UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

DEPARTAMENTUL: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

**PROIECT**

**BAZE DE DATE**

COORDONATOR ȘTIINȚIFIC:

VASILE SILVIU-LAURENȚIU

STUDENT:

COJOCARU ANDREI-LAURENȚIU

BUCUREȘTI

2022

UNIVERSITATEA DIN BUCURESTI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

DEPARTAMENTUL: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

**BAZA DE DATE A UNUI SPITAL**

COORDONATOR ȘTIINȚIFIC:

VASILE SILVIU-LAURENȚIU

STUDENT:

COJOCARU ANDREI-LAURENȚIU

BUCUREȘTI

2022

**Cuprins**

1. Prezentarea modelului

2. Regulile modelului

3. Diagrama Entitate-Relatie

3.1. Reprezentarea diagramei

3.2. Descrierea entitatilor, atributelor, cheilor, relatiilor si a cardinalitatilor

3.2.1. Descrierea entitatilor, atributelor si a cheilor

3.2.1.1. Tabelul cabinet\_medical

3.2.1.2. Tabelul doctor

3.2.1.3. Tabelul proiect

3.2.1.4. Tabelul pacient

3.2.1.5. Tabelul programare

3.2.1.6.Tabelul tratament

3.2.1.7.Tabelul terapie

3.2.1.8.Tabelul medicament

3.2.2. Descrierea relatiilor si a cardinalitatilor

3.2.2.1 cabinet\_medical – doctor

3.2.2.2 doctor – proiect

3.2.2.3 doctor - programare

3.2.2.4 pacient – programare

3.2.2.5 programare – tratament

3.2.2.6 tratament – terapie

3.2.2.7 tratament – medicament

4. Diagrama Conceptuala

4.1. Reprezentare diagrama

4.2 Descrierea constrangerilor de integritate

4.2.1 Tabelul cabinet\_medical

4.2.2 Tabelul doctor

4.2.3 Tabelul proiect

4.2.4 Tabelul doctor\_atasat\_la\_proiect

4.2.5 Tabelul pacient

4.2.6 Tabelul programare

4.2.7 Tabelul tratament

4.2.8 Tabelul prescriptii\_programare

4.2.9 Tabelul terapie

4.2.10 Tabelul medicament

4.2.11 Tabelul terapie\_inclusa\_in\_tratament

4.2.12 Tabelul medicamente\_continute\_de\_tratament

4.3 Schemele relationale

4.3.1 Schemele relationale

4.3.2 Descrierea constrangerilor on delete

5. Scriptul SQL

5.1 Introducere

5.2 Etapa de drop

5.3 Crearea tabelelor si a constrangerilor

5.3.1 Crearea tabelului cabinet\_medical

5.3.2 Crearea tabelului doctor

5.3.3 Crearea tabelului proiect

5.3.4 Crearea tabelului doctor\_atasat\_la\_proiect

5.3.5 Crearea tabelului pacient

5.3.6 Crearea tabelului programare

5.3.7 Crearea tabelului tratament

5.3.8 Crearea tabelului prescriptii\_programare

5.3.9 Crearea tabelului terapie

5.3.10 Crearea tabelului medicament

5.3.11 Crearea tabelului terapie\_inclusa\_in\_tratament

5.3.12 Crearea tabelului medicamente\_continute\_de\_tratament

5.4 Introducerea datelor in baza de date

5.4.1 Introducerea datelor in tabelul cabinet\_medical

5.4.2 Introducerea datelor in tabelul doctor

5.4.3 Introducerea datelor in tabelul proiect

5.4.4 Introducerea datelor in tabelul doctor\_atasat\_la\_proiect

5.4.5 Introducerea datelor in tabelul pacient

5.4.6 Introducerea datelor in tabelul programare

5.4.7 Introducerea datelor in tabelul tratament

5.4.8 Introducerea datelor in tabelul prescriptii\_programare

5.4.9 Introducerea datelor in tabelul terapie

5.4.10 Introducerea datelor in tabelul medicament

5.4.11 Introducerea datelor in tabelul terapie\_inclusa\_in\_tratament

1. Prezentarea modelului

Tema aleasa pentru proiect este „Baza de date a unui spital”. Modelul creat poate fi aplicat in cadrul mai multor spitale, dar acesta urmareste strict activitatea unui singur spital.

Este foarte important ca activitatea unui spital sa fie monitorizata cu ajutorul unei baze de date structurata corect deoarece avem de aface cu un volum foarte mare de date.

De asemenea, este important sa reducem posibilitatea de a introduce erori in baza de date, urmarea unui astfel de eveniment putand avea consecinte grave pentru pacienti.

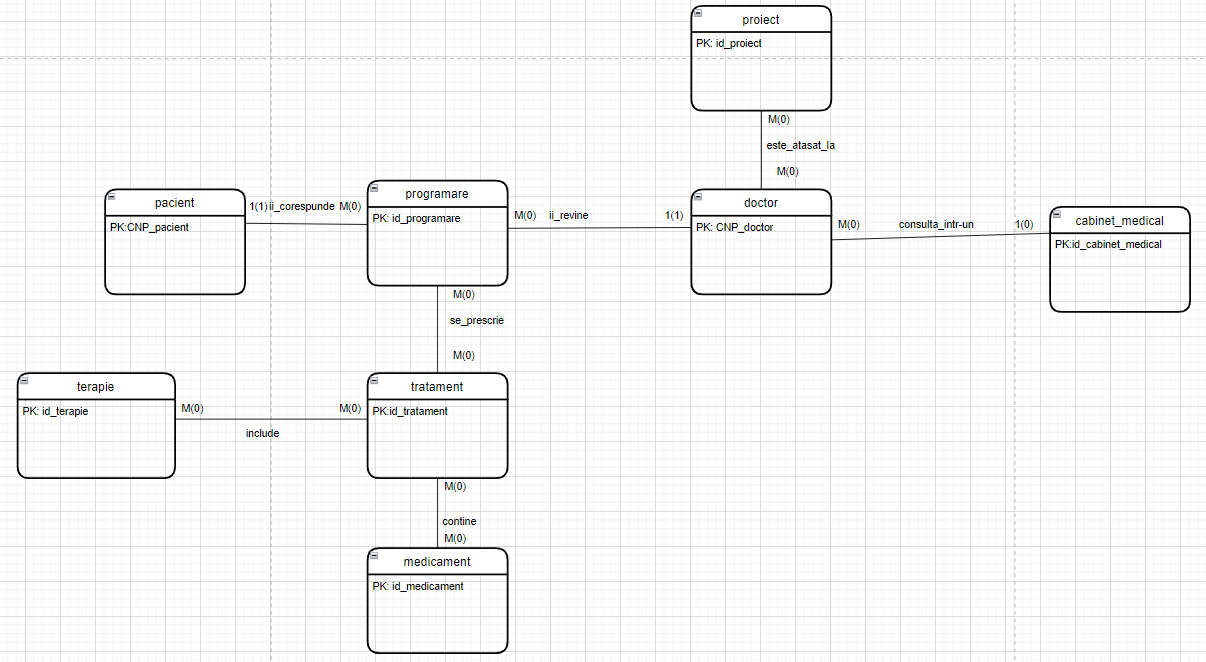
Avand in vedere cele mentionate, am realizat o baza de date pentru stocarea a cateva date administrative (doctori, cabinete medicale, proiecte de cercetare stiintifica), date despre pacienti, programarile lor, tratamentele administrate, cat si despre tipurile de terapii si medicamente continute de tratamente.

2. Regulile modelului

* 1. Unui pacient ii corespund mai multe programari sau niciuna.
* 2. O programare ii corespunde unui singur pacient.
* 3. O programare ii corespunde macar unui pacient.
* 4. In cadrul unei programari se pot prescrie mai multe tratamente.
* 5. In cadrul unei programari se poate sa nu se prescrie niciun tratament.
* 6. Un tratament poate fi prescris in cadrul mai multor programari.
* 7. Un tratament poate sa nu fie prescris in cadrul niciunei programari.
* 8. Un tratament poate include mai multe terapii sau niciuna.
* 9. O terapie poate fi inclusa in mai multe tratamente sau niciuna.
* 10. Un tratament poate contine mai multe medicamente sau niciunul.
* 11. Un medicament poate fi continut in mai multe tratamente sau niciunul.
* 12. Unui doctor ii revin mai multe programari sau niciuna.
* 13. O programare ii revine unui singur doctor.
* 14. O programare ii revine macar unui doctor.
* 15. Un doctor consulta intr-un singur cabinet medical.
* 16. Un doctor poate sa nu consulte intr-un cabinet medical. Acesta se poate deplasa la domiciliul pacientului.
* 17. Un doctor poate fi atasat la mai multe proiecte de cercetare sau niciunul.
* 18. La un proiect de cercetare pot fi atasati mai multi doctori sau niciunul.

3. Diagrama Entitate-Relatie

3.1 Reprezentarea diagramei



3.2. Descrierea entitatilor, atributelor, cheilor, relatiilor si a cardinalitatilor

3.2.1 Descrierea entitatilor, atributelor si a cheilor

3.2.1.1 Tabelul cabinet\_medical

Tabelul cabinet\_medical stocheaza datele elementare in legatura cu cabinetele medicale din baza de date.

|  |  |  |  |
| --- | --- | --- | --- |
| Cheie | Denumire atribut | Descriere | Tipul atributului |
| PK | id\_cabinet\_medical | Identificator unic al cabinetului medical | decimal(4) |
|  | etaj | Etajul unde se afla cabinetul medical | decimal(1) |
|  | corp\_cladire | Corpul spitalului in care se afla cabinetul | char(1) |

3.2.1.2 Tabelul doctor

Tabelul doctor stocheaza datele despre doctorii care lucreaza in spital.

|  |  |  |  |
| --- | --- | --- | --- |
| Cheie | Denumire atribut | Descriere | Tipul atributului |
| PK | CNP | Identificator unic al unui doctor | decimal(13) |
|  | prenume | Prenumele doctorului | varchar(25) |
|  | nume | Numele doctorului | varchar(25) |
|  | telefon | Data de contact | varchar(20) |
|  | email | Data de contact | varchar(50) |
|  | data\_angajare | Data la care doctorul a fost angajat | date |
|  | salariu | Salariul net al doctorului | decimal(8) |
|  | puncte\_comision | Procent din salariu care se adauga la salariu | decimal(2,2) |
|  | denumire\_specialitate | Domeniul in care doctorul este specializat | varchar(20) |
| FK | id\_cabinet\_medical | Cabinetul in care lucreaza | decimal(4) |

3.2.1.3 Tabelul proiect

Tabelul proiect stocheaza datele despre proiectele de cercetare in care sunt implicati doctorii din spital.

|  |  |  |  |
| --- | --- | --- | --- |
| Cheie | Denumire atribut | Descriere | Tipul atributului |
| PK | id\_proiect | Identificator unic al unui proiect | decimal(2) |
|  | data\_inceput | Data la care a fost pornit proiectul | date |
|  | buget\_alocat | Valoarea totala alocata pentru proiect | decimal(10) |
|  | nume\_proiect | Numele proiectului | varchar(200) |
|  | nume\_laborator\_desfasurare | Numele laboratorului in care are loc proiectul | varchar(14) |

3.2.1.4 Tabelul pacient

Tabelul pacient stocheaza datele despre pacientii care si-au facut o programare sau mai multe.

|  |  |  |  |
| --- | --- | --- | --- |
| Cheie | Denumire atribut | Descriere | Tipul atributului |
| PK | CNP\_pacient | Identificator unic al unui pacient | decimal(13) |
|  | nume\_pacient | Numele pacientului | varchar(25) |
|  | prenume\_pacient | Prenumele pacientului | varchar(25) |
|  | telefon | Data de contact | varchar(20) |
|  | sex | Sexul pacientului | varchar(8), |
|  | data\_nasterii | Data nasterii a pacientului | date |

3.2.1.5 Tabelul programare

Tabelul programare stocheaza datele despre programarile pe care pacientii si le-au facut la doctori.

|  |  |  |  |
| --- | --- | --- | --- |
| Cheie | Denumire atribut | Descriere | Tipul atributului |
| PK | id\_programare | Identificator unic al unei programari | decimal(10) |
| FK | CNP\_pacient | Pacientul care vine la programare | decimal(13) |
| FK | CNP\_doctor | Doctorul la care s-a facut programarea | decimal(13) |
|  | data | Data la care este stabilita programarea | datetime |

3.2.1.6 Tabelul tratament

Tabelul tratament stocheaza datele despre toate tratamentele prescrise de doctori in cadrul programarilor.

|  |  |  |  |
| --- | --- | --- | --- |
| Cheie | Denumire atribut | Descriere | Tipul atributului |
| PK | id\_tratament | Identificator unic al unui tratament | decimal(4) |
|  | nume\_tratament | Numele tratamentului care contine implicit si diagnosticul | varchar(60) |
|  | tip\_tratament | Tipul tratamentului | varchar(11) |

3.2.1.7 Tabelul terapie

Tabelul terapie stocheaza datele despre toate terapiile folosite de doctori in tratarea pacientilor.

|  |  |  |  |
| --- | --- | --- | --- |
| Cheie | Denumire atribut | Descriere | Tipul atributului |
| PK | id\_terapie | Identificator unic al unei terapii | decimal(2) |
|  | nume\_terapie | Numele terapiei | varchar(20) |
|  | durata | Timp alocat terapiei in zile | decimal(3) |

3.2.1.8 Tabelul medicament

Tabelul medicament stocheaza datele despre toate medicamentele folosite de doctori in tratarea pacientilor.

|  |  |  |  |
| --- | --- | --- | --- |
| Cheie | Denumire atribut | Descriere | Tipul atributului |
| PK | id\_medicament | Identificator unic al unei terapii | decimal(5) |
|  | nume\_medicament | Numele medicamentului | varchar(50) |
|  | nume\_producator | Numele producatorului medicamentului | varchar(20) |
|  | forma\_farmaceutica | Modul de administrare al medicamentului | varchar(20) |

3.2.2 Descrierea relatiilor si a cardinalitatilor

3.2.2.1 cabinet\_medical – doctor

Relatia: Un doctor consulta pacienti intr-un cabinet medical.

Cardinalitati:

* Cardinalitate maximala

- In cate cabinete medicale poate lucra un doctor? =>1

-Cati doctori lucreaza intr-un cabinet medical? =>multi

* Cardinalitate minimala

-In cate cabinete medicale trebuie sa lucreze un doctor? =>0

-Cati doctori pot lucra intr-un cabinet medical? =>0

3.2.2.2 doctor – proiect

Relatia: Un doctor este atasat la proiecte.

* Cardinalitate maximala

-La cate proiecte poate fi atasat un doctor? =>multe

-Cati doctori pot fi atasati la un proiect? => multi

* Cardinalitate minima

-La cate proiecte trebuie sa fie atasat un doctor? =>0

-Cati doctori trebuie sa fie atasati la proiect? =>0

3.2.2.3 doctor-programare

Relatia: O programare ii revine unui doctor.

* Cardinalitate maximala

-Cate programari ii revin unui doctor? =>multe

-La cati doctori revine o programare? =>1

* Cardinalitate minima

-Cate programari ii pot reveni unui doctor? =>0

-La cati doctori trebuie sa revina o programare? =>1

3.2.2.3 pacient-programare

Relatia: Un pacient solicita o programare.

* Cardinalitate maximala

-Cate programari poate solicita un pacient? =>multe

-De cati pacienti poate fi solicitata o programare? =>1

* Cardinalitate minimala

-Cate programari trebuie sa solicite un pacient? =>0

-De cati pacienti trebuie sa fie solicitata o programare? =>1

3.2.2.4 programare-tratament

Relatia: In urma unei programari se prescrie un tratament.

* Cardinalitate maximala

-Cate tratamente se pot prescrie in urma unei programari? => multe

-In urma cator programari se poate prescrie un tratament? => multe

* Cardinalitate minimala

-Cate tratamente trebuie sa fie prescrise in urma unei programari? =>0

- In urma cator programari trebuie prescris un tratament? =>0

3.2.2.5 tratament-terapie

Relatia: Un tratament include terapii.

* Cardinalitate maximala

-In cate tratamente poate fi inclusa o terapie? =>multe

-Un tratament cate terapii poate sa includa? =>multe

* Cardinalitate minima

- In cate tratamente trebuie sa fie inclusa o terapie? =>0

-Un tratament cate terapii trebuie sa includa? =>0

3.2.2.6 tratament-medicament

Relatia: Un tratament contine medicamente.

* Cardinalitate maximala

-Cate tratamente pot contine un medicament? =>multe

-Un tratament cate medicamente poate sa contina? =>multe

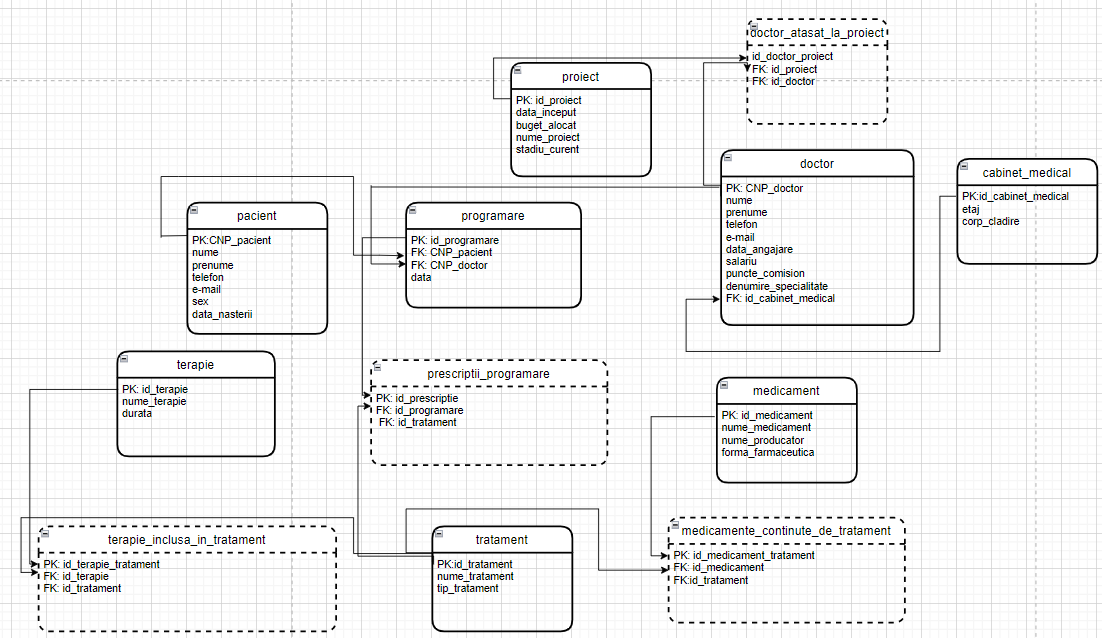
* Cardinalitate minima

- Cate tratamente trebuie sa contina un medicament? =>0

-Un tratament cate terapii trebuie sa contina? =>0

4. Diagrama Conceptuala

4.1 Reprezentare diagrama



4.2 Descrierea constrangerilor de integritate

4.2.1 Tabelul cabinet\_medical

* Constrangeri de tip primary key

-id\_cabinet\_medical\_pk, pentru atributul id\_cabinet\_medical

* Constrangeri de tip not null

-etaj\_not\_null, pentru atributul etaj

-corp\_cladire\_not\_null, pentru atributul corp\_cladire

=> Cabinetul medical trebuie sa fie obligatoriu localizat.

4.2.2 Tabelul doctor

* Constrangeri de tipul primary key

-CNP\_doctor\_pk, pentru atributul CNP

* Constrangeri de tipul not null

-prenume\_doctor\_not\_null, pentru atributul prenume

-nume\_doctor\_not\_null, pentru atributul nume

=> Este nevoie ca numele si prenumele doctorului sa fie publice.

-data\_angajare\_not\_null, pentru atributul data\_angajare

=> Este important ca data angajarii sa fie cunoscuta deoarece aceasta poate determina diferite sporuri salariale anuale.

-denumire\_specialitate\_not\_null, pentru atributul denumire\_specialitate

=> Pacientul trebuie sa stie specialitatea medicului pentru a se putea programa conform nevoilor sale.

-telefon\_doctor\_not\_null, pentru atributul telefon

-email\_doctor\_not\_null, pentru atributul email

=> Telefonul si email sunt date importante prin care doctoral poate fi contactat.

* Constrangeri de tipul unique

-uq\_telefon\_doctor, pentru atributul telefon

-uq\_email\_doctor, pentru atributul email

=> Doi doctori nu pot avea aceleasi date de contact.

* Constrangeri de tipul foreign key

-id\_cabinet\_medical\_fk, pentru atributul id\_cabinet\_medical

=> Face legatura cu tabelul cabinet\_medical(id\_cabinet\_medical) si determina cabinetul medical in care lucreaza un doctor (daca acesta lucreaza intr-unul).

4.2.3 Tabelul proiect

* Constrangeri de tipul primary key

-id\_proiect\_pk, pentru atributul id\_proiect

* Constrangeri de tipul not null

-data\_inceput\_not\_null, pentru atributul data\_inceput

=> Este important de stiut data la care a pornit proiectul.

-nume\_proiect\_not\_null, pentru atributul nume\_proiect

=> Este esentialala aceasta constrangere deoarece prin aceasta se cunoaste tematica si scopul final al proiectului.

-buget\_alocat\_not\_null, pentru atributul buget\_alocat

=> Este important sa se stie bugetul alocat deoarece in functie de acesta se pot lua diferite decizii in legatura cu achizitionarea de material necesare, de exemplu.

* Constrangeri de tipul unique

-uq\_nume\_proiect, pentru atributul nume\_proiect

=> Nu putem avea 2 proiecte cu acelasi nume, si implicit cu aceeasi tematica.

4.2.4 Tabelul doctor\_atasat\_la\_proiect

* Constrangeri de tipul primary key

-id\_doctor\_proiect\_pk, pentru atributul id\_doctor\_proiect

Constrangeri de tipul foreign key

- CNP\_doctor\_proiect\_fk, pentru atributul CNP\_doctor\_proiect

=> Face legatura cu tabelul doctor (CNP) si determina doctoral care este implicat in proiect.

- id\_proiect\_fk, pentru atributul id\_proiect

=> Face legatura cu tabelul proiect(id\_proiect) si determina proiectul la care este atasat doctoral in cauza.

4.2.5 Tabelul pacient

* Constrangeri de tipul primary key

-CNP\_pacient\_pk, pentru atributul CNP

* Constrangeri de tipul not null

- prenume\_pacient\_not\_null, pentru atributul prenume

- nume\_pacient\_not\_null, pentru atributul nume

=> Este nevoie ca numele si prenumele pacientului sa fie cunoscute.

- telefon\_pacient\_not\_null, pentru atributul telefon

=> Este nevoie ca telefonul sa fie cunoscute pentru a putea contacta pacientul.

* Constrangeri de tipul unique

- uq\_telefon\_pacient, pentru atributul telefon

=> Nu pot exista 2 pacienti cu acelasi numar de telefon.

4.2.6 Tabelul programare

* Constrangeri de tipul primary key

-id\_programare\_pk, pentru atributul id\_programare

* Constrangeri de tipul not null

- CNP\_pacient\_programare\_not\_null, pentru atributul CNP\_pacient\_programare

- CNP\_doctor\_programare\_not\_null, pentru atributul

CNP\_doctor\_programare

=> Nu putem avea o programare de la care sa lipseasca pacientul sau doctoral.

- data\_not\_null, pentru atributul data

=> Este obligatoriu ca data programarii sa fie cunoscuta.

* Constrageri de tipul foreign key

- CNP\_pacient\_programare\_fk, pentru atributul CNP\_pacient\_programare

=> Face legatura cu tabelul pacient (CNP) si determina pacientul care participa la programare.

- CNP\_doctor\_programare\_fk, pentru atributul CNP\_doctor\_programare

=> Face legatura cu tabelul doctor (CNP) si determina doctorul care participa la programare.

4.2.7 Tabelul tratament

* Constrangeri de tipul primary key

-id\_tratament\_pk, pentru atributul id\_tratament

* Constrangeri de tipul not null

- nume\_tratament\_not\_null, pentru atributul nume\_tratament

=> Este important ca numele tratamentului sa fie cunoscut deoarece prin intermediul acestuia se identifica si diagnosticul pacientului.

- tip\_tratament\_not\_null, pentru atributul tip\_tratament

=> Informeaza in legatura cu stadiul bolii diagnosticate si cu scopul tratamentului.

4.2.8 Tabelul prescriptii\_programare

* Constrangeri de tipul primary key

-id\_prescripitie\_pk, pentru atributul id\_prescriptie

* Constrangeri de tipul foreign key

-id\_tratament\_fk, pentru atributul id\_tratament

=> Face legatura cu tabelul tratament (id\_tratament) si determina tratamentul care se prescrie dupa programare.

-id\_programare\_fk, pentru atributul id\_programare

=> Face legatura cu tabelul programare (id\_programare) si determina programarea in cadrul careia se prescrie tratamentul respectiv.

4.2.9 Tabelul terapie

* Constrangeri de tipul primary key

-id\_terapie\_pk, pentru atributul id\_terapie

* Constrangeri e tipul not null

- nume\_terapie\_not\_null, pentru atributul nume\_terapie

=> Este obligatoriu sa se cunosca numele pentru a stii cu exactitate tipul de terapie.

4.2.10 Tabelul medicament

* Constrangeri de tipul primary key

-id\_medicament\_pk, pentru atributul id\_medicament

* Constrangeri de tipul not null

- nume\_medicament\_not\_null, pentru atributul nume\_medicament

- nume\_producator\_not\_null, pentru atributul nume\_producator

=> Ambele constrangeri sunt necesare deoarece orice medicament inregistrat are un nume si un producator.

4.2.11 Tabelul terapie\_inclusa\_in\_tratament

* Constrangeri de tipul primary key

-id\_terapie\_tratament\_pk, pentru atributul id\_terapie\_tratament

* Constrangeri de tipul foreign key

-id\_terapie\_fk, pentru atributul id\_terapie

=> Face legatura cu tabelul terapie (id\_terapie) si determina terapia care este inclusa in tratamentul respective.

- id\_tratament\_terapie\_fk, pentru atributul id\_tratament

=> Face legatura cu tabelul tratament (id\_tratament) si determina tratamentul care include acel tip de terapie.

4.2.12 Tabelul medicamente\_continute\_de\_tratament

* Constrangeri de tipul primary key

- id\_medicament\_tratament\_pk, pentru atributul id\_medicament\_tratament

* Constrangeri de tipul foreign key

-id\_medicament\_fk, pentru atributul id\_medicament

=> Face legatura cu tabelul medicament (id\_medicament) si determina medicamentul care este continut de tratamentul respective.

- id\_tratament\_medicament\_fk, pentru atributul id\_tratament

=> Face legatura cu tabelul tratament (id\_tratament) si determina tratamentul care include acel tip de medicament.

4.3 Scheme relationale

4.3.1 Schemele relationale

Schemele relationale atasate diagramei conceptuale sunt:

* cabinet\_medical (id\_cabinet\_medical#, etaj, corp\_cladire);
* doctor (CNP#, prenume, nume, telefon, email, data\_angajare, salariu, puncte\_comision, denumire\_specialitate, id\_cabinet\_medical(FK));
* proiect (id\_proiect#, data\_inceput, buget\_alocat, nume\_proiect, nume\_laborator\_desfasurare, stadiu\_curent);
* doctor\_atasat\_la\_proiect (id\_doctor\_proiect#, id\_doctor(FK), id\_proiect(FK));
* pacient (CNP#, nume, prenume, telefon, sex, data\_nasterii);
* programare (id\_programare#, CNP\_pacient\_programare (FK), CNP\_doctor\_programare (FK), datetime);
* tratament (id\_tratament#, nume\_tratament, tip\_tratament);
* prescriptii\_programare (id\_prescriptie#, id\_tratament(FK), id\_programare(FK));
* terapie (id\_terapie#, nume\_terapie, durata\_in\_zile);
* medicament (id\_medicament#, nume\_medicament, nume\_producator, forma\_farmaceutica);
* terapie\_inclusa\_in\_tratament (id\_terapie\_tratament, id\_terapie(FK), id\_tratament\_terapie(FK));
* medicamente\_continute\_de\_tratament (id\_medicament\_tratament, id\_medicament(FK), id\_tratament(FK));

4.3.2 Descrierea constrangerilor on delete

Exista 11 constrangeri on delete care corespund in parte cate unei constrangeri de tip foreign key.

* id\_cabinet\_medical\_fk foreign key, on delete set null

-atunci cand un cabinet medical este sters, toti doctorii care lucrau in acel cabinet vor ramane in spital, dar fara un cabinet medical specificat;

-on delete cascade nu ar fi fost o optiune buna, deoarece s-ar fi sters in acelasi timp cu cabinetul medical si doctorii care lucrau in el;

* CNP\_doctor\_proiect\_fk, on delete cascade

-atunci cand se sterge un doctor practic si valoarea inregistrarii se pierde;

* id\_proiect\_fk, on delete cascade
* -atunci cand se sterge un proiect practic si valoarea inregistrarii se pierde;
* CNP\_pacient\_programare\_fk, on delete cascade

-atunci cand se sterge un pacient practic si valoarea inregistrarii se pierde partial (ne intereseaza sa stim programarile cu exactitate pentru fiecare pacient);

-daca dorim sa stergem un pacient din baza de date cel mai probabil nu ne mai intereseaza datele in legatura cu acesta;

* CNP\_doctor\_programare\_fk, on delete cascade

-atunci cand se sterge un doctor practic si valoarea inregistrarii se pierde partial (ne intereseaza sa stim programarile cu exactitate pentru fiecare doctor);

- daca dorim sa stergem un doctor din baza de date cel mai probabil nu ne mai intereseaza datele in legatura cu acesta (nu mai lucreaza in acel spital, de exemplu);

* id\_tratament\_fk, on delete cascade

-atunci cand se sterge un tratament practic si valoarea inregistrarii se pierde;

* id\_programare

-atunci cand se sterge o programare practic si valoarea inregistrarii se pierde;

* id\_terapie\_fk, on delete cascade

-atunci cand se sterge o terapie practic si valoarea inregistrarii se pierde;

* id\_tratament\_terapie\_fk, on delete cascade

-atunci cand se sterge un tratament practic si valoarea inregistrarii se pierde;

* id\_medicament\_fk, on delete cascade

-atunci cand se sterge un medicament practic si valoarea inregistrarii se pierde;

* id\_tratament\_medicament\_fk, on delete cascade

-atunci cand se sterge un tratament practic si valoarea inregistrarii se pierde;

5. Scriptul SQL

5.1 Introducere

Scriptul SQL, salvat cu numele script\_sql\_andrei\_cojocaru\_261.sql, a fost scris folosind MySQL Workbench, iar pentru baza de date locala, Mysql Server.

Pentru prima rulare a scriptului, trebuie ignorate primele instructiuni de drop pentru tabele, deoarece acestea nu au fost inca create. De la a doua rulare, intreg scriptul poate fi rulat fara probleme.

5.2 Etapa de drop

Atunci cand am efectuat instructiunile drop, am avut grija ca mai intai sa sterg tabelele care contin chei externe in ordinea potrivita.

drop table doctor\_atasat\_la\_proiect;

drop table prescriptii\_programare;

drop table terapie\_inclusa\_in\_tratament;

drop table medicamente\_continute\_de\_tratament;

drop table proiect;

drop table terapie;

drop table medicament;

drop table programare;

drop table tratament;

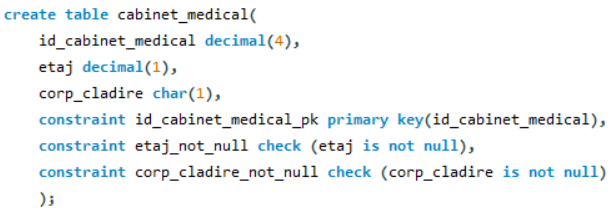
drop table pacient;

drop table doctor;

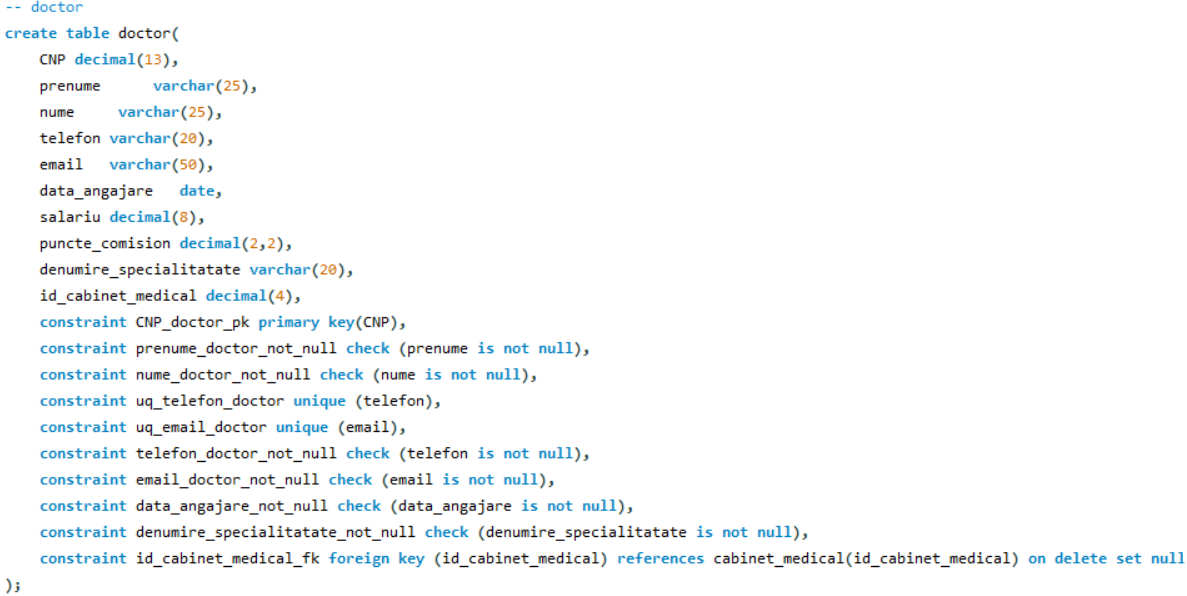
drop table cabinet\_medical;

5.3 Crearea tabelelor, inclusive a constragerilor

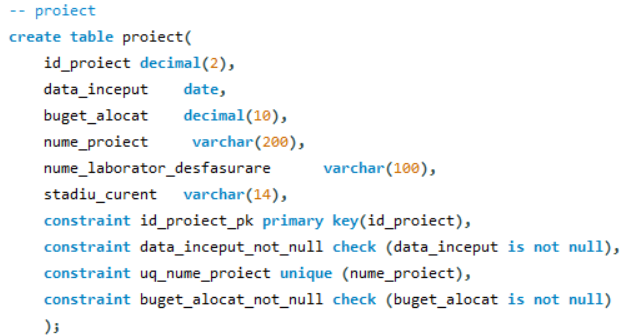
5.3.1 Crearea tabelului cabinet\_medical



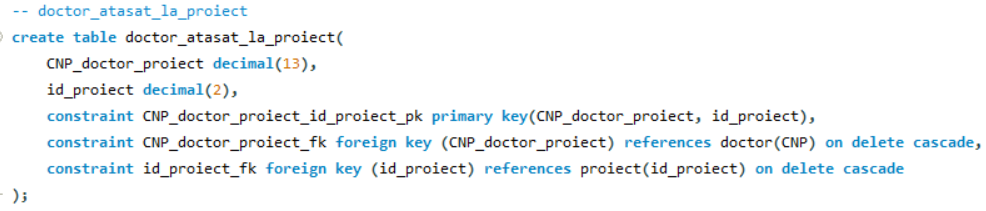
5.3.1 Crearea tabelului doctor



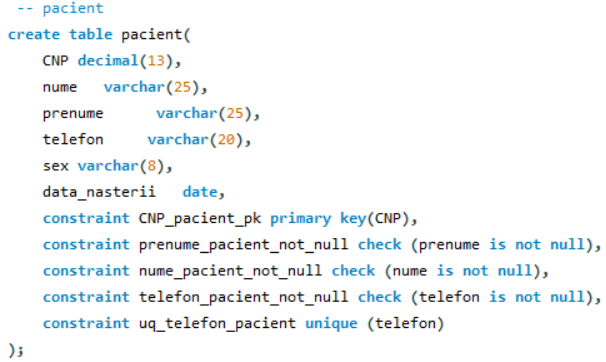
5.3.2 Crearea tabelului proiect



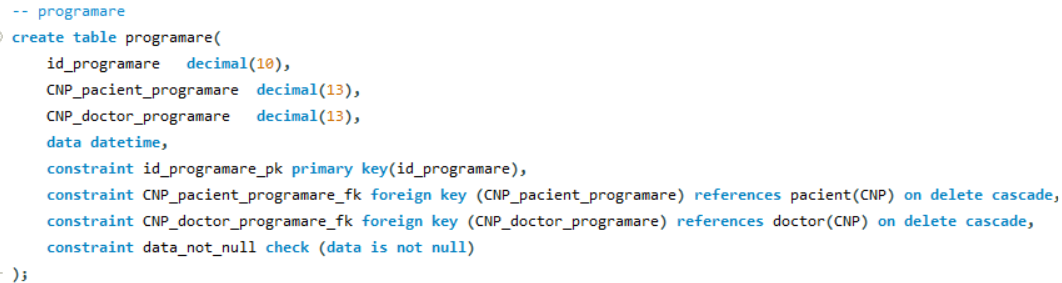
5.3.3 Crearea tabelului doctor\_atasat\_la\_proiect



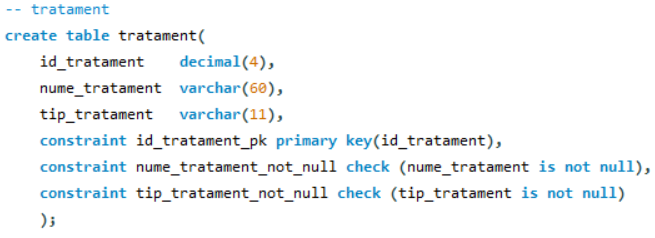
5.3.4 Crearea tabelului pacient



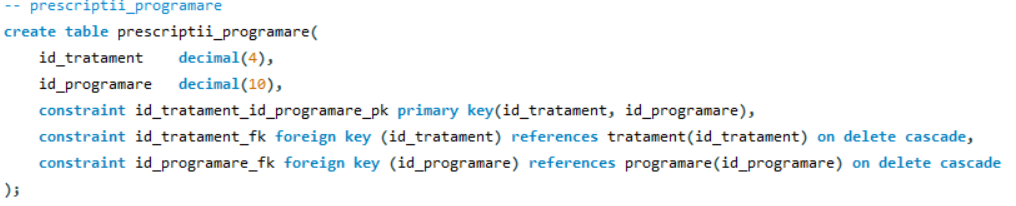
5.3.5 Crearea tabelului programare



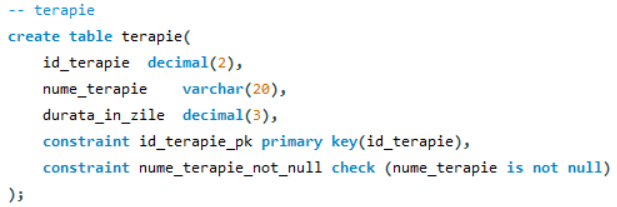
5.3.6 Crearea tabelului tratament



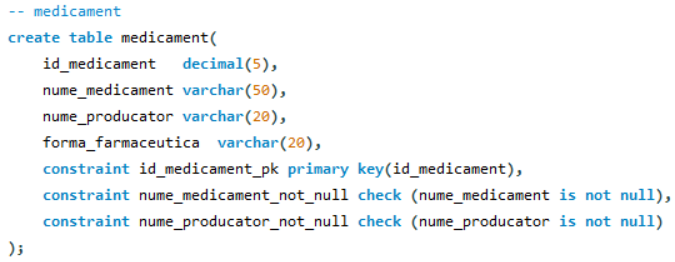
5.3.7 Crearea tabelului prescriptii\_programare



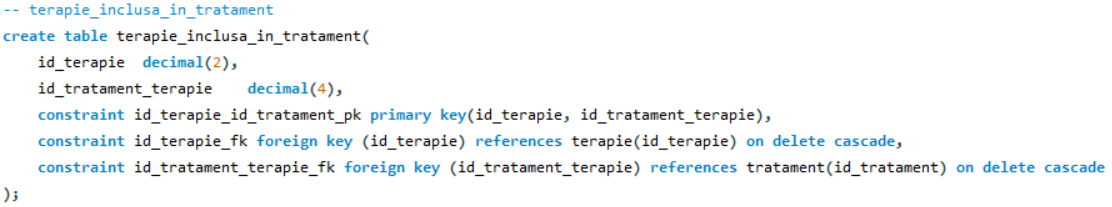
5.3.8 Crearea tabelului terapie



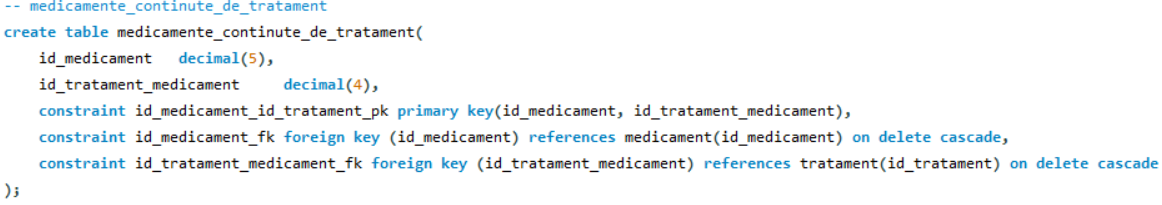
5.3.9 Crearea tabelului medicament



5.3.10 Crearea tabelului terapie\_inclusa\_in\_tratament



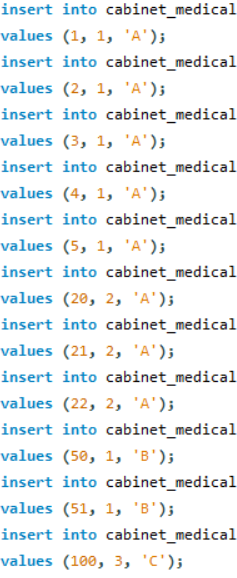
5.3.11 Crearea tabelului -- medicamente\_continute\_de\_tratament



5.4 Introducerea datelor in baza de date

In realitate, in unele tabele se pot regasi un numar extrem de mare de intrari. In continuare, doar voi exemplifica cu cateva inserari in fiecare table pentru a demonstra functionalitatea.

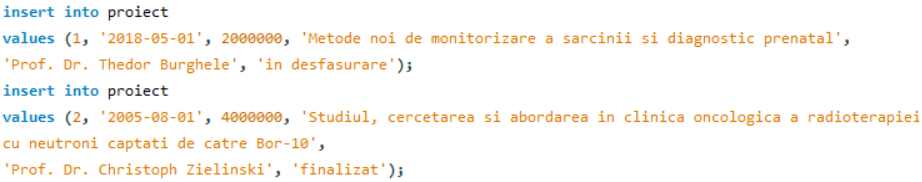
5.4.1 Introducerea datelor in tabelul cabinet\_medical



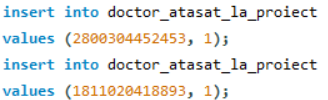
5.4.2 Introducerea datelor in tabelul doctor



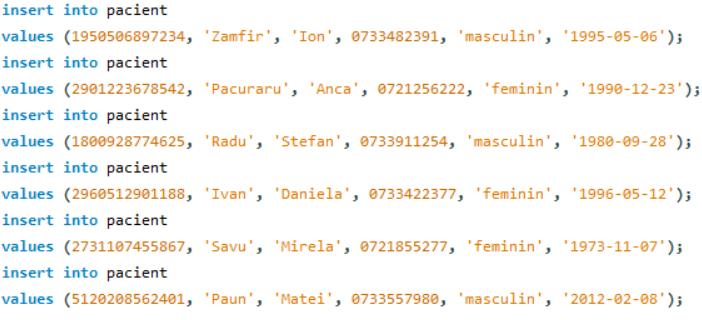
5.4.2 Introducerea datelor in tabelul proiect



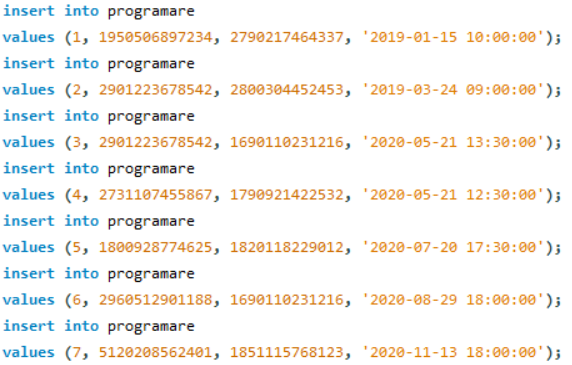
5.4.3 Introducerea datelor in tabelul doctor\_atasat\_la\_proiect



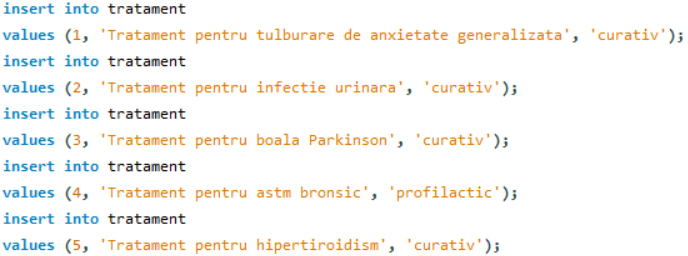
5.4.4 Introducerea datelor in tabelul pacient



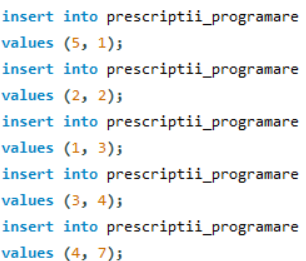
5.4.5 Introducerea datelor in tabelul programare



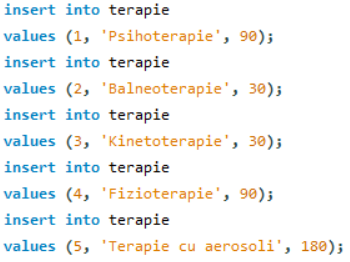
5.4.6 Introducerea datelor in tabelul tratament



5.4.7 Introducerea datelor in tabelul prescriptii\_programare



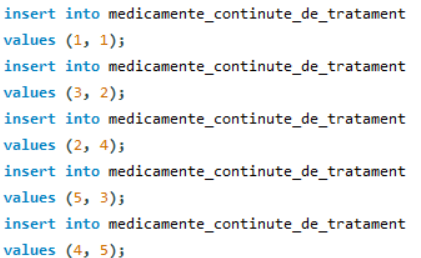
5.4.8 Introducerea datelor in tabelul terapie



5.4.9 Introducerea datelor in tabelul medicament



5.4.10 Introducerea datelor in tabelul medicamente\_continute\_de\_tratament



5.4.11 Introducerea datelor in tabelul terapie\_inclusa\_in\_tratament

