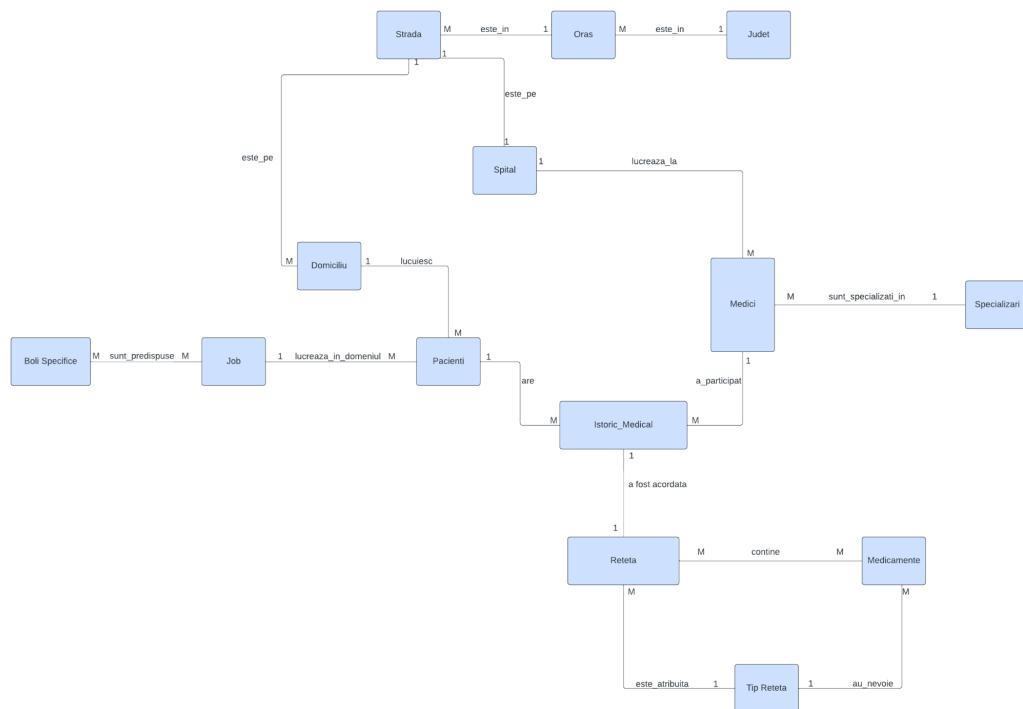


## Proiect SGBD

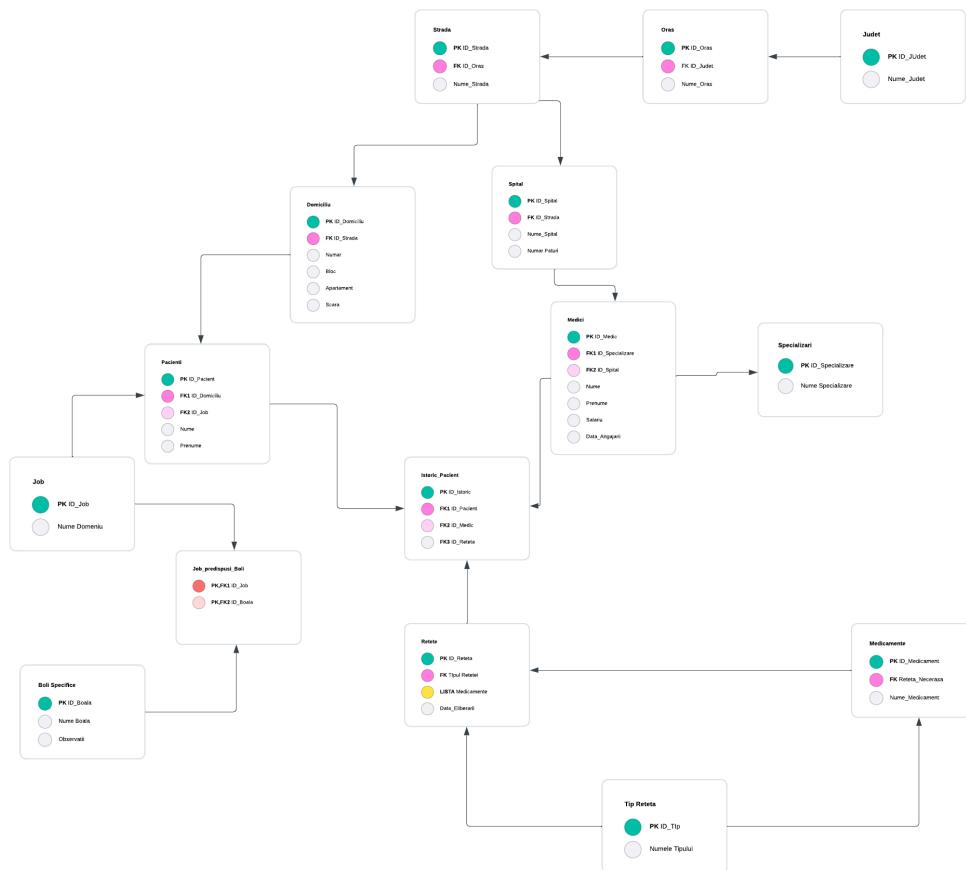
### 1. Prezentați pe scurt baza de date (utilitatea ei)

Baza de date este destinată gestionării informațiilor în cadrul unui sistem de sănătate. Aceasta are mai multe spitale din toată țara, medici care lucrează în diferite spitale. Totodată există pacienți care au un istoric medical strâns legat de un medic și o rețetă unică eliberată de acesta.

### 2. Realizați diagrama entitate-relație (ERD): entitățile, relațiile și atributele trebuie definite în limba română (vezi curs SGBD / model de diagrama ERD; nu se va accepta alt format)



3. Pornind de la diagrama entitate-relație realizată diagrama conceptuală a modelului propus, integrând toate atributele necesare: entitățile, relațiile și atributele trebuie definite în limba română.



4. Implementați în Oracle diagrama conceptuală realizată: definiți toate tabelele, definind toate constrângerile de integritate necesare (chei primare, cheile externe etc).

| <b>Constrangeri</b>  | <b>Definirea tabelelor</b>   |
|--|--|
| Numele specializării nu are cum să fie null și trebuie să fie unic în tabela specializări. | <pre> CREATE TABLE SPECIALIZARI(     ID_SPECIALIZARE INT PRIMARY KEY,     NUME_SPECIALIZARE VARCHAR2(50) NOT NULL unique ); </pre> |
| Nu pot să apara două județe cu același nume  | <pre> CREATE TABLE judete (     ID_JUDET INT PRIMARY KEY, </pre>   |

|   |   |
|---|---|
|   | <pre>NUME_JUDET VARCHAR2(50) NOT NULL unique );</pre>   |
| Cheia externă ID_Judet nu poate sa fie null pentru ca nu există orașe care să nu facă parte dintr-un județ. Numele orașului poate să se repete fiind într-un județ diferit.   | <pre>CREATE TABLE orase( ID_ORAS INT PRIMARY KEY, ID_JUDET INT NOT NULL, NUME_ORAS VARCHAR2(50) NOT NULL, FOREIGN KEY (ID_JUDET) REFERENCES judete (ID_JUDET) );</pre>  |
| Cheia externă ID_Oras nu poate sa fie null pentru ca nu există străzi care să nu facă parte dintr-un oraș. Numele străzii poate să se repete fiind într-un oraș diferit.  | <pre>CREATE TABLE Strazi( ID_STRADA INT PRIMARY KEY, NUME_STRADA VARCHAR2(50) NOT NULL, ID_ORAS INT NOT NULL, FOREIGN KEY (ID_ORAS) REFERENCES orase (ID_ORAS) );</pre>   |
| Un spital are obligatoriu un nume și o strada pe care se află. Pentru numărul de pături am adăugat default 20 de paturi   | <pre>CREATE TABLE SPITALE ( ID_SPITAL INT PRIMARY KEY, ID_STRADA INT NOT NULL, NUME_SPITAL VARCHAR2(100) NOT NULL, NUMAR_PATURI INT DEFAULT 20, FOREIGN KEY (ID_STRADA) REFERENCES Strazi(ID_STRADA) );</pre>   |
| Medicii care nu au o specializare sunt directori ai spitalului respectiv, dar ei au un nume, prenume și un spital la care fac parte. Salariul default pe care îl poate primii un medic este de 2.500 RON, iar data angajării dacă aceasta nu este specificată este data de azi. | <pre>CREATE TABLE MEDICI( ID_MEDIC INT PRIMARY KEY, ID_SPECIALIZARE INT, ID_SPITAL INT NOT NULL, NUME VARCHAR2(50) NOT NULL, PRENUME VARCHAR2(50) NOT NULL, SALARIU INT DEFAULT 2500, DATA_ANGAJARI DATE DEFAULT SYSDATE, FOREIGN KEY (ID_SPECIALIZARE) REFERENCES SPECIALIZARI (ID_SPECIALIZARE), FOREIGN KEY (ID_SPITAL) REFERENCES SPITAL (ID_SPITAL) );</pre> |
| Orice domiciliu are un număr și o strada pe care se află, dar există și cazurile în care persoana respectivă locuiește la bloc și atunci este nevoie de numărul blocului, apartament și scara.  | <pre>CREATE TABLE Domiciliu( ID_DOMICILIU INT PRIMARY KEY, ID_STRADA INT NOT NULL , NUMAR INT not null, BLOC INT default NULL, APARTAMENT INT default NULL, SCARA VARCHAR(1) default NULL, FOREIGN KEY (ID_STRADA) REFERENCES Strazi (ID_STRADA) );</pre>   |

|   |  |
|---|--|
| Orice Job are o denumire specifică și unică   | <pre>CREATE TABLE Joburi (     ID_JOB INT PRIMARY KEY,     NUME_DOMENIU VARCHAR2(100) NOT NULL );</pre>  |
| Bolile specifice au un nume unic și o observație medicală (cu ce se poate trata, simptome, descriere a bolii)   | <pre>CREATE TABLE BOLI_SPECIFICE(     ID_BOALA INT PRIMARY KEY,     NUME_BOALA VARCHAR2(100) NOT NULL unique,     OBSERVATII VARCHAR2(100) DEFAULT NULL );</pre>   |
| Relația dintre Job și Boli Specifice este o relație de tip many-to-many aceasta a fost rezolvată cu o tabelă asociativă. Fiind o tabelă asociativă aceasta are o cheie primară compusă.                                     | <pre>CREATE TABLE JOB_PREDISPUSSI_BOLI(     ID_BOALA INT,     ID_JOB INT,     FOREIGN KEY (ID_BOALA) REFERENCES BOLI_SPECIFICE(ID_BOALA),     FOREIGN KEY (ID_JOB) REFERENCES JOB(ID_JOB),     PRIMARY KEY (ID_JOB, ID_BOALA));</pre>  |
| Un pacient trebuie să aibă nume, prenume și domiciliu. Există persoane care nu au un loc de munca ceea ce înseamnă că nu au un job.   | <pre>CREATE TABLE PACIENTI(     ID_PACIENT INT PRIMARY KEY,     ID_DOMICILIU INT NOT NULL,     ID_JOB INT DEFAULT NULL,     NUME VARCHAR2(50) NOT NULL ,     PRENUME VARCHAR2(50) NOT NULL,     FOREIGN KEY (ID_JOB) REFERENCES JOB(ID_JOB),     FOREIGN KEY (ID_DOMICILIU) REFERENCES DOMICILIU(ID_DOMICILIU));</pre> |
| Retetele au un anumit tip care poate să fie eliberat de persoane specializate în domeniu  | <pre>CREATE TABLE TIP_RETETE(     ID_TIP INT PRIMARY KEY,     NUMELE_TIPULUI VARCHAR2(50) NOT NULL );</pre>  |
| Reteta conține un data eliberării, dacă nu e specificată atunci data este cea curentă.<br>O reteta conține medicamente, am decis că id-urile medicamentelor să fie tinute într-o varray pentru a putea fi accesate mai usor | <pre>CREATE TABLE RETETE(     ID_RETETA INT PRIMARY KEY,     TIP_RETETA INT,     DATA_ELIBERARII DATE DEFAULT SYSDATE,     FOREIGN KEY (TIP_RETETA) REFERENCES TIP_RETETA(ID_TIP) ); create or replace type Lista as varray(20) of number(2); alter table RETETE add Lista_Medicamente Lista;</pre>                    |
| Fiecare medicament are un tip de reteta pentru care poate să fie eliberat, nu există medicamente fără nume.   | <pre>CREATE TABLE MEDICAMENTE (     ID_MEDICAMENT INT PRIMARY KEY,     TIP_RETETA INT,     NUME_MEDICAMENT VARCHAR2(100) NOT NULL,     FOREIGN KEY (TIP_RETETA) REFERENCES TIP_RETETA(ID_TIP));</pre>  |
| Fiecare pacient are un istoric medical,   | <pre>CREATE TABLE ISTORIC_PACIENTI (</pre>   |

pentru a putea tine evidenta a ce pastile a consumat si ce boli a avut in trecut.  
Fiecare istoric medical trebuie sa aiba un pacient, medic, si o reteta ce i-a fost eliberata.

```

ID_ISTORIC INT PRIMARY KEY,
ID_PACIENT INT NOT NULL ,
ID_MEDIC INT NOT NULL ,
ID_RETETA INT NOT NULL UNIQUE ,
FOREIGN KEY (ID_PACIENT) REFERENCES
PACIENTI(ID_PACIENT),
FOREIGN KEY (ID_MEDIC) REFERENCES
MEDICI(ID_MEDIC),
FOREIGN KEY (ID_RETETA) REFERENCES
RETETE(ID_RETETA)
);

```

5. Adăugați informații coerente în tabelele create (minim 5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru tabela asociativă).

- Pentru anumite tabele am folosit și sequence uri pentru a ușura munca. În tabela de mai jos sunt arătate toate sequence urile folosite pentru inserturi.

| <i>Judet</i>  | <i>Oras</i>  | <i>Strada</i>  | <i>Istoric Medical</i>  |
|---|--|--|---|
| create sequence<br>seq_judet<br>start with 1<br>increment by 1; | create sequence<br>seq_oras<br>start with 1<br>increment by 1; | create sequence<br>seq_strada<br>start with 1<br>increment by 1; | create sequence<br>seq_istoric<br>start with 1<br>increment by 1; |

Inserturi pentru tabela Județ:

```

insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Alba');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Arad');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Arges');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Bacau');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Bihor');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Bistrita-Nasaud');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Botosani');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Brasov');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Braila');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Bucuresti');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Buzau');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Caras-Severin');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Calarasi');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Cluj');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Constanta');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Covasna');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Dambovita');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Dolj');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Galati');

```

```

insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Giurgiu');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Gorj');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Harghita');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Hunedoara');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Ialomita');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Iasi');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Ilfov');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Maramures');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Mehedinti');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Mures');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Neamt');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Olt');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Prahova');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Satu Mare');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Salaj');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Sibiu');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Suceava');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Teleorman');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Timis');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Tulcea');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Vaslui');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Valcea');
insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Vrancea');

```

```

199  insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Tuicea');
200  insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Vaslui');
201  insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Valcea');
202  insert into judete(id_judet, nume_judet)VALUES (seq_judet.nextval, 'Vrancea');
203  COMMIT;
204  ✓ select * from judete;
205
206
207
208
209  insert into Strazi(id_strada, nume_strada) values (seq_strada.nextval,'Strada Stefan cel Mare'); -- Bucuresti si Vaslui, Braila
210  insert into Strazi(id_strada, nume_strada) values (seq_strada.nextval,'Soseaua Mihai Bravu'); -- Bucuresti
211  insert into Strazi(id_strada, nume_strada) values (seq_strada.nextval,'Strada Hrisovului'); -- Bucuresti
212  insert into Strazi(id_strada, nume_strada) values (seq_strada.nextval,'Strada Exercitiului'); -- Pitesti
213  insert into Strazi(id_strada, nume_strada) values (seq_strada.nextval,'Strada Nicolae Iorga'); -- Pitesti, Bucuresti, Bacau
214  insert into Strazi(id_strada, nume_strada) values (seq_strada.nextval,'Strada Mihail Kogalniceanu'); -- Vaslui,Arad
215  insert into Strazi(id_strada, nume_strada) values (seq_strada.nextval,'Strada Grivitei'); -- Bucuresti, Braila
216  insert into Strazi(id_strada, nume_strada) values (seq_strada.nextval,'Strada Titu Maiorescu'); -- Voluntari

```

| ID_JUDET | NUME_JUDET |
|----------|------------|
| 1        | Alba       |
| 2        | Arad       |
| 3        | Arges      |
| 4        | Bacau      |
| 5        | Bihor      |

### Inserturi pentru tabela Oras:

```

INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval,(SELECT ID_JUDET FROM
judete WHERE nume_judet = 'Arges'), 'Pitesti');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval,(SELECT ID_JUDET FROM
judete WHERE nume_judet = 'Alba'), 'Alba Iulia');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval,(SELECT ID_JUDET FROM
judete WHERE nume_judet = 'Bacau'), 'Bacau');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval,(SELECT ID_JUDET FROM
judete WHERE nume_judet = 'Arad'), 'Arad');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval,(SELECT ID_JUDET FROM
judete WHERE nume_judet = 'Bihor'), 'Oradea');

```

```

INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Botosani'), 'Botosani');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Bistrita-Nasaud'), 'Bistrita');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Braila'), 'Braila');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Brasov'), 'Brasov');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Satu Mare'), 'Satu Mare');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Caras-Severin'), 'Resita');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Calarasi'), 'Calarasi');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Prahova'), 'Ploiesti');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Cluj'), 'Cluj-Napoca');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Bucuresti'), 'Bucuresti');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Constanta'), 'Constanta');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Buzau'), 'Buzau');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Dambovita'), 'Targoviste');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Covasna'), 'Sfantu Gheorghe');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Dolj'), 'Craiova');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Gorj'), 'Targu Jiu');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Harghita'), 'Miercurea Ciuc');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Hunedoara'), 'Deva');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Ialomita'), 'Slobozia');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Iasi'), 'Iasi');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Ilfov'), 'Voluntari');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Maramures'), 'Baia Mare');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Mehedinti'), 'Drobeta-Turnu Severin');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Mures'), 'Targu Mures');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Neamt'), 'Piatra Neamt');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Olt'), 'Slatina');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Salaj'), 'Zalau');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Sibiu'), 'Sibiu');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval, (SELECT ID_JUDET FROM
                judete WHERE nume_judet = 'Valcea'), 'Ramnicu Valcea');

```

```

INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval,(SELECT ID_JUDET FROM
judete WHERE nume_judet = 'Vrancea'), 'Focșani');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval,(SELECT ID_JUDET FROM
judete WHERE nume_judet = 'Suceava'), 'Suceava');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval,(SELECT ID_JUDET FROM
judete WHERE nume_judet = 'Teleorman'), 'Alexandria');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval,(SELECT ID_JUDET FROM
judete WHERE nume_judet = 'Timiș'), 'Timișoara');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval,(SELECT ID_JUDET FROM
judete WHERE nume_judet = 'Tulcea'), 'Tulcea');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval,(SELECT ID_JUDET FROM
judete WHERE nume_judet = 'Vaslui'), 'Vaslui');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval,(SELECT ID_JUDET FROM
judete WHERE nume_judet = 'Galati'), 'Galati');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval,(SELECT ID_JUDET FROM
judete WHERE nume_judet = 'Giurgiu'), 'Giurgiu');

INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval,(SELECT ID_JUDET FROM judete WHERE nume_judet = 'Suceava'), 'Suceava');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval,(SELECT ID_JUDET FROM judete WHERE nume_judet = 'Teleorman'), 'Alexandria');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval,(SELECT ID_JUDET FROM judete WHERE nume_judet = 'Timiș'), 'Timișoara');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval,(SELECT ID_JUDET FROM judete WHERE nume_judet = 'Tulcea'), 'Tulcea');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq_oras.nextval,(SELECT ID_JUDET FROM judete WHERE nume_judet = 'Vaslui'), 'Vaslui');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq.oras.nextval,(SELECT ID_JUDET FROM judete WHERE nume_judet = 'Galati'), 'Galati');
INSERT INTO orase (ID_ORAS, ID_JUDET, nume_oras) VALUES (seq.oras.nextval,(SELECT ID_JUDET FROM judete WHERE nume_judet = 'Giurgiu'), 'Giurgiu');

Commit;
✓ select * from orase;
delete orase;

```

*-- Arad*

Output ORASE

| ID_ORAS | ID_JUDET | NUME_ORAS    |
|---------|----------|--------------|
| 4       | 2        | 1 Alba Iulia |
| 5       | 3        | 4 Bacău      |
| 6       | 4        | 2 Arad       |
| 7       | 5        | 5 Oradea     |

### Inserturi pentru tabela Strada:

```

-- Arad
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Arad'), 'Bulevardul Revoluției');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Arad'), 'Strada Vasile Goldiș');

-- Arges
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Pitești'), 'Bulevardul Nicolae Balcescu');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Pitești'), 'Strada Alexandru Odobescu');

-- Alba Iulia
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Alba Iulia'), 'Bulevardul Regele Ferdinand');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Alba Iulia'), 'Strada Avram Iancu');

-- Bacău

```

```

INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Bacau'), 'Bulevardul Unirii');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Bacau'), 'Strada Mihai Eminescu');

-- Bistrita-Nasaud
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Bistrita'), 'Bulevardul Decebal');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Bistrita'), 'Strada Liviu Rebreanu');

-- Bihor
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Oradea'), 'Bulevardul Dacia');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Oradea'), 'Strada Republicii');

-- Botosani
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Botosani'), 'Bulevardul Mihai Eminescu');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Botosani'), 'Strada Stefan cel Mare');

-- Brasov
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Brasov'), 'Bulevardul Eroilor');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Brasov'), 'Strada Republicii');

-- Buzau
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Buzau'), 'Bulevardul Independentei');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Buzau'), 'Strada Mihai Viteazu');

-- Braila
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Braila'), 'Bulevardul Dorobantilor');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Braila'), 'Strada Mihail Kogalniceanu');

-- Bucuresti
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Bucuresti'), 'Bulevardul Unirii');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Bucuresti'), 'Calea Victoriei');

-- Caras-Severin
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Resita'), 'Bulevardul 1 Decembrie');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Resita'), 'Strada Avram Iancu');

-- Calarasi
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Calarasi'), 'Bulevardul Dorobantilor');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Calarasi'), 'Strada 1 Mai');

```

```
-- Cluj
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Cluj-Napoca'), 'Bulevardul Eroilor');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Cluj-Napoca'), 'Strada Horea');

-- Constanta
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Constanta'), 'Bulevardul Mamaia');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Constanta'), 'Strada Tomis');

-- Dolj
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Craiova'), 'Bulevardul Carol I');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Craiova'), 'Strada Madona Dudu');

-- Galati
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Galati'), 'Bulevardul Dorobantilor');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Galati'), 'Strada Eroilor');
-- Covasna
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Sfantu Gheorghe'), 'Bulevardul Mihai Viteazu');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Sfantu Gheorghe'), 'Strada Nicolae Balcescu');

-- Dambovita
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Targoviste'), 'Bulevardul Carol I');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Targoviste'), 'Strada Matei Basarab');

-- Giurgiu
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Giurgiu'), 'Bulevardul Unirii');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Giurgiu'), 'Strada Eroilor');

-- Harghita
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Miercurea Ciuc'), 'Bulevardul Harghita');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Miercurea Ciuc'), 'Strada Unirii');
-- Gorj
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Targu Jiu'), 'Bulevardul Ecaterina Teodoroiu');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Targu Jiu'), 'Strada General Magheru');

-- Hunedoara
```

```

INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Deva'), 'Bulevardul 1 Decembrie');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Deva'), 'Strada Mihai Eminescu');

-- Ialomița
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Slobozia'), 'Bulevardul Independentei');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Slobozia'), 'Strada Mihail Kogălniceanu');

-- Maramureș
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Baia Mare'), 'Bulevardul Republicii');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Baia Mare'), 'Strada Mihai Viteazu');

-- Iasi
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Iasi'), 'Bulevardul Stefan cel Mare');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Iasi'), 'Strada Palat');

-- Ilfov
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Bucuresti'), 'Bulevardul Pipera');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Bucuresti'), 'Strada Bucuresti');

-- Târgu Mureș
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Târgu Mureș'), 'Strada Avram Iancu');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Târgu Mureș'), 'Strada Unirii');

-- Piatra Neamț
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Piatra Neamț'), 'Bulevardul Decebal');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Piatra Neamț'), 'Strada Stefan cel Mare');

-- Slatina
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Slatina'), 'Bulevardul Republicii');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Slatina'), 'Strada Mihai Viteazu');

-- Ploiești
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Ploiești'), 'Bulevardul Independentei');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Ploiești'), 'Strada Gheorghe Doja');

-- Satu Mare
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Satu Mare'), 'Strada Horea');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Satu Mare'), 'Bulevardul Vasile Lucaciu');

```

```
-- Zalau
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Zalau'), 'Bulevardul Mihai Viteazul');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Zalau'), 'Strada Avram Iancu');

-- Sibiu
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Sibiu'), 'Bulevardul Victoriei');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Sibiu'), 'Strada Nicolae Balcescu');

-- Suceava
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Suceava'), 'Bulevardul George Enescu');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Suceava'), 'Strada Stefan cel Mare');

-- Alexandria
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Alexandria'), 'Bulevardul 1 Decembrie');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Alexandria'), 'Strada Alexandru Ioan Cuza');

-- Timisoara
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Timisoara'), 'Bulevardul Revolutiei');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Timisoara'), 'Strada Marasesti');

-- Tulcea
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Tulcea'), 'Bulevardul Republicii');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Tulcea'), 'Strada Isaccei');

-- Vaslui
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Vaslui'), 'Bulevardul Stefan cel Mare');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Vaslui'), 'Strada Cuza Voda');

-- Ramnicu Valcea
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Ramnicu Valcea'), 'Bulevardul General Praporgescu');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Ramnicu Valcea'), 'Strada Calea lui Traian');

-- Focsani
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Focsani'), 'Bulevardul Unirii');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras
FROM orase WHERE nume_oras = 'Focsani'), 'Strada Cuza Voda');
```

```
-- Ramnicu Valcea
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras FROM orase WHERE nume_oras = 'Ramnicu Valcea'), 'Bulevardul General Praporgescu');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras FROM orase WHERE nume_oras = 'Ramnicu Valcea'), 'Strada Calea lui Traian');

-- Focșani
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras FROM orase WHERE nume_oras = 'Focșani'), 'Bulevardul Unirii');
INSERT INTO Strazi (id_strada,id_oras, nume_strada) VALUES (seq_strada.nextval,(SELECT id_oras FROM orase WHERE nume_oras = 'Focșani'), 'Strada Cuza Vodă');

Commit;
select * from Strazi;
-- Primele 20 de inserții
select * from Domiciliu;
```

| ID_STRADA | NUME_STRADA                   | ID_ORAS |
|-----------|-------------------------------|---------|
| 3         | 2 Strada Vasile Goldis        | 4       |
| 4         | 3 Bulevardul Nicolae Balcescu | 1       |
| 5         | 4 Strada Alexandru Odobescu   | 1       |
| 6         | 5 Bulevardul Regele Ferdinand | 2       |
| 7         | 6 Strada Avram Iancu          | 2       |

Inserturi pentru tabela Domiciliu:

```
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (1, 1, 12, 6, 3, 'B');
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (2, 2, 45, NULL, NULL, NULL);
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (3, 3, 78, NULL, NULL, NULL);
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (4, 4, 9, 11, 15, 'G');
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (5, 5, 23, 8, 7, 'I');
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (6, 6, 56, NULL, NULL, NULL);
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (7, 7, 34, 10, 8, 'E');
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (8, 8, 67, NULL, NULL, NULL);
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (9, 9, 41, 7, 5, 'C');
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (10, 10, 14, NULL, NULL, NULL);
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (11, 11, 88, 4, 12, 'G');
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (12, 12, 27, NULL, NULL, NULL);
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (13, 13, 19, 5, 6, 'D');
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (14, 14, 52, NULL, NULL, NULL);
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (15, 15, 63, 3, 9, 'B');
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (16, 16, 72, NULL, NULL, NULL);
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (17, 17, 31, 1, 4, 'F');
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (18, 18, 49, NULL, NULL, NULL);
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (19, 19, 77, 0, 11, 'E');
```

```
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (20, 20, 22, NULL, NULL, NULL);
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (21, 1, 56, NULL, NULL, NULL);
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (22, 2, 14, 1, 2, 'B');
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (23, 3, 29, 4, 7, 'E');
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (24, 4, 8, NULL, NULL, NULL);
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (25, 5, 41, 2, 5, 'C');
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (26, 6, 63, NULL, NULL, NULL);
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (27, 7, 22, NULL, NULL, NULL);
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (28, 8, 48, 3, 9, 'D');
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (29, 9, 77, 1, 12, 'B');
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (30, 10, 33, 5, 4, 'F');
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (31, 11, 19, NULL, NULL, NULL);
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (32, 12, 58, NULL, NULL, NULL);
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (33, 13, 12, 4, 3, 'E');
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (34, 14, 27, NULL, NULL, NULL);
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (35, 15, 39, NULL, NULL, NULL);
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (36, 16, 18, 2, 7, 'G');
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (37, 17, 51, NULL, NULL, NULL);
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (38, 18, 32, 1, 4, 'A');
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (39, 19, 67, NULL, NULL, NULL);
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (40, 20, 44, 3, 8, 'B');
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (41, 21, 23, NULL, NULL, NULL);
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (42, 22, 56, NULL, NULL, NULL);
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (43, 23, 11, 4, 6, 'C');
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (44, 24, 28, 1, 2, 'D');
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (45, 25, 49, NULL, NULL, NULL);
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (46, 26, 17, 3, 5, 'E');
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (47, 27, 32, 2, 8, 'F');
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (48, 28, 22, 4, 6, 'G');
```

```
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (49, 29, 14, NULL, NULL, NULL);
INSERT INTO Domiciliu (ID_DOMICILIU, ID_STRADA, NUMAR, BLOC, APARTAMENT, SCARA) VALUES (50, 30, 37, 1, 3, 'A');
```

The screenshot shows the MySQL Workbench interface. At the top, there is a query editor window with the following content:

```
select * from Domiciliu;
```

Below it is another window titled "Output" showing the results of the query:

|   | ID_DOMICILIU | ID_STRADA | NUMAR | BLOC   | APARTAMENT | SCARA         |
|---|--------------|-----------|-------|--------|------------|---------------|
| 1 | 1            | 1         | 12    | 6      |            | 3 B           |
| 2 | 2            | 2         | 45    | <null> |            | <null> <null> |
| 3 | 3            | 3         | 78    | <null> |            | <null> <null> |
| 4 | 4            | 4         | 9     | 11     |            | 15 6          |
| 5 | 5            | 5         | 23    | 8      |            | 7 I           |

### Inserturi pentru tabela Spital:

```
INSERT INTO SPITALE (ID_SPITAL, ID_STRADA, NUME_SPITAL, NUMAR_PATURI) VALUES (1, 55, 'Spitalul Municipal Targu Mures', 150);
INSERT INTO SPITALE (ID_SPITAL, ID_STRADA, NUME_SPITAL, NUMAR_PATURI) VALUES (2, 58, 'Spitalul Judetean Piatra Neamt', 200);
INSERT INTO SPITALE (ID_SPITAL, ID_STRADA, NUME_SPITAL, NUMAR_PATURI) VALUES (3, 60, 'Spitalul Municipal Slatina', 180);
INSERT INTO SPITALE (ID_SPITAL, ID_STRADA, NUME_SPITAL, NUMAR_PATURI) VALUES (4, 62, 'Spitalul de Urgenta Ploiesti', 250);
INSERT INTO SPITALE (ID_SPITAL, ID_STRADA, NUME_SPITAL, NUMAR_PATURI) VALUES (5, 64, 'Spitalul Judetean Satu Mare', 120);
INSERT INTO SPITALE (ID_SPITAL, ID_STRADA, NUME_SPITAL, NUMAR_PATURI) VALUES (6, 66, 'Spitalul Municipal Zalau', 100);
INSERT INTO SPITALE (ID_SPITAL, ID_STRADA, NUME_SPITAL, NUMAR_PATURI) VALUES (7, 68, 'Spitalul Clinic Judetean de Urgenta Sibiu', 300);
INSERT INTO SPITALE (ID_SPITAL, ID_STRADA, NUME_SPITAL, NUMAR_PATURI) VALUES (8, 70, 'Spitalul Judetean Suceava', 180);
INSERT INTO SPITALE (ID_SPITAL, ID_STRADA, NUME_SPITAL, NUMAR_PATURI) VALUES (9, 72, 'Spitalul Judetean Alexandria', 150);
INSERT INTO SPITALE (ID_SPITAL, ID_STRADA, NUME_SPITAL, NUMAR_PATURI) VALUES (10, 74, 'Spitalul Clinic Judetean de Urgenta Timisoara', 350);
INSERT INTO SPITALE (ID_SPITAL, ID_STRADA, NUME_SPITAL, NUMAR_PATURI) VALUES (11, 53, 'Spitalul Regina Maria', 550);
```

```

INSERT INTO SPITALE (ID_SPITAL, ID_STRADA, NUME_SPITAL, NUMAR_PATURI) VALUES (5, 64, 'Spitalul Județean Satu Mare', 120);
INSERT INTO SPITALE (ID_SPITAL, ID_STRADA, NUME_SPITAL, NUMAR_PATURI) VALUES (6, 66, 'Spitalul Municipal Zalau', 100);
INSERT INTO SPITALE (ID_SPITAL, ID_STRADA, NUME_SPITAL, NUMAR_PATURI) VALUES (7, 68, 'Spitalul Clinic Județean de Urgență Sibiu', 300);
INSERT INTO SPITALE (ID_SPITAL, ID_STRADA, NUME_SPITAL, NUMAR_PATURI) VALUES (8, 70, 'Spitalul Județean Suceava', 180);
INSERT INTO SPITALE (ID_SPITAL, ID_STRADA, NUME_SPITAL, NUMAR_PATURI) VALUES (9, 72, 'Spitalul Județean Alexandria', 150);
INSERT INTO SPITALE (ID_SPITAL, ID_STRADA, NUME_SPITAL, NUMAR_PATURI) VALUES (10, 74, 'Spitalul Clinic Județean de Urgență Timișoara', 350);
INSERT INTO SPITALE (ID_SPITAL, ID_STRADA, NUME_SPITAL, NUMAR_PATURI) VALUES (11, 55, 'Spitalul Regina Maria', 550);
COMMIT;
select * from SPITALE

```

| ID_SPITAL | ID_STRADA | NUME_SPITAL                       | NUMAR_PATURI |
|-----------|-----------|-----------------------------------|--------------|
| 1         | 1         | 55 Spitalul Municipal Târgu Mureș | 150          |
| 2         | 2         | 58 Spitalul Județean Piatra Neamț | 200          |
| 3         | 3         | 60 Spitalul Municipal Satu Mare   | 180          |
| 4         | 4         | 62 Spitalul de Urgență Ploiești   | 250          |
| 5         | 5         | 64 Spitalul Județean Satu Mare    | 120          |

### Inserturi pentru tabela Specializari:

```

INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (1, 'Cardiologie');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (2, 'Neurologie');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (3, 'Chirurgie Generală');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (4, 'Dermatologie');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (5, 'Gastroenterologie');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (6, 'Endocrinologie');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (7, 'Hematologie');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (8, 'Imunologie');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (9, 'Nefrologie');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (10, 'Oftalmologie');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (11, 'Ortopedie');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (12, 'Oncologie');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (13, 'Pediatrie');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (14, 'Psihiatrie');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (15, 'Radiologie');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (16, 'Reumatologie');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (17, 'Urologie');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (18, 'Pneumologie');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (19, 'Genetica Medicală');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (20, 'Medicina Internă');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (21, 'Medicina de Familie');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (22, 'Chirurgie Cardiovasculară');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (23, 'Chirurgie Plastica');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (24, 'Neonatologie');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (25, 'Geriatrie');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (26, 'Otorinolaringologie');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (27, 'Medicina de Urgență');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (28, 'Fiziatrie');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (29, 'Chirurgie Vasculară');
INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (30, 'Medicina Sportivă');

```

```

551 INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (26, 'Otorinolaringologie');
552 INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (27, 'Medicina de Urgenta');
553 INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (28, 'Fiziatrie');
554 INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (29, 'Chirurgie Vasculara');
555 INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (30, 'Medicina Sportiva');
556 COMMIT;
557 select * from SPECIALIZARI order by ID_SPECIALIZARE;
558
559
560
561
562
563
564
565
566
567

```

The screenshot shows the execution of an SQL script in a code editor and the resulting table output in a separate window.

**Code Editor:**

```

551 INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (26, 'Otorinolaringologie');
552 INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (27, 'Medicina de Urgenta');
553 INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (28, 'Fiziatrie');
554 INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (29, 'Chirurgie Vasculara');
555 INSERT INTO SPECIALIZARI (ID_SPECIALIZARE, NUME_SPECIALIZARE) VALUES (30, 'Medicina Sportiva');
556 COMMIT;
557 select * from SPECIALIZARI order by ID_SPECIALIZARE;
558
559
560
561
562
563
564
565
566
567

```

**Output Window:**

| ID_SPECIALIZARE | NUME_SPECIALIZARE  |
|-----------------|--------------------|
| 1               | Cardiologie        |
| 2               | Neurologie         |
| 3               | Chirurgie Generală |
| 4               | Dermatologie       |
| 5               | Gastroenterologie  |

### Inserturi pentru tabela Medici:

```

INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI)
VALUES (1, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Popescu',
'Ion', 3000, TO_DATE('2022-01-15', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI)
VALUES (2, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Ionescu',
'Ana', 2800, TO_DATE('2022-02-20', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI)
VALUES (3, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Dumitru',
'Maria', 2600, TO_DATE('2022-03-25', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI)
VALUES (4, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Georgescu',
'Mihai', 3200, (TO_DATE('2021-01-01', 'YYYY-MM-DD') + DBMS_RANDOM.VALUE *
(TO_DATE('2023-12-02', 'YYYY-MM-DD') - TO_DATE('2021-01-01', 'YYYY-MM-DD'))));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI)
VALUES (5, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Mihai',
'Diana', 3000, TO_DATE('2022-05-15', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI)
VALUES (6, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Stanescu',
'Gabriel', 2800, TO_DATE('2022-06-20', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI)
VALUES (7, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Cristea',
'Ana-Maria', 2600, TO_DATE('2022-07-25', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI)
VALUES (8, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Gheorghe',
'Ioan', 3200, TO_DATE('2022-08-10', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI)

```

```
VALUES (9, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Iancu',
'Elena', 3000, TO_DATE('2022-09-15', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI)
VALUES (10, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Vasilescu',
'Andrei', 2800, TO_DATE('2022-10-20', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(11, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Voicu', 'Roxana',
2905, TO_DATE('2023-03-9', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(12, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Coman', 'Alexandrina',
3652, TO_DATE('2011-06-2', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(13, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Dragomir', 'Sorin',
3322, TO_DATE('2015-08-25', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(14, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Florea', 'Robert',
3282, TO_DATE('2004-09-18', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(15, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Bogdan', 'Sorin',
4424, TO_DATE('2005-04-13', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(16, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Frumosu', 'Lavinia',
4094, TO_DATE('2016-06-4', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(17, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Bogdan', 'Silvian',
3333, TO_DATE('2012-09-12', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(18, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Sorin', 'Alexandru',
4529, TO_DATE('2003-07-27', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(19, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Serban', 'George',
3974, TO_DATE('2021-06-17', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(20, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Stefanescu', 'Elena',
4854, TO_DATE('2008-09-7', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(21, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Ardelean', 'Ion',
3445, TO_DATE('2015-06-11', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(22, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Tomescu', 'Adela',
3411, TO_DATE('2010-05-26', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(23, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Stanciu', 'Dana',
3843, TO_DATE('2014-07-17', 'YYYY-MM-DD'));
```

```
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(24, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Ardelean', 'Valentin',
3158, TO_DATE('2022-02-14', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(25, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Radulescu',
'Luminița', 4659, TO_DATE('2005-02-17', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(26, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Sorin', 'Marian',
4817, TO_DATE('2016-01-28', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(27, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Octavian', 'Vasile',
4488, TO_DATE('2006-05-10', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(28, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Badea', 'Loredana',
3054, TO_DATE('2004-08-18', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(29, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Oprea', 'Valeriu',
4585, TO_DATE('2012-01-7', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(30, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Cretu', 'Elena',
3790, TO_DATE('2001-09-5', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(31, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Gheorghita', 'Alina',
3407, TO_DATE('2015-04-22', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(32, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Grosu', 'Teodora',
3449, TO_DATE('2017-01-23', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(33, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Oprea', 'Andrei',
4755, TO_DATE('2002-08-26', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(34, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Dobre', 'Alexandrina',
3712, TO_DATE('2012-04-18', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(35, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Sava', 'Răzvan',
3034, TO_DATE('2020-09-24', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(36, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Cretu', 'Sorin',
4438, TO_DATE('2004-09-20', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(37, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Tomescu', 'Radu',
3945, TO_DATE('2016-06-15', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
```

```
(38, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Nistor', 'Anca',
3981, TO_DATE('2018-02-21', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(39, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Ghitescu', 'Marius',
3703, TO_DATE('2012-08-19', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(40, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Ardelean', 'Eduard',
4983, TO_DATE('2021-07-19', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(41, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Lucian', 'Octavian',
4889, TO_DATE('2015-05-8', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(42, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Valentin', 'Elena',
3779, TO_DATE('2021-04-1', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(43, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Aurelian', 'Elena',
3454, TO_DATE('2018-07-27', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(44, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Ghitescu',
'Alexandra', 3975, TO_DATE('2006-07-1', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(45, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Ghitescu', 'Radu',
4478, TO_DATE('2014-03-4', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(46, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Burlacu', 'Marian',
4078, TO_DATE('2005-05-22', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(47, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Victor', 'Tudor',
2806, TO_DATE('2018-02-17', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(48, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Dima', 'Monica',
4888, TO_DATE('2011-02-19', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(49, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Cretu', 'Alex',
4331, TO_DATE('2021-01-27', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(50, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Popescu', 'Daniela',
3376, TO_DATE('2016-03-23', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(51, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Stanciu', 'Paul',
3838, TO_DATE('2023-06-22', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(52, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Ionita', 'Claudiu',
3207, TO_DATE('2010-07-19', 'YYYY-MM-DD'));
```

```
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(53, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Sorin', 'Daniela',
3603, TO_DATE('2023-07-26', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(54, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Tanase', 'Bianca',
2830, TO_DATE('2013-02-20', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(55, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Florin', 'Lavinia',
3106, TO_DATE('2023-02-3', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(56, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Gheorghita',
'Emilian', 3424, TO_DATE('2017-03-5', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(57, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Oprea', 'Elena',
3302, TO_DATE('2021-02-14', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(58, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Negreanu', 'Elena',
4028, TO_DATE('2003-09-10', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(59, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Bogdan', 'Bogdan',
4342, TO_DATE('2015-02-17', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(60, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Ilieșcu', 'Nicolae',
3868, TO_DATE('2006-06-19', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(61, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Florea', 'Daniela',
4229, TO_DATE('2012-01-19', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(62, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Tomescu', 'Ema',
4473, TO_DATE('2014-01-23', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(63, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Eugen', 'Luminița',
3647, TO_DATE('2019-09-20', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(64, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Grosu', 'Alina',
4422, TO_DATE('2009-04-28', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(65, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Balanescu', 'Monica',
4190, TO_DATE('2023-02-16', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(66, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Octavian', 'Florin',
3999, TO_DATE('2021-03-9', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
```

```
(67, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Ghitescu', 'Răzvan',  
3359, TO_DATE('2005-01-9', 'YYYY-MM-DD'));  
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,  
DATA_ANGAJARI) VALUES  
(68, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Nistor', 'Simona',  
3864, TO_DATE('2021-05-14', 'YYYY-MM-DD'));  
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,  
DATA_ANGAJARI) VALUES  
(69, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Serban', 'Elena',  
3378, TO_DATE('2010-03-20', 'YYYY-MM-DD'));  
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,  
DATA_ANGAJARI) VALUES  
(70, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Dima', 'Ionel',  
2840, TO_DATE('2015-06-14', 'YYYY-MM-DD'));  
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,  
DATA_ANGAJARI) VALUES  
(71, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Ionita', 'Loredana',  
4796, TO_DATE('2011-04-25', 'YYYY-MM-DD'));  
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,  
DATA_ANGAJARI) VALUES  
(72, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Dumitache', 'Diana',  
3401, TO_DATE('2023-08-2', 'YYYY-MM-DD'));  
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,  
DATA_ANGAJARI) VALUES  
(73, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Chiriac', 'Bogdan',  
2937, TO_DATE('2000-09-28', 'YYYY-MM-DD'));  
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,  
DATA_ANGAJARI) VALUES  
(74, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Florin', 'Andrei',  
3731, TO_DATE('2019-09-11', 'YYYY-MM-DD'));  
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,  
DATA_ANGAJARI) VALUES  
(75, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Dobre', 'Sorin',  
4749, TO_DATE('2006-01-12', 'YYYY-MM-DD'));  
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,  
DATA_ANGAJARI) VALUES  
(76, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Oprea', 'Teodora',  
4581, TO_DATE('2010-07-28', 'YYYY-MM-DD'));  
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,  
DATA_ANGAJARI) VALUES  
(77, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Radu', 'Elena',  
3035, TO_DATE('2011-05-25', 'YYYY-MM-DD'));  
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,  
DATA_ANGAJARI) VALUES  
(78, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Horia', 'Dumitru',  
3801, TO_DATE('2022-06-18', 'YYYY-MM-DD'));  
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,  
DATA_ANGAJARI) VALUES  
(79, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Rosu', 'Victor',  
4837, TO_DATE('2009-02-25', 'YYYY-MM-DD'));  
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,  
DATA_ANGAJARI) VALUES  
(80, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Diaconu', 'Stefan',  
2991, TO_DATE('2001-01-6', 'YYYY-MM-DD'));  
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,  
DATA_ANGAJARI) VALUES  
(81, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Hagi', 'Elena',  
4061, TO_DATE('2001-08-4', 'YYYY-MM-DD'));
```

```
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(82, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Vasiliu', 'Luminița',
3784, TO_DATE('2014-05-16', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(83, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Lucian', 'Silvian',
3120, TO_DATE('2014-08-8', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(84, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Frumosu', 'Octavia',
4328, TO_DATE('2003-09-26', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(85, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Lupu', 'Monica',
3541, TO_DATE('2009-08-20', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(86, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Marian', 'Georgiana',
3089, TO_DATE('2016-01-1', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(87, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Zoltan', 'Dorina',
4002, TO_DATE('2020-08-26', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(88, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Vasiliu', 'Anca',
4889, TO_DATE('2001-07-16', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(89, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Ciprian', 'Bogdan',
4682, TO_DATE('2007-02-24', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(90, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Vasilescu',
'Alexandru', 4942, TO_DATE('2007-01-12', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(91, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Ionita', 'Alexandru',
3630, TO_DATE('2003-04-27', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(92, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Dumitru', 'Ionel',
2865, TO_DATE('2011-04-26', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(93, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Badea', 'Florin',
4998, TO_DATE('2007-09-24', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(94, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Dragos', 'Şerban',
4476, TO_DATE('2004-02-3', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(95, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Aurelian', 'Irina',
4128, TO_DATE('2020-03-17', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
```

```
(96, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Popescu', 'Octavian',
4590, TO_DATE('2006-05-9', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(97, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Dragomir',
'Sebastian', 3776, TO_DATE('2008-02-20', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(98, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Radu', 'Larisa',
3162, TO_DATE('2020-06-10', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(99, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Badea', 'Anca',
3381, TO_DATE('2004-07-15', 'YYYY-MM-DD'));
INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU,
DATA_ANGAJARI) VALUES
(100, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Voicu', 'Andrei',
4252, TO_DATE('2015-09-20', 'YYYY-MM-DD'));

781 ◻ INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU, DATA_ANGAJARI) VALUES
782 ◻ (94, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Drago', 'Sebastian', 4476, TO_DATE('2004-02-3', 'YYYY-MM-DD'));
783 ◻ INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU, DATA_ANGAJARI) VALUES
784 ◻ (95, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Aurelian', 'Irina', 4128, TO_DATE('2020-03-17', 'YYYY-MM-DD'));
785 ◻ INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU, DATA_ANGAJARI) VALUES
786 ◻ (96, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Popescu', 'Octavian', 4590, TO_DATE('2006-05-9', 'YYYY-MM-DD'));
787 ◻ INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU, DATA_ANGAJARI) VALUES
788 ◻ (97, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Dragomir', 'Sebastian', 3776, TO_DATE('2008-02-20', 'YYYY-MM-DD'));
789 ◻ INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU, DATA_ANGAJARI) VALUES
790 ◻ (98, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Radu', 'Larisa', 3162, TO_DATE('2020-06-10', 'YYYY-MM-DD'));
791 ◻ INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU, DATA_ANGAJARI) VALUES
792 ◻ (99, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Badea', 'Anca', 3381, TO_DATE('2004-07-15', 'YYYY-MM-DD'));
793 ◻ INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU, DATA_ANGAJARI) VALUES
794 ◻ (100, ROUND(DBMS_RANDOM.VALUE(1, 30)), ROUND(DBMS_RANDOM.VALUE(1, 11)), 'Voicu', 'Andrei', 4252, TO_DATE('2015-09-20', 'YYYY-MM-DD'));
795 COMMIT;
796
797 ✓ select * from MEDICI;
798
```

| ID_MEDIC | ID_SPECIALIZARE | ID_SPITAL | NUME  | PRENUME   | SALARIU | DATA_ANGAJARI |
|----------|-----------------|-----------|-------|-----------|---------|---------------|
| 97       | 97              | 8         | Drago | Sebastian | 3776    | 2008-02-20    |
| 98       | 98              | 1         | Radu  | Larisa    | 3162    | 2020-06-10    |
| 99       | 99              | 3         | Badea | Anca      | 3381    | 2004-07-15    |
| 100      | 100             | 4         | Voicu | Andrei    | 4252    | 2015-09-20    |

### Inserturi pentru tabela Job:

```
INSERT INTO Joburi (ID_JOB, NUME_DOMENIU) VALUES (1, 'Programare');
INSERT INTO Joburi (ID_JOB, NUME_DOMENIU) VALUES (2, 'Design Grafic');
INSERT INTO Joburi (ID_JOB, NUME_DOMENIU) VALUES (3, 'Marketing Digital');
INSERT INTO Joburi (ID_JOB, NUME_DOMENIU) VALUES (4, 'Management');
INSERT INTO Joburi (ID_JOB, NUME_DOMENIU) VALUES (5, 'Finante');
INSERT INTO Joburi (ID_JOB, NUME_DOMENIU) VALUES (6, 'Inginerie Electrica');
INSERT INTO Joburi (ID_JOB, NUME_DOMENIU) VALUES (7, 'Resurse Umane');
INSERT INTO Joburi (ID_JOB, NUME_DOMENIU) VALUES (8, 'Vanzari');
INSERT INTO Joburi (ID_JOB, NUME_DOMENIU) VALUES (9, 'Logistica');
INSERT INTO Joburi (ID_JOB, NUME_DOMENIU) VALUES (10, 'Asistenta Medicala');
INSERT INTO Joburi (ID_JOB, NUME_DOMENIU) VALUES (11, 'Educatie');
INSERT INTO Joburi (ID_JOB, NUME_DOMENIU) VALUES (12, 'Jurnalism');
INSERT INTO Joburi (ID_JOB, NUME_DOMENIU) VALUES (13, 'Consultanta IT');
INSERT INTO Joburi (ID_JOB, NUME_DOMENIU) VALUES (14, 'Arhitectura');
INSERT INTO Joburi (ID_JOB, NUME_DOMENIU) VALUES (15, 'Servicii Clienti');
INSERT INTO Joburi (ID_JOB, NUME_DOMENIU) VALUES (16, 'Turism');
INSERT INTO Joburi (ID_JOB, NUME_DOMENIU) VALUES (17, 'Proiectare Mecanica');
INSERT INTO Joburi (ID_JOB, NUME_DOMENIU) VALUES (18, 'Arte Plastice');
INSERT INTO Joburi (ID_JOB, NUME_DOMENIU) VALUES (19, 'Biotehnologie');
INSERT INTO Joburi (ID_JOB, NUME_DOMENIU) VALUES (20, 'Telecomunicatii');
```

```

INSERT INTO Joburi (ID_JOB, NUME_DOMENIU) VALUES (16, 'Turism');
INSERT INTO Joburi (ID_JOB, NUME_DOMENIU) VALUES (17, 'Proiectare Mecanica');
INSERT INTO Joburi (ID_JOB, NUME_DOMENIU) VALUES (18, 'Arte Plastice');
INSERT INTO Joburi (ID_JOB, NUME_DOMENIU) VALUES (19, 'Biotehnologie');
INSERT INTO Joburi (ID_JOB, NUME_DOMENIU) VALUES (20, 'Telecomunicatii');
Commit;
SELECT * FROM Joburi;

-- Inserarea datelor pentru tabela boli_specifice cu date fictive
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (1, 'Gripa', 'Simptome: febra, tuse, oboseala');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (2, 'Raceala', 'Simptome: nas infundat, stranut, dureri de gat');



```

### Inserturi pentru tabela Boli Specifice:

```

INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (1, 'Gripa', 'Simptome: febra, tuse, oboseala');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (2, 'Raceala', 'Simptome: nas infundat, stranut, dureri de gat');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (3, 'Diabet', 'Necesita gestionarea nivelului de zahar din sange');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (4, 'Hipertensiune', 'Cresterea presiunii arteriale');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (5, 'Osteoporoza', 'Scaderea densitatii osoase');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (6, 'Alergii', 'Reactii exagerate la alergeni specifici');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (7, 'Asthm', 'Dificultati in respiratie');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (8, 'Depresie', 'Stare persistenta de tristete');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (9, 'Anxietate', 'Sentimente exagerate de nelinieste');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (10, 'Artrita', 'Inflamarea articulatiilor');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (11, 'Parkinson', 'Tulburari motorii si tremuraturi');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (12, 'Cataracta', 'Opacifierea cristalinului ochiului');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (13, 'Diaree', 'Tranzit intestinal rapid');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (14, 'Anemia', 'Scaderea numarului de globule rosii');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (15, 'Epilepsie', 'Tulburari ale activitatii electrice cerebrale');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (16, 'Obstructie Pulmonara', 'Blocarea fluxului de aer in plamani');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (17, 'Leucemie', 'Proliferare anormala a celulelor albe din sange');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (18, 'Insuficienta Renala', 'Scaderea functiei renale');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (19, 'Migrena', 'Dureri de cap severe');

```

```

INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (20, 'Gastrita',
'Inflamarea mucoasei stomacului');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (21, 'Fibromialgie',
'Dureri musculare si oboseala cronica');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (22, 'Hepatita',
'Inflamarea ficatului');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (23, 'Psoriazis',
'Afectiune a pielii caracterizata prin descuamare');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (24, 'Tiroidita',
'Inflamarea glandei tiroide');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (25, 'Ulcer', 'Leseuni
ale mucoasei digestive');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (26, 'Scleroza Multipla',
'Leziuni ale sistemului nervos central');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (27, 'Pancreatita',
'Inflamarea pancreasului');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (28, 'Sindromul Ovarului Policistic',
'Dezechilibru hormonal la femei');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (29, 'Tromboza Venoasa',
'Formarea cheagurilor de sange in vene');
INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (30, 'Tuberculoza',
'Bacterie care afecteaza plamanii');

```

```

840    INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (17, 'Leucemie', 'Proliferare abnormala a celulelor albe din sange');
841    INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (18, 'Insuficienta Renala', 'Scadere functiei renale');
842    INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (19, 'Anemie', 'Dureri de cap si obosinte');
843    INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (20, 'Gastrita', 'Inflamarea mucoasei stomacului');
844    INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (21, 'Fibromialgia', 'Dureri musculare si oboseala cronica');
845    INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (22, 'Hepatita', 'Afectiune a pielii caracterizata prin descuamare');
846    INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (23, 'Psoriazis', 'Inflamarea glandei tiroide');
847    INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (24, 'Tiroidita', 'Inflamarea pancreasului');
848    INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (25, 'Ulcer', 'Leseuni ale mucositei digestive');
849    INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (26, 'Scleroza Multipla', 'Leziuni ale sistemului nervos central');
850    INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (27, 'Pancreatita', 'Inflamarea pancreasului');
851    INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (28, 'Sindromul Ovarului Policistic', 'Dezechilibru hormonal la femei');
852    INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (29, 'Tromboza Venoasa', 'Formarea cheagurilor de sange in vene');
853    INSERT INTO boli_specifice (ID_BOALA, NUME_BOALA, OBSERVATII) VALUES (30, 'Tuberculoza', 'Bacterie care afecteaza plamanii');

854    Commit;
855    select * from BOLI_SPECIFICE;
856    -- inserarea datelor pentru tabela job_predispusi_boli
857
858

```

|    | ID_BOALA | NUME_BOALA                    | OBSERVATII                            |
|----|----------|-------------------------------|---------------------------------------|
| 27 | 27       | Pancreatita                   | Inflamarea pancreasului               |
| 28 | 28       | Sindromul Ovarului Policistic | Dezechilibru hormonal la femei        |
| 29 | 29       | Tromboza Venoasa              | Formarea cheagurilor de sange in vene |
| 30 | 30       | Tuberculoza                   | Bacterie care afecteaza plamanii      |

856 C Copilot is available. It's recommended to install it. // Install update Hide forever (29 minutes ago)

856

C

R

U

F

T

S

P

O

N

D

A

M

L

V

W

X

Y

Z

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

```

INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (11, 2);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (11, 5);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (12, 12);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (13, 3);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (13, 9);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (14, 8);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (14, 15);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (15, 13);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (15, 16);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (16, 4);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (16, 7);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (17, 10);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (17, 13);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (18, 14);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (18, 19);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (19, 15);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (19, 16);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (20, 11);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (20, 14);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (21, 19);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (21, 20);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (22, 18);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (22, 20);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (23, 14);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (23, 19);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (24, 12);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (24, 17);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (25, 5);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (25, 9);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (26, 1);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (26, 13);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (27, 17);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (27, 18);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (28, 9);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (28, 15);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (29, 14);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (29, 20);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (30, 3);
INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (30, 16);

```

```

932 INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (28, 13);
933 INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (28, 13);
934
935 INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (27, 17);
936 INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (27, 18);
937
938 INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (28, 9);
939 INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (28, 15);
940
941 INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (29, 14);
942 INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (29, 20);
943
944
945 INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (30, 3);
946 INSERT INTO job_predispsi_boli (ID_BOALA, ID_JOB) VALUES (30, 16);
947
948
949 ✓ select * from JOB_PREDISPSI_BOLI;
950 -- Insertarea datelor pentru tabela Tip_Retete cu date fictive
951 INSERT INTO TIP_RETETE (ID_TIP, NUMELE_TIPULUI) VALUES (1, 'Reteta simpla');

```

Output JOB\_PREDISPSI\_BOLI

| ID_BOALA | ID_JOB |
|----------|--------|
| 52       | 18     |
| 53       | 21     |
| 54       | 23     |
| 55       | 21     |
| 56       | 22     |

ms

Problems Services

Copilot is available. It's recommended to install it. // Install update Hide forever (31 minutes ago)

949:30 CRLF UTF-8 4 spaces

### Inserturi pentru tabela Tip Reteta:

```
INSERT INTO TIP_RETETE (ID_TIP, NUMELE_TIPULUI) VALUES (1, 'Reteta simpla');
```

```

INSERT INTO TIP_RETETE (ID_TIP, NUMELE_TIPULUI) VALUES (2, 'Reteta compusa');
INSERT INTO TIP_RETETE (ID_TIP, NUMELE_TIPULUI) VALUES (3, 'Reteta electronica');
INSERT INTO TIP_RETETE (ID_TIP, NUMELE_TIPULUI) VALUES (4, 'Reteta urgență');
INSERT INTO TIP_RETETE (ID_TIP, NUMELE_TIPULUI) VALUES (5, 'Reteta copii');
INSERT INTO TIP_RETETE (ID_TIP, NUMELE_TIPULUI) VALUES (6, 'Reteta cronică');
INSERT INTO TIP_RETETE (ID_TIP, NUMELE_TIPULUI) VALUES (7, 'Reteta pentru afecțiuni dermatologice');
INSERT INTO TIP_RETETE (ID_TIP, NUMELE_TIPULUI) VALUES (8, 'Reteta pentru afecțiuni respiratorii');
INSERT INTO TIP_RETETE (ID_TIP, NUMELE_TIPULUI) VALUES (9, 'Reteta pentru afecțiuni cardiace');
INSERT INTO TIP_RETETE (ID_TIP, NUMELE_TIPULUI) VALUES (10, 'Reteta pentru afecțiuni digestive');

150 INSERT INTO TIP_RETETE (ID_TIP, NUMELE_TIPULUI) VALUES (4, 'Reteta urgență');
151 INSERT INTO TIP_RETETE (ID_TIP, NUMELE_TIPULUI) VALUES (5, 'Reteta copii');
152 INSERT INTO TIP_RETETE (ID_TIP, NUMELE_TIPULUI) VALUES (6, 'Reteta cronică');
153 INSERT INTO TIP_RETETE (ID_TIP, NUMELE_TIPULUI) VALUES (7, 'Reteta pentru afecțiuni dermatologice');
154 INSERT INTO TIP_RETETE (ID_TIP, NUMELE_TIPULUI) VALUES (8, 'Reteta pentru afecțiuni respiratorii');
155 INSERT INTO TIP_RETETE (ID_TIP, NUMELE_TIPULUI) VALUES (9, 'Reteta pentru afecțiuni cardiace');
156 INSERT INTO TIP_RETETE (ID_TIP, NUMELE_TIPULUI) VALUES (10, 'Reteta pentru afecțiuni digestive');
157 Commit;
158 ✓ select * from TIP_RETETE;
159 select * from MEDICAMENTE;

```

| ID_TIP | NUMELE_TIPULUI       |
|--------|----------------------|
| 1      | 1 Reteta simpla      |
| 2      | 2 Reteta compusa     |
| 3      | 3 Reteta electronica |
| 4      | 4 Reteta urgență     |
| 5      | 5 Reteta copii       |

### Inserturi pentru tabela Medicamente:

```

INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (1, 3, 'Paracetamol');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (2, 5, 'Ibuprofen');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (3, 2, 'Amoxicilin');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (4, 8, 'Omeprazol');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (5, 1, 'Aspirină');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (6, 4, 'Loratadină');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (7, 9, 'Captopril');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (8, 6, 'Metformin');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (9, 7, 'Atorvastatină');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (10, 10, 'Escitalopram');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (11, 3, 'Diazepam');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (12, 5, 'Ranitidină');

```

```
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (13, 2, 'Ceftriaxonă');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (14, 8, 'Lansoprazol');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (15, 1, 'Naproxen');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (16, 4, 'Desloratadină');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (17, 9, 'Enalapril');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (18, 6, 'Gliclazidă');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (19, 7, 'Simvastatină');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (20, 10, 'Fluoxetină');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (21, 3, 'Clonazepam');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (22, 5, 'Dexametazonă');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (23, 2, 'Azitromicină');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (24, 8, 'Pantoprazol');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (25, 1, 'Ibuprofen');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (26, 4, 'Fexofenadină');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (27, 9, 'Ramipril');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (28, 6, 'Glibenclamidă');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (29, 7, 'Rosuvastatină');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (30, 10, 'Sertralin');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (31, 3, 'Alprazolam');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (32, 5, 'Diclofenac');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (33, 2, 'Ciprofloxacină');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (34, 8, 'Esomeprazol');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (35, 1, 'Acetaminofen');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (36, 4, 'Levocetirizină');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (37, 9, 'Losartan');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (38, 6, 'Pioglitazonă');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (39, 7, 'Pravastatină');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (40, 10, 'Venlafaxină');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (41, 3, 'Zolpidem');
```

```

INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (42, 5, 'Meloxicam');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (43, 2, 'Penicilină');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (44, 8, 'Rabeprazol');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (45, 1, 'Celecoxib');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (46, 4, 'Bilastină');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (47, 9, 'Perindopril');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (48, 6, 'Repaglinidă');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (49, 7, 'Ezetimib');
INSERT INTO Medicamente (id_medicament, tip_reteta, nume_medicament)
VALUES (50, 10, 'Mirtazapin');

```

| ID_MEDICAMENT | TIP_RETETA | NUME_MEDICAMENT |
|---------------|------------|-----------------|
| 45            | 65         | 1 Celecoxib     |
| 46            | 46         | 4 Bilastină     |
| 47            | 47         | 9 Perindopril   |
| 48            | 48         | 6 Repaglinidă   |
| 49            | 49         | 7 Ezetimib      |
| 50            | 50         | 10 Mirtazapin   |

### Inserturi pentru tabela Retete:

```

INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
VALUES
(1, 1, (TO_DATE('2021-01-01', 'YYYY-MM-DD') + DBMS_RANDOM.VALUE * (TO_DATE('2023-12-02', 'YYYY-MM-DD') - TO_DATE('2021-01-01', 'YYYY-MM-DD'))), Lista(5, 15, 25, 35, 45));
INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
VALUES
(2, 1, (TO_DATE('2021-01-01', 'YYYY-MM-DD') + DBMS_RANDOM.VALUE * (TO_DATE('2023-12-02', 'YYYY-MM-DD') - TO_DATE('2021-01-01', 'YYYY-MM-DD'))), Lista(5, 15));
INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
VALUES
(3, 1, (TO_DATE('2021-01-01', 'YYYY-MM-DD') + DBMS_RANDOM.VALUE * (TO_DATE('2023-12-02', 'YYYY-MM-DD') - TO_DATE('2021-01-01', 'YYYY-MM-DD'))), Lista(5, 15, 25));
INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
VALUES
(4, 1, (TO_DATE('2021-01-01', 'YYYY-MM-DD') + DBMS_RANDOM.VALUE * (TO_DATE('2023-12-02', 'YYYY-MM-DD') - TO_DATE('2021-01-01', 'YYYY-MM-DD'))), Lista(5, 15, 25, 35));

```

```
(5, 1, (TO_DATE('2021-01-01', 'YYYY-MM-DD') + DBMS_RANDOM.VALUE * (TO_DATE('2023-12-02', 'YYYY-MM-DD') - TO_DATE('2021-01-01', 'YYYY-MM-DD'))), Lista(5));
INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
VALUES
(6, 1, (TO_DATE('2021-01-01', 'YYYY-MM-DD') + DBMS_RANDOM.VALUE * (TO_DATE('2023-12-02', 'YYYY-MM-DD') - TO_DATE('2021-01-01', 'YYYY-MM-DD'))), Lista(5, 15, 25, 35, 45));
INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
VALUES
(7, 1, TO_DATE('2022-02-18', 'YYYY-MM-DD'), Lista(5, 15, 25));
INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
VALUES
(8, 1, (TO_DATE('2021-01-01', 'YYYY-MM-DD') + DBMS_RANDOM.VALUE * (TO_DATE('2023-12-02', 'YYYY-MM-DD') - TO_DATE('2021-01-01', 'YYYY-MM-DD'))), Lista(5, 15, 25, 35));
INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
VALUES
(9, 1, (TO_DATE('2021-01-01', 'YYYY-MM-DD') + DBMS_RANDOM.VALUE * (TO_DATE('2023-12-02', 'YYYY-MM-DD') - TO_DATE('2021-01-01', 'YYYY-MM-DD'))), Lista(5, 15));
INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
VALUES
(10, 1, (TO_DATE('2021-01-01', 'YYYY-MM-DD') + DBMS_RANDOM.VALUE * (TO_DATE('2023-12-02', 'YYYY-MM-DD') - TO_DATE('2021-01-01', 'YYYY-MM-DD'))), Lista(5, 15, 25, 35));

INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
VALUES(21, 3, TO_DATE('2021-06-15', 'YYYY-MM-DD'), Lista(1, 11, 21, 31, 41));
INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
VALUES(22, 3, TO_DATE('2022-07-22', 'YYYY-MM-DD'), Lista(1, 11));
INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
VALUES(23, 3, TO_DATE('2022-01-10', 'YYYY-MM-DD'), Lista(1, 11, 21));
INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
VALUES(24, 3, (TO_DATE('2021-01-01', 'YYYY-MM-DD') + DBMS_RANDOM.VALUE *
(TO_DATE('2023-12-02', 'YYYY-MM-DD') - TO_DATE('2021-01-01', 'YYYY-MM-DD'))), Lista(1, 11, 21,
31));
INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
VALUES(25, 3, (TO_DATE('2021-01-01', 'YYYY-MM-DD') + DBMS_RANDOM.VALUE *
(TO_DATE('2023-12-02', 'YYYY-MM-DD') - TO_DATE('2021-01-01', 'YYYY-MM-DD'))), Lista(1));
INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
VALUES(26, 3, (TO_DATE('2021-01-01', 'YYYY-MM-DD') + DBMS_RANDOM.VALUE *
(TO_DATE('2023-12-02', 'YYYY-MM-DD') - TO_DATE('2021-01-01', 'YYYY-MM-DD'))), Lista(1, 11, 21,
31, 41));
INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
VALUES(27, 3, TO_DATE('2022-02-18', 'YYYY-MM-DD'), Lista(1, 11, 21));
INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
VALUES(28, 3, (TO_DATE('2021-01-01', 'YYYY-MM-DD') + DBMS_RANDOM.VALUE *
(TO_DATE('2023-12-02', 'YYYY-MM-DD') - TO_DATE('2021-01-01', 'YYYY-MM-DD'))), Lista(1, 11, 21,
31));
INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
VALUES(29, 3, (TO_DATE('2021-01-01', 'YYYY-MM-DD') + DBMS_RANDOM.VALUE *
(TO_DATE('2023-12-02', 'YYYY-MM-DD') - TO_DATE('2021-01-01', 'YYYY-MM-DD'))), Lista(1, 11));
INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
VALUES(30, 3, (TO_DATE('2021-01-01', 'YYYY-MM-DD') + DBMS_RANDOM.VALUE *
(TO_DATE('2023-12-02', 'YYYY-MM-DD') - TO_DATE('2021-01-01', 'YYYY-MM-DD'))), Lista(1, 11, 21,
31));

INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
VALUES(31, 4, (TO_DATE('2021-01-01', 'YYYY-MM-DD') + DBMS_RANDOM.VALUE *
(TO_DATE('2023-12-02', 'YYYY-MM-DD') - TO_DATE('2021-01-01', 'YYYY-MM-DD'))), Lista(6, 16, 26,
36, 46));
INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
```









```

INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
VALUES(96, 10, (TO_DATE('2021-01-01', 'YYYY-MM-DD') + DBMS_RANDOM.VALUE *
(TO_DATE('2023-12-02', 'YYYY-MM-DD') - TO_DATE('2021-01-01', 'YYYY-MM-DD'))), Lista(10, 20,
30, 40, 50));
INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
VALUES(97, 10, TO_DATE('2022-02-18', 'YYYY-MM-DD'), Lista(10, 20, 30));
INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
VALUES(98, 10, (TO_DATE('2021-01-01', 'YYYY-MM-DD') + DBMS_RANDOM.VALUE *
(TO_DATE('2023-12-02', 'YYYY-MM-DD') - TO_DATE('2021-01-01', 'YYYY-MM-DD'))), Lista(10, 20,
30, 40));
INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
VALUES(99, 10, (TO_DATE('2021-01-01', 'YYYY-MM-DD') + DBMS_RANDOM.VALUE *
(TO_DATE('2023-12-02', 'YYYY-MM-DD') - TO_DATE('2021-01-01', 'YYYY-MM-DD'))), Lista(10, 20));
INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
VALUES(100, 10, (TO_DATE('2021-01-01', 'YYYY-MM-DD') + DBMS_RANDOM.VALUE *
(TO_DATE('2023-12-02', 'YYYY-MM-DD') - TO_DATE('2021-01-01', 'YYYY-MM-DD'))), Lista(10, 20,
30, 40));
1221 :INSERI INIU RETETE (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
1222 :VALUES(97, 10, TO_DATE('2022-02-18', 'YYYY-MM-DD'), Lista(10, 20, 30));
1223 :INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
1224 :VALUES(98, 10, (TO_DATE('2021-01-01', 'YYYY-MM-DD') + DBMS_RANDOM.VALUE * (TO_DATE('2023-12-02', 'YYYY-MM-DD') - TO_DATE('2021-01-01', 'YYYY-MM-DD'))), Lista(10, 20, 30, 40));
1225 :INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
1226 :VALUES(99, 10, (TO_DATE('2021-01-01', 'YYYY-MM-DD') + DBMS_RANDOM.VALUE * (TO_DATE('2023-12-02', 'YYYY-MM-DD') - TO_DATE('2021-01-01', 'YYYY-MM-DD'))), Lista(10, 20));
1227 :INSERT INTO Retete (id_reteta, tip_reteta, data_eliberarii, lista_medicamente)
1228 :VALUES(100, 10, (TO_DATE('2021-01-01', 'YYYY-MM-DD') + DBMS_RANDOM.VALUE * (TO_DATE('2023-12-02', 'YYYY-MM-DD') - TO_DATE('2021-01-01', 'YYYY-MM-DD'))), Lista(10, 20, 30, 40));
1229 :
1230 :COMMIT;
1231 ✓ select * from RETETE;
1232 :
1233 SELECT table_name
1234 FROM user_tables
1235 WHERE num_rows = 0;
1236 :
1237 :
1238 select * from PACIENTI;
1239 alter table pacienti
1240 add Data_Nastere Date not null;

Output RETETE
ID_RETETA TIP_RETETA DATA_ELBERARII LISTA_MEDICAMENTE
97 97 10 2022-02-18 {10,20,30}
98 98 10 2022-09-26 05:02:40 {10,20,30,40}
99 99 10 2023-06-26 08:44:54 {10,20}
100 100 10 2022-02-10 01:11:23 {10,20,30,40}

5 ms
Problems Services
Copilot is available. It's recommended to install it. // Install update Hide forever (34 minutes ago)
1231:14 CRLF UTF-8 4 spaces

```

### Inserturi pentru tabela Pacienti:

```

INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(1, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Popescu', 'Ion',
TO_DATE('1990-01-15', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(2, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Ionescu', 'Ana',
TO_DATE('1985-05-20', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(3, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Vasilescu',
'Maria', TO_DATE('1992-08-10', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(4, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Dumitrescu',
'George', TO_DATE('1988-03-03', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(5, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Radulescu',
'Elena', TO_DATE('1995-11-25', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(6, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Constantinescu',
'Mihai', TO_DATE('1987-06-18', 'YYYY-MM-DD'));

```

```

INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(7, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Gheorghe', 'Ana',
TO_DATE('1993-02-12', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(8, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Stoica',
'Alexandru', TO_DATE('1984-07-30', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(9, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Munteanu',
'Andreea', TO_DATE('1991-09-08', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(10, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Cristea',
'Cristian', TO_DATE('1989-12-05', 'YYYY-MM-DD'));

INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(11, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Georgescu',
'Andrei', TO_DATE('1994-04-22', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(12, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Dinu', 'Elena',
TO_DATE('1986-10-15', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(13, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Ilie', 'Ioana',
TO_DATE('1997-07-07', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(14, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Voicu', 'Adrian',
TO_DATE('1983-12-30', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(15, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Matei',
'Gabriela', TO_DATE('1990-02-18', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(16, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Antonescu',
'Raluca', TO_DATE('1988-05-12', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(17, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Iancu', 'Vlad',
TO_DATE('1995-09-25', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(18, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Rosu', 'Laura',
TO_DATE('1987-03-07', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(19, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Florescu',
'Catalin', TO_DATE('1992-11-10', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(20, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Mazilu',
'Simona', TO_DATE('1989-06-02', 'YYYY-MM-DD'));

INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(21, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Radu',
'Ana-Maria', TO_DATE('1993-08-14', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(22, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Stefanescu',
'Razvan', TO_DATE('1988-01-27', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(23, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Pavel', 'Elena',
TO_DATE('1991-05-05', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(24, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Costin',
'Andreea', TO_DATE('1985-12-18', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)

```

```

VALUES(25, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Dragomir',
'Cristian', TO_DATE('1996-10-03', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(26, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Toma',
'Gabriela', TO_DATE('1987-03-21', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(27, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Gavrilă',
'Daniel', TO_DATE('1994-06-28', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(28, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Sava', 'Iulia',
TO_DATE('1989-09-10', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(29, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Iovan', 'Mirela',
TO_DATE('1992-02-13', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(30, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Vlad', 'Silviu',
TO_DATE('1986-11-07', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(31, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Neacsu',
'Roxana', TO_DATE('1992-04-05', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(32, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Mihai', 'Victor',
TO_DATE('1987-09-18', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(33, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Ivan', 'Diana',
TO_DATE('1995-02-10', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(34, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Florea',
'Adrian', TO_DATE('1990-07-23', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(35, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Voinea', 'Andra',
TO_DATE('1984-12-06', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(36, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Albu',
'Catalina', TO_DATE('1993-10-21', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(37, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Coman', 'Ionut',
TO_DATE('1988-05-14', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(38, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Vasile',
'Radu', TO_DATE('1991-08-27', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(39, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Dobre', 'Bogdan',
TO_DATE('1986-01-30', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(40, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Nita', 'Larisa',
TO_DATE('1994-03-03', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(41, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Popa', 'Iuliana',
TO_DATE('1989-06-12', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(42, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Marin', 'Eugen',
TO_DATE('1994-11-25', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(43, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Stanciu',
'Mihaela', TO_DATE('1992-02-18', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)

```

```

VALUES(44, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Apostol',
'Victor', TO_DATE('1987-09-30', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(45, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Dragan',
'Camelia', TO_DATE('1996-04-03', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(46, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Oprea',
'Sebastian', TO_DATE('1988-08-16', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(47, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Pascu', 'Roxana',
TO_DATE('1993-01-19', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(48, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Serban',
'George', TO_DATE('1985-12-22', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(49, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Balan',
'Daniela', TO_DATE('1990-03-07', 'YYYY-MM-DD'));
INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
VALUES(50, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Dima',
'Valentin', TO_DATE('1986-07-15', 'YYYY-MM-DD'));

1334 : VALUES(45, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Stanciu', 'Mihaila', TO_DATE('1992-02-18', 'YYYY-MM-DD'));
1335 : INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
1336 : VALUES(46, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Apostol', 'Victor', TO_DATE('1987-09-30', 'YYYY-MM-DD'));
1337 : INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
1338 : VALUES(45, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Dragan', 'Camelia', TO_DATE('1996-04-03', 'YYYY-MM-DD'));
1339 : INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
1340 : VALUES(46, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Oprea', 'Sebastian', TO_DATE('1988-08-16', 'YYYY-MM-DD'));
1341 : INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
1342 : VALUES(47, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Pascu', 'Roxana', TO_DATE('1993-01-19', 'YYYY-MM-DD'));
1343 : INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
1344 : VALUES(48, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Serban', 'George', TO_DATE('1985-12-22', 'YYYY-MM-DD'));
1345 : INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
1346 : VALUES(49, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Balan', 'Daniela', TO_DATE('1990-03-07', 'YYYY-MM-DD'));
1347 : INSERT INTO Pacienti (id_pacient, id_domiciliu, id_job, nume, prenume, data_nastere)
1348 : VALUES(50, CEIL(DBMS_RANDOM.VALUE(1, 50)), CEIL(DBMS_RANDOM.VALUE(1, 20)), 'Dima', 'Valentin', TO_DATE('1986-07-15', 'YYYY-MM-DD'));

1349
1350 ✓ select * from PACIENTI;
1351
1352 declare

```

Output PACIENTI

| ID_PACIENT | ID_DOMICILIU | ID_JOB | NUME      | PRENUME   | DATA_NASTERE |
|------------|--------------|--------|-----------|-----------|--------------|
| 48         | 48           | 44     | Io Serban | Io George | 1985-12-22   |
| 49         | 49           | 37     | 13 Balan  | Daniela   | 1990-03-07   |
| 50         | 50           | 46     | 12 Dina   | Valentin  | 1986-07-15   |

Inserturi pentru tabela istoric pacienti(bloc pl/sql):

```

declare
random number(3);
  type tabel is table of number;
  random_retete tabel;
begin
random_retete := tabel();
for i in 1..100 loop
random_retete.extend;
random_retete(i) := i;
end loop;

for i in 1..100 loop
random := round(DBMS_RANDOM.VALUE(1,100));
while random_retete.EXISTS(random) = false loop
random := round(DBMS_RANDOM.VALUE(1,100));
  end loop;
insert into ISTORIC_PACIENTI(id_istoric, id_pacient, id_medic, id_reteta)
values
(i,round(DBMS_RANDOM.VALUE(1,50)),round(DBMS_RANDOM.value(1,100)),random_retete(random));
random_retete.DELETE(random);
  end loop;
end;

```

```
1371 for i in 1..100 loop
1372   random := round(DBMS_RANDOM.VALUE(1,100));
1373   while random_reteete.EXISTS(random) = false loop
1374     random := round(DBMS_RANDOM.VALUE(1,100));
1375   end loop;
1376   insert into ISTORIC_PACIENTI(id_istoric, id_pacient, id_medic, id_reteete)
1377   values (i,round(DBMS_RANDOM.VALUE(1,50)),round(DBMS_RANDOM.VALUE(1,100)),random_reteete(random));
1378   random_reteete.DELETE(random);
1379   end loop;
1380 end;
1381
1382 commit;
1383 select * from ISTORIC_PACIENTI;
1384
1385
1386 select * from TIP_BETEATA;
1387
1388
```

## Exercitiul 6

Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze toate cele 3 tipuri de colecții studiate. Apelați subprogramul.

Directorii Spitalelor doresc o procedura prin intermediul careia sa poata vedea in functie de preferinte:

**a. Toate Rețetele:**

- O privire de ansamblu asupra tuturor rețetelor eliberate de toți medicii până în prezent, oferind o imagine completă a activității spitalului.

**b. Rețete în Intervalul de Date:**

- Posibilitatea de a selecta rețetele eliberate într-un anumit interval de timp, pentru o analiză mai detaliată a activității medicale într-un anumit interval.

**c. Rețete de la un Anumit Spital:**

- Explorarea rețetelor eliberate de la un spital municipal specific, pentru a evalua contribuția fiecărui spital la activitatea medicală generală.

**d. Rețete de la un Anumit Medic:**

- Identificarea rețetelor eliberate de un medic anume, furnizând informații detaliate despre contribuția fiecărui medic în parte.

**e. Combinări Personalizate:**

- Posibilitatea de a combina cerințe, cum ar fi rețetele eliberate de la un anumit medic într-un anumit interval de timp sau rețetele de la un anumit spital

```

create or replace procedure Raport_Medical(
    data_start in date default null,
    data_finish in date default null,
    numespital in varchar2 default null,
    numecomplet in varchar2 default null)
is
    ---exceptii
    incorrect_date exception;
    incorrect_name exception;
    incorrect_hospital exception ;
    type vector_medicamente is varray(10) of MEDICAMENTE.NUME_MEDICAMENT%type; -- am facut
vectorul care o sa contine o lista de medicamente
    type reteta_medicamente_rec is record (nume_reteta TIP_RETETE.NUMELE_TIPULUI%type,
                                              medicamente vector_medicamente
                                         ); -- recordul pentru a retine numele retetei si
medicamentele asociatie
    type tabel_imbricat_rmr is table of reteta_medicamente_rec; -- o sa retinem intr-un tabel
imbricat pentru ca un medic are mai multe rețete emise
    type tabel_indexat_mr is table of tabel_imbricat_rmr index by varchar2(50);
    -- tabelul indexat o sa retine cheia: numele complet al medicului
    -- valoare: tabela cu rețetele eliberate de el
    tabel_medic_retete tabel_indexat_mr;
    tabel_recorduri tabel_imbricat_rmr ; -- folosim constructor pentru ca e tabel imbricat
indexxx number(3);
    data_start_p date;
    data_finish_p date;

begin
    if data_finish is null then
        data_finish_p := sysdate;
    else
        data_finish_p := data_finish;
    end if;
    if data_start is null then
        data_start_p:=to_date('1999/01/01','YYYY-MM-DD');
    else

```

```

        data_start_p := data_start;
    end if;
    DBMS_OUTPUT.PUT_LINE(data_finish_p || ' start:'|| data_start_p);
    if data_finish - data_start <= 0 then
        Raise incorrect_date;

    DBMS_OUTPUT.PUT_LINE(data_start_p);
    DBMS_OUTPUT.PUT_LINE(data_finish_p);
end if;
for medic in (
    select m.ID_MEDIC,m.nume,m.prenume from MEDICI m
left join SPITALE s on m.ID_SPITAL = s.ID_SPITAL
where ID_MEDIC in (select ID_MEDIC from ISTORIC_PACIENTI) and (numespital is null or
numespital = S.NUME_SPITAL)
    ) loop
    indexx :=1;
    tabel_recorduri := tabel_imbricat_rmr();
    for retete_medicamente in (
        select M.NUME || ' ' || M.PRENUME as Complet, R2.Lista_Medicamente,
tr.NUMELE_TIPULUI,R2.DATA_ELIBERARII from ISTORIC_PACIENTI ip
            left join MEDICI M on ip.ID_MEDIC = M.ID_MEDIC
            left join RETETE R2 on ip.ID_RETETA = R2.ID_RETETA
            left join TIP_RETETA tr on r2.TIP_RETETA = tr.ID_TIP
        where (M.NUME = medic.NUME and M.PRENUME = medic.PRENUME) and (r2.DATA_ELIBERARII
between data_start_p and data_finish_p)
    ) loop
        tabel_recorduri.extend;
        tabel_recorduri(indexx).nume_reteta := retete_medicamente.NUMELE_TIPULUI;
        tabel_recorduri(indexx).medicamente := vector_medicamente();

    tabel_recorduri(indexx).medicamente.extend(retete_medicamente.Lista_Medicamente.COUNT);
        for i in retete_medicamente.Lista_Medicamente.first..
retete_medicamente.Lista_Medicamente.LAST loop
            select NUME_MEDICAMENT into tabel_recorduri(indexx).medicamente(i) from MEDICAMENTE
where ID_MEDICAMENT = retete_medicamente.Lista_Medicamente(i);

        end loop;
        indexx := indexx+1;
    end loop;
    if tabel_recorduri.COUNT > 0 then
        tabel_medic_retete(medic.NUME || ' ' || medic.PRENUME) := tabel_recorduri;
    end if;
    end loop;

if numecomplet is null then

    if tabel_medic_retete.count = 0 then
        raise no_data_found;
    end if;

    for med in ( select ID_MEDIC,nume || ' ' || prenume as NumeCompletn from MEDICI
where ID_MEDIC in (select ID_MEDIC from ISTORIC_PACIENTI)
    ) loop
        if tabel_medic_retete.EXISTS(med.NumeCompletn) then
            DBMS_OUTPUT.PUT_LINE('| Retetele eliberate de Dr.'||med.NumeCompletn||' |');
        for i in tabel_medic_retete(med.NumeCompletn).first ..
tabel_medic_retete(med.NumeCompletn).last loop

```

```

        DBMS_OUTPUT.PUT_LINE('=====');
        DBMS_OUTPUT.PUT_LINE('Tipul Retetei:');
' || tabel_medic_retete(med.NumeComplet)(i).nume_reteta);
        DBMS_OUTPUT.PUT_LINE('Medicamente elibarate: ');
        for j in tabel_medic_retete(med.NumeComplet)(i).medicamente.first
..tabel_medic_retete(med.NumeComplet)(i).medicamente.last loop
        DBMS_OUTPUT.PUT_LINE('- ' || tabel_medic_retete(med.NumeComplet)(i).medicamente(j));
        end loop;
        DBMS_OUTPUT.PUT_LINE('=====');

        end loop;

    end if;

    end loop;
else
    if tabel_medic_retete.count = 0 then
        raise no_data_found;
    end if;
    if tabel_medic_retete.EXISTS(numecomplet) = false then
        raise incorrect_name;
    end if;
    DBMS_OUTPUT.PUT_LINE('| Retetele eliberate de Dr.' || numecomplet || ' |');
        for i in tabel_medic_retete(numecomplet).first ..
tabel_medic_retete(numecomplet).last loop
            DBMS_OUTPUT.PUT_LINE('=====');
            DBMS_OUTPUT.PUT_LINE('Tipul Retetei:');
' || tabel_medic_retete(numecomplet)(i).nume_reteta);
            DBMS_OUTPUT.PUT_LINE('Medicamente elibarate: ');
            for j in tabel_medic_retete(numecomplet)(i).medicamente.first
..tabel_medic_retete(numecomplet)(i).medicamente.last loop
            DBMS_OUTPUT.PUT_LINE('- ' || tabel_medic_retete(numecomplet)(i).medicamente(j));
            end loop;
            DBMS_OUTPUT.PUT_LINE('=====');

            end loop;

    end if;
exception
    when incorrect_date then
        DBMS_OUTPUT.PUT_LINE('Intervalul de date in care cautati rapoarte este incorect!');
    when incorrect_name then
        DBMS_OUTPUT.PUT_LINE('Numele medicului pentru care doriti sa socateti raportul este
gresit! ');
    when NO_DATA_FOUND then
        dbms_output.PUT_LINE('Nu exista o reteta eliberata intre datele respective sau numele
spitalului este gresit! ');
    when others then
        DBMS_OUTPUT.PUT_LINE('A aparut o alta eroare, va rugam sa contactati staff-ul:
Eroare[' || SQLERRM || ']');
    end Raport_Medical;

declare
data_startstring varchar2(15) := &data_st;

```

```

data_finishstring varchar2(10) :=&data_fi;
data_finish date:=null;
data_start date:=null;
nume_spital varchar(60) := &nume_spit;
nume_persoana varchar(50):= &numecautat;
begin
  data_finish := to_date(data_finishstring,'YYYY-MM-DD');
  data_start:=to_date(data_startstring,'YYYY-MM-DD');

  Raport_Medical(data_start => data_start, data_finish => data_finish, numespital =>
  nume_spital, numecomplet => nume_persoana);
end;

```

```

1400
1401 create or replace procedure Raport_Medical(
1402   data_start in date default null,
1403   data_finish in date default null,
1404   numespital in varchar2 default null,
1405   numecomplet in varchar2 default null)
1406
1407   R Raport_Medical()
1408   ole [@localhost] ×
1409
1410   When others then
1411     DBMS_OUTPUT.PUT_LINE('A aparut o alta eroare, va rugam sa contactati staff-ul: Eroare['||SQLERRM||]');
1412
1413 end Report_Medical;
[2024-01-03 17:47:52] completed in 18 ms

```

Rezultatele apelarii procedurii cu diferite variabile de intrare:

| Date de intrare | Output |
|-----------------|--------|
|                 |        |

| <p><b>Parameters</b></p> <pre>#2 ...archar2(10) null #3 ...varchar2(10) null #4 nume_spital va null #5 nume_persoana : 'Radu Elena'</pre> <p><b>Execute</b>    <b>Close</b></p>   | <pre>1525 1526 1527 1528 ✓ declare 1529   data_startstring varchar2(10) := &amp;data_st; 1530   data_finishingstring varchar2(10) :=&amp;data_fi; 1531   data_finish date:=null; 1532   data_start date:=null;</pre> <p>end;</p> <p>[2024-01-04 14:32:04] completed in 34 ms<br/>  Retetele eliberate de Dr.Radu Elena  <br/>*****<br/>Tipul Retetei: Reteta compusa<br/>Medicamente elibarate:<br/>- Amoxicilin<br/>- Cetirizina<br/>- Acitromoxina<br/>- Ciprofloxacină<br/>*****<br/>Tipul Retetei: Reteta simplă<br/>Medicamente elibarate:<br/>- Aspirină<br/>- Naproxen<br/>*****</p>   |                                   |            |                |   |   |                                   |   |    |                                   |   |    |                               |   |    |                                  |   |   |                                |   |   |                             |
|---|---|-----------------------------------|------------|----------------|---|---|-----------------------------------|---|----|-----------------------------------|---|----|-------------------------------|---|----|----------------------------------|---|---|--------------------------------|---|---|-----------------------------|
| <p><b>Parameters</b></p> <pre>#1 ...archar2(10) '2023-09-01' #2 ...varchar2(10) '2023-10-01' #3 nume_sp: null #4 nume_per null</pre> <p><b>Execute</b>    <b>Close</b></p>  | <pre>1555 ✓ declare 1556   data_startstring varchar2(15) := &amp;data_st; 1557   data_finishingstring varchar2(10) :=&amp;data_fi; 1558   data_finish date:=null; 1559   data_start date:=null; 1560   nume_spital varchar(60) := &amp;nume_spital; 1561   nume_persoana varchar(50):= &amp;nume_persoana; 1562   begin 1563     data_finish := to_date(data_finishingstring,'YYYY-MM-DD'); 1564     data_start:=to_date(data_startstring,'YYYY-MM-DD'); 1565 1566     Report_Medical(data_start =&gt; data_start, data_finish =&gt; data_finish, numespital =&gt; nume_spital) 1567   end;</pre> <p>select * from RETETE where DATA_ELIBERARI BETWEEN to_date('2023-09-01','YYYY-MM-DD') AND to_date('2023-10-01','YYYY-MM-DD')</p> <p>select * from TSTORIC_PACIENTII where TO_RETETA = RR</p> <p>ms</p> <table border="1"> <thead> <tr> <th>ID_RETETA</th> <th>TIP_RETETA</th> <th>DATA_ELIBERARI</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>8</td> <td>1 2023-09-25</td> </tr> <tr> <td>2</td> <td>18</td> <td>2 2023-09-25</td> </tr> <tr> <td>3</td> <td>46</td> <td>5 2023-09-19 05:56:35</td> </tr> <tr> <td>4</td> <td>88</td> <td>9 2023-09-14 20:29:23</td> </tr> </tbody> </table> <p>Problems    Services</p>   | ID_RETETA                         | TIP_RETETA | DATA_ELIBERARI | 1 | 8 | 1 2023-09-25                      | 2 | 18 | 2 2023-09-25                      | 3 | 46 | 5 2023-09-19 05:56:35         | 4 | 88 | 9 2023-09-14 20:29:23            |   |   |                                |   |   |                             |
| ID_RETETA   | TIP_RETETA  | DATA_ELIBERARI                    |            |                |   |   |                                   |   |    |                                   |   |    |                               |   |    |                                  |   |   |                                |   |   |                             |
| 1   | 8   | 1 2023-09-25                      |            |                |   |   |                                   |   |    |                                   |   |    |                               |   |    |                                  |   |   |                                |   |   |                             |
| 2   | 18  | 2 2023-09-25                      |            |                |   |   |                                   |   |    |                                   |   |    |                               |   |    |                                  |   |   |                                |   |   |                             |
| 3   | 46  | 5 2023-09-19 05:56:35             |            |                |   |   |                                   |   |    |                                   |   |    |                               |   |    |                                  |   |   |                                |   |   |                             |
| 4   | 88  | 9 2023-09-14 20:29:23             |            |                |   |   |                                   |   |    |                                   |   |    |                               |   |    |                                  |   |   |                                |   |   |                             |
| <p><b>Parameters</b></p> <pre>#1 ...archar2(10) := '2023-09-01' #2 ...varchar2(10) := null #3 nume_spital varchar(60) := 'Spitalul Municipal Targu Mures' #4 nume_persoana varchar(50):= null</pre> <p><b>Execute</b>    <b>Close</b></p> | <pre>1555 ✓ declare 1556   data_startstring varchar2(15) := &amp;data_st; 1557   data_finishingstring varchar2(10) :=&amp;data_fi; 1558   data_finish date:=null; 1559   data_start date:=null; 1560   nume_spital varchar(60) := &amp;nume_spital; 1561   nume_persoana varchar(50):= &amp;nume_persoana; 1562   begin 1563     data_finish := to_date(data_finishingstring,'YYYY-MM-DD'); 1564     data_start:=to_date(data_startstring,'YYYY-MM-DD'); 1565 1566     Report_Medical(data_start =&gt; data_start, data_finish =&gt; data_finish, numespital =&gt; nume_spital, nume_persoana=&gt; nume_persoana) 1567   end;</pre> <p>select * from RETETE where DATA_ELIBERARI BETWEEN to_date('2023-09-01','YYYY-MM-DD') AND to_date('2023-10-01','YYYY-MM-DD')</p> <p>ms</p> <table border="1"> <thead> <tr> <th>ID_SPITAL</th> <th>ID_STRADA</th> <th>NUM_SPITAL</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>55 Spitalul Municipal Targu Mures</td> </tr> <tr> <td>2</td> <td>2</td> <td>55 Spitalul Judetean Piatra Neamt</td> </tr> <tr> <td>3</td> <td>3</td> <td>60 Spitalul Municipal Slatina</td> </tr> <tr> <td>4</td> <td>4</td> <td>62 Spitalul de Urgenta Plouzești</td> </tr> <tr> <td>5</td> <td>5</td> <td>64 Spitalul Județean Satu Mare</td> </tr> <tr> <td>6</td> <td>6</td> <td>66 Spitalul Municipal Zalău</td> </tr> </tbody> </table> <p>Problems    Services</p> | ID_SPITAL                         | ID_STRADA  | NUM_SPITAL     | 1 | 1 | 55 Spitalul Municipal Targu Mures | 2 | 2  | 55 Spitalul Judetean Piatra Neamt | 3 | 3  | 60 Spitalul Municipal Slatina | 4 | 4  | 62 Spitalul de Urgenta Plouzești | 5 | 5 | 64 Spitalul Județean Satu Mare | 6 | 6 | 66 Spitalul Municipal Zalău |
| ID_SPITAL   | ID_STRADA   | NUM_SPITAL                        |            |                |   |   |                                   |   |    |                                   |   |    |                               |   |    |                                  |   |   |                                |   |   |                             |
| 1   | 1   | 55 Spitalul Municipal Targu Mures |            |                |   |   |                                   |   |    |                                   |   |    |                               |   |    |                                  |   |   |                                |   |   |                             |
| 2   | 2   | 55 Spitalul Judetean Piatra Neamt |            |                |   |   |                                   |   |    |                                   |   |    |                               |   |    |                                  |   |   |                                |   |   |                             |
| 3   | 3   | 60 Spitalul Municipal Slatina     |            |                |   |   |                                   |   |    |                                   |   |    |                               |   |    |                                  |   |   |                                |   |   |                             |
| 4   | 4   | 62 Spitalul de Urgenta Plouzești  |            |                |   |   |                                   |   |    |                                   |   |    |                               |   |    |                                  |   |   |                                |   |   |                             |
| 5   | 5   | 64 Spitalul Județean Satu Mare    |            |                |   |   |                                   |   |    |                                   |   |    |                               |   |    |                                  |   |   |                                |   |   |                             |
| 6   | 6   | 66 Spitalul Municipal Zalău       |            |                |   |   |                                   |   |    |                                   |   |    |                               |   |    |                                  |   |   |                                |   |   |                             |

## Exercitiul 7

Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat, dependent de celălalt cursor. Apelați subprogramul

**Direcția spitalului dorește un raport detaliat privind pacienții care au solicitat servicii de urgență în alte spitale, în afara localității de domiciliu, pe parcursul unui an specificat. Scopul acestui raport este de a identifica și analiza nevoile urgente ale comunității în afara facilităților medicale disponibile local.**

```

create or replace type number_list is varray(40) of number;

create or replace procedure PacientiSpitaleInfo(
an in date default null
)
is

--cursoare
cursor domiciliat is (select id_domiciliu,s.ID_ORAS from Domicilii left join Strazi s on
s.ID_STRADA = Domicilii.ID_STRADA);
cursor pacie(dom Domiciliu.ID_DOMICILIU%type) is select NUME || ' ' ||
PRENUME,Data_Nastere,J.NUME_DOMENIU,ID_PACIENT from Pacienti
left join Joburi J on PACIENTI.ID_JOB = J.ID_JOB
where dom = ID_DOMICILIU;
cursor spitale(istoric number_list, domoras Strazi.ID_ORAS%type) is (select
s2.NUME_SPITAL,s3.ID_ORAS from Medici m left join SPITALE s2 on m.ID_SPITAL = s2.ID_SPITAL
left join Strazi s3 on
s3.ID_STRADA = s2.ID_SPITAL
where m.ID_MEDIC in (select * from table ( istoric
)) and ID_ORAS != domoras);
--- variabile
datepersonale personale;
spit hospital;
tabelamedici number_list;
contor number := 0;
anul_end date;

--exceptii
begin
if an is not null then
    anul_end := add_months(an,12);
end if;
select count(*) into contor from retete where an is null or DATA_ELIBERARII between an and
anul_end;
if contor = 0 then
    raise no_data_found;
end if;
for dom in domiciliat loop
    open pacie(dom.ID_DOMICILIU);

loop
    contor :=0;
    fetch pacie into datepersonale;

```

```

        exit when pacie%notfound;
        select ID_MEDIC bulk collect into tabelamedici from ISTORIC_PACIENTI IP left join
RETETE R2 on R2.ID_RETETA = IP.ID_RETETA where ID_PACIENT = datepersonale.cod and (an is null
or r2.DATA_LIBERARII between an and anul_end); -- medicii care i-au facut ceva pacientului
--- obtinem spitale din toata tara la care a fost

        open spitale(tabelamedici,dom.ID_ORAS);
        fetch spitale into spit.Nume,spit.codoras;

        if spitale%found then
            DBMS_OUTPUT.PUT_LINE('Nume Complet: '||datepersonale.NumeComplet);
            DBMS_OUTPUT.PUT_LINE('Data de nastere: '|| datepersonale.datanastere);
            DBMS_OUTPUT.PUT_LINE('Domeniu de lucru: '|| datepersonale.domeniu);
            DBMS_OUTPUT.PUT_LINE('Spitalele la care a fost in afara Orasului: ');
            loop

                exit when spitale%notfound;
                contor:= contor+1;
                select NUME_ORAS into spit.numeOras from orase where ID_ORAS =
spit.codoras;
                DBMS_OUTPUT.PUT_LINE('- '||spit.Nume||', '||spit.numeOras || ';');
                fetch spitale into spit.Nume,spit.codoras;

                end loop;
                DBMS_OUTPUT.PUT_LINE('Total de plata suplimentar:'||contor*100);
                DBMS_OUTPUT.PUT_LINE('=====');

            end if;
            close spitale;

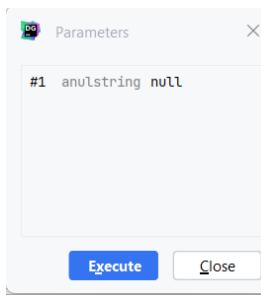
        end loop;
        close pacie;

    end loop;
exception
    when no_data_found then
        DBMS_OUTPUT.PUT_LINE('Nu exista pacientii pentru anii respectivi!');
    when others then
        DBMS_OUTPUT.PUT_LINE('Contacteaza administratorul si precizeaza codul de eroare!
[,'||sqlcode||', '|| SQLERRM||']');
        end PacientiSpitaleInfo;

declare
    anulstring varchar(5) := &anul;
    anstart date;
    begin
        anstart:= to_date(anulstring,'YYYY');
        PacientiSpitaleInfo(an => anstart);
    end;

```

## Rezultatele apelarii procedurii

| Date de intrare  | Output   |
|--|--|
|  <p>Parameters</p> <pre>#1 anulstring null</pre> <p>Execute Close</p> | <pre> 1650   end loop; 1651   exception 1652     when no_data_found then 1653       DBMS_OUTPUT.PUT_LINE( 'Nu exista pacientii pentru anii respectivi!'); 1654     when others then 1655       DBMS_OUTPUT.PUT_LINE( 'Contacteaza administratorul si precizeaza codul de eroare! ['  sqlcode  ', '   SQLERRM  ']"); 1656   end PacientiSpitaleInfo; 1657   --- exercitiul 8 1658 1659  declare 1660    anulstring varchar(5) := &amp;anul; 1661    anstart date; 1662    begin 1663      anstart:= to_date(anulstring,'YYYY'); 1664      PacientiSpitaleInfo(an =&gt; anstart); 1665    end; 1666 1667 1668 1669 1670 1671 1672 1673 1674 1675 1676 1677 1678 1679 1680 1681 1682 1683 1684 1685 1686 1687 1688 1689 1690 1691 1692 1693 1694 1695 1696 1697 1698 1699 1700 1701 1702 1703 1704 1705 1706 1707 1708 1709 1710 1711 1712 1713 1714 1715 1716 1717 1718 1719 1720 1721 1722 1723 1724 1725 1726 1727 1728 1729 1730 1731 1732 1733 1734 1735 1736 1737 1738 1739 1740 1741 1742 1743 1744 1745 1746 1747 1748 1749 1750 1751 1752 1753 1754 1755 1756 1757 1758 1759 1760 1761 1762 1763 1764 1765 1766 1767 1768 1769 1770 1771 1772 1773 1774 1775 1776 1777 1778 1779 1780 1781 1782 1783 1784 1785 1786 1787 1788 1789 1790 1791 1792 1793 1794 1795 1796 1797 1798 1799 1800 1801 1802 1803 1804 1805 1806 1807 1808 1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821 1822 1823 1824 1825 1826 1827 1828 1829 1830 1831 1832 1833 1834 1835 1836 1837 1838 1839 1840 1841 1842 1843 1844 1845 1846 1847 1848 1849 1850 1851 1852 1853 1854 1855 1856 1857 1858 1859 1860 1861 1862 1863 1864 1865 1866 1867 1868 1869 1870 1871 1872 1873 1874 1875 1876 1877 1878 1879 1880 1881 1882 1883 1884 1885 1886 1887 1888 1889 1890 1891 1892 1893 1894 1895 1896 1897 1898 1899 1900 1901 1902 1903 1904 1905 1906 1907 1908 1909 1910 1911 1912 1913 1914 1915 1916 1917 1918 1919 1920 1921 1922 1923 1924 1925 1926 1927 1928 1929 1930 1931 1932 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942 1943 1944 1945 1946 1947 1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032 2033 2034 2035 2036 2037 2038 2039 2040 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050 2051 2052 2053 2054 2055 2056 2057 2058 2059 2060 2061 2062 2063 2064 2065 2066 2067 2068 2069 2070 2071 2072 2073 2074 2075 2076 2077 2078 2079 2080 2081 2082 2083 2084 2085 2086 2087 2088 2089 2090 2091 2092 2093 2094 2095 2096 2097 2098 2099 20100 20101 20102 20103 20104 20105 20106 20107 20108 20109 20110 20111 20112 20113 20114 20115 20116 20117 20118 20119 20120 20121 20122 20123 20124 20125 20126 20127 20128 20129 20130 20131 20132 20133 20134 20135 20136 20137 20138 20139 20140 20141 20142 20143 20144 20145 20146 20147 20148 20149 20150 20151 20152 20153 20154 20155 20156 20157 20158 20159 20160 20161 20162 20163 20164 20165 20166 20167 20168 20169 20170 20171 20172 20173 20174 20175 20176 20177 20178 20179 20180 20181 20182 20183 20184 20185 20186 20187 20188 20189 20190 20191 20192 20193 20194 20195 20196 20197 20198 20199 20200 20201 20202 20203 20204 20205 20206 20207 20208 20209 20210 20211 20212 20213 20214 20215 20216 20217 20218 20219 20220 20221 20222 20223 20224 20225 20226 20227 20228 20229 202210 202211 202212 202213 202214 202215 202216 202217 202218 202219 202220 202221 202222 202223 202224 202225 202226 202227 202228 202229 202230 202231 202232 202233 202234 202235 202236 202237 202238 202239 202240 202241 202242 202243 202244 202245 202246 202247 202248 202249 202250 202251 202252 202253 202254 202255 202256 202257 202258 202259 202260 202261 202262 202263 202264 202265 202266 202267 202268 202269 202270 202271 202272 202273 202274 202275 202276 202277 202278 202279 202280 202281 202282 202283 202284 202285 202286 202287 202288 202289 202290 202291 202292 202293 202294 202295 202296 202297 202298 202299 2022100 2022101 2022102 2022103 2022104 2022105 2022106 2022107 2022108 2022109 2022110 2022111 2022112 2022113 2022114 2022115 2022116 2022117 2022118 2022119 2022120 2022121 2022122 2022123 2022124 2022125 2022126 2022127 2022128 2022129 2022130 2022131 2022132 2022133 2022134 2022135 2022136 2022137 2022138 2022139 2022140 2022141 2022142 2022143 2022144 2022145 2022146 2022147 2022148 2022149 2022150 2022151 2022152 2022153 2022154 2022155 2022156 2022157 2022158 2022159 20221510 20221511 20221512 20221513 20221514 20221515 20221516 20221517 20221518 20221519 20221520 20221521 20221522 20221523 20221524 20221525 20221526 20221527 20221528 20221529 202215210 202215211 202215212 202215213 202215214 202215215 202215216 202215217 202215218 202215219 202215220 202215221 202215222 202215223 202215224 202215225 202215226 202215227 202215228 202215229 202215230 202215231 202215232 202215233 202215234 202215235 202215236 202215237 202215238 202215239 202215240 202215241 202215242 202215243 202215244 202215245 202215246 202215247 202215248 202215249 202215250 202215251 202215252 202215253 202215254 202215255 202215256 202215257 202215258 202215259 202215260 202215261 202215262 202215263 202215264 202215265 202215266 202215267 202215268 202215269 202215270 202215271 202215272 202215273 202215274 202215275 202215276 202215277 202215278 202215279 202215280 202215281 202215282 202215283 202215284 202215285 202215286 202215287 202215288 202215289 202215290 202215291 202215292 202215293 202215294 202215295 202215296 202215297 202215298 202215299 2022152100 2022152101 2022152102 2022152103 2022152104 2022152105 2022152106 2022152107 2022152108 2022152109 2022152110 2022152111 2022152112 2022152113 2022152114 2022152115 2022152116 2022152117 2022152118 2022152119 2022152120 2022152121 2022152122 2022152123 2022152124 2022152125 2022152126 2022152127 2022152128 2022152129 2022152130 2022152131 2022152132 2022152133 2022152134 2022152135 2022152136 2022152137 2022152138 2022152139 2022152140 2022152141 2022152142 2022152143 2022152144 2022152145 2022152146 2022152147 2022152148 2022152149 2022152150 2022152151 2022152152 2022152153 2022152154 2022152155 2022152156 2022152157 2022152158 2022152159 2022152160 2022152161 2022152162 2022152163 2022152164 2022152165 2022152166 2022152167 2022152168 2022152169 2022152170 2022152171 2022152172 2022152173 2022152174 2022152175 2022152176 2022152177 2022152178 2022152179 2022152180 2022152181 2022152182 2022152183 2022152184 2022152185 2022152186 2022152187 2022152188 2022152189 2022152190 2022152191 2022152192 2022152193 2022152194 2022152195 2022152196 2022152197 2022152198 2022152199 2022152200 2022152201 2022152202 2022152203 2022152204 2022152205 2022152206 2022152207 2022152208 2022152209 2022152210 2022152211 2022152212 2022152213 2022152214 2022152215 2022152216 2022152217 2022152218 2022152219 2022152220 2022152221 2022152222 2022152223 2022152224 2022152225 2022152226 2022152227 2022152228 2022152229 2022152230 2022152231 2022152232 2022152233 2022152234 2022152235 2022152236 2022152237 2022152238 2022152239 2022152240 2022152241 2022152242 2022152243 2022152244 2022152245 2022152246 2022152247 2022152248 2022152249 2022152250 2022152251 2022152252 2022152253 2022152254 2022152255 2022152256 2022152257 2022152258 2022152259 2022152260 2022152261 2022152262 2022152263 2022152264 2022152265 2022152266 2022152267 2022152268 2022152269 2022152270 2022152271 2022152272 2022152273 2022152274 2022152275 2022152276 2022152277 2022152278 2022152279 2022152280 2022152281 2022152282 2022152283 2022152284 2022152285 2022152286 2022152287 2022152288 2022152289 2022152290 2022152291 2022152292 2022152293 2022152294 2022152295 2022152296 2022152297 2022152298 2022152299 20221522100 20221522101 20221522102 20221522103 20221522104 20221522105 20221522106 20221522107 20221522108 20221522109 20221522110 20221522111 20221522112 20221522113 20221522114 20221522115 20221522116 20221522117 20221522118 20221522119 20221522120 20221522121 20221522122 20221522123 20221522124 20221522125 20221522126 20221522127 20221522128 20221522129 20221522130 20221522131 20221522132 20221522133 20221522134 20221522135 20221522136 20221522137 20221522138 20221522139 20221522140 20221522141 20221522142 20221522143 20221522144 20221522145 20221522146 20221522147 20221522148 20221522149 20221522150 20221522151 20221522152 20221522153 20221522154 20221522155 20221522156 20221522157 20221522158 20221522159 20221522160 20221522161 20221522162 20221522163 20221522164 20221522165 20221522166 20221522167 20221522168 20221522169 20221522170 20221522171 20221522172 20221522173 20221522174 20221522175 20221522176 20221522177 20221522178 20221522179 20221522180 20221522181 20221522182 20221522183 20221522184 20221522185 20221522186 20221522187 20221522188 20221522189 20221522190 20221522191 20221522192 20221522193 20221522194 20221522195 20221522196 20221522197 20221522198 20221522199 20221522200 20221522201 20221522202 20221522203 20221522204 20221522205 20221522206 20221522207 20221522208 20221522209 20221522210 20221522211 20221522212 20221522213 20221522214 20221522215 20221522216 20221522217 20221522218 20221522219 20221522220 20221522221 20221522222 20221522223 20221522224 20221522225 20221522226 20221522227 20221522228 20221522229 202215222210 202215222211 202215222212 202215222213 202215222214 202215222215 202215222216 202215222217 202215222218 202215222219 202215222220 202215222221 202215222222 202215222223 202215222224 202215222225 202215222226 202215222227 202215222228 202215222229 2022152222210 2022152222211 2022152222212 2022152222213 2022152222214 2022152222215 2022152222216 2022152222217 2022152222218 2022152222219 2022152222220 2022152222221 2022152222222 2022152222223 2022152222224 2022152222225 2022152222226 2022152222227 2022152222228 2022152222229 20221522222210 20221522222211 20221522222212 20221522222213 20221522222214 20221522222215 20221522222216 20221522222217 20221522222218 20221522222219 20221522222220 20221522222221 20221522222222 20221522222223 20221522222224 20221522222225 20221522222226 20221522222227 20221522222228 20221522222229 202215222222210 202215222222211 202215222222212 202215222222213 202215222222214 202215222222215 202215222222216 202215222222217 202215222222218 202215222222219 202215222222220 202215222222221 202215222222222 202215222222223 202215222222224 202215222222225 202215222222226 202215222222227 202215222222228 202215222222229 2022152222222210 2022152222222211 2022152222222212 2022152222222213 2022152222222214 2022152222222215 2022152222222216 2022152222222217 2022152222222218 2022152222222219 2022152222222220 2022152222222221 2022152222222222 2022152222222223 2022152222222224 2022152222222225 2022152222222226 2022152222222227 2022152222222228 2022152222222229 20221522222222210 20221522222222211 20221522222222212 20221522222222213 20221522222222214 20221522222222215 20221522222222216 20221522222222217 20221522222222218 20221522222222219 20221522222222220 20221522222222221 20221522222222222 20221522222222223 20221522222222224 20221522222222225 20221522222222226 20221522222222227 20221522222222228 20221522222222229 2</pre> |

## Exercitiul 8

Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele definite. Definiți minim 2 excepții proprii. Apelați subprogramul astfel încât să evidențiați toate cazurile definite și tratate.

**Un pacient dorește să își schimbe adresa de domiciliu și vine la spital pentru a updatea informațiile din baza de date. Acesta poate să vina și cu cererea de a-și schimba și locul de muncă trecut în baza de date.**

**Conform regulilor Spitalului facând aceste schimbări pacientului trebuie să i se elibereze Istoricul Medical, iar în cazul schimbării locului de muncă eliberarea fisei în care sunt precizate bolile specifice ale fostului loc de munca.**

Pentru rezolvarea acestui exercițiu am realizat următoarele:

Am creat un obiect de tipul adresa

```
create or replace type Adresa as Object(
    judet varchar2(50),
    oras varchar2(50),
    strada varchar2(50),
    numar number,
    bloc number,
    apartament number,
    scara varchar2(2),
    Member function coduri return number_list,
    constructor function Adresa(numejudet varchar2, numeoras varchar2, strada varchar2, p_numar
    number, bl number, ap number, sc varchar2) return self as result,
    member function GetJudet return varchar2,
    member function GetOras return varchar2,
    member function GetStrada return varchar2,
    member function GetNumar return number,
    member function GetBloc return number,
    member function GetApartament return number,
    member function GetScara return varchar2,
    member function Afisareadresa return varchar2
);
create or replace type body Adresa as
    constructor function Adresa(numejudet varchar2, numeoras varchar2, strada varchar2, p_numar
    number, bl number, ap number, sc varchar2) return self as result
        is begin
            self.judet := numejudet;
            self.ORAS:=numeoras;
            self.STRADA:=strada;
            self.NUMAR:=p_numar;
            self.BLOC:=bl;
            self.APARTAMENT:=ap;
            self.SCARA:=sc;
            return;
        end Adresa;

    member function Afisareadresa return varchar2 is
        adresa_completa varchar2(100);
```

```

begin
    if self.JUDET is null or self.STRADA is null or self.ORAS is null or self.NUMAR is
null then
        return 'Adresa incompleta!';
    end if;
    adresa_completa:= 'Jud.' || self.JUDET || ', Mun.' || self.ORAS || ', Str.' ||
self.STRADA || ', nr.' || self.NUMAR;
    if self.APARTAMENT is not null and self.BLOC is not null and self.SCARA is not null
then
        adresa_completa:= adresa_completa || ' bl.' || self.BLOC || ' sc.' || self.SCARA
|| 'ap.' || self.APARTAMENT;
    else
        return adresa_completa;
    end if;
    return adresa_completa;

    end Afisareadresa;
member function coduri return number_list is
cod number_list:=number_list();

begin
    cod.extend(3);

    cod(1):=-1;cod(2):=-1; cod(3):=-1;
select ID_JUDET into cod(1) from Judet where NUME_JUDET = SELF.JUDET;
select ID_ORAS into cod(2) from ORAS where NUME_ORAS = SELF.ORAS;
select ID_STRADA into cod(3) from STRADA where NUME_STRADA = SELF.STRADA;
return cod;
exception
    when no_data_found then
        return cod;
end coduri;

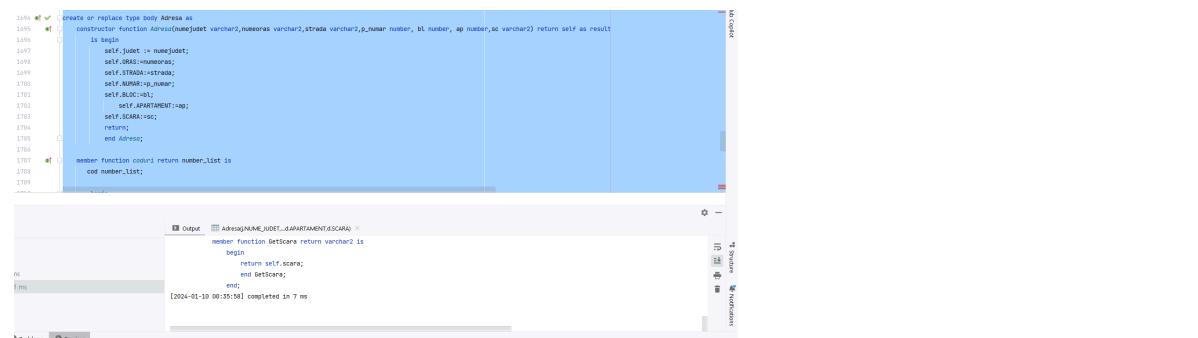
-- geteri si setteri
member function GetJudet return varchar2 is
begin
    return self.judet;
    end GetJudet;
member function GetOras return varchar2 is
begin
    return self.oras;
    end GetOras;
member function GetStrada return varchar2 is
begin
    return self.strada;
    end GetStrada;
member function GetNumar return number is
begin
    return self.numar;
    end GetNumar;
member function GetBloc return number is
begin
    return self.bloc;
    end GetBloc;
member function GetApartament return number is
begin
    return self.apartament;
end

```

```

    end GetApartament;
member function GetScara return varchar2 is
begin
    return self.scara;
end GetScara;
end;

```



```

1594 ✓ create or replace type body Adresa as
1595   constructor function Adresa(numejudet varchar2,oras varchar2,strada varchar2,p_numar number, d1 number, ap number,sc varchar2) return self as result
1596   is begin
1597     self.JUDET := numejudet;
1598     self.ORAIS := oras;
1599     self.IDSTRADA := strada;
1600     self.NUMAR := p_numar;
1601     self.D1 := d1;
1602     self.APARTAMENT := ap;
1603     self.SCARA := sc;
1604     return;
1605   end Adresa;
1606
1607   member function coduri return number list is
1608   cod number list;
1609

```

Output: Adresa(NUME\_JUDET,...APARTAMENT,SCARA)

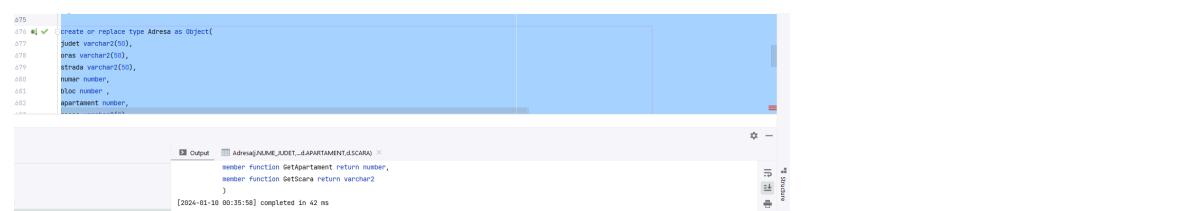
```

    member function GetScara return varchar2 is
    begin
        return self.scara;
    end GetScara;
end;

```

[2024-01-10 00:39:58] completed in 7 ms

Problems Services



```

675 ✓ create or replace type Adresa as Object(
676   judest varchar2(10),
677   oras varchar2(10),
678   strada varchar2(10),
679   numar,
680   bloc number,
681   apartament number,
682   );

```

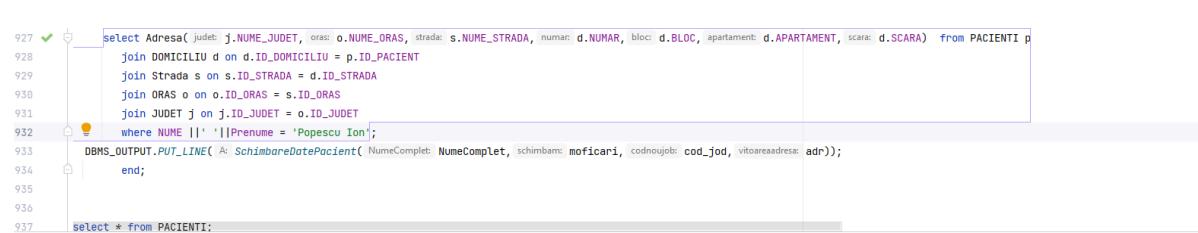
```

676 ✓ Output: Adresa(NUME_JUDET,...APARTAMENT,SCARA)
677   member function GetScara return varchar2;
678   member function GetApartament return number;
679

```

[2024-01-10 00:35:58] completed in 42 ms

UTI:7770DB: create or replace type body Adresa as



```

927 ✓ select Adresa(j.judest, o.oras, s.strada, n.numar, b.bloc, a.apartament, s.scara) from PACIENTI p
928   join DOMICILIU d on d.ID_DOMICILIU = p.ID_PACIENT
929   join Strada s on s.ID_STRADA = d.ID_STRADA
930   join ORAS o on o.ID_ORAS = s.ID_ORAS
931   join JUDET j on j.ID_JUDET = o.ID_JUDET
932   where NUME ||' '||Prenume = 'Popescu Ion';
933   DBMS_OUTPUT.PUT_LINE('SchimbareDatePacient(NumeComplet: NumeCompleta, schimbam: moficari, codnoujob: cod_jed, vitoareaadresa: adr)');
934
935
936
937 select * from PACIENTI;

```

Output: Adresa(NUME\_JUDET,...APARTAMENT,SCARA)

ADRESA(J.NUME\_JUDET,O.NUME\_ORAS,S.NUME\_STRADA,D.NUMAR,D.BLOC,D.APARTAMENT,D.SCARA)

1 {Arad,Arad,Bulevardul Revolutiei,12,6,3,8}

Două proceduri stocate **AfisareIstoricPacient** și **BoliJob** care o să afiseze istoricul pacientului și bolile specifice la un anumit job.

```

create or replace procedure AfisareIstoricPacient(codpacient
ISTORIC_PACIENTI.ID_PACIENT%type)
is
  istoric_not_found exception;
  pragma exception_init ( istoric_not_found,-20004);
  ---tipuri de date
  type reteta is record( data_eliberarii date, Med Lista,nume_medic
varchar(50),specializare SPECIALIZARI.NUME_SPECIALIZARE%type);
  type raport is table of reteta;
  ---variabile
  Istoric raport;

```

```

begin
    select r2.DATA_ELIBERARII,r2.Lista_Medicamente,M.NUME || ' ' ||
M.PRENUME,s2.NUME_SPECIALIZARE bulk collect into Istoric from ISTORIC_PACIENTI
        left join RETETE R2 on ISTORIC_PACIENTI.ID_RETETA = R2.ID_RETETA
        left join MEDICI M on ISTORIC_PACIENTI.ID_MEDIC = M.ID_MEDIC
        left join SPECIALIZARI S2 on M.ID_SPECIALIZARE = S2.ID_SPECIALIZARE
            where ID_PACIENT = codpacient;
    if Istoric.COUNT = 0 then
        raise_application_error(-20004,'Nu exista istoricul pacientului!');
    end if;
    for i in 1..Istoric.COUNT loop
        DBMS_OUTPUT.PUT_LINE('Dr. '||Istoric(i).nume_medic||', Specializare: '||Istoric(i).specializare);
        DBMS_OUTPUT.PUT_LINE('A eliberat in data:' || Istoric(i).data_eliberarii);
        DBMS_OUTPUT.PUT_LINE('Reteta care cuprinde medicamentele: ');
        for medicamente in (select NUME_MEDICAMENT from MEDICAMENTE where ID_MEDICAMENT
in (select * from table ( Istoric(i).Med ))) loop
            DBMS_OUTPUT.PUT_LINE(' - '||medicamente.NUME_MEDICAMENT);
        end loop;
    DBMS_OUTPUT.PUT_LINE('=====');
    end loop;
end AfisareIstoricPacient;

```

754 ✓ create or replace procedure AfisareIstoricPacient(codpacient ISTORIC\_PACIENTI.ID\_PACIENT%type)
755 is
756
757
758 istoric\_not\_found exception;
759 pragma exception\_init ( istoric\_not\_found,-20004);
760 --tipuri de date
761
762 type reteta is record( data\_eliberarii date, Med Lista,nume\_medic varchar(30),specializare SPECIALIZARI.NUME\_SPECIALIZARE%type);
763 type report is table of reteta;
764 --variabile
765
766 Istoric report;
767
768 begin
 select r2.DATA\_ELIBERARII,r2.Lista\_Medicamente,M.NUME || ' ' || M.PRENUME,s2.NUME\_SPECIALIZARE bulk collect into Istoric from ISTORIC\_PACIENTI
 left join RETETE R2 on ISTORIC\_PACIENTI.ID\_RETETA = R2.ID\_RETETA

The screenshot shows the Oracle SQL Developer interface. The code editor at the top contains the PL/SQL procedure definition. Below it, the 'Output' tab of the results window is active, displaying the generated report for a patient. The report includes the doctor's name, specializations, and the list of medications prescribed. The execution time is shown as 17 ms.

```

create or replace procedure BoliJob(codjob number) is
    cursor boli(idj number)      is (select j.NUME_DOMENIU,bs.NUME_BOALA,bs.OBSERVATII from
JOB_PREDISPUSI_BOLI p
        inner join Joburi j on p.ID_JOB = j.ID_JOB
        join BOLI_SPECIFICE BS on BS.ID_BOALA = p.ID_BOALA
        where p.ID_JOB = idj);
    type bolitot is record(domeniu joburi.NUME_DOMENIU%type, boala
BOLI_SPECIFICE.NUME_BOALA%type, obs BOLI_SPECIFICE.OBSERVATII%type);
    dbo bolitot;
    no_boli exception ;
begin
    open boli(codjob);
    fetch boli into dbo;
    if boli%notfound then
        raise no_boli;
    end if;
    DBMS_OUTPUT.PUT_LINE('DOMENIU :'|| DBO.domeniu);
    loop
        DBMS_OUTPUT.PUT_LINE('Nume boala :'|| DBO.boala);
        DBMS_OUTPUT.PUT_LINE('Observatii :'|| DBO.obs);
        fetch boli into dbo;
        exit when boli%notfound;
    end loop;
end BoliJob;

```

```

        end loop;
exception
when no_boli then
    DBMS_OUTPUT.PUT_LINE('Jobul nu are boli specifice!');
when others then
    DBMS_OUTPUT.PUT_LINE(sqlerrm);

end BoliJob;

1790  ✓ create or replace procedure BoliJob(codjob number) is
1791      cursor boliti(id number)  is (select j.NUME_DOMENIU,bs.NUME_BOALA,bs.OBSERVATII from JOB_PREDISPOZITI_BOLI p
1792          inner join Job j on p.ID_JOB = j.ID_JOB
1793          join BOLI_SPECIFICE bs on bs.ID_BOALA = p.ID_BOALA
1794          where p.ID_JOB = id);
1795      type boliti is record(domenu job.NUME_DOMENIU%type, boala BOLI_SPECIFICE.NUME_BOALA%type, obs BOLI_SPECIFICE.OBSERVATII%type);
1796      boliti;
1797      no_boli exception;
1798  begin
1799      if no_boli then
2000          DBMS_OUTPUT.PUT_LINE('Jobul nu are boli specifice!');
2001          when others then
2002              DBMS_OUTPUT.PUT_LINE(sqlerrm);
2003      end if;
2004  end BoliJob;

```

### Funcția principală pentru rezolvarea exercitiului 8:

```

create or replace SchimbareDatePacient
(NumeComplet varchar2, schimbam_in varchar2, codnoujob in number, vitoareaadresa Adresa)
return varchar2
is
-- exceptii

incorrect_nume_judet exception;
no_job_specified exception ;
same_domiciliu exception ;
no_town_street exception ;
done_job exception ;

-- tipuri de date

--variabile
type informatiidomiciliu is record (
    codpacient PACIENTI.ID_PACIENT%type,
    coddomiciliu DOMICILII.Id_domiciliu%type,
    codstrada Strazi.id_strada%type,
    codoras orase.id_oras%type,
    codjob JOBURI.id_job%type
);
oldinfo informatiidomiciliu;
iduri number_list :=number_list();
cod number;
dom number:=-1;

begin

    select p.ID_PACIENT,d.ID_DOMICILIU,d.ID_STRADA,s.ID_ORAS,j.ID_JOB into oldinfo from
PACIENTI p
    join Domicilii d on d.ID_DOMICILIU = p.ID_PACIENT

```

```

join Strazi s on s.ID_STRADA = d.ID_STRADA
join joburi J on p.ID_JOB = J.ID_JOB
where NUME ||' '|| Prenume = NumeCompleta;

if schimbam = 'Job' or schimbam is null then
    if codnoujob is null then
        raise no_job_specified;
    end if;
    BoliJob(oldinfo.codjob);
    update PACIENTI
    set ID_JOB = codnoujob
    where ID_PACIENT = oldinfo.codpatient;
    if schimbam = 'Job' then
        raise done_job;
    end if;

end if;
iduri.extend(3);
iduri:= vitoareaadresa.CODURI();
if vitoareaadresa.GetStrada() is null or vitoareaadresa.GetOras() is null or
vitoareaadresa.GetNumar() is null then
    raise no_town_street;
end if;

if iduri(2) = -1 then
    select ID_ORAS+1 into cod from(select * from Orase order by ID_ORAS desc) where
ROWNUM=1;

    insert into Orase(id_oras, id_judet, nume_oras) VALUES
(cod,iduri(1),vitoareaadresa.GETORAS());
    DBMS_OUTPUT.PUT_LINE('S-a adaugat in baza de date strada: ' ||
vitoareaadresa.GetOras());

    iduri(2):=cod;
    DBMS_OUTPUT.PUT_LINE('Strada');
    select ID_STRADA+1 into cod from(select * from strazi order by ID_STRADA desc)
where ROWNUM=1;
    insert into Strazi(id_strada, nume_strada, id_oras) VALUES
(cod,vitoareaadresa.GETSTRADA(),iduri(2));
    DBMS_OUTPUT.PUT_LINE('S-a adaugat in baza de date strada: ' ||
vitoareaadresa.GetStrada());

    iduri(3):=cod;
    elsif iduri(3) =-1 then --- daca nu exista strada in bd

        select ID_STRADA+1 into cod from(select * from Strazi order by ID_STRADA desc)
where ROWNUM=1;
        insert into Strazi(id_strada, nume_strada, id_oras) VALUES
(cod,vitoareaadresa.GETSTRADA(),iduri(2));
        DBMS_OUTPUT.PUT_LINE('S-a adaugat in baza de date strada: ' ||
vitoareaadresa.GetStrada());

```

```

        iduri(3) :=cod;

    end if;
--- daca exista strada si orasul, verificam daca exista domiciliul respectiv
begin
    select ID_DOMICILIU into dom from Domicilii where ID_STRADA = iduri(3)
    and numar = vitoareaadresa.GETNUMAR() and BLOC = vitoareaadresa.GETBLOC() and
APARTAMENT = vitoareaadresa.GETAPARTAMENT() and SCARA = vitoareaadresa.GETSCARA();
    if dom = oldinfo.coddomiciliu then
        raise same_domiciliu;
    end if;
    exception
    when no_data_found then
        dom := -1;
end;

if dom != -1 then
    update PACIENTI set ID_DOMICILIU = dom
    where ID_PACIENT = oldinfo.codpacient;
else
    select ID_DOMICILIU+1 into cod from(select * from domicilii order by
ID_DOMICILIU desc) where ROWNUM=1;
    insert into Domicilii(id_domiciliu, id_strada, numar,bloc,APARTAMENT,SCARA)
VALUES
(cod,iduri(3),vitoareaadresa.GETNUMAR(),vitoareaadresa.GETBLOC(),vitoareaadresa.GETAPARTAMENT
(),vitoareaadresa.GetScara());
    update PACIENTI set ID_DOMICILIU = cod
    where ID_PACIENT = oldinfo.codpacient;
end if;

AfisareIstoricPacient(oldinfo.codpacient);
return 'Noua adresa: ' || vitoareaadresa.AFISAREADRESA();
exception
when no_data_found then
return 'Nu exista pacientul in baza de date';
when too_many_rows then
return 'Mai multi pacienti cu acelasi nume!';
when no_town_street then
return 'Nu putem sa cautam/adaugam o adresa fara Oras/Strada/Numar';
when same_domiciliu then
return 'Persoana vrea sa se mute in aceeasi casa';
when incorrect_nume_judet then
return 'Nu a fost acordat un nume de judet corect, nu s-a schimbat domiciliul!';
when no_job_specified then
return 'Nu ai specificat un job, dar vrei sa il schimbi!';
when done_job then
return 'Am schimbat cu succes locul de munca al pacientului!';
when others then
return sqlerrm;

end SchimbareDatePacient;

```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor with a large blue background. The code is a PL/SQL function named `SchimbareDatePacient`. The function takes three parameters: `CNameCompleta` (varchar2), `schimbam` (varchar2), and `codnoujob` (number). It returns a varchar2 value. The function body includes exception handling for `incorrect_num_judet`, `no_job_specified`, and `same_domiciliu`. It also contains comments about date types and variables. A cursor is declared at the end of the function body.

```
1820 1821 1822 1823 1824 1825 1826 1827 1828 1829 1830 1831 1832
create or replace function SchimbareDatePacient
(CNameCompleta varchar2, schimbam in varchar2,codnoujob in number, vitoareadresa Adresa)
return varchar2
is
-- exceptii
incorrect_num_judet exception;
no_job_specified exception ;
same_domiciliu exception ;
-- tipuri de date
--variabile
cursor cur1 is
  select * from Adresa(j.NUME_JUDET,d.APARTAMENT,d.SCARA)
```

In the bottom-right pane, there is an "Output" tab showing the results of the function execution. The output shows the function was completed in 16 ms. The output text is:

```
when others then
    return sqlerrm;
```

```
ms
[2024-01-10 00:41:28] completed in 16 ms
```

## Rezultatele apelarii functiei

| Input  | Output   |
|--|--|
| <p>Parameters</p> <pre>#18 judet varchar2(50):= 'Arad' #19 oras varchar2(50):= 'Arad' #20 stra varchar2(50) := 'Bulevardul Revoluției' #21 numb number:= 12 #22 bloc number:= 6 #23 apt number:= 3 #24 sc varchar(2):=_ 'B' #25 NumeComplet varchar: 'Popescu Ion' #26 moficari varchar2(50) 'Domiciliu' #27 cod_jod number := _ null</pre> <p>Execute Close</p> | <pre>1925 ✓ declare 1926   judet varchar2(50):=&amp;jud; 1927   oras varchar2(50):=&amp;oras; 1928   stra varchar2(50) := &amp;str; 1929   numb number:=&amp;nmb; 1930   bloc number:= &amp;blk; 1931   apt number:=&amp;apt; 1932   sc varchar(2):=&amp;sc; 1933   adr Adresa; 1934   NumeComplet varchar2(50):= &amp;nume; 1935   moficari varchar2(50) := &amp;mof;</pre> <p>Output AdresajNUME_JUDET...d.APARTAMENT.dSCARA ×</p> <pre>begin   adr:= Adresa(judet,oras,str,a,numb,bloc,apt,sc);   DBMS_OUTPUT.PUT_LINE(SchimbareDatePacient(NumeComplet,moficari,cod_jod,adr)); end;</pre> <p>[2024-01-10 01:02:44] completed in 21 ms<br/>Persoana vrea sa se mute in aceeasi casa</p>                               |
| <p>Parameters</p> <pre>#18 judet varchar2(50):= null #19 oras varchar2(50):= 'Arad' #20 stra varchar2(50) := 'Bulevardul Revoluției' #21 numb number:= 12 #22 bloc number:= 6 #23 apt number:= 3 #24 sc varchar(2):=_ 'B' #25 NumeComplet varchar: 'Popescu Ion' #26 moficari varchar2(50) 'Domiciliu' #27 cod_jod number := _ null</pre> <p>Execute Close</p>   | <pre>1930 ✓ declare 1931   judet varchar2(50):=&amp;jud; 1932   oras varchar2(50):=&amp;oras; 1933   stra varchar2(50) := &amp;str; 1934   numb number:=&amp;nmb; 1935   bloc number:= &amp;blk; 1936   apt number:=&amp;apt; 1937   sc varchar(2):=&amp;sc; 1938   adr Adresa; 1939   NumeComplet varchar2(50):= &amp;nume; 1940   moficari varchar2(50) := &amp;mof;</pre> <p>Output AdresajNAME_JUDET...d.APARTAMENT.dSCARA ×</p> <pre>begin   adr:= Adresa(judet,oras,str,a,numb,bloc,apt,sc);   DBMS_OUTPUT.PUT_LINE(SchimbareDatePacient(NumeComplet,moficari,cod_jod,adr)); end;</pre> <p>[2024-01-10 01:07:48] completed in 12 ms<br/>Nu a fost acordat un nume de județ corect, nu s-a schimbat domiciliul!</p> |
| <p>Parameters</p> <pre>#18 judet varchar2(50):= null #19 oras varchar2(50):= null #20 stra varchar2(50) := 'Bulevardul Revoluției' #21 numb number:= 12 #22 bloc number:= 6 #23 apt number:= 3 #24 sc varchar(2):=_ 'B' #25 NumeComplet varchar: 'Popescu Ion' #26 moficari varchar2(50) 'Domiciliu' #27 cod_jod number := _ null</pre> <p>Execute Close</p>     | <pre>1930 ✓ declare 1931   judet varchar2(50):=&amp;jud; 1932   oras varchar2(50):=&amp;oras; 1933   stra varchar2(50) := &amp;str; 1934   numb number:=&amp;nmb; 1935   bloc number:= &amp;blk; 1936   apt number:=&amp;apt; 1937   sc varchar(2):=&amp;sc; 1938   adr Adresa; 1939   NumeComplet varchar2(50):= &amp;nume; 1940   moficari varchar2(50) := &amp;mof;</pre> <p>Output AdresajNUME_JUDET...d.APARTAMENT.dSCARA ×</p> <pre>begin   adr:= Adresa(judet,oras,str,a,numb,bloc,apt,sc);   DBMS_OUTPUT.PUT_LINE(SchimbareDatePacient(NumeComplet,moficari,cod_jod,adr)); end;</pre> <p>[2024-01-10 01:08:32] completed in 12 ms<br/>Nu putem sa cautam/adaugam o adresa fara Oras/Stads/Numar</p>              |

**Execution 1:**

```

Parameters
#18 judet varchar2(50):= null
#19 oras varchar2(50):= null
#20 stra varchar2(50) := 'Bulevardul Revoluției'
#21 numb number:= 12
#22 bloc number:= 6
#23 apt number:= 3
#24 sc varchar(2):= 'B'
#25 NumeComplet varchar2 'Popescu Ion'
#26 moficari varchar2(50) 'Job'
#27 cod_jod number := null

declare
    judet varchar2(50):=&jud;
    oras varchar2(50):=&ora;
    stra varchar2(50) := &str;
    numb number:=&nmb;
    bloc number:= &blk;
    apt number:=&apt;
    sc varchar(2):=&sc;
    adr Adresa;
    NumeComplet varchar2(50):=&neme;
    moficari varchar2(50) := &mof;
    cod_jod number := &njd;
begin
    adr:= Adresa(judet, oras, stra, numb, bloc, apartament: apt, scane: sc);
    DBMS_OUTPUT.PUT_LINE( A: schimbareDatePacient(NumeComplet, moficari, cod_jod, vitoareaadresa: adr));
end;
/

```

Output

```

Adresa(JUDET..._DAPARTAMENT.DSCARA) ×
begin
adr:= Adresa(judet,oras,str,a,numb,bloc,apt,sc);
DBMS_OUTPUT.PUT_LINE(SchimbareDatePacient(NumeComplet,moficari,cod_jod,adr));
end;

[2024-01-10 01:10:12] completed in 13 ms
Nu ai specificat un job, dar vezi sa il schimbi!

```

**Execution 2:**

```

Parameters
#18 judet varchar2(50):= null
#19 oras varchar2(50):= null
#20 stra varchar2(50) := 'Bulevardul Revoluției'
#21 numb number:= 12
#22 bloc number:= 6
#23 apt number:= 3
#24 sc varchar(2):= 'B'
#25 NumeComplet varchar2 'Popescu Ion'
#26 moficari varchar2(50) 'Job'
#27 cod_jod number := 2

declare
    judet varchar2(50):=&jud;
    oras varchar2(50):=&ora;
    stra varchar2(50) := &str;
    numb number:=&nmb;
    bloc number:= &blk;
    apt number:=&apt;
    sc varchar(2):=&sc;
    adr Adresa;
    NumeComplet varchar2(50):=&neme;
    moficari varchar2(50) := &mof;
    cod_jod number := &njd;
begin
    adr:= Adresa(judet,oras,str,a,numb,bloc,apt,sc);
end;
/

```

Output

```

Adresa(JUDET..._DAPARTAMENT.DSCARA) ×
Observati: Simptome: nas infundat, stramut, dureri de gat
None boala: Atergic
Observati: Reactii exagerate la alergeni specifici
None boala: Parkinson
Observati: Tulburari motorice si tremururi
Am schimbat cu succes locul de munca al pacientului!

```

**Execution 3:**

```

Parameters
#18 judet varchar2(50):= 'Bacau'
#19 oras varchar2(50):= 'Bacau'
#20 stra varchar2(50) := 'Neamului'
#21 numb number:= 12
#22 bloc number:= 6
#23 apt number:= 3
#24 sc varchar(2):= 'B'
#25 NumeComplet varchar2 'Popescu Ion'
#26 moficari varchar2(50) 'Domiciliu'
#27 cod_jod number := null

declare
    judet varchar2(50):=&jud;
    oras varchar2(50):=&ora;
    stra varchar2(50) := &str;
    numb number:=&nmb;
    bloc number:= &blk;
    apt number:=&apt;
    sc varchar(2):=&sc;
    adr Adresa;
    NumeComplet varchar2(50):=&neme;
    moficari varchar2(50) := &mof;
    cod_jod number := &njd;
begin
    adr:= Adresa(judet,oras,str,a,numb,bloc,apt,sc);
end;
/

```

Output

```

ORAS ×
- Loratadină
- Desloratadină
- Fexofenadină
- Levocetirizină
=====
Noua adresa: Jud.IIfov, Mun.Domnesti, Str.Traian, nr.32

```

**Table PACIENTI**

| ID_PACIENT | ID_DOMICILIU | ID_JOB | NUME           | PRENUME | DATA_NASTERE |
|------------|--------------|--------|----------------|---------|--------------|
| 1          | 3            | 2      | Popescu        | Ion     | 1990-01-15   |
| 2          | 26           | 3      | Ionescu        | Ana     | 1985-05-20   |
| 3          | 47           | 2      | Vasilescu      | Maria   | 1992-08-10   |
| 4          | 22           | 18     | Dumitrescu     | George  | 1988-03-03   |
| 5          | 29           | 14     | Radulescu      | Elena   | 1984-07-14   |
| 6          | 25           | 12     | Constantinescu | Mihai   | 1980-12-05   |

**Table PACIENTI**

| ID_PACIENT | ID_DOMICILIU | ID_JOB | NUME           | PRENUME | DATA_NASTERE |
|------------|--------------|--------|----------------|---------|--------------|
| 19         | 1            | 1      | Popescu        | Ion     | 1990-01-15   |
| 19         | 2            | 2      | Vasilescu      | Maria   | 1992-08-10   |
| 19         | 3            | 3      | Radulescu      | Elena   | 1984-07-14   |
| 19         | 4            | 4      | Constantinescu | Mihai   | 1980-12-05   |
| 53         | 53           | 53     | Popescu        | Ion     | 1990-01-15   |
| 26         | 26           | 26     | Vasilescu      | Maria   | 1992-08-10   |
| 47         | 47           | 47     | Radulescu      | Elena   | 1984-07-14   |
| 22         | 22           | 22     | Constantinescu | Mihai   | 1980-12-05   |

## Exercitiul 9

Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip procedură care să utilizeze într-o singură comandă SQL 5 dintre tabelele definite. Tratați toate exceptiile care pot apărea, incluzând exceptiile

**NO\_DATA\_FOUND** și **TOO\_MANY\_ROWS**. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

**Conducerea spitalelor dorește sa vada cei mai activi medici din fiecare spital(activitatea medicului este determinată de numărul de pacienți pe care îi are), -- dacă există 2 medici care au aceeași activitate bonusul o să fie redus la jumătate(spre exemplu dacă sunt 2 medici cu cea mai mare activitate pe țara acestea o sa primeasca 0.025 per pacient).**

Răsplata o să fie în funcție de următoarele:

- Cel mai activ medic din țara pe luna respectivă o să primeasca 0.05 per pacient ajutat
- Cel mai activ medic din județ o să primeasca 0.03 per pacient ajutat
- Cel mai activ medic dintr-un spital o să primeasca 0.02 per pacient ajutat
- Dacă un medic a primit un bonus acesta nu mai poate primii alt bonus mai mic decat cel primit

Pentru rezolvarea acestui exercițiu am realizat următoarele:

Un pachet care contine un record si un nested-table de acel record

```
create or replace package MediciActiviP is
    type rec is record (
        codmedic number,
        numecomplet varchar(60),
        specializare SPECIALIZARI.NUME_SPECIALIZARE%type,
        spitalul SPITALE.NUME_SPITAL%type,
        judetul JUDETE.NUME_JUDET%type,
        numeoras ORASE.NUME_ORAS%type, totalpacienti number);
    type tabelaMedici is table of rec;
end MediciActiviP;
```

Procedura:

```
create or replace procedure MediciActivi(anul date)  is
    --- exceptii
    no_active_judet exception;
    no_active_spital exception;
    ---variabile
    Tabela tabelaMedici;
    medic rec;
    multi boolean:=false;
    salariuafis number;
    no_marire number;
begin
    SELECT M2.ID_MEDIC,M2.NUME || '
    '||M2.PRENUME,s5.NUME_SPECIALIZARE,s5.NUME_SPITAL,o.NUME_ORAS,j.NUME_JUDET, COUNT(ip.ID_MEDIC)
    bulk collect into Tabela
    FROM MEDICI m2
```

```

left JOIN (select * from ISTORIC_PACIENTI join RETETE r on ISTORIC_PACIENTI.ID_RETETA =
r.ID_RETETA where DATA_ELIBERARII between anul and add_months(anul,12) ) ip ON M2.ID_MEDIC
= ip.ID_MEDIC
join SPITALE S5 on S5.ID_SPITAL = M2.ID_SPITAL
join Strazi st on st.ID_STRADA = s5.ID_STRADA
join orase o on o.ID_ORAS = st.ID_ORAS
join judete j on j.ID_JUDET = o.ID_JUDET
join SPECIALIZARI S5 on M2.ID_SPECIALIZARE = S5.ID_SPECIALIZARE
GROUP BY M2.NUME, M2.PRENUME,m2.ID_MEDIC,
s5.NUME_SPECIALIZARE,s5.NUME_SPITAL,o.NUME_ORAS,j.NUME_JUDET
order by m2.ID_MEDIC
;

select max(totalpacienti) into no_marire from Table(Tabela);
if no_marire = 0 then
  raise no_data_found;
end if;
--- marim salariul celui mai bun din tara
begin
  select * into medic from Table(Tabela) where totalpacienti = (select max(totalpacienti)
from Table(Tabela));
  DBMS_OUTPUT.PUT_LINE('Cel mai activ medic din Tara este Dr. '|| medic.nume complet);
  DBMS_OUTPUT.PUT_LINE('Lucreaza la spitalul: ' || medic.spitalul);
  update MEDICI set SALARIU = round(salariu +( salariu *(medic.totalpacienti *0.05)))
  where ID_MEDIC = medic.codmedic;
  select SALARIU into salariuafis from MEDICI where ID_MEDIC = medic.codmedic;
  Tabela(medic.codmedic).totalpacienti:=-1;
  DBMS_OUTPUT.PUT_LINE('Noul sau salariu este ' || salariuafis);

exception
when too_many_rows then
  multi := true;
end;
--- in cazul in care sunt mai multi pe tara atunci coeficientul
if multi = true then
  DBMS_OUTPUT.PUT_LINE('Mai multi medici se afla la egalitate pentru bonusul pe tara!');
  DBMS_OUTPUT.PUT_LINE('Bonusul a fost redus la 0.05 per 2 pacienti');
  DBMS_OUTPUT.PUT_LINE('Cei mai buni medici din tara sunt:');
  for med in (select * from Table(Tabela) where totalpacienti = (select
max(totalpacienti) from Table(Tabela))) loop
    DBMS_OUTPUT.PUT_LINE('Dr. ' || med.nume complet);
    update MEDICI set SALARIU = round(salariu +( salariu *(med.totalpacienti
*(0.05/2))))
    where ID_MEDIC = med.codmedic;
    select SALARIU into salariuafis from MEDICI where ID_MEDIC = med.codmedic;
    Tabela(medic.codmedic).totalpacienti:=-1;
    DBMS_OUTPUT.PUT_LINE('Noul sau salariu este ' || salariuafis);
    end loop;
  multi:=false;
end if;
for judet in (select * from judete) loop
begin
  select max(totalpacienti) into no_marire from Table(Tabela) where judetul =
judet.NUME_JUDET;
  if no_marire = 0 then
    raise no_active_judet;
  end if;

```

```

select * into medic from Table(Tabela) where totalpacienti = (select max(totalpacienti)
from Table(Tabela) where judetul = judet.NUME_JUDET) and judetul = judet.NUME_JUDET;
DBMS_OUTPUT.PUT_LINE('Cel mai activ medic din '||judet.NUME_JUDET||' este Dr. '||medic.numecomplet);
DBMS_OUTPUT.PUT_LINE('Lucreaza la spitalul: ' || medic.spitalul);
update MEDICI set SALARIU = round(salariu +( salariu *(medic.totalpacienti *0.03)))
where ID_MEDIC = medic.codmedic;
select SALARIU into salariuafis from MEDICI where ID_MEDIC = medic.codmedic;
Tabela(medic.codmedic).totalpacienti:=-1;
DBMS_OUTPUT.PUT_LINE('Noul sau salariu este ' || salariuafis);
exception
when no_active_judet then
DBMS_OUTPUT.PUT_LINE('Nu exista medici activi in judetul '|| judet.NUME_JUDET);
multi:=false;
when no_data_found then
DBMS_OUTPUT.PUT_LINE('Nu exista medici in Judetul '|| judet.NUME_JUDET );
when too_many_rows then
multi := true;
end;
if multi = true then
DBMS_OUTPUT.PUT_LINE('Mai multi medici se afla la egalitate pentru bonusul din
judetul '||judet.NUME_JUDET||'!');
DBMS_OUTPUT.PUT_LINE('Bonusul a fost redus la 0.03 per 2 pacienti');
DBMS_OUTPUT.PUT_LINE('Cei mai buni medici din '||judet.NUME_JUDET||' sunt:');
for med in (select * into medic from Table(Tabela)
where totalpacienti = (select max(totalpacienti)
from Table(Tabela) where judetul = judet.NUME_JUDET) and judetul = judet.NUME_JUDET) loop
DBMS_OUTPUT.PUT_LINE('Dr. ' || med.numecomplet);
update MEDICI set SALARIU = round(salariu +( salariu
*(med.totalpacienti *(0.03)/2)))
where ID_MEDIC = med.codmedic;
select SALARIU into salariuafis from MEDICI where ID_MEDIC =
med.codmedic;
Tabela(med.codmedic).totalpacienti:=-1;
DBMS_OUTPUT.PUT_LINE('Noul sau salariu este ' || salariuafis);
end loop;
multi:=false;
end if;
end loop;

--- cei mai bun din spitale
for spital in (select * from SPITALE) loop
begin
select max(totalpacienti) into no_marire from Table(Tabela) where spitalul
= spital.NUME_SPITAL;
if no_marire = 0 then
raise no_active_spital;
end if;
select * into medic from Table(Tabela) where totalpacienti = (select max(totalpacienti)
from Table(Tabela) where spitalul = spital.NUME_SPITAL) and spitalul = spital.NUME_SPITAL;
DBMS_OUTPUT.PUT_LINE('Cel mai activ medic din '||spital.NUME_SPITAL||' este Dr. '||medic.numecomplet);
DBMS_OUTPUT.PUT_LINE('Avand specializarea: ' || medic.specializare);
update MEDICI set SALARIU = round(salariu +( salariu *(medic.totalpacienti *0.03)))
where ID_MEDIC = medic.codmedic;
select SALARIU into salariuafis from MEDICI where ID_MEDIC = medic.codmedic;
Tabela(medic.codmedic).totalpacienti:=-1;

```

```

DBMS_OUTPUT.PUT_LINE('Noul sau salariu este ' || salariuafis);
exception
when no_active_spital then
    multi:=false;
DBMS_OUTPUT.PUT_LINE('Nu exista medici activi in spitalul ' || medic.spitalul);
when no_data_found then
DBMS_OUTPUT.PUT_LINE('Nu exista medici in Spitalul '|| medic.spitalul );
when too_many_rows then
    multi := true;
end;
if multi = true then
    DBMS_OUTPUT.PUT_LINE('Mai multi medici se afla la egalitate pentru bonusul
din'||spital.NUME_SPITAL||'!');
    DBMS_OUTPUT.PUT_LINE('Bonusul a fost redus la 0.02 per 2 pacienti');
    DBMS_OUTPUT.PUT_LINE('Cei mai buni medici din'||spital.NUME_SPITAL||'
sunt:');
for med in (select * into medic from Table(Tabela)
                           where totalpacienti = (select max(totalpacienti)
from Table(Tabela) where spitalul = spital.NUME_SPITAL) and spitalul = spital.NUME_SPITAL)
loop
    DBMS_OUTPUT.PUT_LINE('Dr. ' || med.numecomplet);
    update MEDICI set SALARIU = round(salariu +( salariu
*(med.totalpacienti *(0.02)/2)))
        where ID_MEDIC = med.codmedic;
    select SALARIU into salariuafis from MEDICI where ID_MEDIC =
med.codmedic;
    Tabela(med.codmedic).totalpacienti:=-1;
    DBMS_OUTPUT.PUT_LINE('Noul sau salariu este ' || salariuafis);
end loop;
end if;
end loop;

exception
when No_Data_Found then
DBMS_OUTPUT.PUT_LINE('Nu exista medici activi in acea perioada!');
end MediciActivi;

```

```

2009 ✓ create or replace procedure MediciActivi(anul date) is
2010   --- exceptii
2011   no_active_judet exception;
2012   no_active_spital exception;
2013   ---variabile
2014   Tabela MediciActiviP.tabelaMedici;
2015   medic MediciActiviP.rec;
2016   multi boolean:=false;
⑧ MediciActivi()

```

Output Result 22 ×

```

0 ms
[2024-01-11 17:05:29] completed in 21 ms

```

## Rezultatele apelarii procedurii

| Date de intrare   | Output   |
|---|--|
| <p>Parameters</p> <pre>#3 _, 'YYYY' '2024'</pre> <p><b>Execute</b> <b>Close</b></p> | <pre>155 ✓ declare 156     anul_date := to_date(&amp;n, 'YYYY'); 157 158 begin 159     MedicActiv(anul: anul); 160 end; 161 162 -- O procedura stocata care sa 163 164 165 ---LMD la nivel de comanda 166 --- acestei triggeri se folosesc cand nu conteaza informatiile 167 168 <u>scind la nivel de linie</u></pre> <p><b>Output</b> <b>Result 22</b></p> <pre>anul_date := to_date('2024', 'YYYY'); begin     MedicActiv(anul); end;</pre> <p>[2024-01-11 17:19:39] completed in 14 ms<br/>Nu exista medici activi in acea perioada!</p>  |
| <p>Parameters</p> <pre>#3 _, 'YYYY' '2022'</pre> <p><b>Execute</b> <b>Close</b></p> | <pre>2194 commit; 2195 ✓ declare 2196     anul_date := to_date(&amp;n, 'YYYY'); 2197 2198 begin 2199     MedicActiv(anul: anul); 2200 end; 2201 2202 -- O procedura stocata care sa 2203 2204 2205 ---LMD la nivel de comanda 2206 --- acestei triggeri se folosesc cand nu conteaza informatiile 2207 2208 <u>scind la nivel de linie</u></pre> <p><b>Output</b> <b>Result 22</b></p> <p>Nu exista medici in Judetul Vaslui<br/>Nu exista medici in Judetul Vrancea<br/>Cel mai activ medic din Spitalul Municipal Targu Mures este Dr. Dima Ionel<br/>Avand specializarea: Pneumologie<br/>Noul sau salariu este 3010<br/>Cel mai activ medic din Spitalul Judetean Piatra Neamt este Dr. Oprea Elena<br/>Avand specializarea: Nefrologie<br/>Noul sau salariu este 3401<br/>Cel mai activ medic din Spitalul Municipal Slatina este Dr. Stanciu Paul<br/>Avand specializarea: Medicina de Uroenta</p> |

**Output MEDICI**

| ID_MEDIC | ID_SPECIALIZARE | ID_SPITAL | NUME | PRENUME | SALARIU | DATA_ANGAJARI   |
|----------|-----------------|-----------|------|---------|---------|-----------------|
| 1        | 48              | 19        | 9    | Dima    | Monica  | 4860 2011-02-19 |
| 2        | 70              | 18        | 1    | Dima    | Ionel   | 2840 2015-06-14 |

**Output MEDICI**

| ID_MEDIC | ID_SPECIALIZARE | ID_SPITAL | NUME | PRENUME | SALARIU | DATA_ANGAJARI   |
|----------|-----------------|-----------|------|---------|---------|-----------------|
| 1        | 48              | 19        | 9    | Dima    | Monica  | 4937 2011-02-19 |
| 2        | 70              | 18        | 1    | Dima    | Ionel   | 3010 2015-06-14 |

## Exercitiul 10

Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.

**Pe durata perioadei de angajare, este acceptat sa se poată efectua inserarea unui medic în tabela specifică. Perioada de angajare o să fie deschisă/ închisă prin intermediul unei proceduri.**

Pentru rezolvarea acestui exercițiu am realizat următoarele:

```
create or replace package Perioada_angajare is
    data_start date;
    data_finish date;
    procedure SchimbaAngajarea(startd varchar2 ,finish varchar2 );
end;
create or replace package body Perioada_angajare is
    procedure SchimbaAngajarea(startd varchar2 ,finish varchar2 )
    is
    begin
        data_start := to_date(startd,'YYYY-MM-DD');
        data_finish := to_date(finish,'YYYY-MM-DD');
    end SchimbaAngajarea;
end;
declare
    stringstar varchar2(11) := &start;
    stringend varchar2(11) :=&end;
begin
    Perioada_angajare.SchimbaAngajarea(stringstar,stringend);
end;
```

Trigger-ul:

```
create or replace trigger PERIOADAANG
before insert on MEDICI
begin
    if sysdate not between Perioada_angajare.data_start and Perioada_angajare.data_finish
then
        RAISE_APPLICATION_ERROR(-20005,'Nu poti angaja medici pentru ca nu suntem in
perioada de angajare');
    end if;
end;
```

| Date de intrare   | Output   |
|---|--|
| <div style="border: 1px solid #ccc; padding: 10px; width: fit-content;"> <p>Parameters</p> <pre>#4 stringsta '2023/01/01' #5 stringend '2023/02/02'</pre> <p><b>Execute</b> <b>Close</b></p> </div> | <div style="display: flex; align-items: flex-start;"> <div style="flex: 1; margin-right: 20px;"> <pre>2318  END; 2319  INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU, DATA_ANGAJARI) 2320   VALUES (102,2,1,'Marian', 'Ion', 3000, sysdate);</pre> <p>(2023/01/01) ORA-20005: Nu poti angaja medici pentru ca nu suntem in perioada de angajare ORA-06512: at "UTILIZATOR.PERIOADAANG", line 3 ORA-04088: error during execution of trigger 'UTILIZATOR.PERIOADAANG' Position: 12</p> </div> <div style="flex: 1;"> </div> </div> |

The screenshot shows the Oracle SQL Developer interface. On the left, there's a 'Parameters' window with two variables defined: #4 stringdate '2023/01/01' and #5 stringdate '2024/02/02'. Below it are 'Execute' and 'Close' buttons. The main area contains a code editor with the following PL/SQL code:

```

2319  INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU, DATA_ANGAJARI)
2320  VALUES (102,2,1, 'Mariam', 'Ion', 3000, sysdate);
2321  select * from Medici;
2322  create or replace trigger PERIODANG
2323    before insert on MEDICI
2324    begin
2325      if sysdate not between Perioada_angajare.data_start and Perioada_angajare.data_finish then
2326        RAISE_APPLICATION_ERROR(-20005,'Nu poti angaja medici pentru ca nu suntem in perioada de angajare');
2327      end if;
2328    end;

```

Below the code editor is an 'Output' window titled 'MEDICI' showing the result of the 'Execute' command:

```

[2024-01-11 18:43:49] completed in 12 ms
UTILIZATOR> INSERT INTO MEDICI (ID_MEDIC, ID_SPECIALIZARE, ID_SPITAL, NUME, PRENUME, SALARIU, DATA_ANGAJARI)
VALUES (102,2,1, 'Mariam', 'Ion', 3000, sysdate)
[2024-01-11 18:43:56] 1 row affected in 11 ms

```

## Exercitiul 11

Definiți un trigger de tip LMD la nivel de linie. Declanșați trigger-ul

**Conducerea Spitalelor a decis ca în anul 2024 nu mai doresc sa angajeze mai mult de (*numărul de paturi din acel spital / 50*) medici la un spital.**

**Dacă un medic dorește să se transfere acesta trebuie să atunci acesta trebuie să aibă un pacient care locuiește în acel județ și trebuie să respecte locurile disponibile(*același calcul ca mai sus*)**

```

CREATE OR REPLACE TRIGGER angajaremedic
for Insert or Update ON MEDICI compound trigger
type spitalspecializare is record (spital Medici.ID_SPITAL%type, specializare
Medici.ID_SPECIALIZARE%type);
type tabelimpMedici is table of spitalspecializare index by pls_integer;
Medi tabelimpMedici;
before statement is
auxiliar spitalspecializare;
begin
for medic in (select * from Medici) loop
auxiliar.specializare := medic.ID_SPECIALIZARE;
auxiliar.spital := medic.ID_SPITAL;
Medi(medic.ID_MEDIC) := auxiliar;
end loop;

end before statement;
before each row is
counter number:=0;
numarpaturi spital.NUMAR_PATURI%type;
numespital SPITALE.NUME_SPITAL%type;
stradaspital spital.E.ID_SPITAL%type;
judetspital orasE.ID_JUDET%type;
judete number_list;
begin
select NUME_SPITAL,NUMAR_PATURI, ID_STRADA into numespital,numarpaturi,stradaspital from
SPITALE where ID_SPITAL = :new.ID_SPITAL;
if inserting then
for i in 1..Medi.Count loop
if Medi(i).specializare = :new.ID_SPECIALIZARE and Medi(i).spital = :new.ID_SPITAL then
counter := counter+1;
end if;
end loop;
end if;
end;

```

```

    end if;
    if counter >= ceil(numarpaturi/50)then -- luam cu ceil limita superioara adica daca
calculul da 2,5 lasam sa fie 3 locuri
        raise_application_error(-20010,'Sunt prea multe locuri ocupate in spitalul ' ||
numespital);
    end if;
end loop;
end if;-- in cazul in care un medic se transfera
if updating then
if :new.ID_SPITAL <>:old.ID_SPITAL then
    SELECT DISTINCT o.ID_JUDET bulk collect into judete
        FROM ISTORIC_PACIENTI i
    JOIN pacienti p ON p.ID_PACIENT = i.ID_PACIENT
    JOIN domiciliI d ON d.ID_DOMICILIU = p.ID_DOMICILIU
    JOIN STRAZI s ON s.ID_STRADA = d.ID_STRADA
    LEFT JOIN orasE o ON s.ID_ORAS = o.ID_ORAS
WHERE i.id_medic = :old.ID_MEDIC;
if sql%notfound then
    raise_application_error(-20012,'Medicul nu are pacienti la activ!!');
end if;
select o.id_judet into judetspital from strazi
join orasE o on strazi.ID_ORAS =o.ID_ORAS where strazi.ID_STRADA = stradaspital;
if judete.EXISTS(judetspital) then
    if :new.ID_SPECIALIZARE is not null then
        for i in 1..Medi.Count loop
if Medi(i).specializare = :new.ID_SPECIALIZARE and Medi(i).spital = :new.ID_SPITAL then
            counter := counter+1;
        end if;
        if counter >= ceil(numarpaturi/50)then -- luam cu ceil limita superioara adica daca
calculul da 2,5 lasam sa fie 3 locuri
            raise_application_error(-20010,'Sunt prea multe locuri ocupate in spitalul ' ||
numespital);
        end if;
        end loop;
    else
        for i in 1..Medi.Count loop
if Medi(i).specializare = :old.ID_SPECIALIZARE and Medi(i).spital = :new.ID_SPITAL then
            counter := counter+1;
        end if;
        if counter >= ceil(numarpaturi/50)then -- luam cu ceil limita superioara adica daca
calculul da 2,5 lasam sa fie 3 locuri
            raise_application_error(-20010,'Sunt prea multe locuri ocupate in spitalul ' ||
numespital);
        end if;
        end loop;
    end if;
    else
        raise_application_error(-20011,'Medicul nu are pacienti in judetul in care vrea sa se
muta');
    end if;
    end if;
end if;
end before each row;
end;

```

Insertul în cazul în care medicul nou angajat nu are loc

```

2118  insert into Medic(id_medic, id_specializare, id_spital, nume, prenume, salariu, data_angajari) select 102,21,5,'Marian','Segevescu',5000,sysdate from DUAL;
2119  declare
2120
2121 begin
2122     select count(*) from MEDICI where ID_SPITAL =5 and ID_SPECIALIZARE=21;
2123     select ceil(NUMAR_PATURI/50) from SPITAL where ID_SPITAL=5;

```

[72000][20010]  
ORA-20010: Sunt prea multe locuri ocupate în spitalul Spitalul Județean Satu Mare  
ORA-06512: at "UTILIZATOR.ANGAJAREMEDIC", line 30  
ORA-04088: error during execution of trigger 'UTILIZATOR.ANGAJAREMEDIC'  
Position: 12

```

919  select count(*) from MEDICI where ID_SPITAL =5 and ID_SPECIALIZARE=21;
920  setvar < from SPITAL;
921  insert into Medic(id_medic, id_specializare, id_spital, nume, prenume, salariu, data_angajari) values (102,21,5,'Marian','Segevescu',5000,sysdate);
922 exception
923 when others then
924   DBMS_OUTPUT.PUT_LINE( A: sqlcode );
925   DBMS_OUTPUT.PUT_LINE( A: sqlerrm );
926 end;
927
928 select * from RETETE;
929 --- 100
930 create table userlogin(
931  ID NUMBER PRIMARY KEY ,
932  Username VARCHAR2(10),
933  Name_Password VARCHAR2(100),
934  Datelogin timestamp(3)
935 );

```

Output count(\*)NUMBER  
1 COUNT(\*) :  
1 3

Output celNUMAR\_PATURISI/50NUMBER  
1 celNUMAR\_PATURISI/50NUMBER :  
1 1

### Update in momentul cand Medicul nu are pacienti:

```

1922  update Medic set ID_SPITAL = 6 where ID_MEDIC = 101;
1923  insert into Medic(id_medic, id_specializare, id_spital, nume, prenume, salariu, data_angajari) values (102,21,5,'Marian','Segevescu',5000,sysdate);
1924 exception
1925 when others then
1926   DBMS_OUTPUT.PUT_LINE( A: sqlcode );

```

[72000][20012]  
ORA-20012: Medicul nu are pacienti la activ!  
ORA-06512: at "UTILIZATOR.ANGAJAREMEDIC", line 45  
ORA-04088: error during execution of trigger 'UTILIZATOR.ANGAJAREMEDIC'  
Position: 9

Output celNUMAR\_PATURISI/50NUMBER  
1 celNUMAR\_PATURISI/50NUMBER :  
1 0

Output UTILIZATOR> update Medic set ID\_SPITAL = 6 where ID\_MEDIC = 101  
[2024-01-07 20:34:06] [72000][20012]  
[2024-01-07 20:34:06] ORA-20012: Medicul nu are pacienti la activ!  
[2024-01-07 20:34:06] ORA-06512: at "UTILIZATOR.ANGAJAREMEDIC", line 45  
[2024-01-07 20:34:06] ORA-04088: error during execution of trigger 'UTILIZATOR.ANGAJAREMEDIC'  
[2024-01-07 20:34:06] Position: 7

```

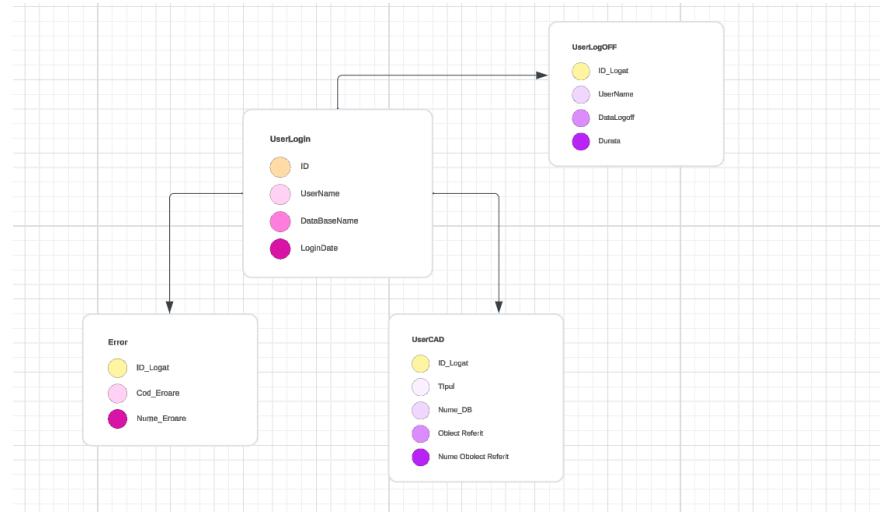
1922  update Medic set ID_SPITAL = 6 where ID_MEDIC = 2;
1923  insert into Medic(id_medic, id_specializare, id_spital, nume, prenume, salariu, data_angajari) values (102,21,5,'Marian','Segevescu',5000,sysdate);
1924 exception
1925 when others then
1926   DBMS_OUTPUT.PUT_LINE( A: sqlcode );
1927   DBMS_OUTPUT.PUT_LINE( A: sqlerrm );
1928 end;
1929
1930 select * from RETETE;
```

[72000][20011]  
ORA-20011: Medicul nu are pacienti in judetul in care vrea sa se mute  
ORA-06512: at "UTILIZATOR.ANGAJAREMEDIC", line 71  
ORA-04088: error during execution of trigger 'UTILIZATOR.ANGAJAREMEDIC'  
Position: 7

Output celNUMAR\_PATURISI/50NUMBER  
1 celNUMAR\_PATURISI/50NUMBER :  
1 0

## Exercitiul 12

Definiți un trigger de tip LDD. Declanșați trigger-ul.



Am realizat 4 tabele și 4 triggeri care o să insereze date în ele în momentul în care

Creem tabelele

| Poza cu tabele  | Comanda de Create table   |
|---|---|
| <pre> 1988 ✓  select * from UserLogin; 1989   select * from UserLogOff; 1990   select * from ErrorLog; 1991   select * from UserCAD; 1992 1993   delete UserLogOff; 1994   delete UserCAD; 1995   delete ErrorLog; 1996   delete UserLogin; 1997 1998 select to_char(sys_context('userenv','sessionid')) from Dual; 1999 </pre> | <pre> create table UserLogin(     ID number primary key ,     UserName varchar2(30),     Nume_BazaDate varchar2(50),     DataLogare timestamp(3) );  alter table UserLogin add sesiune varchar2(30); </pre> |

|   |  |
|---|--|
| <pre> 1909 ✓  select * from UserLogOff; 1910   select * from ErrorLog; 1911   select * from UserCAD; 1912 1913 1914 delete UserLogOff; 1915 delete UserCAD; 1916 delete ErrorLog; 1917 delete UserLogin; 1918 1919 select to_char(sys_context('userenv','sessionid')) from Dual; </pre> | <pre> create table UserLogOff( ID_Logare number, UserName varchar2(30), Nume_BazaDate varchar2(50), DataDeLogare timestamp(3), foreign key (ID_Logare) references UserLogin(ID) ); </pre>                                    |
| <pre> 1910 ✓  select * from ErrorLog; 1911   select * from UserCAD; 1912 1913 1914 delete UserLogOff; 1915 delete UserCAD; 1916 delete ErrorLog; 1917 delete UserLogin; 1918 1919 select to_char(sys_context('userenv','sessionid')) from Dual; </pre>                                  | <pre> create table ErrorLog( ID_Logare number, Cod_Eroare varchar2(30), Eroare varchar2(2000), Data timestamp(3), foreign key (ID_Logare) references UserLogin(ID) ); </pre>   |
| <pre> 1911 ✓  select * from UserCAD; 1912 1913 delete UserLogOff; 1914 delete UserCAD; 1915 delete ErrorLog; 1916 delete UserLogin; 1917 1918 select to_char(sys_context('userenv','sessionid')) from Dual; </pre>  | <pre> create table UserCAD( ID_Logare number, Tipul varchar2(20), Nume_BD varchar2(50), Obiect_Referit varchar2(30), Nume_Obiect varchar2(30), data timestamp(3), foreign key (ID_Logare) references UserLogin(ID) ); </pre> |

Un user se conecteaza la baza de date(trigger LDD care insereaza date in tabela *UserLogin*)

The screenshot shows the Oracle SQL Developer interface. On the left, the Object Navigator displays the schema structure under the database connection '@localhost'. A trigger named 'LoguriErrori' is selected. The code editor on the right contains the DDL for creating the trigger:

```

1896    insert into ErrorLog(id_logare, cod_eroare, eroare, data) values (errorr,cod,text,sysimestamp);
1897
1898    end inserterr;
1899
1900    create or replace trigger LoguriErrori
1901        after servererror on schema
1902            call inserterr;
1903
1904    SELECT * FROM ALL_ERRORS WHERE NAME = 'CADSCHEMA' AND TYPE = 'TRIGGER' AND OWNER = 'UTILIZATOR';
1905
1906    select * from UserCAD;
1907
1908    select * from UserLogin;
1909    select * from UserLogOff;
1910    select * from ErrorLog;
1911    select * from UserCAD;
1912
1913    delete UserLogOff;
1914
1915    DIP

```

Below the code editor, the Services pane shows the database connection and session details. The Output pane displays the results of the trigger execution, showing four rows inserted into the 'USERLOGIN' table:

| ID | USERNAME     | NUME_BAZADATE | DATALOGARE              | SESIUNE |
|----|--------------|---------------|-------------------------|---------|
| 1  | 4 UTILIZATOR | XE            | 2024-01-07 00:54:47,343 | 170622  |
| 2  | 3 UTILIZATOR | XE            | 2024-01-07 00:37:23,398 | 170620  |
| 3  | 1 UTILIZATOR | XE            | 2024-01-07 00:30:45,745 | 170608  |
| 4  | 2 UTILIZATOR | XE            | 2024-01-07 00:31:08,044 | 170610  |

```

create or replace procedure insertinto
is
    e_logat number;
    ultimalogare UserLogin.ID%type;
    prima_logare exception;
begin
begin
    select ID into ultimalogare from (select ID from UserLogin where UserName =
sys.LOGIN_USER() order by ID desc ) where ROWNUM = 1;

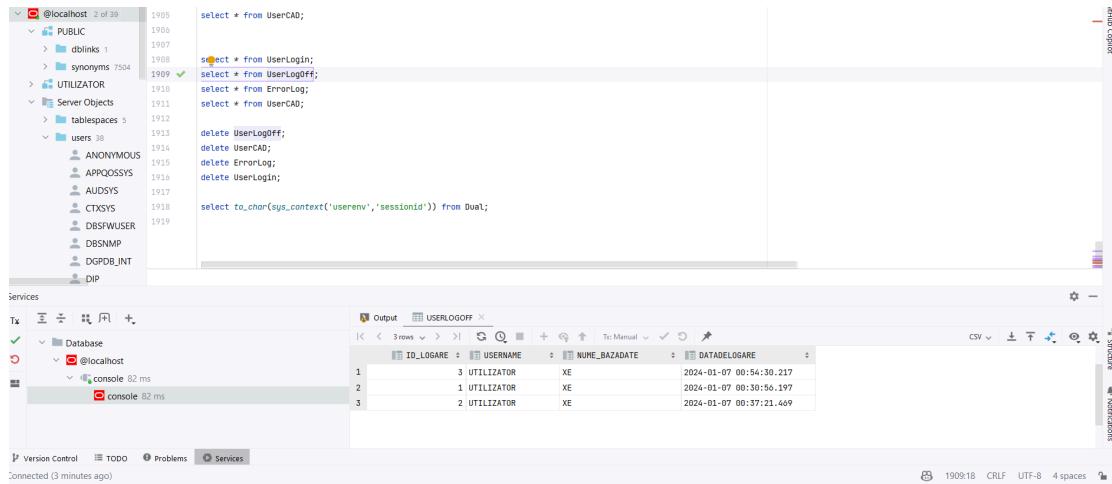
    select COUNT(*) into e_logat from UserLogOff where UserName = sys.LOGIN_USER() and
ID_Logare = ultimalogare;
    exception
    when no_data_found then
    ultimalogare:=-1;
end;

if e_logat>0 or ultimalogare =-1 then
    insert into UserLogin(UserLogin.ID, username, nume_bazadate, datalogare,sesiune)
values
(seq_login.nextval,sys.LOGIN_USER(),sys.DATABASE_NAME(),sys.timestamp,sys_context('userenv','sessionid'));
    end if;
End;

create or replace trigger Login
after logon on DATABASE
call insertinto;

```

Un user se deconecteaza de la baza de date(trigger LDD care insereaza date in tabela *UserLogOff*)



The screenshot shows the Oracle SQL Developer interface. On the left, the object browser displays various database objects like PUBLIC, dblinks, synonyms, UTILIZATOR, Server Objects, tablespaces, and users. In the center, a SQL editor window contains the following code:

```

1905 select * from UserCAD;
1906
1907
1908 select * from UserLogin;
1909 select * from UserLogOff;
1910 select * from ErrorLog;
1911 select * from UserCAD;
1912
1913 delete UserLogOff;
1914 delete UserCAD;
1915 delete ErrorLog;
1916 delete UserLogin;
1917
1918 select to_char(sys_context('userenv','sessionid')) from Dual;
1919

```

Below the editor is a results grid titled "USERLOGOFF" with the following data:

| ID_LOGARE | USERNAME     | NUME_BAZADATE | DATADELOGARE            |
|-----------|--------------|---------------|-------------------------|
| 1         | 3 UTILIZATOR | XE            | 2024-01-07 00:54:30.217 |
| 2         | 1 UTILIZATOR | XE            | 2024-01-07 00:30:56.197 |
| 3         | 2 UTILIZATOR | XE            | 2024-01-07 00:37:21.469 |

```

create or replace procedure insertintologoff
is
logoff UserLogin%rowtype;
inchis number;
begin

    select * into logoff from (select * from UserLogin where UserName =
sys.LOGIN_USER() order by ID desc) where rownum=1;
    select count(*) into inchis from UserLogOff where ID_Logare = logoff.ID;

    if logoff.sesiune != to_char(sys_context('userenv','sessionid')) and inchis=0 then
        insert into UserLogOff(id_logare, username, nume_bazadate, datadelogare) values
(logoff.ID,sys.LOGIN_USER(),sys.DATABASE_NAME(),systimestamp);
        end if;
exception
    when no_data_found then
        raise_application_error(-2100,'A aparut o eroare la log off!!');
end;
create or replace trigger Logoff
before logoff on Database
call insertintologoff;

```

Un user produce o eroare in baza de date((trigger LDD care insereaza date in tabela *ErrorLog*)

The screenshot shows the Oracle SQL Developer interface. At the top, there is a code editor window containing PL/SQL code. A tooltip is visible over the line 'increment by 1;'. Below the code editor is a message box with the error message: '[42000][95] ORA-00955: name is already used by an existing object' at position 17. The bottom part of the screenshot shows the 'Output' tab with a table named 'ERRORLOG' containing three rows of data. The columns are 'ID\_LOGARE', 'COD\_EROARE', 'ERODARE', and 'DATA'. The first row has ID\_LOGARE 1, COD\_EROARE 1, ERODARE 'savepoint 'ORACLE\_SVPL\_41' never established in this session or is invalid', and DATA '2024-01-07 01:00:27.440'. The second row has ID\_LOGARE 2, COD\_EROARE 2, ERODARE 'Missing keyword', and DATA '2024-01-07 00:31:35.472'. The third row has ID\_LOGARE 3, COD\_EROARE 4, ERODARE 'name is already used by an existing object', and DATA '2024-01-07 01:00:02.159'.

| ID_LOGARE | COD_EROARE | ERODARE  | DATA                    |
|-----------|------------|--|-------------------------|
| 1         | 1          | savepoint 'ORACLE_SVPL_41' never established in this session or is invalid | 2024-01-07 01:00:27.440 |
| 2         | 2          | Missing keyword  | 2024-01-07 00:31:35.472 |
| 3         | 4          | name is already used by an existing object                                 | 2024-01-07 01:00:02.159 |

```

create or replace procedure inserterror
is
errorr UserLogin.ID%type;
erroare varchar2(4020);
cod varchar2(20);
text varchar2(2000);
begin
erroare := DBMS_UTILITY.FORMAT_ERROR_STACK;
cod := SUBSTR(erroare, INSTR(erroare, 'ORA-', 1), 9);
text := SUBSTR(erroare,11);

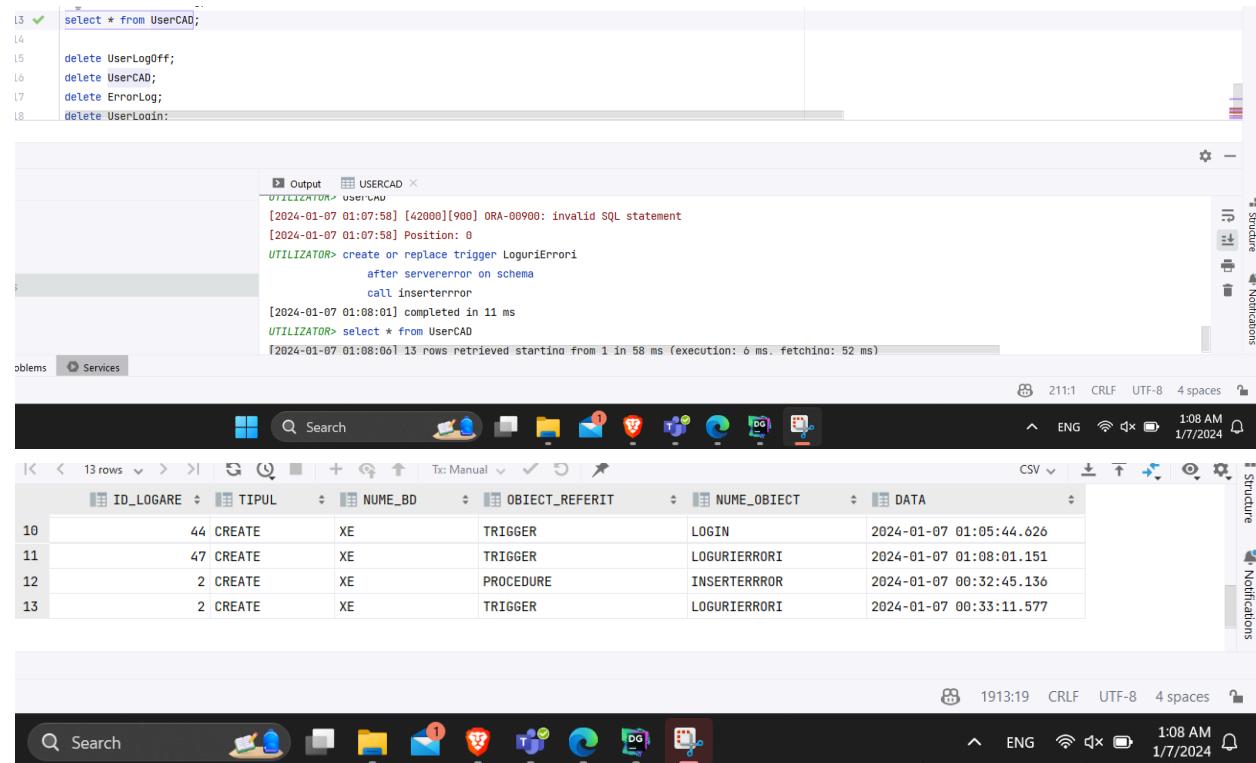
select ID into errorr from UserLogin where UserName = sys.LOGIN_USER() and rownum <=1
order by ID desc ;

insert into ErrorLog(id_logare, cod_eroare, eroare, data) values
(errorr,cod,text,sysimestamp);

end inserterr;
create or replace trigger LoguriErrori
after servererror on schema
call inserterr

```

Un user creaza/insereaza sau sterge ceva din baza de date((trigger LDD care insereaza date in tabela *UserCAD*)



The screenshot shows a SQL developer interface. At the top, there is a session history window with the following content:

```

L3 ✓ select * from UserCAD;
L4
L5 delete UserLogOff;
L6 delete UserCAD;
L7 delete ErrorLog;
L8 delete UserLogin;

```

Below the session history is an output window titled "USERCAD" showing the following log entries:

```

[2024-01-07 01:07:58] [42000][900] ORA-00900: invalid SQL statement
[2024-01-07 01:07:58] Position: 0
UTILIZATOR> create or replace trigger LoguriErrori
           after servererror on schema
           call inserterrerror
[2024-01-07 01:08:01] completed in 11 ms
UTILIZATOR> select * from UserCAD
[2024-01-07 01:08:06] 13 rows retrieved starting from 1 in 58 ms (execution: 6 ms. fetchina: 52 ms)

```

At the bottom of the interface is a table viewer showing data from the UserCAD table:

| ID_LOGARE | TIPUL | NUME_BD | OBIECT_REFERIT | NUME_OBJECT | DATA          |                         |
|-----------|-------|---------|----------------|-------------|---------------|-------------------------|
| 10        | 44    | CREATE  | XE             | TRIGGER     | LOGIN         | 2024-01-07 01:05:44.626 |
| 11        | 47    | CREATE  | XE             | TRIGGER     | LOGURIERRORI  | 2024-01-07 01:08:01.151 |
| 12        | 2     | CREATE  | XE             | PROCEDURE   | INSERTERRRROR | 2024-01-07 00:32:45.136 |
| 13        | 2     | CREATE  | XE             | TRIGGER     | LOGURIERRORI  | 2024-01-07 00:33:11.577 |

```

create or replace procedure InsertCAD
is

    cad UserLogin.ID%type;
begin
    select ID into cad from UserLogin where UserName = sys.LOGIN_USER() order by ID desc FETCH
FIRST 1 ROW ONLY ;
    insert into UserCAD(id_logare, tipul, nume_bd, obiect_referit, nume_object, data) values
(cad,sys.SYSEVENT(),sys.DATABASE_NAME(),sys.DICTIONARY_OBJ_TYPE(),sys.DICTIONARY_OBJ_NAME(),
systimestamp);

End;

create or replace trigger CADSchema
    after create or drop or alter on schema
    call InsertCAD

```

## Exercitiul 13

Definiți un pachet care să conțină toate obiectele definite în cadrul proiectului.

Pachetul Utility care contine proceduri și variabile auxiliare rezolvării proiectului

```

0
7   procedure AfisareIstoricPacient(codpacient ISTORIC_PACIENTI.ID_PACIENT%type);
8     procedure BoliJob(codjob number);
9
10    end Utility;
11
12  create or replace package body Utility is
13    procedure AfisareIstoricPacient...
14
15      procedure BoliJob(codjob number) is
16        cursor boli(idj number)  is (select j.NUME_DOMENIU,bs.NUME_BOALA,bs.OBSERVATII from JOB_PREDISPUSI_BOLI p
17          inner join Joburi j on p.ID_JOB = j.ID_JOB
18          join BOLI_SPECIFICE BS on BS.ID_BOALA = p.ID_BOALA
19
20      begin
21        DBMS_OUTPUT.PUT_LINE(sqlerrm);
22
23        end BoliJob;
24      end Utility;
25
26  [2024-01-11 21:22:13] completed in 32 ms

```

```

create or replace package Utility is
  type rec is record (
    codmedic      number,
    numecomplet   varchar(60),
    specializare  SPECIALIZARI.NUME_SPECIALIZARE%type,
    spitalul     SPITALE.NUME_SPITAL%type,
    judetul       judete.NUME_JUDET%type,
    numeoras      orase.NUME_ORAS%type, totalpacienti number);
  type tabelaMedici is table of rec;

procedure AfisareIstoricPacient(codpacient ISTORIC_PACIENTI.ID_PACIENT%type);
procedure BoliJob(codjob number);

end Utility;

create or replace package body Utility is
  procedure AfisareIstoricPacient(codpacient ISTORIC_PACIENTI.ID_PACIENT%type)
  is

  istoric_not_found exception;
  pragma exception_init ( istoric_not_found,-20004);
  ---tipuri de date

  type reteta is record( data_eliberarii date, Med Lista,nume_medic
varchar(50),specializare SPECIALIZARI.NUME_SPECIALIZARE%type);
  type raport is table of reteta;
  ---variabile

  Istoric raport;
  begin
    select r2.DATA_ELIBERARII,r2.Lista_Medicamente,M.NUME || ' ' ||
M.PRENUME,s2.NUME_SPECIALIZARE bulk collect into Istoric from ISTORIC_PACIENTI
      left join RETETE R2 on ISTORIC_PACIENTI.ID_RETETA = R2.ID_RETETA

```

```

        left join MEDICI M on ISTORIC_PACIENTI.ID_MEDIC = M.ID_MEDIC
        left join SPECIALIZARI S2 on M.ID_SPECIALIZARE = S2.ID_SPECIALIZARE
        where ID_PACIENT = codpacient;
    if Istoric.COUNT = 0 then
        raise_application_error(-20004,'Nu exista istoricul pacientului!');
    end if;
    for i in 1..Istoric.COUNT loop
        DBMS_OUTPUT.PUT_LINE('Dr. '||Istoric(i).nume_medic||', Specializare: '||Istoric(i).specializare);
        DBMS_OUTPUT.PUT_LINE('A eliberat in data:' || Istoric(i).data_eliberarii);
        DBMS_OUTPUT.PUT_LINE('Reteta care cuprinde medicamentele: ');
        for medicamente in (select NUME_MEDICAMENT from MEDICAMENTE where ID_MEDICAMENT in (select * from table ( Istoric(i).Med ))) loop
            DBMS_OUTPUT.PUT_LINE('- '||medicamente.NUME_MEDICAMENT);
        end loop;
    DBMS_OUTPUT.PUT_LINE('=====');
    end loop;

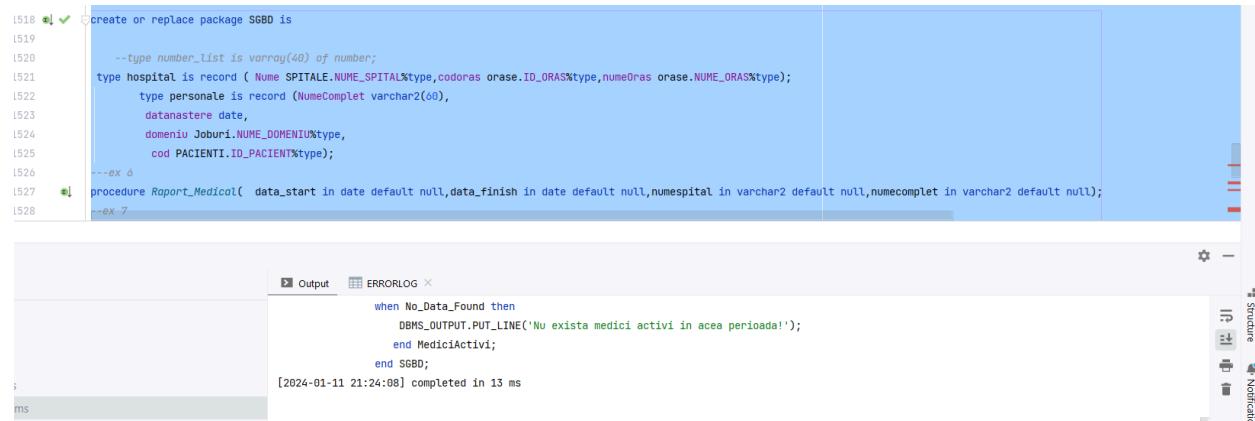
end AfisareIstoricPacient;

procedure BoliJob(codjob number) is
cursor boli(idj number)      is (select j.NUME_DOMENIU,bs.NUME_BOALA,bs.OBSERVATII from
JOB_PREDISPUSI_BOLI p
inner join Joburi j on p.ID_JOB = j.ID_JOB
join BOLI_SPECIFICE BS on BS.ID_BOALA = p.ID_BOALA
where p.ID_JOB = idj);
type bolitot is record ( domeniu Joburi.NUME_DOMENIU%type,boala
BOLI_SPECIFICE.NUME_BOALA%type,obs BOLI_SPECIFICE.OBSERVATII%type);
dbo bolitot;
no_boli exception ;
begin
open boli(codjob);
fetch boli into dbo;
if boli%notfound then
    raise no_boli;
end if;
DBMS_OUTPUT.PUT_LINE('DOMENIU :'|| DBO.domeniu);
loop
    DBMS_OUTPUT.PUT_LINE('Nume boala :'|| DBO.boala);
    DBMS_OUTPUT.PUT_LINE('Observatii :'|| DBO.obs);
    fetch boli into dbo;
    exit when boli%notfound;
end loop;
exception
when no_boli then
    DBMS_OUTPUT.PUT_LINE('Jobul nu are boli specifice!');
when others then
    DBMS_OUTPUT.PUT_LINE(sqlerrm);

end BoliJob;
end Utility;

```

### Pachetul SGBD care contine toate obiectele definite în cadrul proiectului



```

L518 ④ ✓ create or replace package SGBD is
L519
L520     --type number_list is varray(40) of number;
L521     type hospital is record ( Nume_SPITALE.NUME_SPITAL%type,codoras.orase.ID_ORAS%type,numeOras.orase.NUME_ORAS%type);
L522         type personale is record (NumeComplet varchar2(60),
L523             datanastere date,
L524             domeniu Joburi.NUME_DOMENIU%type,
L525             cod PACIENTI.ID_PACIENT%type);
L526     ---ex 6
L527     procedure Raport_Medical( data_start in date default null,data_finish in date default null,numsipital in varchar2 default null,numecomplet in varchar2 default null);
L528     ---ex 7

```

[2024-01-11 21:24:08] completed in 13 ms

```

create or replace package SGBD is

    --type number_list is varray(40) of number;
    type hospital is record ( Nume_SPITALE.NUME_SPITAL%type,codoras.orase.ID_ORAS%type,numeOras.orase.NUME_ORAS%type);
        type personale is record (NumeComplet varchar2(60),
            datanastere date,
            domeniu Joburi.NUME_DOMENIU%type,
            cod PACIENTI.ID_PACIENT%type);
    ---ex 6
    procedure Raport_Medical( data_start in date default null,data_finish in date default null,numsipital in varchar2 default null,numecomplet in varchar2 default null);
    ---ex 7
    procedure PacientiSpitaleInfo(an in date default null);
    ---ex 8
    function SchimbareDatePacient(NumeComplet varchar2, schimbam in varchar2,codnoujob in
    number, vitoareaadresa Adresa) return varchar2;
        procedure MediciActivi(anul date);
    end SGBD;
    create or replace package body SGBD is
    procedure Raport_Medical(
        data_start in date default null,
        data_finish in date default null,
        numsipital in varchar2 default null,
        numecomplet in varchar2 default null)
    is

        ---exceptii
        incorrect_date exception;
        incorrect_name exception;
        incorrect_hospital exception ;
        type vector_medicamente is varray(10) of MEDICAMENTE.NUME_MEDICAMENT%type; -- am facut
vectorul care o sa contina o lista de medicamente
        type reteta_medicamente_rec is record (nume_reteta TIP_RETETE.NUMELE_TIPULUI%type,
            medicamente vector_medicamente
        ); -- recordul pentru a retine numele retetei si
medicamentele asociatii

```

```

type tabel_imbricat_rmr is table of reteta_medicamente_rec; -- o sa retinem intr-un tabel
imbricat pentru ca un medic are mai multe retete emise
type tabel_indexat_mr is table of tabel_imbricat_rmr index by varchar2(50);
-- tabelul indexat o sa retine cheia: numele complet al medicului
-- valoare: tabela cu retetele elibererate de el
tabel_medic_retete tabel_indexat_mr;
tabel_recorduri tabel_imbricat_rmr ; -- folosim constructor pentru ca e tabel imbricat
indexxx number(3);
data_start_p date;
data_finish_p date;

begin
if data_finish is null then
    data_finish_p := sysdate;
else
    data_finish_p := data_finish;
end if;
if data_start is null then
    data_start_p:=to_date('1999/01/01','YYYY-MM-DD');
else
    data_start_p := data_start;
end if;
DBMS_OUTPUT.PUT_LINE(data_finish_p ||' start:'|| data_start_p);
if data_finish - data_start <= 0 then
    Raise incorrect_date;

DBMS_OUTPUT.PUT_LINE(data_start_p);
DBMS_OUTPUT.PUT_LINE(data_finish_p);
end if;
for medic in (
    select m.ID_MEDIC,m.nume,m.prenume from MEDICI m
left join SPITALE s on m.ID_SPITAL = s.ID_SPITAL
where ID_MEDIC in (select ID_MEDIC from ISTORIC_PACIENTI) and (numespital is null or
numespital = S.NUME_SPITAL)
) loop
indexxx :=1;
tabel_recorduri := tabel_imbricat_rmr();
for retete_medicamente in (
    select M.NUME || ' ' || M.PRENUME as Complet, R2.Lista_Medicamente,
tr.NUMELE_TIPULUI,R2.DATA_ELIBERARII from ISTORIC_PACIENTI ip
        left join MEDICI M on ip.ID_MEDIC = M.ID_MEDIC
        left join RETETE R2 on ip.ID_RETETA = R2.ID_RETETA
        left join TIP_RETETA tr on r2.TIP_RETETA = tr.ID_TIP
        where (M.NUME = medic.NUME and M.PRENUME = medic.PRENUME) and (r2.DATA_ELIBERARII
between data_start_p and data_finish_p)
) loop
tabel_recorduri.extend;
tabel_recorduri(indexxx).nume_reteta := retete_medicamente.NUMELE_TIPULUI;
tabel_recorduri(indexxx).medicamente := vector_medicamente();

tabel_recorduri(indexxx).medicamente.extend(retete_medicamente.Lista_Medicamente.COUNT);
    for i in retete_medicamente.Lista_Medicamente.first..
retete_medicamente.Lista_Medicamente.LAST loop
        select NUME_MEDICAMENT into tabel_recorduri(indexxx).medicamente(i) from MEDICAMENTE
where ID_MEDICAMENT = retete_medicamente.Lista_Medicamente(i);

    end loop;
    indexxx := indexxx+1;

```

```

        end loop;
      if tabel_recorduri.COUNT > 0 then
        tabel_medic_retete(medic.NUME || ' ' || medic.PRENUME) := tabel_recorduri;
      end if;
      end loop;

    if numecomplet is null then

      if tabel_medic_retete.count = 0 then
        raise no_data_found;
      end if;

      for med in ( select ID_MEDIC,nume || ' ' || prenume as NumeComplet from MEDICI
      where ID_MEDIC in (select ID_MEDIC from ISTORIC_PACIENTI)
        ) loop
        if tabel_medic_retete.EXISTS(med.NumeComplet) then
          DBMS_OUTPUT.PUT_LINE('| Retetele eliberate de Dr.'||med.NumeComplet||' |');
          for i in tabel_medic_retete(med.NumeComplet).first ..
          tabel_medic_retete(med.NumeComplet).last loop
            DBMS_OUTPUT.PUT_LINE('=====');
            DBMS_OUTPUT.PUT_LINE('Tipul Retetei:
'||tabel_medic_retete(med.NumeComplet)(i).nume_reteta);
            DBMS_OUTPUT.PUT_LINE('Medicamente elibarate: ');
            for j in tabel_medic_retete(med.NumeComplet)(i).medicamente.first
              ..tabel_medic_retete(med.NumeComplet)(i).medicamente.last loop
                DBMS_OUTPUT.PUT_LINE('- '||tabel_medic_retete(med.NumeComplet)(i).medicamente(j));
              end loop;
            DBMS_OUTPUT.PUT_LINE('=====');

          end loop;
        end if;
      end loop;
    else
      if tabel_medic_retete.count = 0 then
        raise no_data_found;
      end if;
      if tabel_medic_retete.EXISTS(numecomplet) = false then
        raise incorrect_name;
      end if;
      DBMS_OUTPUT.PUT_LINE('| Retetele eliberate de Dr.'||numecomplet||' |');
      for i in tabel_medic_retete(numecomplet).first ..
      tabel_medic_retete(numecomplet).last loop
        DBMS_OUTPUT.PUT_LINE('=====');
        DBMS_OUTPUT.PUT_LINE('Tipul Retetei:
'||tabel_medic_retete(numecomplet)(i).nume_reteta);
        DBMS_OUTPUT.PUT_LINE('Medicamente elibarate: ');
        for j in tabel_medic_retete(numecomplet)(i).medicamente.first
          ..tabel_medic_retete(numecomplet)(i).medicamente.last loop
            DBMS_OUTPUT.PUT_LINE('- '||tabel_medic_retete(numecomplet)(i).medicamente(j));
          end loop;
        DBMS_OUTPUT.PUT_LINE('=====');

      end loop;
    end if;
  end if;
end if;

```

```

    end if;
exception
    when incorrect_date then
        DBMS_OUTPUT.PUT_LINE('Intervalul de date in care cautati rapoarte este incorect!');
    when incorrect_name then
        DBMS_OUTPUT.PUT_LINE('Numele medicului pentru care doriti sa socateti raportul este
gresit!');
    when NO_DATA_FOUND then
        dbms_output.PUT_LINE('Nu exista o reteta eliberata intre datele respective sau numele
spitalului este gresit!');
    when others then
        DBMS_OUTPUT.PUT_LINE('A aparut o alta eroare, va rugam sa contactati staff-ul:
Eroare['||SQLERRM||']);
    end Report_Medical;
procedure PacientiSpitaleInfo(
an in date default null
)
is

    --cursoare
    cursor domiciliat is (select id_domiciliu,s.ID_ORAS from Domicilii left join Strazi s on
s.ID_STRADA = Domicilii.ID_STRADA);
    cursor pacie(dom Domiciliu.ID_DOMICILIU%type) is select NUME || ' ' ||
PRENUME,Data_Nastere,J.NUME_DOMENIU,ID_PACIENT from Pacienti
    left join Joburi J on PACIENTI.ID_JOB = J.ID_JOB
                                where dom = ID_DOMICILIU;
    cursor spitale(istoric number_list, domoras Strazi.ID_ORAS%type) is (select
s2.NUME_SPITAL,s3.ID_ORAS from Medici m left join SPITALE s2 on m.ID_SPITAL = s2.ID_SPITAL
                                left join Strazi s3 on
s3.ID_STRADA = s2.ID_SPITAL
                                where m.ID_MEDIC in (select * from table ( istoric
)) and ID_ORAS != domoras);
    --- variabile
    datepersonale personale;
    spit hospital;
    tabelamedici number_list;
    contor number := 0;
    anul_end date;

    --exceptii
begin
    if an is not null then
        anul_end := add_months(an,12);
    end if;
    select count(*) into contor from retete where an is null or DATA_ELIBERARII between an and
anul_end;
    if contor = 0 then
        raise no_data_found;
    end if;
    for dom in domiciliat loop
        open pacie(dom.ID_DOMICILIU);

        loop
            contor :=0;
            fetch pacie into datepersonale;
            exit when pacie%notfound;

```

```

select ID_MEDIC bulk collect into tabelamedici from ISTORIC_PACIENTI IP left join
RETETE R2 on R2.ID_RETETA = IP.ID_RETETA where ID_PACIENT = datepersonale.cod and (an is null
or r2.DATA_ELIBERARII between an and anul_end); -- medicii care i-au facut ceva pacientului
--- obtinem spitale din toata tara la care a fost

open spitale(tabelamedici,dom.ID_ORAS);
fetch spitale into spit.Nume,spit.codoras;

if spitale%found then
    DBMS_OUTPUT.PUT_LINE('Nume Complet: '||datepersonale.NumeComplet);
    DBMS_OUTPUT.PUT_LINE('Data de nastere: '|| datepersonale.datanastere);
    DBMS_OUTPUT.PUT_LINE('Domeniu de lucru: '|| datepersonale.domeniu);
    DBMS_OUTPUT.PUT_LINE('Spitalele la care a fost in afara Orasului: ');
    loop

        exit when spitale%notfound;
        contor:= contor+1;
        select NUME_ORAS into spit.numeOras from orase where ID_ORAS =
spit.codoras;
        DBMS_OUTPUT.PUT_LINE('- '||spit.Nume||', '||spit.numeOras || ';');
        fetch spitale into spit.Nume,spit.codoras;

    end loop;
    DBMS_OUTPUT.PUT_LINE('Total de plata suplimentar:'||contor*100);
    DBMS_OUTPUT.PUT_LINE('=====');

end if;
close spitale;

end loop;
close pacie;

end loop;
exception
when no_data_found then
    DBMS_OUTPUT.PUT_LINE('Nu exista pacientii pentru anii respectivi!');
when others then
    DBMS_OUTPUT.PUT_LINE('Contacteaza administratorul si precizeaza codul de eroare!
[,'||sqlcode'||', '|| SQLERRM||']");
    end PacientiSpitaleInfo;
    function SchimbareDatePacient
(NumeComplet varchar2, schimbam in varchar2,codnoujob in number, vitoareaadresa Adresa)
return varchar2
is
-- exceptii

incorrect_nume_judet exception;
no_job_specified exception ;
same_domiciliu exception ;
no_town_street exception ;
done_job exception ;

-- tipuri de date

```

```
--variabile
type informatiidomiciliu is record (
    codpacient PACIENTI.ID_PACIENT%type,
    coddomiciliu DOMICILII.Id_domiciliu%type,
    codstrada Strazi.id_strada%type,
    codoras orase.id_oras%type,
    codjob JOBURI.id_job%type
);
oldinfo informatiidomiciliu;
iduri number_list :=number_list();
cod number;
dom number:=-1;

begin

    select p.ID_PACIENT,d.ID_DOMICILIU,d.ID_STRADA,s.ID_ORAS,j.ID_JOB into oldinfo from
PACIENTI p
        join Domicilii d on d.ID_DOMICILIU = p.ID_PACIENT
        join Strazi s on s.ID_STRADA = d.ID_STRADA
        join joburi J on p.ID_JOB = J.ID_JOB
        where NUME ||' '|| Prenume = NumeComplet;

    if schimbam = 'Job' or schimbam is null then
        if codnoujob is null then
            raise no_job_specified;
        end if;
        Utility.BoliJob(oldinfo.codjob);
        update PACIENTI
        set ID_JOB = codnoujob
        where ID_PACIENT = oldinfo.codpacient;
        if schimbam = 'Job' then
            raise done_job;
        end if;

    end if;
    iduri.extend(3);
    iduri:= vitoareaadresa.CODURI();
    if vitoareaadresa.GetStrada() is null or vitoareaadresa.GetOras() is null or
vitoareaadresa.GetNumar() is null then
        raise no_town_street;
    end if;

    if iduri(2) = -1 then
        select ID_ORAS+1 into cod from(select * from Orase order by ID_ORAS desc) where
ROWNUM=1;

        insert into Orase(id_oras, id_judet, nume_oras) VALUES
(cod,iduri(1),vitoareaadresa.GETORAS());
        DBMS_OUTPUT.PUT_LINE('S-a adaugat in baza de date strada: '||vitoareaadresa.GetOras());
    end if;

end;
```

```

        iduri(2):=cod;
        DBMS_OUTPUT.PUT_LINE('Strada');
        select ID_STRADA+1 into cod from(select * from strazi order by ID_STRADA desc)
where ROWNUM=1;
        insert into Strazi(id_strada, nume_strada, id_oras) VALUES
(cod,vitoareaadresa.GETSTRADA(),iduri(2));
        DBMS_OUTPUT.PUT_LINE('S-a adaugat in baza de date strada: ' ||
vitoareaadresa.GetStrada());;

        iduri(3):=cod;
        elsif iduri(3) ==1 then --- daca nu exista strada in bd

        select ID_STRADA+1 into cod from(select * from Strazi order by ID_STRADA desc)
where ROWNUM=1;
        insert into Strazi(id_strada, nume_strada, id_oras) VALUES
(cod,vitoareaadresa.GETSTRADA(),iduri(2));
        DBMS_OUTPUT.PUT_LINE('S-a adaugat in baza de date strada: ' ||
vitoareaadresa.GetStrada());
        iduri(3):=cod;

        end if;
--- daca exista strada si orasul, verificam daca exista domiciliul respectiv
begin
        select ID_DOMICILIU into dom from Domicilii where ID_STRADA = iduri(3)
        and numar = vitoareaadresa.GETNUMAR() and BLOC = vitoareaadresa.GETBLOC() and
APARTAMENT = vitoareaadresa.GETAPARTAMENT() and SCARA = vitoareaadresa.GETSCARA();
        if dom = oldinfo.coddomiciliu then
            raise same_domiciliu;
        end if;
        exception
        when no_data_found then
            dom := -1;
    end;

    if dom != -1 then
        update PACIENTI set ID_DOMICILIU = dom
        where ID_PACIENT = oldinfo.codpacient;
    else
        select ID_DOMICILIU+1 into cod from(select * from domicilii order by
ID_DOMICILIU desc) where ROWNUM=1;
        insert into Domicilii(id_domiciliu, id_strada, numar,bloc,APARTAMENT,SCARA)
VALUES
(cod,iduri(3),vitoareaadresa.GETNUMAR(),vitoareaadresa.GETBLOC(),vitoareaadresa.GETAPARTAMENT(),
),vitoareaadresa.GetScara());
        update PACIENTI set ID_DOMICILIU = cod
        where ID_PACIENT = oldinfo.codpacient;
    end if;

    Utility.AfisareIstoricPacient(oldinfo.codpacient);
    return 'Noua adresa: ' || vitoareaadresa.AFISAREADRESA();
exception
when no_data_found then
    return 'Nu exista pacientul in baza de date';

```

```

when too_many_rows then
return 'Mai multi pacienti cu acelasi nume!';
when no_town_street then
return 'Nu putem sa cautam/adaugam o adresa fara Oras/Strada/Numar';
when same_domiciliu then
return 'Persoana vrea sa se mute in aceeasi casa';
when incorrect_num_judet then
return 'Nu a fost acordat un nume de judet corect, nu s-a schimbat domiciliul!';
when no_job_specified then
return 'Nu ai specificat un job, dar vrei sa il schimbi!';
when done_job then
return 'Am schimbat cu succes locul de munca al pacientului!';
when others then
    return sqlerrm;

end SchimbareDatePatient;
procedure MediciActivi(anul date)  is
--- exceptii
no_active_judet exception;
no_active_spital exception;
---variabile
Tabela Utility.tabelaMedici;
medic Utility.rec;
multi boolean:=false;
    salariuafis number;
    no_marire number;
begin

    SELECT M2.ID_MEDIC,M2.NUME ||'
'||M2.PRENUME,s5.NUME_SPECIALIZARE,s5.NUME_SPITAL,o.NUME_ORAS,j.NUME_JUDET, COUNT(ip.ID_MEDIC)
bulk collect  into Tabela
FROM MEDICI m2
left JOIN (select * from ISTORIC_PACIENTI join RETETE r on ISTORIC_PACIENTI.ID_RETETA =
r.ID_RETETA where DATA_ELIBERARII between anul and add_months(anul,12) ) ip ON M2.ID_MEDIC
= ip.ID_MEDIC
join SPITALE S5 on S5.ID_SPITAL = M2.ID_SPITAL
join Strazi st on st.ID_STRADA = s5.ID_STRADA
join orase o on o.ID_ORAS = st.ID_ORAS
join judete j on j.ID_JUDET = o.ID_JUDET
join SPECIALIZARI S5 on M2.ID_SPECIALIZARE = S5.ID_SPECIALIZARE
GROUP BY M2.NUME, M2.PRENUME,m2.ID_MEDIC,
s5.NUME_SPECIALIZARE,s5.NUME_SPITAL,o.NUME_ORAS,j.NUME_JUDET
order by m2.ID_MEDIC
;

select max(totalpacienti) into no_marire from Table(Tabela);
if no_marire = 0 then
    raise no_data_found;
end if;
--- marim salariul celui mai bun din tara
begin
    select * into medic from Table(Tabela) where totalpacienti = (select max(totalpacienti)
from Table(Tabela));
    DBMS_OUTPUT.PUT_LINE('Cel mai activ medic din Tara este Dr. '|| medic.numecomplet);
    DBMS_OUTPUT.PUT_LINE('Lucreaza la spitalul: ' || medic.spitalul);

```

```

update MEDICI set SALARIU = round(salariu +( salariu *(medic.totalpacienti *0.05)))
  where ID_MEDIC = medic.codmedic;
select SALARIU into salariuafis from MEDICI where ID_MEDIC = medic.codmedic;
Tabela(medic.codmedic).totalpacienti:=-1;
DBMS_OUTPUT.PUT_LINE('Noul sau salariu este ' || salariuafis);

exception
when too_many_rows then
  multi := true;
end;
--- in cazul in care sunt mai multi pe tara atunci coeficientul
if multi = true then
  DBMS_OUTPUT.PUT_LINE('Mai multi medici se afla la egalitate pentru bonusul pe tara!');
  DBMS_OUTPUT.PUT_LINE('Bonusul a fost redus la 0.05 per 2 pacienti');
  DBMS_OUTPUT.PUT_LINE('Cei mai buni medici din tara sunt:');
  for med in (select * from Table(Tabela) where totalpacienti = (select
max(totalpacienti) from Table(Tabela))) loop
    DBMS_OUTPUT.PUT_LINE('Dr. ' || med.numecomplet);
    update MEDICI set SALARIU = round(salariu +( salariu *(med.totalpacienti
*(0.05/2))));
    where ID_MEDIC = med.codmedic;
    select SALARIU into salariuafis from MEDICI where ID_MEDIC = med.codmedic;
    Tabela(medic.codmedic).totalpacienti:=-1;
    DBMS_OUTPUT.PUT_LINE('Noul sau salariu este ' || salariuafis);
    end loop;
    multi:=false;
  end if;
  for judet in (select * from judete) loop
    begin
      select max(totalpacienti) into no_marire from Table(Tabela) where judetul =
judet.NUME_JUDET;
      if no_marire = 0 then
        raise no_active_judet;
      end if;
      select * into medic from Table(Tabela) where totalpacienti = (select max(totalpacienti)
from Table(Tabela) where judetul = judet.NUME_JUDET) and judetul = judet.NUME_JUDET;
      DBMS_OUTPUT.PUT_LINE('Cel mai activ medic din'||judet.NUME_JUDET|| este Dr. ' ||
medic.numecomplet);
      DBMS_OUTPUT.PUT_LINE('Lucreaza la spitalul: ' || medic.spitalul);
      update MEDICI set SALARIU = round(salariu +( salariu *(medic.totalpacienti *0.03)))
      where ID_MEDIC = medic.codmedic;
      select SALARIU into salariuafis from MEDICI where ID_MEDIC = medic.codmedic;
      Tabela(medic.codmedic).totalpacienti:=-1;
      DBMS_OUTPUT.PUT_LINE('Noul sau salariu este ' || salariuafis);
    exception
      when no_active_judet then
        DBMS_OUTPUT.PUT_LINE('Nu exista medici activi in judetul'|| judet.NUME_JUDET);
        multi:=false;
      when no_data_found then
        DBMS_OUTPUT.PUT_LINE('Nu exista medici in Judetul'|| judet.NUME_JUDET );
      when too_many_rows then
        multi := true;
      end;
      if multi = true then
        DBMS_OUTPUT.PUT_LINE('Mai multi medici se afla la egalitate pentru bonusul din
judetul '||judet.NUME_JUDET||'!');
        DBMS_OUTPUT.PUT_LINE('Bonusul a fost redus la 0.03 per 2 pacienti');
        DBMS_OUTPUT.PUT_LINE('Cei mai buni medici din'||judet.NUME_JUDET|| sunt:');
      end;
    end;
  end if;
end;

```

```

        for med in (select * into medic from Table(Tabela)
                     where totalpacienti = (select max(totalpacienti)
from Table(Tabela) where judetul = judet.NUME_JUDET) and judetul = judet.NUME_JUDET) loop
                    DBMS_OUTPUT.PUT_LINE('Dr. ' || med.numecomplet);
                    update MEDICI set SALARIU = round(salariu +( salariu
* (med.totalpacienti *(0.03)/2)))
                    where ID_MEDIC = med.codmedic;
                    select SALARIU into salariuafis from MEDICI where ID_MEDIC =
med.codmedic;
                    Tabela(med.codmedic).totalpacienti:= -1;
                    DBMS_OUTPUT.PUT_LINE('Noul sau salariu este ' || salariuafis);
                end loop;
                multi:=false;
            end if;
        end loop;

--- cei mai bun din spitale
        for spital in (select * from SPITALE) loop
            begin
                select max(totalpacienti) into no_marire from Table(Tabela) where spitalul
= spital.NUME_SPITAL;
                if no_marire = 0 then
                    raise no_active_spital;
                end if;
                select * into medic from Table(Tabela) where totalpacienti = (select max(totalpacienti)
from Table(Tabela) where spitalul = spital.NUME_SPITAL) and spitalul = spital.NUME_SPITAL;
                DBMS_OUTPUT.PUT_LINE('Cel mai activ medic din '||spital.NUME_SPITAL||' este Dr. '||
medic.numecomplet);
                DBMS_OUTPUT.PUT_LINE('Avand specializarea: ' || medic.specializare);
                update MEDICI set SALARIU = round(salariu +( salariu *(medic.totalpacienti *0.03)))
                where ID_MEDIC = medic.codmedic;
                select SALARIU into salariuafis from MEDICI where ID_MEDIC = medic.codmedic;
                Tabela(medic.codmedic).totalpacienti:= -1;
                DBMS_OUTPUT.PUT_LINE('Noul sau salariu este ' || salariuafis);
            exception
                when no_active_spital then
                    multi:=false;
                DBMS_OUTPUT.PUT_LINE('Nu exista medici activi in spitalul ' ||medic.spitalul);
                when no_data_found then
                DBMS_OUTPUT.PUT_LINE('Nu exista medici in Spitalul '|| medic.spitalul );
                when too_many_rows then
                    multi := true;
                end;
                if multi = true then
                    DBMS_OUTPUT.PUT_LINE('Mai multi medici se afla la egalitate pentru bonusul
din'||spital.NUME_SPITAL||'!');
                    DBMS_OUTPUT.PUT_LINE('Bonusul a fost redus la 0.02 per 2 pacienti');
                    DBMS_OUTPUT.PUT_LINE('Cei mai buni medici din '||spital.NUME_SPITAL||'
sunt:');
                for med in (select * into medic from Table(Tabela)
                             where totalpacienti = (select max(totalpacienti)
from Table(Tabela) where spitalul = spital.NUME_SPITAL) and spitalul = spital.NUME_SPITAL)
loop
                    DBMS_OUTPUT.PUT_LINE('Dr. ' || med.numecomplet);
                    update MEDICI set SALARIU = round(salariu +( salariu
* (med.totalpacienti *(0.02)/2)))
                    where ID_MEDIC = med.codmedic;
            end;
        end;
    end;

```

```

        select SALARIU into salariuafis from MEDICI where ID_MEDIC =
med.codmedic;
        Tabela(med.codmedic).totalpacienti:= -1;
        DBMS_OUTPUT.PUT_LINE('Noul sau salariu este ' || salariuafis);
    end loop;
end if;
end loop;

exception
when No_Data_Found then
    DBMS_OUTPUT.PUT_LINE('Nu exista medici activi in acea perioada!');
end MediciActivi;
end SGBD;

```

Pachetul TriggerUtility care contine toate obiectele triggerilor

```

2130  create or replace package TriggerUtility is
2131      data_start date;
2132      data_finish date;
2133      procedure SchimbaAngajarea(startd varchar2 ,finish varchar2 );
2134      procedure insertinto;
2135      procedure insertintologoff;
2136      procedure InsertCAD;
2137      procedure inserterror;
2138  end TriggerUtility;
2139
2140  create or replace package body TriggerUtility is
2141      procedure SchimbaAngajarea...;
2142      procedure insertinto ...;
2143
2144  end TriggerUtility;

```

Output window:

```

insert into ErrorLog(id_logare, cod_eroare, eroare, data) values (errorr,cod,text,sysimestamp);

end inserterror;
end TriggerUtility;
[2024-01-11 21:30:14] completed in 6 ms

```

```

create or replace package TriggerUtility is
    data_start date;
    data_finish date;
    procedure SchimbaAngajarea(startd varchar2 ,finish varchar2 );
    procedure insertinto;
    procedure insertintologoff;
    procedure InsertCAD;
    procedure inserterror;
end TriggerUtility;

create or replace package body TriggerUtility is
    procedure SchimbaAngajarea(startd varchar2 ,finish varchar2 )
    is
    begin
        data_start := to_date(startd,'YYYY-MM-DD');
        data_finish := to_date(finish,'YYYY-MM-DD');
    end SchimbaAngajarea;
    procedure insertinto
    is
        e_logat number;
        type lastlogin is record(ultimalogare UserLogin.ID%type, data_logare timestamp);
        login lastlogin;
        sesiunee varchar(20);

```

```

        prima_logare exception;
begin
begin
    select ID,DataLogare into login.ultimalogare,login.data_logare from (select
ID,DataLogare from UserLogin where UserName = sys.LOGIN_USER() order by ID desc ) where ROWNUM
= 1;

    select COUNT(*) into e_logat from UserLogOff where UserName = sys.LOGIN_USER() and
ID_Logare = login.ultimalogare and DataDelogare > login.data_logare;
exception
when no_data_found then
login.ultimalogare:=-1;
end;
if e_logat>0 or login.ultimalogare ==-1 then
    select to_char(sys_context('userenv','sessionid')) into sesiunee from Dual where
user = sys.LOGIN_USER();
    DBMS_OUTPUT.PUT_LINE('Sesiune:' ||sesiunee ||' User: '||sys.LOGIN_USER());
    insert into UserLogin(UserLogin.ID, username, nume_bazadate, datalogare,sesiune)
values (seq_login.nextval,sys.LOGIN_USER(),sys.DATABASE_NAME(),systimestamp,sesiunee);
    end if;
end;
procedure insertintologoff
is
logoff UserLogin%rowtype;
inchis number;
sesiune number;
begin

    select * into logoff from (select * from UserLogin where UserName =
sys.LOGIN_USER() order by ID desc) where rownum =1;
    select count(*) into inchis from UserLogOff where ID_Logare = logoff.ID;
    select sys_context('userenv','sessionid') into sesiune from DUal;
    if logoff.sesiune != sesiune and inchis=0 then
    insert into UserLogOff(id_logare, username, nume_bazadate, datadelogare) values
(logoff.ID,sys.LOGIN_USER(),sys.DATABASE_NAME(),systimestamp);
    end if;
exception
when no_data_found then
    raise_application_error(-2100,'A aparut o eroare la log off!!');
end;

procedure InsertCAD
is

cad UserLogin.ID%type;
begin
    select ID into cad from UserLogin where UserName = sys.LOGIN_USER() order by ID desc FETCH
FIRST 1 ROW ONLY ;
    insert into UserCAD(id_logare, tipul, nume_bd, obiect_referit, nume_object, data) values
(cad,sys.SYSEVENT(),sys.DATABASE_NAME(),sys.DICTIONARY_OBJ_TYPE(),sys.DICTIONARY_OBJ_NAME(),
systimestamp);

end;
procedure inserterrор
is
errorr UserLogin.ID%type;
erroare varchar2(4020);

```

```

cod varchar2(20);
text varchar2(2000);
begin
erroare := DBMS_UTILITY.FORMAT_ERROR_STACK;
cod := SUBSTR(erroare, INSTR(erroare, 'ORA-', 1), 9);
text := SUBSTR(erroare,11);

select ID into errorr from UserLogin where UserName = sys.LOGIN_USER() and rownum <=1
order by ID desc ;

insert into ErrorLog(id_logare, cod_eroare, eroare, data) values
(errorr,cod,text,systimestamp);

end inserterr;
end;

```

## Exercitiul 14

Definiți un pachet care să includă tipuri de date complexe și obiecte necesare unui flux de acțiuni integrate, specifice bazei de date definite (minim 2 tipuri de date, minim 2 funcții, minim 2 proceduri).

```

create or replace package InactiveMedic is

type meds is record (
    lastreteta date,
    codmedic Medici.Id_medic%type,
    numecomplet varchar(100)
);
type tabela is table of meds;
procedure afisaremediciinactive(an date, scadere in number);
End;

create or replace package body InactiveMedic is
function collectallmedic(an date) return tabela is
medicii tabela:=tabela();

begin
    select r.DATA_ELIBERARII,medici.ID_MEDIC,Nume || ' ' || Prenume bulk collect into
medicii from MEDICI
        join ISTORIC_PACIENTI IP on MEDICI.ID_MEDIC = IP.ID_MEDIC
        join RETETE r on IP.ID_RETETA = r.ID_RETETA
    where r.DATA_ELIBERARII > an;
    return medicii;
end collectallmedic;

procedure rasplataactivi(medicia tabela) is
begin
    for i in 1..medicia.count loop
        update MEDICI set SALARIU = salariu+50
        where ID_MEDIC = medicia(i).codmedic;

    end loop;
    for i in 1..medicia.COUNT loop

```

```

        DBMS_OUTPUT.PUT_LINE('Am marit cu 50 de lei salariul dr.'||

medicia(i).numecomplet);
      end loop;
    end rasplataactivi;
function scaredesalar(activi tabela,scadere number) return boolean
is

begin
  for i in (select * from Medici where ID_MEDIC in (select codmedic from Table ( activi
))) loop
    update MEDICI set SALARIU = salariu - scadere
    where ID_MEDIC = i.ID_MEDIC;
  end loop;
  return true;
  exception
  when others then
    return false;
    end scaredesalar;
procedure afisaremediciinactive(an date,scadere in number) is
  mediciactivi tabela;
begin
  mediciactivi:= collectallmedic(an);
  if scadere is not null then
    if scaredesalar(mediciactivi,scadere) then
      DBMS_OUTPUT.PUT_LINE('Am scazut salarile medicilor inactivi cu'||scadere);
    end if;
    end if;

    for i in (select * from Medici where ID_MEDIC in (select codmedic from Table ( mediciactivi ))) loop
      DBMS_OUTPUT.PUT_LINE('Dr. '||i.NUME || ' ' || i.PRENUME);
      DBMS_OUTPUT.PUT_LINE('A fost inactiv anul acesta');
      end loop;
    rasplataactivi(mediciactivi);
  exception
  when no_data_found then
    DBMS_OUTPUT.PUT_LINE('Toti medicii au fost inactivi');
  end afisaremediciinactive;

end InactiveMedic;
declare
stringd varchar2(11):=&an;
datafinal date;
begin
datafinal:=to_date(stringd,'YYYY-MM-DD');
InactiveMedic.afisaremediciinactive(datafinal,-10);
end;

```

```
518     type tabela is table of meds;
519  520  procedure afisaremedicinactivi(an date, scadere in number);
521  522  end;
523  524  create or replace package body InactiveMedic is
525  526  function collectAllMedic(an date) return tabela is
527  528  medicii tabela:=tabela();
529  530  begin
531  532  select r.DATA_ELIBERARI,medici.ID_MEDIC,Nume || ' ' || Prenume bulk collect into medicii from MEDICI
533  534  join ISTORIC_PACIENTI IP on MEDICI.ID_MEDIC = IP.ID_MEDIC
535  536  join RETETE r on IP.ID_RETETA = r.ID_RETETA
537  538  where r.DATA_ELIBERARI > an;
539  540  return medicii;
541  end InactiveMedic;
```

Output Result 6

```
end afisaremedicinactivi;
```

```
end InactiveMedic;
```

[2024-01-12 22:46:13] completed in 11 ms