Evidenta consultatiilor la o clinica medicala

Marius-Gabriel Ciurea

334AA

**Descrierea cerintei :** “Aplicatie pt evidenta consultatiilor la o clinica medicala.”

Aplicatia dezvoltata e bazata pe principiul de usurare a procesului de management al datelor, in cazul unei clinici medicale.

**Tehnologii folosite (proiectul este o aplicatie web) :**

* Python3 (cu framework-ul Flask, bazat pe engine-ul de template-uri Jinja) - pentru partea de back-end
* HTML, CSS - pentru partea de front-end
* Microsoft SQL Server - baza de date

**Prezentarea bazei de date :**

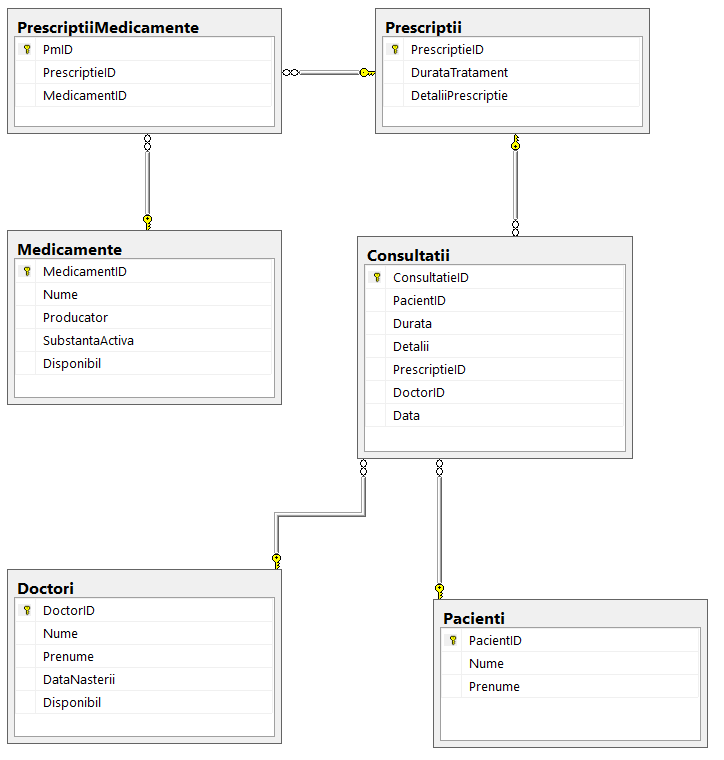
Intrucat, conform cerintei, aplicatia trebuie focusata in jurul consultatiilor de la clinica, baza de date este de asemenea focusata in jurul tabelei “Consultatii”.

Legate de aceasta sunt tabelele “Pacienti”, “Doctori” si “Prescriptii”.

Intre acestea se stabilesc urmatoarele legaturi :

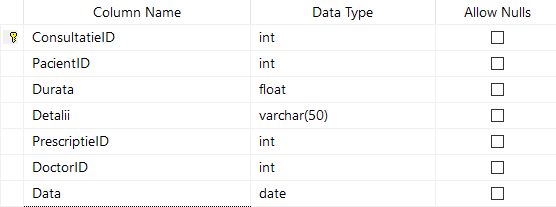
* + Consultatii : Doctori – N:1 (un doctor poate sustine mai multe consultatii, dar o consultatie e sustinuta de un singur doctor)
  + Consultatii : Pacienti – N:1( un pacient poate avea mai multe consultatii, dar o consultatie este facuta unui singur pacient)
  + Consultatii : Prescriptii – 1:1 (o consultatie corespunde unei singura prescriptii, iar o prescriptie apartine unei singure consultatii)

Fara o legatura directa cu tabela “Consultatii” este tabela “Medicamente”, aceasta fiind legata de tabela Prescriptii printr-o relatie N:N (un medicament poate aparea in mai multe prescriptii, iar o prescriptie poate contine mai multe medicamente).

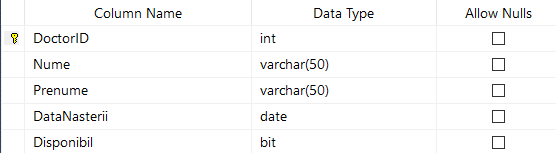
Aceasta legatura este facilitata de tabela (de legatura) “PrescriptiiMedicamente”.

**Prezentarea tabelelor :**

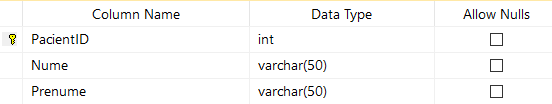
Consultatii :



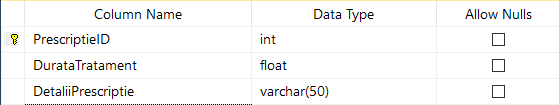
Doctori :



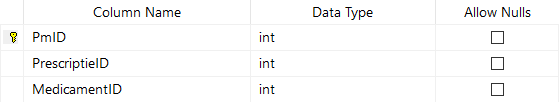
Pacienti :



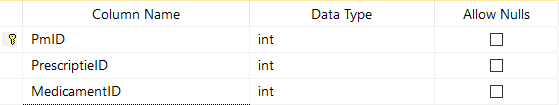
Prescriptii :



Medicamente :



PrescriptiiMedicamente :



***Nota :*** In forma actuala, tabelele sunt minimale din perspectiva datelor pe care le contin si au ca scop buna functionare a aplicatiei. In cadrul unei baze de date reale, tabelele ar fi mai mari, continand mult mai multe date despre fiecare obiect in parte; datele suplimentare nu ar fi modificat substantial functionarea aplicatiei, dar ar fi ingreunat interactiunea cu utilizatorul la prima vedere (ex. : o consultatie ar ocupa un ecran intreg si aplicatia ar fi mai greu de prezentat).

**Constrangeri de integritate :**

Desi in forma actuala tabelele nu necesita constrangeri suplimentare de integritate, in urma completarii cu date suplimentare exista posibilitatea necesitatii acestora (ex. : adaugarea coloanei “Sex” in tabela “Pacienti” ar necesita constrangerea de integritate ‘Masculin’ sau ‘Feminin’ asupra campului varchar(50) ).

**Functionarea aplicatiei :**

**Password-protected :**

Intrucat baza de date a unei clinici medicale ar trebui sa fie privata si inaccesibila persoanelor neautorizate, pentru accesul la informatii, respectiv la editarea acestora, este nevoie de o parola.

Astfel, in urma accesarii url-ului corespunzator (in cazul nostru, localhost:5000 ), utilizatorul este intampinat de pagina de login :



Introducerea unei parole gresite duce la reincarcarea paginii de login, impreuna cu un mesaj : “Parola introdusa este gresita”.



**Meniul principal :**

In urma introducerii parolei corecte (in cazul de fata, parola corecta este “parola”), utilizatorul este redirectionat catre meniul principal.

Din meniul principal avem access la diferite informatii, cu posibilitatea de intoarcere la pagina principala.

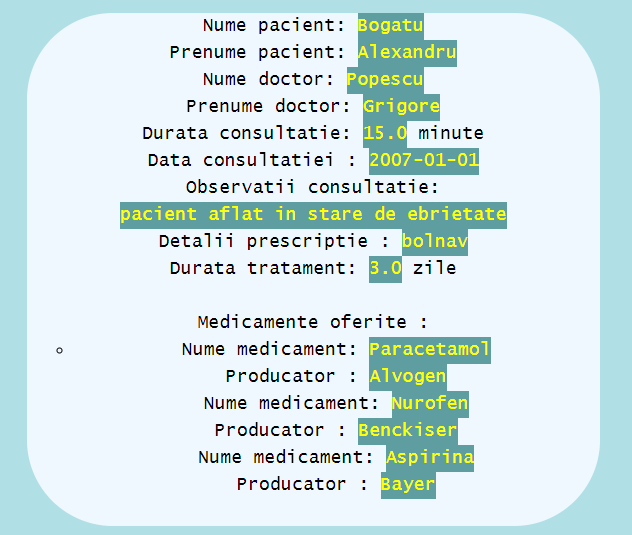
Nota : Mi-am luat libertatea sa insotesc meniul de un titlu ce poate fi transformat intr-un acronim vesel care sa insoteasca paginile aplicatiei.



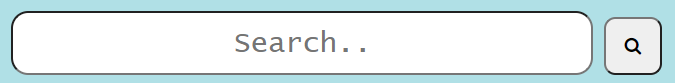
**Consultatii :**

Pagina “Consultatii” contine informatii despre toate consultatiile efectuate la clinica, acestea fiind afisate sub forma de lista, fiecare consultatie fiind delimitata de un chenar.

In cadrul unei consultatii, informatiile utile sunt scoase in evidenta.

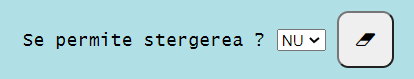


In cadrul paginii “Consultatii”, utilizatorul are mai multe posibilitati :

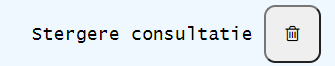
* Se poate intoarce la meniul principal, prin apasarea butonului “Home”
* Poate cauta o consultatie dupa numele/prenumele pacientului sau al doctorului responsabil de aceasta
* Poate cauta o consultatie dupa anul in care a fost facuta

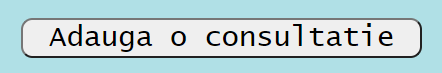


* Poate activa optiunea de stergere a datelor, prin selectarea optiunii “DA” in meniul “Se permite stergerea” si apasarea butonului alaturat

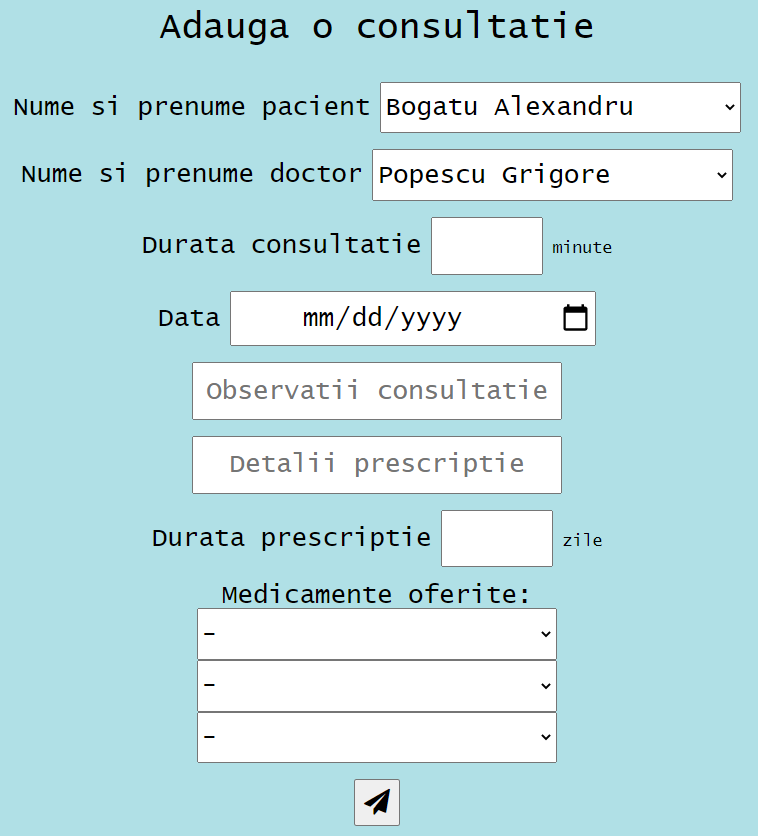


Activarea acestei functii determina aparitia unei optiuni suplimentare in chenarul fiecarei consultatii :



* Poate adauga o consultatie prin accesarea meniului de adaugare

Apasarea butonului trimite utilizatorul catre pagina corespunzatoare introducerii unei consultatii



**Pacienti :**

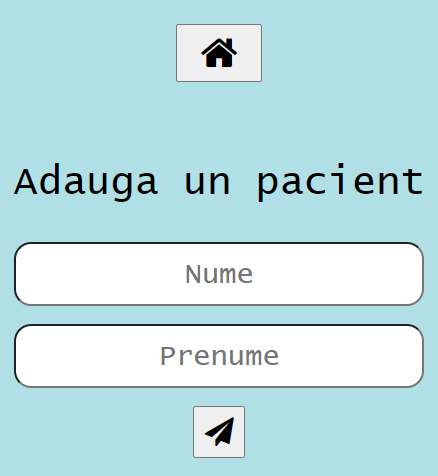
Pagina contine informatii despre pacientii tratati la clinica, precum numele si prenumele pacientului, cat si numarul si numele doctorilor care l-au tratat.

Din punct de vedere al optiunilor pentru utilizatori, pagina este similara cu meniul “Consultatii”.



Observatie : un numar de doctori egal cu 0 semnifica faptul ca pacientul a fost inregistrat, dar nu exista nicio consultatie pentru acesta

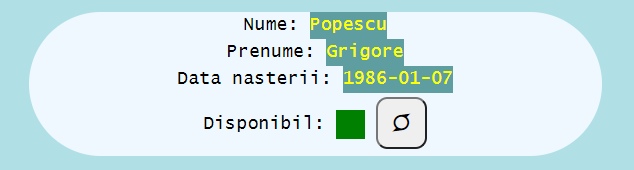
In urma apasarii butonului “Adauga un pacient”, utilizatorul este directionat catre meniul de adaugare.



**Doctori :**

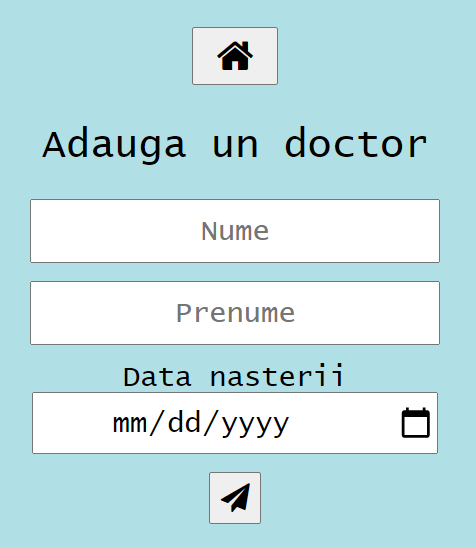
Pagina ofera informatie despre doctorii clinicii, inclusiv disponibilitatea acestora.

Pe langa optiunile prezente la paginile anterioare, pagina “Doctori” ofera utilizatorului optiunea de a schimba disponibilitatea unui doctor, prin apasarea unui buton.



Intrucat accesul la baza de date este permis doar persoanelor autorizate, acestea au posibilitatea sa altereze informatia din baza de date (modificare, stergere).

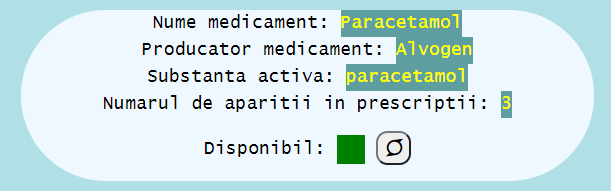
Pagina de adaugare a unui doctor este similara celei de adaugare a unui pacient.



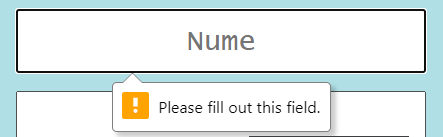
**Medicamente :**

Pagina corespunzatoare medicamentelor este asemanatoare cu cea pentru doctori prin faptul ca prezinta optiunea de disponibilitate, care poate fi updatata prin apasarea butonului alaturat.

Pe langa aceasta, pagina contine toate optiunile mentionate precedent.



 Meniul de adaugare a unui medicament este simplu si foarte asemanator cu cele mentionate precedent.

Mentiune : In cadrul meniurilor de introducere a datelor in baza de date, campurile sunt protejate impotriva trimiterii acestora fara valori.

**Interogari :**

In cadrul aplicatiei sunt folosite 26 de interogari diferite. Numarul mare este datorat existentei optiunilor de “Cautare”, prezenta la foarte multe dintre pagini, dar si complexitatii aplicatiei.

Din totalul de interogari, 4 sunt interogari complexe, restul fiind interogari simple dintre care majoritatea necesita legaturi intre tabele (JOIN).

Majoritatea interogarilor contin un parametru variabil (sub forma de string).

Pe langa interogari se afla 6 comenzi de DELETE, 2 de UPDATE, respectiv 6 de INSERT.

**Lista interogarilor folosite :**

**Mentiune :** Interogarile vor fi prezentate sub forma de string (“ “) pentru a scoate in evidenta elemental variabil. Aceasta abordare are dezavantajul ca expune baza de date la injectii SQL, dar din moment ce nu au acces la aplicatie decat persoanele autorizate, nu este o problema semnificativa.

* **Interogari simple :**

“SELECT DISTINCT d.Nume, d.Prenume, p.PacientId

FROM Doctori d INNER JOIN Consultatii c ON c.DoctorId = d.DoctorId

INNER JOIN Pacienti p ON p.PacientId = c.PacientId

WHERE p.Pacientid = " + str(person['id']) + "

ORDER BY p.Pacientid"

"SELECT PrescriptieId

FROM Consultatii

WHERE ConsultatieId = " + str(idConsultatieSters)

"SELECT p.Nume, p.Prenume, d.Nume, d.Prenume, c.Durata, c.Detalii, c.Data, c.ConsultatieId

FROM Consultatii c INNER JOIN Pacienti p on c.PacientID = p.PacientID INNER JOIN Doctori d ON c.DoctorID = d.DoctorID

WHERE p.Nume LIKE '%" + str(formsData) + "%' or p.Prenume LIKE '%" + str(formsData) + "%'

OR d.Nume LIKE '%" + str(formsData) + "%'

OR d.Prenume LIKE '%" + str(formsData) + "%'

ORDER BY c.Data DESC"

"SELECT p.Nume, p.Prenume, d.Nume, d.Prenume, c.Durata, c.Detalii, c.Data, c.ConsultatieId

FROM Consultatii c INNER JOIN Pacienti p ON c.PacientID = p.PacientID INNER JOIN Doctori d ON c.DoctorID = d.DoctorID

ORDER BY c.Data DESC"

“SELECT m.Nume, m.Producator

FROM Medicamente m

INNER JOIN PrescriptiiMedicamente pm

ON m.MedicamentID = pm.MedicamentID

INNER JOIN Consultatii c ON c.PrescriptieID = pm.PrescriptieID

WHERE c.ConsultatieID = " + str(consultatieID)

“SELECT Disponibil

FROM Medicamente

WHERE MedicamentId = " + str(medicamentID)

“SELECT p.Nume, p.Prenume

FROM Pacienti p"

“SELECT d.Nume, d.Prenume

FROM Doctori d"

“SELECT m.Nume

FROM Medicamente m")

“SELECT TOP 1 PrescriptieId

FROM Prescriptii

WHERE DurataTratament = '" + str(durataPrescriptie) + "' AND DetaliiPrescriptie = '" + str(detaliiPrescriptie) + "' "

“SELECT p.PacientId

FROM Pacienti p

WHERE p.Nume = '" + str(numePacient) + "' AND p.Prenume = '" + str(prenumePacient) + "'"

“SEELCT d.DoctorId

FROM Doctori d

WHERE d.Nume = '" + str(numeDoctor) + "' AND d.Prenume = '" + str(prenumeDoctor) + "'"

“SELECT MedicamentId

FROM Medicamente

WHERE Nume = '" + str(med) + "' "

“SELECT Disponibil

FROM Doctori

WHERE DoctorId = " + str(doctorID)

"SELECT DoctorID, Nume, Prenume, DataNasterii, Disponibil

FROM Doctori

WHERE Nume LIKE '%" + str(formsData) + "%'

OR Prenume LIKE '%" + str(formsData) + "%'"

“SELECT DoctorID, Nume, Prenume, DataNasterii, Disponibil FROM Doctori"

“SELECT p.PacientId, p.Nume, p.Prenume, COUNT(DISTINCT c.DoctorId) FROM Pacienti p LEFT JOIN Consultatii c ON p.PacientID = c.PacientID WHERE p.Nume LIKE '%" + str(formsData) + "%' OR p.Prenume LIKE '%" + str(formsData) + "%'

GROUP BY p.PacientId, p.Nume, p.Prenume

ORDER BY p.PacientId"

"SELECT p.PacientId, p.Nume, p.Prenume, COUNT(DISTINCT c.DoctorId)

FROM Pacienti p LEFT JOIN Consultatii c ON p.PacientID = c.PacientID GROUP BY p.PacientId, p.Nume, p.Prenume

ORDER BY p.PacientId"

"SELECT p.PacientId, p.Nume, p.Prenume, COUT(DISTINCT c.DoctorId) FROM Pacienti p LEFT JOIN Consultatii c ON p.PacientID = c.PacientID GROUP BY p.PacientId, p.Nume, p.Prenume

ORDER BY p.PacientId"

“SELECT DISTINCT p.DetaliiPrescriptie, p.DurataTratament, COUNT(DISTINCT pm.MedicamentID)

FROM Prescriptii p LEFT JOIN Consultatii c

ON c.PrescriptieID = p.PrescriptieID

LEFT JOIN PrescriptiiMedicamente pm

ON p.PrescriptieID = pm.PrescriptieID

WHERE c.ConsultatieID = "+ str(consultatieID) + "

GROUP BY p.DetaliiPrescriptie, p.DurataTratament"

"SELECT m.MedicamentId, m.Nume, m.Producator, m.SubstantaActiva, COUNT(pm.MedicamentId), m.Disponibil

FROM Medicamente m LEFT JOIN PrescriptiiMedicamente pm

ON m.MedicamentID = pm.MedicamentID

WHERE m.nume LIKE '%" + formsData + "%'

OR m.producator LIKE '%" + formsData + "%'

OR m.SubstantaActiva LIKE '%" + formsData + "%'

GROUP BY m.Nume, m.Producator, m.SubstantaActiva, m.Disponibil, m.MedicamentId"

"SELECT m.MedicamentId, m.Nume, m.Producator, m.SubstantaActiva, COUNT(pm.MedicamentId), m.Disponibil

FROM Medicamente m

LEFT JOIN PrescriptiiMedicamente pm

ON m.MedicamentID = pm.MedicamentID

GROUP BY m.Nume, m.Producator, m.SubstantaActiva, m.Disponibil, m.MedicamentId"

* **Interogari complexe :**

“SELECT p.Nume, p.Prenume, d.Nume, d.Prenume, c.Durata, c.Detalii, c.Data, c.ConsultatieId

FROM Consultatii c INNER JOIN Pacienti p ON c.PacientID = p.PacientID

INNER JOIN Doctori d ON c.DoctorID = d.DoctorID

WHERE c.ConsultatieID IN (SELECT cc.ConsultatieId

FROM Consultatii cc

WHERE YEAR(cc.Data) = " + str(receivedDate) + ") "

“SELECT DoctorID, Nume, Prenume, DataNasterii, Disponibil

FROM Doctori

WHERE DoctorId IN (SELECT dd.DoctorId

FROM Doctori dd

WHERE YEAR(dd.DataNasterii) = " + str(birthYear) + ")"

“SELECT p.PacientId, p.Nume, p.Prenume, COUNT(DISTINCT c.DoctorId)

FROM Pacienti p LEFT JOIN Consultatii c ON p.PacientID = c.PacientID

WHERE p.PacientId IN (SELECT pp.PacientId

FROM Pacienti pp

LEFT JOIN Consultatii cc ON pp.PacientID = cc.PacientID

GROUP BY p.PacientId

HAVING COUNT(DISTINCT cc.DoctorId) = " + str(nrDoctori) + ")

GROUP BY p.Pacientid, p.Nume, p.Prenume

ORDER BY p.PacientId”

“SELECT m.MedicamentId, m.Nume, m.Producator, m.SubstantaActiva, COUNT(pm.MedicamentId), m.Disponibil

FROM Medicamente m LEFT JOIN PrescriptiiMedicamente pm

ON m.MedicamentID = pm.MedicamentID

WHERE m.MedicamentID IN (SELECT mm.MedicamentId

FROM Medicamente mm

LEFT JOIN PrescriptiiMedicamente ppm

ON mm.MedicamentID = ppm.MedicamentID

GROUP BY mm.MedicamentID

HAVING COUNT(ppm.MedicamentId) = " + str(nrAparitii) + ")

GROUP BY m.Nume, m.Producator, m.SubstantaActiva, m.Disponibil, m.MedicamentId”

**Insert, Update, Delete :**

* **Insert :**

“INSERT INTO Prescriptii (DurataTratament, DetaliiPrescriptie)

VALUES ('" + str(durataPrescriptie) + "','" + str(detaliiPrescriptie) + "')"

INSERT INTO Consultatii (PacientId, Durata, Detalii, PrescriptieId, DoctorId, Data)

VALUES (" + str(pacientId[0][0]) + "," + str(durataConsultatie) + ",'" + str(detaliiConsultatie) + "'," + str(prescriptieId[0][0]) + "," + str(doctorId[0][0]) + ", '" + str(dataConsultatie) + "')"

INSERT INTO PrescriptiiMedicamente (PrescriptieId, MedicamentId) VALUES (" + str(prescriptieId[0][0]) + "," + str(medicamentId[0][0]) + ")"

“INSERT INTO Pacienti (Nume, Prenume)

VALUES ('" + str(numePacient) + "','" + str(prenumePacient) + "')")

INSERT INTO Doctori (Nume, Prenume, Disponibil, DataNasterii) VALUES ('" + str(numeDoctor) + "','" + str(prenumeDoctor) + "' ,1, '" + dataNasterii + "')"

“INSERT INTO Medicamente (Nume, Producator, SubstantaActiva, Disponibil)

VALUES ('" + str(nume) + "','" + str(producator) + "','" + str(substantaActiva) + "',1" + ")")

* **Update :**

"UPDATE Doctori SET Disponibil = " + str(disp) + "

WHERE DoctorId = " + str(doctorID)

“UPDATE Medicamente SET Disponibil = " + str(disp) + "

WHERE MedicamentID = " + str(medicamentID))

* **Delete :**

“DELETE FROM Doctori

WHERE DoctorId = " + str(idDoctorSters))

"DELETE FROM Pacienti

WHERE PacientId = " + str(idPacientSters)

“DELETE FROM Consultatii

WHERE ConsultatieId = " + str(idConsultatieSters))

“DELETE FROM PrescriptiiMedicamente

WHERE PrescriptieId = " + str(prescriptieID))

"DELETE FROM Prescriptii

WHERE PrescriptieId = " + str(prescriptieID)

“DELETE FROM Medicamente

WHERE MedicamentId = " + str(idMedicamentSters))