NOT NULL
- Câmpul ClientID din animale, câmpul CategorieID din servicii, câmpurile AnimalID și AfectiuneID din animale_cu_afectiuni, câmpurile ServiciuID, MedicID, AnimalID din consultatii, câmpurile ClientID și ConsultatieID din reviews sunt NOT NULL
- Câmpul Scor din reviews poate să ia doar valori de la 1 la 10.
- Câmpul Stare din animale_cu_afectiuni poate lua ca valori doar ,Vindecat' sau ,Nevindecat'

Implementarea aplicației

În primul rând, baza de date a fost creată folosind MySQL, conectarea la server se face prin intermediul aplicației MAMP. Aceasta oferă ca interfață site-ul phpMyAdmin, prin intermediul acesteia s-au scris query-urile pentru crearea tabelelor și introducerea de valori în acestea, de asemenea și pentru testarea de noi queries.

Pentru aplicația propriu-zisă s-a lucrat cu limbajul de programare vizuală C#, folosind ca IDE Microsoft Visual Studio 2022. Tipul de aplicație este desktop, lucrându-se cu tipologia de Windows Forms App în mediul Visual Studio. Aceasta oferă de a introduce manual elementele pentru un Forms App fără a scrie cod. De asemenea, se pot scrie în fișierele fiecărui form funcțiile care implementează serviciile aplicației.

Așadar, aplicația este alcătuită din mai multe forms prin care utilizatorul poate să traverseze cu ajutorul unor butoane de pe meniul principal, primul forms care apare la pornirea aplicației. Aici apar opțiunile de a accesa forms-uri pentru: editarea tabelelor de clienți și veterinari sau vizualizarea de statistici legate clienți, veterinari, animale, afecțiuni medicale sau servicii. Rezultatele query-urilor sunt afișate în datagrids și se execută prin apăsarea de butoane. Pentru query-urile cu parametri variabili se pot introduce date în textbox-uri sau se pot selecta rânduri din tabele deja afișate.

Pentru a putea executa query-uri de MySQL în C# s-a utilizat librăria Data.MySQLClient care oferă mai multe metode pentru a putea lucra cu o bază de date din MySQL.

Conexiunea la baza de date necesită crearea unui obiect de tip MySqlConnection, acesta primește ca dată de intrare un string de conectare. Pentru baza de date cabinetveterinar, string-ul arată astfel:

"datasource=localhost;port=3306;username=root;password=root;database=cabinetveterinar"

Așadar se utilizează port-ul 3306, username-ul este root, iar parola este tot root.

Codul sursă este localizat în folder-ul AplicatieCabinetVeterinar\AplicatieCabinetVeterinar din arhivă.

Editarea de baze de date

S-au implementat funcționalități de update, insert și delete pentru tabelele de clienți și veterinari. Astfel pentru fiecare tabelă există un forms separat în care se pot efectua cele 3 operații pentru un tabel.

Insert

În textbox-uri se introduc valori pentru fiecare câmp și apoi se apasă butonul de adaugă. În cazul în care nu se introduc toate datele necesare se afișează un mesaj de avertisment. Noua intrare se poate apoi vedea în lista cu înregistrările tabelului.

- Veterinari: **INSERT INTO `veterinari` (`Nume`, `Prenume`, `CNP`, `Telefon`, `Email`, `Adresă`, `Oraș`, `DataNastere`, `Specializare`) VALUES (@nume,@prenume,@cnp,@telefon,@email,@adresa,@oras,@data,@specializare)**
- Clienți: **„INSERT INTO `clienti` (`Nume`, `Prenume`, `CNP`, `Telefon`, `Email`, `Adresă`, `Oraș`, `DataNastere`) VALUES (@nume,@prenume,@cnp,@telefon,@email,@adresa,@oras,@data)”**

Update

Se alege înregistrarea care se dorește actualizată din tabelul afișat, se completează textbox-urile și se apasă pe butonul de Update. Intrarea apare automat actualizată în tabel.

- Veterinari: **„UPDATE `veterinari` SET `Nume` = @nume, `Prenume` = @prenume, `Telefon` = @telefon, `Email` = @email, `Adresă` = @adresa, `Oraș` = @oras, `DataNastere` = @data, `Specializare` = @specializare WHERE `CNP` = @cnp”**
- Clienți: **„UPDATE `clienti` SET `Nume` = @nume, `Prenume` = @prenume, `Telefon` = @telefon, `Email` = @email, `Adresă` = @adresa, `Oraș` = @oras, `DataNastere` = @data WHERE `CNP` = @cnp”**

Delete

Se alege înregistrarea care se dorește a fi ștearsă și se apasă pe butonul de Șterge. Lista se va actualiza și nu va mai apărea intrarea respectivă.

- Veterinari: **"DELETE FROM `veterinari` WHERE `CNP` = @cnp"**
- Clienți: **"DELETE FROM `clienti` WHERE `CNP` = @cnp"**

Query-urile simple cu JOIN-uri

1. Afișarea tuturor consultațiilor la care a luat parte un veterinar
**„SELECT Cl.Nume AS NumeClient, A.NumeAnimal AS Nume_Animal,
C.DataEfectuare AS Data, S.Denumire AS Serviciu
FROM consultatii C JOIN animale A ON C.AnimalID = A.AnimalID
JOIN veterinari V ON V.MedicID = C.MedicID
JOIN servicii S ON S.ServiciuID = C.ServiciuID
JOIN clienti Cl ON A.ClientID = Cl.ClientID**

WHERE V.CNP = @CNP”

Apare un parametru variabil, acest query se execută pentru veterinarul selectat din lista de veterinari afișată în form. Dacă nu este selectat niciunul, apare un mesaj de eroare. Se află în forms-ul pentru statistici veterinari.

2. Afișarea scorului mediu primit de un veterinar în toate recenziile consultațiilor sale
„SELECT AVG(R.Scor) AS Recenzie_medie, V.Nume AS NumeMedic, V.Prenume AS PrenumeMedic

FROM reviews R JOIN consultatii C ON C.ConsultatieID = R.ConsultatieID

JOIN veterinari V ON V.MedicID = C.MedicID

WHERE V.CNP = @CNP

GROUP BY V.Nume, V.Prenume”

Apare un parametru variabil, acest query se execută pentru veterinarul selectat din lista de veterinari afișată în form. Dacă nu este selectat niciunul, apare un mesaj de eroare. Se află în forms-ul pentru statistici veterinari.

3. Afișarea sumei totale cheltuite pe consultații de un client
„SELECT SUM(S.Preț) AS Total, Cl.Nume AS NumeClient, Cl.Prenume AS PrenumeClient

FROM servicii S JOIN consultatii C ON S.ServiciuID = C.ServiciuID

JOIN animale A ON C.AnimalID = A.AnimalID

JOIN clienti Cl ON A.ClientID = Cl.ClientID

WHERE Cl.CNP = @CNP

GROUP BY Cl.Nume, Cl.Prenume”

Apare un parametru variabil, acest query se execută pentru clientul selectat din lista de clienți afișată în form. Dacă nu este selectat niciunul, apare un mesaj de eroare. Se află în forms-ul pentru statistici clienți.

4. Afișarea tuturor animalelor unui client
„SELECT A.NumeAnimal AS Nume, A.Vârstă AS Vârstă, A.Specie AS Specie, A.Rasă AS Rasă

FROM animale A JOIN clienti Cl ON A.ClientID = Cl.ClientID

WHERE Cl.CNP = @CNP”

Apare un parametru variabil, acest query se execută pentru clientul selectat din lista de clienți afișată în form. Dacă nu este selectat niciunul, apare un mesaj de eroare. Se află în forms-ul pentru statistici clienți.

5. Afișarea datei celui mai recent diagnostic pentru fiecare afecțiune din baza de date

```
„SELECT Af.Denumire AS NumeAfecțiune, MAX(Aaf.DataDiagnostic) AS
CelMaiRecentDiagnostic
FROM afectiunimedicale Af JOIN animale_cu_afectiuni Aaf ON Af.AfectiuneID =
Aaf.AfectiuneID
GROUP BY Af.Denumire
ORDER BY MAX(Aaf.DataDiagnostic) DESC”
```

Se afișează prin apăsarea unui buton în aplicație, rezultatele sunt ordonate descrescător după data diagnosticului, apare în statistici boli.

6. Afișarea tuturor animalelor diagnosticate cu un anumit status

```
„SELECT A.NumeAnimal AS Nume, A.Vârstă AS Vârstă, A.Specie AS Specie, A.Rasă
AS Rasă, Aaf.DataDiagnostic AS DataDiagnostic, Af.Denumire
FROM animale_cu_afectiuni Aaf JOIN animale A ON Aaf.AnimalID = A.AnimalID
JOIN afectiuni Af ON Aaf.AfectiuneID = Af.AfectiuneID
WHERE Aaf.Stare = @stare”
```

Apare un parametru variabil, dintr-un drop down list se alege starea ‚Vindecăt’ sau ‚Nevindecăt’, iar la apăsarea butonului se afișează toate animalele diagnosticate cu acel status, apare în statistici boli.

7. Afișarea tuturor consultațiilor care au loc într-o perioadă delimitată de utilizator

```
„SELECT Cl.Nume AS NumeClient, A.NumeAnimal AS Nume_Animal,
C.DataEfectuare AS Data, S.Denumire AS Serviciu, V.Nume AS NumeVeterinar,
V.Prenume AS PrenumeVeterinar
FROM consultatii C JOIN animale A ON C.AnimalID = A.AnimalID
JOIN veterinari V ON V.MedicID = C.MedicID
JOIN servicii S ON S.ServiciuID = C.ServiciuID
JOIN clienti Cl ON A.ClientID = Cl.ClientID
WHERE C.DataEfectuare >= @BeginDate AND C.DataEfectuare <= @Date”
```

Apar 2 parametri variabili, anume zilele care delimitează perioada, acestea trebuie trecute în textbox-uri înainte de execuție, altfel apare un mesaj de eroare. Apare în statistici boli.

8. Afișarea categoriilor de servicii împreună cu toate serviciile din categorie

```
„SELECT S.Denumire AS Denumire, Cat.Denumire AS Categorie, S.Preț AS Preț,
S.Detalii AS Detalii
FROM servicii S JOIN categoriiservicii Cat ON S.CategorieID = Cat.CategorieID”
```

Apare direct la încărcarea forms-ului pentru statistici boli.

Query-uri complexe cu subcereri

1. Afișarea animalelor din baza de date care nu au avut niciun diagnostic
„SELECT A.NumeAnimal, A.Vârstă, A.Specie, COALESCE(A.Rasă,'Nu are') AS Rasă,
CONCAT(C.Nume,' ',C.Prenume) AS NumeClient FROM animale A
JOIN clienti C ON A.ClientID=C.ClientID
WHERE AnimalID NOT IN (SELECT AnimalID FROM animale_cu_afectiuni)”

Se execută prin apăsarea butonului, apare în forms-ul de statistici animale.

2. Afișarea animalelor care au efectuat mai multe consultații decât media de consultații per animal

„SELECT A.NumeAnimal, A.Vârstă, A.Specie, COALESCE(A.Rasă,'Nu are') AS Rasă,
CONCAT(C.Nume,' ',C.Prenume) AS NumeClient, COUNT(Co.ConsultatieID) AS
NrConsultatii
FROM animale A JOIN clienti C ON A.ClientID=C.ClientID JOIN consultatii Co ON
Co.AnimalID = A.AnimalID
GROUP BY A.NumeAnimal, A.Vârstă, A.Specie, A.Rasă, C.Nume, C.Prenume
HAVING COUNT(Co.ConsultatieID) > (SELECT AVG(NrCon) FROM (SELECT
COUNT(C2.ConsultatieID) AS NrCon FROM consultatii C2 GROUP BY C2.AnimalID)
AS Subquery)”

Se execută prin apăsarea butonului, apare în forms-ul de statistici animale.

3. Afișarea tuturor veterinarilor care au acumulat o sumă mai mare prin consultații decât veterinarul ales din listă împreună cu sumele respective

„SELECT V.Nume, V.Prenume, SUM(S.Preț) AS Total
FROM veterinari V JOIN consultatii Co ON V.MedicID = Co.MedicID JOIN servicii S
ON Co.ServiciuID = S.ServiciuID
GROUP BY V.Nume, V.Prenume
HAVING SUM(S.Preț) > (SELECT SUM(S2.Preț) FROM servicii S2 JOIN consultatii Co2
ON S2.ServiciuID = Co2.ServiciuID JOIN veterinari V2 ON Co2.MedicID ON
V2.MedicID WHERE V2.CNP = @CNP)”

Apare un parametru variabil, acest query se execută pentru veterinarul selectat din lista de veterinari afișată în form. Dacă nu este selectat niciunul, apare un mesaj de eroare. Se află în forms-ul pentru statistici veterinari. În plus, într-un textbox se afișează și suma totală acumulată de medic prin query-ul următor:

"SELECT SUM(S.Pret) FROM servicii S JOIN consultatii Co ON S.ServiciuID = Co.ServiciuID JOIN veterinari V ON Co.MedicID = V.MedicID WHERE V.CNP = @CNP"

4. Afișarea clienților care a căror scoruri medii din recenzii sunt similare cu scorul mediu al clientului selectat împreună cu scorul

„SELECT CONCAT(Cl.Nume, ' ', Cl.Prenume) AS Nume, AVG(R.Scor) AS Medie
FROM clienti Cl JOIN reviews R ON Cl.ClientID = R.ClientID
WHERE Cl.CNP != @CNP1
GROUP BY Cl.Nume, Cl.Prenume
HAVING ABS(AVG(R.Scor) - (SELECT AVG(R2.Scor) FROM reviews R2 JOIN clienti Cl2 ON R2.ClientID=Cl2.ClientID WHERE Cl2.CNP = @CNP2)) <= 0.5”

Apare un parametru variabil, acest query se execută pentru clientul selectat din lista de clienți afișată în form. Dacă nu este selectat niciunul, apare un mesaj de eroare. Se află în forms-ul pentru statistici clienți. În plus, se afișează într-un textbox și recenzia medie dată de client prin query-ul următor:

"SELECT AVG(R.Scor) FROM reviews R JOIN clienti Cl ON R.ClientID=Cl.ClientID
WHERE Cl.CNP = @CNP"

5. Afișarea celui mai frecvent serviciu pentru fiecare categorie

„SELECT
C.Denumire AS Categorie,
(
SELECT TOP 1 S.Denumire
FROM servicii S
JOIN consultatii Co ON S.ServiciuID = Co.ServiciuID
WHERE S.CategorieID = C.CategorieID
GROUP BY S.Denumire
ORDER BY COUNT(Co.ConsultatieID) DESC
) AS ServiciuPopular
FROM categoriiservicii C”

Query-ul se execută automat la încărcarea forms-ului, apare în statistici boli.

Bibliografie

<https://www.youtube.com/watch?v=VX4wl7qlcbA&t=6468s&pp=ygUebXlzcWwgdmIzdWFsIHNoZWVpbyBmb3JtcyBhcHAg>

<https://www.basedash.com/blog/how-to-connect-mysql-to-visual-studio>

Cursul de Baze de Date