**GESTIUNEA UNUI SPITAL**

**Ion Apelia-Cosmina**

**Grupa 231**

**Universitatea din Bucuresti**

**Facultatea de Matematica si Informatica**

**Cuprins lucrare:**

[--1 Descrierea modelului 2](#_Toc104068781)

[--2 Restrictii de functionare 2](#_Toc104068782)

[--3 Entitati 2](#_Toc104068783)

[--4 Relatii 3](#_Toc104068784)

[--5 Descrierea atributelor 4](#_Toc104068785)

[*.*--6 Diagrama E/R 7](#_Toc104068786)

[--7 Diagrama conceptuala 8](#_Toc104068787)

[--8 Scheme relationale 9](#_Toc104068788)

[--9 Normalizari 9](#_Toc104068789)

[--10 Crearea unei secvențe 10](#_Toc104068790)

[--11 Crearea tabelelor 10](#_Toc104068791)

# --1 Descrierea modelului

Modelul de date ce urmeaza a fi descris va gestiona informatii legate de organizarea si functionarea unui spital, mai exact gestiunea lucratorilor si a pacientilor internati in cadrul acestuia.

Spitalul are mai multe **cladiri**, fiecare avand un numar variabil de **etaje**. In cadrul fiecarui etaj se incadreaza **sali** de diferite tipuri. Un etaj poate gazdui un singur **departament** (ATI, Medicina interna, Boli infectioase, etc). Departamentul reprezinta principala unitate organizatorica a spitalului, ce imparte angajatii in functie de specializarea lor, pentru a putea oferi pacientilor cel mai potrivit tratament, in functiile de nevoile acestora.

**Angajatii** pot avea mai multe **job**-uri, pornind de la cele de maxima importanta pentru spital, medici, asistente, pana la agenti de securitate, ingrijjitori etc. Medicii si asistentele pot avea in grija pacienti si le pot pune un **diagnostic**. Diagnosticul determina aplicarea unui **tratament**.

**Pacientii** sunt internati in sali.

# --2 Restrictii de functionare

* Intr-o cladire pot exista mai multe departamente.
* Un departament poate ocupa un singur etaj dintr-o cladire.
* Un angajat poate avea un singur job in cadrul spitalului, deci lucreaza intr-un singur departament.
* Un departament poate fi condus de un singur angajat (mannager).
* Un pacient poate avea un singur diagnostic in cadrul internarii curente.

# --3 Entitati

Pentru modelul de date referitor la gestiunea unui spital, au fost introduse entitatile: CLADIRE, ETAJ, SALA, DEPARTAMENT, ANGAJAT, JOB, PACIENT, DIAGNOSTIC, TRATAMENT, IMPLICA.

CLADIRE= constructie cu unul sau mai multe etaje ce deserveste ca spatiu de desfasurare a diverselor activitati medicale. Cheiea primara a entitatii este *id\_cladire*.

ETAJ=subdiviziune a unei cladiri din cadrul spitalului, ce gazduieste un departament. Poate avea acces general sau restrictionat (numai pentru angajati). Cheie primara: *id\_etaj.*

SALA=incapere ce se gaseste la nivelul etajelor, poate avea functii multiple, precum internare de pacienti, laborator, birou, cantina etc. Poate avea un numar variabil de locuri, in functie de scopul pe care il deserveste. Entitatea are cheia primara *id\_sala.*

DEPARTAMENT=forma de organizare a spitalului, ce imparte diversele domenii medicale in sectii mai usor de gestionat. Un departament ocupa un singur etaj al unei cladiri, are un singur manager insa poate avea mai multi angajati. Cheia primara a entitatii este *id\_dept.*

ANGAJAT= un angajat este o persoana ce presteaza servicii paltite in cadrul institutiei descrise. Acesta poate avea diverse sarcini, in functie de studii si de domeniul sau de competenta. Cheie primara: id\_angajat.

JOB=ocupatie a a unui angajat. Jobul confera angajatului un salariu, dar si un bonus, in functie de orele lucrate peste program. Entitatea are cheia primara *id\_job.*

PACIENT=persoana ce necesita asistenta medicala. Un pacient este internat intr-o sala, acestuia i se pune un diagnostic si i se administreaza un tratament. Cheia primara a entitatii este *id\_pac.*

DIAGNOSTIC= fisa medicala a pacientului, ce contine informatii despre problema de care sufera, data la care s-a descoperit prezenta bolii respective, precum si planul de tratament. Entitatea are cheia primara *id\_diagnostic.*

TRATAMENT= Totalitatea mijloacelor medicamentoase aplicate pentru a rezolva un diagnostic. Cheia primara este *plan\_tratament*.

# --4 Relatii

CLADIRE se\_gaseste DEPARTAMENT= relaţie care leagă entităţile CLADIRE şi DEPARTAMENT, reflectând legătura dintre acestea (intr-o cladire a spitalului se pot gasi mai multe departamente si cel putin unul). Cardinalitate minima 1:1, cardinalitate maxima 1:m, asa cum am precizat mai sus.

DEPARTAMENT ocupa ETAJ=relatie care creaza o legatura intre entitatiele DEPARTAMENT si ETAJ. Un departament poate ocupa doar un etaj. Cardinalitate 1:1

ETAJ sunt SALA= relatie ce leaga entitatile ETAJ si SALA, mai exact sugereaza ca la nivelul unui etaj exista mai multe sali. Cardinalitate minima 1:1, maxima 1:m (adica suprafata unui etaj poate gazdui una sau mai multe sali).

ANGAJAT lucreaza\_in DEPARTAMENT= relatia leaga entitatile ANGAJAT de DEPARTAMENT, reflectând legătura dintre acestea (angajatii sunt arondati la un departament al spitalului). Cardinalitate minima 1:1 (1 angajat lucreaza intr-un departament), maxima m:1 (mai multi angajati lucreaza intr-un departament. Conform regulilor de functionare a modelului, un angajat nu poate lucra ij mai multe departamente.).

ANGAJAT conduce DEPARTAMENT= relatia creaza o alta legatura intre entitatile ANGAJAT si DEPARTAMENT cu scopul de a asigna un manager unui departament. Cardinalitatea acestei relatii poate fi 1:0 (angajatul nu conduce nici un departament) sau 1:1 (un angajat conduce un singur departament).

ANGAJAT se\_ocupa\_cu JOB= relatie care leaga entitatile ANGAJAT si JOB, reflectand legatura dintre ele, si anume, aceea ca un angajat se ocupa cu un job. Cardinalitate 1:1 (adica un angajat poate avea un singur job).

ANGAJAT stabileste DIAGNOSTIC= relatie de tip many-to-many ce leaga entitatile ANGAJAT si DIAGNOSTIC, pe baza faptului ca un angajat poate pune un diagnostic. Cardinalite minima 0:0 (nici un angajat nu pune un diagnostic), maxima m:n (multi angajati pot pune diagnostice <<n diagnostice>>).

DIAGNOSTIC implica TRATAMENT= relatia de tip many-to-many leaga entitatile DIAGNOSTIC si TRATAMENT, reflectand faptul ca un diagnostic trebuie ameliorat cu ajutorul unui(unor) tratament. Cardinalitate minima 1:1, maxima m:m.

ANGAJAT are\_cabinet\_in SALA= relatia leaga entitatile ANGAJAT si SALA, reflectand faptul ca un medic (angajat) are cabinetul intr-o sala. Cardinalitate M:1.

PACIENT internat\_in SALA= relatie ce leaga entitatile PACIENT si SALA (indica sala in care este internat pacientul). Cardinalitate minima 0:1 (0 pacienti internati in sala) si maxima m:1.

# --5 Descrierea atributelor

Entitatea *CLADIRE* are ca atribute: *id\_cladire* (variabila de tip caracter, de lungime maxima 5, care reprezinta codul cladirii), *nr\_de etaje* (variabila de tip numar, de lungime maxima 2 ce reprezinta numarul de etaje dintr-o cladire) *adresa* (variabila de tip caracter, de lungime maxima 25, ce reprezinta adresa cladirii).

Entitatea ETAJ are ca atribute: *id\_etaj* (variabila de tip varchar2, de lungime maxima 5, ce reprezinta etajele dintr-o cladire a spitalului), *id\_dept* (variabila de tip caracter, de lungime maxima 5 ce reprezinta codul departamentului care ocupa etajul respectiv. Atributul corespunde unei valori a cheii primare din tabelul DEPARTAMENT), *tip\_acces* (variabila de tip caracter, de lungime maxima 10 ce precizeaza daca etajul este accesibil tuturor (general) sau doar angajatilor) și nr\_Sali, (variabila de tip NUMBER, de lungime mazima 2 de precizeaza nr de Sali ce se gasesc la etajul respectiv).

Entitatea SALA are ca atribute *id\_sala* (varaiabila de tip caracter de lungime maxima 5 ce reprezinta codul salii respective), *id\_etaj* (variabila de tip varchar2, de lungime maxima 5, ce reprezinta etajul la nivelu caruia se gaseste sala curenta. Atributul corespunde unei valori a cheii primare din tabelul ETAJ.), *nr\_locuri* (variabila de tip intreg, de lungime maxima 3, ce reprezinta capacitatea salii, numarul de persoane ce pot fi prezente in acelasi timp in sala).

Entitatea DEPARTAMENT are ca atribute: *id\_dept* (variabila de tip caracter, de lungime maxima 5 ce reprezinta codul departamentului), *id\_cladire* (variabila de tip caracter, de lungime maxima 5, care reprezinta codul cladirii in care se gaseste departamentul. Atributul corespunde unei valori a cheii primare din tabelul CLADIRE), *manager\_id* (variabial de tip intreg, de lungime maxima 5, ce reprezinta codul managerului departamentului curent), *dept\_desc* (variabila de tip carater, de lungime maxima 20 ce reprezinta descrierea, denumirea departamentului).

Entitatea ANGAJAT are ca atribute: *id\_angajat* (variabila de tip intreg, de lungime maxima 5, ce reprezinta codul angajatului)*, id\_dept* (variabila de tip caracter, de lungime maxima 5 ce reprezinta codul departamentului in care lucreaza angajatul. Atributul corespunde unei valori a cheii primare din tabelul DEPARTAMENT) *, id\_job* (variabila de tip caracter, de lungime maxima 5, ce reprezinta codul unui job pe care il are un angajat. Atributul corespunde unei valori a cheii primare din tabelul JOB)*, id\_diagnstic* (variabila de tip caracter, de lungime maxima 10, ce reprezinta un diagnostic pus de un angajat al spitalului. Atributul corespunde unei valori a cheii primare din tabelul DIAGNOSTIC.)*, id\_sala* (varaiabila de tip caracter de lungime maxima 5 ce reprezinta codul salii in care angajatul isi are cabinetul. Atributul corespunde unei valori a cheii primare din tabelul SALA)*, nume* (variabila de tip caracter, de lungime maxima 10, ce reprezinta numele angajatului)*, prenume* (variabila de tip caracter, de lungime maxima 20, ce reperezinta prenumele angajatului)*, oras* (variabila de tip caracter, de lungime maxima 10, ce reprezinta orasul de resedinta al angajatului)*, adresa* (variabila de tip caracter, de lugime maxima 20 ce reprezinta adresa angajatului)*.*

Entitatea JOB are drept atribute: *id\_job* (variabila de tip char de lungime maxima 5 , ce reprezinta codul jobului)*, titlu\_job* (variabila de tip char, de lungime maxima 15, ce reprezinta denumirea jobului)*, salariu* (variabila de tip intreg, de lungime maxima 6, ce trprezinta salariul unui angajat)*, bonus* (variabila de tip intreg, de lungime maxima 5, ce reprezinta bonusul pe care il poate obtine un angajat in urma realizarii ce ore suplimentare)*.*

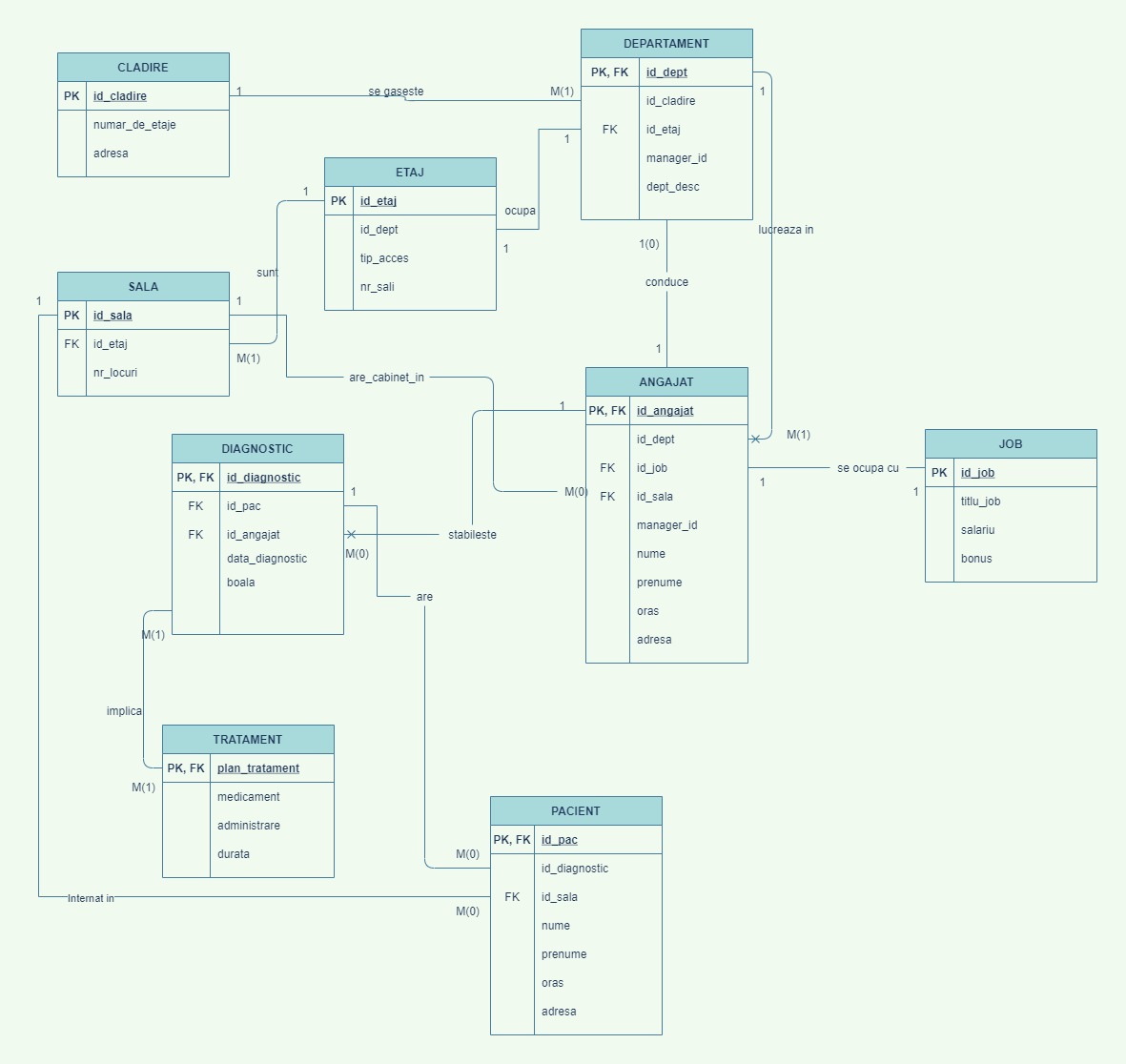
­­

Entitatea DIAGNOSTIC are atributele: *id\_diagnostic* (variabila de tip caracter, de lungime maxima 10, ce reprezinta codul unui diagnostic)*, id\_pac* (variabila de tip char, de lungime maxima 10, ce reprezinta codul pacientului caruia i s-a pus diagnosticul. Atributul corespunde unei valori a cheii primare din tabelul PACIENT*), id\_angajat* (variabila de tip intreg, de lungime maxima 5, ce reprezinta codul angajatului care a pus diagnosticul. Atributul corespunde unei valori a cheii primare din tabelul ANGAJAT )*, data\_diagnostic* (variabila de tip data calendaristica ce reprezinta data in care s-a stabilit diagnosticul)*, bolala* (variabila de tip caracter, de lungime maxima 15, ce reprezinta numele bolii diagnosticate)*.*

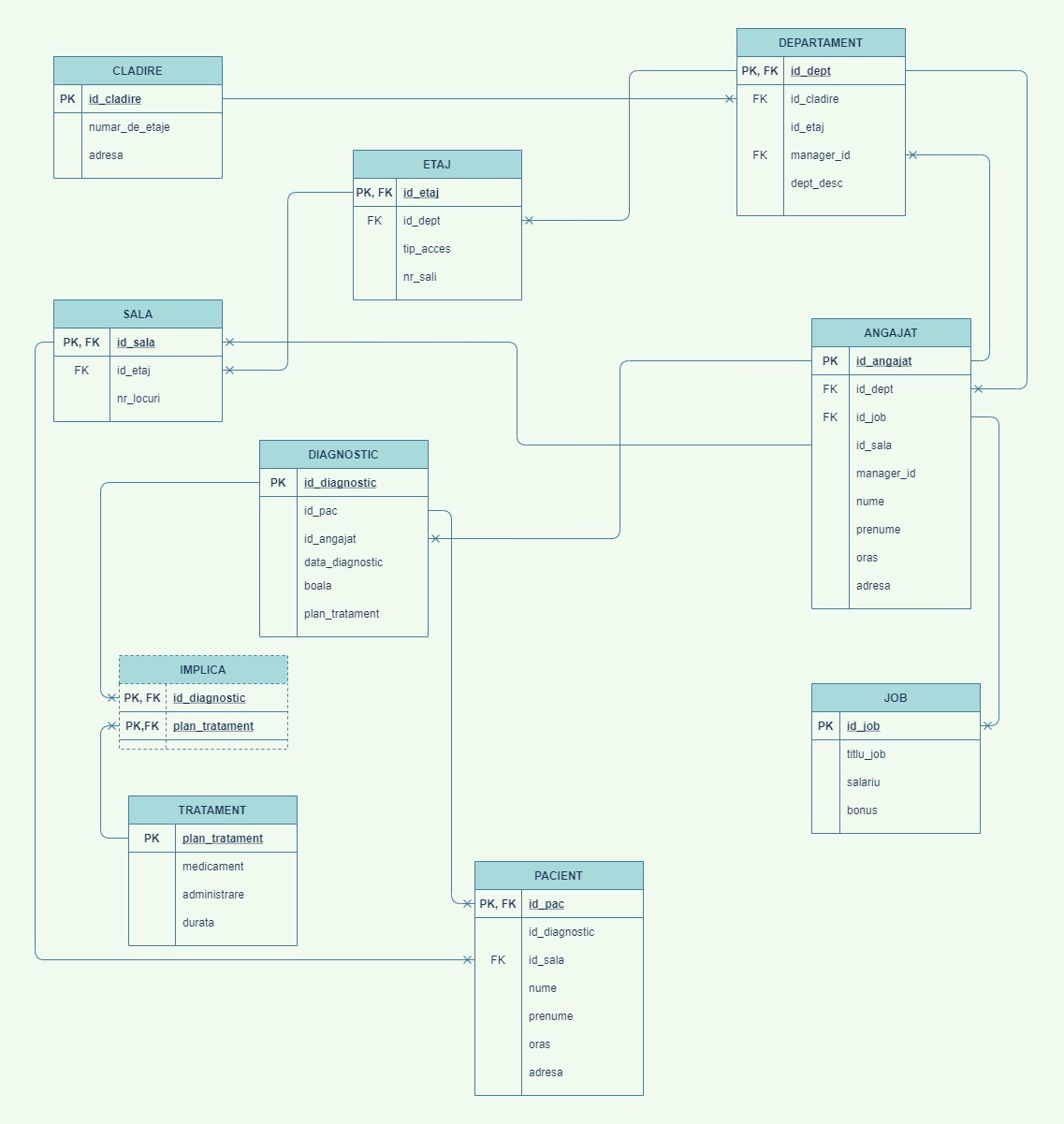
TRATAMENT are urmatoarele atribute: *plan\_tratament* (variabila de tip intreg, de lungime maxima 4, ce reprezinta codul unui tratament), *medicament* (variabila de tip caracter, de lungime maxima 20 ce reprezinta numele unui medicament), *administrare* (variabila de tip caracter, de lungime maxima 20 ce reprezinta modul de administrare a medicamentului), *durata* (variabila de tipvarchar2, de lungime maxima 20 ce reprezinta numarul de zile de tratament).

Entitatea PACIENT are atributele: *id\_pac* (variabila de tip numar, de lungime maxima 10, ce reprezinta codul pacientului, id\_diagnostic (variabila de tip varcahr2, de lungime maxima 10 bytes, ce reprezinta id-ul diagnosticului asociat pacientului, *id\_sala* (varaiabila de tip caracter de lungime maxima 5 ce reprezinta codul salii in care este internat pacentul. Atributul corespunde unei valori a cheii primare din tabelul SALA)*, nume* (variabila de tip caracter, de lungime maxima 10, ce reprezinta numele pacientului)*, prenume* (variabila de tip caracter, de lungime maxima 20, ce reperezinta prenumele pacientului)*, oras* (variabila de tip caracter, de lungime maxima 10, ce reprezinta orasul de resedinta al pacientului)*, adresa* (variabila de tip caracter, de lugime maxima 15 ce reprezinta adresa pacientului).

# *.*--6 Diagrama E/R



# --7 Diagrama conceptuala



# --8 Scheme relationale

**Schemele relationale** corespunzatoare diagramei conceptuale de pe pagina anterioara sunt urmatoarele:

CLADIRE(id\_cladire#, nr\_de\_etaje, adresa)

ETAJ(id\_etaj#, id\_dept, tip\_acces, nr\_sali)  
SALA(id\_sala#, nr\_etaj, nr\_locuri)

DEPARTAMENT(id\_dept#, id\_cladire, id\_etaj, manager\_id, dept\_descr)

ANGAJAT(id\_angajat#, id\_dept, id\_job, id\_sala, manager\_id, nume, prenume, oras, adresa)  
JOB(id\_job#, titlu\_job, salariu, bonus)

DIAGNOSTIC(id\_diagnostic#, id\_pac, id-angajat, data\_diagnostic, boala)

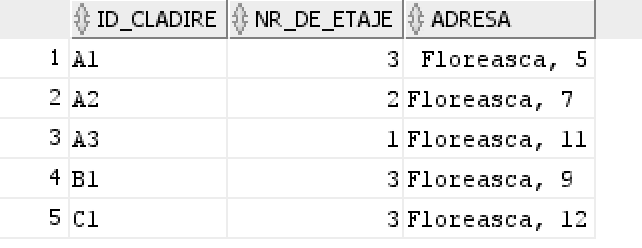
IMPLICA(id\_diagnostic#, plan\_tratament#)  
TRATAMENT(plan\_tratement#, medicament, administrare, durata)  
PACIENT(id\_pac#, id\_diagnostic, id\_sala, nume, prenume, oras, adresa)

# --9 Normalizari

--FN1:

Pentru normalizarea fn1 voi considera tabelul CLADIRE:

Se obesrva ca atributul adrresa contine valori multiple: de exemplu in datele introduse in baza de date (tabela de mai jos), cladirea cu id\_cladire=A1 are adresa Floreasca 5, reprezentand strada si respective numarul cladirii.



Pentru a adduce tabelul CLADIRE la fn1, atributul adresa ar trebui inlocuit de alte 2 atribute, respective strada si numar.

Totusi, in baza de date am utilizat tabelul ce nu este in fn1, deoarece atributul adresa nu este folosit in queries.

--FN2

Pentru a islustra normalizarea pana la forma fn2, voi considera tabelul IMPLICA ce contine 2 chei primare: id\_diagnostic si plan\_tratament. In baza de date create, tabelul nu mai contine si alte atribute, in afara celor 2 mentionate, insa cu scopul realizarii normalizarii la fn2, voi considera urmatoarea forma a acestui tabel:

IMPLICA(id\_diagnostic#, plan\_tratament#, medicament)

In acest exemplu, se observa faptul ca atributul medicament depinde doar de o parte a cheii primare compuse, mai exact de atribulul plan\_tratament.

Pentru a aduce tabelul la fn2, trebuie impartit tabelul in 2, astfel:

IMPLICA(id\_diagnostic#, plan\_tratament#)  
TABEL\_2(plan\_tratament#, medicament)

In acest fel, tabelul implica va fi in fn2

--FN3

Pentru a ilustra normalizarea FN3, voi considera urmatorul tabel, un mix intre tabelele ANGAJAT si JOB. Voi numi tabelul NOTFN3

NOTFN3(id\_angajat#, id\_job, titlu\_job, nume, prenume)

In tabelul dat ca exemplu exista o relatie de dependent tranzitiva, intre id\_job si titlu\_job. Motiv pentru care nu este in FN3.

Pentru a fi in FN3 putem separa tabelul dat in:

TABEL\_1(id\_angajat#, id\_job, nume, prenume)

TABEL\_2(id\_job#, titlu\_job)

Sau in tabelele ANGAJAT si JOB, asa cum a fost gandit in baza de date implementata.

# --10 Crearea unei secvențe

CREATE SEQUENCE secventa

start with 1

increment by 1

minvalue 1

maxvalue 100000

nocycle;

--folosita pt inserarae datelor (id\_pac) in tabelul PACIENT

# --11 Crearea tabelelor

CREATE TABLE CLADIRE (

id\_cladire VARCHAR(5) PRIMARY KEY,

nr\_de\_etaje NUMBER(2),

adresa VARCHAR(25)

);

INSERT INTO CLADIRE (ID\_CLADIRE, NR\_DE\_ETAJE, ADRESA)

VALUES ('A1', 3, ' Floreasca, 5');

INSERT INTO CLADIRE (ID\_CLADIRE, NR\_DE\_ETAJE, ADRESA)

VALUES ('A2', 2, 'Floreasca, 7');

INSERT INTO CLADIRE (ID\_CLADIRE, NR\_DE\_ETAJE, ADRESA)

VALUES ('A3', 1, 'Floreasca, 11');

INSERT INTO CLADIRE (ID\_CLADIRE, NR\_DE\_ETAJE, ADRESA)

VALUES ('B1', 3, 'Floreasca, 9');

INSERT INTO CLADIRE (ID\_CLADIRE, NR\_DE\_ETAJE, ADRESA)

VALUES ('C1', 3, 'Floreasca, 12');

--Departament---------------------------------------------------------------------

CREATE TABLE DEPARTAMENT (

id\_dept VARCHAR(5) PRIMARY KEY,

id\_cladire VARCHAR(5),

id\_etaj varchar(5),

id\_angajat number(5),

manager\_id number(5),

dept\_descr VARCHAR(20)

);

INSERT INTO DEPARTAMENT (ID\_DEPT, ID\_CLADIRE, ID\_ETAJ, MANAGER\_ID, DEPT\_DESCR)

VALUES ('U', 'A1', 'A1\_1', 1, 'Urgente');

INSERT INTO DEPARTAMENT (ID\_DEPT, ID\_CLADIRE, ID\_ETAJ, MANAGER\_ID, DEPT\_DESCR)

VALUES ('ATI', 'A1', 'A1\_2', 12, 'Terapie intensiva');

INSERT INTO DEPARTAMENT (ID\_DEPT, ID\_CLADIRE, ID\_ETAJ, MANAGER\_ID, DEPT\_DESCR)

VALUES ('MI', 'A1', 'A1\_3', 5, 'Medicina interna');

INSERT INTO DEPARTAMENT (ID\_DEPT, ID\_CLADIRE, ID\_ETAJ, MANAGER\_ID, DEPT\_DESCR)

VALUES ('OFT', 'A2', 'A2\_1', 6, 'Oftalmologie');

INSERT INTO DEPARTAMENT (ID\_DEPT, ID\_CLADIRE, ID\_ETAJ, MANAGER\_ID, DEPT\_DESCR)

VALUES ('DERM', 'A2', 'A2\_2', 10, 'Dermatologie');

INSERT INTO DEPARTAMENT (ID\_DEPT, ID\_CLADIRE, ID\_ETAJ, MANAGER\_ID, DEPT\_DESCR)

VALUES ('NEF', 'B1', 'B3\_1', 8, 'Nefrologie');

--Etaj----------------------------------------------------------------------------

CREATE TABLE ETAJ (

id\_etaj varchar(5) PRIMARY KEY,

id\_dept VARCHAR(5),

tip\_acces VARCHAR(10),

nr\_sali NUMBER (2)

);

INSERT INTO ETAJ (ID\_ETAJ, ID\_DEPT, TIP\_ACCES, NR\_SALI)

VALUES ('A1\_1', 'U', 'general', 5);

INSERT INTO ETAJ (ID\_ETAJ, ID\_DEPT, TIP\_ACCES, NR\_SALI)

VALUES ('A1\_2', 'ATI', 'angajati', 7);

INSERT INTO ETAJ (ID\_ETAJ, ID\_DEPT, TIP\_ACCES, NR\_SALI)

VALUES ('A1\_3', 'MI', 'angajati', 4);

INSERT INTO ETAJ (ID\_ETAJ, ID\_DEPT, TIP\_ACCES, NR\_SALI)

VALUES ('A2\_1', 'OFT', 'general', 2);

INSERT INTO ETAJ (ID\_ETAJ, ID\_DEPT, TIP\_ACCES, NR\_SALI)

VALUES ('A2\_2', 'DERM', 'general', 2);

INSERT INTO ETAJ (ID\_ETAJ, ID\_DEPT, TIP\_ACCES, NR\_SALI)

VALUES ('B1\_1', 'NEF', 'general', 6);

--SALA-----------------------------------------------------------------------------

CREATE TABLE SALA(

id\_sala VARCHAR(5) PRIMARY KEY,

id\_etaj varchar(5),

nr\_locuri NUMBER(3)

);

INSERT INTO SALA (ID\_SALA, ID\_ETAJ, NR\_LOCURI)

VALUES ('U1', 'A1\_1', 3);

INSERT INTO SALA (ID\_SALA, ID\_ETAJ, NR\_LOCURI)

VALUES ('Ati1', 'A1\_2', 15);

INSERT INTO SALA (ID\_SALA, ID\_ETAJ, NR\_LOCURI)

VALUES ('Ati2', 'A1\_2', 20);

INSERT INTO SALA (ID\_SALA, ID\_ETAJ, NR\_LOCURI)

VALUES ('Oft1', 'A2\_2', 4);

INSERT INTO SALA (ID\_SALA, ID\_ETAJ, NR\_LOCURI)

VALUES ('Nef1', 'B1\_1', 2);

INSERT INTO SALA (ID\_SALA, ID\_ETAJ, NR\_LOCURI)

VALUES ('MI1', 'A1\_3', 10);

INSERT INTO SALA (ID\_SALA, ID\_ETAJ, NR\_LOCURI)

VALUES ('MI2', 'A1\_3', 7);

INSERT INTO SALA (ID\_SALA, ID\_ETAJ, NR\_LOCURI)

VALUES ('Mi3', 'A1\_3', 1);

INSERT INTO SALA (ID\_SALA, ID\_ETAJ, NR\_LOCURI)

VALUES ('U2', 'A1\_1', 2);

--LEGATURI\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ALTER TABLE DEPARTAMENT

ADD CONSTRAINT leg\_cladire\_dept

FOREIGN KEY (id\_cladire) REFERENCES CLADIRE (id\_cladire) ON DELETE SET NULL;

ALTER TABLE ETAJ

ADD CONSTRAINT leg\_dept\_etaj

FOREIGN KEY (id\_dept) REFERENCES DEPARTAMENT (id\_dept) ON DELETE SET NULL;

ALTER TABLE SALA

ADD CONSTRAINT leg\_etaj\_sala

FOREIGN KEY (id\_etaj) REFERENCES ETAJ (id\_etaj) ON DELETE SET NULL;

--Angajat---------------------------------------------------------------------------------

CREATE TABLE ANGAJAT (

id\_angajat NUMBER(5) PRIMARY KEY,

id\_dept VARCHAR(5),

id\_job VARCHAR(5),

id\_sala VARCHAR(5),

manager\_id NUMBER(5),

nume VARCHAR(10),

prenume VARCHAR(20),

oras VARCHAR(10),

adresa VARCHAR(15)

);

INSERT INTO ANGAJAT (ID\_ANGAJAT, ID\_DEPT, ID\_JOB, ID\_SALA, MANAGER\_ID, NUME, PRENUME, ORAS, ADRESA)

VALUES (1, 'U', 'Med', 'U1', NULL, 'Nelu', 'Constantin', 'Bucuresti', 'str. sarii');

INSERT INTO ANGAJAT (ID\_ANGAJAT, ID\_DEPT, ID\_JOB, ID\_SALA, MANAGER\_ID, NUME, PRENUME, ORAS, ADRESA)

VALUES (2, 'U', 'Asist', NULL, 1, 'Ion', 'Agripina', 'Cluj', 'str. Siret');

INSERT INTO ANGAJAT (ID\_ANGAJAT, ID\_DEPT, ID\_JOB, ID\_SALA, MANAGER\_ID, NUME, PRENUME, ORAS, ADRESA)

VALUES (3, 'MI', 'Med', NULL, 3, 'Maria', 'Andreea', 'Giurgiu', 'Str. Campaniei');

INSERT INTO ANGAJAT (ID\_ANGAJAT, ID\_DEPT, ID\_JOB, ID\_SALA, MANAGER\_ID, NUME, PRENUME, ORAS, ADRESA)

VALUES (4, 'MI', 'An', NULL, 3, 'Georgescu', 'Adrian', 'Bucuresti', 'Str. Proiect');

INSERT INTO ANGAJAT (ID\_ANGAJAT, ID\_DEPT, ID\_JOB, ID\_SALA, MANAGER\_ID, NUME, PRENUME, ORAS, ADRESA)

VALUES (5, 'MI', 'Ch', 'Mi3', NULL, 'Neculce', 'Marian', 'Bucuresti', 'Str Scanteii');

INSERT INTO ANGAJAT (ID\_ANGAJAT, ID\_DEPT, ID\_JOB, ID\_SALA, MANAGER\_ID, NUME, PRENUME, ORAS, ADRESA)

VALUES (6, 'OFT', 'Med', NULL, NULL, 'Stan', 'Maria', 'Bucuresti', 'str. Teilor');

INSERT INTO ANGAJAT (ID\_ANGAJAT, ID\_DEPT, ID\_JOB, ID\_SALA, MANAGER\_ID, NUME, PRENUME, ORAS, ADRESA)

VALUES (7, 'OFT', 'Asist', NULL, 6, 'Dinu', 'Alexandra', 'Giurgiu', 'str. Pietrelor');

INSERT INTO ANGAJAT (ID\_ANGAJAT, ID\_DEPT, ID\_JOB, ID\_SALA, MANAGER\_ID, NUME, PRENUME, ORAS, ADRESA)

VALUES (8, 'NEF', 'Med', 'Nef1', NULL, 'Stanescu', 'Paul', 'Pitesti', 'str. Valurilor');

INSERT INTO ANGAJAT (ID\_ANGAJAT, ID\_DEPT, ID\_JOB, ID\_SALA, MANAGER\_ID, NUME, PRENUME, ORAS, ADRESA)

VALUES (9, 'NEF', 'Asist', NULL, 8, 'Velicu', 'Cristina', 'Bucuresti', 'str. Sadoveanu');

INSERT INTO ANGAJAT (ID\_ANGAJAT, ID\_DEPT, ID\_JOB, ID\_SALA, MANAGER\_ID, NUME, PRENUME, ORAS, ADRESA)

VALUES (10, 'DERM', 'Med', NULL, NULL, 'Oprea', 'Luminita', 'Ploiesti', 'str. Unirii');

INSERT INTO ANGAJAT (ID\_ANGAJAT, ID\_DEPT, ID\_JOB, ID\_SALA, MANAGER\_ID, NUME, PRENUME, ORAS, ADRESA)

VALUES (11, 'DERM', 'Asist', NULL, 10, 'Oancea', 'Georgiana', 'Galati', 'str. Plopilor');

INSERT INTO ANGAJAT (ID\_ANGAJAT, ID\_DEPT, ID\_JOB, ID\_SALA, MANAGER\_ID, NUME, PRENUME, ORAS, ADRESA)

VALUES (12, 'ATI', 'Med', NULL, NULL, 'Udrea', 'Ileana', 'Iasi', 'str. Depozit');

INSERT INTO ANGAJAT (ID\_ANGAJAT, ID\_DEPT, ID\_JOB, ID\_SALA, MANAGER\_ID, NUME, PRENUME, ORAS, ADRESA)

VALUES (13, 'ATI', 'Med', NULL, 12, 'Mihail ', 'Constantin', 'Bucuresti', 'str. Florilor');

INSERT INTO ANGAJAT (ID\_ANGAJAT, ID\_DEPT, ID\_JOB, ID\_SALA, MANAGER\_ID, NUME, PRENUME, ORAS, ADRESA)

VALUES (14, 'ATI', 'Asist', NULL, 12, 'Manea', 'Thomas', 'Bucuresti', 'bd. Timisoara');

--LEGATURI\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ALTER TABLE ANGAJAT

ADD CONSTRAINT leg\_dept\_angajat

FOREIGN KEY (id\_dept) REFERENCES DEPARTAMENT (id\_dept) ON DELETE SET NULL;

ALTER TABLE DEPARTAMENT

ADD CONSTRAINT leg\_angajat\_dept

FOREIGN KEY (manager\_id) REFERENCES angajat(id\_angajat) ON DELETE SET NULL;

--JOB------------------------------------------------------------------------------

CREATE TABLE JOB (

id\_job VARCHAR(5) PRIMARY KEY,

titlu\_job VARCHAR(15),

salariu NUMBER(6),

bonus NUMBER(5)

);

INSERT INTO JOB (ID\_JOB, TITLU\_JOB, SALARIU, BONUS)

VALUES ('Med', 'Medic Rezident', 5000, 1000);

INSERT INTO JOB (ID\_JOB, TITLU\_JOB, SALARIU, BONUS)

VALUES ('Asist', 'Asistent', 3000, 500);

INSERT INTO JOB (ID\_JOB, TITLU\_JOB, SALARIU, BONUS)

VALUES ('Sec', 'Agent sec.', 1500, 1500);

INSERT INTO JOB (ID\_JOB, TITLU\_JOB, SALARIU, BONUS)

VALUES ('An', 'Anestezist', 7000, 4500);

INSERT INTO JOB (ID\_JOB, TITLU\_JOB, SALARIU, BONUS)

VALUES ('Ch', 'Chirurg', 10000, 6000);

--LEG\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ALTER TABLE Angajat

ADD CONSTRAINT leg\_angajat\_job

FOREIGN KEY (id\_job) REFERENCES Job (id\_job) ON DELETE SET NULL;

--Diagnostic------------------------------------------------------------------------

CREATE TABLE DIAGNOSTIC (

id\_diagnostic VARCHAR(10) PRIMARY KEY,

id\_pac VARCHAR(10),

id\_angajat NUMBER(5),

data\_diagnostic date default (sysdate),

boala VARCHAR(20)

);

INSERT INTO DIAGNOSTIC (ID\_DIAGNOSTIC, ID\_PAC, ID\_ANGAJAT, DATA\_DIAGNOSTIC, BOALA)

VALUES ('frac', '1', 1, to\_date('10-06-2020', 'DD-MM-YYYY'), 'Fractura');

INSERT INTO DIAGNOSTIC (ID\_DIAGNOSTIC, ID\_PAC, ID\_ANGAJAT, DATA\_DIAGNOSTIC, BOALA)

VALUES ('ir', '2', 8, to\_date('11-08-2020', 'DD-MM-YYYY'), 'Insuficienta Renala');

INSERT INTO DIAGNOSTIC (ID\_DIAGNOSTIC, ID\_PAC, ID\_ANGAJAT, DATA\_DIAGNOSTIC, BOALA)

VALUES ('mio', '3', 6, to\_date('14-03-21', 'DD-MM-YYYY'), 'miopie');

INSERT INTO DIAGNOSTIC (ID\_DIAGNOSTIC, ID\_PAC, ID\_ANGAJAT, DATA\_DIAGNOSTIC, BOALA)

VALUES ('ast', '4', 6, to\_date('19-11-19', 'DD-MM-YYYY'), 'astigmatism');

INSERT INTO DIAGNOSTIC (ID\_DIAGNOSTIC, ID\_PAC, ID\_ANGAJAT, DATA\_DIAGNOSTIC, BOALA)

VALUES ('derm', '5', 10, NULL, 'dermatita');

INSERT INTO DIAGNOSTIC (ID\_DIAGNOSTIC, ID\_PAC, ID\_ANGAJAT, DATA\_DIAGNOSTIC, BOALA)

VALUES ('traum', '6', 1, NULL, 'traumatim');

INSERT INTO DIAGNOSTIC (ID\_DIAGNOSTIC, ID\_PAC, ID\_ANGAJAT, DATA\_DIAGNOSTIC, BOALA)

VALUES ('apend', '7', 3, NULL, 'apendicita acuta');

INSERT INTO DIAGNOSTIC (ID\_DIAGNOSTIC, ID\_PAC, ID\_ANGAJAT, DATA\_DIAGNOSTIC, BOALA)

VALUES ('inf', '8', 1, to\_date('03-03-2021', 'DD-MM-YYYY'), 'infectie');

INSERT INTO DIAGNOSTIC (ID\_DIAGNOSTIC, ID\_PAC, ID\_ANGAJAT, DATA\_DIAGNOSTIC, BOALA)

VALUES ('sept', '9', 1, NULL, 'septicemie');

INSERT INTO DIAGNOSTIC (ID\_DIAGNOSTIC, ID\_PAC, ID\_ANGAJAT, DATA\_DIAGNOSTIC, BOALA)

VALUES ('lit', '10', 8, to\_date('01/09/2021', 'DD-MM-YYYY'), 'litiaza renala');

INSERT INTO DIAGNOSTIC (ID\_DIAGNOSTIC, ID\_PAC, ID\_ANGAJAT, DATA\_DIAGNOSTIC, BOALA)

VALUES ('infa', '11', 13, to\_date('05-07-2021', 'DD-MM-YYYY'), 'infarct');

--LEG\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ALTER TABLE Diagnostic

ADD CONSTRAINT leg\_angajat\_diagnostic

FOREIGN KEY (id\_angajat) REFERENCES ANGAJAT (id\_angajat) ON DELETE SET NULL;

--PACIENT-----------------------------------------------------------------

CREATE TABLE PACIENT (

id\_pac number(10) PRIMARY KEY,

id\_diagnostic VARCHAR(10),

--id\_angajat NUMBER(5),

id\_sala VARCHAR(5),

nume VARCHAR(10),

prenume VARCHAR(20),

oras VARCHAR(10),

adresa VARCHAR(15)

);

--SECVENTA~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

CREATE SEQUENCE secventa

start with 1

increment by 1

minvalue 1

maxvalue 100000

nocycle;

--LEG\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ALTER TABLE Pacient

ADD CONSTRAINT leg\_diagnostic\_pacient

FOREIGN KEY (id\_diagnostic) REFERENCES Diagnostic (id\_diagnostic) ON DELETE SET NULL;

ALTER TABLE PACIENT

ADD CONSTRAINT leg\_sala\_pacient

FOREIGN KEY (id\_sala) REFERENCES SALA (id\_sala) ON DELETE SET NULL;

INSERT INTO PACIENT (ID\_PAC, ID\_DIAGNOSTIC, ID\_SALA, NUME, PRENUME, ORAS, ADRESA)

VALUES (secventa.nextval, 'frac', 'U1', 'Ilioiu', 'Ana', 'Bucuresti', 'Str. Astra');

INSERT INTO PACIENT (ID\_PAC, ID\_DIAGNOSTIC, ID\_SALA, NUME, PRENUME, ORAS, ADRESA)

VALUES (secventa.nextval, 'traum', 'U1', 'Sambure', 'Costin', 'Cluj', 'Str. Jiului');

INSERT INTO PACIENT (ID\_PAC, ID\_DIAGNOSTIC, ID\_SALA, NUME, PRENUME, ORAS, ADRESA)

VALUES (secventa.nextval, 'mio', 'Oft1', 'Capsuna', 'Marian', 'Bucuresti', 'Str. Macesului');

INSERT INTO PACIENT (ID\_PAC, ID\_DIAGNOSTIC, ID\_SALA, NUME, PRENUME, ORAS, ADRESA)

VALUES (secventa.nextval, 'ir', 'Nef1', 'Marin', 'Andreea', 'Giurgiu', 'Str. Turcilor');

INSERT INTO PACIENT (ID\_PAC, ID\_DIAGNOSTIC, ID\_SALA, NUME, PRENUME, ORAS, ADRESA)

VALUES (secventa.nextval, 'ast', 'Oft1', 'Hodosan', 'Toma', 'Bucuresti', 'Str. Pierzaniei');

INSERT INTO PACIENT (ID\_PAC, ID\_DIAGNOSTIC, ID\_SALA, NUME, PRENUME, ORAS, ADRESA)

VALUES (secventa.nextval, 'ir', 'Nef1', 'Neculce', 'Elena', 'Bucuresti', 'Str. Crinului');

INSERT INTO PACIENT (ID\_PAC, ID\_DIAGNOSTIC, ID\_SALA, NUME, PRENUME, ORAS, ADRESA)

VALUES (secventa.nextval, 'traum', 'U1', 'Tomsa', 'Popa', 'Bucuresti', 'Str. Ploii');

INSERT INTO PACIENT (ID\_PAC, ID\_DIAGNOSTIC, ID\_SALA, NUME, PRENUME, ORAS, ADRESA)

VALUES (secventa.nextval, 'frac', 'U1', 'Nedelciu', 'Adriana', 'Iasi', 'Str. Castanelor');

INSERT INTO PACIENT (ID\_PAC, ID\_DIAGNOSTIC, ID\_SALA, NUME, PRENUME, ORAS, ADRESA)

VALUES (secventa.nextval, 'frac', 'U2', 'Dinu', 'Cosmin', 'Bucuresti', 'Str. Iasomiei');

--TRATAMENT-------------------------------------------------------------------

CREATE TABLE TRATAMENT (

plan\_tratament NUMBER(4) PRIMARY KEY,

medicament VARCHAR(20),

administrare VARCHAR(20),

durata VARCHAR(20)

);

INSERT INTO TRATAMENT (PLAN\_TRATAMENT, MEDICAMENT, ADMINISTRARE, DURATA)

VALUES (1, 'Ghips', 'imediata', '4 saptamani');

INSERT INTO TRATAMENT (PLAN\_TRATAMENT, MEDICAMENT, ADMINISTRARE, DURATA)

VALUES (2, 'dializa', NULL, 'nedeterminata');

INSERT INTO TRATAMENT (PLAN\_TRATAMENT, MEDICAMENT, ADMINISTRARE, DURATA)

VALUES (3, 'prescriptie ochelari', NULL, 'nedeterminata');

INSERT INTO TRATAMENT (PLAN\_TRATAMENT, MEDICAMENT, ADMINISTRARE, DURATA)

VALUES (4, 'prescriptie ochelari', NULL, 'nedeterminata');

INSERT INTO TRATAMENT (PLAN\_TRATAMENT, MEDICAMENT, ADMINISTRARE, DURATA)

VALUES (5, 'antibiotic', 'zilnica', '2 saptamani');

INSERT INTO TRATAMENT (PLAN\_TRATAMENT, MEDICAMENT, ADMINISTRARE, DURATA)

VALUES (6, 'Analgezic', 'zilnica', '1 saptamana');

INSERT INTO TRATAMENT (PLAN\_TRATAMENT, MEDICAMENT, ADMINISTRARE, DURATA)

VALUES (7,'operatie','imediata','2 zile');

INSERT INtO TRATAMENT (PLAN\_TRATAMENT, MEDICAMENT, ADMINISTRARE, DURATA)

VALUES (8,'antibiotice','zilnica','3 saptamani');

INSERT INTO TRATAMENT (PLAN\_TRATAMENT, MEDICAMENT, ADMINISTRARE, DURATA)

VALUES (9,'antibiotice','zilnica','3 saptamani');

INSERT INTO TRATAMENT (PLAN\_TRATAMENT, MEDICAMENT, ADMINISTRARE, DURATA)

VALUES (10,'operatie, tratament','imediata','1 saptamana');

INSERT INTO TRATAMENT (PLAN\_TRATAMENT, MEDICAMENT, ADMINISTRARE, DURATA)

VALUES (11,'anticoagulante','imediata','5 zile');

--IMPLICA---------------------------------------------------------------------

CREATE TABLE IMPLICA (

id\_diagnostic VARCHAR(10),

plan\_tratament NUMBER(4),

CONSTRAINT PK\_implica PRIMARY KEY (id\_diagnostic, plan\_tratament)

);

--LEG\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ALTER TABLE IMPLICA

ADD CONSTRAINT leg\_diagnostic\_implica

FOREIGN KEY (id\_diagnostic) REFERENCES Diagnostic (id\_diagnostic) ON DELETE SET NULL;

--LEG\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ALTER TABLE IMPLICA

ADD CONSTRAINT leg\_tratament\_implica

FOREIGN KEY (plan\_tratament) REFERENCES TRATAMENT (plan\_tratament) ON DELETE SET NULL;

INSERT INTO IMPLICA (ID\_DIAGNOSTIC, PLAN\_TRATAMENT)

VALUES ('frac', 1);

INSERT INTO IMPLICA (ID\_DIAGNOSTIC, PLAN\_TRATAMENT)

VALUES ('ir', 2);

INSERT INTO IMPLICA (ID\_DIAGNOSTIC, PLAN\_TRATAMENT)

VALUES ('mio', 3);

INSERT INTO IMPLICA (ID\_DIAGNOSTIC, PLAN\_TRATAMENT)

VALUES ('ast', 4);

INSERT INTO IMPLICA (ID\_DIAGNOSTIC, PLAN\_TRATAMENT)

VALUES ('derm', 5);

INSERT INTO IMPLICA (ID\_DIAGNOSTIC, PLAN\_TRATAMENT)

VALUES ('traum', 6);

INSERT INTO IMPLICA (ID\_DIAGNOSTIC, PLAN\_TRATAMENT)

VALUES ('apend', 7);

INSERT INTO IMPLICA (ID\_DIAGNOSTIC, PLAN\_TRATAMENT)

VALUES ('inf', 8);

INSERT INTO IMPLICA (ID\_DIAGNOSTIC, PLAN\_TRATAMENT)

VALUES ('sept', 9);

INSERT INTO IMPLICA (ID\_DIAGNOSTIC, PLAN\_TRATAMENT)

VALUES ('lit', 10);

INSERT INTO IMPLICA (ID\_DIAGNOSTIC, PLAN\_TRATAMENT)

VALUES ('inf', 11);

ALTER TABLE DEPARTAMENT

DROP COLUMN id\_angajat;

