- Gestiunea activităților unor spitale -

Mai 2023

Roșu Ana-Maria, grupa 131
Facultatea de Matematică și Informatică
Universitatea din București

Cuprins

1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.....3
2. Prezentarea constrângerilor.....4
3. Descrierea entităților.....4
4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.....6
5. Descrierea atributelor.....8
6. Diagrama entitate - relație.....13
7. Diagrama conceptuală..... 15
8. Scheme relaționale.....17
9. Realizarea normalizarii.....17
10. Crearea unei secvențe.....17
11. Crearea tabelelor si inserarea datelor.....18
Crearea tabelelor:..... 18
Inserarea datelor:.....28
12. Implementarea a 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:..... 54
1. Cererea 1.....55
2. Cererea 2.....56
3. Cererea 3.....57
4. Cererea 4.....58
5. Cererea 5.....59
13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri 60
1. Operație de actualizare.....60
2. Operație de actualizare.....61
3. Operație de ștergere.....63
15. Implementarea următoarelor 3 cereri:..... 66
O cerere care utilizează operația division.....66
O cerere care implementează analiza top-n.....68
O cerere care utilizează operația outer-join pe minimum 4 tabele..... 69

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

Tema acestui proiect o reprezintă **gestiunea unei bazei de date pentru spitale**. Această bază de date ajută spitalele pentru a ține evidenţa medicilor, a pacienților, a consultațiilor, a rețetelor eliberate, cât și a transporturilor cu ambulanța.

Spitalul este format din mai multe secții, precum: pediatrie, dermatologie, neurologie, cardiologie etc., care, la rândul lor, au mai mulți medici. În fiecare secție sunt consultați mai mulți pacienți, pe baza recomandării medicului de familie. În urma acestei recomandări, pacientul este văzut de un medic specialist, la spital, în cadrul unei consultații. Pe baza acestei consultații, medicul stabilește diagnosticul pacientului și o rețetă, în cazul în care este nevoie. Tratamentul dat de medicul specialist care se ocupă de pacientul respectiv poate fi de tip medicamentos sau nu. Baza de date ține evidența și a medicamentelor prescrise unui pacient.

În plus, sunt păstrate informaţii despre toate consultațiile pe care le-a avut fiecare pacient în parte, și toate tratamentele care i-au fost prescrise, de-a lungul timpului.

De asemenea, modelul analizează informaţii legate de localizarea şi accesarea spitalelor, precum și a angajatilor.

2. Prezentarea constrângerilor, (restricții, reguli) impuse asupra modelului.

- Un spital se află la o singură adresă.
- Într-un spital lucrează cel puțin un angajat.
- În spital, lucrează angajați de tip medic sau șoferi de ambulanță.
- Un angajat poate lucra într-un singur spital.
- Un medic poate trata mai mulți pacienți.
- Un pacient poate consulta mai mulți medici.
- Între medic și pacient pot exista mai multe consultații.
- O consultatie generează un singur diagnostic.
- Un tratament se eliberează pe baza unui singur diagnostic.
- Un tratament poate include zero sau mai multe medicamente.
- Un medicament poate aparține unuia sau mai multor tratamente.
- În urma unei consultații, se emite o factură. O factura aparține unei singure consultații. O consultație emite o singură factură.

3. Descrierea entităților, incluzând precizarea cheii primare

Pentru modelul de date proiectat, structurile SPITAL, ADRESA, ANGAJAT, AMBULANȚĂ, SECȚIE, PACIENT, CONSULTAȚIE, DIAGNOSTIC, RETETA, MEDICAMENT, FACTURĂ reprezintă entități. Voi prezenta entitățile modelului de date, dând o descriere fiecăreia. Pentru fiecare entitate voi preciza cheia primară.

SPITAL = instituție în care se tratează pacienții. Cheia primară a entității este id.

ADRESA = entitate care identifică adresa în care se află un anumit spital sau la care locuiește un pacient. Cheia primară a entităţii este id.

SECȚIE = unitate de asistență medicală, componentă a unui spital, unde pacienții primesc îngrijire și tratament medical. Cheia primară a entității este id.

AMBULANȚĂ = autovehicul folosit pentru transportarea pacienților la spital. Cheia primară a entității este id.

ANGAJAT = persoană fizică, angajată într-un spital, care contribuie la buna funcționare a spitalului. Medicii oferă pacienților diagnostic, tratament și sfaturi medicale, șoferii transportă pacienții la spital în cazuri de urgență. Cheia primară a entității este id.

PACIENT = persoană fizică care persoană care primește îngrijiri medicale la spital. se prezintă într-un spital și dorește atât depistarea bolii de care suferă, conform simptomelor lui, cât și primirea unui tratament; reține date relevante despre acesta. Cheia primară a entității este id.

CONSULTAȚIE = reprezintă întâlnirile dintre pacienți și medici, în cadrul cărora se fac diverse investigații pe baza simptomelor pacientului. Cheia primară a entității este id.

DIAGNOSTIC = diagnosticul primit în urma evaluării pacientului de către medic în cadrul consultației. Cheia primară a entității este id.

RETETA = reteta eliberată pe baza diagnosticului. Cheia primară a entității este id_diagnostic, id_medicament.

MEDICAMENT = medicament ce face parte din schema unui tratament. Cheia primara este id.

FACTURĂ = factura conține date referitoare la serviciile acordate pacientului și suma pe care trebuie să o plătească

4. Descrierea relațiilor, incluzând precizarea cardinalității acestora

Voi prezenta relaţiile modelului de date, dând o descriere a fiecăreia. Denumirile acestor legături sunt sugestive, reflectând conţinutul acestora şi entităţile pe care le leagă. Pentru fiecare relaţie voi preciza cardinalitatea minimă şi maximă.

SPITAL_se_afla_ADRESA = relaţie care leagă entităţile SPITAL şi ADRESA, reflectând legătura dintre acestea (ce adresă are un spital). Relaţia are cardinalitatea minimă 0:1 (un spital trebuie să aibă o adresă, dar la o adresă nu trebuie neapărat să se afle un spital) şi cardinalitatea maximă 1:1 (un spital poate avea o singură adresă, iar la o adresă se poate găsi un singur spital).

SPITAL_are_SECTIE = relaţie care leagă entităţile SPITAL şi SECTIE, reflectând legătura dintre acestea (ce secții componente are un spital). Relaţia are cardinalitatea minimă 1:1 (un spital trebuie să aibă cel puțin o secție și o secție trebuie să aparțină unui singur spital) şi cardinalitatea maximă 1:n (un spital poate avea mai multe secții, iar o secție poate aparține unui singur spital).

SECTIE_are_MEDIC = relaţie care leagă entităţile SECTIE şi MEDIC, reflectând legătura dintre acestea (în ce secție lucrează un medic). Ea are cardinalitatea minimă 1:1 (o secție **trebuie** să aibă cel puțin un medic și un medic **trebuie** să lucreze într-o singură secție) şi cardinalitatea maximă 1:n (o secție **poate** avea mai mulți medici, iar un medic **poate** să lucreze într-o singură secție).

SPITAL_are_SOFER = relaţie care leagă entităţile SOFER şi SPITAL, reflectând legătura dintre acestea (în ce spital lucrează un șofer). Ea are cardinalitatea minimă 1:1 (un spital **trebuie** să aibă cel puțin un șofer și un șofer **trebuie** să lucreze într-un singur spital) şi cardinalitatea maximă 1:n (un spital **poate** avea mai mulți șoferi, iar un șofer **poate** să lucreze într-un singur spital).

SPITAL_are_AMBULANTA = relaţie care leagă entităţile SPITAL şi AMBULANTA, reflectând legătura dintre acestea (ce ambulanțe are un spital). Relaţia are cardinalitatea minimă 1:1 (un spital trebuie să aibă cel puțin o

ambulanță și o ambulanță trebuie să aparțină unui singur spital) şi cardinalitatea maximă 1:n (un spital poate avea mai multe ambulanțe, iar o ambulanță poate aparține unui singur spital).

MEDIC_are_CONSULTATIE = relaţie dintre entităţile MEDIC şi CONSULTAȚIE, reflectând legătura dintre acestea (de ce consultații se ocupă un medic). Ea are cardinalitatea minimă 1:0 (un medic nu trebuie neapărat să aibă o consultație dar o consultație trebuie să fie susținută un medic) şi cardinalitatea maximă 1:n (un medic poate avea mai multe consultații, dar o consultație poate fi susținută de un singur medic).

PACIENT_are_CONSULTATIE = relaţie dintre entităţile PACIENT şi CONSULTAȚIE, reflectând legătura dintre acestea (ce consultații are un pacient). Ea are cardinalitatea minimă 1:1 (un pacient trebuie să aibă o consultație, iar la o consultație trebuie sa participe un pacient) şi cardinalitatea maximă 1:n (un pacient poate avea mai multe consultații, dar la o consultație poate participa un singur pacient).

PACIENT_are_ADRESA = relaţie dintre entităţile PACIENT şi ADRESA, reflectând legătura dintre acestea (ce adresă are un pacient). Ea are cardinalitatea minimă 0:1 (un pacient trebuie să aibă o adresă, la o adresă nu trebuie neapărat să locuiască cineva) şi cardinalitatea maximă n:1 (un pacient poate să aibă o singură adresă la care locuiește, dar la o adresă pot locui mai mulți pacienți).

CONSULTATIE_stabileste_DIAGNOSTIC = relaţie dintre entităţile CONSULTATIE şi DIAGNOSTIC, reflectând legătura dintre acestea (ce diagnostic este stabilit în urma unei consultații). Ea are cardinalitatea minimă 1:1 (o consultație trebuie să genereze un diagnostic, un diagnostic trebuie să aparțină unei consultații) şi cardinalitatea maximă 1:n (

CONSULTATIE_emite_FACTURA = relaţie dintre entităţile CONSULTATIE şi FACTURA, reflectând legătura dintre acestea (ce factura este emisa în urma unei consultații). Ea are cardinalitatea minimă 1:1 şi cardinalitatea maximă 1:1.

DIAGNOSTIC_are_RETETA = relaţie dintre entităţile DIAGNOSTIC şi RETETA, reflectând legătura dintre acestea (ce retetă este recomandată pentru un diagnostic). Ea are cardinalitatea minimă 1:0 şi cardinalitatea maximă 1:1.

RETETA_include_MEDICAMENT = relaţie dintre entităţile RETETA şi MEDICAMENT, reflectând legătura dintre acestea (ce medicamente include un tratament). Ea are cardinalitatea minimă 0:0 şi cardinalitatea maximă m:n ((un tratament poate să includă 0 sau mai multe medicamente, un medicament poate să fie inclus în 0 sau în mai multe tratamente).

SOFER_preia_PACIENT_cu_AMBULANTA = relaţie de tip 3 ce leagă entităţile SOFER, PACIENT şi AMBULANTA, reflectând ce pacient a fost preluat de ce șofer, cu ce ambulanță. Denumirea acestei relaţii va fi “preia”.

5. Descrierea atributelor

Entitatea ADRESA are ca atribute:
id = variabilă de tip întreg, de lungime maximă 4, care reprezintă id-ul unei adrese
strada = variabilă de tip șir de caractere, de lungime maximă 25, care reprezintă numele străzii
numar = variabilă de tip întreg, de lungime maximă 3, care reprezintă numărul străzii
oras = variabilă de tip șir de caractere, de lungime maximă 15, care reprezintă numele străzii, și care nu poate fi nulă
judet = variabilă de tip șir de caractere, de lungime maximă 15, care reprezintă numele străzii

Entitatea SPITAL are ca atribute:
id = variabilă de tip întreg, de lungime maximă 3, care reprezintă id-ul unui spital
nume = variabilă de tip șir de caractere, de lungime maximă 40, care reprezintă numele spitalului, și care nu poate fi nulă.
data_infintarii = variabilă de tip dată calendaristică, reprezintă data înființării unui spital

id_adresa = variabilă de tip întreg, de lungime maximă 4, care reprezintă id-ul adresei în care se află spitalul. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul ADRESA.

Entitatea SECTIE are ca atribute:
id number(4) = variabilă de tip întreg, de lungime maximă 4, care reprezintă id-ul unei secții
nume = variabilă de tip șir de caractere, de lungime maximă 40, care reprezintă numele secției, și care nu poate fi nulă
nr_paturi = variabilă de tip întreg, de lungime maximă 3, care reprezintă numărul de paturi al secției
id_sef = variabilă de tip întreg, de lungime maximă 3, care reprezintă șeful secției. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul MEDIC.
id_spital = variabilă de tip întreg, de lungime maximă 3, care reprezintă id-ul spitalului din care face parte secția

Entitatea TIP_ANG are ca atribute:
id = variabilă de tip șir de caractere, de lungime maximă 5, care reprezintă tipul angajaților care pot exista în baza de date a spitalelor

Entitatea ANGAJAT are ca atribute:
id = variabilă de tip întreg, de lungime maximă 4, care reprezintă id-ul unui angajat
id_tip_ang = variabilă de tip șir de caractere, de lungime maximă 5, care reprezintă tipul angajatului (medic sau șofer). Atributul trebuie să corespundă la o valoare a cheii primare din tabelul TIP_ANG.
nume = variabilă de tip șir de caractere, de lungime maximă 8, care reprezintă numele angajatului, și care nu poate fi nulă.
prenume = variabilă de tip șir de caractere, de lungime maximă 12, care reprezintă prenumele angajatului, și care nu poate fi nulă.
data_angajarii = variabilă de tip dată calendaristică, reprezintă data angajării unui angajat, are ca valoare default ziua curentă, în cazul în care ea nu este specificată
salariu = variabilă de tip real, care reprezintă salariul angajatului

Subentitatea MEDIC are ca atribute:

id = variabilă de tip întreg, de lungime maximă 4, care reprezintă id-ul unui medic. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul ANGAJAT
id_tip_ang = variabilă de tip șir de caractere, de lungime maximă 5, care reprezintă tipul angajatului (medic sau șofer). Atributul trebuie să corespundă la o valoare a cheii primare din tabelul TIP_ANG, are valoare default 'Medic' și trebuie să fie egal cu 'Medic'.
tip = variabilă de tip șir de caractere, de lungime maximă 10, care reprezintă tipul medicului. De exemplu, poate să fie specialist, rezident sau eventual poate lua alte valori.
id_sectie = variabilă de tip întreg, de lungime maximă 4, care reprezintă id-ul secției în care lucrează în prezent medicul. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul SECTIE

Subentitatea SOFER are ca atribute:
id = variabilă de tip întreg, de lungime maximă 4, care reprezintă id-ul unui șofer. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul ANGAJAT
id_tip_ang = variabilă de tip șir de caractere, de lungime maximă 5, care reprezintă tipul angajatului (medic sau șofer). Atributul trebuie să corespundă la o valoare a cheii primare din tabelul TIP_ANG, are valoare default 'Sofer' și trebuie să fie egal cu 'Sofer'.
id_spital = variabilă de tip întreg, de lungime maximă 3, care reprezintă id-ul spitalului în care lucrează în prezent șoferul. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul SPITAL

Entitatea PACIENT are ca atribute:
id = variabilă de tip întreg, de lungime maximă 3, care reprezintă id-ul unui pacient
nume = variabilă de tip șir de caractere, de lungime maximă 10, care reprezintă numele pacientului, și care nu poate fi nulă.
prenume = variabilă de tip șir de caractere, de lungime maximă 10, care reprezintă prenumele pacientului, și care nu poate fi nulă.
data_nasterii = variabilă de tip dată calendaristică, care reprezintă data nașterii a pacientului

sex = variabilă de tip șir de caractere, de lungime maximă 1, care reprezintă sexul pacientului
id_adresa = variabilă de tip întreg, de lungime maximă 4, care reprezintă id-ul adresei pacientului. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul ADRESA.

Entitatea AMBULANTA are ca atribute:
id = variabilă de tip întreg, de lungime maximă 3, care reprezintă id-ul unei ambulanțe
nr_inmatriculare = variabilă de tip șir de caractere, de lungime maximă 8, care reprezintă numărul de înmatriculare a ambulanței, și care nu poate fi nulă.
capacitate = variabilă de tip întreg, de lungime maximă 1, care reprezintă numărul de pacienți pe care îl poate transporta concomitent o ambulanță
id_spital = variabilă de tip întreg, de lungime maximă 3, care reprezintă id-ul spitalului de care aparține ambulanța. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul SPITAL
Entitatea SERVICIU_AMBULANTA are ca atribute:
id_sofer = variabilă de tip întreg, de lungime maximă 4, care reprezintă id-ul șoferului care conduce ambulanța cu care se efectuează serviciul. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul SOFER
id_pacient = variabilă de tip întreg, de lungime maximă 3, care reprezintă id-ul pacientului căruia i se acordă serviciul de transport cu ambulanța. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul PACIENT
id_ambulanta = variabilă de tip întreg, de lungime maximă 3, care reprezintă id-ul ambulanțe cu care este transportat pacientul. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul AMBULANTA
data_serviciu = variabilă de tip dată calendaristică, reprezintă data efectuării serviciului, are ca valoare default ziua curentă, în cazul în care ea nu este specificată
motiv = variabilă de tip șir de caractere, de lungime maximă 30, care reprezintă motivul pentru care s-a apelat la serviciu. De exemplu, poate să fie accident rutier, fractură osoasă sau alte valori.

Entitatea MEDICAMENT are ca atribute:

id = variabilă de tip întreg, de lungime maximă 4, care reprezintă id-ul medicamentului
nume = variabilă de tip șir de caractere, de lungime maximă 20, care reprezintă numele medicamentului, și care nu poate fi nulă
companie = variabilă de tip șir de caractere, de lungime maximă 20, care reprezintă numele companiei producătoare a medicamentului
data_expirarii = variabilă de tip dată calendaristică, reprezintă data expirării medicamentului

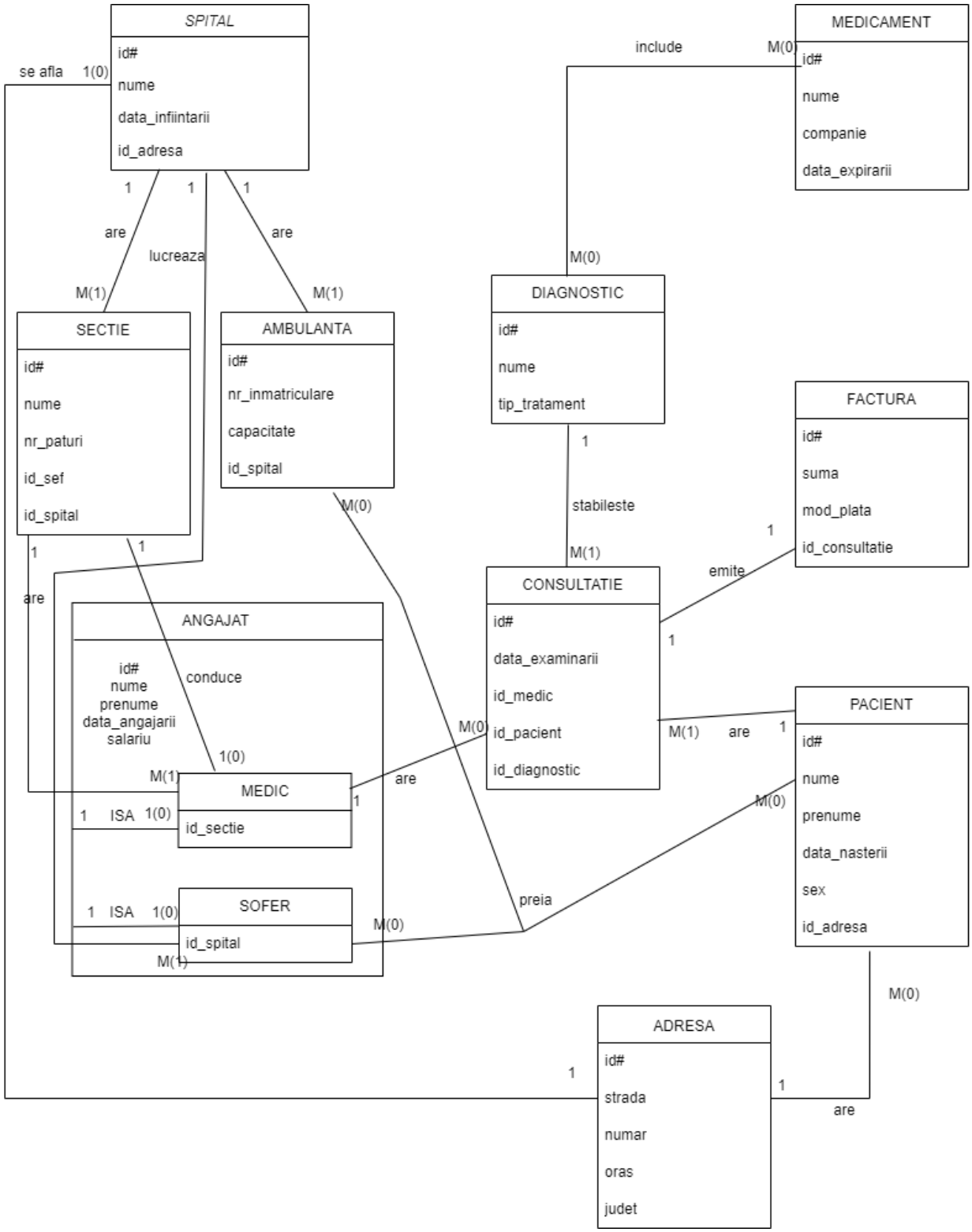
Entitatea DIAGNOSTIC are ca atribute:
id = variabilă de tip întreg, de lungime maximă 4, care reprezintă id-ul diagnosticului
nume = variabilă de tip șir de caractere, de lungime maximă 25, care reprezintă numele diagnosticului, și care nu poate fi nulă
tip_tratament = variabilă de tip șir de caractere, de lungime maximă 25, care reprezintă tipul tratamentului pentru diagnostic. De exemplu, poate să fie medicamentos, exerciții fizice, terapie, dieta sau alte valori.
Entitatea RETETA are ca atribute:
id_diagnostic = variabilă de tip întreg, de lungime maximă 4, care reprezintă id-ul diagnosticului pentru care este prescrisă rețeta. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul DIAGNOSTIC
id_medicament = variabilă de tip întreg, de lungime maximă 4, care reprezintă id-ul medicamentului prescris. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul MEDICAMENT
doza = variabilă de tip șir de caractere, de lungime maximă 20, care reprezintă doza prescrisă pentru medicamentul respectiv
durata = variabilă de tip șir de caractere, de lungime maximă 20, care reprezintă cat timp trebuie să se administreze medicamentul

Entitatea CONSULTATIE are ca atribute:
id = variabilă de tip întreg, de lungime maximă 4, care reprezintă id-ul consultației
data_examinarii = variabilă de tip dată calendaristică, reprezintă data la care are loc consultația între medic și pacient, are ca valoare default ziua curentă, în cazul în care ea nu este specificată

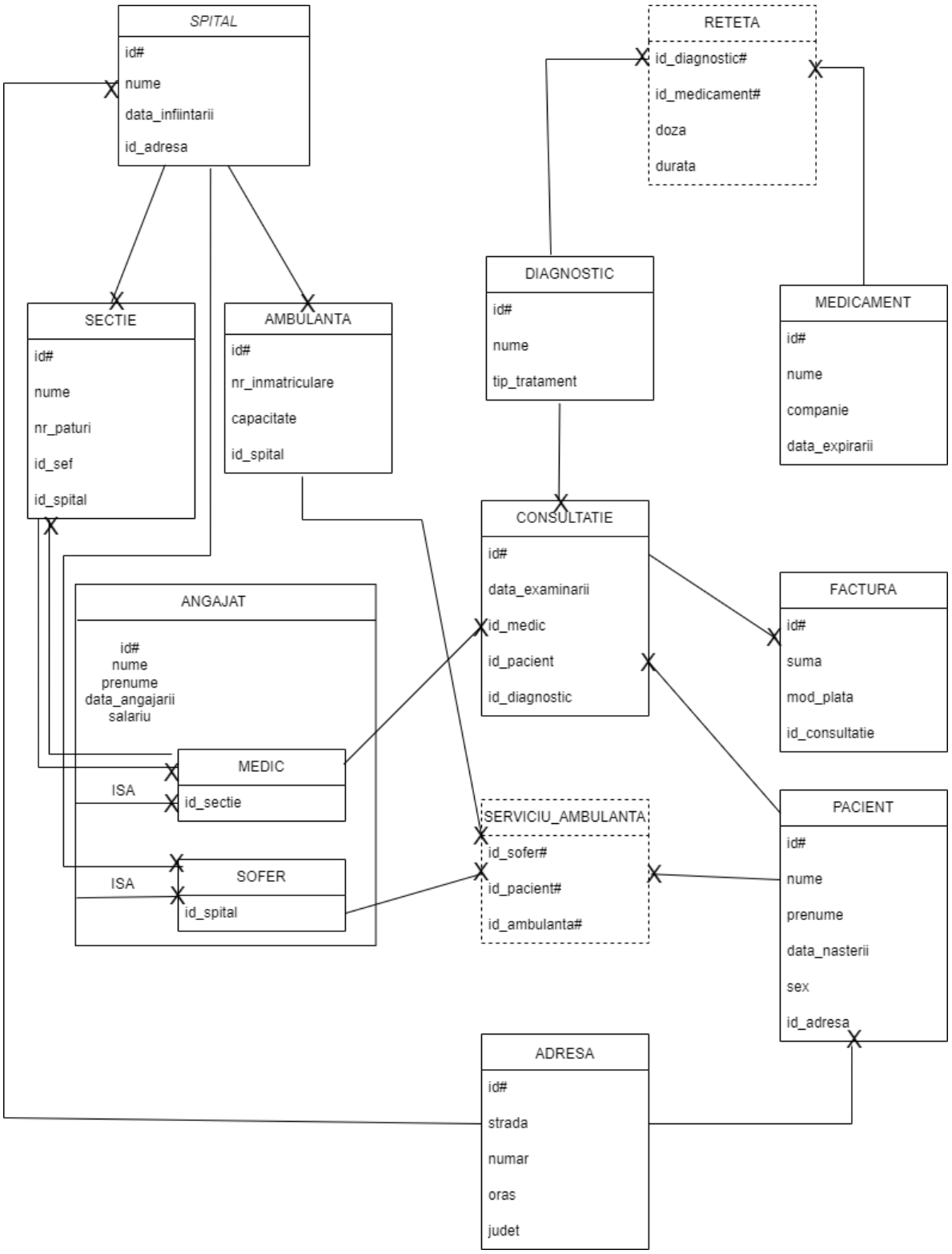
id_medic = variabilă de tip întreg, de lungime maximă 4, care reprezintă id-ul medicului care a consultat pacientul. Atributul nu poate fi null, trebuie să corespundă la o valoare a cheii primare din tabelul MEDIC.
id_pacient = variabilă de tip întreg, de lungime maximă 3, care reprezintă id-ul pacientului evaluat în cadrul consultației. Atributul nu poate fi null, trebuie să corespundă la o valoare a cheii primare din tabelul PACIENT.
id_diagnostic = variabilă de tip întreg, de lungime maximă 4, care reprezintă id-ul diagnosticului dat pacientului în urma consultației . Atributul nu poate fi null, trebuie să corespundă la o valoare a cheii primare din tabelul DIAGNOSTIC

Entitatea FACTURA are ca atribute:
id = variabilă de tip întreg, de lungime maximă 5, care reprezintă id-ul facturii
suma = variabilă de tip întreg, de lungime maximă 6, care reprezintă suma facturii. Atributul nu poate fi null.
mod_plata = variabilă de tip șir de caractere, de lungime maximă 5, care reprezintă metoda de plata a facturii (cash, card)
id_consultatie = variabilă de tip întreg, de lungime maximă 4, care reprezintă id-ul consultației în urma căreia s-a emis factura . Atributul nu poate fi null, trebuie să corespundă la o valoare a cheii primare din tabelul DIAGNOSTIC

6. Diagrama entitate - relație



7. Diagrama conceptuală



8. Scheme relaționale

SPITAL (id#, nume, data_infintarii, id_adresa)
ADRESA (id#, strada, numar, oras, judet)
SECTIE (id#, nume, nr_paturi, id_sef, id_spital)
AMBULANTA (id#, nr_inmatriculare, capacitate, id_spital)
ANGAJAT (id#, nume, prenume, data_angajarii, salariu)
MEDIC (id#, nume, prenume, data_angajarii, salariu, id_sectie)
SOFER (id#, nume, prenume, data_angajarii, salariu, id_spital)
SERVICIU_AMBULANTA (id_sofer#, id_pacient#, id_ambulanta#, motiv)
PACIENT (id#, nume, prenume, data_nasterii, sex, id_adresa)
CONSULTATIE (id#, data_examinarii, id_medic, id_pacient, id_diagnostic)
DIAGNOSTIC (id#, nume, tip_tratament)
RETETA (id_diagnostic#, id_medicament#, doza, durata)
MEDICAMENT (id#, nume, companie, data_expirarii)
FACTURA (id#, suma, mod_plata, id_consultatie)

9. Realizarea normalizarii

Forma normala 1
O relație R este în prima formă normală dacă fiecărui atribut care o compune îi corespunde o valoare atomica.

Schema NON-FN1:
PACIENT (id**#**, nume, prenume, data_nasterii, sex, **adresa**) Schema noastră ar fi scoasă din FN1 dacă în loc să existe două entități ADRESA și PACIENT ar exista doar entitatea PACIENT, deoarece adresa este compusă din mai multe valori (stradă, numar, județ, oraș).

Forma normala 2
O relație R este în a doua formă normală dacă și numai dacă relația R este in FN1 și fiecare atribut care nu este cheie (nu participa la cheia primara) este dependent de întreaga cheie primară.

Schema NON-FN2:
MEDIC(id#, specializare, id_sectie)
Atributul specializare prezintă o dependență funcțională cheia parțială “id_sectie”. Dependența funcțională determinată de cheia parțială "id_sectie" este eliminată prin ștergerea coloanei specializare din tabela "Medici".

Forma normala 3
O relaţie R este în a treia formă normală dacă și numai dacă: relația R este în FN2 și fiecare atribut care nu este cheie (nu participă la cheia primară) depinde direct de cheia primară.

Schema NON-FN3:
{id_angajat#} -> {nume, prenume, salariu, sex, id_spital} - id_angajat determina functional atributele selectate - {id_angajat#} -> {id_spital} -> {data_infiintare, nume, id_adresa}
Pentru a eliminara dependențele uncționale tranzitive, avem proiectiile:
ANGAJAT(cod_angajat#, nume, prenume, salariu, sex, id_spital)
SPITAL(id#, data_infiintare, nume, id_adresa)

10. Crearea unei secvențe utilizată în inserarea înregistrărilor în tabele

```
create sequence seq_med
start with 1400
maxvalue 9990
nocycle
nocache;
```

create sequence seq_med
start with 1400
maxvalue 9999
nocycle
nocache;

11. Crearea tabelelor si inserarea datelor

Crearea tabelelor:

```
--ADRESE--

create table adrese (
    id number(4) constraint pk_adr primary key,
    strada varchar2(25),
    numar number(3),
    oras varchar(15) not null,
    judet varchar(15)
);
```

--ADRESE--

create table adrese (
id number(4) constraint pk_adr primary key,
strada varchar2(25),
numar number(3),
oras varchar(15) not null,
judet varchar(15)
);

```
--PACIENTI--

create table pacienti (
    id number(3) constraint pk_pac primary key,
    nume varchar2(10) not null,
    prenume varchar2(10) not null,
    data_nasterii date,
    sex varchar2(1),
    id_adresa number(4) not null,
    constraint fk_pac foreign key(id_adresa) references adrese
);
```

--PACIENTI--

```
create table pacienti (
   id number(3) constraint pk_pac primary key,
   nume varchar2(10) not null,
   prenume varchar2(10) not null,
   data_nasterii date,
   sex varchar2(1),
   id_adresa number(4) not null,
   constraint fk_pac foreign key(id_adresa) references adrese
);
```

```
--SPITALE--

create table spitale (
    id number(3) primary key,
    nume varchar2(40) not null,
    data_infintarii date,
    id_adresa number(4) unique,
    constraint fk_sp foreign key(id_adresa) references adrese
);
```

--SPITALE--

```
create table spitale (
   id number(3) primary key,
   nume varchar2(40) not null,
   data_infintarii date,
   id_adresa number(4) unique,
   constraint fk_sp foreign key(id_adresa) references adrese
);
```

```sql
--SECTII--

create table sectii (
    id number(4) primary key,
    nume varchar2(40) not null,
    nr_paturi number(3),
    id_sef number(3) unique,
    id_spital number(3),
    foreign key(id_spital) references spitale(id)
);
```

--SECTII--

create table sectii (
id number(4) primary key,
nume varchar2(40) not null,
nr_paturi number(3),
id_sef number(3) unique,
id_spital number(3),
foreign key(id_spital) references spitale(id)
);

```sql
--ANGAJATI--
--un angajat poate fi de un singur tip: fie medic, fie sofer

--pentru a obtine asta, creez o tabela care contine diferitele tipuri de angajati
--pe care sistemul le va permite
create table tip_ang (
    id varchar(5) primary key
);
insert into tip_ang values ('Medic');
insert into tip_ang values ('Sofer');
--acum voi adauga in tabela angajati coloana id_tip_ang;
--fiecarui angajat trebuie sa ii fie asignat un tip => not null;
--adaug constrangere unique(id, id_tip_ang) ca sa pot seta o cheie straina
--in subtabele care sa referentieze acest tuplu;
create table angajati (
    id number(4) primary key,
    id_tip_ang varchar(5) references tip_ang(id) not null,
    nume varchar2(8) not null,
    prenume varchar2(12) not null,
    data_angajarii date default sysdate,
    salariu number(8,2),
    constraint ang_min_sal check (salariu > 0),
    constraint ang_alt_pk unique (id, id_tip_ang)
);
```

--ANGAJATI–

--un angajat poate fi de un singur tip: fie medic, fie sofer

--pentru a obtine asta, creez o tabela care contine diferitele tipuri de angajati
--pe care sistemul le va permite

create table tip_ang (
  id varchar(5) primary key
);
insert into tip_ang values ('Medic');
insert into tip_ang values ('Sofer');

--acum voi adauga in tabela angajati coloana id_tip_ang;
--fiecarui angajat trebuie sa ii fie asignat un tip => not null;
--adaug constrangere unique(id, id_tip_ang) ca sa pot seta o cheie straina
--in subtabele care sa referentieze acest tuplu;

```
create table angajati (
   id number(4) primary key,
   id_tip_ang varchar(5) references tip_ang(id) not null,
   nume varchar2(8) not null,
   prenume varchar2(12) not null,
   data_angajarii date default sysdate,
   salariu number(8,2),
   constraint ang_min_sal check (salariu > 0),
   constraint ang_alt_pk unique (id, id_tip_ang)
);
--subtabelele vor avea o cheie straina care sa referentieze angajatul
--dar in loc sa referentieze doar angajat(id) va referentia tuplul (id, id_tip_ang)
--in felul acesta e garantat ca angajatul va fi de un singur tip
--si ca pot avea date doar in subtabela corespunzatoare pt acel tip

--MEDICI--

create table medici (
    id number(4) primary key,
    id_tip_ang varchar(5) default 'Medic' check (id_tip_ang = 'Medic'),
    tip varchar2(10),
    id_sectie number(4),
    foreign key(id_sectie) references sectii(id),
    foreign key(id, id_tip_ang) references angajati(id, id_tip_ang)
);
```

--subtabelele vor avea o cheie straina care sa referentieze angajatul
--dar in loc sa referentieze doar angajat(id) va referentia tuplul (id, id_tip_ang)
--in felul acesta e garantat ca angajatul va fi de un singur tip
--si ca pot avea date doar in subtabela corespunzatoare pt acel tip

--MEDICI--

```
create table medici (
   id number(4) primary key,
   id_tip_ang varchar(5) default 'Medic' check (id_tip_ang = 'Medic'),
   tip varchar2(10),
   id_sectie number(4),
   foreign key(id_sectie) references sectii(id),
   foreign key(id, id_tip_ang) references angajati(id, id_tip_ang)
);
```

```sql
--acum ca am creat tabela medici, care la randul ei referentia tabela sectii
--pot altera tabela sectii sa aiba cheie straina catre id-ul unui medic,
--reprezentand seful pe sectia respectiva
alter table sectii add (constraint sect_sef_fk foreign key (id_sef) references medici(id));
alter table sectii drop constraint sect_sef_fk;

--SOFERI--

create table soferi (
    id number(4) primary key,
    id_tip_ang varchar(5) default 'Sofer' check (id_tip_ang = 'Sofer'),
    id_spital number(3),
    foreign key(id_spital) references spitale(id),
    foreign key(id, id_tip_ang) references angajati(id, id_tip_ang)
);
```

--acum ca am creat tabela medici, care la randul ei referentia tabela sectii
--pot altera tabela sectii sa aiba cheie straina catre id-ul unui medic,
--reprezentand seful pe sectia respectiva
alter table sectii add (constraint sect_sef_fk foreign key (id_sef) references medici(id));
alter table sectii drop constraint sect_sef_fk;

--SOFERI--

create table soferi (
id number(4) primary key,
id_tip_ang varchar(5) default 'Sofer' check (id_tip_ang = 'Sofer'),
id_spital number(3),
foreign key(id_spital) references spitale(id),
foreign key(id, id_tip_ang) references angajati(id, id_tip_ang)
);

```sql
create view medici_complet as
select a.*, m.tip, m.id_sectie, s.nume as nume_sectie
from angajati a inner join medici m
on a.id = m.id
inner join sectii s
on m.id_sectie = s.id;

--select * from medici_complet;

create view soferi_complet as
select angajati.*, id_spital, spitale.nume as nume_spital
from angajati inner join soferi
on angajati.id = soferi.id
inner join spitale
on soferi.id_spital = spitale.id;
```

create view medici_complet as
select a.*, m.tip, m.id_sectie, s.nume as nume_sectie
from angajati a inner join medici m
on a.id = m.id
inner join sectii s
on m.id_sectie = s.id;

create view soferi_complet as
select angajati.*, id_spital, spitale.nume as nume_spital
from angajati inner join soferi
on angajati.id = soferi.id
inner join spitale
on soferi.id_spital = spitale.id;

```sql
--AMBULANTE--

create table ambulante (
    id number(3) primary key,
    nr_inmatriculare varchar2(8) not null,
    capacitate number(1),
    id_spital number(3),
    foreign key(id_spital) references spitale(id)
);
```

--AMBULANTE--

```
create table ambulante (
   id number(3) primary key,
   nr_inmatriculare varchar2(8) not null,
   capacitate number(1),
   id_spital number(3),
   foreign key(id_spital) references spitale(id)
);
```

```
--SERVICII_AMBULANTA--

create table servicii_amb (
    id_sofer number(4),
    id_pacient number(3),
    id_ambulanta number(3),
    data_serviciu date default sysdate,
    motiv varchar2(30) not null,
    constraint serv_amb_pk primary key(id_sofer, id_pacient, id_ambulanta),
    constraint serv_amb_fk_sof foreign key(id_sofer) references soferi(id),
    constraint serv_amb_fk_pac foreign key(id_pacient) references pacienti(id),
    constraint serv_amb_fk_amb foreign key(id_ambulanta) references ambulante(id)
);
```

--SERVICII_AMBULANTA--

```
create table servicii_amb (
   id_sofer number(4),
   id_pacient number(3),
   id_ambulanta number(3),
   data_serviciu date default sysdate,
   motiv varchar2(30) not null,
   constraint serv_amb_pk primary key(id_sofer, id_pacient, id_ambulanta),
   constraint serv_amb_fk_sof foreign key(id_sofer) references soferi(id),
   constraint serv_amb_fk_pac foreign key(id_pacient) references pacienti(id),
   constraint serv_amb_fk_amb foreign key(id_ambulanta) references ambulante(id)
);
```

```sql
--MEDICAMENTE--

create table medicamente (
    id number(4) primary key,
    nume varchar(20),
    companie varchar(20),
    data_expirarii date
);
alter table medicamente
modify nume not null;
```

--MEDICAMENTE--

create table medicamente (
id number(4) primary key,
nume varchar(20),
companie varchar(20),
data_expirarii date
);
alter table medicamente
modify nume not null;

```sql
--DIAGNOSTICE--

create table diagnostice (
    id number(4) primary key,
    nume varchar2(25) not null,
    tip_tratament varchar2(25),
    constraint uc_diag_nume_tip UNIQUE (nume, tip_tratament)
);
```

--DIAGNOSTICE--

create table diagnostice (
id number(4) primary key,
nume varchar2(25) not null,
tip_tratament varchar2(25),
constraint uc_diag_nume_tip UNIQUE (nume, tip_tratament)
);

```
--RETETE--

create table retete (
    id_diagnostic number(4),
    id_medicament number(4),
    doza varchar2(20),
    durata varchar2(20),
    constraint ret_pk primary key(id_diagnostic, id_medicament),
    constraint ret_fk_diag foreign key(id_diagnostic) references diagnostice(id),
    constraint ret_fk_mdm foreign key(id_medicament) references medicamente(id)
);
```

--RETETE--

create table retete (
  id_diagnostic number(4),
  id_medicament number(4),
  doza varchar2(20),
  durata varchar2(20),
  constraint ret_pk primary key(id_diagnostic, id_medicament),
  constraint ret_fk_diag foreign key(id_diagnostic) references diagnostice(id),
  constraint ret_fk_mdm foreign key(id_medicament) references medicamente(id)
);

```
--CONSULTATII--

create table consultatii (
    id number(4) primary key,
    data_examinarii date default sysdate,
    id_medic number(4) not null,
    id_pacient number(3) not null,
    id_diagnostic number(4) not null,
    constraint cons_fk_med foreign key(id_medic) references medici(id),
    constraint cons_fk_pac foreign key(id_pacient) references pacienti(id),
    constraint cons_fk_diag foreign key(id_diagnostic) references diagnostice(id)
);
```

--CONSULTATII--

create table consultatii (
  id number(4) primary key,
  data_examinarii date default sysdate,
  id_medic number(4) not null,
  id_pacient number(3) not null,

```
    id_diagnostic number(4) not null,
    constraint cons_fk_med foreign key(id_medic) references medici(id),
    constraint cons_fk_pac foreign key(id_pacient) references pacienti(id),
    constraint cons_fk_diag foreign key(id_diagnostic) references diagnostice(id)
);
```

```
--FACTURI--

create table facturi (
    id number(5) primary key,
    suma number(6) not null,
    mod_plata varchar2(5),
    id_consultatie number(4) not null,
    constraint fact_fk_cons foreign key(id_consultatie) references consultatii(id),
    constraint fact_uniq_cons unique(id_consultatie)
);
```

--FACTURI--

create table facturi (
    id number(5) primary key,
    suma number(6) not null,
    mod_plata varchar2(5),
    id_consultatie number(4) not null,
    constraint fact_fk_cons foreign key(id_consultatie) references consultatii(id),
    constraint fact_uniq_cons unique(id_consultatie)
);

Inserarea datelor:

—INSERARE ADRESE–

```
create sequence seq_adr
start with 1200
maxvalue 9999
nocycle
nocache;

insert into adrese values (seq_adr.nextval, 'Str. Varsatorului', 2, 'Magurele', 'Ilfov');
```

```
insert into adrese values (seq_adr.nextval, 'Calea Floreasca', 8, 'Sector 1', 'Bucuresti');
insert into adrese values (seq_adr.nextval, 'Bdul Independentei', 1, 'Iasi', 'Iasi');
insert into adrese values (seq_adr.nextval, 'Bdul Corneliu Coposu', 24, 'Sibiu', 'Sibiu');
insert into adrese values (seq_adr.nextval, 'Bdul 1 Decembrie 1918', 21, 'Suceava',
'Suceava');
insert into adrese values (seq_adr.nextval, 'Str. Dornei', 114, 'Sector 1', 'Bucuresti');
insert into adrese values (seq_adr.nextval, 'Str. Rasadnitei', 8, 'Sector 6', 'Bucuresti');
insert into adrese values (seq_adr.nextval, 'Bd. Ion Mihalache', 37, 'Sector 1', 'Bucuresti');
insert into adrese values (seq_adr.nextval, 'Strada Victoriei', 10, 'Bucure?ti', 'Ilfov');
insert into adrese values (seq_adr.nextval, 'Strada Libertă?ii', 15, 'Cluj-Napoca', 'Cluj');
insert into adrese values (seq_adr.nextval, 'Strada Unirii', 5, 'Timi?oara', 'Timi?');
insert into adrese values (seq_adr.nextval, 'Bulevardul Revolu?iei', 7, 'Ia?i', 'Ia?i');
insert into adrese values (seq_adr.nextval, 'Strada Mihai Viteazu', 23, 'Craiova', 'Dolj');
insert into adrese values (seq_adr.nextval, 'Strada Eroilor', 12, 'Constan?a', 'Constan?a');
insert into adrese values (seq_adr.nextval, 'Bulevardul Decebal', 18, 'Bra?ov', 'Bra?ov');
insert into adrese values (seq_adr.nextval, 'Strada Dimitrie Cantemir', 9, 'Ploie?ti',
'Prahova');
insert into adrese values (seq_adr.nextval, 'Aleea Gheorghe Lazăr', 4, 'Oradea', 'Bihor');
insert into adrese values (seq_adr.nextval, 'Bdl Tudor Vladimirescu', 14, 'Arad', 'Arad');
insert into adrese values (seq_adr.nextval, 'Strada Avram Iancu', 20, 'Târgu Mure?',
'Mure?');
insert into adrese values (seq_adr.nextval, 'Strada Vasile Alecsandri', 8, 'Sibiu', 'Sibiu');
insert into adrese values (seq_adr.nextval, 'Bulevardul Carol I', 17, 'Gala?i', 'Gala?i');
insert into adrese values (seq_adr.nextval, 'Strada Nicolae Bălcescu', 3, 'Boto?ani',
'Boto?ani');
insert into adrese values (seq_adr.nextval, 'Strada 1 Decembrie', 11, 'Bacău', 'Bacău');
insert into adrese values (seq_adr.nextval, 'Bdl Alexandru Ioan Cuza', 22, 'Pite?ti', 'Arges');
insert into adrese values (seq_adr.nextval, 'Strada ?tefan cel Mare', 6, 'Brăila', 'Brăila');
insert into adrese values (seq_adr.nextval, 'Strada Vasile Pârvan', 13, 'Rm Vâlcea',
'Vâlcea');
insert into adrese values (seq_adr.nextval, 'Bulevardul Iuliu Maniu', 19, 'Suceava',
'Suceava');
insert into adrese values (seq_adr.nextval, 'Strada Mihai Eminescu', 15, 'Ia?i', 'Ia?i');
insert into adrese values (seq_adr.nextval, 'Strada Avram Iancu', 7, 'Cluj-Napoca', 'Cluj');
insert into adrese values (seq_adr.nextval, 'Strada Tudor Vladimirescu', 20, 'Timi?oara',
'Timi?');
insert into adrese values (seq_adr.nextval, 'Strada 1 Decembrie', 5, 'Craiova', 'Dolj');
insert into adrese values (seq_adr.nextval, 'Strada Unirii', 12, 'Oradea', 'Bihor');
```

```
select * from adrese;
```

Script Output | Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 32 in 0,03 seconds

	ID	STRADA	NUMAR	ORAS	JUDET
1	1200	Str. Varsatorului	2	Magurele	Ilfov
2	1201	Calea Floreasca	8	Sector 1	Bucuresti
3	1202	Bdul Independentei	1	Iasi	Iasi
4	1203	Bdul Corneliu Coposu	24	Sibiu	Sibiu
5	1204	Bdul 1 Decembrie 1918	21	Suceava	Suceava
6	1205	Str. Dornei	114	Sector 1	Bucuresti
7	1206	Str. Rasadnitei	8	Sector 6	Bucuresti
8	1207	Bd. Ion Mihalache	37	Sector 1	Bucuresti
9	1208	Strada Victoriei	10	București	Ilfov
10	1209	Strada Libertății	15	Cluj-Napoca	Cluj
11	1210	Strada Unirii	5	Timișoara	Timiș
12	1211	Bulevardul Revoluției	7	Iași	Iași
13	1212	Strada Mihai Viteazu	23	Craiova	Dolj
14	1213	Strada Eroilor	12	Constanța	Constanța
15	1214	Bulevardul Decebal	18	Brașov	Brașov
16	1215	Strada Dimitrie Cantemir	9	Ploiești	Prahova
17	1216	Aleea Gheorghe Lazăr	4	Oradea	Bihor
18	1217	Bdl Tudor Vladimirescu	14	Arad	Arad
19	1218	Strada Avram Iancu	20	Târgu Mureș	Mureș
20	1219	Strada Vasile Alecsandri	8	Sibiu	Sibiu
21	1220	Bulevardul Carol I	17	Galați	Galați
22	1221	Strada Nicolae Bălcescu	3	Botoșani	Botoșani
23	1222	Strada 1 Decembrie	11	Bacău	Bacău
24	1223	Bdl Alexandru Ioan Cuza	22	Pitești	Arges
25	1224	Strada Ștefan cel Mare	6	Brăila	Brăila
26	1225	Strada Vasile Pârvan	13	Rm Vâlcea	Vâlcea
27	1226	Bulevardul Iuliu Maniu	19	Suceava	Suceava
28	1227	Strada Mihai Eminescu	15	Iași	Iași
29	1228	Strada Avram Iancu	7	Cluj-Napoca	Cluj
30	1229	Strada Tudor Vladimirescu	20	Timișoara	Timiș
31	1230	Strada 1 Decembrie	5	Craiova	Dolj
32	1231	Strada Unirii	12	Oradea	Bihor

-- INSERARE PACIENTI –

```
create  sequence seq_pac
start with 100
maxvalue 999
nocycle
nocache;
```

```
insert into pacienti values (seq_pac.nextval, 'Popescu', 'Ana', to_date('12-05-1990',
'DD-MM-YYYY'), 'F', 1203);
insert into pacienti values (seq_pac.nextval, 'Ionescu', 'Mihai', to_date('21-08-1985',
'DD-MM-YYYY'), 'M', 1209);
insert into pacienti values (seq_pac.nextval, 'Apostol', 'Maria', to_date('03-11-1992',
'DD-MM-YYYY'), 'F', 1210);
insert into pacienti values (seq_pac.nextval, 'Radu', 'Ion', to_date('17-02-1978',
'DD-MM-YYYY'), 'M', 1222);
insert into pacienti values (seq_pac.nextval, 'Dumitru', 'Elena', to_date('25-06-1982',
'DD-MM-YYYY'), 'F', 1202);
insert into pacienti values (seq_pac.nextval, 'Stancu', 'Andrei', to_date('08-03-1996',
'DD-MM-YYYY'), 'M', 1212);
insert into pacienti values (seq_pac.nextval, 'Constantin', 'Laura', to_date('30-09-1994',
'DD-MM-YYYY'), 'F', 1220);
insert into pacienti values (seq_pac.nextval, 'Florescu', 'Alexandru', to_date('14-07-1989',
'DD-MM-YYYY'), 'M', 1208);
insert into pacienti values (seq_pac.nextval, 'Gheorghe', 'Ioana', to_date('19-04-1991',
'DD-MM-YYYY'), 'F', 1227);
insert into pacienti values (seq_pac.nextval, 'Stan', 'Daniel', to_date('07-12-1980',
'DD-MM-YYYY'), 'M', 1200);
insert into pacienti values (seq_pac.nextval, 'Vasilescu', 'Cristina', to_date('29-01-1987',
'DD-MM-YYYY'), 'F', 1205);
insert into pacienti values (seq_pac.nextval, 'Marin', 'Adrian', to_date('11-10-1984',
'DD-MM-YYYY'), 'M', 1222);
insert into pacienti values (seq_pac.nextval, 'Dobre', 'Georgiana', to_date('02-02-1993',
'DD-MM-YYYY'), 'F', 1201);
insert into pacienti values (seq_pac.nextval, 'Munteanu', 'Sorin', to_date('15-11-1979',
'DD-MM-YYYY'), 'M', 1202);
insert into pacienti values (seq_pac.nextval, 'Neagu', 'Andreea', to_date('28-07-1995',
'DD-MM-YYYY'), 'F', 1203);
insert into pacienti values (seq_pac.nextval, 'Diaconu', 'Dumitru', to_date('04-04-1986',
'DD-MM-YYYY'), 'M', 1210);
insert into pacienti values (seq_pac.nextval, 'Georgescu', 'Elena', to_date('22-01-1981',
'DD-MM-YYYY'), 'F', 1209);
insert into pacienti values (seq_pac.nextval, 'Stoica', 'Marius', to_date('05-09-1997',
'DD-MM-YYYY'), 'M', 1207);
insert into pacienti values (seq_pac.nextval, 'Pavel', 'Andreea', to_date('11-06-1990',
'DD-MM-YYYY'), 'F', 1206);
insert into pacienti values (seq_pac.nextval, 'Gavril', 'Iulia', to_date('27-03-1983',
'DD-MM-YYYY'), 'F', 1204);
insert into pacienti values (seq_pac.nextval, 'Voinea', 'Ionut', to_date('09-12-1977',
'DD-MM-YYYY'), 'M', 1205);
insert into pacienti values (seq_pac.nextval, 'Radulescu', 'Simona', to_date('17-10-1992',
'DD-MM-YYYY'), 'F', 1208);
insert into pacienti values (seq_pac.nextval, 'Barbu', 'Gheorghe', to_date('13-02-1988',
'DD-MM-YYYY'), 'M', 1211);
insert into pacienti values (seq_pac.nextval, 'Iordache', 'Catalina', to_date('26-09-1980',
'DD-MM-YYYY'), 'F', 1213);
```

```
insert into pacienti values (seq_pac.nextval, 'Grosu', 'Adrian', to_date('02-07-1987',
'DD-MM-YYYY'), 'M', 1212);
insert into pacienti values (seq_pac.nextval, 'Nistor', 'Elena', to_date('16-03-1993',
'DD-MM-YYYY'), 'F', 1215);
insert into pacienti values (seq_pac.nextval, 'Iancu', 'Mihai', to_date('30-12-1984',
'DD-MM-YYYY'), 'M', 1214);
insert into pacienti values (seq_pac.nextval, 'Matei', 'Ana', to_date('19-01-1989',
'DD-MM-YYYY'), 'F', 1211);
insert into pacienti values (seq_pac.nextval, 'Florea', 'Alexandru', to_date('04-10-1978',
'DD-MM-YYYY'), 'M', 1217);
insert into pacienti values (seq_pac.nextval, 'Dinu', 'Maria', to_date('15-06-1982',
'DD-MM-YYYY'), 'F', 1219);
insert into pacienti values (seq_pac.nextval, 'Stefan', 'Vasile', to_date('29-03-1996',
'DD-MM-YYYY'), 'M', 1218);
insert into pacienti values (seq_pac.nextval, 'Sandu', 'Andreea', to_date('03-08-1994',
'DD-MM-YYYY'), 'F', 1210);
insert into pacienti values (seq_pac.nextval, 'Gheorghiu', 'Adrian', to_date('12-11-1989',
'DD-MM-YYYY'), 'M', 1216);
insert into pacienti values (seq_pac.nextval, 'Dumitrache', 'Ioana', to_date('23-02-1991',
'DD-MM-YYYY'), 'F', 1220);
insert into pacienti values (seq_pac.nextval, 'Mihai', 'Marian', to_date('06-09-1980',
'DD-MM-YYYY'), 'M', 1221);
insert into pacienti values (seq_pac.nextval, 'Niculae', 'Elena', to_date('20-07-1987',
'DD-MM-YYYY'), 'F', 1222);
insert into pacienti values (seq_pac.nextval, 'Cristea', 'Cristian', to_date('03-03-1995',
'DD-MM-YYYY'), 'M', 1223);
insert into pacienti values (seq_pac.nextval, 'Marinescu', 'Raluca', to_date('16-12-1986',
'DD-MM-YYYY'), 'F', 1224);
insert into pacienti values (seq_pac.nextval, 'Tudor', 'Andrei', to_date('28-10-1992',
'DD-MM-YYYY'), 'M', 1225);
insert into pacienti values (seq_pac.nextval, 'Dobre', 'Gabriela', to_date('12-02-1993',
'DD-MM-YYYY'), 'F', 1226);
insert into pacienti values (seq_pac.nextval, 'Stanciu', 'Mihai', to_date('25-09-1979',
'DD-MM-YYYY'), 'M', 1227);
insert into pacienti values (seq_pac.nextval, 'Pop', 'Cristina', to_date('07-06-1995',
'DD-MM-YYYY'), 'F', 1220);
insert into pacienti values (seq_pac.nextval, 'Gheorghe', 'Andrei', to_date('21-03-1983',
'DD-MM-YYYY'), 'M', 1200);
insert into pacienti values (seq_pac.nextval, 'Muntean', 'Elena', to_date('02-12-1981',
'DD-MM-YYYY'), 'F', 1201);
insert into pacienti values (seq_pac.nextval, 'Dinca', 'George', to_date('15-08-1997',
'DD-MM-YYYY'), 'M', 1231);
insert into pacienti values (seq_pac.nextval, 'Popescu', 'Roxana', to_date('28-05-1990',
'DD-MM-YYYY'), 'F', 1230);
insert into pacienti values (seq_pac.nextval, 'Ionescu', 'Mihai', to_date('08-08-1985',
'DD-MM-YYYY'), 'M', 1229);
insert into pacienti values (seq_pac.nextval, 'Apostol', 'Maria', to_date('19-11-1992',
'DD-MM-YYYY'), 'F', 1228);
```

```
insert into pacienti values (seq_pac.nextval, 'Radu', 'Ion', to_date('01-02-1978',
'DD-MM-YYYY'), 'M', 1228);
insert into pacienti values (seq_pac.nextval, 'Dumitru', 'Elena', to_date('13-06-1982',
'DD-MM-YYYY'), 'F', 1227);
insert into pacienti values (seq_pac.nextval, 'Stancu', 'Andrei', to_date('26-03-1996',
'DD-MM-YYYY'), 'M', 1226);
insert into pacienti values (seq_pac.nextval, 'Constantin', 'Laura', to_date('09-10-1994',
'DD-MM-YYYY'), 'F', 1225);
insert into pacienti values (seq_pac.nextval, 'Florescu', 'Alexandru', to_date('23-07-1989',
'DD-MM-YYYY'), 'M', 1224);
insert into pacienti values (seq_pac.nextval, 'Gheorghe', 'Ioana', to_date('05-05-1991',
'DD-MM-YYYY'), 'F', 1223);
insert into pacienti values (seq_pac.nextval, 'Stan', 'Daniel', to_date('18-12-1980',
'DD-MM-YYYY'), 'M', 1222);
insert into pacienti values (seq_pac.nextval, 'Vasilescu', 'Cristina', to_date('30-01-1987',
'DD-MM-YYYY'), 'F', 1221);
insert into pacienti values (seq_pac.nextval, 'Marin', 'Adrian', to_date('11-10-1984',
'DD-MM-YYYY'), 'M', 1220);
insert into pacienti values (seq_pac.nextval, 'Dobre', 'Georgiana', to_date('24-02-1993',
'DD-MM-YYYY'), 'F', 1219);
insert into pacienti values (seq_pac.nextval, 'Munteanu', 'Sorin', to_date('08-12-1979',
'DD-MM-YYYY'), 'M', 1218);
insert into pacienti values (seq_pac.nextval, 'Neagu', 'Andreea', to_date('21-07-1995',
'DD-MM-YYYY'), 'F', 1217);
insert into pacienti values (seq_pac.nextval, 'Diaconu', 'Dumitru', to_date('04-04-1986',
'DD-MM-YYYY'), 'M', 1216);
insert into pacienti values (seq_pac.nextval, 'Georgescu', 'Elena', to_date('17-01-1981',
'DD-MM-YYYY'), 'F', 1215);
insert into pacienti values (seq_pac.nextval, 'Stoica', 'Marius', to_date('30-08-1997',
'DD-MM-YYYY'), 'M', 1214);
insert into pacienti values (seq_pac.nextval, 'Pavel', 'Andreea', to_date('13-06-1990',
'DD-MM-YYYY'), 'F', 1213);
insert into pacienti values (seq_pac.nextval, 'Gavril', 'Iulia', to_date('27-03-1983',
'DD-MM-YYYY'), 'F', 1212);
insert into pacienti values (seq_pac.nextval, 'Voinea', 'Ionut', to_date('09-12-1977',
'DD-MM-YYYY'), 'M', 1211);
insert into pacienti values (seq_pac.nextval, 'Radulescu', 'Simona', to_date('22-10-1992',
'DD-MM-YYYY'), 'F', 1210);
insert into pacienti values (seq_pac.nextval, 'Barbu', 'Gheorghe', to_date('06-02-1988',
'DD-MM-YYYY'), 'M', 1209);
insert into pacienti values (seq_pac.nextval, 'Iordache', 'Catalina', to_date('19-09-1980',
'DD-MM-YYYY'), 'F', 1208);
insert into pacienti values (seq_pac.nextval, 'Grosu', 'Adrian', to_date('03-07-1987',
'DD-MM-YYYY'), 'M', 1207);
insert into pacienti values (seq_pac.nextval, 'Nistor', 'Elena', to_date('17-03-1993',
'DD-MM-YYYY'), 'F', 1206);
insert into pacienti values (seq_pac.nextval, 'Iancu', 'Mihai', to_date('01-01-1985',
'DD-MM-YYYY'), 'M', 1205);
```

```
insert into pacienti values (seq_pac.nextval, 'Matei', 'Ana', to_date('21-01-1989',
'DD-MM-YYYY'), 'F', 1204);
insert into pacienti values (seq_pac.nextval, 'Florea', 'Alexandru', to_date('05-10-1978',
'DD-MM-YYYY'), 'M', 1203);
insert into pacienti values (seq_pac.nextval, 'Dinu', 'Maria', to_date('16-06-1982',
'DD-MM-YYYY'), 'F', 1202);
insert into pacienti values (seq_pac.nextval, 'Stefan', 'Vasile', to_date('29-03-1996',
'DD-MM-YYYY'), 'M', 1201);
```

insert into pacienti values (seq_pac.nextval, 'Sandu', 'Andreea', to_date('12-08-1994', 'DD-MM-YYYY'), 'F', 1200);

```
select * from pacienti;
```

ript Output | Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 71 in 0,157 seconds

	ID	NUME	PRENUME	DATA_NASTERII	SEX	ID_ADRESA
1	100	Popescu	Ana	12-05-1990	F	1203
2	101	Ionescu	Mihai	21-08-1985	M	1209
3	102	Apostol	Maria	03-11-1992	F	1210
4	103	Radu	Ion	17-02-1978	M	1222
5	104	Dumitru	Elena	25-06-1982	F	1202
6	105	Stancu	Andrei	08-03-1996	M	1212
7	106	Constantin	Laura	30-09-1994	F	1220
8	107	Florescu	Alexandru	14-07-1989	M	1208
9	108	Gheorghe	Ioana	19-04-1991	F	1227
10	109	Stan	Daniel	07-12-1980	M	1200
11	110	Vasilescu	Cristina	29-01-1987	F	1205
12	111	Marin	Adrian	11-10-1984	M	1222
13	112	Dobre	Georgiana	02-02-1993	F	1201
14	113	Munteanu	Sorin	15-11-1979	M	1202
15	114	Neagu	Andreea	28-07-1995	F	1203
16	115	Diaconu	Dumitru	04-04-1986	M	1210
17	116	Georgescu	Elena	22-01-1981	F	1209
18	117	Stoica	Marius	05-09-1997	M	1207
19	118	Pavel	Andreea	11-06-1990	F	1206
20	119	Gavril	Iulia	27-03-1983	F	1204
21	120	Voinea	Ionut	09-12-1977	M	1205
22	121	Radulescu	Simona	17-10-1992	F	1208
23	122	Barbu	Gheorghe	13-02-1988	M	1211
24	123	Iordache	Catalina	26-09-1980	F	1213
25	124	Grosu	Adrian	02-07-1987	M	1212
26	125	Nistor	Elena	16-03-1993	F	1215
27	126	Iancu	Mihai	30-12-1984	M	1214
28	127	Matei	Ana	19-01-1989	F	1211
29	128	Florea	Alexandru	04-10-1978	M	1217
30	129	Dinu	Maria	15-06-1982	F	1219
31	130	Stefan	Vasile	29-03-1996	M	1218
32	131	Sandu	Andreea	03-08-1994	F	1210
33	132	Gheorghiu	Adrian	12-11-1989	M	1216
34	133	Dumitrache	Ioana	23-02-1991	F	1220

-- INSERARE SPITALE –

```
create  sequence seq_sp
start with 200
increment by 1
maxvalue 999
nocycle
nocache;

insert into spitale values (seq_sp.nextval, 'SPITALUL JUD DE URGENTA SF. IOAN CEL NOU', to_date('08-IUN-1997'), 1204);
insert into spitale values (seq_sp.nextval, 'SPITALUL CLINIC JUD DE URGENTA SIBIU', to_date('20-IAN-1980'), 1203);
insert into spitale values (seq_sp.nextval, 'SPITALUL CLINIC DE URGENTA FLOREASCA',to_date('30-APR-2004'), 1201);
insert into spitale values (seq_sp.nextval, 'SPITALUL JUD DE URGENTA SF. SPIRIDON',to_date('02-MAI-2002'), 1202);
insert into spitale values (seq_sp.nextval, 'SPITALUL CLINIC SF. MARIA', to_date('13-OCT-1950'), 1207);
```

```
select * from spitale;
```

ipt Output | Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 5 in 0,124 seconds

	ID	NUME	DATA_INFINTARII	ID_ADRESA
1	200	SPITALUL JUD DE URGENTA SF. IOAN CEL NOU	08-06-1997	1204
2	201	SPITALUL CLINIC JUD DE URGENTA SIBIU	20-01-1980	1203
3	202	SPITALUL CLINIC DE URGENTA FLOREASCA	30-04-2004	1201
4	204	SPITALUL CLINIC SF. MARIA	13-10-1950	1207
5	205	SPITALUL JUD DE URGENTA SF. SPIRIDON	02-05-2002	1206

-- INSERARE SECTII –

```
create sequence seq_sec
start with 2000
nocycle
nocache;

insert into sectii values (seq_sec.nextval, 'Cardiologie', 20, null, 200);
insert into sectii values (seq_sec.nextval, 'Ortopedie', 15, null, 201);
insert into sectii values (seq_sec.nextval, 'Ginecologie', 25, null, 202);
insert into sectii values (seq_sec.nextval, 'Neurologie', 30, null, 200);
```

insert into sectii values (seq_sec.nextval, 'Pediatrie', 40, null, 204);
insert into sectii values (seq_sec.nextval, 'Endocrinologie', 18, null, 205);
insert into sectii values (seq_sec.nextval, 'Oftalmologie', 12, null, 201);
insert into sectii values (seq_sec.nextval, 'Dermatologie', 16, null, 202);
insert into sectii values (seq_sec.nextval, 'Urologie', 22, null, 200);

insert into sectii values (seq_sec.nextval, 'Oncologie', 10, null, 200);
insert into sectii values (seq_sec.nextval, 'Pneumologie', 20, null, 201);
insert into sectii values (seq_sec.nextval, 'Cardiologie', 16, null, 200);
insert into sectii values (seq_sec.nextval, 'Neurologie', 32, null, 200);

```
select *
from sectii;
```

	ID	NUME	NR_PATURI	ID_SEF	ID_SPITAL
1	2000	Cardiologie	20	(null)	200
2	2001	Ortopedie	15	(null)	201
3	2002	Ginecologie	25	(null)	202
4	2004	Pediatrie	40	(null)	204
5	2005	Endocrinologie	18	(null)	205
6	2006	Oftalmologie	12	(null)	201
7	2007	Dermatologie	16	(null)	202
8	2011	Neurologie	30	(null)	200
9	2012	Urologie	22	(null)	200
10	2013	Oncologie	10	(null)	200
11	2014	Pneumologie	20	(null)	201
12	2015	Cardiologie	16	(null)	200
13	2016	Neurologie	32	(null)	200

-- INSERARE MEDICI –

create sequence seq_med
start with 1400
maxvalue 9990
nocycle
nocache;

insert into angajati values (seq_med.nextval, 'Medic', 'Popescu', 'Ioan', to_date('12-03-2008', 'DD-MM-YYYY'), 80000);
insert into medici(id, tip, id_sectie) values (seq_med.currval, 'rezident', 2000);

insert into angajati values (seq_med.nextval, 'Medic', 'Ionescu', 'Ana', to_date('05-07-2010', 'DD-MM-YYYY'), 75000);
insert into medici (id, tip, id_sectie) values (seq_med.currval, 'specialist', 2001);

insert into angajati values (seq_med.nextval, 'Medic', 'Popa', 'Maria', to_date('20-09-2012', 'DD-MM-YYYY'), 90000);

```
insert into medici (id, tip, id_sectie) values (seq_med.currval, 'specialist', 2002);

insert into angajati values (seq_med.nextval, 'Medic', 'Vasile', 'Mihai',
to_date('08-04-2015', 'DD-MM-YYYY'), 82000);
insert into medici (id, tip, id_sectie) values (seq_med.currval, 'rezident', 2004);

insert into angajati values (seq_med.nextval, 'Medic', 'George', 'Elena',
to_date('16-11-2013', 'DD-MM-YYYY'), 78000);
insert into medici (id, tip, id_sectie) values (seq_med.currval, 'specialist', 2005);

insert into angajati values (seq_med.nextval, 'Medic', 'Dumitru', 'Andrei',
to_date('30-01-2016', 'DD-MM-YYYY'), 85000);
insert into medici (id, tip, id_sectie) values (seq_med.currval, 'specialist', 2006);

insert into angajati values (seq_med.nextval, 'Medic', 'Costea', 'Ioana',
to_date('14-08-2014', 'DD-MM-YYYY'), 76000);
insert into medici (id, tip, id_sectie) values (seq_med.currval, 'rezident', 2007);

insert into angajati values (seq_med.nextval, 'Medic', 'Stanescu', 'Adrian',
to_date('10-06-2011', 'DD-MM-YYYY'), 92000);
insert into medici (id, tip, id_sectie) values (seq_med.currval, 'specialist', 2011);

insert into angajati values (seq_med.nextval, 'Medic', 'Iordache', 'Cristina',
to_date('22-02-2017', 'DD-MM-YYYY'), 81000);
insert into medici (id, tip, id_sectie) values (seq_med.currval, 'specialist', 2012);

insert into angajati values (seq_med.nextval, 'Medic', 'Mihai', 'Alexandru',
to_date('07-12-2019', 'DD-MM-YYYY'), 79000);
insert into medici (id, tip, id_sectie) values (seq_med.currval, 'rezident', 2000);

insert into angajati values (seq_med.nextval, 'Medic', 'Marin', 'Ana-Maria',
to_date('18-10-2018', 'DD-MM-YYYY'), 83000);
insert into medici (id, tip, id_sectie) values (seq_med.currval, 'specialist', 2002);

insert into angajati values (seq_med.nextval, 'Medic', 'Cristea', 'Ion', to_date('05-08-2019',
'DD-MM-YYYY'), 75000);
insert into medici (id, tip, id_sectie) values (seq_med.currval, 'rezident', 2000);

insert into angajati values (seq_med.nextval, 'Medic', 'Cretu', 'Maria',
to_date('12-03-2020', 'DD-MM-YYYY'), 79000);
insert into medici (id, tip, id_sectie) values (seq_med.currval, 'rezident', 2001);

insert into angajati values (seq_med.nextval, 'Medic', 'Popa', 'Darius',
to_date('21-09-2020', 'DD-MM-YYYY'), 72000);
insert into medici (id, tip, id_sectie) values (seq_med.currval, 'rezident', 2004);

insert into angajati values (seq_med.nextval, 'Medic', 'Laura', 'Elena',
to_date('10-06-2021', 'DD-MM-YYYY'), 78000);
insert into medici (id, tip, id_sectie) values (seq_med.currval, 'rezident', 2011);
```

```
insert into angajati values (seq_med.nextval, 'Medic', 'Matei', 'Alexandru',
to_date('28-02-2022', 'DD-MM-YYYY'), 84000);
insert into medici (id, tip, id_sectie) values (seq_med.currval, 'specialist', 2005);

insert into angajati values (seq_med.nextval, 'Medic', 'Mihai', 'Cristina',
to_date('15-11-2022', 'DD-MM-YYYY'), 77000);
insert into medici (id, tip, id_sectie) values (seq_med.currval, 'specialist', 2012);

insert into angajati values (seq_med.nextval, 'Medic', 'Radu', 'Stefan',
to_date('03-08-2020', 'DD-MM-YYYY'), 82000);
insert into medici (id, tip, id_sectie) values (seq_med.currval, 'specialist', 2006);

insert into angajati values (seq_med.nextval, 'Medic', 'Nastase', 'Adriana',
to_date('19-04-2019', 'DD-MM-YYYY'), 79000);
insert into medici (id, tip, id_sectie) values (seq_med.currval, 'specialist', 2007);

insert into angajati values (seq_med.nextval, 'Medic', 'Ilie', 'Mihai', to_date('07-01-2018',
'DD-MM-YYYY'), 83000);
insert into medici (id, tip, id_sectie) values (seq_med.currval, 'specialist', 2002);

insert into angajati values (seq_med.nextval, 'Medic', 'Badea', 'Marius',
to_date('07-10-2018', 'DD-MM-YYYY'), 60000);
insert into medici (id, tip, id_sectie) values (seq_med.currval, 'rezident', 2002);
```

```
select * from medici_complet;
```

Script Output | Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 21 in 0,335 seconds

	ID	ID_TIP_ANG	NUME	PRENUME	DATA_ANGAJARII	SALARIU	TIP	ID_SECTIE	NUME_SECTIE
1	1400	Medic	Popescu	Ioan	12-03-2008	80000	rezident	2000	Cardiologie
2	1401	Medic	Ionescu	Ana	05-07-2010	75000	specialist	2001	Ortopedie
3	1402	Medic	Popa	Maria	20-09-2012	90000	specialist	2002	Ginecologie
4	1403	Medic	Vasile	Mihai	08-04-2015	82000	rezident	2004	Pediatrie
5	1404	Medic	George	Elena	16-11-2013	78000	specialist	2005	Endocrinologie
6	1405	Medic	Dumitru	Andrei	30-01-2016	85000	specialist	2006	Oftalmologie
7	1406	Medic	Costea	Ioana	14-08-2014	76000	rezident	2007	Dermatologie
8	1407	Medic	Stanescu	Adrian	10-06-2011	92000	specialist	2011	Neurologie
9	1408	Medic	Iordache	Cristina	22-02-2017	81000	specialist	2012	Urologie
10	1409	Medic	Mihai	Alexandru	07-12-2019	79000	rezident	2000	Cardiologie
11	1410	Medic	Marin	Ana-Maria	18-10-2018	83000	specialist	2002	Ginecologie
12	1411	Medic	Cristea	Ion	05-08-2019	75000	rezident	2000	Cardiologie
13	1412	Medic	Cretu	Maria	12-03-2020	79000	rezident	2001	Ortopedie
14	1413	Medic	Popa	Darius	21-09-2020	72000	rezident	2004	Pediatrie
15	1414	Medic	Laura	Elena	10-06-2021	78000	rezident	2011	Neurologie
16	1415	Medic	Matei	Alexandru	28-02-2022	84000	specialist	2005	Endocrinologie
17	1416	Medic	Mihai	Cristina	15-11-2022	77000	specialist	2012	Urologie
18	1417	Medic	Radu	Stefan	03-08-2020	82000	specialist	2006	Oftalmologie
19	1418	Medic	Nastase	Adriana	19-04-2019	79000	specialist	2007	Dermatologie
20	1419	Medic	Ilie	Mihai	07-01-2018	83000	specialist	2002	Ginecologie
21	1420	Medic	Badea	Marius	07-10-2018	60000	rezident	2002	Ginecologie

-- INSERARE SOFERI –

```
create sequence seq_sof
start with 1000
nocycle
nocache;

insert into angajati values (seq_sof.nextval, 'Sofer', 'Pop', 'George', to_date('22-10-2008',
'DD-MM-YYYY'), 30000);
insert into soferi(id, id_spital) values (seq_sof.currval, 200);

insert into angajati values (seq_sof.nextval, 'Sofer', 'Ionescu', 'Marius',
to_date('15-07-2011', 'DD-MM-YYYY'), 32000);
insert into soferi (id, id_spital) values (seq_sof.currval, 201);

insert into angajati values (seq_sof.nextval, 'Sofer', 'Popescu', 'Alexandru',
to_date('05-09-2014', 'DD-MM-YYYY'), 29000);
insert into soferi (id, id_spital) values (seq_sof.currval, 202);

insert into angajati values (seq_sof.nextval, 'Sofer', 'George', 'Andrei',
to_date('28-02-2017', 'DD-MM-YYYY'), 31000);
insert into soferi (id, id_spital) values (seq_sof.currval, 205);

insert into angajati values (seq_sof.nextval, 'Sofer', 'Tutu', 'Cosmin', to_date('23-06-2016',
'DD-MM-YYYY'), 35000);
insert into soferi (id, id_spital) values (seq_sof.currval, 200);

insert into angajati values (seq_sof.nextval, 'Sofer', 'Ionescu', 'Maria',
to_date('01-01-2010', 'DD-MM-YYYY'), 35000);
insert into soferi(id, id_spital) values (seq_sof.currval, 201);

insert into angajati values (seq_sof.nextval, 'Sofer', 'Popescu', 'Alexandru',
to_date('05-03-2012', 'DD-MM-YYYY'), 33000);
insert into soferi(id, id_spital) values (seq_sof.currval, 202);

insert into angajati values (seq_sof.nextval, 'Sofer', 'Mihai', 'Ana', to_date('10-07-2014',
'DD-MM-YYYY'), 29000);
insert into soferi(id, id_spital) values (seq_sof.currval, 200);

insert into angajati values (seq_sof.nextval, 'Sofer', 'Popa', 'Andrei', to_date('15-09-2016',
'DD-MM-YYYY'), 27000);
insert into soferi(id, id_spital) values (seq_sof.currval, 204);

insert into angajati values (seq_sof.nextval, 'Sofer', 'Costic', 'Cătălina',
to_date('20-12-2018', 'DD-MM-YYYY'), 25000);
insert into soferi(id, id_spital) values (seq_sof.currval, 205);
```

```
select * from soferi_complet;
```

Script Output | Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 10 in 1,972 seconds

	ID	ID_TIP_ANG	NUME	PRENUME	DATA_ANGAJARII	SALARIU	ID_SPITAL	NUME_SPITAL
1	1001	Sofer	Ionescu	Marius	15-07-2011	32000	201	SPITALUL CLINIC JUD DE URGENTA SIBIU
2	1002	Sofer	Popescu	Alexandru	05-09-2014	29000	202	SPITALUL CLINIC DE URGENTA FLOREASCA
3	1003	Sofer	George	Andrei	28-02-2017	31000	205	SPITALUL JUD DE URGENTA SF. SPIRIDON
4	1004	Sofer	Tutu	Cosmin	23-06-2016	35000	200	SPITALUL JUD DE URGENTA SF. IOAN CEL NOU
5	1005	Sofer	Ionescu	Maria	01-01-2010	35000	201	SPITALUL CLINIC JUD DE URGENTA SIBIU
6	1006	Sofer	Popescu	Alexandru	05-03-2012	33000	202	SPITALUL CLINIC DE URGENTA FLOREASCA
7	1007	Sofer	Mihai	Ana	10-07-2014	29000	200	SPITALUL JUD DE URGENTA SF. IOAN CEL NOU
8	1008	Sofer	Popa	Andrei	15-09-2016	27000	204	SPITALUL CLINIC SF. MARIA
9	1009	Sofer	Costic	Cătălina	20-12-2018	25000	205	SPITALUL JUD DE URGENTA SF. SPIRIDON
10	1000	Sofer	Pop	George	22-10-2008	30000	200	SPITALUL JUD DE URGENTA SF. IOAN CEL NOU

```
select * from angajati;
```

Script Output | Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 31 in 0,004 seconds

	ID	ID_TIP_ANG	NUME	PRENUME	DATA_ANGAJARII	SALARIU
1	1400	Medic	Popescu	Ioan	12-03-2008	80000
2	1001	Sofer	Ionescu	Marius	15-07-2011	32000
3	1002	Sofer	Popescu	Alexandru	05-09-2014	29000
4	1003	Sofer	George	Andrei	28-02-2017	31000
5	1004	Sofer	Tutu	Cosmin	23-06-2016	35000
6	1005	Sofer	Ionescu	Maria	01-01-2010	35000
7	1006	Sofer	Popescu	Alexandru	05-03-2012	33000
8	1007	Sofer	Mihai	Ana	10-07-2014	29000
9	1008	Sofer	Popa	Andrei	15-09-2016	27000
10	1009	Sofer	Costic	Cătălina	20-12-2018	25000
11	1401	Medic	Ionescu	Ana	05-07-2010	75000
12	1402	Medic	Popa	Maria	20-09-2012	90000
13	1403	Medic	Vasile	Mihai	08-04-2015	82000
14	1404	Medic	George	Elena	16-11-2013	78000
15	1405	Medic	Dumitru	Andrei	30-01-2016	85000
16	1406	Medic	Costea	Ioana	14-08-2014	76000
17	1407	Medic	Stanescu	Adrian	10-06-2011	92000
18	1408	Medic	Iordache	Cristina	22-02-2017	81000
19	1409	Medic	Mihai	Alexandru	07-12-2019	79000
20	1410	Medic	Marin	Ana-Maria	18-10-2018	83000
21	1000	Sofer	Pop	George	22-10-2008	30000
22	1411	Medic	Cristea	Ion	05-08-2019	75000
23	1412	Medic	Cretu	Maria	12-03-2020	79000
24	1413	Medic	Popa	Darius	21-09-2020	72000
25	1414	Medic	Laura	Elena	10-06-2021	78000
26	1415	Medic	Matei	Alexandru	28-02-2022	84000
27	1416	Medic	Mihai	Cristina	15-11-2022	77000
28	1417	Medic	Radu	Stefan	03-08-2020	82000
29	1418	Medic	Nastase	Adriana	19-04-2019	79000
30	1419	Medic	Ilie	Mihai	07-01-2018	83000
31	1420	Medic	Badea	Marius	07-10-2018	60000

-- INSERARE AMBULANTE --

```
create sequence seq_amb
start with 300
nocycle
nocache;

insert into ambulante values (seq_amb.nextval, 'AB123CD', 1, 200);
insert into ambulante values (seq_amb.nextval, 'EF456GH', 2, 201);
insert into ambulante values (seq_amb.nextval, 'IJ789KL', 1, 202);
insert into ambulante values (seq_amb.nextval, 'MN012OP', 2, 205);
insert into ambulante values (seq_amb.nextval, 'QR345ST', 1, 204);
insert into ambulante values (seq_amb.nextval, 'IF023CL', 1, 200);
insert into ambulante values (seq_amb.nextval, 'MF256GT', 2, 201);
insert into ambulante values (seq_amb.nextval, 'TM989KE', 1, 202);
insert into ambulante values (seq_amb.nextval, 'GJ612OI', 2, 205);
insert into ambulante values (seq_amb.nextval, 'B305STP', 1, 200);
```

```
select * from ambulante;
```

Script Output | Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 15 in 0,347 seconds

	ID	NR_INMATRICULARE	CAPACITATE	ID_SPITAL
1	305	IF145LL	2	200
2	306	MF957GI	2	201
3	307	GJ709PO	2	202
4	308	AR302LP	1	205
5	309	TM375ST	2	200
6	310	IF023CL	1	200
7	311	MF256GT	2	201
8	312	TM989KE	1	202
9	313	GJ612OI	2	205
10	314	B305STP	1	200
11	300	AB123CD	1	200
12	301	EF456GH	2	201
13	302	IJ789KL	1	202
14	303	MN012OP	2	205
15	304	QR345ST	1	204

--INSERARE SERVICII_AMBULANTA–

```
insert into servicii_amb values (1000, 100, 300, TO_DATE('01-05-2023', 'DD-MM-YYYY'),
'Accident rutier');
insert into servicii_amb values (1001, 101, 301, TO_DATE('02-06-2022', 'DD-MM-YYYY'),
'Probleme respiratorii');
insert into servicii_amb values (1002, 102, 302, TO_DATE('03-01-2023', 'DD-MM-YYYY'),
'Accident vascular cerebral');
insert into servicii_amb values (1003, 103, 303, TO_DATE('04-02-2020', 'DD-MM-YYYY'),
'Dureri abdominale severe');
```

```
insert into servicii_amb values (1004, 104, 304, TO_DATE('05-10-2020', 'DD-MM-YYYY'),
'Fractură osoasă');
insert into servicii_amb values (1005, 105, 305, TO_DATE('06-12-2021', 'DD-MM-YYYY'),
'Criză de astm');
insert into servicii_amb values (1006, 106, 306, TO_DATE('07-11-2022', 'DD-MM-YYYY'),
'Le?in');
insert into servicii_amb values (1007, 107, 307, TO_DATE('08-07-2023', 'DD-MM-YYYY'),
'Alergie severă');
insert into servicii_amb values (1008, 108, 308, TO_DATE('09-08-2023', 'DD-MM-YYYY'),
'Sângerare abundentă');
insert into servicii_amb values (1000, 110, 309, TO_DATE('10-04-2020', 'DD-MM-YYYY'),
'Arsură gravă');
insert into servicii_amb values (1000, 111, 300, TO_DATE('11-03-2021', 'DD-MM-YYYY'),
'Intoxica?ie alimentară');
insert into servicii_amb values (1001, 112, 301, TO_DATE('12-05-2022', 'DD-MM-YYYY'),
'Convulsii');
insert into servicii_amb values (1002, 113, 302, TO_DATE('13-06-2023', 'DD-MM-YYYY'),
'Accident rutier');
insert into servicii_amb values (1003, 114, 303, TO_DATE('14-10-2022', 'DD-MM-YYYY'),
'Intoxica?ie alimentară');
insert into servicii_amb values (1004, 115, 304, TO_DATE('15-05-2023', 'DD-MM-YYYY'),
'Le?in');
insert into servicii_amb values (1003, 114, 308, TO_DATE('14-09-2022', 'DD-MM-YYYY'),
'Convulsii');
```

insert into servicii_amb values (1004, 115, 313, TO_DATE('15-01-2023', 'DD-MM-YYYY'),'Fractura osoasa');

```
select * from servicii_amb;
```

cript Output × | Query Result × | Query Result 1 × | Query Result 2 ×

SQL | All Rows Fetched: 17 in 0,042 seconds

	ID_SOFER	ID_PACIENT	ID_AMBULANTA	DATA_SERVICIU	MOTIV
1	1000	100	300	01-05-2023	Accident rutier
2	1001	101	301	02-06-2022	Probleme respiratorii
3	1002	102	302	03-01-2023	Accident vascular cerebral
4	1003	103	303	04-02-2020	Dureri abdominale severe
5	1004	104	304	05-10-2020	Fractură osoasă
6	1005	105	305	06-12-2021	Criză de astm
7	1006	106	306	07-11-2022	Leșin
8	1007	107	307	08-07-2023	Alergie severă
9	1008	108	308	09-08-2023	Sângerare abundentă
10	1000	111	300	11-03-2021	Intoxicație alimentară
11	1001	112	301	12-05-2022	Convulsii
12	1002	113	302	13-06-2023	Accident rutier
13	1003	114	303	14-10-2022	Intoxicație alimentară
14	1004	115	304	15-05-2023	Leșin
15	1000	110	309	10-04-2020	Arsură gravă
16	1003	114	308	14-09-2022	Convulsii
17	1004	115	313	15-01-2023	Fractura osoasa

--INSERARE MEDICAMENTE–

create sequence seq_mdm
start with 4000
nocycle
nocache;

insert into medicamente values (seq_mdm.nextval, 'Paracetamol', 'XYZ Company', to_date('06-01-2023', 'DD-MM-YYYY'));
insert into medicamente values (seq_mdm.nextval, 'Ibuprofen', 'ABC Pharma', to_date('15-07-2023', 'DD-MM-YYYY'));
insert into medicamente values (seq_mdm.nextval, 'Amoxicilină', 'DEF Pharma', to_date('28-02-2024', 'DD-MM-YYYY'));
insert into medicamente values (seq_mdm.nextval, 'Omeprazol', 'GHI Corporation', to_date('11-10-2023', 'DD-MM-YYYY'));
insert into medicamente values (seq_mdm.nextval, 'Aspirină', 'JKL Drugs', to_date('20-09-2023', 'DD-MM-YYYY'));
insert into medicamente values (seq_mdm.nextval, 'Simvastatină', 'MNO Pharmaceuticals', to_date('04-05-2024', 'DD-MM-YYYY'));

```
insert into medicamente values (seq_mdm.nextval, 'Metformină', 'PQR Labs',
to_date('31-10-2023', 'DD-MM-YYYY'));
insert into medicamente values (seq_mdm.nextval, 'Ventolin', 'STU Company',
to_date('15-03-2024', 'DD-MM-YYYY'));
insert into medicamente values (seq_mdm.nextval, 'Cetirizină', 'VWX Pharmaceuticals',
to_date('08-10-2023', 'DD-MM-YYYY'));
insert into medicamente values (seq_mdm.nextval, 'Loratadină', 'YZA Corporation',
to_date('01-01-2024', 'DD-MM-YYYY'));
insert into medicamente values (seq_mdm.nextval, 'Sertralină', 'BCD Drugs',
to_date('12-05-2023', 'DD-MM-YYYY'));
insert into medicamente values (seq_mdm.nextval, 'Losartan', 'EFG Pharmaceuticals',
to_date('20-11-2023', 'DD-MM-YYYY'));
insert into medicamente values (seq_mdm.nextval, 'Gabapentină', 'HIJ Labs',
to_date('05-10-2024', 'DD-MM-YYYY'));
insert into medicamente values (seq_mdm.nextval, 'Metoprolol', 'KLM Company',
to_date('15-09-2023', 'DD-MM-YYYY'));
insert into medicamente values (seq_mdm.nextval, 'Rosuvastatină', 'NOP
Pharmaceuticals', to_date('15-02-2024', 'DD-MM-YYYY'));
insert into medicamente values (seq_mdm.nextval, 'Ranitidină', 'QRS Drugs',
to_date('10-01-2023', 'DD-MM-YYYY'));
insert into medicamente values (seq_mdm.nextval, 'Clonazepam', 'STU
Pharmaceuticals', to_date('20-03-2024', 'DD-MM-YYYY'));
insert into medicamente values (seq_mdm.nextval, 'Levotiroxină', 'VWX Labs',
to_date('08-05-2023', 'DD-MM-YYYY'));
insert into medicamente values (seq_mdm.nextval, 'Furosemid', 'YZA Company',
to_date('15-01-2024', 'DD-MM-YYYY'));
insert into medicamente values (seq_mdm.nextval, 'Ciprofloxacină', 'BCD
Pharmaceuticals', to_date('20-12-2023', 'DD-MM-YYYY'));
insert into medicamente values (seq_mdm.nextval, 'Xanax', 'DPD Meds',
to_date('10-08-2023', 'DD-MM-YYYY'));
```

```
select * from medicamente;
```

Script Output | Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 21 in 0,1 seconds

	ID	NUME	COMPANIE	DATA_EXPIRARII
1	4000	Paracetamol	XYZ Company	06-01-2023
2	4001	Ibuprofen	ABC Pharma	15-07-2023
3	4002	Amoxicilină	Gripovit	28-02-2024
4	4003	Omeprazol	GHI Corporation	11-10-2023
5	4004	Aspirină	JKL Drugs	20-09-2023
6	4005	Simvastatină	MNO Pharmaceuticals	04-05-2024
7	4006	Metformină	PQR Labs	31-10-2023
8	4007	Ventolin	STU Company	15-03-2024
9	4008	Cetirizină	VWX Pharmaceuticals	08-10-2023
10	4009	Loratadină	YZA Corporation	01-01-2024
11	4010	Sertralină	BCD Drugs	12-05-2023
12	4011	Losartan	EFG Pharmaceuticals	20-11-2023
13	4012	Gabapentină	HIJ Labs	05-10-2024
14	4013	Metoprolol	Gripovit	15-09-2023
15	4014	Rosuvastatină	NOP Pharmaceuticals	15-02-2024
16	4015	Ranitidină	Gripovit	10-01-2023
17	4016	Clonazepam	STU Pharmaceuticals	20-03-2024
18	4017	Levotiroxină	VWX Labs	08-05-2023
19	4018	Furosemid	YZA Company	15-01-2024
20	4019	Ciprofloxacină	BCD Pharmaceuticals	20-12-2023
21	4020	Xanax	DPD Meds	10-08-2023

--INSERARE DIAGNOSTICE–

```
create sequence seq_diag
start with 5025
nocycle
nocache;

insert into diagnostice values (seq_diag.nextval, 'Diabet', 'medicamentos');
insert into diagnostice values (seq_diag.nextval, 'Hipertensiune arterială',
'medicamentos' );
insert into diagnostice values (seq_diag.nextval, 'Gripă', 'medicamentos');
insert into diagnostice values (seq_diag.nextval, 'Insuficien?ă cardiacă',
'medicamentos');
insert into diagnostice values (seq_diag.nextval, 'Osteoporoză', 'exercitii fizice' );
insert into diagnostice values (seq_diag.nextval, 'Anemie','suplimente alimentare');
insert into diagnostice values (seq_diag.nextval, 'Migrenă','medicamentos, repaus' );
insert into diagnostice values (seq_diag.nextval, 'Ulcer gastric','medicamentos si dieta' );
insert into diagnostice values (seq_diag.nextval, 'Depresie','terapie si medicamentos');
insert into diagnostice values (seq_diag.nextval, 'Astm bron?ic', 'medicamentos,
inhalatoare' );
insert into diagnostice values (seq_diag.nextval, 'Diabet', 'dieta');
```

```
select * from diagnostice;
```

Script Output | Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 11 in 0,099 seconds

	ID	NUME	TIP_TRATAMENT
1	5030	Anemie	suplimente alimentare
2	5034	Astm bronșic	medicamentos, inhalatoare
3	5033	Depresie	terapie
4	5035	Diabet	dieta
5	5025	Diabet	medicamentos
6	5027	Gripă	medicamentos
7	5026	Hipertensiune arterială	medicamentos
8	5028	Insuficiență cardiacă	medicamentos
9	5031	Migrenă	medicamentos, repaus
10	5029	Osteoporoză	exercitii fizice
11	5032	Ulcer gastric	medicamentos si dieta

--INSERARE RETETE–

insert into retete (id_diagnostic, id_medicament, doza, durata) values (5025, 4000, '5 mg', '7 zile');
insert into retete (id_diagnostic, id_medicament, doza, durata) values (5025, 4011, '12 mg', '14 zile');
insert into retete (id_diagnostic, id_medicament, doza, durata) values (5025, 4010, '9 mg', '4 zile');
insert into retete (id_diagnostic, id_medicament, doza, durata) values (5026, 4001, '10 mg', '14 zile');
insert into retete (id_diagnostic, id_medicament, doza, durata) values (5026, 4012, '12 mg', '10 zile');
insert into retete (id_diagnostic, id_medicament, doza, durata) values (5027, 4002, '15 mg', '10 zile');
insert into retete (id_diagnostic, id_medicament, doza, durata) values (5027, 4013, '8 mg', '21 zile');
insert into retete (id_diagnostic, id_medicament, doza, durata) values (5028, 4003, '20 mg', '21 zile');
insert into retete (id_diagnostic, id_medicament, doza, durata) values (5028, 4014, '15 mg', '14 zile');
insert into retete (id_diagnostic, id_medicament, doza, durata) values (5026, 4004, '8 mg', '14 zile');
insert into retete (id_diagnostic, id_medicament, doza, durata) values (5027, 4015, '10 mg', '7 zile');

insert into retete (id_diagnostic, id_medicament, doza, durata) values (5025, 4002, '7 mg', '5 zile');
insert into retete (id_diagnostic, id_medicament, doza, durata) values (5030, 4005, '12 mg', '7 zile');
insert into retete (id_diagnostic, id_medicament, doza, durata) values (5030, 4016, '18 mg', '14 zile');
insert into retete (id_diagnostic, id_medicament, doza, durata) values (5031, 4006, '18 mg', '14 zile');
insert into retete (id_diagnostic, id_medicament, doza, durata) values (5031, 4017, '25 mg', '10 zile');
insert into retete (id_diagnostic, id_medicament, doza, durata) values (5032, 4007, '25 mg', '10 zile');
insert into retete (id_diagnostic, id_medicament, doza, durata) values (5032, 4018, '15 mg', '21 zile');
insert into retete (id_diagnostic, id_medicament, doza, durata) values (5030, 4008, '15 mg', '21 zile');
insert into retete (id_diagnostic, id_medicament, doza, durata) values (5030, 4019, '12 mg', '14 zile');
insert into retete (id_diagnostic, id_medicament, doza, durata) values (5031, 4020, '6 mg', '8 zile');
insert into retete (id_diagnostic, id_medicament, doza, durata) values (5032, 4015, '2 mg', '7 zile');
insert into retete (id_diagnostic, id_medicament, doza, durata) values (5034, 4009, '10 mg', '14 zile');

```
select * from retete;
```

cript Output × | Query Result × | Query Result 1 × | Query Result 2 ×

SQL | All Rows Fetched: 23 in 0,1 seconds

	ID_DIAGNOSTIC	ID_MEDICA...	DOZA	DURATA
1	5025	4000	5 mg	7 zile
2	5025	4011	12 mg	14 zile
3	5025	4010	9 mg	4 zile
4	5026	4001	10 mg	14 zile
5	5026	4012	12 mg	10 zile
6	5027	4002	15 mg	10 zile
7	5027	4013	8 mg	21 zile
8	5028	4003	20 mg	21 zile
9	5028	4014	15 mg	14 zile
10	5026	4004	8 mg	14 zile
11	5027	4015	10 mg	7 zile
12	5025	4002	7 mg	5 zile
13	5030	4005	12 mg	7 zile
14	5030	4016	18 mg	14 zile
15	5031	4006	18 mg	14 zile
16	5031	4017	25 mg	10 zile
17	5032	4007	25 mg	10 zile
18	5032	4018	15 mg	21 zile
19	5030	4008	15 mg	21 zile
20	5030	4019	12 mg	14 zile
21	5031	4020	6 mg	8 zile
22	5032	4015	2 mg	7 zile
23	5034	4009	10 mg	14 zile

--INSERARE CONSULATATII–

```
create sequence seq_cons
start with 1
nocycle
nocache;

insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1400, 150, 5025, to_date('18.10.2022', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1401, 151, 5026, to_date('19.08.2020', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1402, 152, 5027, to_date('20.06.2021', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1403, 153, 5028, to_date('21.04.2023', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1404, 154, 5029, to_date('22.02.2023', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1405, 155, 5030, to_date('23.10.2023', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1406, 156, 5031, to_date('24.10.2019', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1407, 157, 5032, to_date('25.11.2022', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1408, 158, 5033, to_date('26.09.2019', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1409, 159, 5034, to_date('27.10.2018', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1410, 160, 5035, to_date('22.12.2022', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1411, 161, 5025, to_date('29.10.2020', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1412, 162, 5026, to_date('10.05.2023', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1413, 163, 5027, to_date('31.10.2022', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1414, 164, 5028, to_date('01.11.2010', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1415, 165, 5029, to_date('02.11.2013', 'DD.MM.YYYY'));
```

```
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1416, 166, 5030, to_date('03.11.2015', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1417, 167, 5031, to_date('04.01.2023', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1418, 168, 5032, to_date('05.11.2022', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1419, 169, 5033, to_date('06.11.2020', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1420, 170, 5034, to_date('07.11.2021', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1400, 150, 5035, to_date('08.10.2022', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1401, 151, 5025, to_date('09.12.2020', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1402, 153, 5026, to_date('10.11.2021', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1403, 154, 5027, to_date('11.07.2021', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1404, 151, 5028, to_date('12.01.2023', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1405, 154, 5029, to_date('13.03.2023', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1406, 160, 5030, to_date('14.04.2023', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1407, 160, 5031, to_date('15.10.2022', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1408, 160, 5032, to_date('16.12.2020', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1400, 100, 5025, to_date('30.06.2022', 'DD.MM.YYYY'));
insert into consultatii (id, id_medic, id_pacient, id_diagnostic, data_examinarii)
values (seq_cons.nextval, 1408, 101, 5027, to_date('30.06.2022', 'DD.MM.YYYY'));
```

```
select * from consultatii;
```

Script Output × | Query Result × | Query Result 1 × | Query Result 2 ×

SQL | All Rows Fetched: 32 in 0,059 seconds

	ID	DATA_EXAMINARII	ID_MEDIC	ID_PACIENT	ID_DIAGNOSTIC
1	1	18-10-2022	1400	150	5025
2	2	19-08-2020	1401	151	5026
3	3	20-06-2021	1402	152	5027
4	4	21-04-2023	1403	153	5028
5	5	22-02-2023	1404	154	5029
6	6	23-05-2023	1405	155	5030
7	7	24-10-2019	1406	156	5031
8	8	25-11-2022	1407	157	5032
9	9	26-09-2019	1408	158	5033
10	10	27-10-2018	1409	159	5034
11	11	22-12-2022	1410	160	5035
12	12	29-10-2020	1411	161	5025
13	13	10-05-2023	1412	162	5026
14	14	31-10-2022	1413	163	5027
15	15	01-11-2010	1414	164	5028
16	16	02-11-2013	1415	165	5029
17	17	03-11-2015	1416	166	5030
18	18	04-01-2023	1417	167	5031
19	19	05-11-2022	1418	168	5032
20	20	06-11-2020	1419	169	5033
21	21	07-11-2021	1420	170	5034
22	22	08-10-2022	1400	150	5035
23	23	09-12-2020	1401	151	5025
24	24	10-11-2021	1402	153	5026
25	25	11-07-2021	1403	154	5027
26	26	12-01-2023	1404	151	5028
27	27	13-03-2023	1405	154	5029
28	28	14-04-2023	1406	160	5030
29	29	15-10-2022	1407	160	5031
30	30	16-12-2020	1408	160	5032
31	31	30-06-2022	1400	100	5025
32	32	30-06-2022	1408	101	5027

--INSERARE FACTURI–

```
create sequence seq_fac
start with 7000
nocycle
nocache;

insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 200, 'cash', 1);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 150, 'card', 2);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 300, 'cash', 3);
```

```
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 180,
'card', 4);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 250,
'cash', 5);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 220,
'card', 6);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 280,
'cash', 7);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 190,
'card', 8);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 320,
'cash', 9);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 170,
'card', 10);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 220,
'cash', 11);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 300,
'card', 12);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 190,
'cash', 13);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 260,
'card', 14);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 240,
'cash', 15);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 180,
'card', 16);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 290,
'cash', 17);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 210,
'card', 18);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 280,
'cash', 19);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 230,
'card', 20);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 290,
'cash', 21);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 220,
'card', 22);
```

```
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 310,
'cash', 23);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 200,
'card', 24);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 270,
'cash', 25);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 190,
'card', 26);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 280,
'cash', 27);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 230,
'card', 28);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 320,
'cash', 29);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 250,
'card', 30);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 150,
'cash', 31);
insert into facturi (id, suma, mod_plata, id_consultatie) values (seq_fac.nextval, 50,
'cash', 32);
```

```
select * from facturi;
```

Script Output | Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 32 in 0,002 seconds

	ID	SUMA	MOD_PLATA	ID_CONSULTATIE
1	7000	200	cash	1
2	7001	150	card	2
3	7002	300	cash	3
4	7003	180	card	4
5	7004	250	cash	5
6	7005	220	card	6
7	7006	280	cash	7
8	7007	190	card	8
9	7008	320	cash	9
10	7009	170	card	10
11	7010	220	cash	11
12	7011	300	card	12
13	7012	190	cash	13
14	7013	260	card	14
15	7014	240	cash	15
16	7015	180	card	16
17	7016	290	cash	17
18	7017	210	card	18
19	7018	280	cash	19
20	7019	230	card	20
21	7020	290	cash	21
22	7021	220	card	22
23	7022	310	cash	23
24	7023	200	card	24
25	7024	270	cash	25
26	7025	190	card	26
27	7026	280	cash	27
28	7027	230	card	28
29	7028	320	cash	29
30	7029	250	card	30
31	7036	150	cash	31
32	7037	50	cash	32

12. Implementarea a 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:

• subcereri sincronizate în care intervin cel puțin 3 tabele
• subcereri nesincronizate în clauza FROM
• grupări de date cu subcereri nesincronizate în care intervin cel putin 3 tabele, funcții grup, filtrare la nivel de grupuri (în cadrul aceleiași cereri)
• ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri)
• utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE
• utilizarea a cel puțin 1 bloc de cerere (clauza WITH)

1. Cererea 1

Să se afișeze spitalele din afara Bucureștiului care au un număr de secții mai mare decât media numărului de secții al spitalelor din București.

Cererea utilizează grupări de date cu subcereri nesincronizate în care intervin cel puțin 3 tabele, funcții grup, filtrare la nivel de grupuri, 2 funcții pe șiruri de caractere, subcerere nesincronizată în clauza FROM.

```
select s.id, s.nume, a.judet, count(sec.id) numar_sectii
from spitale s
join adrese a on a.id = s.id_adresa
join sectii sec on sec.id_spital = s.id
where lower(a.judet) != 'bucuresti'
group by s.id, s.nume, a.judet
having count(sec.id) > (select avg(numar_sectii)
                        from (
                            select s.id, count(sec.id) numar_sectii
                            from spitale s
                            join adrese a on a.id = s.id_adresa
                            join sectii sec on sec.id_spital = s.id
                            where initcap(a.judet) = 'Bucuresti'
                            group by s.id )
);
```

	ID	NUME	JUDET	NUMAR_SECTII
1	200	SPITALUL JUD DE URGENTA SF. IOAN CEL NOU	Suceava	6
2	201	SPITALUL CLINIC JUD DE URGENTA SIBIU	Sibiu	3

```
select s.id, s.nume, a.judet, count(sec.id) numar_sectii
from spitale s
join adrese a on a.id = s.id_adresa
join sectii sec on sec.id_spital = s.id
where lower(a.judet) != 'bucuresti'
group by s.id, s.nume, a.judet
having count(sec.id) > (select avg(numar_sectii)
                        from (
                            select s.id, count(sec.id) numar_sectii
                            from spitale s
                            join adrese a on a.id = s.id_adresa
                            join sectii sec on sec.id_spital = s.id
                            where initcap(a.judet) = 'Bucuresti'
                            group by s.id
```

)

)

2. Cererea 2

Să se afișeze numele, prenumele, vârsta pacienților cu număr maxim de consultații și diagnosticele asociate lui/lor.
Cererea utilizează un 1 bloc de cerere (clauza WITH), subcerere nesincronizată în clauza FROM, grupari de date, filtrare la nivel de grupuri, functii grup

```
with pac as (
    select p.id pac_id
    from pacienti p
    join consultatii c on c.id_pacient = p.id
    group by p.id
    having count(c.id_pacient) = ( select max(nr_consultatii) nr_max_cons
                                   from (
                                           select count(id_pacient) nr_consultatii
                                           from consultatii
                                           group by id_pacient )
                                 )
)
select p.id, p.nume, p.prenume, extract(year from sysdate) - extract(year from p.data_nasterii) varsta, d.nume
from pacienti p
join pac on pac.pac_id = p.id
join consultatii c on c.id_pacient = p.id
join diagnostice d on c.id_diagnostic = d.id;
```

	ID	NUME	PRENUME	VARSTA	NUME_1
1	160	Diaconu	Dumitru	37	Diabet
2	160	Diaconu	Dumitru	37	Anemie
3	160	Diaconu	Dumitru	37	Migrenă
4	160	Diaconu	Dumitru	37	Ulcer gastric

```
with pac as (
   select p.id pac_id
   from pacienti p
   join consultatii c on c.id_pacient = p.id
   group by p.id
   having count(c.id_pacient) = ( select max(nr_consultatii) nr_max_cons
                                  from (
                                          select count(id_pacient) nr_consultatii
                                          from consultatii
                                          group by id_pacient
                                          )
                                 )
)
```

select p.id, p.nume, p.prenume, extract(year from sysdate) - extract(year from p.data_nasterii) varsta, d.nume
from pacienti p
join pac on pac.pac_id = p.id
join consultatii c on c.id_pacient = p.id
join diagnostice d on c.id_diagnostic = d.id;

3. Cererea 3

Să se afișeze, pentru fiecare diagnostic în parte, numărul total de consultații și încasările provenite din plata cu cardul. Rezultatele sunt afișate în ordine descrescătoare a numărului de consultații, respectiv a încasărilor.

Cererea utilizează funcțiile NVL și DECODE, funcții grup, grupări de date, funcție pe șiruri de caractere

```
select d.nume, count(c.id) nr_consultatii,
sum(nvl(decode(upper(f.mod_plata), 'CARD', f.suma), 0)) suma_plata_card
from diagnostice d
join consultatii c on d.id = c.id_diagnostic
join facturi f on c.id = f.id_consultatie
group by d.nume
order by nr_consultatii desc, suma_plata_card desc;
```

	NUME	NR_CONSULTATII	SUMA_PLATA_CARD
1	Diabet	6	520
2	Gripă	4	260
3	Anemie	3	450
4	Ulcer gastric	3	440
5	Insuficiență cardiacă	3	370
6	Hipertensiune arterială	3	350
7	Migrenă	3	210
8	Osteoporoză	3	180
9	Depresie	2	230
10	Astm bronșic	2	170

select d.nume, count(c.id) nr_consultatii,
sum(nvl(decode(upper(f.mod_plata), 'CARD', f.suma), 0)) suma_plata_card
from diagnostice d
join consultatii c on d.id = c.id_diagnostic
join facturi f on c.id = f.id_consultatie

group by d.nume
order by nr_consultatii desc, suma_plata_card desc;

4. Cererea 4

Sa se afișeze numele și nivelul de experienta al medicilor de pe secția de neurologie al spitalelor care se afla în orașul Suceava. Se considera medic avansat cel cu experiența de lucru sub 5 ani și medic experimentat cel cu vechime de peste 5 ani.

Cererea utilizează expresia CASE, funcție pe date calendaristice, funcții pe șiruri de caractere.

```
select a.nume || ' ' || a.prenume nume_medic, sec.nume nume_sectie,
    case
        when months_between(sysdate,a.data_angajarii) < 60 then 'Avansat'
        when months_between(sysdate,a.data_angajarii) >= 60 then 'Experimentat'
    end categorie_medic, sp.nume nume_spital
from  medici m
join angajati a on a.id = m.id
join sectii sec on sec.id = m.id_sectie
join spitale sp on sp.id = sec.id_spital
join adrese ad on ad.id = sp.id_adresa
where lower(sec.nume) = 'cardiologie'
and lower(ad.oras) = 'suceava';
```

	NUME_MEDIC	NUME_SECTIE	CATEGORIE_MEDIC	NUME_SPITAL
1	Mihai Alexandru	Cardiologie	Avansat	SPITALUL JUD DE URGENTA SF. IOAN CEL NOU
2	Popescu Ioan	Cardiologie	Experimentat	SPITALUL JUD DE URGENTA SF. IOAN CEL NOU
3	Cristea Ion	Cardiologie	Avansat	SPITALUL JUD DE URGENTA SF. IOAN CEL NOU

select a.nume || ' ' || a.prenume nume_medic, sec.nume nume_sectie,
  case
    when months_between(sysdate,a.data_angajarii) < 60 then 'Avansat'
    when months_between(sysdate,a.data_angajarii) >= 60 then 'Experimentat'

end categorie_medic, sp.nume nume_spital
from medici m
join angajati a on a.id = m.id
join sectii sec on sec.id = m.id_sectie
join spitale sp on sp.id = sec.id_spital
join adrese ad on ad.id = sp.id_adresa
where lower(sec.nume) = 'cardiologie'
and lower(ad.oras) = 'suceava';

5. Cererea 5

Numărul consultațiilor de la secția de ortopedie în cadrul cărora pacientului i s-au prescris medicamente care expira în următoarele 3 luni.
Cererea utilizează subcereri sincronizate în care intervin 5 tabele, funcție pe date calendaristice, funcții pe șiruri de caractere, funcții grup

```
select count(c.id) nr_cons
from consultatii c
join medici med on med.id = c.id_medic
join sectii sec on med.id_sectie = sec.id
where c.id_pacient in ( select p.id
                        from pacienti p
                        join consultatii cons on cons.id_pacient = p.id
                        join diagnostice diag on diag.id = cons.id_diagnostic
                        join retete ret on ret.id_diagnostic = diag.id
                        join medicamente m on ret.id_medicament = m.id
                        where p.id = c.id_pacient
                        and m.data_expirarii between sysdate and add_months(sysdate, 3)
                        )
and lower(sec.nume) = 'ortopedie';
```

	NR_CONS
1	3

select count(c.id) nr_cons

```
from consultatii c
join medici med on med.id = c.id_medic
join sectii sec on med.id_sectie = sec.id
where c.id_pacient in ( select p.id
                from pacienti p
                join consultatii cons on cons.id_pacient = p.id
                join diagnostice diag on diag.id = cons.id_diagnostic
                join retete ret on ret.id_diagnostic = diag.id
                join medicamente m on ret.id_medicament = m.id
                where p.id = c.id_pacient
                and m.data_expirarii between sysdate and add_months(sysdate, 3)
                )
and lower(sec.nume) = 'ortopedie';
```

13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri

1. Operație de actualizare

Să se actualizeze adresa pacienților care au avut consultații după data de 1 mai 2023 cu adresa Sibiu, Strada Vasile Alecsandri, nr 8.

Adresa pacienților care au avut consultații după data de 1 mai 2023, **înainte** de actualizare:

```
select id_pacient, a.oras, data_examinarii
from consultatii
join pacienti p on p.id = id_pacient
join adrese a on a.id = p.id_adresa
where data_examinarii >= to_date('01-05-2023', 'DD-MM-YYYY');
```

	ID_PACIENT	ORAS	STRADA	NUMAR	DATA_EXAMINARII
1	155	Botoșani	Strada Nicolae Bălcescu	3	23-05-2023
2	162	Brașov	Bulevardul Decebal	18	10-05-2023

select id_pacient, a.oras, data_examinarii
from consultatii
join pacienti p on p.id = id_pacient
join adrese a on a.id = p.id_adresa
where data_examinarii >= to_date('01-05-2023', 'DD-MM-YYYY');

Executarea operației de actualizare a datelor:

```
update pacienti
set id_adresa = (
    select id
    from adrese
    where lower(oras) = 'sibiu' and lower(strada) = 'strada vasile alecsandri' and numar =
)
where id in (
    select id_pacient
    from consultatii
    where data_examinarii >= to_date('01-05-2023', 'DD-MM-YYYY')
);
```

```
2 rows updated.
```

update pacienti
set id_adresa = (select id
from adrese
where lower(oras) = 'sibiu' and lower(strada) = 'strada vasile alecsandri')
where id in (select id_pacient
from consultatii
where data_examinarii >= to_date('01-05-2023', 'DD-MM-YYYY')
);

Adresa pacienților care au avut consultații după data de 1 mai 2023, **dupa** actualizare:

	ID_PACIENT	ORAS	STRADA	NUMAR	DATA_EXAMINARII
1	155	Sibiu	Strada Vasile Alecsandri	8	23-05-2023
2	162	Sibiu	Strada Vasile Alecsandri	8	10-05-2023

2. Operație de actualizare

Să se actualizeze compania producătoare a medicamentelor utilizate în tratamentul diagnosticului ‘Gripă’ în ‘Gripovit’.

Companiile medicamentelor utilizate în tratamentul diagnosticului ‘Gripa’, **înainte** de actualizare:

```
select m.id, m.nume, m.companie
from medicamente m
where m.id in (
                select r.id_medicament
                from retete r
                join diagnostice d on r.id_diagnostic = d.id
                where lower(d.nume) like 'gripă');
```

t Output × | Query Result × | Query Result 1 × | Query Result 2 ×

SQL | All Rows Fetched: 3 in 0,227 seconds

ID	NUME	COMPANIE
4002	Amoxicilină	DEF Pharma
4013	Metoprolol	KLM Company
4015	Ranitidină	QRS Drugs

```
select m.id, m.nume, m.companie
from medicamente m
where m.id in (
            select r.id_medicament
            from retete r
            join diagnostice d on r.id_diagnostic = d.id
            where lower(d.nume) like 'gripă');
```

Executarea operației de actualizare a datelor:

```sql
update medicamente
set companie = 'Gripovit'
where id in (
    select r.id_medicament
    from retete r
    join diagnostice d on r.id_diagnostic = d.id
    where lower(d.nume) like 'gripă'
);
```

```
3 rows updated.
```

update medicamente
set companie = 'Gripovit'
where id in (
  select r.id_medicament
  from retete r
  join diagnostice d on r.id_diagnostic = d.id
  where lower(d.nume) like 'gripă'
);

Companiile medicamentelor utilizate în tratamentul diagnosticului ‘Gripa’, **după** actualizare:

	ID	NUME	COMPANIE
1	4002	Amoxicilină	Gripovit
2	4013	Metoprolol	Gripovit
3	4015	Ranitidină	Gripovit

3. Operație de ștergere

Să se șteargă toate secțiile pe care nu lucrează niciun medic.

Toate secțiile, **înainte** de ștergere:

```
select *
from sectii;
```

	ID	NUME	NR_PATURI	ID_SEF	ID_SPITAL
1	2000	Cardiologie	20	(null)	200
2	2001	Ortopedie	15	(null)	201
3	2002	Ginecologie	25	(null)	202
4	2004	Pediatrie	40	(null)	204
5	2005	Endocrinologie	18	(null)	205
6	2006	Oftalmologie	12	(null)	201
7	2007	Dermatologie	16	(null)	202
8	2011	Neurologie	30	(null)	200
9	2012	Urologie	22	(null)	200
10	2013	Oncologie	10	(null)	200
11	2014	Pneumologie	20	(null)	201
12	2015	Cardiologie	16	(null)	200
13	2016	Neurologie	32	(null)	200

select *
from sectii;

Secțiile pe care nu lucrează niciun medic:

```
select *
from sectii
where id not in (
                select distinct id_sectie
                from medici
);
```

	ID	NUME	NR_PATURI	ID_SEF	ID_SPITAL
1	2013	Oncologie	10	(null)	200
2	2014	Pneumologie	20	(null)	201
3	2015	Cardiologie	16	(null)	200
4	2016	Neurologie	32	(null)	200

select *
from sectii
where id not in (
        select distinct id_sectie
        from medici

);

Executarea operației de ștergere:

```
delete from sectii
where id not in (
    select distinct id_sectie
    from medici
);
```

```
4 rows deleted.
```

where id not in (
  select distinct id_sectie
  from medici
);

Toate secțiile, **după** ștergere:

	ID	NUME	NR_PATURI	ID_SEF	ID_SPITAL
1	2000	Cardiologie	20	(null)	200
2	2001	Ortopedie	15	(null)	201
3	2002	Ginecologie	25	(null)	202
4	2004	Pediatrie	40	(null)	204
5	2005	Endocrinologie	18	(null)	205
6	2006	Oftalmologie	12	(null)	201
7	2007	Dermatologie	16	(null)	202
8	2011	Neurologie	30	(null)	200
9	2012	Urologie	22	(null)	200

15. Implementarea următoarelor 3 cereri:

- **o cerere ce utilizează operația division**
- **o cerere care implementează analiza top-n**
- **o cerere care utilizeaza operația outer-join pe minimum 4 tabele**

O cerere care utilizează operația division.

Să se afișeze codurile și numele medicamentelor care au fost prescrise în toate consultațiile care au avut loc în ultima zi din luna iunie a anului precedent.

```
--varianta 1
select m.id, m.nume
from medicamente m
where not exists (
    select *
    from consultatii c
    where c.data_examinarii = to_date('30.06.' || to_char(extract(year from sysdate) - 1), 'DD.MM.YYYY')
    and not exists (
          select *
          from retete r
          where r.id_diagnostic = c.id_diagnostic
          and r.id_medicament = m.id )
);
```

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 1 in 0,002 seconds

	ID	NUME
1	4002	Amoxicilină

```
select m.id, m.nume
from medicamente m
where not exists (
                select *
                from consultatii c
                where c.data_examinarii = to_date('30.06.' || to_char(extract(year from
            sysdate) - 1), 'DD.MM.YYYY')
```

```
                and not exists (
                        select *
                        from retete r
                        where r.id_diagnostic = c.id_diagnostic
                        and r.id_medicament = m.id )
);
```

```
--varianta 2
select m.id, m.nume
from medicamente m
where m.id in (
    select r.id_medicament
    from retete r
    join diagnostice d on r.id_diagnostic = d.id
    join consultatii c on c.id_diagnostic = d.id
    where c.data_examinarii = to_date('30.06.' || to_char(extract(year from sysdate) - 1), 'DD.MM.YYYY')
    group by r.id_medicament
    having count(distinct c.id) = (
        select count(*)
        from consultatii
        where data_examinarii = to_date('30.06.' || to_char(extract(year from sysdate) - 1), 'DD.MM.YYYY')
    )
);
```

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 1 in 0,088 seconds

	ID	NUME
1	4002	Amoxicilină

```
select m.id, m.nume
from medicamente m
where m.id in (
   select r.id_medicament
   from retete r
   join diagnostice d on r.id_diagnostic = d.id
   join consultatii c on c.id_diagnostic = d.id
   where c.data_examinarii = to_date('30.06.' || to_char(extract(year from sysdate) - 1),
'DD.MM.YYYY')
   group by r.id_medicament
   having count(distinct c.id) = (
      select count(*)
      from consultatii
      where data_examinarii = to_date('30.06.' || to_char(extract(year from sysdate) - 1),
'DD.MM.YYYY')
   )
```

);

O cerere care implementează analiza top-n.

Să se afișeze primele 2 spitale care au cel mai mare număr de servicii cu ambulanța efectuate.

```
with aux as
      (select a.id_spital, count(a.id_spital) nr_serv
       from servicii_amb sa
       join ambulante a on a.id = sa.id_ambulanta
       group by a.id_spital
)
select *
from (
  select s.id, s.nume, aux.nr_serv
  from spitale s
  join aux on s.id = aux.id_spital
  order by aux.nr_serv desc
)
where rownum <= 2;
```

Script Output × | Query Result × | Query Result 1 × | Query Result 2 ×

SQL | All Rows Fetched: 2 in 0,003 seconds

	ID_SPITAL	NUME	NR_SERV
1	205	SPITALUL JUD DE URGENTA SF. SPIRIDON	5
2	200	SPITALUL JUD DE URGENTA SF. IOAN CEL NOU	4

with aux as

(select a.id_spital, count(a.id_spital) nr_serv

from servicii_amb sa

```
            join ambulante a on a.id = sa.id_ambulanta
            group by a.id_spital )
select *
from (
  select s.id, s.nume, aux.nr_serv
  from spitale s
  join aux on s.id = aux.id_spital
  order by aux.nr_serv desc
  )
where rownum <= 2;
```

O cerere care utilizează operația outer-join pe minimum 4 tabele.

Sa se afiseze toți pacienții, inclusiv cei pentru care nu este înregistrată nicio consultatie, iar pentru cei care au înregistrate consultații, să se afișeze data examinării și id-ul medicului, precum și diagnosticul asociat consultațiilor, rețetele pentru medicamentele incluse în tratament, în cazul în care acesta este de tip medicamentos, iar pentru cei care nu au tratament de tip medicamentos să se afișeze null în toate câmpurile destinate rețetelor.

Să se afișeze rezultatele crescător în funcție de data la care a avut loc consultația, în funcție de id-ul pacientului și de tipul tratamentului.

```
select p.id, p.nume || ' ' || p.prenume nume_pacient, c.data_examinarii, c.id_medic,
d.nume nume_diagnostic, d.tip_tratament, r.id_medicament,m.nume nume_medicament, r.doza, r.durata
from pacienti p
left join consultatii c on p.id = c.id_pacient
left join diagnostice d on d.id = c.id_diagnostic
left join retete r on r.id_diagnostic = d.id
left join medicamente m on m.id = r.id_medicament
order by c.data_examinarii, p.id, d.tip_tratament;
```

Script Output | Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 130 in 0,121 seconds

	ID	NUME_PACIENT	DATA_EXAMINARII	ID_MEDIC	NUME_DIAGNOSTIC	TIP_TRATAMENT	ID_MEDICAMENT	NUME_MEDICAMENT	DOZA	DURATA
1	164	Gavril Iulia	01-11-2010	1414	Insuficiență cardiacă	medicamentos	4014	Rosuvastatină	15 mg	14 zile
2	164	Gavril Iulia	01-11-2010	1414	Insuficiență cardiacă	medicamentos	4003	Omeprazol	20 mg	21 zile
3	165	Voinea Ionut	02-11-2013	1415	Osteoporoză	exercitii fizice	(null)	(null)	(null)	(null)
4	166	Radulescu Simona	03-11-2015	1416	Anemie	suplimente alimentare	4005	Simvastatină	12 mg	7 zile
5	166	Radulescu Simona	03-11-2015	1416	Anemie	suplimente alimentare	4019	Ciprofloxacină	12 mg	14 zile
6	166	Radulescu Simona	03-11-2015	1416	Anemie	suplimente alimentare	4008	Cetirizină	15 mg	21 zile
7	166	Radulescu Simona	03-11-2015	1416	Anemie	suplimente alimentare	4016	Clonazepam	18 mg	14 zile
8	159	Neagu Andreea	27-10-2018	1409	Astm bronșic	medicamentos, inhalatoare	4009	Loratadină	10 mg	14 zile
9	158	Munteanu Sorin	26-09-2019	1408	Depresie	terapie	(null)	(null)	(null)	(null)
10	156	Marin Adrian	24-10-2019	1406	Migrenă	medicamentos, repaus	4020	Xanax	6 mg	8 zile
11	156	Marin Adrian	24-10-2019	1406	Migrenă	medicamentos, repaus	4006	Metformină	18 mg	14 zile
12	156	Marin Adrian	24-10-2019	1406	Migrenă	medicamentos, repaus	4017	Levotiroxină	25 mg	10 zile
13	151	Constantin Laura	19-08-2020	1401	Hipertensiune arterială	medicamentos	4004	Aspirină	8 mg	14 zile

80	155	Vasilescu Cristina	23-05-2023	1405	Anemie	suplimente alimentare	4016	Clonazepam	18 mg	14 zile
81	155	Vasilescu Cristina	23-05-2023	1405	Anemie	suplimente alimentare	4008	Cetirizină	15 mg	21 zile
82	155	Vasilescu Cristina	23-05-2023	1405	Anemie	suplimente alimentare	4019	Ciprofloxacină	12 mg	14 zile
83	102	Apostol Maria	(null)	(n...	(null)	(null)	(null)	(null)	(null)	(null)
84	103	Radu Ion	(null)	(n...	(null)	(null)	(null)	(null)	(null)	(null)
85	104	Dumitru Elena	(null)	(n...	(null)	(null)	(null)	(null)	(null)	(null)
86	105	Stancu Andrei	(null)	(n...	(null)	(null)	(null)	(null)	(null)	(null)
87	106	Constantin Laura	(null)	(n...	(null)	(null)	(null)	(null)	(null)	(null)
88	107	Florescu Alexandru	(null)	(n...	(null)	(null)	(null)	(null)	(null)	(null)
89	108	Gheorghe Ioana	(null)	(n...	(null)	(null)	(null)	(null)	(null)	(null)
90	109	Stan Daniel	(null)	(n...	(null)	(null)	(null)	(null)	(null)	(null)

```
select p.id, p.nume || ' ' || p.prenume nume_pacient, c.data_examinarii, c.id_medic,
d.nume nume_diagnostic, d.tip_tratament, r.id_medicament,m.nume
nume_medicament, r.doza, r.durata
from pacienti p
left join consultatii c on p.id = c.id_pacient
left join diagnostice d on d.id = c.id_diagnostic
left join retete r on r.id_diagnostic = d.id
left join medicamente m on m.id = r.id_medicament
order by c.data_examinarii, p.id, d.tip_tratament;
```