*Nume: Logofatu Georgiana-Lavinia Grupa: C.R. 3.2.A*

***CABINET MEDICAL***

Tema acestui proiect este gestiunea unei baze de date pentru un cabinet medical.

In acest cabinet sunt angajati 15 de medici si exista 10 sectii:

1. Cardiologie
2. Neurologie
3. Reumatologie
4. Endocrinologie
5. Pediatrie
6. Oncologie
7. Oftalmologie
8. Diabet
9. ORL
10. Gastroenterologie

Pe fiecare sectie sunt specializati mai multi medici, astfel:

Sectia 1 – 3 medici

Sectia 2 – 1 medic

Sectia 3 – 1 medic

Sectia 4 – 2 medici

Sectia 5 – 2 medici

Sectia 6 – 1 medic

Sectia 7 – 1 medic

Sectia 8 – 1 medic

Sectia 9 – 2 medici

Sectia 10 – 1 medic

De asemenea, in fiecare sectie exista un asistent care se ocupa cu programarile pentru consultatii. Fiecare consultatie este facuta de catre un asistent, pe baza unei programari. Apoi, in urma consultatiei, medicul poate elibera o reteta.

Baza de date contine 7 tabele:

1. ***Sectii*** – contine numarul, denumirea si descrierea sectiei
2. ***Medici*** – contine ID-ul fiecarui medic, sectia si alte date personale (CNP, nume, prenume, salariu, email, telefon)
3. ***Pacienti*** – contine ID-ul fiecarui pacient si alte date personale (CNP, nume, prenume, data nasterii, localitate si asigurare)

*Daca un pacient are asigurare, pretul unei consultatii se injumatateste!*

1. ***Asistenti*** – contine ID-ul fiecarui asistent, sectia si alte date personale (CNP, nume, prenume)
2. ***Programare*** – contine numarul programarii si alte date despre aceasta (asistentul, pacientul, medicul si data)
3. ***Consultatie*** – contine numarul consultatiei, pretul, numarul progamarii, medicul si pacientul
4. ***Reteta*** – contine numarul retetei, numarul consultatiei in baza careia este eliberata, pacient, medic si medicamentul recomandat.

Crearea tabelelor:

Pentru a specifica direct tipul de data CNP, am creat domeniul CNP\_tip:

create domain CNP\_tip as char(13);

create table Asistenti

(

ID int,

CNP CNP\_tip NOT NULL UNIQUE,

Nume varchar(50),

Prenume varchar(50),

Sectie int,

primary key(ID)

)

create table Pacienti

(

ID int,

CNP CNP\_tip NOT NULL UNIQUE,

Nume varchar(50),

Prenume varchar(50),

Data\_nasterii date,

Localitate varchar(50),

Asigurare bool,

primary key(ID)

)

create table Medici

(

ID int,

CNP CNP\_tip NOT NULL UNIQUE,

Nume varchar(50),

Prenume varchar(50),

Sectie int,

Salariu int,

email varchar(50),

Telefon varchar(15),

primary key(ID)

)

create table Sectii

(

Numar int,

Denumire varchar(50),

Descriere varchar(200),

primary key(Numar)

)

create table Programare

(

Numar int,

ID\_Asistent int,

ID\_Pacient int,

ID\_Medic int,

Data date,

primary key(Numar)

)

create table Consultatie

(

Numar int,

ID\_Pacient int,

ID\_Medic int,

Nr\_Programare int,

Pret int,

primary key(Numar)

)

create table Reteta

(

Numar int,

Numar\_Consultatie int,

ID\_Pacient int,

ID\_Medic int,

Medicament varchar(50)

)

**Constraints:**

***Tabelul Programare:***

alter table programare add constraint fk\_id\_asistent

foreign key (ID\_Asistent) references Asistenti(ID)

on delete cascade on update cascade

alter table programare add constraint fk\_id\_pacient

foreign key (ID\_Pacient) references Pacienti(ID)

on delete cascade on update cascade

alter table programare add constraint fk\_id\_medic

foreign key (ID\_Medic) references Medici(ID)

on delete cascade on update cascade

***Tabelul Asistenți:***

alter table Asistenti add constraint fk\_sectie

foreign key (Sectie) references Sectii(Numar)

***Tabelul Consultație:***

alter table consultatie add constraint fk\_id\_pacient

foreign key (ID\_Pacient) references Pacienti(ID)

on delete cascade on update cascade

alter table consultatie add constraint fk\_id\_medic

foreign key (ID\_Medic) references Medici(ID)

on delete cascade on update cascade

alter table consultatie add constraint fk\_nr\_programare

foreign key (Nr\_Programare) references Programare(Numar)

on delete cascade on update cascade

***Tabelul Medici:***

alter table Medici add constraint fk\_sectie

foreign key (Sectie) references Sectii(Numar)

***Tabelul Rețeta:***

alter table reteta add constraint fk\_id\_pacient

foreign key (ID\_Pacient) references Pacienti(ID)

on delete cascade on update cascade

alter table reteta add constraint fk\_id\_medic

foreign key (ID\_Medic) references Medici(ID)

on delete cascade on update cascade

alter table reteta add constraint fk\_nr\_consultatie

foreign key (Numar\_Consultatie) references Consultatie(Numar)

on delete cascade on update cascade

**UPDATE:**

1. Sa se majorize cu 1000 salariul medicilor care au salariul cuprins intre 4000 si

5000 lei.

Table

Description automatically generated

*update medici*

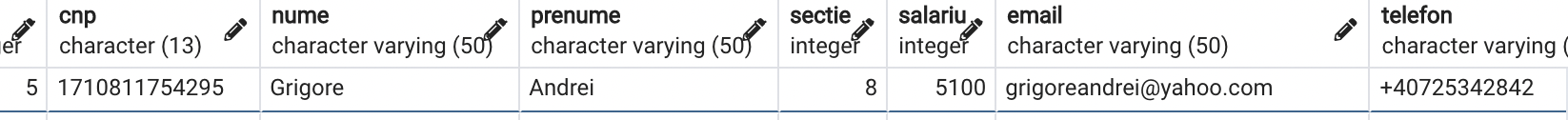
*set salariu = salariu + 1000*

*where salariu between 4000 and 5000*

Table

Description automatically generated

1. Sa se actualizeze numarul de telefon si emailul pentru medicul cu numele “Grigore”



*update medici*

*set telefon = '+40784162637', email = 'grigoreandrei@gmail.com'*

*where nume ='Grigore'*



1. Sa se reduca la jumatate pretul consultatiilor pentru pacientii care au asigurare

*update consultatie*

*set pret = pret / 2*

*from pacienti*

*where consultatie.id\_pacient = pacienti.id and pacienti.asigurare = 'true'*

Graphical user interface, text, application, email

Description automatically generated

Din 30 de pacienti, 15 au asigurare

1. Pe retetele une exista medicamentul “Neurovet”, sa se schimbe in “Neuromax”

Table

Description automatically generated

*update reteta*

*set medicament = 'Neuromax'*

*where medicament = 'Neurovet'*

Table

Description automatically generated

**DELETE:**

1. Sa se stearga pacientii care sunt nascuti intre 2003 si 2005.

*delete from pacienti*

*where extract(year from data\_nasterii) between 2003 and 2005*

1. Sa se stearga programarea pacientului cu numarul 2.

*delete from programare*

*where id\_pacient = 2*

1. Sa se stearga medicul cu numele Manea Lucian.

*delete from medici*

*where nume = 'Manea' and prenume = 'Lucian'*

1. Sa se stearga asistenul care lucreaza la sectia 1.

*delete from asistenti*

*where sectie = 1*

**Normalizari:**

FN1:

Este respectata forma normala deoarece fiecarui atribut ii corespunde o valoare atomica.

Contraexemplu: Daca in tabelul Medici, atributul telefon continea o multime de valori, ca de exemplu {+49765451264, +40251428494 }, atunci trebuia sa formam entitatea Telefon in care aveam telefonul personal si cel de la serviciu.

FN2:

O relatie X este in a doua forma normala daca si numai daca:

- Relatia X este in FN1

- fiecare atribut care nu este cheie(nu face parte din cheia primara) este dependent de intreaga cheie primara.

Exemplu: toate tabelele sunt dependente de cheia primara, tabelul medicamente(pk-id\_produs), denumiremedicament, pretmedicament sunt dependente de cheia primara samd.

contraexemplu: daca nu am avea tabelul Sectii, iar in tabelul Medici am avea coloanele din tabelul Sectii, coloanele din tabelul Sectii nu ar depinde de cheia primara din tabelul Medici.

FN3:

O relatie X este in a3a forma normala daca si numai daca:

-Relatia X este in FN2

-fiecare atribut care nu este cheie(nu participa la o cheie) depinde direct de cheia primara

-este respectata de toate tabelele, deoarece fiecare atribut(coloana) care nu este cheie, depinde de cheie, de intreaga cheie si numai de cheie.

contraexemplu: daca in tabelul Medici, am avea coloanele din tabelul Sectii(numar, denumire, descriere), denumire, descriere nu ar depinde de cheia primara a tabelului Medici, ea ar depinde de numar sectie, care la randul ei va depinde de cheia primara din Medici=> dependenta tranzitiva.

Deci coloanele denumire si descriere vor fi mutate in alta tabela(sectii)=>eliminarea dependentei tranzitive

**Interogari:**

**INNER JOIN:**

1. Sa se afiseze numele pacientilor care au avut consultatie la endocrinologie

*select pacienti.nume, pacienti.prenume, sectii.denumire*

*from pacienti inner join consultatie*

*on pacienti.id = consultatie.id\_pacient*

*inner join medici*

*on medici.id = consultatie.id\_medic*

*inner join sectii*

*on medici.sectie = sectii.numar*

*where sectii.denumire = 'Endocrinologie'*

Table

Description automatically generated

1. Sa se afiseze numele doctorilor impreuna cu sectia la care lucreaza

*select medici.nume, medici.prenume, sectii.denumire*

*from medici inner join sectii*

*on medici.sectie = sectii.numar*

*Table

Description automatically generated*

1. Numele pacientilor care au platit cea mai scumpa consultatie

*select pacienti.nume, pacienti.prenume, consultatie.pret*

*from pacienti inner join consultatie on pacienti.id = consultatie.id\_pacient*

*where consultatie.pret= (select max(pret) from consultatie)*

*Graphical user interface, application, table

Description automatically generated*

1. Sa se afiseze numele si prenumele pacientilor si doctorilor corespunzatoare fiecarei consultatii

*select consultatie.numar, pacienti.nume as nume\_pacient, pacienti.prenume as prenume\_pacient,*

*medici.nume as nume\_medic, medici.prenume as prenume\_medic*

*from pacienti inner join consultatie on consultatie.id\_pacient = pacienti.id*

*inner join medici on consultatie.id\_medic = medici.id*

Graphical user interface

Description automatically generated

**OUTER JOIN:**

1. Sa se afiseze toti medicii impreuna cu retetele prescrise(chiar daca nu au prescris nicio reteta)

*select medici.nume, medici.prenume, reteta.numar, reteta.medicament*

*from medici left outer join reteta*

*on medici.id = reteta.id\_medic*

*Table

Description automatically generated*

1. Sa se afiseze toti medicii, precum si cei care au acelasi nume de familie cu al unui pacient.

*select medici.nume as nume\_medici, medici.prenume as prenume\_medici,*

*pacienti.nume as nume\_pacienti, pacienti.prenume as prenume\_pacienti*

*from pacienti right outer join medici*

*on medici.nume=pacienti.nume*

*Table

Description automatically generated*

**GROUP BY:**

1. Sa se afiseze numarul total de medici din fiecare sectie

*select sectie, count(\*)total\_medici*

*from medici*

*group by sectie*

*order by sectie*

Table

Description automatically generated

1. Sa se afiseze suma salariilor medicilor pentru fiecare sectie

*select sectie, sum(salariu)total\_salarii*

*from medici*

*group by sectie*

*order by sectie*

Table

Description automatically generated

1. Sa se afiseze media salariilor medicilor pentru fiecare sectie

*select sectii.denumire, round(avg(salariu),0)medie\_salariu*

*from sectii, medici*

*where medici.sectie = sectii.numar*

*group by sectii.denumire*

Table

Description automatically generated

1. Sa se afiseze cate retete a prescris fiecare medic.

*select medici.nume, count(\*)total\_retete*

*from medici, reteta*

*where reteta.id\_medic = medici.id*

*group by medici.nume*

Table

Description automatically generated

1. Sa se afiseze salariul minim al medicilor din fiecare sectie, ordonat descrescator

*select min(salariu), sectii.denumire*

*from medici, sectii*

*where medici.sectie = sectii.numar*

*group by sectii.denumire*

*order by min(salariu) desc*

*Table

Description automatically generated*

**Subinterogari:**

1. Sa se afiseze medicul cu cel mai mare salariu

*select nume, prenume, salariu*

*from medici*

*where salariu = (select max(salariu) from medici)*

*Graphical user interface, application, table

Description automatically generated*

1. Sa se afiseze consultatiile care au pretul mai mare decat media preturilor consultatiilor

*select numar, pret*

*from consultatie*

*where pret >=(select AVG(pret)from consultatie)*

*order by pret*

Table

Description automatically generated

1. Sa se afiseze retetele prescrise de medicul Badescu Lavinia, daca exista

*select reteta.numar, reteta.medicament*

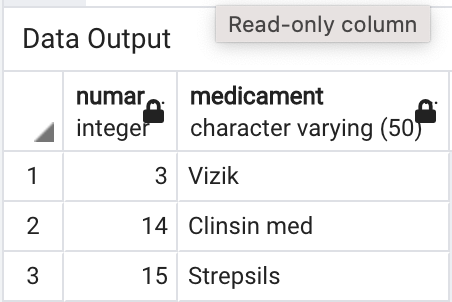
*from reteta*

*where exists*

*(select medici.nume, medici.prenume*

*from medici*

*where reteta.id\_medic = medici.id and medici.nume = 'Badescu' and medici.prenume = 'Lavinia')*



1. Sa se afiseze numele si cnp-urile medicilor care lucreaza in aceeasi sectie ca si medicul cu cnp = ‘2910817017193’

*select medici.nume, medici.prenume, medici.cnp, sectii.denumire*

*from medici, sectii*

*where sectie in(select sectie*

*from medici*

*where cnp = '2910817017193')*

*and medici.sectie = sectii.numar*

Table

Description automatically generated

1. Sa se afiseze pacientii programati in data de 20.10.2021

*select nume, prenume*

*from pacienti*

*where exists*

*(select programare.data*

*from programare where programare.id\_pacient = pacienti.id and programare.data = '2021-10-20')*

Graphical user interface, application, table

Description automatically generated

1. Sa se afiseze ce medici au efectuat o consultatie in perioada 01.10 – 15.10 2021

*select consultatie.numar, medici.nume, medici.prenume, programare.data*

*from medici, consultatie, programare*

*where*

*programare.numar in (select programare.numar*

*from programare*

*where programare.data between '2021-10-01' and '2021-10-15')*

*and programare.numar = consultatie.numar*

*and consultatie.id\_medic = medici.id*

*Table

Description automatically generated*

1. Sa se afiseze sectile care au un angajat cu salariul mai mare de 6000 lei

*select sectii.denumire*

*from sectii*

*where exists(select 1 from medici where salariu > 6000*

*and medici.sectie = sectii.numar)*

*Table

Description automatically generated*

1. Sa se afiseze numele medicilor care au salariul mai mare decat toti medicii din sectia 1

*select nume, prenume, salariu*

*from medici*

*where salariu > all(select salariu*

*from medici*

*where sectie = 1)*

*Table

Description automatically generated*

1. Sa se afiseze asistentii care au facut o programare

*select asistenti.nume, asistenti.prenume, programare.numar*

*from asistenti, programare*

*where exists(select programare.numar*

*where programare.id\_asistent = asistenti.id)*

*Table

Description automatically generated*

**Vederi:**

1. Sa se realizeze un tabel virtual cu toti medicii care au consultat pacienti de gen feminin.

*create view v\_medici\_pacienti*

*as select medici.nume as nume\_medici, medici.prenume as prenume\_medici,*

*pacienti.nume as nume\_pacietni, pacienti.prenume as prenume\_pacienti*

*from medici, pacienti, consultatie*

*where medici.id = consultatie.id\_medic*

*and pacienti.id = consultatie.id\_pacient*

*and substring(pacienti.CNP,1,1) IN ( '2', '6')*

1. Sa se realizeze un table virtual cu toti pacientii din Craiova care au primit o reteta.

*create view v\_pacienti\_retete*

*as select pacienti.nume, pacienti.prenume, pacienti.localitate, reteta.medicament*

*from pacienti, reteta*

*where pacienti.id = reteta.id\_pacient*

*and pacienti.localitate = 'Craiova'*

1. Sa se realizeze un table virtual cu toti asistentii care lucreaza in sectii, ordonat alfabetic in functie de numele fiecarei sectii.

*create view v\_asistenti\_sectii*

*as select sectii.denumire, asistenti.nume, asistenti.prenume*

*from asistenti, sectii*

*where asistenti.sectie = sectii.numar*

*order by sectii.denumire*