

# **ADMINISTRAREA BAZEI DE DATE A UNEI CLINICI**

---

**PROIECT BAZE DE DATE**

**GONTESCU MARIA-RUXANDRA**

**01.07.2022**

# Cuprins

- 1.Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.
- 2.Prezentarea constrângerilor(restricții, reguli) impuse asupra modelului.
- 3.Descrierea entităților, incluzând precizarea cheii primare.
- 4.Descrierea relațiilor, incluzând precizarea cardinalității acestora.
- 5.Descrierea atributelor, incluzând tipul de date și eventualele constrângerii, valori implicate, valori posibile ale atributelor.
- 6.Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.
- 7.Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6.
- 8.Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.
- 9.Realizarea normalizării până la forma normală 3(FN1-FN3).
- 10.Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea .
- 11.Formulați în limbaj natural și implementați 5 cereri SQL
- 12.Implementarea a 3 operații de actualizare sau suprimare a datelor utilizând subcereri.
- 13.Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele(punctul 10).
- 14.Crearea unei vizualizări compuse. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.
- 15.Crearea unui index care să optimizeze o cerere de tip căutare cu 2 criterii.  
Specificați cererea.
- 16.Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outer-join pe minimum 4 tabele și două cereri ce utilizează operația division.
- 17.Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile operatorilor algebrei relaționale. Cererea va fi exprimată prin expresie algebraică, arbore algebraic și limbaj (SQL), atât anterior cât și ulterior optimizării.
- 18.a. Realizarea normalizării BCNF, FN4, FN5.  
b. Aplicarea denormalizării, justificând necesitatea acesteia.

## 1. Descrierea modelului real a utilității acestuia și a regulilor de funcționare.

Vom prezenta modelul de date, restricțiile pe care trebuie să le respecte și vom încerca, într-o maniera didactică, să construim diagrama E/R corespunzătoare. Vom considera, în abordarea inițială, anumite situații care nu sunt optime, în sensul în care pot genera redundanță, anomalii la reactualizare sau nu permit rezolvarea anumitor interogări asupra modelului. Vom încerca să arătăm care sunt deficiențele modelului, situațiile care le-au generat și cum pot fi corectate.

Baza de date conține informații cu privire la locațiile unei rețele de clinici, sectoarele și facilitățile acesteia (cabinete medicale, laboratoare de analize, saloane), cât și personalul care lucrează în cadrul clinicii.

Scopul creării acestei baze de date este de a putea să evidențieze organizarea personalului din cadrul clinicii (personal medical, administrativ și pacienți).

În cadrul unei clinici există mai multe categorii de personal. Personalul medical este alcătuit din medici, asistenți medicali și brancardieri. În plus, există și TESA (director, contabili, economiști). Aceștia reprezintă angajații ai unei clinici.

O clinică poate avea mai multe secții, care conțin cabinete, laboratoare de analize și saloane la care pot fi internați pacienții. În cadrul instituției medicale pacienții pot apela în mod facil la diverse servicii prin intermediul unei programări. Serviciile medicale pot fi realizate atât de medici în ceea ce privește investigațiile amănunte, controalele medicale, cât și de asistenții medicali în cazul unor analize de rutina. Se va achita pe baza unei facturi care va conține prețul final al serviciilor de care a beneficiat pacientul. Prețul inițial poate fi micșorat în cazul în care pacientul este asigurat la Casa de Sănătate sau dacă este client fidel.

## 2. Restrictii de functionare

Modelul de date respecta anumite restricții de funcționare.

- Un medic este angajat la cel putin o clinica, iar o clinica angajează cel putin un medic.
- O clinică are cel putin o sectie.
- O clinică are cel putin un medic, un asistent medical, un brancardier și un angajat TESA.
- Într-un salon se poate afla cel mult 2 pacienți.
- Pentru a fi pacient în cadrul unei clinici, o persoana trebuie să aibă cel puțin o programare
- O programare pentru consult medical poate fi realizată numai de un medic, în timp ce analizele medicale, EKG-urile pot fi realizate și de asistenții medicali.
- Pe factura sunt incluse serviciile de care a beneficiat respectivul pacient al clinicii.
- O factura nu poate să fie vida.
- Persona de contact a unui cabinet poate fi numai medic.
- Cabinetele conțin un asistent\_sef
- Într-un salon există 2 pacienți distincți, de același sex.

### 3. Entități

Pentru modelul de date referitor la gestiunea unei clinici, structurile CLINICA, SECTIE\_CLINICA, CABINET, LABORATOR, SALON, CATEGORII\_DE\_PERSONAL, MEDIC, ASISTENȚI\_MEDICALI, BRANCARDIERI, TESA, ANGAJATI, SALON, PROGRAMARI etc. reprezintă entități.

Vom prezenta entitățile modelului de date, dând o descriere completa a fiecăreia. De asemenea, pentru fiecare entitate se va preciza cheia primara.

Toate entitățile care vor fi prezentate sunt independente, cu excepția entităților dependente ....

CLINICA = instituția în care se desfășoară procesele medicale. Cheia primară este id\_clinica.

SECTIE\_CLINICA = modalitate de clasificare a serviciilor din cadrul clinicii. Cheia primară este cod\_sectie.

CABINET = modalitate de clasificare în cadrul unei secții. Cheia primară este id\_cabinet.

LABORATOARE = în cadrul laboratorului se realizează analizele medicale de către un anumit asistent medical. Acestea depind de cabinete. Cheia primară este id\_laborator.

SALOANE = în salon sunt cazați pacienții care sunt internați. Cheia primară este id\_salon.

CATEGORII\_DE\_PERSONAL = modalitate de a clasifica personalul din clinica.

MEDICI = persoane fizice care pot realiza consulturi medicale. Cheia primară este id\_medic.

ASISTENȚI\_MEDICALI = persoane fizice care lucrează în cadrul clinicii. Asistenți medicali se ocupă în principal de realizarea

analizelor, pot fi asignați unui cabinet sau unui salon. Cheia primară este id\_asistent.

TESA = persoane fizice care lucrează în departamentul administrativ(contabili, directori, economiști). Cheia primară este id\_tesa.

BRANCARDIERI = persoane fizice care realizează servicii în cadrul clinicii. Cheia primară este id\_brancardier.

PACIENTI =persoane fizice beneficiază de servicii în cadrul unei clinici. Cheia primară este id\_patient.

INTERNATI = persoane fizice care se află în regim de internare în cadrul unui salon. Cheia primară este id\_internat.A

PROGRAMARI\_CONSULT = un pacient se programează pentru consult la un medic.Consultul are loc într-un cabinet. Cheia primară este id\_programare.

PROGRAMARI\_ANALIZE = un pacient este programat pentru a face analize în cadrul unui cabinet de către un asistent medical. Cheia primară este id\_analiza.

SERVICII = fiecare secție din cadrul clinicii oferă anumite facilități. Cheia primară este id\_serviciu.

CONSULT\_SERVICII = pentru a beneficia consult medical este nevoie de programare pentru acestea.Cheia primară este id\_consult\_servicii.

ANALIZE\_SERVICII = pentru a beneficia analize medicale este nevoie de programare pentru acestea.Cheia primară este id\_analiza\_servicii.

#### 4. Relații

Vom prezenta relațiile modelului de date, dând o descriere completă a fiecareia. De fapt, denumirile acestor legături sunt sugestive, reflectând conținutul acestora și entitățile pe care le leagă. Pentru fiecare relație se va preciza cardinalitatea minimă și maximă.

MEDIC\_este\_angajat\_la\_CLINCA = relație de tip many-to-many care leagă entitățile MEDIC și CLINICA, reflectând legătura dintre acestea (la o clinica este angajat un anumit doctor). Ea are cardinalitatea minimă 1:0 (o clinica trebuie să aibă cel puțin un doctor angajat și un doctor poate să fie angajat la cel puțin o clinica) și cardinalitatea maximă m:n (o clinica poate avea mai mulți doctori angajați – de regulă, aceasta este situația – și un doctor poate să lucre la mai multe facultăți).

CLINICA\_are\_SINDICAT = relație care leagă entitățile SINDICAT și CLINICA, reflectând legătura dintre acestea (pentru o clinica, există un sindicat). Relația are cardinalitatea minimă și maximă 1:1.

CLINICA\_are\_SECTIE\_CLINICA = relație care leagă entitățile CLINICA și SECTIE\_CLINICA, reflectând legătura dintre acestea (pentru o clinica, există una sau mai multe secții). Relația are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:n.

SECTIE\_CLINICA\_are\_CABINET = relație dintre SECTIE și CABINET, reflectând legătura dintre acestea (ce cabinete din clinică sunt la o anumită secție). Ea are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:n.

CABINET\_are\_LABORATOR = relație dintre entitățile CABINET și LABORATOR, reflectând legătura dintre acestea (ce laboratoare de analize aparțin de un cabinet). Ea are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:n.

CABINET\_are\_SALOANE = relație dintre entitățile CABINET și SALOANE, reflectând legătura dintre acestea (ce saloane aparțin de un cabinet). Ea are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:n.

SALOANE\_contin\_INTERNATI = relație între entitățile SALOANE și INTERNATI, reflectând legătura dintre acestea (ce pacienți sunt cazați în

anumite saloane). Ea are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:n.

SECTIE\_CLINICA\_ofera\_SERVICII = relație între entitățile SECTIE\_CLINICA și SERVICII, reflectând legătura dintre acestea( ce servicii sunt oferite de anumite secții dintr-o clinica). Ea are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:n.

CLINICA\_contine\_CATEGORII\_DE\_PERSONAL = Ea are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:n.

CATEGORII\_DE\_PERSONAL\_reprezinta\_MEDICI = Ea are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:n.

CATEGORII\_DE\_PERSONAL\_reprezinta\_ASISTENȚI\_MEDICALI = Ea are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:n.

CATEGORII\_DE\_PERSONAL\_reprezinta\_BRANCARDIERI = Ea are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:n.

CATEGORII\_DE\_PERSONAL\_reprezinta\_TESA = Ea are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:n.

CABINET\_are\_PROGRAMARE\_CONSULT = Ea are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:n.

MEDICII\_tin\_PROGRAMARE\_CONSULT = Ea are cardinalitatea minimă 1:1 și cardinalitatea maximă 1:n.

## 5. Atribute

- Entitatea independentă FACULTATE are ca atrbute:

id\_clinica = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unei clinici.

denumire = variabilă de tip caracter, de lungime maximă 100, care reprezintă denumirea clinicii.

adresă = variabilă de tip caracter, de lungime maximă 100, care reprezintă adresa clinicii. telefon = variabilă de tip caracter, de lungime 10, care reprezintă numărul de telefon al clinicii.

mail = variabilă de tip caracter, de lungime maximă 50, care reprezintă adresa de mail a unei clinici.

fax = variabilă de tip caracter, de lungime maximă 50, care reprezintă fax-ul unei clinici.

cod\_postal = variabilă de tip caracter, de lungime maximă 6, care reprezintă codul postal al unei clinici.

CLINICA						
	<u>id_clinica#</u>					
		<u>denumire</u>				
			<u>adresa</u>			
				<u>telefon</u>		
					<u>mail</u>	
						<u>fax</u>
						<u>cod_postal</u>

id_clinica	denumire	adresa	telefon	mail	fax	cod_postal
PK	NOT NULL	NOT NULL	NOT NULL	UNIQUE	-	-

- Entitatea SECTIE\_CLINICA are ca atrbute:

cod\_sectie = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul sectiei.

denumire = variabilă de tip caracter, de lungime maximă 100, care reprezintă denumirea sectiei.

id\_clinica= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul clinicii din care face parte secția. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul CLINICA.

SECTIE_CLINICA
cod_sectie#
denumire
id_clinica

cod_sectie	denumire	id_clinica
PK	NOT NULL	FK

- Entitatea CABINET are ca atrbute:

cod\_cabinet = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul seriei.

denumire = variabilă de tip caracter, de lungime maximă 100, care reprezintă denumirea cabinet.

cod\_sectie = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul secției din care face parte seria. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul SECTIE\_CLINICA.

CABINET
cod_cabinet#
denumire
cod_sectie

cod_cabinet	denumire	cod_sectie
PK	NOT NULL	FK

- Entitatea LABORATOR are ca atrbute:

cod\_laborator= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul laboratorului.

denumire = variabilă de tip caracter, de lungime maximă 100, care reprezintă denumirea laboratorului.

cod\_cabinet= variabilă de tip întreg, de lungime maximă 5, care reprezintă codul cabinetului din care face parte laboratorul. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul CABINET.

LABORATOR		
cod_laborator#	denumire	cod_cabinet
PK	NOT NULL	FK

- Entitatea LABORATOARE are ca atribute:

id_laborator	nume	numar_sala	etaj	id_cabinet	id_responsabil
PK	NOT NULL	UNIQUE	NUMBER	FK	FK

- Entitatea SALOANE are ca atribute:

id_salon	numar_sala	etaj	id_cabinet	id_internat1	id_internat2	id_responsabil
PK	UNIQUE NOT NULL	NUMBER	FK	FK	FK	FK

- Entitatea INTERNATI are ca atribute:

id_internat	nume	prenume	sex	asigurat	telefon
PK	NOT NULL	NOT NULL	NOT NULL	NUMBER	UNIQUE NOT NULL

- Entitatea CATEGORII\_DE\_PERSONAL are ca atribute:

id_categorie	nume_categoria_personal	id_clinica
PK	NOT NULL	FK

- Entitatea MEDICI are ca atribute:

id_medic	nume	prenume	data_nastereii	sex	telefon	mail	nationalitate	id_categorie
PK	NOT NULL	NOT NULL	NOT NULL	NOT NULL	UNIQUE	UNIQUE	NOT NULL	FK

- Entitatea ASISTENTI\_MEDICALI are ca atribute:

id_asistent	nume	prenume	data_nastereii	sex	telefon	mail	nationalitate	id_categorie
PK	NOT NULL	NOT NULL	NOT NULL	NOT NULL	UNIQUE	UNIQUE	NOT NULL	FK

- Entitatea BRANCARDIERI are ca atribute:

id_brancaardier	nume	prenume	data_nastereii	sex	telefon	mail	nationalitate	id_categorie
PK	NOT NULL	NOT NULL	NOT NULL	NOT NULL	UNIQUE	UNIQUE	NOT NULL	FK

- Entitatea TESA are ca atribute:

id_tesa	nume	prenume	data_nastereii	sex	telefon	mail	nationalitate	id_categorie
PK	NOT NULL	NOT NULL	NOT NULL	NOT NULL	UNIQUE	UNIQUE	NOT NULL	FK

- Entitatea PACIENTI are ca atribute:

id_pacient		nume	prenume	Asigurat	
PK		NOT NULL	NOT NULL	NUMBER	

- Entitatea INTERNATI are ca atribute:

id_internat		nume	prenume	sex	telefon	Asigurat
PK		NOT NULL	NOT NULL	NOT NULL	UNIQUE	Asigurat

- Entitatea PROGRAMARI\_CONSULT are ca atribute:

id_programare		data_consult	id_pacient	id_medic	id_cabinet
PK		NOT NULL	FK	FK	FK

- Entitatea PROGRAMARI\_ANALIZE are ca atribute:

id_analiza		data_consult	id_pacient	id_asistent	id_laborator
PK		NOT NULL	FK	FK	FK

- Entitatea SERVICII are ca atribute:

<code>id_serviciu</code>	<code>nume</code>	<code>pret</code>	<code>id_sectie</code>
PK	NOT NULL	NUMBER	FK

- Entitatea CONSULT\_SERVICII are ca atribute:

<code>id_consult_serviciu</code>	<code>id_programare_consult</code>	<code>id_serviciu</code>
PK	NUMBER	FK

- Entitatea ANALIZE\_SERVICII are ca atribute:

<code>id_analiza_serviciu</code>	<code>id_programare_analiza</code>	<code>id_serviciu</code>
PK	NUMBER	FK

## 8. Scheme relationale

**Schemele relationale** corespunzătoare diagramei conceptuale sunt următoarele:

CLINICA ( id\_clinica#, denumire, adresa, telefon, mail, fax,  
cod\_postal);

SECTIE\_CLINICA ( cod\_sectie#, denumire, id\_clinica#);

CATEGORII\_DE\_PERSONAL ( id\_categorie#,  
nume\_categorie\_personal, #id\_clinica);

MEDICI ( id\_medic#, nume, prenume, data\_nasterii, sex, telefon,  
mail, nationalitate, id\_categorie#);

ASISTENȚI\_MEDICALI ( id\_asistent#, nume, prenume,  
data\_nasterii, sex, telefon, mail, nationalitate, id\_categorie#);

BRANCARDIERI ( id\_brancardier#, nume, prenume, data\_nasterii,  
sex, telefon, mail, nationalitate, id\_categorie#);

TESA ( id\_tesa#, nume, prenume, data\_nasterii, sex, telefon, mail,  
nationalitate, id\_categorie#);

CABINET ( id\_cabinet#, nume, numar\_sala, etaj,  
id\_persoana\_contact#, id\_asistent\_sef#, id\_sectie#);

LABORATOARE ( id\_laborator#, nume, numar\_sala, etaj,  
id\_cabinet#, id\_responsabil#);

PACIENTI ( id\_pacient#, nume, prenume, asigurat);

INTERNATI ( id\_internat#, nume, prenume, sex, asigurat, telefon);

SALOANE ( id\_salon#, numar\_sala, etaj, id\_cabinet#, id\_internat1#,  
id\_internat2#, id\_responsabil#);

PROGRAMARI\_CONSULT ( id\_programare#, data\_consult,  
id\_pacient#, id\_medic#, id\_cabinet#);

PROGRAMARI\_ANALIZE ( id\_analiza#, data\_analyze, id\_pacient#,  
id\_asistent#, id\_laborator#)

SERVICII ( id\_serviciu#, nume, pret, id\_sectie#);

CONSULT\_SERVICII (id\_consult\_servicii#,  
id\_programare\_consult#, id\_serviciu#);

ANALIZE\_SERVICII ( id\_analiza\_servicii#, id\_programare\_analiza#,  
id\_serviciu#);

## 9.Normalizare și denormalizare (18)

### **Forma normală 1 (FN1)**

O relație se află în FN1 dacă fiecărui atribut care o compune îi corespunde o valoare indivizabilă.

Forma normală 1 este și cea care impune și faptul că fiecare înregistrare să fie definită astfel încât să fie identificată unic prin intermediul unei chei primare.

În cadrul bazei de date pe care o construiesc, voi utiliza un exemplu pentru a prezenta aflarea acesteia în forma normală 1.

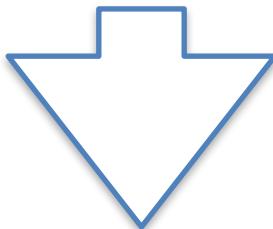
Voi lua entitățile CLINICA și SECTIE\_CLINICA. În cadrul unei clinici, denumirea secțiilor nu se repetă, dar în cadrul unui unei rețele de clinici acest lucru nu este garantat, de multe ori chiar întâmplându-se acest lucru.

#### 1. Mai multe valori semnifică același camp

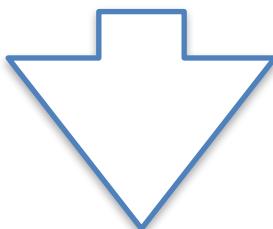
CLINICA	SECTIE_CLINICA
Sanador Buzesti	Chirurgie, Ortopedie, Oncologie
Helpsana	Pediatrie, Neonatologie, Ortopedie

A selecta intregările pe baza câmpurilor care pot conține valori semnificative sunt foarte dificile și greu de implementat. Spre exemplu, o interogare care ar selecta acele clinici care au și secție de Pediatrie și secție de Chirurgie, ar trebui să parcurgem fiecare sir din ”SECTIE\_CLINICA”, să identificăm subșirurile Pediatrie și Chirurgie și să selectăm numai acele înregistrări în care apar ambele subșiruri.

CLINICA	SECTIE_CLINICA
Sanador Buzesti	Chirurgie, Ortopedie, Oncologie
Helpsana	Pediatrie, Neonatologie, Ortopedie



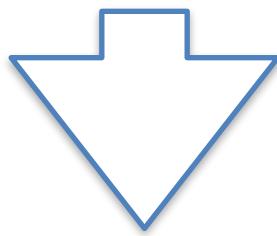
CLINICA	SECTIE_CLINICA
Sanador Buzesti	Chirurgie
Sanador Buzesti	Ortopedie
Sanador Buzesti	Oncologie
Helpsana	Pediatrie
Helpsana	Neonatologie
Helpsana	Ortopedie



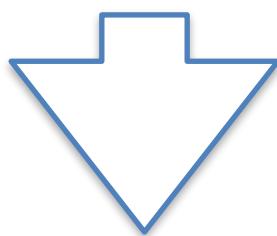
ID	CLINICA	SECTIE_CLINICA
1	Sanador Buzesti	Chirurgie
2	Sanador Buzesti	Ortopedie
3	Sanador Buzesti	Oncologie
4	Helpsana	Pediatrie
5	Helpsana	Neonatologie
6	Helpsana	Ortopedie

2. Mai multe coloane reprezintă același tip de date.

CLINICA	SECTIE_CLINICA	SECTIE_CLINICA(1)	SECTIE_CLINICA(2)
Sanador	Chirurgie	Ortopedie	Oncologie
Buzesti			
Helpsana	Pediatrie	Neonatologie	Ortopedie



CLINICA	SECTIE_CLINICA
Sanador Buzesti	Chirurgie
Helpsana	Pediatrie
Sanador Buzesti	Ortopedie
Helpsana	Neonatologie
Sanador Buzesti	Oncologie
Helpsana	Ortopedie



ID	CLINICA	SECTIE_CLINICA
1	Sanador Buzesti	Chirurgie
2	Sanador Buzesti	Ortopedie
3	Sanador Buzesti	Oncologie
4	Helpsana	Pediatrie
5	Helpsana	Neonatologie
6	Helpsana	Ortopedie

Pentru a asigura unicitatea unei inregistrari, se va utiliza cheia primara. In exemplul de mai sus, prin introducerea unei coloane aditionale de tip intreg se asigura unicitatea fiecarei inregistrari.

- **Forma Normală 2 (FN2)**

O relație se află în a doua formă normal dacă și numai dacă această relație

este deja în FN1 și fiecare atribut care nu este cheie primară este dependent de întreaga cheie primară.

FN2 interzice existența dependențelor funcționale parțiale în cadrul relației.

Dacă unul sau mai multe elemente sunt dependente funcțional numai de o parte a cheii primare, atunci ele trebuie să fie separate în tabele diferite. Dacă tabela are o cheie primară formată din numai un atribut, atunci ea este automat în FN2.

Pentru baza mea de date, voi exemplifica pentru cazul diagramei SERVICII.

<u>id_serviciu</u>	<u>nume</u>	<u>pret</u>	<u>id_sectie</u>
1	Consult si analize medicale complete pentru operatii	120	20
2	Consult si analize medicale complete pentru stabilirea unui diagnostic ortopedic	140	21
3	Consult si analize medicale complete pentru stabilirea unui diagnostic oncologic	450	22
4	Consult si analize medicale complete pentru stabilirea unui diagnostic pediatric	150	23
5	Consult si analize medicale complete pentru stabilirea unui diagnostic al nou-nascutilor	440	24

O secție poate avea mai multe servicii, un serviciu poate fi al mai multor clinici.

Relația este în FN1 – avem identificator unic pentru toate intrările din table.

Fiecare atribut care nu este cheie (nu participă la cheia primară) este dependent de întreaga cheie primară – în cazul nostru atributul nume, nu este cheie și trebuie să depindă direct de întreaga cheie primară id\_serviciu# și id\_sectie# -> aceste attribute nu depind direct de întreaga cheie primară deoarece se observă dependența directă dintre nume și id\_serviciu, însemnând că nume depinde direct doar de o parte a cheii primare, și anume doar de id\_serviciu -> relația se află în FN2.

Așteptăm ca:

- {id\_serviciu#} -> {nume} – id\_serviciu determină funcțional id\_sectie

Se aplică regula Casey Delobel și va rezulta faptul că relația este în FN2, deoarece în tabela nu se află și numele secției de unde se poate beneficia de respectivul serviciu.

- **Forma Normală 3 (FN3)**

O relație este în a treia formă normală dacă și numai dacă este în FN2 și fiecare atribut care nu este cheie depinde direct de cheia primară.

Iau tabela MEDICI. Inițial, aceasta ar fi fost de forma:

id_medic#	nume	prenume	data_nastereii	sex	telefon	mail	nationalitate	nume_categorie_personal	id_clinica
P1	N1	PR1	01-02-2000	M	7213423	-	RO	C	1
P2	N2	PR2	01-02-2000	M	72166423	-	RO	C	2
P3	N3	PR3	01-02-2000	M	72133423	-	RO	C	3

Se poate observa că atributul id\_clinica depinde de atributul nume\_categorie\_personal care depinde la randul său de cheia primară id\_categorie, astfel nefiind în NF3.

Pentru a aduce în FN3, separăm atributele despre CATEGORII\_DE\_PERSONAL din MEDICI, apărând astfel tabela CATEGORII\_DE\_PERSONAL. Astfel, înlocuiesc aceste atribută cu cheia străină id\_categorie (apărând astfel o relație one-to-many între CATEGORII\_DE\_PERSONAL și MEDICI) pentru a determina mai ușor din ce CATEGORIE\_DE\_PERSONAL face un MEDIC parte.

id_medic#	nume	prenume	data_nastereii	sex	telefon	mail	nationalitate	nume_categorie_personal	id_clinica
P1	N1	PR1	01-02-2000	M	7213423	-	RO	C	1
P2	N2	PR2	01-02-2000	M	72166423	-	RO	C	2
P3	N3	PR3	01-02-2000	M	72133423	-	RO	C	3

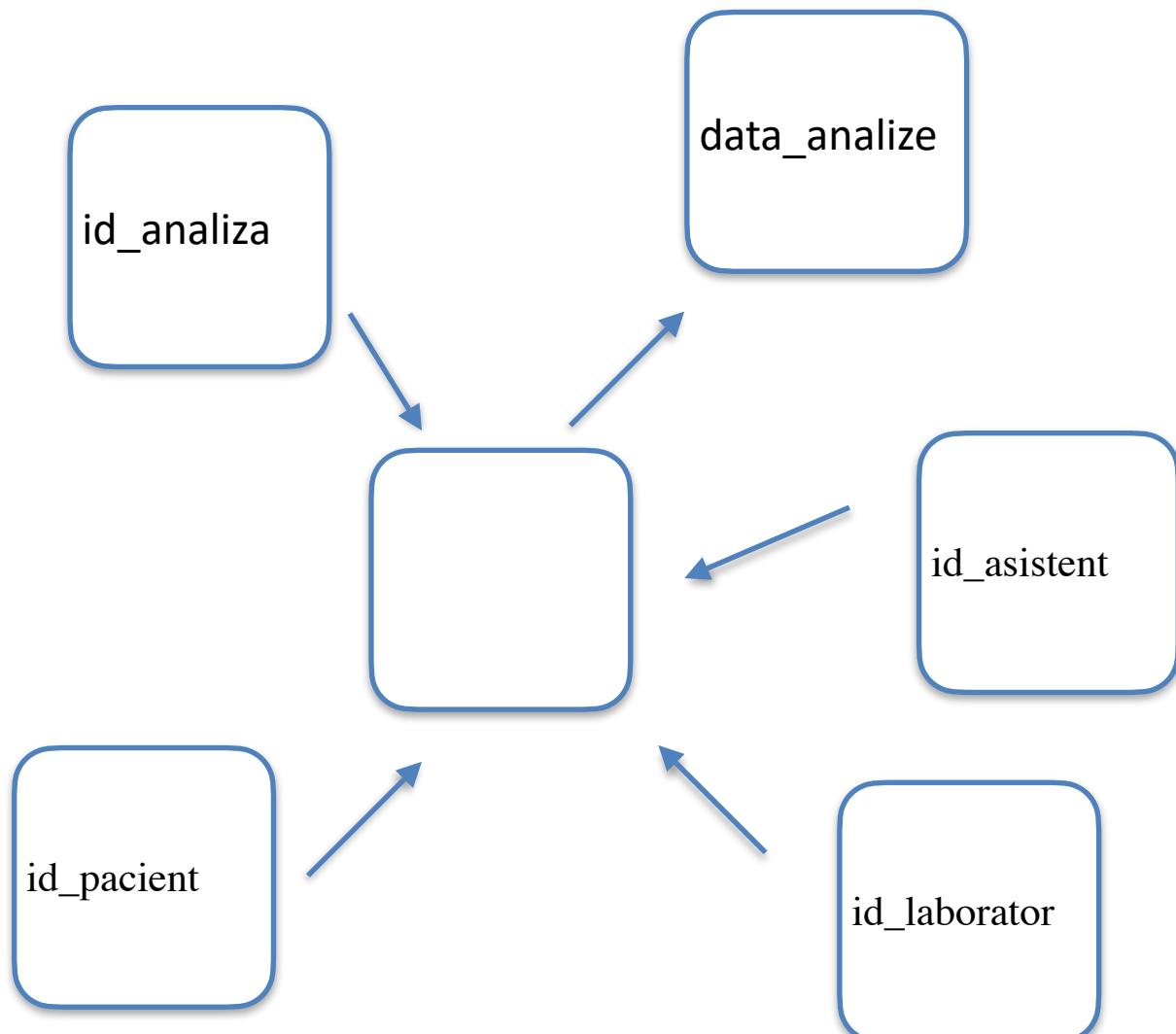
<b>id_categorie</b>	<b>nume_categorie_personal</b>	<b>id_clinica</b>
PK	NOT NULL	FK

- **Forma normală Boyce-Codd (BCNF)**

Forma normală Boyce-Codd se bazează pe dependențele funcționale care iau în considerație toate cheile candidat dintr-o relație.

Pentru relațiile cu o singura cheie candidat, formele FN3 și BCNF sunt echivalente (clinica, sectie\_clinica etc.)

Sa luam relația ASISTENTII\_realizeaza\_PROGRAMARI\_ANALIZE  
(id\_analiza#, data\_analize, id\_pacient#, id\_asistent#, id\_laborator#);



Regula Casey Delobel pentru REALIZEAZA (id\_analiza#, data\_analize, id\_pacient#, id\_asistent#, id\_laborator#) având faptul că date\_analize-> id\_analiza.

REALIZEAZA1(id\_analiza#, data\_analize)

REALIZEAZA2(data\_analize, id\_pacient#, id\_asistent#, id\_laborator#)

- **Forma nomală 4 (FN4)**

FN4 elimină redundanțele datorate relațiilor m:n, adică datorate dependenței multiple.

O relație este în a patra formă nomală dacă și numai dacă este în BCNF și nu conține relații m:n independente.

Iau relația CABINET ( id\_cabinet#, nume, numar\_sala, etaj, id\_persoana\_contact#, id\_asistent\_sef#, id\_sectie#) li presupun că o sală poate avea mai multe denumiri și mai multe persoane de contact.

id\_cabinet → → nume;

id\_cabinet → → id\_persoana\_contact;

Relația Cabinet este în BCNF. Pentru a aduce relația în FN4 o vom descompune prin proiecție în două relații:

CABINET1(id\_cabinet#, nume)

CABINET2(id\_cabinet#, id\_persoana\_contact)

→ CABINET = JOIN(CABINET1, CABINET2)

- **Forma normală 5 (FN5)**

O relație R este în FN5 (numită și forma normală proiecție-uniune) dacă și numai dacă orice dependentă de uniune a lui R este o consecință a unei chei candidat a lui R.

Orice relație care este în FN5 este și în FN4, deoarece fiecare dependență multivaloare poate fi privită ca un caz particular de dependență de uniune. Orice relație poate fi descompusă fără pierderi la uniune într-o mulțime de relații care sunt în FN5.

Pentru a preciza dacă o relație este în FN5, este suficient să cunoaștem cheile candidate și toate dependențele de uniune din R.

Aducerea în FN5 presupune eliminarea join dependentelor.

Sa luăm relația CLINICA și să presupunem că am avea o join dependență în aceasta – iau multimea (fax, adresa, cod\_postal) presupunem ca există multiple dependențe între fiecare dintre perechile din mulțimi (fax,adresa), (adresa,cod\_postal) și (fax, cod\_postal) – se pierde faptul că în acest caz cod\_postal se duce în fax - este un exemplu pur ipotetic.

#### • Denormalizare

Denormalizarea este procesul invers al procesului de normalizare.

Denormalizarea funcționează adăugând date redundante sau grupând date pentru a optimiza performanța.

Denormalizare are rolul de a realiza executarea mai rapidă a interogărilor prin introducerea redundanței, punând accentul pe rapiditatea analizei și scăzând numărul de tabele.

Motivul efectuării denormalizării îl reprezintă costurile unui produs în procesor de interogare printr-o structură supra-normalizată.

Denormalizarea poate fi definită și ca metoda de stocare a îmbinării relațiilor de formă normală superioară ca relație de bază, care se află într-o formă normală inferioară. Reduce numărul de tabele și îmbinările de tabel complicate, deoarece un număr mai mare de îmbinări poate încetini procesul.

Aici abordarea denormalizării, subliniază conceptul că, plasând toate datele într-un singur loc, ar putea elimina necesitatea căutării acestor fișiere multiple pentru a colecta aceste date.

În cadrul bazei mele de date, luând spre exemplu relația CABINET și descompunerea ei de la FN4, este inutil și mult mai costisitor din punct de vedere al timpului de executare să parcurgem datele și din CABINET1 și CABINET2.

## 10.

CREARE CLINICA

```

CREATE TABLE CLINICA
(
    id_clinica NUMBER(5) CONSTRAINT PKEY_CLINICA PRIMARY KEY,
    denumire VARCHAR(100) CONSTRAINT denumire_clinica NOT NULL,
    adresa VARCHAR(100) CONSTRAINT adresa_clinica NOT NULL,
    telefon VARCHAR(20) CONSTRAINT telefon_clinica NOT NULL,
    mail VARCHAR(50) UNIQUE,
    fax VARCHAR(50),
    cod_postal VARCHAR(6)
);

SELECT *
FROM CLINICA;
INSERT INTO CLINICA
VALUES (1, 'Sanador Buzesti', 'Strada Dr. Iacob Felix 32, Bucuresti ', '+4021-314.28.63',
        'secretariat@sanador.buzesti.ro', '+4021-314.28.63','011038');
INSERT INTO CLINICA
VALUES (2,'Helpsana','Strada Henri Coanda 11, Câmpina ','+4021-312.49.48',
        'helpsana@clinica.helpsana.ro','+4021-312.49.48','105600');
INSERT INTO CLINICA
VALUES (3,'Medical Care','Strada Grigore Nicolae nr 1, Câmpina', '+0421-305.38.10',
        'medical_care@yahoo.com','+0421-305.38.10','105600');
UPDATE CLINICA
SET

```

Query Result

ID_CLINICA	DENUMIRE	ADRESA	TELEFON	MAIL
1	Sanador Buzesti	Strada Dr. Iacob Felix 32, Bucuresti	+4021-314.28.63	secretariat@sanador.buzesti.ro
2	Medical Care	Strada Grigore Nicolae nr 1, Câmpina	+0421-305.38.10	medical_care@yahoo.com
3	Helpsana	Strada Henri Coanda 11, Câmpina	+4021-312.49.48	helpsana@clinica.helpsana.ro
4	Medicover	Strada Mihai Eminescu 7-9, Ploiești	+4021-314.35.89	secretariat@medicover.ploiesti.com
5	Lotus	Strada Avrig 14, Ploiești	+4021-313.43.36	spital_lotus@secretaria.medicover.

SECTIE

```

CREATE TABLE SECTIE_CLINICA(cod_sectie NUMBER(5) PRIMARY KEY,
                           denumire VARCHAR(100) NOT NULL,
                           id_clinica NUMBER(5),
                           FOREIGN KEY(id_clinica) REFERENCES CLINICA(id_clinica)
                           );
Select *
from SECTIE_CLINICA;

INSERT INTO SECTIE_CLINICA
VALUES(20,'Chirurgie', 1);

INSERT INTO sectie_clinica
VALUES (21, 'Ortopedie', 1);

INSERT INTO sectie_clinica
VALUES(22,'Oncologie', 1);

INSERT INTO SECTIE_clinica
VALUES(23,'Pediatrie', 2);

INSERT INTO SECTIE_clinica
VALUES (24,'Neonatalogie',2);

INSERT INTO SECTIE_clinica
VALUES (25,'Ortopedie',2);

```

Query Result

ID_CLINICA	DENUMIRE	ADRESA	TELEFON	MAIL
1	Sanador Buzesti	Strada Dr. Iacob Felix 32, Bucuresti	+4021-314.28.63	secretariat@sanador.buzesti.ro
2	Medical Care	Strada Grigore Nicolae nr 1, Câmpina	+0421-305.38.10	medical_care@yahoo.com
3	Helpsana	Strada Henri Coanda 11, Câmpina	+4021-312.49.48	helpsana@clinica.helpsana.ro
4	Medicover	Strada Mihai Eminescu 7-9, Ploiești	+4021-314.35.89	secretariat@medicover.ploiesti.com
5	Lotus	Strada Avrig 14, Ploiești	+4021-313.43.36	spital_lotus@secretaria.medicover.

```

-----CATEGORII DE PERSONAL-----
CREATE TABLE CATEGORII_DE_PERSONAL (id_categorie NUMBER(5) PRIMARY KEY,
                                     nume_categorie_personal VARCHAR(5) NOT NULL,
                                     id_clinica NUMBER(5),
                                     FOREIGN KEY(id_clinica) REFERENCES CLINICA(id_clinica)
                                     );

ALTER TABLE CATEGORII_DE_PERSONAL MODIFY NUME_CATEGORIE_PERSONAL VARCHAR(50);
select *
from categorii_de_personal;
INSERT INTO categorii_de_personal
VALUES (1, 'Medici', 1);
INSERT INTO categorii_de_personal
VALUES (2, 'Asistenti medicali', 1);
INSERT INTO CATEGORII_DE_PERSONAL
VALUES (3, 'BRANCARDIERI', 1);
INSERT INTO CATEGORII_DE_PERSONAL
VALUES (4, 'TESA', 1);
INSERT INTO CATEGORII_DE_PERSONAL
VALUES (12, 'Medici', 2);
INSERT INTO CATEGORII_DE_PERSONAL
VALUES (22, 'Asistenti medicali', 2 );
INSERT INTO CATEGORII_DE_PERSONAL
VALUES (32, 'Brancardieri', 2);
INSERT INTO CATEGORII_DE_PERSONAL
VALUES (42, 'TESA', 2 );
-----MEDICI-----

```

```

-----MEDICI-----
CREATE TABLE MEDICI (id_medic number(5) primary key,
                     nume VARCHAR(100) not null,
                     prenume varchar(150) not null,
                     data_nasterii date not null,
                     sex varchar (10) not null,
                     telefon varchar (13) unique,
                     mail varchar(30) unique,
                     nationalitate varchar(30) not null,
                     id_categorie number(5),
                     foreign key (id_categorie) references categorii_de_personal(id_categorie) );
alter table medici modify mail varchar (50) ;
select *
from medici;
insert into medici
values (100, 'Ionescu', 'Cristina', TO_DATE('12-04-1970','DD-MM-YYYY'),
        'Femeie', '+40722592883', 'ionescu_cristina3@sanador.com', 'romana', 1);
insert into medici
values (101, 'Mociornita', 'Georgel', TO_DATE('19-05-1959','DD-MM-YYYY'),
        'Barbat', '+40734769001', 'medic_mociornita.georgel@sanador.com', 'romana', 1);
insert into medici
values (102, 'Georgescu', 'Ioan Marcel', TO_DATE('22-01-1969','DD-MM-YYYY'),
        'Barbat', '+40756769441', 'georgescu_ioan.marcel@sanador.com', 'romana', 1);
insert into medici
values (103, 'Margeleanu', 'Narcisa Ioana', TO_DATE('03-09-1989','DD-MM-YYYY'),
        'Femeie', '+40770142772', 'margealnu.ioana-narcisa@sanador.com', 'romana', 1);
insert into medici

```

Query Result x

SQL | All Rows Fetched: 5 in 0.122 seconds

ID_MEDIC	NUME	PRENUME	DATA_NASTERII	SEX	TELEFON	MAIL	NA
1	100	Ionescu	Cristina	12-APR-70	Femeie	+40722592883	ionescu_cristina3@sanador.com
2	101	Mociornita	Georgel	19-MAY-59	Barbat	+40734769001	medic_mociornita.georgel@sanador.com
3	102	Georgescu	Ioan Marcel	22-JAN-69	Barbat	+40756769441	georgescu_ioan.marcel@sanador.com
4	103	Margeleanu	Narcisa Ioana	03-SEP-89	Femeie	+40770142772	margealnu.ioana-narcisa@sanador.com
5	104	Grosu	Maricica	12-APR-60	Femeie	+40722820780	grosu_maricica@sanador.com

```
--ASISTENȚI MEDICALI--  

CREATE TABLE ASISTENȚI_MEDICALI (id_asistent number(5) primary key,  

                                nume varchar(100) not null,  

                                prenume varchar (150) not null,  

                                principal varchar(5) not null,  

                                data_nasterii date not null,  

                                sex varchar (10) not null,  

                                telefon varchar (13) unique,  

                                mail varchar(30) unique,  

                                nationalitate varchar(30) not null,  

                                id_categorie number(5),  

                                foreign key (id_categorie) references categorii_de_personal(id_categorie));  

alter table asistenti_medicali modify mail varchar (50) ;  

select *  

from asistenti_medicali;  

insert into asistenti_medicali  

values (200, 'Marcu', 'Cristina', 'True', TO_DATE('19-07-1978', 'DD-MM-YYYY'),  

       'Femeie', '+40723782880', 'marcu_cristina3@asis.sanador.com', 'romana', 2);  

insert into asistenti_medicali  

values (201, 'Karakantas', 'Stavros', 'True', TO_DATE('24-07-1997', 'DD-MM-YYYY'),  

       'Barbat', '+40743967301', 'karakantas.stavros@asis.sanador.com', 'greaca', 2);  

insert into asistenti_medicali  

values (202, 'Blindariu', 'Ioan Mihnea', 'False', TO_DATE('29-05-1988', 'DD-MM-YYYY'),  

       'Barbat', '+40722300241', 'blindariu.ioan-mihnea@asis.sanador.com', 'romana', 2);  

insert into asistenti_medicali  

values (203, 'Dovanca', 'Mihai George', 'False', TO_DATE('03-04-1969', 'DD-MM-YYYY'),  

       'Barbat', '+40722642700', 'dovanca.mihai-george@asis.sanador.com', 'romana', 2);
```

Query Result

All Rows Fetched: 9 in 0.116 seconds

ID_ASISTENT	NUME	PRENUME	PRINCIPAL	DATA_NASTERII	SEX	TELEFON	MAIL
1	200	Marcu	Cristina	True	19-JUL-78	Femeie	+40723782880 marcu_cristina3@asis.sanador.com
2	201	Karakantas	Stavros	True	24-JUL-97	Barbat	+40743967301 karakantas.stavros@asis.sanador.com
3	202	Blindariu	Ioan Mihnea	False	29-MAY-88	Barbat	+40722300241 blindariu.ioan-mihnea@asis.sanador.com
4	203	Dovanca	Mihai George	False	03-APR-69	Barbat	+40722642700 dovanca.mihai-george@asis.sanador.com
5	204	Ionescu	Maricica	False	17-FEB-60	Femeie	+40760800789 ionescu_maricica@asis.sanador.com
6	205	Pavelescu	Maricica	False	10-MAR-60	Femeie	+40760300789 pavelescu.maricica@asis.sanador.com
7	206	Marculescu	Ioana	False	11-MAR-60	Femeie	+40760300799 marculescu.ioana@asis.sanador.com
8	207	Maiorescu	Daniela	False	10-MAR-60	Femeie	+40768300789 maiorescu.daniela@asis.sanador.com
9	208	Stoica	Dobre	False	10-SEP-60	Barbat	+40760300787 stoica.dobre@asis.sanador.com

Worksheet | Query Builder

```
CREATE TABLE BRANCARDIERI (id_brancardier number(5) primary key,  

                            nume varchar(100) not null,  

                            prenume varchar (150) not null,  

                            data_nasterii date not null,  

                            sex varchar (10) not null,  

                            telefon varchar (13) unique,  

                            mail varchar(50) unique,  

                            nationalitate varchar(30) not null,  

                            id_categorie number(5),  

                            foreign key (id_categorie) references categorii_de_personal(id_categorie));  

select *  

from BRANCARDIERI;  

insert into BRANCARDIERI  

values (300, 'Popescu', 'Georgeta', TO_DATE('01-12-1970', 'DD-MM-YYYY'),  

       'Femeie', '+40720598883', 'popescu.georgeta@br.sanador.com', 'romana', 3);  

insert into BRANCARDIERI  

values (301, 'Sora', 'Andrei Cristian', TO_DATE('29-09-1973', 'DD-MM-YYYY'),  

       'Barbat', '+40744769001', 'sora.andrei-cristian@br.sanador.com', 'romana', 3);  

insert into BRANCARDIERI  

values (302, 'Mohammed', 'Abdul', TO_DATE('22-04-1969', 'DD-MM-YYYY'),  

       'Barbat', '+40756769001', 'mohammed.abdu@br.sanador.com', 'turca', 3);  

insert into BRANCARDIERI  

values (303, 'Hussein', 'Abdul', TO_DATE('09-09-1999', 'DD-MM-YYYY'),  

       'Barbat', '+40720142772', 'hussein.abdul@br.sanador.com', 'egipteană', 3);  

insert into BRANCARDIERI  

values (304, 'Zorba', 'Alexandros', TO_DATE('05-04-1966', 'DD-MM-YYYY'),  

       'Barbat', '+40762820780', 'zorba.alexandros@br.sanador.com', 'greaca', 3);
```

Script Output | Query Result

All Rows Fetched: 5 in 0.112 seconds

ID_BRANCARDIER	NUME	PRENUME	DATA_NASTERII	SEX	TELEFON	MAIL
1	300	Popescu	Georgeta	01-DEC-70	Femeie	+40720598883 popescu.georgeta@br.sanador.com
2	301	Sora	Andrei Cristian	29-SEP-73	Barbat	+40744769001 sora.andrei-cristian@br.sanador.com
3	302	Mohammed	Abdul	22-APR-69	Barbat	+40756769001 mohammed.abdu@br.sanador.com
4	303	Hussein	Abdul	09-SEP-99	Barbat	+40720142772 hussein.abdul@br.sanador.com
5	304	Zorba	Alexandros	05-APR-66	Barbat	+40762820780 zorba.alexandros@br.sanador.com

```

-----TESA-----
CREATE TABLE TESA (id_tesa number(5) primary key,
                    nume varchar(100) not null,
                    prenume varchar (150) not null,
                    functie varchar (150) not null,
                    data_nasterii date not null,
                    sex varchar (10) not null,
                    telefon varchar (13) unique,
                    mail varchar(50) unique,
                    nationalitate varchar(30) not null,
                    id_categorie number(5),
                    foreign key (id_categorie) references categorii_de_personal(id_categorie)

select *
from tesa;
insert into tesa
values (400, 'Dobrii', 'Madalina Natalia','Director executiv', TO_DATE('29-12-1978','DD-MM-YYYY'),
        'Femeie', '+40723782889', 'dobrii.madalina@tesa.sanador.com', 'romana', 4);
insert into tesa
values (401, 'Pisalita', 'Marian','Contabil', TO_DATE('07-07-1997','DD-MM-YYYY'),
        'Barbat', '+40743977301', 'pisalita.marian@tesa.sanador.com', 'greaca', 4);
insert into tesa
values (402, 'Bocioaca', 'Sergiu Vasile','Contabil', TO_DATE('10-05-1958','DD-MM-YYYY'),
        'Barbat', '+40722399241', 'bociaoaca.sergiu-vasile@tesa.sanador.com', 'romana', 4);
insert into tesa
values (403, 'Dovanca', 'Serban Daniel','False', TO_DATE('03-08-1988','DD-MM-YYYY'),
        'Barbat', '+40722642722', 'dovanca.serban-daniel@tesa.sanador.com', 'romana', 4);
insert into tesa
values (404, 'Tabarca', 'Marilena','Economist', TO_DATE('19-03-1969','DD-MM-YYYY'),
        'Femeie', '+40760800782', 'tabarca.marilena@tesa.sanador.com', 'romana', 4);
update TESA
SET
    FUNCTIE = 'Director economic'
WHERE
    id_TESA = 403;|-----CABINET-----
CREATE TABLE CABINET ( id_cabinet NUMBER(5) PRIMARY KEY,
                      nume VARCHAR(100) NOT NULL,
                      numar_sala NUMBER(5),
                      etaj NUMBER(5),
                      id_persoana_contact number (5),
                      FOREIGN KEY (id_persoana_contact) REFERENCES MEDICI(id_medic),
                      id_asistent_sef number (5),
                      foreign key(id_asistent_sef) references ASISTENTI_MEDICALI(id_asistent),
                      id_sectie number (5),
                      foreign key (id_sectie) references SECTIE_clinica(cod_sectie) );

```

```

-----LABORATOARE-----
CREATE TABLE LABORATOARE( id_laborator number (5) primary key,
                           nume varchar(150) not null,
                           numar_sala number(5) unique,
                           etaj number (5),
                           id_cabinet number (5),
                           foreign key (id_cabinet) references CABINET(id_cabinet),
                           id_responsabil number (5),
                           foreign key (id_responsabil) references ASISTENTI_MEDICALI(id_asistent)

);

select *
from laboratoare;
insert into laboratoare
values (1, 'Laborator EKG', 130, 1, 1, 205);
insert into laboratoare
values (2, 'Analize sange 1', 231, 2, 2, 206);
insert into laboratoare
values (3, 'Analize Sange-Tiroidea', 232, 2, 3, 207);
insert into laboratoare
values (4, 'Analize Sange-VSH', 331, 3, 4, 208);
insert into laboratoare
values (5, 'Analize Sange-TGO-TGP', 332, 3, 4, 205);
CREATE TABLE PACIENTI (id_pacient number(5) primary key, --pacienti veniti doar pentru consult
                      nume varchar(50) not null,
                      prenume varchar(50) not null,
                      asigurat number(5) -----0 pentru false, 1 pentru true
);

ALTER TABLE PACIENTI
ADD numar_telefon number(5) unique;
alter table pacienti modify numar_telefon varchar(15);
alter table pacienti
add
select *
from pacienti;
```

```

----- --
[?] create table INTERNATI ( id_internat number(5) primary key,
    nume varchar (50) not null,
    prenume varchar(50) not null,
    sex varchar (10) not null,
    asigurat number(5),
    telefon varchar(15) unique not null);

select *
from internati;
[?] insert into internati
--alter table internati
--add
values (1, 'Angelescu', 'Marcel','Barbat', 1, '+40720300400');
insert into internati
values (2, 'Moineac', 'Vasilel','Barbat', 1, '+40720300405');
insert into internati
values (3, 'Barbulescu', 'Tudorel','Barbat', 0, '+40720300406');
insert into internati
values (4, 'Sorescu', 'Marian','Barbat', 0, '+40720300407');
insert into internati
values (5, 'Sendroiu', 'Cristian','Barbat', 1, '+40720300408');
insert into internati
values (6, 'Mat', 'Marcel','Barbat', 1, '+40720300409');
insert into internati
values (7, 'Rat', 'Cosmin','Barbat', 1, '+40720300410');
insert into internati
values (8, 'Lazarescu', 'Liviu','Barbat', 1, '+40720300411');
insert into internati
values (9, 'Puui', 'Cristian','Barbat', 1, '+40720300412');
insert into internati
values (10, 'Matei', 'Alexandru','Barbat', 1, '+40720300413');

--SALOANE
[?] CREATE TABLE SALOANE (id_salon number (5) primary key,
    numar_sala number (5) unique not null,
    etaj number (5),
    id_cabinet number (5),
    foreign key (id_cabinet) references CABINET(id_cabinet),
    id_internat1 number(5) unique,
    foreign key (id_internat1) references INTERNATI(id_internat),
    id_internat2 number(5) unique,
    foreign key (id_internat2) references INTERNATI(id_internat),
    id_responsabil number(5),
    foreign key (id_responsabil) references asistenti_medicali(id_asistent)
);
----- ..

```

```

----- PROGRAMARI -----
[?] CREATE TABLE PROGRAMARI_CONSULT (id_programare number(5) primary key,
    data_consult timestamp(0) not null,
    id_pacient number(5) ,
    foreign key (id_pacient) references pacienti(id_pacient),
    id_medic number(5) ,
    foreign key(id_medic) references medici(id_medic),
    id_cabinet number(5) ,
    foreign key(id_cabinet) references cabinet(id_cabinet)
);
alter table programari_consult modify data_consult timestamp(0);
drop table programari_consult;
select *
from programari_consult;

insert into programari_consult
values(1, TO_DATE('12-05-2022 12:30','DD-MM-YYYY hh:mi'), 1,100, 1 );
[?] delete
from programari_consult
where
id_proPROGgramare = 2;
insert into programari_consult
values(2, TO_DATE('11-05-2022 12:30','DD-MM-YYYY:hh:mi'), 2,101, 2 );
insert into programari_consult
values(3, TO_DATE('12-05-2022 11:30:00','DD-MM-YYYY:hh:mi:ssam'), 3,102, 3 );
insert into programari_consult
values(4, TO_DATE('12-05-2022 10:30:00','DD-MM-YYYY:hh:mi:ssam'), 4,102, 3 );
insert into programari_consult
values(5, TO_DATE('12-05-2022 09:30:00','DD-MM-YYYY:hh:mi:ssam'), 5,104, 5 );
----- PROGRAMARE -----

```

```

[?] create table PROGRAMARI_ANALIZE( id_analiza number(5) primary key,
    data_analize timestamp(0) not null,
    id_pacient number(5) ,
    foreign key (id_pacient) references pacienti(id_pacient),
    id_asistent number(5) ,
    foreign key(id_asistent) references asistenti_medicali(id_asistent),
    id_laborator number(5) ,
    foreign key(id_laborator) references laboratoare(id_laborator)
);

```

```

-----PROGRAMARI-----
CREATE TABLE PROGRAMARI_CONSULT (id_programare number(5) primary key,
                                  data_consult timestamp(0) not null,
                                  id_pacient number(5),
                                  foreign key (id_pacient) references pacienti(id_pacient),
                                  id_medic number(5),
                                  foreign key(id_medic) references medici(id_medic),
                                  id_cabinet number(5),
                                  foreign key(id_cabinet) references cabinet(id_cabinet)
                                );
alter table programari_consult modify data_consult timestamp(0);
drop table programari_consult;
select *
from programari_consult;

insert into programari_consult
values(1, TO_DATE('12-05-2022 12:30','DD-MM-YYYY hh:mi'), 1,100, 1 );
delete
from programari_consult
where
id_proPROGramare = 2;
insert into programari_consult
values(2, TO_DATE('11-05-2022 12:30','DD-MM-YYYY:hh:mi'), 2,101, 2 );
insert into programari_consult
values(3, TO_DATE('12-05-2022 11:30:00','DD-MM-YYYY:hh:mi:ssam'), 3,102, 3 );
insert into programari_consult
values(4, TO_DATE('12-05-2022 10:30:00','DD-MM-YYYY:hh:mi:ssam'), 4,102, 3 );
insert into programari_consult
values(5, TO_DATE('12-05-2022 09:30:00','DD-MM-YYYY:hh:mi:ssam'), 5,104, 5 );

-----PROGRAMARE-----
create table PROGRAMARI_ANALIZE( id_analiza number(5) primary key,
                                  data_analize timestamp(0) not null,
                                  id_pacient number(5),
                                  foreign key (id_pacient) references pacienti(id_pacient),
                                  id_asistent number(5),
                                  foreign key(id_asistent) references asistenti_medicali(id_asistent),
                                  id_laborator number(5),
                                  foreign key(id_laborator) references laboratoare(id_laborator)
                                );

```

```

-----M2M_PROGRAMARI_SERVICII-----
CREATE TABLE CONSULT_SERVICII (id_consult_servicii number(5) primary key,
                               id_programare_consult number(5),
                               foreign key(id_programare_consult) references programari_consult(id_programar
                               id_serviciu number(5),
                               foreign key(id_serviciu) references SERVICII(id_serviciu)
                             );
select *
from consult_servicii;
insert into consult_servicii
values (1, 1, 1);
insert into consult_servicii
values (2, 2, 2);
insert into consult_servicii
values (3, 3, 3);
insert into consult_servicii
values (4, 4, 4);
insert into consult_servicii
values (5, 5, 5);
insert into consult_servicii
values (6, 6, 6);

-----M2M_ANALIZE_SERVICII-----
CREATE TABLE ANALIZE_SERVICII ( id_analiza_servicii number(5) primary key,
                                 id_programare_analiza number(5),
                                 foreign key(id_programare_analiza) references programari_analyze(id_analiza),
                                 id_serviciu number(5),
                                 foreign key(id_serviciu) references SERVICII(id_serviciu)
                               );
drop table analize_servicii;
select *
from analize_servicii;
insert into analize_servicii
values (1, 1, 1);
insert into analize_servicii
values (2, 2, 2);
insert into analize_servicii
values (3, 3, 3);
insert into analize_servicii
values (4, 4, 4);
insert into analize_servicii
values (5, 5, 5);

```

11.

```
--11
--numele clinicilor au codul postal 105600
select unique concat (concat('Numele clinicii este ', clinica.denumire),
concat(' si are adresa ', clinica.adresa)) as "Rezultat"
from clinica
where clinica.cod_postal = '105600';
```

Script Output x Query Result x  
SQL | All Rows Fetched: 2 in 0.069 seconds  
Rezultat

1 Numele clinicii este Helpsana si are adresa Strada Henri Coanda 11, Câmpina
2 Numele clinicii este Medical Care si are adresa Strada Grigore Nicolae nr 1, Câmpina

```
--afisati numele si prenumele medicilor barbati nascuti in anul 1960
select medici.nume, medici.prenume
from medici
where medici.sex = 'Barbat' and extract(year from medici.data_nasterii) = 1959;
```

Script Output x Query R... x  
SQL | All Rows Fetched: 1 in 0.061 seconds  
NUME PRENUME

1 Mociornita Georgel
----------------------

```
--afisati numele si prenumele medicilor barbati nascuti in anul 1960 care contin litera i in nume
select medici.nume, medici.prenume
from medici
where medici.sex = 'Barbat' and extract(year from medici.data_nasterii) = 1959 and medici.nume like '%i%';
```

Script Output x Query Result x  
SQL | All Rows Fetched: 1 in 0.135 seconds  
NUME PRENUME

1 Mociornita Georgel
----------------------

```
--afisare asistenti medicali care sunt responsabili pentru saloane de internati
-- si care sunt asistenti sefi peste cabinele de la etajul 1
select concat (concat(asistenti_medicali.nume, ' '), asistenti_medicali.prenume) as "Persoanl asistent"
from asistenti_medicali
where asistenti_medicali.id_asistent in
(select id_responsabil
from saloane
where saloane.id_responsabil in
(select cabinet.id_asistent_sef
from cabinet
where cabinet.etaj = 1)
);
```

Script Output x Query Result x  
SQL | All Rows Fetched: 3 in 0.062 seconds  
Persoanl asistent

1 Karakantas Stavros
2 Marcu Cristina
3 Blindariu Ioan Mihnea

```
--pentru fiecare serviciu medical sa se afiseze ce pret are, precum si
--suma totala a preturilor
select servicii.id_serviciu, servicii.nume,
       SUM(DECODE(servicii.id_serviciu, 1, servicii.pret)) Serviciu1,
       SUM(DECODE(id_serviciu, 2, pret)) Serviciu2,
       SUM(DECODE(id_serviciu, 3, pret)) Serviciu3,
       SUM(DECODE(id_serviciu, 4, pret)) Serviciu4,
       SUM(DECODE(id_serviciu, 5, pret)) Serviciu5,
       sum(servicii.pret) as Total
from servicii
```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.09 seconds

ID_SERVICIU	NUME	SERVICIU1	SERVICIU2	SERVICIU3	SERVICIU4	SERVICIU5	TOTAL
1	2 Consult si analize m...	(null)	140	(null)	(null)	(null)	140
2	4 Consult si analize m...	(null)	(null)	(null)	150	(null)	150
3	5 Consult si analize m...	(null)	(null)	(null)	(null)	440	440
4	3 Consult si analize m...	(null)	(null)	450	(null)	(null)	450
5	1 Consult si analize m...	120	(null)	(null)	(null)	(null)	120

```
--sa se pacientii care sunt programati peste 6 luni, ordonati dupa nume
select nume, prenume
from pacienti
where pacienti.id_pacient in
(select id_pacient
from programari_consult
where months_between (sysdate, programari_consult.data_consult)=6
)
order by pacienti.nume;
```

Script Output x Query Result x

SQL | All Rows Fetched: 0 in 0.202 seconds

NUME	PRENU...
------	----------

```
from saluare,
```

```
select nume_categoria_personal,
case id_categoria
when 1 then 'Medic la clinica 1'
when 2 then 'Asistenti medicali la clinica 1'
when 3 then 'Brancardieri la clinica 1'
when 4 then 'tesa la clinica 1'
when 12 then 'Medic la clinica 2 '
when 22 then 'Asistenti medicali la clinica 2'
when 32 then 'Brancardieri la clinica 2'
when 42 then 'tesa la clinica 2'
end clasificare
from categorii_de_personal
order by
nume_categoria_personal;
```

```
from categorii_de_personal;
```

Script Output x Query Result x

SQL | All Rows Fetched: 8 in 1.794 seconds

NUME_CATEGORIE_PERSONAL	CLASIFICARE
1 Asistenti medicali	Asistenti medicali la clinica 2
2 Asistenti medicali	Asistenti medicali la clinica 1
3 BRANCARDIERI	Brancardieri la clinica 1
4 Brancardieri	Brancardieri la clinica 2
5 Medici	Medic la clinica 2
6 Medici	Medic la clinica 1
7 TESA	tesa la clinica 2
8 TESA	tesa la clinica 1

```
--serviciile cu pret mai mare decat cel mediu
with temp(average_pret) as (
  select avg(pret)
  from servicii )
select nume, pret
from servicii, temp
where servicii.pret > temp.average_pret
order by nume;
```

Script Output x Query Result x  
SQL | All Rows Fetched: 2 in 0.172 seconds

NUME	PRET
1 Consult si analize medicale complete pentru stabilirea unui diagnostic al nou-nascutilor	440
2 Consult si analize medicale complete pentru stabilirea unui diagnostic oncologic	450

```
--medici care nu sunt sefi de sectie
select medici.id_medic, medici.prenume, NVL(TO_CHAR(medici.nume), 'Not assigned') "Schimbare"
from medici
join cabinet
on
cabinet.id_persoana_contact = medici.id_medic
where cabinet.etaj = 1;
order by nume;
```

Script Output x Query Result x  
SQL | All Rows Fetched: 3 in 0.064 seconds

ID_MEDIC	PRENUME	Schimbare
1	102 Ioan Marcel	Georgescu
2	100 Cristina	Ionescu
3	101 Georgel	Mociornita

```
--id-ul medicilor, numele acestora, sectia la care lucreaza
select medici.id_medic, medici.nume, cabinet.numar_sala, sectie_clinica.denumire, clinica.denumire
from medici
inner join cabinet on cabinet.id_persoana_contact = medici.id_medic
inner join sectie_clinica on cabinet.id_sectie = sectie_clinica.cod_sectie
inner join clinica on sectie_clinica.id_clinica = clinica.id_clinica
where clinica.id_clinica = 1
order by medici.nume, medici.id_medic , cabinet.numar_sala, sectie_clinica.denumire, clinica.denumire;
```

Script Output x Query Result x  
SQL | All Rows Fetched: 5 in 0.067 seconds

ID_MEDIC	NUME	NUMAR_SALA	DENUMIRE	DENUMIRE_1
1	102 Georgescu	102 Oncologie	Sanador Buzesti	
2	104 Grosu	300 Chirurgie	Sanador Buzesti	
3	100 Ionescu	100 Chirurgie	Sanador Buzesti	
4	103 Margeleleanu	203 Oncologie	Sanador Buzesti	
5	101 Mociornita	101 Ortopedie	Sanador Buzesti	

12. Implementarea a 3 operații de actualizare sau suprimare a datelor utilizând subcereri.

```
UPDATE CLINICA
SET
    ADRESA = 'Strada Dr. Iacob Felix 32, Bucuresti'
WHERE
    id_clinica = 1;
update CLINICA
SET
    ADRESA = 'Strada Grigore Nicolae nr 1, Câmpina'
WHERE
    id_clinica = 3;
```

```
foreign key(id_sectie) references sectie_clinica(cod_sectie);

update
servicii
set pret = pret * 1.19
where id_sectie in
(select cod_sectie
from sectie_clinica
where sectie_clinica.denumire like '0%');

drop table servicii;
select *
from servicii;
insert into servicii
values (1, 'Consult si analize medicale complete pentru operatii', 120.5, 20);
insert into servicii
```

Script Output x | Task completed in 0.07 seconds  
2 rows updated.  
2 rows updated.  
2 rows updated.  
2 rows updated.

```
update cabinet
set nume = 'Cabinet general'
where cabinet.id_cabinet in
(select id_cabinet
from cabinet
where etaj = 1);
```

```

--division
--sa se obtina asistentii care au post de asistent sef in cabinete care contin la nume a doua si a treia litera
--'ab'
select *
from asistenti_medicali
where not exists
(select id_asistent_sef
from cabinet
where asistenti_medicali.id_asistent = cabinet.id_asistent_sef
minus
select id_responsabil
from laboratoare
where nume like '_ab%');

```

Script Output x Query Result x

All Rows Fetched: 4 in 0.107 seconds

ID_ASISTENT	NUME	PRENUME	PRINCIPAL	DATA_NASTERII	SEX	TELEFON	MAIL	
1	205	Pavelescu	Maricica	False	10-MAR-60	Femeie	+40760300789	pavelescu.maricica@asis.san...
2	206	Marculescu	Ioana	False	11-MAR-60	Femeie	+40760300799	marculescu.ioana@asis.sanad...
3	207	Maiorescu	Daniela	False	10-MAR-60	Femeie	+40768300789	maiorescu.daniela@asis.sanad...
4	208	Stoica	Dobre	False	10-SEP-60	Barbat	+40760300787	stoica.dobre@asis.sanador.co...

14.Crearea unei vizualizări compuse. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.

```

--14.Crearea unei vizualizări compuse.
--Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și
create view servicii_cu_pret_peste_avg as select servicii.nume, servicii.pret
from servicii
join sectie_clinica on servicii.id_serviciu = sectie_clinica.cod_sectie
where servicii.pret > (select avg(servicii.pret) from servicii);
--un exemplu de operație LMD nepermisă.

insert into servicii_cu_pret_peste_avg values ('Servicii clienti', 300);

```

Script Output x Query Result x

Task completed in 0.168 seconds

2 rows updated.

View SERVICII CU PRET PESTE\_AVG created.

Error starting at line : 642 in command -
insert into servicii\_cu\_pret\_peste\_avg values ('Servicii clienti', 300)
Error report -
ORA-01400: cannot insert NULL into ("GRUPA144"."SERVICII"."ID\_SERVICIU")

15. Crearea unui index care să optimizeze o cerere de tip căutare cu 2 criterii. Specificați cererea.

```
--15
create index idx on pacienti ( nume, prenume, asigurat)
where PACIENTI.nume like 'A%' and PACIENTI.asigurat = 1;
```

## 17. Algebra relațională

Sa se afiseze numele și prenumele persoanelor de contact (medicilor) femei din cadrul unui cabinet care lucrează la secție de Chirurgie

```
select medici.nume, medici.prenume
from medici
join cabinet on (medici.id_medic =
cabinet.id_persoana_contact)
JOIN SECTIE_CLINICA ON
SECTIE_CLINICA.COD_SECTIE = cabinet.ID_SECTIE
where medici.sex = 'Femeie' and medici.nationalitate =
'romana' AND sectie_clinica.denumire like 'Chirurgie';
```

Expresie Algebrică:

R1 = JOIN(medici, cabinet)

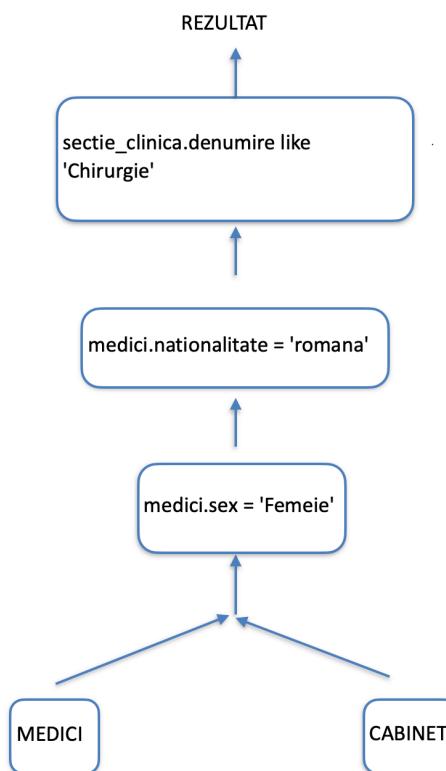
R2 = join(R1, sectie\_clinica)

R3 = SELECT(R2, medici.sex = 'Femeie')

R4 = SELECT(R3, medici.nationalitate = 'romana')

R5 = SELECT(R4, sectie\_clinica.denumire like 'Chirurgie')

REZULTAT = R5 = PROJECT(R4, nume, prenume)



16. Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outerjoin pe minimum 4 tabele și două cereri ce utilizează operația division.

## OUTER-JOIN

```
--16 outer join
select clinica.denumire, clinica.adresa
from clinica
outer join sectie_clinica on (sectie_clinica.cod_sectie = clinica.id_clinica)
outer join saloane on (saloane.id_salon = cabinet.id_cabinet)
outer join internati on (saloane.id_internat1 = internati.id_internat )
or (saloane.id_internat2 = internati.id_internat )
where internati.sex = 'Barbat';
```

