 **UNIVERSITATEA TEHNICĂ “GHEORGHE ASACHI” IAȘI**

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**

**SPECIALIZAREA CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI**

**DISCIPLINA BAZE DE DATE**

Temă Baze de Date

**Farmacia**

(Gestiunea activității unei farmacii)

**Coordonator: Avram Sorin**

**Student: Ganea Luiza-Andreea**

**Grupa: 1308A**

**Iasi, 2023**

Cuprins

[Titlul proiectului: Farmacie (Gestiunea activității unei farmacii) 3](#_Toc124077877)

[Tehnologii utilizate 3](#_Toc124077878)

[Interfața grafică 4](#_Toc124077879)

[Tranzacții 4](#_Toc124077880)

[Descrierea cerințelor și modul de organizare a proiectului 5](#_Toc124077881)

[Descrierea funcțională a aplicației: 7](#_Toc124077882)

[Descrierea detaliată a entităților și a relațiilor dintre tabele: 7](#_Toc124077883)

[Conflicte 8](#_Toc124077884)

[Constrângeri 9](#_Toc124077885)

[Diagrama Entitate Relație 11](#_Toc124077886)

[Conexiunea cu baza de date: 11](#_Toc124077887)

# Titlul proiectului: Farmacie (Gestiunea activității unei farmacii)

Analiza, proiectarea și implementarea unei baze de date și a aplicației aferente care să modeleze activitatea unei farmacii cu privire la gestiunea clienților, produselor si a activității acesteia (aprovizionări și vânzări).

Aplicația permite înregistrarea clienților și medicamentelelor în baza de date a farmaciei, pentru ca, ulterior, pe baza id-urilor acestora fiind posibile vânzările. La introducerea medicamentelor în baza de date, nu este necesară specificarea cantității (deci, initial este **zero**) întrucât aceasta se stabilește în urma aprovizionărilor, respective vânzarilor.

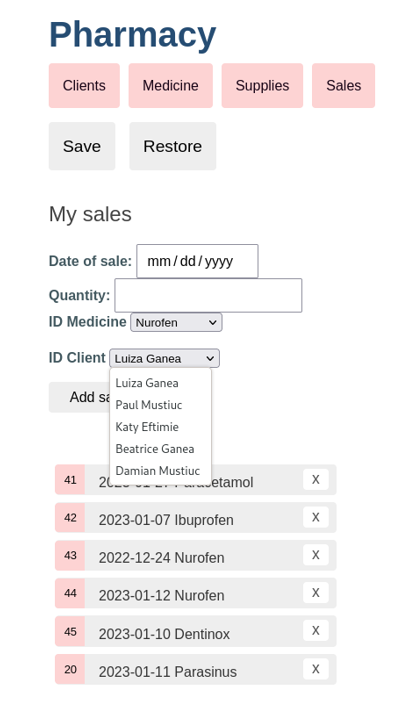
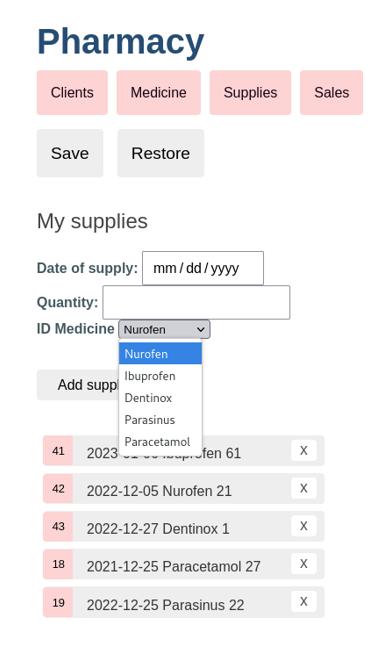
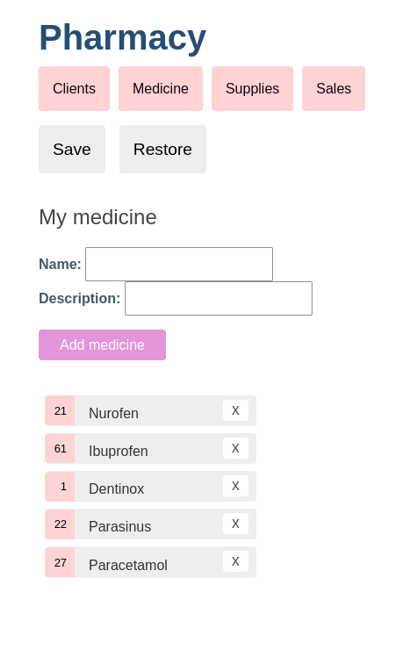
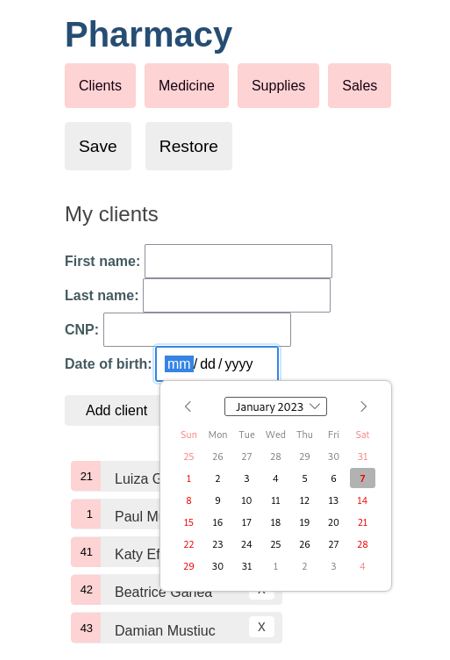
# Tehnologii utilizate:

**Back-end**-ul a fost realizat în *Java*, utilizând framework-ul *Spring* pentru a expune sub forma unui API REST metodele CRUD necesare.

Pentru realizarea bazei de date am folosit *Oracle*, iar pentru a testa instrucțiunile SQL folosite am folosit *Oracle SQL Developer*.

Pentru dezvoltarea interfeței grafice (**Front-end**) am dezvoltat un client pentru API-ul menționat anterior utilizând framework-ul *Angular* și limbajele *HTML, CSS* si *TypeScript*.

# Interfața grafică:



# Tranzacții:

Pe interfață sunt plasate doua butoane ce permit executarea tranzacțiilor. **Save** ne salvează starea actuală a aplicației iar **Restore** ne permite să ne întoarcem la starea salvată.

# Descrierea cerințelor și modul de organizare a proiectului:

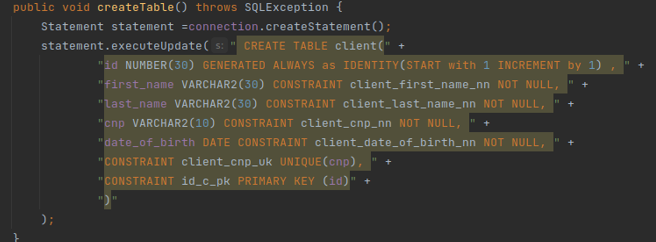
Structura urmează modelul unui magazin generic, într-o farmacie fiind necesară ținerea unei evidențe asupra clienților, medicamentelor, aprovizionărilor și vânzărilor. Aplicația web pe care am dezvoltat-o vine în ajutorul farmaciștilor, fiind foarte ușor și sigur de utilizat.

Operațiile SQL utilizate sunt CRUD (Create, Read, Update, Delete) iar informațiile de care avem nevoie sunt cele legate de:

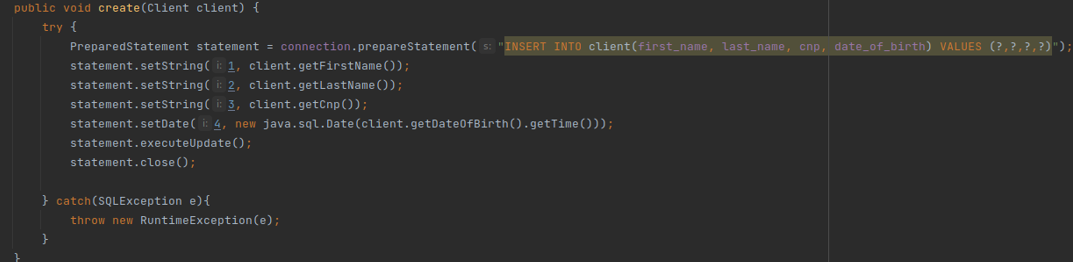
**-Clienți :** avem nevoie de datele de identificare ale clienților (nume, prenume, cnp, data nașterii) pentru a-i stoca în baza noastră de date. Ulterior, clienților le pot fi asociate vânzări de medicamente prin intermediul id-urilor lor (id-uri generate automat). Nu se pot elimina din baza de date clienții ai căror id-uri corespund cel puțin unei vânzări existente!

**Creare tabela Clienți:**

Pentru generarea id-urilor folosim autoincrementul (creăm o secvență care să înceapă de la un anumit număr și să incrementeze numărul atunci când înserăm date). La fel procedăm și în cazul celorlalte tabele.



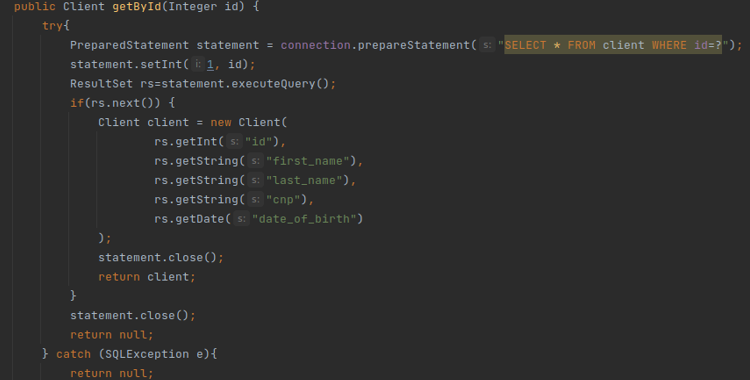
**Adăugare client:**



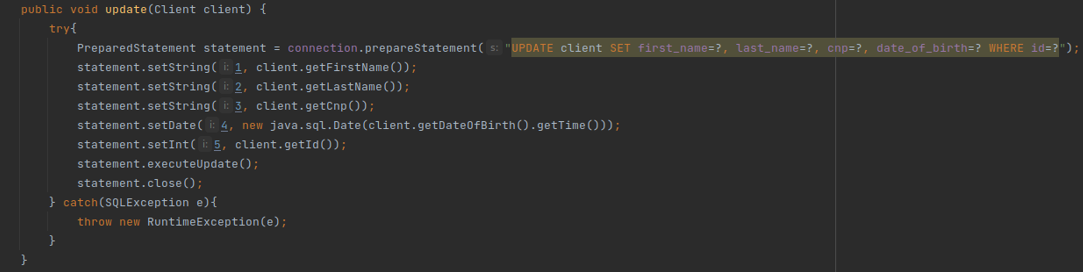
**Obținere clienți:**



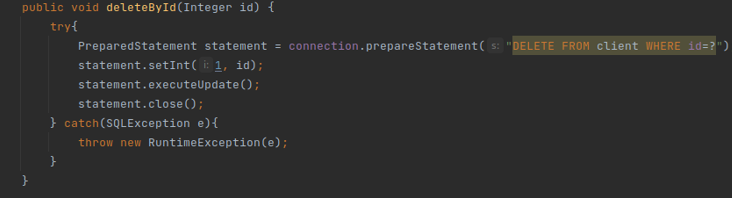
**Obținere client după id:**



**Actualizare client:**



**Ștergere client:**



**-Medicamente:** informațiile de interes legate de medicamente sunt numele, descrierea, cantitatea, necesitarea unei prescripții si vârsta minima pe care trebuie să o aibă clienții care vor să cumpere medicamentul respectiv. Id-urile medicamentelor sunt generate automat la crearea acestora, și sunt folosite pentru a face aprovizionări și vânzări. Nu se pot elimina din baza de date medicamentele pentru care există vânzări!

**-Aprovizionări:** ne interesează id-ul medicamentului pentru care se face aprovizionarea, cantitatea acestuia și data aprovizionării.

**-Vânzări:** asemănător aprovizionărilor, informațiile corespunzătoare unei vânzări sunt id-ul medicamentului vândut, cantitatea acestuia, și data vânzării. De asemenea, este esențial să cunoaștem id-ul clientului care achiziționează medicamentul. Cu privire la cantitatea medicamentului și data vânzării, nu se pot introduce valori care să intre în conflict cu aprovizionările: spre exemplu, nu putem vinde o cantitate indisponibilă de medicamente, și totodata, nu putem face o vânzare înaintea unei aprovizionări care să ne asigure necesarul de medicament din punctul de vedere al cantității.

# Descrierea funcțională a aplicației:

Principalele funcționalități ale aplicației sunt:

⸰Evidența clienților

⸰Evidența medicamentelor

⸰Evidența aprovizionărilor

⸰Evidența vânzărilor

# Descrierea detaliată a entităților și a relațiilor dintre tabele:

**Tabelele** aplicației sunt:

⸰Clienți (Client)

⸰Medicamente (Medicine)

⸰Aprovizionări (Supply)

⸰Vânzări (Sale)

În proiectarea acestei baze de date s-au identificat tipurile de relații 1:1, 1:n și n:1.

Între tabelele **Medicamente** și **Aprovizionări** avem o relație de tip 1:1 (one-to-one) deoarece fiecărui medicament din tabelul Medicamente îi corespunde o intrare în tabelul Aprovizionări. Câmpul comun al acestor două tabele este **id\_medicament**.

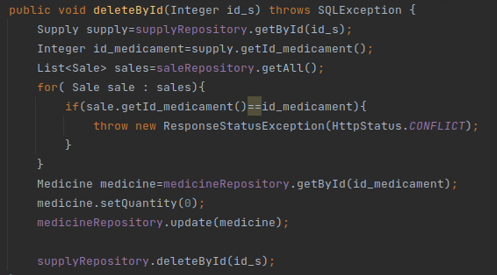
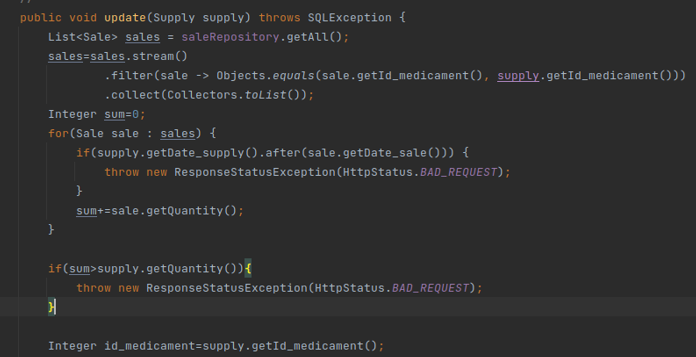
Între tabelele **Vânzări** și **Medicamente** se stabilește o relație de tip n:1 (many-to-one) deoarece un medicament poate fi vândut de mai multe ori. Pentru aceste două tabele avem campul comun **id\_medicament**.

Relatia n:1 (many-to-one) se regăsește și între tabelele **Vânzări** si **Clienți** întrucât un client poate aduce mai multe vanzări. Oricărei vânzări îi corespunde intrarea **id\_client** din tabelul Clienți.

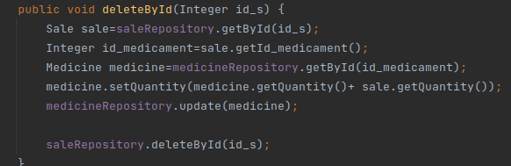
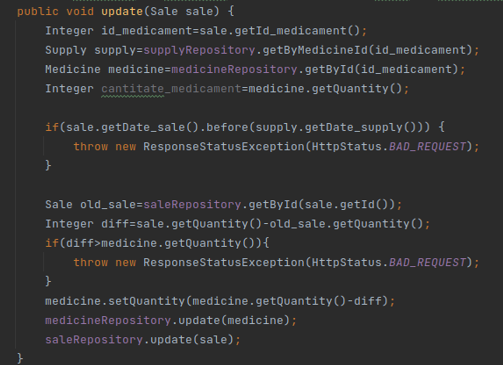
De asemenea, relațiile 1:n (one-to-many) se reflectă prin intermediul tabelelor **Medicamente/Clienți** și **Vânzări**.

# Conflicte:

1. La **adăugarea** unei **aprovizionări**, actualizăm cantitatea medicamentului corespunzător. Astfel, atunci când **modificăm** o aprovizionare, verificăm ca suma cantităților de la toate vânzările medicamentului respectiv să fie mai mica sau egală cu canitatea actuală a medicamentului. La **ștergerea** unei aprovizionări, i se setează canitate zero medicamentului, și se verifică să nu existe nicio vânzare care să îi corespundă acestuia



1. La **adăugarea** unei **vânzări**, verificăm ca data să fie dupa data aprovizionării din punct de vedere cronologic. De asemenea, valoarea cantității vânzării trebuie să fie mai mica sau egală cu cea a medicamentului. La **modificarea** unei vânzări, avem în vedere ca data să fie validă și diferența de cantitate să fie cuprinsă în cantitatea actuală de medicament. La **ștergerea** unei vânzări, creștem cantitatea de medicament cu valoarea corespunzatoare vânzării șterse.

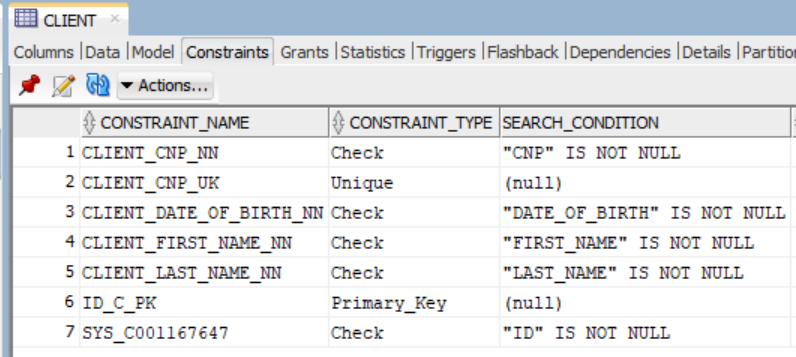


# Constrângeri:

⸰Primary-key-urile sunt generate de baza de date printr-un mechanism de tip autoincrement.

⸰Foreign key-urile sunt selectate dintr-o listă de tip drop-down, valoarea lor fiind preluată automat din aplicație.

**Clienți**



⸰Id **primary key** pentru a avea un identificator unic

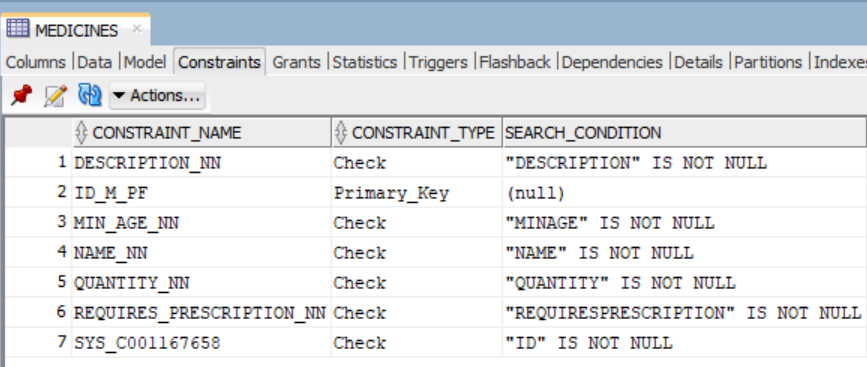
⸰First Name: **not null**

⸰Last Name: **not null**

⸰Cnp **unique** pentru a avea un identificator unic

⸰Date of Birth: **not null**

**Medicamente**



⸰Id: **primary key** pentru a avea un identificator unic

⸰Name: **not null**

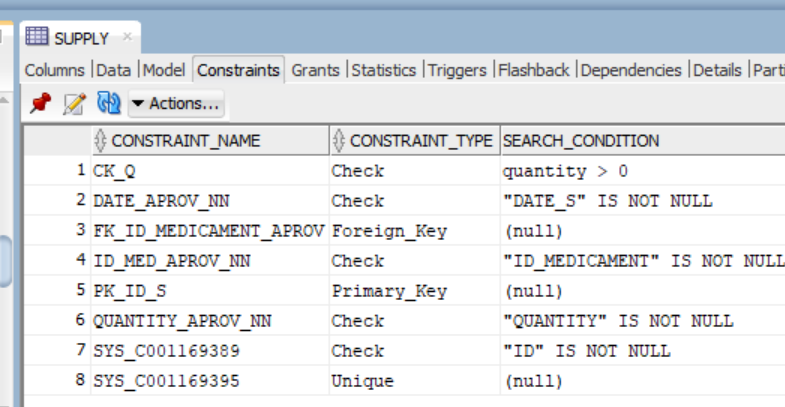
⸰Description: **not null**

⸰Quantity: **not null**

⸰Requires Prescription: **not null**

⸰Minimum age: **not null**

**Aprovizionări**



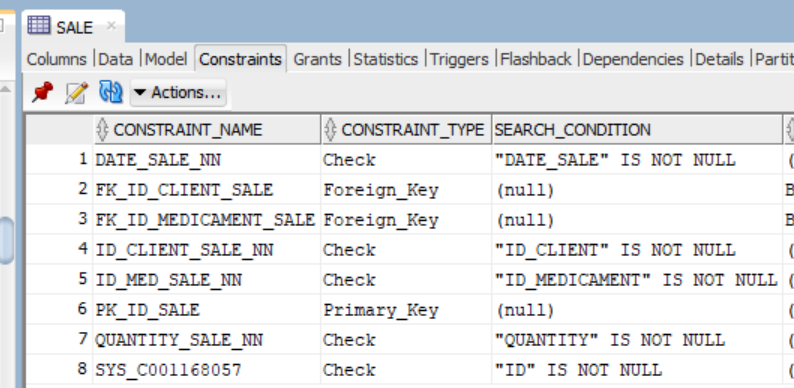
⸰Id: **primary key** pentru a avea un identificator unic

⸰Date of Supply: **not null**

⸰Quantity: **check** (quantity>0) verifică să fie cantitatea adăugată o valoare pozitivă

⸰Id medicine: **foreign key** crează legatura între 2 câmpuri din tabelele **Aprovizionări** și **Medicament**

**Vânzări**



⸰Id: **primary key** pentru a avea un identificator unic

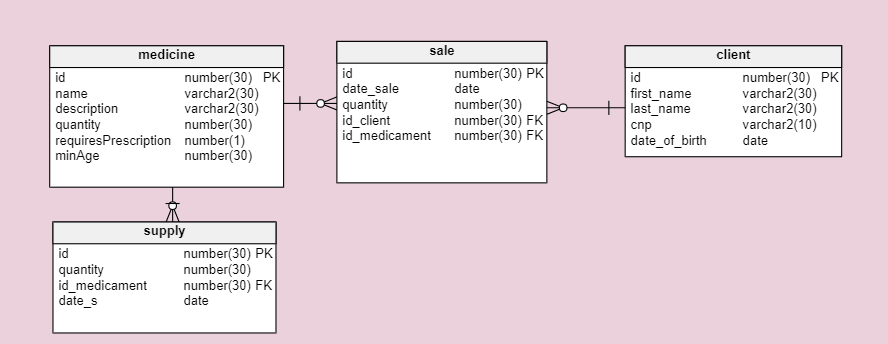
⸰Date of Sale: **not null**

⸰Quantity: **not null**

⸰Id client: **foreign key** crează legatura între 2 câmpuri din tabelele **Vânzări** și **Clienți**

⸰Id medicine: **foreign key** crează legatura între 2 câmpuri din tabelele **Vânzări** și **Medicamente**

# Diagrama Entitate Relație:



# Conexiunea cu baza de date:

