

# Aplicație pentru evidența medicamentelor într-o farmacie

Curcă Ștefan

Grupa 331 AA

#### Lista funcționalităților farmaciei:

- 1) **Gestionarea medicamentelor:** adăugarea, actualizarea și stergerea de medicamente din baza de date, inclusiv detalii precum denumire, preț, cantitatea disponibilă pe stoc, forma farmaceutică, etc.
- 2) **Gestionarea furnizorilor:** adăugarea și actualizarea de informații despre furnizori, inclusiv detalii de contact.
- 3) **Gestionarea rețetelor medicale:** verificarea și validarea acestora, precum și înregistrarea medicamentelor eliberate pe baza rețetelor.
- 4) **Monitorizarea stocului:** urmărirea cantității disponibile pentru fiecare medicament în timp real cu posibilitatea configurării unei avertizări pentru a notifica personalul farmaciei când stocul unui medicament atinge un nivel minim.
- 5) **Monitorizarea datelor de expirare:** Urmărirea datelor de expirare ale medicamentelor și generarea automată a alertelor înainte de expirarea acestora
- 6) **Gestionarea pacienților:** Înregistrarea și gestionarea datelor personale ale pacienților cu rețetă medicală.

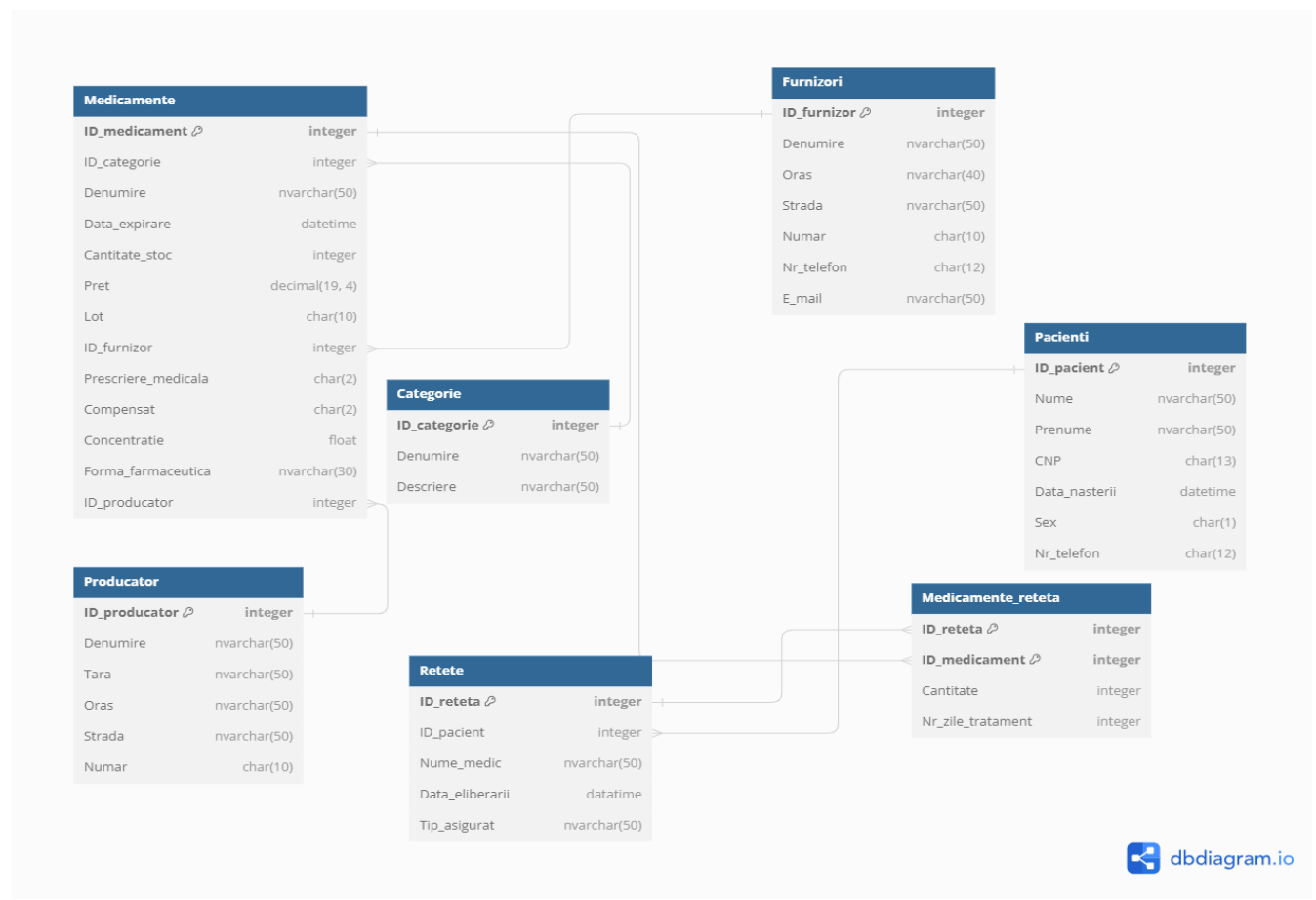
#### Restricții de proiectare:

- 1) Un medicament poate aparține unei singure categorii iar o categorie poate conține mai multe medicamente.
- 2) Un medicament are un singur producător, pot exista medicamente cu aceeași compoziție dar care sunt produse de altă companie sub un alt nume.
- 3) Un medicament are un singur furnizor, iar furnizorul poate comercializa mai multe medicamente.
- 4) Un pacient poate avea mai multe rețete dar rețeta este personalizată fiecărui pacient.

#### Entitățile bazei de date:

- Medicamente
- Categorii
- Furnizori
- Pacienți
- Producători
- Rețete

## Diagrama bazei de date:



## Identificarea relațiilor între tabele:

	Medicamente	Categorii	Furnizori	Pacienți	Producători	Rețete
Medicamente		1:1	1:1	1:N	1:1	1:N
Categorii	1:N					
Furnizori	1:N					
Pacienți	1:N					1:N
Producători	1:N					
Rețete	1:N			1:1		

## **Medicamente**

Din perspectiva tabelului MEDICAMENTE, vom încerca să identificăm tabelele cu care relaționează, și în ce formă.

Medicamente-Categorii: un medicament aparține unei singure categorii, există o relație de tip unu cu unu între aceste două tabele (1:1).

Medicamente-Furnizori: un medicament este achiziționat de la un singur furnizor, farmacia nu achiziționează în același timp un medicament de la doi furnizori diferiți, există o relație de tip unu cu unu între aceste două tabele (1:1).

Medicamente-Pacienți: un medicament poate fi achiziționat de mai mulți pacienți, există o relație de tip unu cu mai mulți între aceste două tabele (1:N).

Medicamente-Produsător: un medicament este produs de către o singură companie, poate exista un medicament cu aceeași compoziție însă acesta este produs de către altă companie sub o altă denumire, există o relație de tip unu cu unu între aceste două tabele (1:1).

Medicamente-Rețete: un medicament se poate afla pe mai multe rețete, există o relație de tip unu cu mai mulți între aceste două tabele (1:N).

## **Categorii**

Din perspectiva tabelului CATEGORII, vom încerca să identificăm tabelele cu care relaționează, și în ce formă.

Categorii-Medicamente: o categorie poate conține mai multe medicamente, există o relație de tip unu cu mai mulți între aceste două tabele (1:N).

## **Furnizori**

Din perspectiva tabelului FURNIZORI, vom încerca să identificăm tabelele cu care relaționează, și în ce formă.

Furnizori-Medicamente: un furnizor poate comercializa mai multe medicamente, există o relație de tip unu cu mai mulți între aceste două tabele (1:N).

## **Pacienți**

Din perspectiva tabelului PACIENȚI, vom încerca să identificăm tabelele cu care relaționează, și în ce formă.

Pacienți-Medicamente: un pacient poate avea prescrise mai multe medicamente, există o relație de tip unu cu mai mulți între aceste două tabele (1:N).

Pacienți-Rețete: un pacient poate primi mai multe rețete medicale, există o relație de tip unu cu mai mulți între aceste două tabele (1:N).

## **Producători**

Din perspectiva tabelului PRODUCĂTORI, vom încerca să identificăm tabelele cu care relaționează, și în ce formă.

Producători-Medicamente: un producător poate crea mai multe medicamente, există o relație de tip unu cu mai mulți între aceste două tabele (1:N).

## **Rețete**

Din perspectiva tabelului REȚETE, vom încerca să identificăm tabelele cu care relaționează, și în ce formă.

Rețete-Medicamente: o rețetă poate conține mai multe medicamente, există o relație de tip unu cu mai mulți între aceste două tabele (1:N).

Rețete-Pacienți: o rețetă are un singur pacient, fiecare rețetă este personalizată pentru un pacient anume, există o relație de tip unu cu unu între aceste două tabele (1:1).

Relațiile oficiale care există între tabelele bazei de date:

**Medicamente-Rețete** = 1:N + 1:N = N:N

**Medicamente-Furnizori** = 1:1 + 1:N = 1:N

**Medicamente-Producători** = 1:1 + 1:N = 1:N

**Medicamente-Categorii** = 1:1 + 1:N = 1:N

**Medicamente-Pacienți** = 1:N + 1:N = N:N

**Rețete-Pacienți** = 1:N + 1:1 = 1:N

Constrângeri de integritate:

## **MEDICAMENTE**

- Câmpul ID\_Medicament este cheie primară a entității medicamente și în plus acest câmp se autoincrementează.

- ID\_Furnizor, ID\_Categorie și ID\_Producător sunt chei externe pentru entitatea medicamente.
- Câmpul Prescriere\_medicală va fi marcat prin „DA” sau „NU”, astfel încât câmpul se referă la faptul că medicamentul în cauză are nevoie de prescriere medicală (rețetă) de la medic pentru a putea fi cumpărat de către un client. Pentru a preveni introducerea altor valori pentru acest câmp, am folosit o constrângere de tip CHECK în care am specificat că singurele valori acceptate sunt „DA” sau „NU”. În plus am utilizat o constrângere DEFAULT pentru acest câmp, căruia i-am setat valoarea „NU” drept valoare implicită.
- Câmpul Compensat va fi marcat prin „DA” sau „NU”, astfel încât câmpul se referă la faptul că medicamentul în cauză este compensat de casa de asigurări de sănătate. Pentru a preveni introducerea altor valori pentru acest câmp, am folosit o constrângere de tip CHECK în care am specificat că singurele valori acceptate sunt „DA” sau „NU”. Am utilizat pentru același câmp și o constrângere DEFAULT, căruia i-am setat valoarea implicită „NU”.
- Câmpurile pentru care este obligatoriu să introducem valori pentru fiecare înregistrare sunt: Denumire, ID\_Categorie, ID\_Producător, ID\_Furnizor, Data\_expirării, Cantitate\_stoc, Preț, Concentrație, Compensat și Prescriere\_medicală.

### **Categorii**

- Câmpul ID\_Categorie este cheie primară a entității categorii, în plus acest câmp se autoincrementează.
- Pentru câmpul Denumire este obligatoriu să introducem valori pentru fiecare înregistrare.

### **Furnizori**

- Câmpul ID\_Furnizor este cheie primară a entității furnizori, de asemenea câmpul acesta se autoincrementează.
- Câmpului Nr\_Telefon i s-a adăugat o constrângere UNIQUE, deoarece mai mulți furnizori diferiți nu pot avea același număr de telefon.
- Câmpurile pentru care este obligatoriu să introducem valori pentru fiecare înregistrare sunt: Denumire, Oraș, Stradă, Număr, Nr\_telefon.

### **Pacienți**

- Câmpul ID\_Pacient este cheie primară a entității pacienți, acest câmp se autoincrementează.
- Câmpului CNP i s-a adăugat o constrângere UNIQUE, deoarece mai mulți pacienți nu pot avea același CNP.
- Câmpul Sex va fi marcat prin „M” sau „F”. Pentru a preveni introducerea altor valori pentru acest câmp, am folosit o constrângere de tip CHECK în care am specificat că singurele valori acceptate sunt „M” sau „F”. Am utilizat pentru același câmp și o constrângere DEFAULT, căruia i-am setat valoarea implicită „M”.

- Câmpurile pentru care este obligatoriu să introducem valori pentru fiecare înregistrare sunt: Nume, Prenume, CNP si Nr\_telefon.

### **Producători**

- Câmpul ID\_Producător este cheie primară a entității producători, în plus acest câmp se autoincrementează.
- Câmpurile pentru care este obligatoriu să introducem valori pentru fiecare înregistrare sunt: Denumire, Țară, Oraș, Stradă și Număr.

### **Rețete**

- Câmpul ID\_Rețetă este cheie primară a entității rețete, de asemenea acest câmp se autoincrementează.
- Câmpul ID\_Pacient este cheie externă pentru entitatea rețete.
- Câmpurile pentru care este obligatoriu să introducem valori pentru fiecare înregistrare sunt: Nume\_medic, Data\_emiterii și Tip\_asigurat.

### **Medicamente\_rețetă**

- Câmpurile ID\_Rețetă și ID\_Medicament sunt chei primare ale acestui tabel de legătură. De asemenea acestea sunt și chei externe.
- Câmpurile pentru care este obligatoriu introducerea de valori pentru fiecare înregistrare sunt: Cantitate și Nr\_zile\_tratament.

### Funcționarea aplicației:

Aplicația este dezvoltată în limbajul de programare C# și are scopul de a facilita comunicarea personalului dintr-o farmacie cu o bază de date stocată în SQL Server Management. Deoarece aplicația este destinată utilizării interne accesul fiind posibil doar pentru angajații farmaciei nu există și o parte de autentificare. Interfața grafică este ușor de utilizat, facilitează navigarea personalului prin funcționalitățile esențiale, cum ar fi gestionarea medicamentelor, a rețetelor medicale și a altor entități. Personalul farmaciei poate accesa și actualiza informațiile despre medicamente, inclusiv stocurile disponibile, datele de expirare, detalii despre producători, precum și alte informații ale bazei de date. Aplicația oferă facilități pentru interogări avansate în baza de date pentru a extrage informații complexe și rapoarte personalizate. Aplicația este concepută pentru a fi compatibilă cu diverse dispozitive, asigurând accesul la informații în orice moment și de pe orice tip de dispozitiv.

### Interogări simple:

- 1) Afișarea medicamentelor dintr-o categorie introdusă de la tastatură.  

```
SELECT M.*  
FROM Medicamente M, Categorii C  
WHERE M.ID_Categorie = C.ID_Categorie AND C.Denumire LIKE '%' + {0} + '%'
```
- 2) Afișarea pacienților prin alegerea tipului de asigurat  

```
SELECT DISTINCT P.*  
FROM Pacienti P, Retete R  
WHERE P.ID_Pacient = R.ID_Pacient AND R.Tip_asigurat LIKE '%' + {0} + '%'
```
- 3) Afișarea pacienților ce au avut o reteta emisa in decursul acestei luni.  

```
SELECT P.*  
FROM Pacienti P, Retete R  
WHERE P.ID_Pacient = R.ID_Pacient AND R.Data_emiterii >= DATEADD(MONTH, -  
1, GETDATE())
```
- 4) Afișarea medicamentelor produse de un producător introdus de la tastatură.  

```
SELECT M.*  
FROM Medicamente M, Producatori P  
WHERE M.ID_Producator = P.ID_Producator AND P.Denumire LIKE '%' + {0} + '%'
```
- 5) Afișarea medicamentelor cumpărate de către un pacient prin introducerea CNP-ului acestuia de la tastatură.  

```
SELECT M.*  
FROM Medicamente M, Pacienti P, Medicamente_reteta MR, Retete R  
WHERE P.ID_Pacient = R.ID_Pacient AND R.ID_Reteta = MR.ID_Reteta AND  
MR.ID_Medicament = M.ID_Medicament AND P.CNP LIKE '%' + {0} + '%'
```
- 6) Afișarea pacienților ce au cumpărat un anumit medicament introdus de la tastatură  

```
SELECT DISTINCT P.*  
FROM Pacienti P, Retete R, Medicamente_reteta MR, Medicamente M  
WHERE P.ID_Pacient = R.ID_Pacient AND R.ID_Reteta = MR.ID_Reteta AND  
MR.ID_Medicament = M.ID_Medicament AND M.Denumire LIKE '%' + {0} + '%'
```
- 7) Afișarea medicamentelor comercializate de un furnizor introdus de la tastatură  

```
SELECT M.*  
FROM Medicamente M, Furnizori F  
WHERE M.ID_Furnizor = F.ID_Furnizor AND F.Denumire LIKE '%' + {0} + '%'
```



Interogări complexe:

- 1) Afișarea medicamentelor produse de fiecare producător, care expiră cel mai devreme.

```
SELECT M.*
FROM Medicamente M JOIN
(SELECT ID_Producator, MIN(Data_expirarii) AS Data_expirarii
FROM Medicamente
GROUP BY ID_Producator) Y ON M.ID_Producator = Y.ID_Producator AND
M.Data_expirarii = Y.Data_expirarii JOIN Producatori P ON M.ID_Producator =
P.ID_Producator
ORDER BY M.Data_expirarii
```

- 2) Afișarea medicamentelor ce au valoarea concentrației peste media valorii concentrației a tuturor medicamentelor

```
SELECT M.*
FROM Medicamente M
WHERE M.Concentratie > (SELECT AVG(Concentratie) FROM Medicamente)
ORDER BY M.Concentratie
```

- 3) Afișarea medicamentelor cu prețul cel mai ridicat comercializate de fiecare furnizor

```
SELECT M.*
FROM Medicamente M JOIN
(SELECT ID_Furnizor, MAX(Pret) AS Pret
FROM Medicamente
GROUP BY ID_Furnizor) Y ON M.ID_Furnizor = Y.ID_Furnizor AND M.Pret = Y.Pret
JOIN Furnizori F ON M.ID_Furnizor = F.ID_Furnizor
ORDER BY M.Pret
```

- 4) Afișarea pacienților ce au vârsta peste media de vârstă a tuturor pacienților

```
SELECT P.*
FROM Pacienti P
WHERE DATEDIFF(YEAR, P.Data_nasterii, GETDATE()) >
(SELECT AVG(DATEDIFF(YEAR, Data_nasterii, GETDATE()))
FROM Pacienti)
ORDER BY P.Data_nasterii
```

- 5) Afișarea medicamentelor cu stocul cel mai redus din fiecare categorie

```
SELECT M.*
FROM Medicamente M JOIN
(SELECT ID_Categorie, MIN(Cantitate_stoc) AS Cantitate_stoc
FROM Medicamente
GROUP BY ID_Categorie) Y ON M.ID_Categorie = Y.ID_Categorie AND
M.Cantitate_stoc = Y.Cantitate_stoc JOIN Categori C ON M.ID_Categorie =
C.ID_Categorie
ORDER BY M.Cantitate_stoc
```