

Prisacaru Elena-Catalina

Simionesei Loredana-Marinela

Grupa A5, An 2

C1

Spitalul “Ultima Speranță” este un spital județean care primește oamenii cu diferite probleme de sănătate și apoi îi repartizează pe o anumită secție pentru a le trata problemele cu care se confruntă. Avem nevoie de o baza de date ce va conține informații despre evidența oamenilor și a activității spitalului. Când este primit un pacient, acestuia i se atribuie un grad de urgență și în funcție de acesta pacientul este preluat mai devreme sau mai târziu de către cadrele medicale. Atunci când unui pacient nu i se cunoaște identitatea se anunță poliția, iar în funcție de gradul de urgență, este internat sau nu. În cazul în care este internat i se întocmește o fișă de internare cu numele “Persoană Necunoscută”, respectiv prenumele “Persoană Necunoscută”.

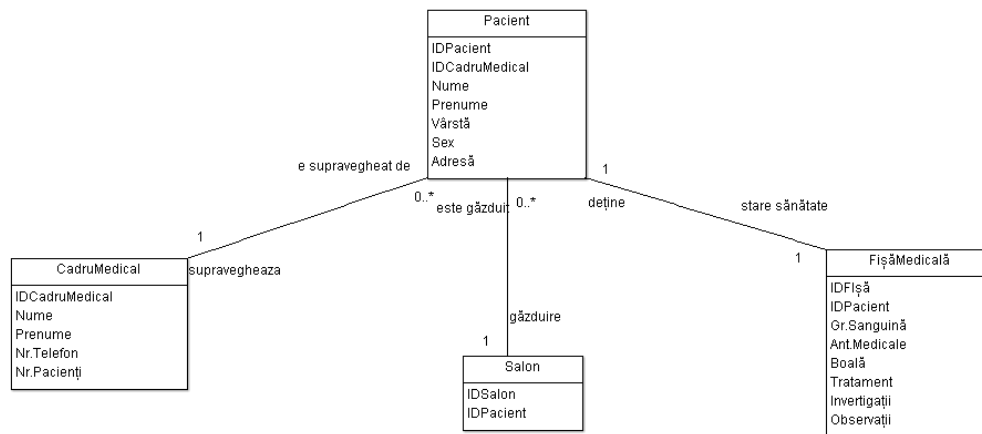
Orice pacient care vine de la UPU este preluat de un singur cadru medical, ulterior acesta îi face un consult medical, pe baza căruia i se întocmește o fișă medicală pacientului respectiv, identifică printr-un număr unic în registrul de date. Mai trebuie specificat faptul că fiecărui pacient i se oferă un pat pentru a i se putea atribui tratamentul necesar.

În fișă medicală se rețin următoarele informații despre pacient: nume, prenume, sex, vârsta, adresă, grupă sanguină, antecedente medicale, tratamente, investigații și observații.

După tratamentul efectuat la UPU, orice pacient care necesită internare este trimis cu tot cu fișă medicală pe secția unde a fost repartizat. Acolo cadrul medical îl repartizează pe pacient într-un anumit salon și într-un pat liber. Pacienților internți li se face vizită medicală zilnică, în urma căruia medicul trece în rubrica “Observații” din fișă medicală evoluția pacientului.

C2

Diagrama UML



Schema relatională

Pacient (IDPacient, IDCadruMedical, Nume, Prenume, Vârstă, Sex, Adresă)

CadruMedical (IDCadruMedical, Nume, Prenume, Nr.Telefon, Nr.Pacienți)

FișăMedicală (IDFișă, IDPacient, Gr.Sanguină, Ant.Medicale, Boală, Investigații, Observații)

Salon (IDSalon, IDPacient)

C3

PK->Primary Key; FK->Foreign Key

Pacient						
IDPacient (PK)	IDCadruMedical(FK)	Nume	Prenume	Vârstă	Sex	Adresă
IDPacient 1	IDCadruMedical 1	Nume 1	Prenume 1	Vârstă 1	Sex 1	Adresă 1
IDPacient 2	IDCadruMedical 2	Nume 2	Prenume 2	Vârstă 2	Sex 2	Adresă 2

IDPacient->IDCadruMedical

IDPacient ->Nume

IDPacient->Prenume

IDPacient->Vârsta

IDPacient->Sex

IDPacient->Adresă

CadruMedical				
IDCadruMedical(PK)	Nume	Prenume	Nr.Telefon	Nr.Pacienți
IDCadruMedical(PK) 1	Nume 1	Prenume 1	Nr.Telefon 1	Nr.Pacienți 1
IDCadruMedical(PK) 2	Nume 2	Prenume 2	Nr.Telefon 2	Nr.Pacienți 2

IDCadruMedical->Nume

IDCadruMedical->Prenume

IDCadruMedical->Nr.Telefon

IDCadruMedical->Nr.Pacienți

FișăMedicală							
IDFișă	IDPacient(FK)	Gr.Sanguină	Ant.Medicale	Boală	Tratament	Investigații	Observații
IDFișă 1	IDPacient 1	Gr.Sanguină 1	Ant.Medicale 1	Boală 1	Tratament 1	Investigații 1	Observații 1
			Ant.Medicale 2	Boală 2	Tratament 2	Investigații 2	Observații 2
IDFișă 2	IDPacient 2	Gr.Sanguină 2	Ant.Medicale 3	Boală 3	Tratament 3	Investigații 3	Observații 3
			Ant.Medicale 4	Boală 4	Tratament 4	Investigații 4	Observații 4

IDFișă->IDPacient

IDFișă->Gr.Sanguină

IDFișă->->Ant.Medicale

IDFișă->->Boală

IDFișă->->Tratament

IDFișă->->Investigații

IDFișă->->Observații

Salon	
IDSalon	IDPacient(FK)
IDSalon 1	IDPacient 1
	IDPacient 2
IDSalon 2	IDPacient 3
	IDPacient 4

IDSalon->->IDPacient

Normalizare

Pentru a fi în 1NF, fiecare tabel din baza de date trebuie să corespundă îndeplinească următoarele cerințe:

1. Tabelul trebuie să aibă o cheie primară.
2. Fiecare coloană trebuie să aibă un nume unic.
3. Nu există înregistrări identice.

Toate tabelele satisfac cerințele 2,3. Tabelele “FișăMedicală” și “Salon” nu satisfac cerința 1. Pentru a satisface și cerința 1, ne trebuie câte o cheie primară. La tabelul “FișăMedicală” cheia primară va fi “IDFișă”. După această modificare, vom obține două tabele “FișăMedicală” și “Boli”, având următoarea formă:

FișăMedicală			
IDFișă(PK)	IDPacient(FK)	Gr.Sanguină	Ant.Medicale
IDFișă 1	IDPacient 1	Gr.Sanguină 1	Ant.Medicale 1
IDFișă 2	IDPacient 2	Gr.Sanguină 2	Ant.Medicale 2

Boli					
IDTratament(PK)	IDFișă(FK)	Boală	Tratament	Investigații	Observații
IDTratament 1	IDFișă 1	Boală 1	Tratament 1	Investigații 1	Observații 1
IDTratament 2	IDFișă 2	Boală 2	Tratament 2	Investigații 2	Observații 2
IDTratament 3	IDFișă 3	Boală 3	Tratament 3	Invertigații 3	Observații 3

Pentru că tabelul “Salon” nu are cheie primară, nu îi putem atribui acestui tabel cheia primară numită “IDSalon”, deoarece într-un salon pot fi mai mulți pacienți, adică “IDSalon” poate avea mai multe înregistrări. Deci, singura și cea mai bună variantă este concatenarea tabelului “Salon” cu tabela “Pacient”.

Pacient							
IDPacient (PK)	IDCadruMedical(FK)	Nume	Prenume	Vârstă	Sex	Adresă	IDSalon
IDPacient 1	IDCadruMedical 1	Nume 1	Prenume 1	Vârstă 1	Sex 1	Adresă 1	IDSalon 1
IDPacient 2	IDCadruMedical 2	Nume 2	Prenume 2	Vârstă 2	Sex 2	Adresă 2	IDSalon 2

Pentru a fi în 2NF, fiecare tabel din baza de date trebuie să corespundă îndeplinească următoarele cerințe:

1. Să fie în 1NF.
2. Toate atributele depind de întreaga cheie primară, nu doar de o anumită parte a ei. În tabelele de mai sus întâlnim chei primare atomice, adică sunt alcătuite dintr-un singur atribut, de unde rezultă că baza de date este în 2NF.

Pentru a fi în 3NF, fiecare tabel din baza de date trebuie să corespundă îndeplinească următoarele cerințe:

1. Să fie în 2NF.
2. Toate atributele care nu fac parte din cheia primară trebuie să fie independente de celelalte atribute care nu sunt chei primare.

Toate atributele din tabelele de mai sus satisfac condiția 2, de unde rezultă că doar cheia primară poate identifica orice alt atribut din tabela din care face parte. De aici rezultă că baza de date se află în 3NF.

Pentru a fi în BCNF, fiecare tabel din baza de date trebuie să corespundă îndeplinească următoarele cerințe:

1. Să fie în 1NF (din 3NF îl conține pe 2NF, care la rândul ei 2NF îl conține pe 1NF, de unde rezultă faptul că 3NF îl conține pe 1NF).
2. Toate dependențele funcționale a tabeli trebuie să fie bazate pe cheia primară, adică cheia primară trebuie să fie singurul atribut unic din tabel.

Tabela "FișăMedicală" nu respectă condiția 2, deoarece există mai multe atribute ce pot defini dependențele funcționale "IDFișă"(PK) și "IDPacient"(FK). Pentru a rezolva această problemă despărțim tabela "FișăMedicală" folosind ca PK "IDPacient". Inițial, tabela "FișăMedicală" are cheia primară formată din două atribute "IDFișă, IDPacient". Dacă stabilim cheia primară "IDFișă" vom putea defini dependența funcțională IDPacient->IDFișă, pentru că "IDPacient" este unică, adică toate dependențele funcționale a tabeli să fie bazate pe cheia primară.

FișăMedicală		
IDFișă(PK)	Gr.Sanguină	Ant.Medicale
IDFișă 1	Gr.Sanguină 1	Ant.Medicale 1
IDFișă 2	Gr.Sanguină 2	Ant.Medicale 2
IDFișă 3	Gr.Sanguină 3	Ant.Medicale 3

FișăPacient	
IDFișă (PK)(FK)	IDPacient (PK)(FK)
IDFișă 1	IDPacient 1
IDFișă 2	IDPacient 2

Pentru a fi în 4NF, fiecare tabel din baza de date trebuie să corespundă îndeplinească următoarele cerințe:

1. Să fie în BCNF.
2. Toate tabelele nu trebuie să conțină dependențe multivaluate, adică să fie bazat doar pe cheia primară.

Toate tabelele din baza de date satisfac această condiție, de unde rezultă că tabelele se află în 4NF.

Forma finală a tabelelor:

Pacient							
IDPacient (PK)	IDCadruMedical(FK)	Nume	Prenume	Vârstă	Sex	Adresă	IDSalon
IDPacient 1	IDCadruMedical 1	Nume 1	Prenume 1	Vârstă 1	Sex 1	Adresă 1	IDSalon 1
IDPacient 2	IDCadruMedical 2	Nume 2	Prenume 2	Vârstă 2	Sex 2	Adresă 2	IDSalon 2

CadruMedical				
IDCadruMedical(PK)	Nume	Prenume	Nr.Telefon	Nr.Pacienți
IDCadruMedical(PK) 1	Nume 1	Prenume 1	Nr.Telefon 1	Nr.Pacienți 1
IDCadruMedical(PK) 2	Nume 2	Prenume 2	Nr.Telefon 2	Nr.Pacienți 2

FișăMedicală	
IDFișă(PK)	Gr.Sanguină
IDFișă 1	Gr.Sanguină 1
IDFișă 2	Gr.Sanguină 2
IDFișă 3	Gr.Sanguină 3

FișăPacient	
IDFișă (PK)(FK)	IDPacient (PK)(FK)
IDFișă 1	IDPacient 1
IDFișă 2	IDPacient 2

Boli						
IDTratament(PK)	IDFișă(FK)	Boală	Tratament	Investigații	Observații	Ant.Medicale
IDTratament 1	IDFișă 1	Boală 1	Tratament 1	Investigații 1	Observații 1	Ant.Medicale 1
IDTratament 2	IDFișă 2	Boală 2	Tratament 2	Investigații 2	Observații 2	Ant.Medicale 2
IDTratament 3	IDFișă 3	Boală 3	Tratament 3	Investigații 3	Observații 3	Ant.Medicale 3

C4

/*cascade_constraints meaning daca sterg datele din parintele, se sterge automat si datele din copil*/

drop table CadruMedical cascade constraints;

drop table FisaMedicala cascade constraints;

drop table FisaPacient cascade constraints;

drop table Pacient cascade constraints;

```
drop table Boli cascade constraints;
```

```
create table CadruMedical(  
  IDCadruMedical int not null PRIMARY KEY,  
  Nume varchar2(20) not null,  
  Prenume varchar2(20) not null,  
  NrTelefon varchar2(20),  
  NrPacienti int  
);
```

```
create table FisaMedicala(  
  IDFisa int not null PRIMARY KEY,  
  GrSanguina varchar2(5),  
  AntMedicale varchar(20) default 'Necunoscut'  
);
```

```
create table Pacient(  
  IDPacient int not null PRIMARY KEY,  
  IDCadruMedical int not null,  
  Nume varchar2(30) default 'PersNecunoscuta',  
  Prenume varchar2(30) default 'PersNecunoscuta',  
  Varsta int,  
  Sex varchar2(5) not null,  
  Adresa varchar2(30) default 'Necunoscuta',  
  IDSalon int not null  
);
```



```
create table FisaPacient(  
IDFisa int not null,  
IDPacient int not null  
);
```

```
create table Boli(  
IDTratament int not null PRIMARY KEY,  
IDFisa int not null,  
Boala varchar2(20),  
Tratament varchar2(20),  
Investigatii varchar2(20),  
Observatii varchar2(20)  
);
```

```
select * from "FISAMEDICALA";  
select * from "FISAPACIENT";  
select * from BOLI;  
select * from CADRUMEDICAL;  
select * from PACIENT;
```

/*prin alter table am putut pune FK*/

```
alter table Pacient add FOREIGN KEY (IDCadruMedical) REFERENCES  
CadruMedical(IDCadruMedical);  
alter table FisaPacient add FOREIGN KEY (IDPacient) REFERENCES Pacient(IDPacient);  
alter table FisaPacient add FOREIGN KEY (IDFisa) REFERENCES FisaMedicala(IDFisa);  
alter table Boli add FOREIGN KEY (IDFisa) REFERENCES FisaMedicala(IDFisa);
```

/*atunci cand facem update la tabela Boli automat in tabela FisaMedicala o sa puna la AntMedical = Inexistent pentru IDFisa dat la intrare*/

```
CREATE TRIGGER adauga_boala AFTER UPDATE ON Boli
FOR EACH ROW
BEGIN
UPDATE FisaMedicala
SET AntMedicale = 'Inexistent'
WHERE IDFisa = :new.IDFisa;
END;
```

```
insert into CadruMedical values(123,'Badea','Stefan','0720561045',12);
insert into CadruMedical values(189,'Bunea','Andrei','0765356562',5);
insert into CadruMedical values(892,'Andreiescu','Tudor','0784532233',165);
insert into CadruMedical values(100,'Susan','Cosmin','0768952336',17);
insert into CadruMedical values(313,'Raduta','Laurentiu','0798104547',1);
```

```
insert into FisaMedicala values(982,'0','Cardiopatie');
insert into FisaMedicala values(650,'1','Gastrita');
insert into FisaMedicala values(345,'2','Durere Cap');
insert into FisaMedicala values(159,'3','Osteoporoza');
insert into FisaMedicala values(780,'0','Gastita');
insert into FisaMedicala (IDFisa,GrSanguina) values(870,'1');
```

```
insert into Pacient(IDPacient,IDCadruMedical,Varsta,Sex,Adresa,IDSalon)
values(555,123,32,'M','Stefan cel Mare',15);

insert into Pacient(IDPacient,IDCadruMedical,Nume,Prenume,Varsta,Sex,IDSalon)
values(32,189,'Zamfir','Mihai',12,'M',10);

insert into Pacient values(62,100,'Boboc','Florin',20,'M','Creta',60);
```

insert into Pacient values(45,313,'Castor','Cornelia',45,'F','Speranta',31);

insert into Pacient values(935,892,'Danca','Aurelia',100,'F','Crim',23);

insert into FisaPacient values(982,555);

insert into FisaPacient values(345,32);

insert into FisaPacient values(780,45);

insert into FisaPacient values(650,45);

insert into FisaPacient values(159,935);

insert into Boli values(1972,982,'Scolioza','Fizio','Analize','Dureri');

insert into Boli values(6258,345,'Raie','Vaccin','Anomalie','Mancarimi');

insert into Boli values(6912,159,'Durere de Cap','Pastile','Neurolog','Nici una');

insert into boli values(3123,650,'Strabism','Ochelari','Oftalmolog','Exercitii');

insert into Boli values(2331,780,'Salmonela','Sange','Diaree','Dureri de burta');

update Boli set TRATAMENT='Paracetamol' where IDFISA=650;

select * from "FISAMEDICALA";

select * from "FISAPACIENT";

select * from BOLI;

select * from CADRUMEDICAL;

select * from PACIENT;

C5

/*view inerent actualizat*/

drop view Detalii;

```
create view Detalii as select IDTratament,Boala,Observatii from Boli where IDFisa=982 or  
Observatii='Dureri de burta';
```

```
select * from Detalii;
```

```
select * from Boli;
```

```
delete from Detalii where IDTRATAMENT=2331;
```

```
select * from Detalii;
```

```
select * from Boli;
```

```
/*view si trigger*/
```

```
drop view Detalii;
```

```
create view Detalii as select IDTratament,Boala,Observatii from Boli where IDFisa=982 or  
Observatii='Dureri de burta';
```

```
select * from Detalii;
```

```
select * from Boli;
```

```
drop trigger DetaliiDelete;
```

```
create trigger DetaliiDelete instead of delete on Detalii
```

```
for each row
```

```
begin
```

```
delete from Boli
```

```
where IDTratament=:old.IDTratament;
```

```
end;
```

```
delete from Detalii where IDTRATAMENT=1972;
```

```
select * from Detalii;
```

```
select * from Boli;
```

C6

```
/*index cu functie*/
```

```
create index up on Boli(upper(Boala));
```

```
select * from Boli where upper(Boala) = 'STRABISM';
```

```
explain plan
```

```
set statement_id = 'upper_username' for
```

```
select * from Boli where upper(Boala) = 'STRABISM';
```

```
SELECT PLAN_TABLE_OUTPUT FROM TABLE(DBMS_XPLAN.DISPLAY(NULL,  
'upper_username','TYPICAL'));
```