

**Gestiunea unei farmacii**

**Coordonator,**

**Avram Sorin**

**Student,**

**Mahu Petrisor – 1309A**

* **Cerintele si organizarea proiectului**

Aplicatia este creata pentru a gestiona vanzarile care se fac intr-o farmacie (bonuri fiscale), inregistrarea medicamentelor in sistem, a furnizorilor, a clientilor si datelor acestora, cu posibilitatea de a le oferi carduri de fidelitate pentru ca acestia sa poata primi reduceri per bon fiscal. Aplicatia va facilita mentinerea unor centralizari de

Farmacia nu face parte dintr-un lant de farmacii, este o simpla farmacie, cu o varietate redusa de medicamente.

Farmacia se aprovizioneaza de la mai multi furnizori. Acestia aprovizioneaza cu mai multe tipuri de medicamente, dar maxim o data pe zi si nu neaparat zilnic.

Medicamentele vandute se vand doar in cantitati intregi si nu mai mult de 5 itemi din acelasi medicament pe un bon.

Nu se pot vinde mai multe medicamente decat prezinta stocul, implicit nu se pot face comenzi de medicamente pentru a fi ridicate de clienti ulterior.

Pretul medicamentelor nu se schimba in timp.

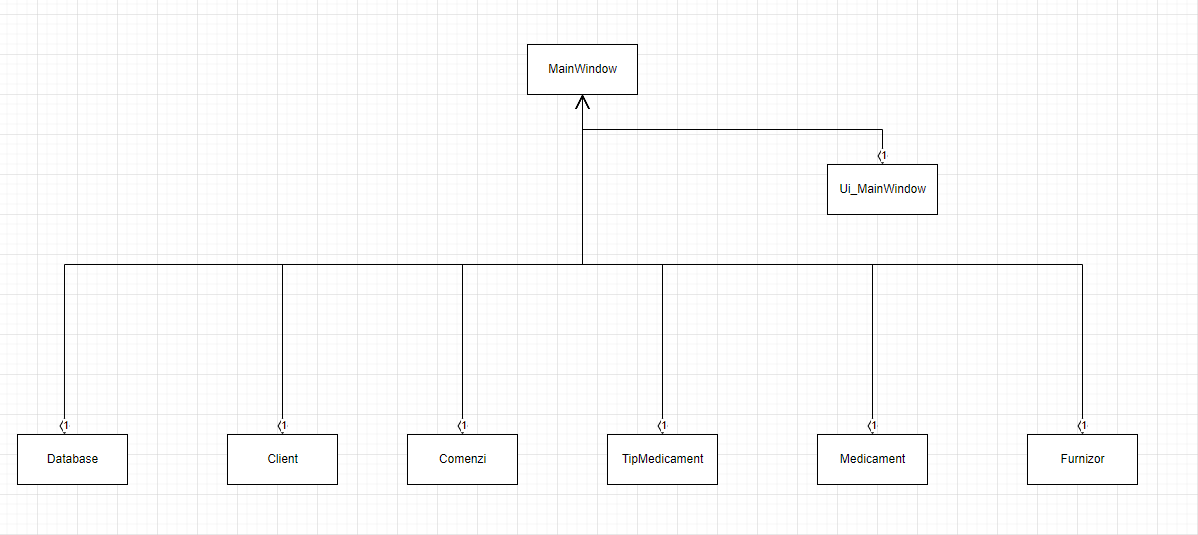
Daca un client are card de fidelitate, acesta va primi o reducere de 10% din valoarea bonului fiscal.

Aplicatia va avea functionalitati precum:

* Date despre medicamente
* Cate bonuri fiscale detine fiecare client
* Afisarea detaliilor despre clienti
* Afisarea pretului de pe fiecare bon sau cele din lunile noiembrie/decembrie
* Medicamentele vandute de-a lungul timpului

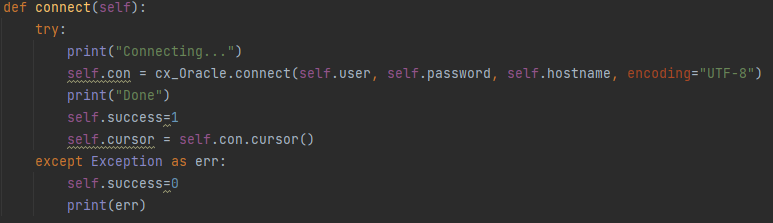
Informatiile de care avem nevoie sunt legate de:

* Tipul medicamentului
* Datele despre medicamentul propriu zis
* Furnizorii
* Informatiile de pe bonurile fiscale
* Clientii si datele acestora in cazul in care le este facut un card de fidelitate
* **Tehnologii folosite pentru front-end si back-end**



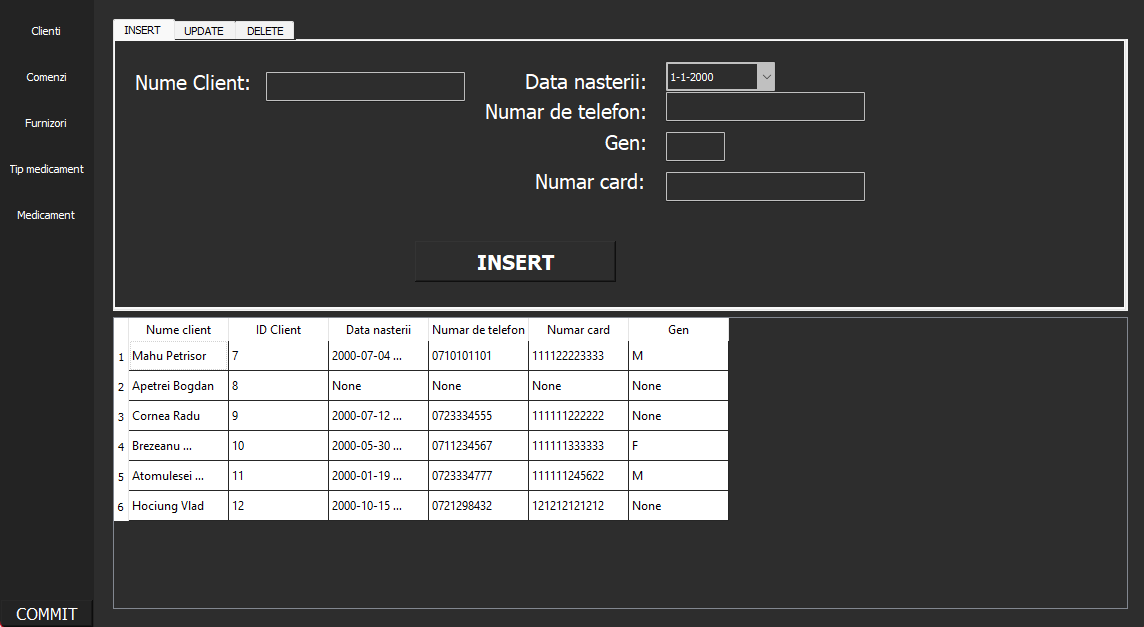
Baza de date a fost proiectata in Oracle. Pe partea de interfata grafica, am folosit libraria PyQt5 din Python. Datele sunt introduse in baza de date prin widget-uri de tip Line Edit si Combo box, precum si unele afisari de inregistrari din tabelele Table Widget.

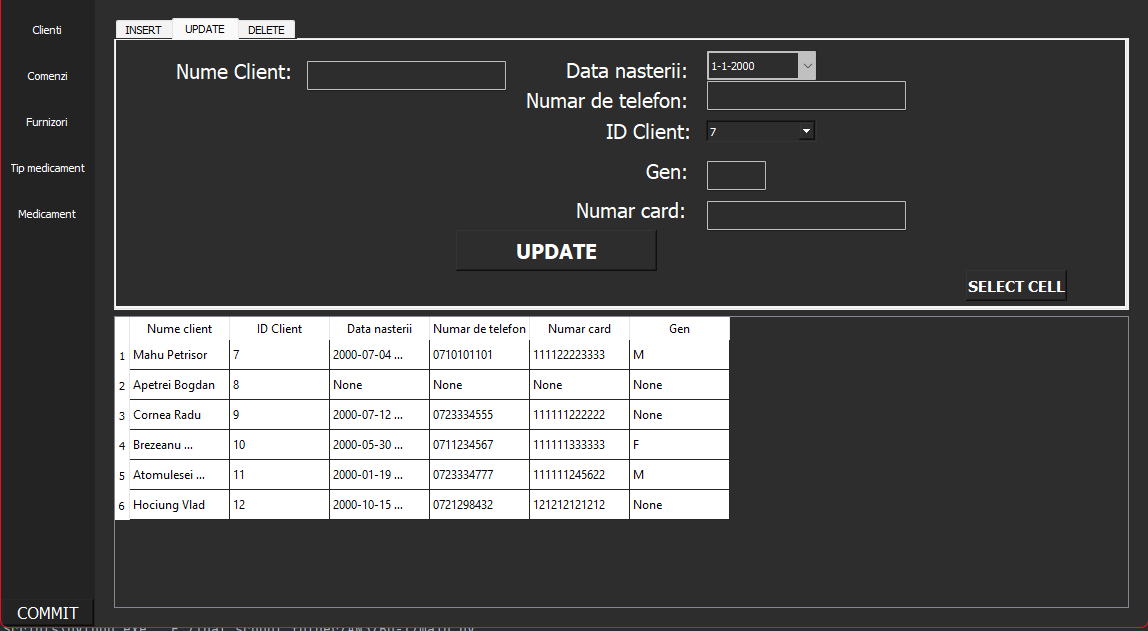
* **Conectarea la baza de date (libraria cx\_oracle)**

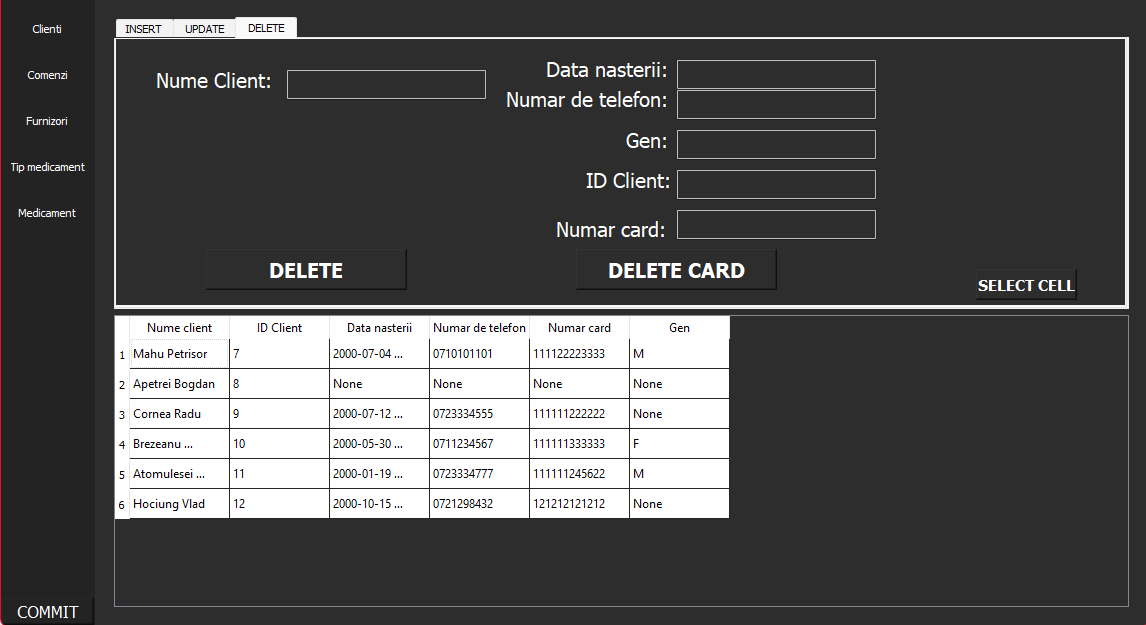


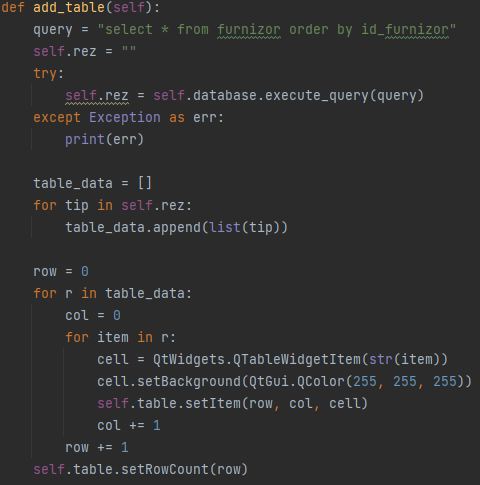
Afisarea interfetei grafice este intarziata pana la finalizarea conexiunii aplicatiei la baza de date.

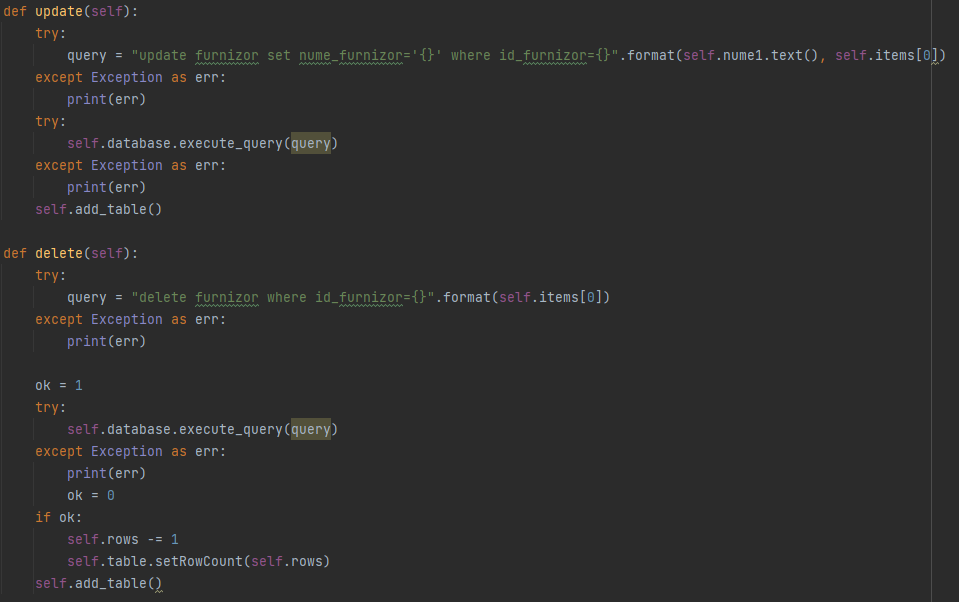
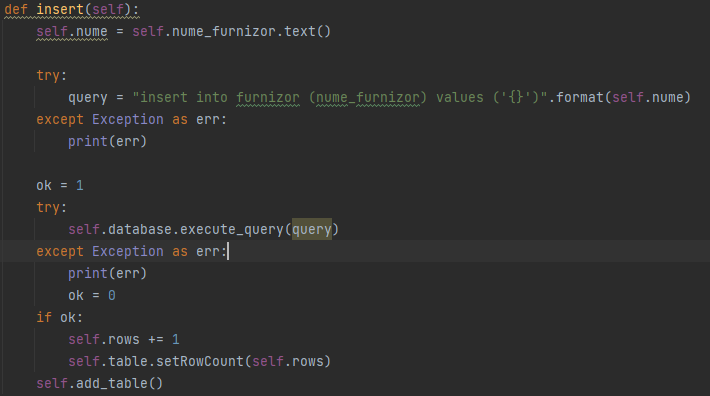
* **Interfata grafica si folosirea acesteia**







In partea din stanga se poate comuta intre tabele (Clienti contine tabelele clienti si date\_card iar comenzi contine tabela bon\_fiscal). In partea de sus putem comuta intre operatiile pe care vrem sa le facem. In Update/Delete tab, prin apasarea pe o inregistrare din tabela, urmata de apasarea butonului Select Cell se va selecta (si afisa in casutele line edit) o inregistrare asupra careia putem executa operatii. Prin apasarea botunului de commit se vor salva modificarile in baza de date. 



Functiile insert, update si delete executa operatiunile descrise in numele lor, iar functia add\_table afiseaza inregistrarile din tabele in TableWidget. Dupa fiecare operatie, datele din widget sunt actualizate cu cele noi.

* **Structura proiectului**

Tabelele folosite in aplicatie sunt:

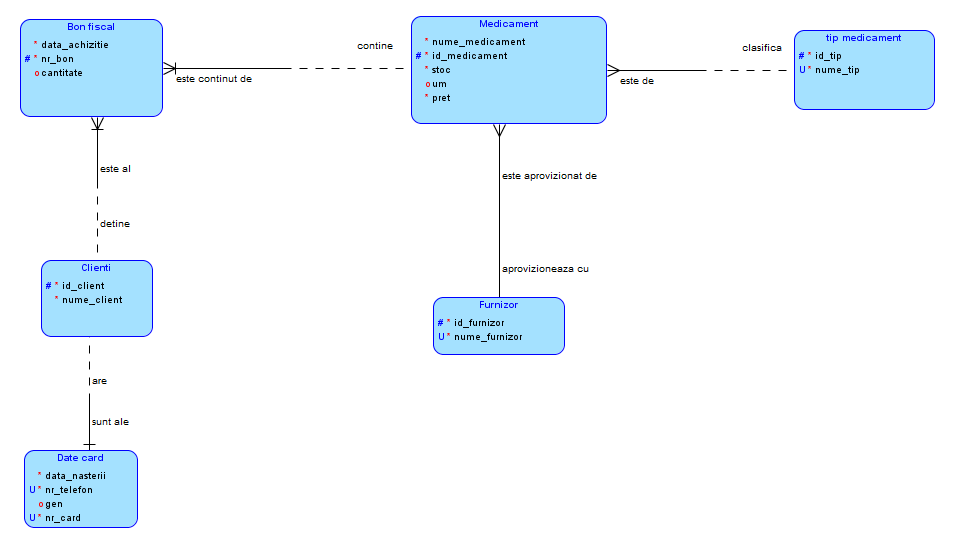
* *tip\_medicament*: contine tipurile de medicamente care pot fi vandute
* *furnizor*: contine id-ul si numele furnizorului
* *medicament*: aici sunt stocate datele despre un medicament specific, ca id-ul sau, tipul de medicament, numele, unitatea de masura, pretul, stocul si cine furnizeaza produsul.
* *bon\_fiscal*: fiecare bon va avea in componenta sa id-ul clientului, produsul cumparat si cantitatea cumparata, data achizitiei si numarul id al bonului.
* *clienti*: contine id-ul clientului si numele sau
* *date\_card*: contine datele despre clientii care au optat pentru un card de fidelitate, date precum data nasterii (clientul trebuie sa fi implinit 18 ani pentru a putea avea un card de fidelitate), numarul de telefon, id-ul sau si numarul unic al cardului.
* **Relationarea entitatilor**

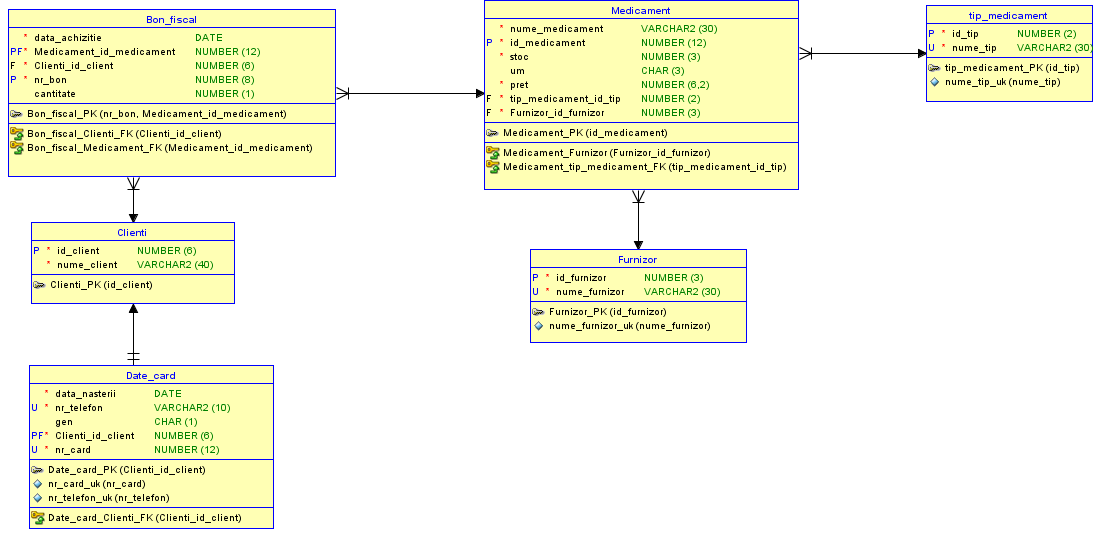
Entitatile au relatii 1:1 si 1:n.

Relatia 1:1 este realizata de catre tabelele *clienti* si *date\_card*, deoarece un client nu poate avea mai multe carduri, iar un card apartine doar unui singur client. Legatura dintre cele doua tabele este realizata prin campul id\_client.

Relatiile 1:n sunt realizate de catre:

* *Tip\_medicament* si *medicament*, deoarece un medicament poate fi doar de un singur tip, dar un tip de medicament poate fi atribuit mai multor medicamente. Legatura dintre cele doua tabele este realizata prin campul id\_tip.
* *Furnizor* si *medicament*, deoarece un furnizor poate aproviziona cu mai multe medicamente, dar un anumit medicament nu poate fi adus de mai multi furnizori. Legatura dintre cele doua tabele este realizata prin campul id\_furnizor.
* *Medicament* si *bon\_fiscal*, deoarece pe un bon fiscal pot exista mai multe medicamente, dar nu poate fi inregistrat de mai multe acelasi medicament. Legatura dintre cele doua tabele este realizata prin campul id\_medicament.
* *Bon\_fiscal* si *clienti*, deoarece clientii pot veni sa cumpere de la farmacie de mai multe ori. Un bon fiscal apartine unui singur client, iar un client poate avea de-a lungul timpului mai multe bonuri fiscale. Legatura dintre cele doua tabele este realizata prin campul id\_client.

**Modelul logic**

**Modelul relational**

* **Constrangerile utilizate**

1. Primary key:

* id\_tip(tabela *tip\_medicament*)
* id\_medicament(tabela *medicament*)
* id\_furnizor(tabela *furnizor*)
* medicament\_id\_medicament si nr\_bon (tabela *bon\_fiscal*), prin care ne asiguram ca nu exista doua bonuri cu acelasi id si acelasi produs.
* id\_client (tabela *clienti*)
* clienti\_id\_client(tabela *date\_card*).

Constrangerea primary key este folosita pentru a identifica in mod unic o inregistrare. Campul respectiv nu poate avea valoare nula.

1. Foreign key:

* tip\_medicament\_id\_tip si furnizor\_id\_furnizor(tabela *medicament*) prin care se face legatura intre tabelele *Medicament* si *tip\_medicament* respectiv *Medicament* si *Furnzior*. Astfel se atribuie unui medicament un tip de medicament inregistrat in sistem, respectiv se retine furnizorul corespunzator medicamentului.
* medicament\_id\_medicament si clienti\_id\_client(tabela *bon\_fiscal*) prin care se face legatura intre tabelele *Medicament* si *Bon\_fiscal*, respectiv *Bon\_fiscal* si *Client*. Astfel se pune in bonul fiscal id-ul medicamentului inregistrat in sistem, respectiv se atribuie bonului fiscal id-ul cumparatorului.
* clienti\_id\_client(tabela *date\_card*) prin care se face legatura intre tabelele *client* si *date\_card*. Astfel se face legatura intre client si cardul de fidelitate, in cazul in care acesta detine unul.

1. Unique key:

* nume\_tip(tabela *tip\_medicament*), deoarece nu ar trebui sa avem in sistem de mai multe ori acelasi tip de medicament.
* nume\_furnizor(tabela *furnizor*), deoarece nu ar trebui sa existe doi furnizor cu acelasi nume.
* nr\_card(tabela *date\_card*) si nr\_telefon, deoarece nu ar trebui sa existe doua carduri cu acelasi numar de identificare sau sa fie atribuite aceluiasi numar de telefon.

1. Check si trigger:

In mare parte sunt folosite verificari care asigura ca unele campuri NOT NULL contin valori valide (ca de exemplu stocul si pretul sa nu fie negative, cantitatea de pe bon sa nu depaseasca valoarea de 5 si sa nu fie mai mica decat 1, campul nr\_card sa aiba obligatoriu 12 cifre in componenta).

Sunt prezente verificari de tip regex prin care se asigura validarea unor campuri ca nume\_furnizor, nume\_client, nume\_medicament care nu trebuie sa contina numere sau ca nr\_telefon sa nu contina litere.

Exista trigger pentru data\_achizitie si data\_nasterii pentru a ne asigura ca bonul nu va fi inregistrat pe o data din viitor, respectv clientul nu a indeplinit varsta de 18 ani pentru a-si putea face un card de fidelitate.

1. Autoincrement

Sistemele de autoincrement sunt folosite pentru cheile primare id\_client(tabela *clienti*), nr\_bon(tabela *bon\_fiscal*), **id\_tip**(tabela *tip\_medicament*), **id\_medicament**(tabela *medicament*), **id\_furnizor**(tabela *furnizor*) pentru a usura inregistrarea lor in sistem, privandu-ne de necesitatea de a le introduce manual.

Pentru toate sistemele de autoincrement se incrementeaza valoarea cu 1. Id\_client porneste de la 1, nr\_bon porneste de la 10000000, **id\_tip** porneste de la 10, **id\_furnizor** porneste de la 500**, id\_medicament** porneste de la 5000.