

Gestiunea unei clinici medicale

Bercă Teodora

EXERCIȚIU 1:

- Prezentați pe scurt baza de date (utilitatea ei):

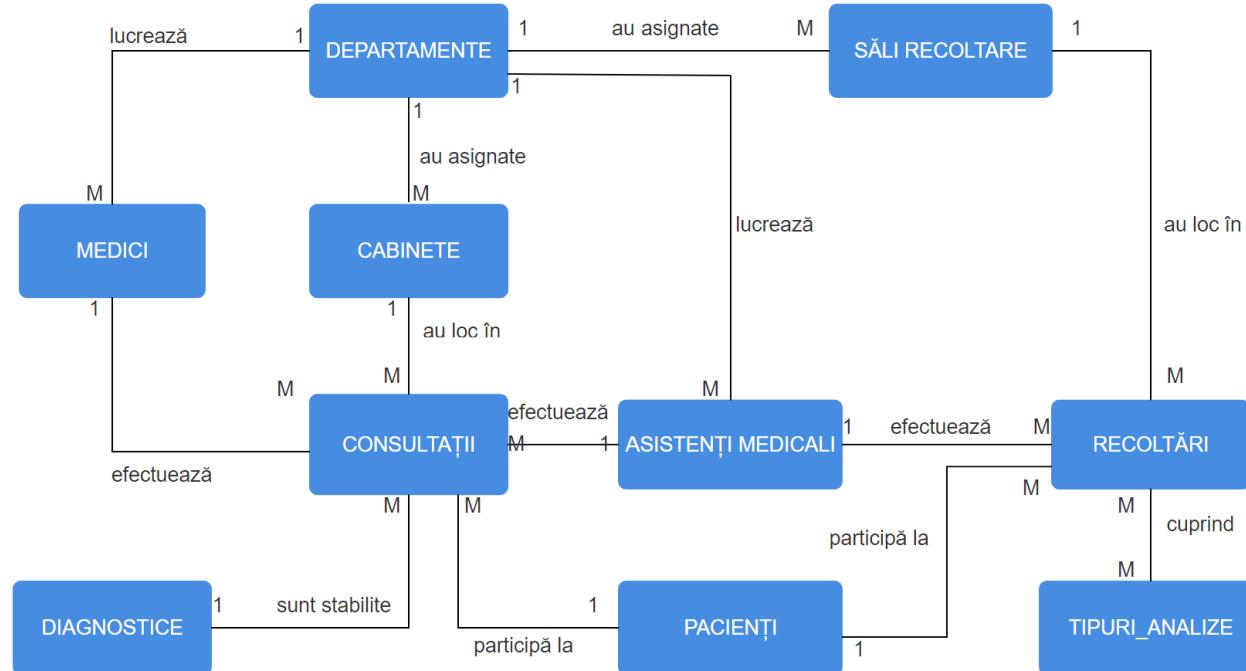
Tema proiectului este o bază de date creată pentru gestionarea activității unei clinici medicale. Clinica medicală pune la dispoziția pacienților un număr mare de cadre medicale bine pregătite, asociate unei game variate de specialități. Pacienții își pot face programări pentru consultații sau pentru recoltarea de analize medicale.

În cadrul clinicii lucrează angajați de mai multe tipuri, în funcție de rolul pe care îl au în cadrul clinicii. Astfel, există medici de diferite specialități, ce lucrează în diferite departamente, și asistenți medicali asociați tot câte unui departament. Clinica este compusă din mai multe departamente, în funcție de specialitate, iar fiecare departament are mai multe cabinete. Un pacient poate opta pentru medicul și pentru specializarea la care dorește să își facă programarea, și pentru ziua și intervalul orar. În ceea ce privește programul de recoltare de analize medicale, acestea sunt recoltate de asistenți medicali în săli special destinate pentru aceste activități.

Utilitatea acestei baze de date constă în faptul că asigură funcționarea optimă a clinicii, conținând date legate de toate aspectele fundamentale în bunul mers al acesteia, înregistrând informațiile necesare în gestionarea activităților întreprinse de clinică.

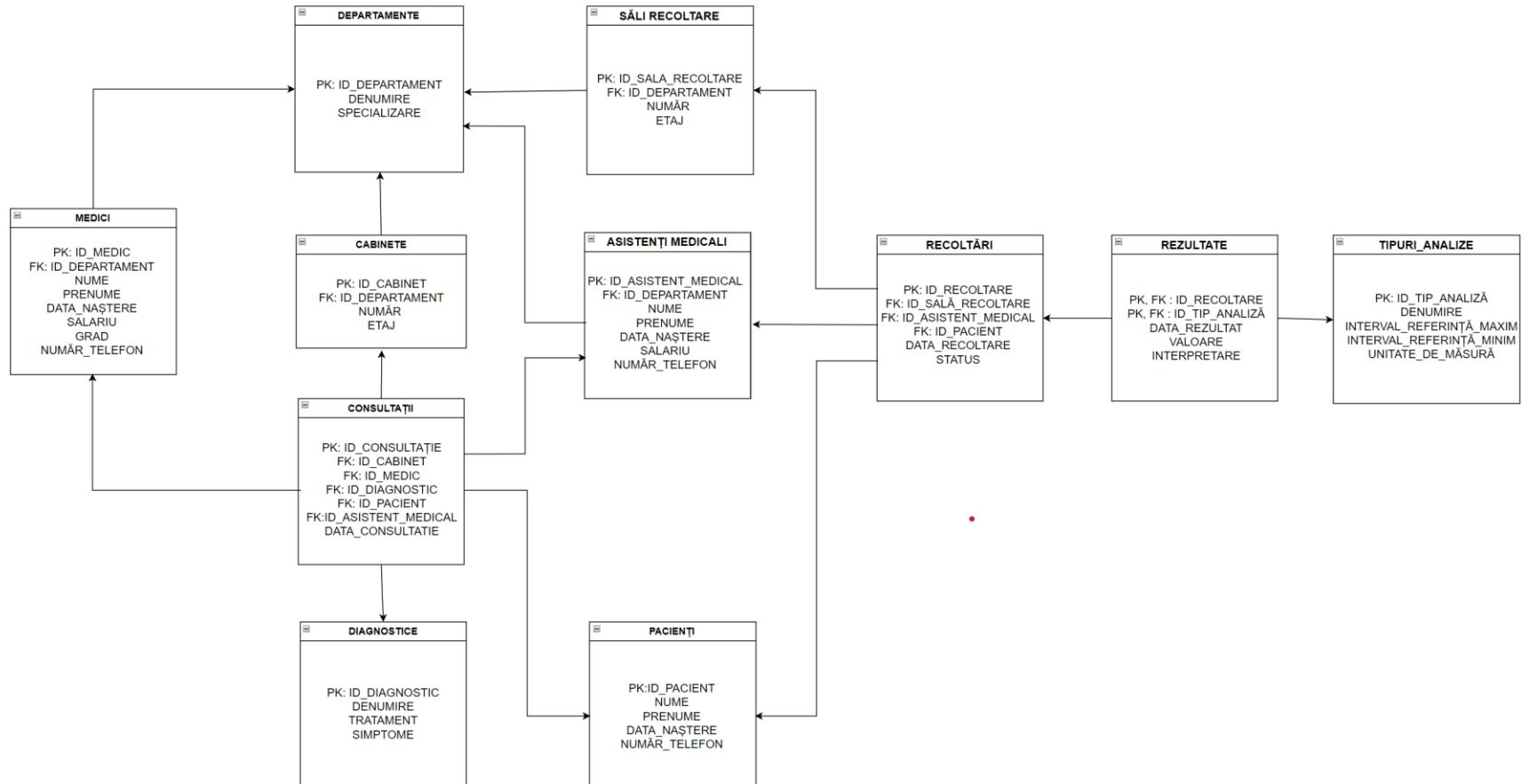
EXERCIȚIU 2:

- Realizați diagrama entitate – relație (ERD):



EXERCIȚIU 3:

- Pornind de la diagrama entitate relatie realizati diagrama conceptuala a modelului propus, integrand toate atributele necesare.



EXERCIȚIU 4:

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

The screenshot shows the Oracle SQL Worksheet interface. The 'Worksheet' tab is selected. In the main area, there is a code editor containing the SQL command to create the 'DEPARTAMENTE' table. The table has three columns: 'ID_DEPARTAMENT' (primary key), 'DENUMIRE' (not null), and 'SPECIALIZARE' (not null). The code is highlighted in yellow. Below the code editor is a 'Script Output' window showing a success message: 'Table DEPARTAMENTE created.' and 'Task completed in 0.104 seconds'.

```
CREATE TABLE DEPARTAMENTE
(
    ID_DEPARTAMENT INT PRIMARY KEY,
    DENUMIRE VARCHAR2(100) NOT NULL,
    SPECIALIZARE VARCHAR2(100) NOT NULL
);
```

Table DEPARTAMENTE created.
Task completed in 0.104 seconds

CREATE TABLE DEPARTAMENTE

```
(  
    ID_DEPARTAMENT INT PRIMARY KEY,  
    DENUMIRE VARCHAR2(100) NOT NULL,  
    SPECIALIZARE VARCHAR2(100) NOT NULL  
);
```

The screenshot shows the Oracle SQL Worksheet interface. The 'Worksheet' tab is selected. In the main area, there is a code editor containing the SQL command to create the 'MEDICI' table. The table has seven columns: 'ID_MEDIC' (primary key), 'NUME' (not null), 'PRENUME' (not null), 'DATA_NASTERE' (date type), 'NUMAR_TELEFON' (varchar2(10)), 'GRAD' (not null), and 'SALARIU' (decimal(10,2) not null). It also includes a foreign key constraint 'ID_DEPARTAMENT' referencing the 'ID_DEPARTAMENT' column in the 'DEPARTAMENTE' table. The code is highlighted in yellow. Below the code editor is a 'Script Output' window showing a success message: 'Table MEDICI created.' and 'Task completed in 0.072 seconds'.

```
CREATE TABLE MEDICI
(
    ID_MEDIC INT PRIMARY KEY,
    NUME VARCHAR2(100) NOT NULL,
    PRENUME VARCHAR2(100) NOT NULL,
    DATA_NASTERE DATE,
    NUMAR_TELEFON VARCHAR2(10),
    GRAD VARCHAR2(50) NOT NULL,
    SALARIU DECIMAL(10,2) NOT NULL,
    ID_DEPARTAMENT INT NOT NULL,
    FOREIGN KEY (ID_DEPARTAMENT) REFERENCES DEPARTAMENTE(ID_DEPARTAMENT)
);
```

Table MEDICI created.
Task completed in 0.072 seconds

CREATE TABLE MEDICI

```
(  
    ID_MEDIC INT PRIMARY KEY,  
    NUME VARCHAR2(100) NOT NULL,  
    PRENUME VARCHAR2(100) NOT NULL,  
    DATA_NASTERE DATE,  
    NUMAR_TELEFON VARCHAR2(10),  
    GRAD VARCHAR2(50) NOT NULL,
```

```
SALARIU DECIMAL(10,2) NOT NULL,  
ID_DEPARTAMENT INT NOT NULL,  
FOREIGN KEY (ID_DEPARTAMENT) REFERENCES DEPARTAMENTE(ID_DEPARTAMENT)  
);
```

```
CREATE TABLE ASISTENTI_MEDICALI
(
    ID_ASISTENT_MEDICAL INT PRIMARY KEY,
    ID_DEPARTAMENT INT NOT NULL,
    NUME VARCHAR2(100) NOT NULL,
    PRENUME VARCHAR2(100) NOT NULL,
    DATA_NASTERE DATE,
    NUMAR_TELEFON VARCHAR2(10),
    SALARIU DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (ID_DEPARTAMENT) REFERENCES DEPARTAMENTE(ID_DEPARTAMENT)
);
```

Table ASISTENTI_MEDICALI created.

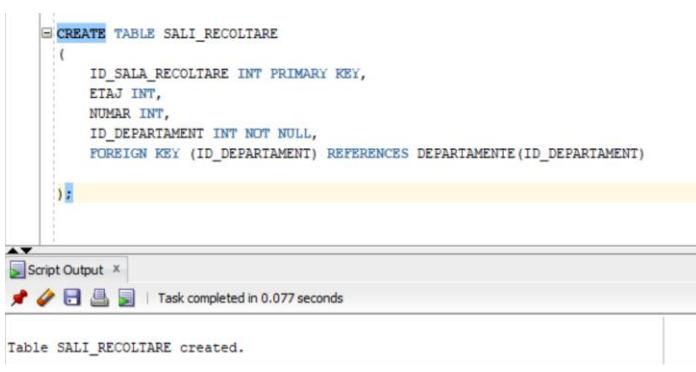
```
CREATE TABLE ASISTENTI_MEDICALI
(
    ID_ASISTENT_MEDICAL INT PRIMARY KEY,
    ID_DEPARTAMENT INT NOT NULL,
    NUME VARCHAR2(100) NOT NULL,
    PRENUME VARCHAR2(100) NOT NULL,
    DATA_NASTERE DATE,
    NUMAR_TELEFON VARCHAR2(10),
    SALARIU DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (ID_DEPARTAMENT) REFERENCES DEPARTAMENTE(ID_DEPARTAMENT)
);
```

```
CREATE TABLE CABINETE
(
    ID_CABINET INT PRIMARY KEY,
    ID_DEPARTAMENT INT NOT NULL,
    NUMAR INT,
    EТАJ INT,
    FOREIGN KEY (ID_DEPARTAMENT) REFERENCES DEPARTAMENTE(ID_DEPARTAMENT)
);
```

Table CABINETE created.

```
CREATE TABLE CABINETE
(
    ID_CABINET INT PRIMARY KEY,
```

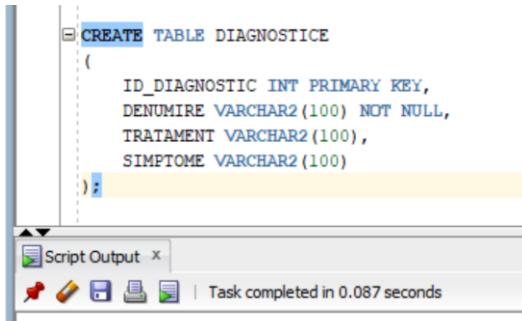
```
ID_DEPARTAMENT INT NOT NULL,  
NUMAR INT,  
ETAJ INT,  
FOREIGN KEY (ID_DEPARTAMENT) REFERENCES DEPARTAMENTE(ID_DEPARTAMENT)  
);
```



```
CREATE TABLE SALI_RECOLTARE  
(  
    ID_SALA_RECOLTARE INT PRIMARY KEY,  
    EТАJ INT,  
    NUMAR INT,  
    ID_DEPARTAMENT INT NOT NULL,  
    FOREIGN KEY (ID_DEPARTAMENT) REFERENCES DEPARTAMENTE(ID_DEPARTAMENT)  
);
```

```
CREATE TABLE SALI_RECOLTARE  
(  
    ID_SALA_RECOLTARE INT PRIMARY KEY,  
    EТАJ INT,  
    NUMAR INT,  
    ID_DEPARTAMENT INT NOT NULL,  
    FOREIGN KEY (ID_DEPARTAMENT) REFERENCES DEPARTAMENTE(ID_DEPARTAMENT)
```

```
);
```



```
CREATE TABLE DIAGNOSTICE  
(  
    ID_DIAGNOSTIC INT PRIMARY KEY,  
    DENUMIRE VARCHAR2(100) NOT NULL,  
    TRATAMENT VARCHAR2(100),  
    SIMPTOME VARCHAR2(100)  
);
```

```
Table SALI_RECOLTARE created.
```

```
Table DIAGNOSTICE created.
```

```
CREATE TABLE DIAGNOSTICE  
(  
    ID_DIAGNOSTIC INT PRIMARY KEY,  
    DENUMIRE VARCHAR2(100) NOT NULL,  
    TRATAMENT VARCHAR2(100),  
    SIMPTOME VARCHAR2(100)
```

```
DENUMIRE VARCHAR2(100) NOT NULL,  
TRATAMENT VARCHAR2(100),  
SIMPTOME VARCHAR2(100)  
);
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor with the following SQL script:

```
CREATE TABLE PACIENTI
(
    ID_PACIENT INT PRIMARY KEY,
    NUME VARCHAR2(100) NOT NULL,
    PRENUME VARCHAR2(100) NOT NULL,
    DATA_NASTERE DATE,
    NUMAR_TELEFON VARCHAR2(10)
);
```

In the bottom-right pane, there is a "Script Output" window with the following message:

```
Table PACIENTI created.
```

```
CREATE TABLE PACIENTI
(
    ID_PACIENT INT PRIMARY KEY,
    NUME VARCHAR2(100) NOT NULL,
    PRENUME VARCHAR2(100) NOT NULL,
    DATA_NASTERE DATE,
    NUMAR_TELEFON VARCHAR2(10)
);
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor with the following SQL script:

```
CREATE TABLE CONSULTATII
(
    ID_CONSULTATIE INT PRIMARY KEY,
    ID_CABINET INT NOT NULL,
    ID_MEDIC INT NOT NULL,
    ID_DIAGNOSTIC INT NOT NULL,
    ID_PACIENT INT NOT NULL,
    ID_ASISTENT_MEDICAL INT NOT NULL,
    DATA_CONSULTATIE DATE,
    FOREIGN KEY (ID_CABINET) REFERENCES CABINETE(ID_CABINET),
    FOREIGN KEY (ID_MEDIC) REFERENCES MEDICI(ID_MEDIC),
    FOREIGN KEY (ID_DIAGNOSTIC) REFERENCES DIAGNOSTICE(ID_DIAGNOSTIC),
    FOREIGN KEY (ID_PACIENT) REFERENCES PACIENTI(ID_PACIENT),
    FOREIGN KEY (ID_ASISTENT_MEDICAL) REFERENCES ASISTENȚI_MEDICALI(ID_ASISTENT_MEDICAL)
);
```

In the bottom-right pane, there is a "Script Output" window with the following message:

```
Table PACIENTI created.
```

```
CREATE TABLE CONSULTATII
(
    ID_CONSULTATIE INT PRIMARY KEY,
    ID_CABINET INT NOT NULL,
    ID_MEDIC INT NOT NULL,
    ID_DIAGNOSTIC INT NOT NULL,
```

```
ID_PACIENT INT NOT NULL,  
ID_ASISTENT_MEDICAL INT NOT NULL,  
DATA_CONSULTATIE DATE,  
FOREIGN KEY (ID_CABINET) REFERENCES CABINETE(ID_CABINET),  
FOREIGN KEY (ID_MEDIC) REFERENCES MEDICI(ID_MEDIC),  
FOREIGN KEY (ID_DIAGNOSTIC) REFERENCES DIAGNOSTICE(ID_DIAGNOSTIC),  
FOREIGN KEY (ID_PACIENT) REFERENCES PACIENTI(ID_PACIENT),  
FOREIGN KEY (ID_ASISTENT_MEDICAL) REFERENCES ASISTENCI_MEDICALI(ID_ASISTENT_MEDICAL)  
);
```

The screenshot shows the Oracle SQL Developer interface with a script editor window containing the following SQL code:

```
CREATE TABLE TIPURI_ANALIZE
(
    ID_TIP_ANALIZA INT PRIMARY KEY,
    DENUMIRE VARCHAR2(100) NOT NULL,
    INTERVAL_REFERINTA_MINIM DECIMAL(5,2) NOT NULL,
    INTERVAL_REFERINTA_MAXIM DECIMAL(5,2) NOT NULL,
    UNITATE_DE_MASURA VARCHAR2(20) NOT NULL
);
```

Below the code, the "Script Output" tab shows the results of the execution:

```
Table PACIENTI created.  
Table CONSULTATII created.  
Table TIPURI_ANALIZE created.
```

```
CREATE TABLE TIPURI_ANALIZE
(
    ID_TIP_ANALIZA INT PRIMARY KEY,
    DENUMIRE VARCHAR2(100) NOT NULL,
    INTERVAL_REFERINTA_MINIM DECIMAL(5,2) NOT NULL,
    INTERVAL_REFERINTA_MAXIM DECIMAL(5,2) NOT NULL,
    UNITATE_DE_MASURA VARCHAR2(20) NOT NULL
);
```

The screenshot shows the Oracle SQL Developer interface with a script editor window containing the following SQL code:

```
CREATE TABLE RECOLTARI
(
    ID_RECOLTARE INT PRIMARY KEY,
    ID_SALA_RECOLTARE INT NOT NULL,
    ID_ASISTENT_MEDICAL INT NOT NULL,
    ID_PACIENT INT NOT NULL,
    DATA_RECOLTARE DATE,
    STATUS VARCHAR2(100),
    FOREIGN KEY (ID_SALA_RECOLTARE) REFERENCES SALI_RECOLTARE(ID_SALA_RECOLTARE),
    FOREIGN KEY (ID_ASISTENT_MEDICAL) REFERENCES ASISTENCI_MEDICALI(ID_ASISTENT_MEDICAL),
    FOREIGN KEY (ID_PACIENT) REFERENCES PACIENTI(ID_PACIENT)
);
```

Below the code, the "Script Output" tab shows the results of the execution:

```
Table RECOLTARI created.
```

```
CREATE TABLE RECOLTARI
(
    ID_RECOLTARE INT PRIMARY KEY,
    ID_SALA_RECOLTARE INT NOT NULL,
    ID_ASISTENT_MEDICAL INT NOT NULL,
    ID_PACIENT INT NOT NULL,
    DATA_RECOLTARE DATE,
    STATUS VARCHAR2(100),
    FOREIGN KEY (ID_SALA_RECOLTARE) REFERENCES SALI_RECOLTARE(ID_SALA_RECOLTARE),
    FOREIGN KEY (ID_ASISTENT_MEDICAL) REFERENCES ASISTENTI_MEDICALI(ID_ASISTENT_MEDICAL),
    FOREIGN KEY (ID_PACIENT) REFERENCES PACIENTI(ID_PACIENT)
);
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor with the following SQL script:

```
CREATE TABLE RECOLTARI
(
    ID_RECOLTARE INT PRIMARY KEY,
    ID_SALA_RECOLTARE INT NOT NULL,
    ID_ASISTENT_MEDICAL INT NOT NULL,
    ID_PACIENT INT NOT NULL,
    DATA_RECOLTARE DATE NOT NULL,
    STATUS VARCHAR2(100),
    FOREIGN KEY (ID_SALA_RECOLTARE) REFERENCES SALI_RECOLTARE(ID_SALA_RECOLTARE),
    FOREIGN KEY (ID_ASISTENT_MEDICAL) REFERENCES ASISTENTI_MEDICALI(ID_ASISTENT_MEDICAL),
    FOREIGN KEY (ID_PACIENT) REFERENCES PACIENTI(ID_PACIENT)
);
```

In the bottom-right pane, there is a 'Script Output' window showing the results of the execution:

```
Table RECOLTARI created.

Table REZULTATE created.
```

```
CREATE TABLE REZULTATE
(
    ID_RECOLTARE INT NOT NULL,
    ID_TIP_ANALIZA INT NOT NULL,
    FOREIGN KEY(ID_RECOLTARE) REFERENCES RECOLTARI(ID_RECOLTARE),
    FOREIGN KEY(ID_TIP_ANALIZA) REFERENCES TIPURI_ANALIZE(ID_TIP_ANALIZA),
    DATA_RESULTATE DATE NOT NULL,
    VALOARE DECIMAL(5,2) NOT NULL,
    INTERPRETARE VARCHAR2(100) NOT NULL
);
```

EXERCITIUL 5:

- Adăugați date coerente în tabelele create.

ID_DEPARTAMENT	DENUMIRE	SPECIALIZARE
1	1 DEP_DERMA	DERMATOLOGIE
2	2 DEP_OFTA	OFTALMOLOGIE
3	3 DEP_ALERG	ALERGOLOGIE
4	4 DEP_CARDIO	CARDIOLOGIE
5	5 DEP_ORL	ORL
6	6 DEP_PED	PEDIATRIE
7	7 DEP_INT	MEDICINA_INTERNA
8	8 DEP_GEN	MEDICINA_GENERALA
9	9 DEP_HOM	HOMEOPATIE
10	10 DEP_ORT	ORTOPEDIE
11	11 DEP_AN	ANALIZE_MEDICALE

```

INSERT INTO DEPARTAMENTE (ID_DEPARTAMENT, DENUMIRE, SPECIALIZARE)
VALUES (1, 'DEP_DERMA', 'DERMATOLOGIE');

INSERT INTO DEPARTAMENTE (ID_DEPARTAMENT, DENUMIRE, SPECIALIZARE)
VALUES (2, 'DEP_OFTA', 'OFTALMOLOGIE');

INSERT INTO DEPARTAMENTE (ID_DEPARTAMENT, DENUMIRE, SPECIALIZARE)
VALUES (3, 'DEP_ALERG', 'ALERGOLOGIE');

INSERT INTO DEPARTAMENTE (ID_DEPARTAMENT, DENUMIRE, SPECIALIZARE)
VALUES (4, 'DEP_CARDIO', 'CARDIOLOGIE');

INSERT INTO DEPARTAMENTE (ID_DEPARTAMENT, DENUMIRE, SPECIALIZARE)
VALUES (5, 'DEP_ORL', 'ORL');

INSERT INTO DEPARTAMENTE (ID_DEPARTAMENT, DENUMIRE, SPECIALIZARE)
VALUES (6, 'DEP_PED', 'PEDIATRIE');

INSERT INTO DEPARTAMENTE (ID_DEPARTAMENT, DENUMIRE, SPECIALIZARE)
VALUES (7, 'DEP_INT', 'MEDICINA_INTERNA');

INSERT INTO DEPARTAMENTE (ID_DEPARTAMENT, DENUMIRE, SPECIALIZARE)
VALUES (8, 'DEP_GEN', 'MEDICINA_GENERALA');

INSERT INTO DEPARTAMENTE (ID_DEPARTAMENT, DENUMIRE, SPECIALIZARE)
VALUES (9, 'DEP_HOM', 'HOMEOPATIE');

INSERT INTO DEPARTAMENTE (ID_DEPARTAMENT, DENUMIRE, SPECIALIZARE)
VALUES (10, 'DEP_ORT', 'ORTOPEDIE');

INSERT INTO DEPARTAMENTE (ID_DEPARTAMENT, DENUMIRE, SPECIALIZARE)
VALUES (11, 'DEP_AN', 'ANALIZE_MEDICALE');

```

Proiect Sisteme de Gestiune a Bazelor de Date, anul 2, semestrul 1
 Berca Teodora

ID_MEDIC	NUME	PRENUME	DATA_NASTERE	NUMAR_TELEFON	GRAD	SALARIU	ID_DEPARTAMENT
1	24 Aleca	Catalina	23-MAI-80	0715545678	Primar	6500	1
2	1 Aleca	Matei	23-MAY-70	0712345678	Primar	6000	1
3	2 Busoiu	Andrei	22-JUN-72	0723456789	Specialist	5000	1
4	3 Badescu	Alexandra	05-JUN-75	0734567891	Specialist	4500	1
5	4 Coltos	Maria	14-AUG-80	0745678912	Specialist	4700	1
6	5 Ghindea	Andreea	28-MAY-69	0756789123	Primar	6200	2
7	6 Burcica	Denisa	14-JAN-74	0767891234	Primar	6500	2
8	7 Marius	Simona	09-AUG-86	0778912345	Primar	6300	2
9	8 Claudiu	Ruxandra	23-OCT-79	0789123456	Specialist	4000	3
10	9 Marcu	Narcisa	03-NOV-89	0791234567	Primar	5500	4
11	10 Bagu	Cristian	23-JUN-70	0711234567	Specialist	7000	5
12	11 Sandu	Ramona	19-APR-87	0711345678	Specialist	7000	5
13	12 Matei	Cristina	28-JUL-67	0711456789	Specialist	6800	6
14	13 Barbu	Anca	01-MAR-87	0767633554	Generalist	5000	7
15	14 Mihailescu	Matei	23-MAY-70	0788294601	Primar	6500	7
16	15 Donea	Maria	23-MAY-70	0791002789	Resident	4000	8
17	16 Priseacariu	Andrei	23-MAY-70	0712900336	Specialist	6100	8
18	17 Popescu	Daniel	23-MAY-99	0766390951	Resident	3900	9
19	18 Radu	Catalina	23-AUG-88	0791276980	Primar	5900	9
20	19 Istrate	Mihai	04-OCT-99	0713178509	Resident	4100	9
21	20 Ilie	Sorin	23-JUL-90	0738301956	Specialist	4700	10
22	21 Tanase	Matei	11-AUG-89	07865643567	Specialist	5000	10
23	22 Maria	Mihai	09-SEP-87	07865345234	Specialist	5500	10
24	23 Mihai	Miruna	08-AUG-80	0765656545	Specialist	6000	10

```

INSERT INTO MEDICI(ID_MEDIC, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON, GRAD, SALARIU, ID_DEPARTAMENT)
VALUES (1,'Aleca','Matei',TO_DATE('1970-05-23','YYYY-MM-DD'),'0712345678','Primar',6000,1);

INSERT INTO MEDICI(ID_MEDIC, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON, GRAD, SALARIU, ID_DEPARTAMENT)
VALUES (2,'Busoiu','Andrei',TO_DATE('1972-06-22','YYYY-MM-DD'),'0723456789','Specialist',5000,1);

INSERT INTO MEDICI(ID_MEDIC, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON, GRAD, SALARIU, ID_DEPARTAMENT)
VALUES (3,'Badescu','Alexandra',TO_DATE('1975-06-05','YYYY-MM-DD'),'0734567891','Specialist',4500,1);

INSERT INTO MEDICI(ID_MEDIC, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON, GRAD, SALARIU, ID_DEPARTAMENT)
VALUES (4,'Coltos','Maria',TO_DATE('1980-08-14','YYYY-MM-DD'),'0745678912','Specialist',4700,1);

INSERT INTO MEDICI(ID_MEDIC, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON, GRAD, SALARIU, ID_DEPARTAMENT)
VALUES (5,'Ghindea','Andreea',TO_DATE('1969-05-28','YYYY-MM-DD'),'0756789123','Primar',6200,2);

INSERT INTO MEDICI(ID_MEDIC, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON, GRAD, SALARIU, ID_DEPARTAMENT)
VALUES (6,'Burcica','Denisa',TO_DATE('1974-01-14','YYYY-MM-DD'),'0767891234','Primar',6500,2);

INSERT INTO MEDICI(ID_MEDIC, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON, GRAD, SALARIU, ID_DEPARTAMENT)
VALUES(7,'Marius','Simona',TO_DATE('1986-08-09','YYYY-MM-DD'),'0778912345','Primar',6300,2);

INSERT INTO MEDICI(ID_MEDIC, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON, GRAD, SALARIU, ID_DEPARTAMENT)
VALUES(8,'Claudiu','Ruxandra',TO_DATE('1979-10-23','YYYY-MM-DD'),'0789123456','Specialist',4000,3);

INSERT INTO MEDICI(ID_MEDIC, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON, GRAD, SALARIU, ID_DEPARTAMENT)
VALUES(9,'Marcu','Narcisa',TO_DATE('1989-11-03','YYYY-MM-DD'),'0791234567','Primar',5500,4);

INSERT INTO MEDICI(ID_MEDIC, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON, GRAD, SALARIU, ID_DEPARTAMENT)
VALUES(10,'Bagu','Cristian',TO_DATE('1970-06-23','YYYY-MM-DD'),'0711234567','Specialist',7000,5);

INSERT INTO MEDICI(ID_MEDIC, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON, GRAD, SALARIU, ID_DEPARTAMENT)
VALUES(11,'Sandu','Ramona',TO_DATE('1987-04-19','YYYY-MM-DD'),'0711345678','Specialist',7000,5);

INSERT INTO MEDICI(ID_MEDIC, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON, GRAD, SALARIU, ID_DEPARTAMENT)

```

Proiect Sisteme de Gestiune a Bazelor de Date, anul 2, semestrul 1
 Berca Teodora

```

VALUES(12,'Matei','Cristina',TO_DATE('1967-07-28','YYYY-MM-DD'),'0711456789','Specialist',6800,6);

INSERT INTO MEDICI(ID_MEDIC, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON, GRAD, SALARIU, ID_DEPARTAMENT)
VALUES(13,'Barbu','Anca',TO_DATE('1987-03-01','YYYY-MM-DD'),'0767633554','Generalist',5000,7);

INSERT INTO MEDICI(ID_MEDIC, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON, GRAD, SALARIU, ID_DEPARTAMENT)
VALUES(14,'Mihailescu','Matei',TO_DATE('1970-05-23','YYYY-MM-DD'),'0788294601','Primar',6500,7);

INSERT INTO MEDICI(ID_MEDIC, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON, GRAD, SALARIU, ID_DEPARTAMENT)
VALUES(15,'Donea','Maria',TO_DATE('1970-05-23','YYYY-MM-DD'),'0791002789','Rezident',4000,8);

INSERT INTO MEDICI(ID_MEDIC, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON, GRAD, SALARIU, ID_DEPARTAMENT)
VALUES(16,'Prisecariu','Andrei',TO_DATE('1970-05-23','YYYY-MM-DD'),'0712900336','Specialist',6100,8);

INSERT INTO MEDICI(ID_MEDIC, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON, GRAD, SALARIU, ID_DEPARTAMENT)
VALUES(17,'Popescu','Daniel',TO_DATE('1999-05-23','YYYY-MM-DD'),'0766389051','Rezident',3900,9);

INSERT INTO MEDICI(ID_MEDIC, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON, GRAD, SALARIU, ID_DEPARTAMENT)
VALUES(18,'Radu','Catalina',TO_DATE('1989-08-23','YYYY-MM-DD'),'0791276980','Primar',5900,9);

INSERT INTO MEDICI(ID_MEDIC, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON, GRAD, SALARIU, ID_DEPARTAMENT)
VALUES(19,'Istrate','Mihai',TO_DATE('1999-10-04','YYYY-MM-DD'),'0713178509','Rezident',4100,9);

INSERT INTO MEDICI(ID_MEDIC, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON, GRAD, SALARIU, ID_DEPARTAMENT)
VALUES(20,'Ilie','Sorin',TO_DATE('1990-07-23','YYYY-MM-DD'),'0738301956','Specialist',4700,10);

INSERT INTO MEDICI(ID_MEDIC, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON, GRAD, SALARIU, ID_DEPARTAMENT)
VALUES (21, 'Tanase', 'Matei', TO_DATE('1989-08-11','YYYY-MM-DD'),'0786543567','Specialist',5000,10);

INSERT INTO MEDICI(ID_MEDIC, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON, GRAD, SALARIU, ID_DEPARTAMENT)
VALUES (22, 'Maria', 'Mihai', TO_DATE('1987-09-09','YYYY-MM-DD'),'0786345234', 'Specialist',5500,10);

INSERT INTO MEDICI(ID_MEDIC, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON, GRAD, SALARIU, ID_DEPARTAMENT)
VALUES (23,'Mihai','Miruna', TO_DATE('1980-08-08','YYYY-MM-DD'),'0765656545', 'Specialist', 6000, 10);

```

ID_ASISTENT_MEDICAL	ID_DEPARTAMENT	NUME	PRENUME	DATA_NASTERE	NUMAR_TELEFON	SALARIU
1	1	1 Marcu	Iuliana	09-AUG-86	0789654676	2500
2	2	2 Elisei	Eliza	09-OCT-90	0784563452	2700
3	3	3 Mircea	Teodora	10-DEC-92	0712543452	3000
4	4	4 Vlad	Costin	06-MAY-73	0756476534	4000
5	5	5 Moise	Catalina	09-AUG-94	0785674564	5000
6	6	6 Florea	Marta	02-FEB-79	0783564785	2300
7	7	7 Mircu	Mia	25-JUN-80	0765754567	3500
8	8	8 Mateescu Roxana	Roxana	09-JAN-67	0712342345	4300
9	9	9 Voicu	Miruna	11-OCT-96	0743562345	4500
10	10	10 Vladimir	Sorana	11-NOV-94	0743256743	5500
11	11	11 Vlad	Xenia	25-DEC-70	0743246780	6000

```

INSERT INTO ASISTENTI_MEDICALI(ID_ASISTENT_MEDICAL, ID_DEPARTAMENT, NUME, PRENUME, DATA_NASTERE,
NUMAR_TELEFON, SALARIU)

```

Proiect Sisteme de Gestiune a Bazelor de Date, anul 2, semestrul 1
Berca Teodora

```
VALUES(1,1,'Marcu','Iuliana',TO_DATE('1986-08-09','YYYY-MM-DD'),'0789654676',2500);

INSERT INTO ASISTENTI_MEDICALI(ID_ASISTENT_MEDICAL, ID_DEPARTAMENT, NUME, PRENUME, DATA_NASTERE,
NUMAR_TELEFON, SALARIU)

VALUES(2,2,'Elisei','Eliza',TO_DATE('1990-10-09','YYYY-MM-DD'),'0784563452',2700);

INSERT INTO ASISTENTI_MEDICALI(ID_ASISTENT_MEDICAL, ID_DEPARTAMENT, NUME, PRENUME, DATA_NASTERE,
NUMAR_TELEFON, SALARIU)

VALUES(3,3,'Mircea','Teodora',TO_DATE('1992-12-10','YYYY-MM-DD'),'0712543452',3000);

INSERT INTO ASISTENTI_MEDICALI(ID_ASISTENT_MEDICAL, ID_DEPARTAMENT, NUME, PRENUME, DATA_NASTERE,
NUMAR_TELEFON, SALARIU)

VALUES(4,4,'Vlad','Costin',TO_DATE('1973-05-06','YYYY-MM-DD'),'0756476534',4000);

INSERT INTO ASISTENTI_MEDICALI(ID_ASISTENT_MEDICAL, ID_DEPARTAMENT, NUME, PRENUME, DATA_NASTERE,
NUMAR_TELEFON, SALARIU)

VALUES(5,5,'Moise','Catalina',TO_DATE('1994-08-09','YYYY-MM-DD'),'0785674564',5000);

INSERT INTO ASISTENTI_MEDICALI(ID_ASISTENT_MEDICAL, ID_DEPARTAMENT, NUME, PRENUME, DATA_NASTERE,
NUMAR_TELEFON, SALARIU)

VALUES(6,6,'Florea','Marta',TO_DATE('1979-02-02','YYYY-MM-DD'),'0783564785',2300);

INSERT INTO ASISTENTI_MEDICALI(ID_ASISTENT_MEDICAL, ID_DEPARTAMENT, NUME, PRENUME, DATA_NASTERE,
NUMAR_TELEFON, SALARIU)

VALUES(7,7,'Mircu','Mia',TO_DATE('1980-06-25','YYYY-MM-DD'),'0765754567',3500);

INSERT INTO ASISTENTI_MEDICALI(ID_ASISTENT_MEDICAL, ID_DEPARTAMENT, NUME, PRENUME, DATA_NASTERE,
NUMAR_TELEFON, SALARIU)

VALUES(8,8,'Mateescu','Roxana',TO_DATE('1967-01-09','YYYY-MM-DD'),'0712342345',4300);

INSERT INTO ASISTENTI_MEDICALI(ID_ASISTENT_MEDICAL, ID_DEPARTAMENT, NUME, PRENUME, DATA_NASTERE,
NUMAR_TELEFON, SALARIU)

VALUES(9,9,'Voicu','Miruna',TO_DATE('1996-10-11','YYYY-MM-DD'),'0743562345',4500);

INSERT INTO ASISTENTI_MEDICALI(ID_ASISTENT_MEDICAL, ID_DEPARTAMENT, NUME, PRENUME, DATA_NASTERE,
NUMAR_TELEFON, SALARIU)

VALUES(10,10,'Vladimir','Sorana',TO_DATE('1994-11-11','YYYY-MM-DD'),'0743256743',5500);

INSERT INTO ASISTENTI_MEDICALI(ID_ASISTENT_MEDICAL, ID_DEPARTAMENT, NUME, PRENUME, DATA_NASTERE,
NUMAR_TELEFON, SALARIU)

VALUES(11,11,'Vlad','Xenia',TO_DATE('1970-12-25','YYYY-MM-DD'),'0743246780',6000);
```

ID_CABINET	ID_DEPARTAMENT	NUMAR	ETAJ
1	1	1	1
2	2	1	2
3	3	2	3
4	4	3	4
5	5	4	1
6	6	5	2
7	7	6	3
8	8	7	4
9	9	8	5
10	10	9	6
11	11	10	7

```
INSERT INTO CABINETE(ID_CABINET, ETAJ,NUMAR, ID_DEPARTAMENT)
VALUES (1,0,1,1);

INSERT INTO CABINETE(ID_CABINET, ETAJ,NUMAR, ID_DEPARTAMENT)
VALUES (2,0,2,1);

INSERT INTO CABINETE(ID_CABINET, ETAJ,NUMAR, ID_DEPARTAMENT)
VALUES (3,0,3,2);

INSERT INTO CABINETE(ID_CABINET, ETAJ,NUMAR, ID_DEPARTAMENT)
VALUES (4,0,4,3);

INSERT INTO CABINETE(ID_CABINET, ETAJ,NUMAR, ID_DEPARTAMENT)
VALUES (5,1,1,4);

INSERT INTO CABINETE(ID_CABINET, ETAJ,NUMAR, ID_DEPARTAMENT)
VALUES (6,1,2,5);

INSERT INTO CABINETE(ID_CABINET, ETAJ,NUMAR, ID_DEPARTAMENT)
VALUES (7,1,3,6);

INSERT INTO CABINETE(ID_CABINET, ETAJ, NUMAR, ID_DEPARTAMENT)
VALUES (8,1,4,7);

INSERT INTO CABINETE(ID_CABINET, ETAJ,NUMAR, ID_DEPARTAMENT)
VALUES (9,1,5,8);

INSERT INTO CABINETE(ID_CABINET, ETAJ,NUMAR, ID_DEPARTAMENT)
VALUES (10,1,6,9);

INSERT INTO CABINETE(ID_CABINET, ETAJ,NUMAR, ID_DEPARTAMENT)
VALUES (11,1,7,10);
```

ID_SALA_RECOLTARE	ETAJ	NUMAR	ID_DEPARTAMENT
1	1	0	5
2	2	0	6
3	3	2	1
4	4	2	2
5	5	1	8
6	6	1	9
7	7	0	7
8	8	0	8
9	9	0	9
10	10	0	10

```
INSERT INTO SALI_RECOLTARE(ID_SALA_RECOLTARE,ETAJ,NUMAR,ID_DEPARTAMENT)
VALUES (1,0,5,11);

INSERT INTO SALI_RECOLTARE(ID_SALA_RECOLTARE,ETAJ,NUMAR,ID_DEPARTAMENT)
VALUES (2,0,6,11);

INSERT INTO SALI_RECOLTARE(ID_SALA_RECOLTARE,ETAJ,NUMAR,ID_DEPARTAMENT)
VALUES (3,2,1,11);

INSERT INTO SALI_RECOLTARE(ID_SALA_RECOLTARE,ETAJ,NUMAR,ID_DEPARTAMENT)
VALUES (4,2,2,11);

INSERT INTO SALI_RECOLTARE(ID_SALA_RECOLTARE,ETAJ,NUMAR,ID_DEPARTAMENT)
VALUES (5,1,8,11);

INSERT INTO SALI_RECOLTARE(ID_SALA_RECOLTARE,ETAJ,NUMAR,ID_DEPARTAMENT)
VALUES (6,1,9,11);

INSERT INTO SALI_RECOLTARE(ID_SALA_RECOLTARE,ETAJ,NUMAR,ID_DEPARTAMENT)
VALUES (7,0,7,11);

INSERT INTO SALI_RECOLTARE(ID_SALA_RECOLTARE,ETAJ,NUMAR,ID_DEPARTAMENT)
VALUES (8,0,8,11);

INSERT INTO SALI_RECOLTARE(ID_SALA_RECOLTARE,ETAJ,NUMAR,ID_DEPARTAMENT)
VALUES (9,0,9,11);

INSERT INTO SALI_RECOLTARE(ID_SALA_RECOLTARE,ETAJ,NUMAR,ID_DEPARTAMENT)
VALUES (10,0,10,11);
```

Proiect Sisteme de Gestiune a Bazelor de Date, anul 2, semestrul 1
 Berca Teodora

ID_DIAGNOSTIC	DENUMIRE	TRATAMENT	SIMPTOME
1	1 Acnee	Antibiotic	Cosuri,puncte negre,noduli
2	2 Dermatita	Crema specifica	Iritatie
3	3 Ulcior	Picaturi	Lacrimare
4	4 Conjurativita	Antibiotice	Mancarime
5	5 Bronsiolita	Oxigenoterapie	Tuse
6	6 Rahitism	Alimentatie echilibrata	Tulburari de crestere
7	7 Rinita	Antihistaminice	Obstructie nazala
8	8 Astm	Inhalator	Respiratie grea
9	9 Amigdalita	Antibiotice	Arsura faringiana
10	10 Deviatie de sept	Operatie	Congestie nazala
11	11 Hipertensiune	Dieta sanatoasa	Dureri de cap
12	12 Lupus	Antiinflamatoare	Dureri musculare
13	13 Tahiocardie	Medicamente	Palpitatii
14	14 Miocardita	Medicamente	Dispnee
15	15 Bronsita	Odiina prelungita	Tuse
16	16 Pneumonie	Antibiotic	Tuse productiva
17	17 Afta bucală	Glicerina boraxata	Umflatura
18	18 Carie	Plomba	Durese
19	19 Dinti strambi	Aparat dentar	Dinti strambi

INSERT INTO DIAGNOSTICE(ID_DIAGNOSTIC, DENUMIRE, TRATAMENT, SIMPTOME)

VALUES (1,'Acnee','Antibiotic','Cosuri,puncte negre,noduli');

INSERT INTO DIAGNOSTICE(ID_DIAGNOSTIC, DENUMIRE, TRATAMENT, SIMPTOME)

VALUES (2,'Dermatita','Crema specifica','Iritatie');

INSERT INTO DIAGNOSTICE(ID_DIAGNOSTIC, DENUMIRE, TRATAMENT, SIMPTOME)

VALUES (3,'Ulcior','Picaturi','Lacrimare');

INSERT INTO DIAGNOSTICE(ID_DIAGNOSTIC, DENUMIRE, TRATAMENT, SIMPTOME)

VALUES (4,'Conjurativita','Antibiotice','Mancarime');

INSERT INTO DIAGNOSTICE(ID_DIAGNOSTIC, DENUMIRE, TRATAMENT, SIMPTOME)

VALUES (5,'Bronsiolita','Oxigenoterapie','Tuse');

INSERT INTO DIAGNOSTICE(ID_DIAGNOSTIC, DENUMIRE, TRATAMENT, SIMPTOME)

VALUES (6,'Rahitism','Alimentatie echilibrata','Tulburari de crestere');

INSERT INTO DIAGNOSTICE(ID_DIAGNOSTIC, DENUMIRE, TRATAMENT, SIMPTOME)

VALUES (7,'Rinita','Antihistaminice','Obstructie nazala');

INSERT INTO DIAGNOSTICE(ID_DIAGNOSTIC, DENUMIRE, TRATAMENT, SIMPTOME)

VALUES (8,'Astm','Inhalator','Respiratie grea');

INSERT INTO DIAGNOSTICE(ID_DIAGNOSTIC, DENUMIRE, TRATAMENT, SIMPTOME)

VALUES (9,'Amigdalita','Antibiotice','Arsura faringiana');

INSERT INTO DIAGNOSTICE(ID_DIAGNOSTIC, DENUMIRE, TRATAMENT, SIMPTOME)

VALUES (10,'Deviatie de sept','Operatie','Congestie nazala');

INSERT INTO DIAGNOSTICE(ID_DIAGNOSTIC, DENUMIRE, TRATAMENT, SIMPTOME)

VALUES (11,'Hipertensiune','Dieta sanatoasa','Dureri de cap');

INSERT INTO DIAGNOSTICE(ID_DIAGNOSTIC, DENUMIRE, TRATAMENT, SIMPTOME)

VALUES (12,'Lupus','Antiinflamatoare','Dureri musculare');

```

INSERT INTO DIAGNOSTICE(ID_DIAGNOSTIC, DENUMIRE, TRATAMENT, SIMPTOME)
VALUES (13,'Tahicardie','Medicamente','Palpitatii');

INSERT INTO DIAGNOSTICE(ID_DIAGNOSTIC, DENUMIRE, TRATAMENT, SIMPTOME)
VALUES (14,'Miocardita','Medicamente','Dispnee');

INSERT INTO DIAGNOSTICE(ID_DIAGNOSTIC, DENUMIRE, TRATAMENT, SIMPTOME)
VALUES (15,'Bronsita','Odihna prelungita','Tuse');

INSERT INTO DIAGNOSTICE(ID_DIAGNOSTIC, DENUMIRE, TRATAMENT, SIMPTOME)
VALUES (16,'Pneumonie','Antibiotic','Tuse productiva');

INSERT INTO DIAGNOSTICE(ID_DIAGNOSTIC, DENUMIRE, TRATAMENT, SIMPTOME)
VALUES (17,'Afta bucală','Glicerina boraxata','Umflatura');

INSERT INTO DIAGNOSTICE(ID_DIAGNOSTIC, DENUMIRE, TRATAMENT, SIMPTOME)
VALUES (18,'Carie','Plomba','Durere');

INSERT INTO DIAGNOSTICE(ID_DIAGNOSTIC, DENUMIRE, TRATAMENT, SIMPTOME)
VALUES (19,'Dinti strambi','Aparat dentar','Dinti strambi');

```

ID_PACIENT	NUME	PRENUME	DATA_NASTERE	NUMAR_TELEFON
1	Dumitrescu	Mara	23-AUG-01	0739905709
2	Popa	Mihaela	23-JUL-90	0790221801
3	Munteanu	Ionut	10-JUL-95	0711916710
4	Gheorghe	Laurentiu	23-JUL-80	0773899018
5	Lazar	Horatiu	23-OCT-90	0718902789
6	Stanciu	Silviu	23-NOV-75	0766748938
7	Anghel	Mihnea	14-APR-83	0790108462
8	Dragomir	Cristina	10-OCT-01	0790036081
9	Craciun	Andreea	03-NOV-88	0755489891
10	Voicu	Miruna	05-JAN-68	0789011528

```

INSERT INTO PACIENTI(ID_PACIENT, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON)
VALUES (1,'Dumitrescu','Mara',TO_DATE('2001-08-23','YYYY-MM-DD'),'0739905709');

INSERT INTO PACIENTI(ID_PACIENT, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON)
VALUES (2,'Popa','Mihaela',TO_DATE('1990-07-23','YYYY-MM-DD'),'0790221801');

INSERT INTO PACIENTI(ID_PACIENT, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON)
VALUES (3,'Munteanu','Ionut',TO_DATE('1995-07-10','YYYY-MM-DD'),'0711916710');

INSERT INTO PACIENTI(ID_PACIENT, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON)
VALUES (4,'Gheorghe','Laurentiu',TO_DATE('1980-07-23','YYYY-MM-DD'),'0773899018');

INSERT INTO PACIENTI(ID_PACIENT, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON)
VALUES (5,'Lazar','Horatiu',TO_DATE('1990-10-23','YYYY-MM-DD'),'0718902789');

INSERT INTO PACIENTI(ID_PACIENT, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON)
VALUES (6,'Stanciu','Silviu',TO_DATE('1975-11-23','YYYY-MM-DD'),'0766748938');

```

Proiect Sisteme de Gestiune a Bazelor de Date, anul 2, semestrul 1
 Berca Teodora

```
INSERT INTO PACIENTI(ID_PACIENT, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON)
VALUES (7,'Anghel','Mihnea',TO_DATE('1983-04-14','YYYY-MM-DD'),'0790108462');

INSERT INTO PACIENTI(ID_PACIENT, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON)
VALUES (8,'Dragomir','Cristina',TO_DATE('2001-10-10','YYYY-MM-DD'),'0790036081');

INSERT INTO PACIENTI(ID_PACIENT, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON)
VALUES (9,'Craciun','Andreea',TO_DATE('1988-11-03','YYYY-MM-DD'),'0755489891');

INSERT INTO PACIENTI(ID_PACIENT, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON)
VALUES (10,'Voicu','Miruna',TO_DATE('1968-01-05','YYYY-MM-DD'),'0789011528');
```

	ID_CONSULTATIE	ID_CABINET	ID_MEDIC	ID_DIAGNOSTIC	ID_PACIENT	ID_ASISTENT_MEDICAL	DATA_CONSULTATIE
1	1	1	1	1	1	1	05-JAN-23
2	2	1	1	1	2	1	05-JAN-23
3	3	1	1	2	3	1	05-JAN-23
4	4	1	2	1	4	1	05-JAN-23
5	5	2	3	1	5	1	06-JAN-23
6	6	2	3	2	6	1	06-JAN-23
7	7	2	3	2	7	1	06-JAN-23
8	8	2	3	2	7	1	07-JAN-23
9	9	2	4	1	8	1	08-JAN-23
10	10	2	4	1	8	1	09-JAN-23
11	11	2	4	1	9	1	09-JAN-23
12	12	3	5	3	1	2	09-JAN-23
13	13	3	5	3	2	2	09-JAN-23
14	14	3	5	4	3	2	09-JAN-23
15	15	3	6	4	4	2	10-JAN-23
16	16	3	6	4	5	2	10-JAN-23
17	17	3	5	3	1	2	11-JAN-23
18	18	3	7	3	6	2	12-JAN-23
19	19	4	8	8	1	3	01-FEB-23
20	20	4	8	8	1	3	02-FEB-23
21	21	5	9	11	7	4	05-MAR-23
22	22	5	9	11	8	4	05-MAR-23
23	23	5	9	13	9	4	05-MAR-23
24	24	5	9	14	10	4	05-MAR-23
25	25	6	10	10	1	5	20-APR-23
26	26	6	10	10	10	5	20-APR-23
27	27	6	11	10	9	5	20-APR-23
28	28	6	11	10	8	5	20-APR-23

```
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,
DATA_CONSULTATIE)
```

```
VALUES (1,1,1,1,1,TO_DATE('2023/01/05 08:30','YYYY/MM/DD HH:MI'));
```

```
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,
DATA_CONSULTATIE)
```

```
VALUES (2,1,1,2,1,TO_DATE('2023/01/05 09:00','YYYY/MM/DD HH:MI'));
```

```
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,
DATA_CONSULTATIE)
```

```
VALUES (3,1,1,2,3,1,TO_DATE('2023/01/05 09:30','YYYY/MM/DD HH:MI'));
```

```
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,
DATA_CONSULTATIE)
```

```
VALUES (4,1,2,1,4,1,TO_DATE('2023/01/05 10:00','YYYY/MM/DD HH:MI'));
```

```
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,
DATA_CONSULTATIE)
```

```
VALUES (5,2,3,1,5,1,TO_DATE('2023/01/06 10:30','YYYY/MM/DD HH:MI'));
```

```
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,
DATA_CONSULTATIE)
```

```
VALUES (6,2,3,2,6,1,TO_DATE('2023/01/06 11:00','YYYY/MM/DD HH:MI'));
```

```
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,  
DATA_CONSULTATIE)  
  
VALUES (7,2,3,2,7,1,TO_DATE('2023/01/06 11:30','YYYY/MM/DD HH:MI'));  
  
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,  
DATA_CONSULTATIE)  
  
VALUES (8,2,3,2,7,1,TO_DATE('2023/01/07 08:00','YYYY/MM/DD HH:MI'));  
  
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,  
DATA_CONSULTATIE)  
  
VALUES (9,2,4,1,8,1,TO_DATE('2023/01/08 08:00','YYYY/MM/DD HH:MI'));  
  
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,  
DATA_CONSULTATIE)  
  
VALUES (10,2,4,1,8,1,TO_DATE('2023/01/09 08:00','YYYY/MM/DD HH:MI'));  
  
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,  
DATA_CONSULTATIE)  
  
VALUES (11,2,4,1,9,1,TO_DATE('2023/01/09 08:30','YYYY/MM/DD HH:MI'));  
  
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,  
DATA_CONSULTATIE)  
  
VALUES (12,3,5,3,1,2,TO_DATE('2023/01/09 08:00','YYYY/MM/DD HH:MI'));  
  
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,  
DATA_CONSULTATIE)  
  
VALUES (13,3,5,3,2,2,TO_DATE('2023/01/09 08:30','YYYY/MM/DD HH:MI'));  
  
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,  
DATA_CONSULTATIE)  
  
VALUES (14,3,5,4,3,2,TO_DATE('2023/01/09 09:00','YYYY/MM/DD HH:MI'));  
  
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,  
DATA_CONSULTATIE)  
  
VALUES (15,3,6,4,4,2,TO_DATE('2023/01/10 08:00','YYYY/MM/DD HH:MI'));  
  
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,  
DATA_CONSULTATIE)  
  
VALUES (16,3,6,4,5,2,TO_DATE('2023/01/10 08:30','YYYY/MM/DD HH:MI'));  
  
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,  
DATA_CONSULTATIE)  
  
VALUES (17,3,5,3,1,2,TO_DATE('2023/01/11 08:00','YYYY/MM/DD HH:MI'));  
  
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,  
DATA_CONSULTATIE)
```

```
VALUES (18,3,7,3,6,2,TO_DATE('2023/01/12 08:00','YYYY/MM/DD HH:MI'));  
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,  
DATA_CONSULTATIE)  
  
VALUES (19,4,8,8,1,3,TO_DATE('2023/02/01 08:00','YYYY/MM/DD HH:MI'));  
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,  
DATA_CONSULTATIE)  
  
VALUES (20,4,8,8,1,3,TO_DATE('2023/02/02 08:00','YYYY/MM/DD HH:MI'));  
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,  
DATA_CONSULTATIE)  
  
VALUES (21,5,9,11,7,4,TO_DATE('2023/03/05 08:00','YYYY/MM/DD HH:MI'));  
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,  
DATA_CONSULTATIE)  
  
VALUES (22,5,9,11,8,4,TO_DATE('2023/03/05 08:30','YYYY/MM/DD HH:MI'));  
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,  
DATA_CONSULTATIE)  
  
VALUES (23,5,9,13,9,4,TO_DATE('2023/03/05 09:00','YYYY/MM/DD HH:MI'));  
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,  
DATA_CONSULTATIE)  
  
VALUES (24,5,9,14,10,4,TO_DATE('2023/03/05 09:30','YYYY/MM/DD HH:MI'));  
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,  
DATA_CONSULTATIE)  
  
VALUES (25,6,10,10,1,5,TO_DATE('2023/04/20 08:00','YYYY/MM/DD HH:MI'));  
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,  
DATA_CONSULTATIE)  
  
VALUES (26,6,10,10,10,5,TO_DATE('2023/04/20 08:30','YYYY/MM/DD HH:MI'));  
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,  
DATA_CONSULTATIE)  
  
VALUES (27,6,11,10,9,5,TO_DATE('2023/04/20 09:00','YYYY/MM/DD HH:MI'));  
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,  
DATA_CONSULTATIE)  
  
VALUES (28,6,11,10,8,5,TO_DATE('2023/04/20 09:30','YYYY/MM/DD HH:MI'));  
  
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,  
DATA_CONSULTATIE)  
  
VALUES (29,7,12,18,5,6,TO_DATE('2023/05/20 08:00','YYYY/MM/DD HH:MI'));  
INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,  
DATA_CONSULTATIE)
```

```

VALUES (30,8,13,12,10,7,TO_DATE('2023/06/01 08:00','YYYY/MM/DD HH:MI'));

INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,
DATA_CONSULTATIE)

VALUES (31,8,13,16,9,7,TO_DATE('2023/06/01 08:30','YYYY/MM/DD HH:MI'));

INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,
DATA_CONSULTATIE)

VALUES (32,8,13,12,8,7,TO_DATE('2023/06/01 09:00','YYYY/MM/DD HH:MI'));

INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,
DATA_CONSULTATIE)

VALUES (33,8,14,12,7,7,TO_DATE('2023/07/21 08:00','YYYY/MM/DD HH:MI'));

INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,
DATA_CONSULTATIE)

VALUES (34,9,15,16,1,8,TO_DATE('2023/08/01 12:00','YYYY/MM/DD HH:MI'));

INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,
DATA_CONSULTATIE)

VALUES (35,9,16,16,2,8,TO_DATE('2023/08/20 12:00','YYYY/MM/DD HH:MI'));

INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,
DATA_CONSULTATIE)

VALUES (36,10,17,5,4,9,TO_DATE('2023/08/21 10:00','YYYY/MM/DD HH:MI'));

INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,
DATA_CONSULTATIE)

VALUES (37,10,18,8,5,9,TO_DATE('2023/08/21 10:30','YYYY/MM/DD HH:MI'));

INSERT INTO CONSULTATII (ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,
DATA_CONSULTATIE)

VALUES (38,10,19,15,6,9,TO_DATE('2023/08/21 11:00','YYYY/MM/DD HH:MI'));

```

ID_TIP_ANALIZA	DENUMIRE	INTERVAL_REFERINTA_MINIM	INTERVAL_REFERINTA_MAXIM	UNITATE_DE_MASURA
1	1 Glicemie	70	100 mg/dl	
2	2 Magneziu	1.6	2.6 mg/dl	
3	3 Vitamina D	50	125 nmol/l	
4	4 Calciu Ionic	3.82	4.82 mg/dl	
5	5 Hemoglobina Glicata	0	5.7 procent	

```

INSERT INTO TIPURI_ANALIZE (ID_TIP_ANALIZA, DENUMIRE, INTERVAL_REFERINTA_MINIM, INTERVAL_REFERINTA_MAXIM,
UNITATE_DE_MASURA)

```

```

VALUES (1,'Glicemie',70,100,'mg/dl');

```

```

INSERT INTO TIPURI_ANALIZE (ID_TIP_ANALIZA, DENUMIRE, INTERVAL_REFERINTA_MINIM, INTERVAL_REFERINTA_MAXIM,
UNITATE_DE_MASURA)

```

```

VALUES (2,'Magneziu',1.6,2.6,'mg/dl');

```

```

INSERT INTO TIPURI_ANALIZE (ID_TIP_ANALIZA, DENUMIRE, INTERVAL_REFERINTA_MINIM, INTERVAL_REFERINTA_MAXIM,
UNITATE_DE_MASURA)

```

VALUES (3,'Vitamina D',50,125,'nmol/l');

INSERT INTO TIPURI_ANALIZE (ID_TIP_ANALIZA, DENUMIRE, INTERVAL_REFERINTA_MINIM, INTERVAL_REFERINTA_MAXIM, UNITATE_DE_MASURA)

VALUES (4,'Calciu Ionic',3.82,4.82,'mg/dl');

INSERT INTO TIPURI_ANALIZE (ID_TIP_ANALIZA, DENUMIRE, INTERVAL_REFERINTA_MINIM, INTERVAL_REFERINTA_MAXIM, UNITATE_DE_MASURA)

VALUES (5,'Hemoglobina Glicata',0.5,7,'procent');

ID_RECOLTARE	ID_SALA_RECOLTARE	ID_ASISTENT_MEDICAL	ID_PACIENT	DATA_RECOLTARE	STATUS
1	1	1	11	1 21-AUG-23	Efectuata
2	2	2	11	2 21-SEP-23	Efectuata
3	3	3	11	3 22-SEP-23	Efectuata
4	4	4	11	1 21-OCT-23	Efectuata
5	5	5	11	1 21-DEC-23	Efectuata
6	6	5	11	2 21-NOV-23	Efectuata

INSERT INTO RECOLTARI (ID_RECOLTARE, ID_SALA_RECOLTARE, ID_ASISTENT_MEDICAL, ID_PACIENT, DATA_RECOLTARE, STATUS)

VALUES (1,1,11,1,TO_DATE('2023/08/21 11:00','YYYY/MM/DD HH:MI'),'Efectuata');

INSERT INTO RECOLTARI (ID_RECOLTARE, ID_SALA_RECOLTARE, ID_ASISTENT_MEDICAL, ID_PACIENT, DATA_RECOLTARE, STATUS)

VALUES (2,2,11,2,TO_DATE('2023/09/21 08:00','YYYY/MM/DD HH:MI'),'Efectuata');

INSERT INTO RECOLTARI (ID_RECOLTARE, ID_SALA_RECOLTARE, ID_ASISTENT_MEDICAL, ID_PACIENT, DATA_RECOLTARE, STATUS)

VALUES (3,3,11,3,TO_DATE('2023/09/22 08:00','YYYY/MM/DD HH:MI'),'Efectuata');

INSERT INTO RECOLTARI (ID_RECOLTARE, ID_SALA_RECOLTARE, ID_ASISTENT_MEDICAL, ID_PACIENT, DATA_RECOLTARE, STATUS)

VALUES (4,4,11,1,TO_DATE('2023/10/21 08:00','YYYY/MM/DD HH:MI'),'Efectuata');

INSERT INTO RECOLTARI (ID_RECOLTARE, ID_SALA_RECOLTARE, ID_ASISTENT_MEDICAL, ID_PACIENT, DATA_RECOLTARE, STATUS)

VALUES (5,5,11,1,TO_DATE('2023/12/21 08:00','YYYY/MM/DD HH:MI'),'Efectuata');

INSERT INTO RECOLTARI (ID_RECOLTARE, ID_SALA_RECOLTARE, ID_ASISTENT_MEDICAL, ID_PACIENT, DATA_RECOLTARE, STATUS)

VALUES (6,5,11,2,TO_DATE('2023/11/21 08:30','YYYY/MM/DD HH:MI'),'Efectuata');

ID_RECOLTARE	ID_TIP_ANALIZA	DATA_RESULTATE	VALOARE	INTERPRETARI
1	1	1 30-AUG-23	90	NORMAL
2	1	2 30-AUG-23	2	NORMAL
3	1	3 30-AUG-23	70	NORMAL
4	2	1 25-SEP-23	100	NORMAL
5	2	2 25-SEP-23	1	SCAZUT
6	3	5 30-SEP-23	6	CRESUT
7	4	4 23-OCT-23	4	NORMAL
8	4	5 23-OCT-23	3	NORMAL
9	5	1 25-DEC-23	120	CRESUT
10	6	1 23-NOV-23	80	NORMAL
11	6	2 23-NOV-23	2	NORMAL
12	6	3 23-NOV-23	70	NORMAL

```
INSERT INTO REZULTATE (ID_RECOLTARE, ID_TIP_ANALIZA, DATA_REZULTATE, VALOARE, INTERPRETARE)
VALUES (1, 1, TO_DATE('2023/08/30 08:00','YYYY/MM/DD HH:MI'), 90, 'NORMAL');

INSERT INTO REZULTATE (ID_RECOLTARE, ID_TIP_ANALIZA, DATA_REZULTATE, VALOARE, INTERPRETARE)
VALUES (1, 2, TO_DATE('2023/08/30 08:30','YYYY/MM/DD HH:MI'), 2.0, 'NORMAL');

INSERT INTO REZULTATE (ID_RECOLTARE, ID_TIP_ANALIZA, DATA_REZULTATE, VALOARE, INTERPRETARE)
VALUES (1, 3, TO_DATE('2023/08/30 08:00','YYYY/MM/DD HH:MI'), 70, 'NORMAL');

INSERT INTO REZULTATE (ID_RECOLTARE, ID_TIP_ANALIZA, DATA_REZULTATE, VALOARE, INTERPRETARE)
VALUES (2, 1, TO_DATE('2023/09/25 08:30','YYYY/MM/DD HH:MI'), 100, 'NORMAL');

INSERT INTO REZULTATE (ID_RECOLTARE, ID_TIP_ANALIZA, DATA_REZULTATE, VALOARE, INTERPRETARE)
VALUES (2, 2, TO_DATE('2023/09/25 08:30','YYYY/MM/DD HH:MI'), 1.0, 'SCAZUT');

INSERT INTO REZULTATE (ID_RECOLTARE, ID_TIP_ANALIZA, DATA_REZULTATE, VALOARE, INTERPRETARE)
VALUES (3, 5, TO_DATE('2023/09/30 08:30','YYYY/MM/DD HH:MI'), 6, 'CRESCUT');

INSERT INTO REZULTATE (ID_RECOLTARE, ID_TIP_ANALIZA, DATA_REZULTATE, VALOARE, INTERPRETARE)
VALUES (4, 4, TO_DATE('2023/10/23 08:30','YYYY/MM/DD HH:MI'), 4, 'NORMAL');

INSERT INTO REZULTATE (ID_RECOLTARE, ID_TIP_ANALIZA, DATA_REZULTATE, VALOARE, INTERPRETARE)
VALUES (4, 5, TO_DATE('2023/10/23 08:30','YYYY/MM/DD HH:MI'), 3, 'NORMAL');

INSERT INTO REZULTATE (ID_RECOLTARE, ID_TIP_ANALIZA, DATA_REZULTATE, VALOARE, INTERPRETARE)
VALUES (5, 1, TO_DATE('2023/12/25 08:30','YYYY/MM/DD HH:MI'), 120, 'CRESCUT');

INSERT INTO REZULTATE (ID_RECOLTARE, ID_TIP_ANALIZA, DATA_REZULTATE, VALOARE, INTERPRETARE)
VALUES (6, 1, TO_DATE('2023/11/23 08:30','YYYY/MM/DD HH:MI'), 80, 'NORMAL');

INSERT INTO REZULTATE (ID_RECOLTARE, ID_TIP_ANALIZA, DATA_REZULTATE, VALOARE, INTERPRETARE)
VALUES (6, 2, TO_DATE('2023/11/23 08:30','YYYY/MM/DD HH:MI'), 2.0, 'NORMAL');

INSERT INTO REZULTATE (ID_RECOLTARE, ID_TIP_ANALIZA, DATA_REZULTATE, VALOARE, INTERPRETARE)
VALUES (6, 3, TO_DATE('2023/11/23 08:30','YYYY/MM/DD HH:MI'), 70, 'NORMAL');
```

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.

CERINTA: Dat fiind id-ul unui pacient, să se afișeze numele și prenumele acestuia urmate de istoricul său medical: fiecare set de analize făcut împreună cu rezultatele și toate diagnosticile sale.

```
CREATE OR REPLACE PROCEDURE ISTORIC_PACIENT (V_ID_PACIENTI.ID_PACIENT%TYPE)
```

```
AS
```

```
--VECTORUL IN CARE RETINEM PENTRU O RECOLTARE REZULTATELE ANALIZELOR FACUTE
```

Proiect Sisteme de Gestiune a Bazelor de Date, anul 2, semestrul 1
Berca Teodora

```
TYPE DATE_ANALIZE IS RECORD(
    TIP_ANALIZA TIPURI_ANALIZE.DENUMIRE%TYPE,
    REZULTAT REZULTATE.VALOARE%TYPE,
    INTERPRETARE REZULTATE.INTERPRETARE%TYPE
);

TYPE SET_REZULTATE IS VARRAY(6) OF DATE_ANALIZE;
--TABLOU CU TOATE SETURILE DE ANALIZE

TYPE COLECTIE_SET_REZULTATE IS TABLE OF SET_REZULTATE;
V_COLECTIE_SET_REZULTATE COLECTIE_SET_REZULTATE := COLECTIE_SET_REZULTATE();
--TABLOUL IMBRICAT CU TOATE DIAGNOSTICELE PE CARE LE-A PRIMIT UN PACIENT

TYPE ISTORIC_DIAGNOSTICE IS TABLE OF DIAGNOSTICE%ROWTYPE;
V_ISTORIC_DIAGNOSTICE ISTORIC_DIAGNOSTICE;
--TABLOUL INDEXAT IN CARE RETIN IN ORDINE CRESCATOARE A DATELOR CALENDARISTICE ANALIZELE MEDICALE FACUTE DE
PACIENT

TYPE ISTORIC_RECOLTARI IS TABLE OF RECOLTARI%ROWTYPE INDEX BY PLS_INTEGER;
V_ISTORIC_RECOLTARI ISTORIC_RECOLTARI;
--VARIABLE AJUTATOARE

V_ULTIMA_RECOLTARE RECOLTARI.ID_RECOLTARE%TYPE;
--NUME SI PRENUME

V_NUME PACIENTI.NUME%TYPE;
V_PRENUME PACIENTI.PRENUME%TYPE;

BEGIN
    SELECT NUME
    INTO V_NUME
    FROM PACIENTI
    WHERE ID_PACIENT = V_ID;
    SELECT PRENUME
    INTO V_PRENUME
    FROM PACIENTI
    WHERE ID_PACIENT = V_ID;
    DBMS_OUTPUT.PUT_LINE('NUME: '||V_NUME||' PRENUME: '||V_PRENUME);
--RETIN IN TABLOUL IMBRICAT TOATE DIAGNOSTICELE

SELECT D.ID_DIAGNOSTIC, D.DENUMIRE, D.TRATAMENT, D.SIMPTOME
```

Proiect Sisteme de Gestiune a Bazelor de Date, anul 2, semestrul 1
Berca Teodora

```
BULK COLLECT INTO V_ISTORIC_DIAGNOSTICE
FROM DIAGNOSTICE D
JOIN CONSULTATII C ON C.ID_DIAGNOSTIC = D.ID_DIAGNOSTIC
JOIN PACIENTI P ON C.ID_PACIENT = P.ID_PACIENT
WHERE P.ID_PACIENT = V_ID;
--RETIN IN TABLOUL INDEXAT TOATE RECOLTARILE ORDONATE DUPA DATA CRESCATOR
SELECT R.ID_RECOLTARE, R.ID_SALA_RECOLTARE, R.ID_ASISTENT_MEDICAL, R.ID_PACIENT, R.DATA_RECOLTARE, R.STATUS
BULK COLLECT INTO V_ISTORIC_RECOLTARI
FROM RECOLTARI R
JOIN PACIENTI P ON P.ID_PACIENT = R.ID_PACIENT
WHERE P.ID_PACIENT = V_ID
ORDER BY R.DATA_RECOLTARE ASC;
IF V_ISTORIC_RECOLTARI.COUNT = 0 THEN
    DBMS_OUTPUT.PUT_LINE ('PACIENTUL NU A FACUT ANALIZE MEDICALE');
ELSE
    DBMS_OUTPUT.PUT_LINE('ISTORIC ANALIZE MEDICALE:');
    FOR I IN V_ISTORIC_RECOLTARI.FIRST..V_ISTORIC_RECOLTARI.LAST LOOP
        V_COLECTIE_SET_RESULTATE.EXTEND;
        V_COLECTIE_SET_RESULTATE(V_COLECTIE_SET_RESULTATE.LAST):=SET_RESULTATE();
        V_COLECTIE_SET_RESULTATE(V_COLECTIE_SET_RESULTATE.LAST).EXTEND(5);
        SELECT TA.DENUMIRE, REZ.VALOARE, REZ.INTERPRETARE
        BULK COLLECT INTO V_COLECTIE_SET_RESULTATE(V_COLECTIE_SET_RESULTATE.LAST)
        FROM RECOLTARI R
        JOIN PACIENT P ON P.ID_PACIENT = R.ID_PACIENT
        JOIN RESULTATE REZ ON REZ.ID_RECOLTARE = R.ID_RECOLTARE
        JOIN TIPURI_ANALIZE TA ON TA.ID_TIP_ANALIZA = REZ.ID_TIP_ANALIZA
        WHERE P.ID_PACIENT = V_ID AND R.ID_RECOLTARE = V_ISTORIC_RECOLTARI(I).ID_RECOLTARE;
    END LOOP;
    FOR I IN V_COLECTIE_SET_RESULTATE.FIRST..V_COLECTIE_SET_RESULTATE.LAST LOOP
        DBMS_OUTPUT.PUT_LINE('RECOLTAREA CU ID-UL: '||V_ISTORIC_RECOLTARI(I).ID_RECOLTARE|| ' DE LA DATA:
'||V_ISTORIC_RECOLTARI(I).DATA_RECOLTARE);
        FOR J IN V_COLECTIE_SET_RESULTATE(I).FIRST..V_COLECTIE_SET_RESULTATE(I).LAST LOOP
            DBMS_OUTPUT.PUT_LINE(
```

```
V_COLECTIE_SET_RESULTATE(I)(J).TIP_ANALIZA || '' ||
V_COLECTIE_SET_RESULTATE(I)(J).RESULTAT || '' ||
V_COLECTIE_SET_RESULTATE(I)(J).INTERPRETARE
);

END LOOP;

END LOOP;

END IF;

IF V_ISTORIC_DIAGNOSTICE.COUNT = 0 THEN

DBMS_OUTPUT.PUT_LINE('PACIENTUL NU A PRIMIT DIAGNOSTICE');

ELSE

DBMS_OUTPUT.PUT_LINE('ISTORIC DIAGNOSTICE:');

FOR I IN V_ISTORIC_DIAGNOSTICE.FIRST..V_ISTORIC_DIAGNOSTICE.LAST LOOP

DBMS_OUTPUT.PUT_LINE(
    V_ISTORIC_DIAGNOSTICE(I).DENUMIRE
);

END LOOP;

END IF;

END;

EXECUTE ISTORIC_PACIENT(1);

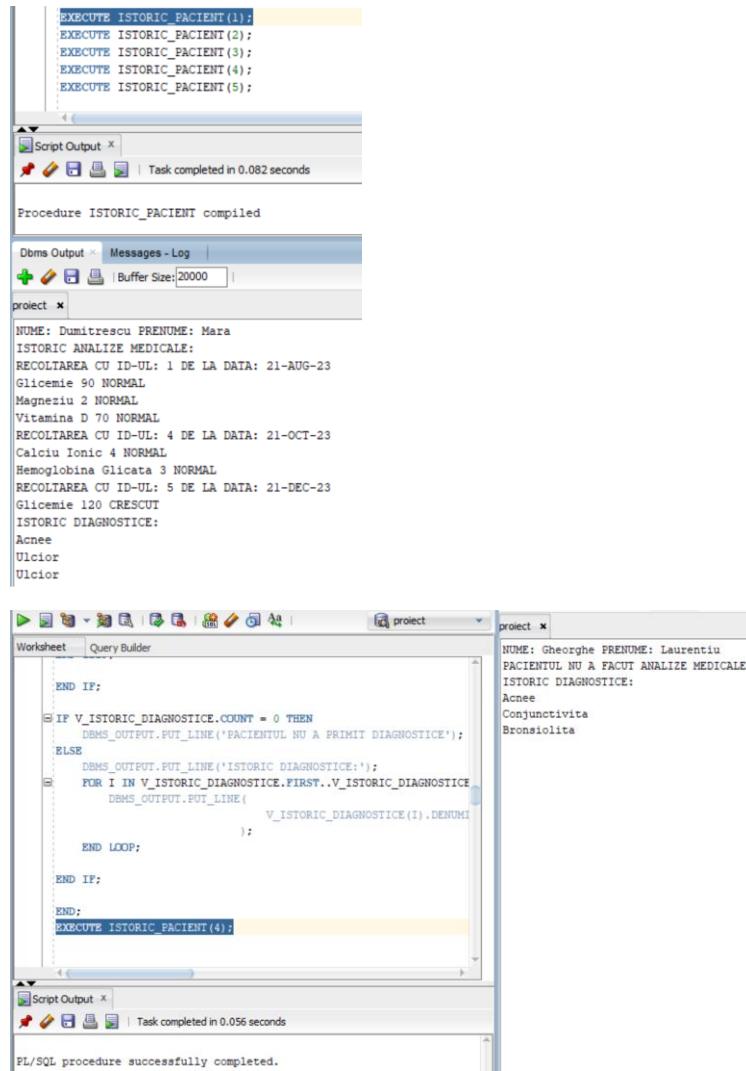
EXECUTE ISTORIC_PACIENT(4);
```

The screenshot shows the Oracle SQL Developer interface with a code editor and a script output window.

In the code editor, the following PL/SQL code is displayed:

```
--EXERCITIUL 6
CREATE OR REPLACE PROCEDURE ISTORIC_PACIENT (V_ID_PACIENTI.ID_PACIENT%TYPE)
AS
--VECTORUL IN CARE RETINEM PENTRU O RECOLTARE REZULTATELE ANALIZELOR FACUTE
TYPE DATE_ANALIZE IS RECORD(
    TIP_ANALIZA TIPURI_ANALIZE.DENUMIRE%TYPE,
    REZULTAT REZULTATE.VALORE%TYPE,
    INTERPRETARE REZULTATE.INTERPRETARE%TYPE
);
TYPE SET_RESULTATE IS VARRAY(6) OF DATE_ANALIZE;
```

In the script output window, the message "Procedure ISTORIC_PACIENT compiled" is shown, indicating the successful compilation of the procedure.



The screenshot shows the Oracle SQL Developer interface. On the left, there is a code editor window containing PL/SQL code. The code includes several EXECUTE statements for a procedure named ISTORIC_PACIENT, which is shown to have been compiled successfully. Below the code editor is a 'Script Output' window showing the results of the execution. The output details a patient's history, including various medical tests and their results (e.g., Glicemie 90 NORMAL, Magneziu 2 NORMAL, Vitamina D 70 NORMAL) and a list of diagnoses (e.g., Acnee, Conjunctivita, Bronsiolita). To the right of the code editor is a 'project' browser.

```

EXECUTE ISTORIC_PACIENT(1);
EXECUTE ISTORIC_PACIENT(2);
EXECUTE ISTORIC_PACIENT(3);
EXECUTE ISTORIC_PACIENT(4);
EXECUTE ISTORIC_PACIENT(5);

Procedure ISTORIC_PACIENT compiled

Dbms Output X Messages - Log | Buffer Size:20000 |
project x

NUME: Dumitrescu PRENUME: Mara
ISTORIC ANALIZE MEDICALE:
RECOLTAREA CU ID-UL: 1 DE LA DATA: 21-AUG-23
Glicemie 90 NORMAL
Magneziu 2 NORMAL
Vitamina D 70 NORMAL
RECOLTAREA CU ID-UL: 4 DE LA DATA: 21-OCT-23
Calcii Ionic 4 NORMAL
Hemoglobina Glicata 3 NORMAL
RECOLTAREA CU ID-UL: 5 DE LA DATA: 21-DEC-23
Glicemie 120 CRESCUT
ISTORIC DIAGNOSTICE:
Acnee
Ulcior
Ulcior

```

```

END IF;
IF V_ISTORIC_DIAGNOSTICE.COUNT = 0 THEN
    DBMS_OUTPUT.PUT_LINE('PACIENTUL NU A PRIMIT DIAGNOSTICE');
ELSE
    DBMS_OUTPUT.PUT_LINE('ISTORIC DIAGNOSTICE:');
    FOR I IN V_ISTORIC_DIAGNOSTICE.FIRST..V_ISTORIC_DIAGNOSTICE.LAST
        DBMS_OUTPUT.PUT_LINE(
            V_ISTORIC_DIAGNOSTICE(I).DENUMIRE);
    END LOOP;
END IF;
END;
EXECUTE ISTORIC_PACIENT(4);

```

PL/SQL procedure successfully completed.

EXERCIȚIU 7:

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

CERINȚĂ: Se dă id-ul unui departament și se cere să se afișeze pentru fiecare medic ce are salariul egal cu salariul maxim din cadrul departamentului numărul de consultații efectuate.

Proiect Sisteme de Gestiune a Bazelor de Date, anul 2, semestrul 1
 Berca Teodora

```

CREATE OR REPLACE PROCEDURE TOP_MEDICI_DEPARTAMENT(V_DEP DEPARTAMENTE.ID_DEPARTAMENT%TYPE)
AS

TYPE ID_MEDICI IS TABLE OF MEDICI.ID_MEDIC%TYPE INDEX BY PLS_INTEGER;

V_MEDICI ID_MEDICI;
V_NUMAR INT;
V_CURSOR INT :=0;
V_TEST DEPARTAMENTE.ID_DEPARTAMENT%TYPE;

```

Script Output X | Task completed in 0.116 seconds

Procedure TOP_MEDICI_DEPARTAMENT compiled

```

FOR I IN C1 LOOP
    V_CURSOR:=V_CURSOR+1;
    OPEN C2(I.ID_MEDIC);
    FETCH C2 INTO V_NUMAR;
    EXIT WHEN C2%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('MEDICUL '||I.NUME||' '||I.PRENUME||' CU SALARIUL ');
    CLOSE C2;
END LOOP;
IF V_CURSOR = 0 THEN
    DBMS_OUTPUT.PUT_LINE('NU EXISTA MEDICI IN ACEST DEPARTAMENT');
END IF;
EXCEPTION
    WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20000, 'NU EXISTA DEP');
END;
EXECUTE TOP_MEDICI_DEPARTAMENT(11);
EXECUTE TOP_MEDICI_DEPARTAMENT(5);

```

PL/SQL procedure successfully completed.

```

END LOOP;
IF V_CURSOR = 0 THEN
    DBMS_OUTPUT.PUT_LINE('NU EXISTA MEDICI IN ACEST DEPARTAMENT');
END IF;
EXCEPTION
    WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20000, 'NU EXISTA DEPARTAMENT CU ID-UL DAT');

END;
EXECUTE TOP_MEDICI_DEPARTAMENT(11);
EXECUTE TOP_MEDICI_DEPARTAMENT(5);
EXECUTE TOP_MEDICI_DEPARTAMENT(1);
EXECUTE TOP_MEDICI_DEPARTAMENT(23);

```

--EXERCITIU 9

Script Output X | Task completed in 0.064 seconds

```

BEGIN TOP_MEDICI_DEPARTAMENT(23); END;
Error report -
ORA-20000: NU EXISTA DEPARTAMENT CU ID-UL DAT
ORA-06512: at "UTILIZATOR.TOP_MEDICI_DEPARTAMENT", line 49
ORA-06512: at line 1
10000. 00000 - "%"
Cause:  The stored procedure 'raise_application_error'
        was called which causes this error to be generated.
Action: Correct the problem as described in the error message or contact
        the application administrator or DBA for more information.

```

CREATE OR REPLACE PROCEDURE TOP_MEDICI_DEPARTAMENT(V_DEP DEPARTAMENTE.ID_DEPARTAMENT%TYPE)

AS

TYPE ID_MEDICI IS TABLE OF MEDICI.ID_MEDIC%TYPE INDEX BY PLS_INTEGER;

V_MEDICI ID_MEDICI;

V_NUMAR INT;

```
V_CURSOR INT :=0;

V_TEST DEPARTAMENTE.ID_DEPARTAMENT%TYPE;

CURSOR C1 IS

SELECT M.ID_MEDIC ID_MEDIC,M.NUME NUME , M.PRENUME PRENUME, D.DENUMIRE DEPARTAMENT, M.SALARIU SALARIU
FROM MEDICI M
JOIN DEPARTAMENTE D ON M.ID_DEPARTAMENT = D.ID_DEPARTAMENT
WHERE M.SALARIU IN (
    SELECT MAX(SALARIU)
    FROM MEDICI
    WHERE ID_DEPARTAMENT = V_DEP
)
AND M.ID_DEPARTAMENT=V_DEP;

CURSOR C2 (V_ID MEDICI.ID_MEDIC%TYPE) IS
SELECT COUNT(C.ID_CONSULTATIE) NUMAR_CONSULTATII
FROM MEDICI M
LEFT JOIN CONSULTATII C ON C.ID_MEDIC = M.ID_MEDIC
HAVING M.ID_MEDIC = V_ID
GROUP BY M.ID_MEDIC;

BEGIN
    SELECT ID_DEPARTAMENT
    INTO V_TEST
    FROM DEPARTAMENTE
    WHERE ID_DEPARTAMENT = V_DEP;
    FOR I IN C1 LOOP
        V_CURSOR:=V_CURSOR+1;
        OPEN C2(I.ID_MEDIC);
        FETCH C2 INTO V_NUMAR;
        EXIT WHEN C2%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('MEDICUL'||I.NUME||' '|I.PRENUME||' CU SALARIUL IN VALOARE DE'||I.SALARIU||' DIN
DEPARTAMENTUL'||I.DEPARTAMENT||' ARE UN NUMAR DE'||V_NUMAR||' CONSULTATII');
        CLOSE C2;
    END LOOP;
```

```
IF V_CURSOR = 0 THEN
    DBMS_OUTPUT.PUT_LINE('NU EXISTA MEDICI IN ACEST DEPARTAMENT');
END IF;

EXCEPTION
    WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20000, 'NU EXISTA DEPARTAMENT CU ID-UL DAT');

END;

EXECUTE TOP_MEDICI_DEPARTAMENT(11);
EXECUTE TOP_MEDICI_DEPARTAMENT(5);
EXECUTE TOP_MEDICI_DEPARTAMENT(23);
```

EXERCIȚIU 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 exceptii proprii. Apelați subprogramul astfel încât să evidențiați toate cazurile definite și tratate.

CERINȚĂ: La introducerea unui departament și a unei valori minime și a uneia maxime să se întoarcă specializarea departamentului, dacă numărul de consultații din acest departament se încadrează în acest interval.

The screenshot shows the Oracle SQL Developer interface with the following details:

- Code Area:** Contains the PL/SQL code for creating a function named `NUMAR_CONSULTATII`. The code includes exception handling for negative numbers and intervals, and it uses a cursor to select specialization counts from a department table.
- Status Bar:** Shows "Task completed in 0.067 seconds".
- Output Area:** Displays the message "Function NUMAR_CONSULTATII compiled".

```

    EXCEPTION
        WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20000,'NU EXISTA DEP');
        WHEN NUMAR_NEGATIV THEN RAISE_APPLICATION_ERROR(-20001,'CAPETELE DE IN');
        WHEN NU_APARTINE_INTERVAL THEN RAISE_APPLICATION_ERROR(-20002,'NUMARUL');
        WHEN ORDINE_GRESITA_CAPETE THEN RAISE_APPLICATION_ERROR(-20003,'CAPETI');
    END NUMAR_CONSULTATII;

    SELECT NUMAR_CONSULTATII(100, 0, 5) FROM DUAL;

```

Script Output | Query Result | SQL | Executing:SELECT NUMAR_CONSULTATII(100, 0, 5) FROM DUAL in 0 seconds

ORA-20000: NU EXISTA DEPARTAMENT CU ID-UL DAT
 ORA-06512: at "UTILIZATOR.NUMAR_CONSULTATII", line 44
 20000. 00000 - "%s"
 *Cause: The stored procedure 'raise_application_error'
 was called which causes this error to be generated.
 *Action: Correct the problem as described in the error message or contact
 the application administrator or DBA for more information.

Worksheet | Query Builder

```

    IF V_NUMAR<MINI OR V_NUMAR>MAXI THEN
        RAISE NU_APARTINE_INTERVAL;
    END IF;

    RETURN V_DEPT;

    EXCEPTION
        WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20000,'NU EXISTA DEPARTAM');
        WHEN NUMAR_NEGATIV THEN RAISE_APPLICATION_ERROR(-20001,'CAPETELE DE INTERV');
        WHEN NU_APARTINE_INTERVAL THEN RAISE_APPLICATION_ERROR(-20002,'NUMARUL DE');
        WHEN ORDINE_GRESITA_CAPETE THEN RAISE_APPLICATION_ERROR(-20003,'CAPETELE D');
    END NUMAR_CONSULTATII;

    SELECT NUMAR_CONSULTATII(1, -5, 10) FROM DUAL;

```

Script Output | Query Result | SQL | Executing:SELECT NUMAR_CONSULTATII(1, -5, 10) FROM DUAL in 0 seconds

ORA-20001: CAPETELE DE INTERVAL TREBUIE SA FIE NUMERE POZITIVE
 ORA-06512: at "UTILIZATOR.NUMAR_CONSULTATII", line 45

Worksheet | Query Builder

```

    IF V_NUMAR<MINI OR V_NUMAR>MAXI THEN
        RAISE NU_APARTINE_INTERVAL;
    END IF;

    RETURN V_DEPT;

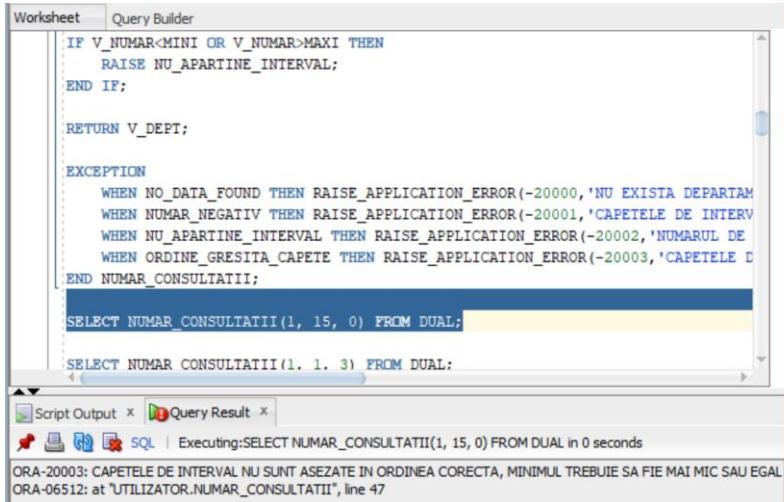
    EXCEPTION
        WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20000,'NU EXISTA DEPARTAM');
        WHEN NUMAR_NEGATIV THEN RAISE_APPLICATION_ERROR(-20001,'CAPETELE DE INTERV');
        WHEN NU_APARTINE_INTERVAL THEN RAISE_APPLICATION_ERROR(-20002,'NUMARUL DE');
        WHEN ORDINE_GRESITA_CAPETE THEN RAISE_APPLICATION_ERROR(-20003,'CAPETELE D');
    END NUMAR_CONSULTATII;

    SELECT NUMAR_CONSULTATII(1, 1, 3) FROM DUAL;

```

Script Output | Query Result | SQL | Executing:SELECT NUMAR_CONSULTATII(1, 1, 3) FROM DUAL in 0 seconds

ORA-20002: NUMARUL DE CONSULTATII DIN ACEST DEPARTAMENT NU SE INCADREAZA IN INTERVALUL INTRODUS
 ORA-06512: at "UTILIZATOR.NUMAR_CONSULTATII", line 46



```

Worksheet | Query Builder
IF V_NUMAR<MINI OR V_NUMAR>MAXI THEN
    RAISE NU_APARTINE_INTERVAL;
END IF;

RETURN V_DEPT;

EXCEPTION
    WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20000,'NU EXISTA DEPARTAMENT');
    WHEN NUMAR_NEGATIV THEN RAISE_APPLICATION_ERROR(-20001,'CAPELELE DE INTERVAL');
    WHEN NU_APARTINE_INTERVAL THEN RAISE_APPLICATION_ERROR(-20002,'NUMARUL DE
    WHEN ORDINE_GRESITA_CAPETE THEN RAISE_APPLICATION_ERROR(-20003,'CAPELELE D
END NUMAR_CONSULTATII;

SELECT NUMAR_CONSULTATII(1, 15, 0) FROM DUAL;

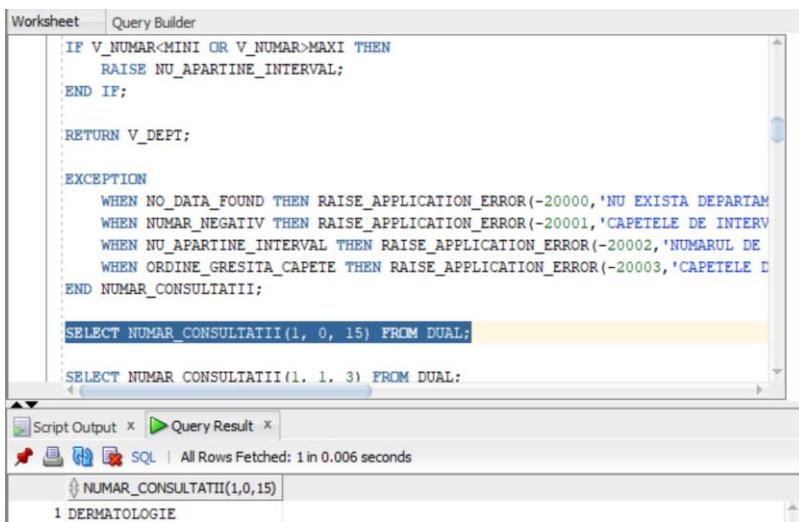
SELECT NUMAR_CONSULTATII(1, 1, 3) FROM DUAL;

```

Script Output | Query Result

SQL | Executing: SELECT NUMAR_CONSULTATII(1, 15, 0) FROM DUAL in 0 seconds

ORA-20003: CAPELELE DE INTERVAL NU SUNT ASEZATE IN ORDINEA CORECTA, MINIMUL TREBUIE SA FIE MAI MIC SAU EGAL
 ORA-06512: at "UTILIZATOR.NUMAR_CONSULTATII", line 47



```

Worksheet | Query Builder
IF V_NUMAR<MINI OR V_NUMAR>MAXI THEN
    RAISE NU_APARTINE_INTERVAL;
END IF;

RETURN V_DEPT;

EXCEPTION
    WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20000,'NU EXISTA DEPARTAMENT');
    WHEN NUMAR_NEGATIV THEN RAISE_APPLICATION_ERROR(-20001,'CAPELELE DE INTERVAL');
    WHEN NU_APARTINE_INTERVAL THEN RAISE_APPLICATION_ERROR(-20002,'NUMARUL DE
    WHEN ORDINE_GRESITA_CAPETE THEN RAISE_APPLICATION_ERROR(-20003,'CAPELELE D
END NUMAR_CONSULTATII;

SELECT NUMAR_CONSULTATII(1, 0, 15) FROM DUAL;

SELECT NUMAR_CONSULTATII(1, 1, 3) FROM DUAL;

```

Script Output | Query Result

SQL | All Rows Fetched: 1 in 0.006 seconds

NUMAR_CONSULTATII(1,0,15)
1 DERMATOLOGIE

CREATE OR REPLACE FUNCTION NUMAR_CONSULTATII(V_ID DEPARTAMENTE.ID_DEPARTAMENT%TYPE, MINI NUMBER, MAXI NUMBER)

RETURN DEPARTAMENTE.SPECIALIZARE%TYPE

IS

NUMAR_NEGATIV EXCEPTION;

NU_APARTINE_INTERVAL EXCEPTION;

ORDINE_GRESITA_CAPETE EXCEPTION;

V_DEPT DEPARTAMENTE.SPECIALIZARE%TYPE;

V_NUMAR NUMBER;

V_SPECIALIZARE DEPARTAMENTE.SPECIALIZARE%TYPE;

BEGIN

IF MINI<0 OR MAXI<0 THEN

RAISE NUMAR_NEGATIV;

Proiect Sisteme de Gestiune a Bazelor de Date, anul 2, semestrul 1
Berca Teodora

```
END IF;

SELECT SPECIALIZARE

INTO V_DEPT

FROM DEPARTAMENTE

WHERE ID_DEPARTAMENT = V_ID;

SELECT COUNT(C.ID_CONSULTATIE)

INTO V_NUMAR

FROM DEPARTAMENTE D

JOIN MEDICI M ON M.ID_DEPARTAMENT = D.ID_DEPARTAMENT

JOIN CONSULTATII C ON C.ID_MEDIC = M.ID_MEDIC

WHERE D.ID_DEPARTAMENT = V_ID;

IF MINI > MAXI THEN

    RAISE ORDINE_GRESITA_CAPETE;

END IF;

IF V_NUMAR<MINI OR V_NUMAR>MAXI THEN

    RAISE NU_APARTINE_INTERVAL;

END IF;

RETURN V_DEPT;

EXCEPTION

    WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20000,'NU EXISTA DEPARTAMENT CU ID-UL DAT');

    WHEN NUMAR_NEGATIV THEN RAISE_APPLICATION_ERROR(-20001,'CAPETELE DE INTERVAL TREBUIE SA FIE NUMERE POZITIVE');

    WHEN NU_APARTINE_INTERVAL THEN RAISE_APPLICATION_ERROR(-20002,'NUMARUL DE CONSULTATII DIN ACEST DEPARTAMENT NU SE INCADREAZA IN INTERVALUL INTRODUS');

    WHEN ORDINE_GRESITA_CAPETE THEN RAISE_APPLICATION_ERROR(-20003,'CAPETELE DE INTERVAL NU SUNT ASEZATE IN ORDINEA CORECTA, MINIMUL TREBUIE SA FIE MAI MIC SAU EGAL CU MAXIMUL');

END NUMAR_CONSULTATII;

SELECT NUMAR_CONSULTATII(100, 0, 5) FROM DUAL;

SELECT NUMAR_CONSULTATII(1, -5, 10) FROM DUAL;

SELECT NUMAR_CONSULTATII(1, 1, 3) FROM DUAL;

SELECT NUMAR_CONSULTATII(1, 15, 0) FROM DUAL;

SELECT NUMAR_CONSULTATII(1, 0, 15) FROM DUAL;
```

EXERCITIU 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 excepțiile care pot apărea, inclusiv excepțiile NO_DATA_FOUND și TOO_MANY_ROWS. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

CERINȚĂ: Dat fiind numele unui medic să se afișeze istoricul său de consultații, cu următoarele date: numele și prenumele pacientului, data la care s-a efectuat consultatia, numarul cabinetului și denumirea diagnosticului primit, fiind afisate doar consultațiile facute pentru pacienti cu varsta mai mica de 50 de ani.

OBSERVAȚIE: Pentru a putea exemplifica excepția TOO_MANY_ROWS am adăugat în tabela de medici o nouă linie.

ID_MEDIC	NUME	PRENUME	DATA_NASTERE	NUMAR_TELEFON	GRAD	SALARIU	ID_DEPARTAMENT
1	24 Aleca	Catalina	23-MAY-80	0715545678	Primar	6500	1

```
INSERT INTO MEDICI(ID_MEDIC, NUME, PRENUME, DATA_NASTERE, NUMAR_TELEFON, GRAD, SALARIU, ID_DEPARTAMENT)
VALUES (24,'Aleca','Catalina',TO_DATE('1980-05-23','YYYY-MM-DD'),'0715545678','Primar',6500,1);
```

The screenshot shows the Oracle SQL Developer interface. The top window is titled 'Worksheet' and contains the PL/SQL code for creating a procedure named 'CONSULTATII_MEDIC'. The code defines a record type 'DATE_CONSULTATIE' with fields: NUME_PACIENT, PRENUME_PACIENT, DATA_CONSULTATIE, NUMAR_CABINET, and DENUMIRE_DIAGNOSTIC. It also defines a table type 'TABLOU' based on 'DATE_CONSULTATIE'. The bottom window is titled 'Script Output' and displays the message 'Procedure CONSULTATII_MEDIC compiled'.

```
CREATE OR REPLACE PROCEDURE CONSULTATII_MEDIC (V_NUME_MEDICI.NUME%TYPE)
AS
  TYPE DATE_CONSULTATIE IS RECORD (
    NUME_PACIENT PACIENTI.NUME%TYPE,
    PRENUME_PACIENT PACIENTI.PRENUME%TYPE,
    DATA_CONSULTATIE CONSULTATII.DATA_CONSULTATIE%TYPE,
    NUMAR_CABINET CABINETE.NUMAR%TYPE,
    DENUMIRE_DIAGNOSTIC DIAGNOSTICE.DENUMIRE%TYPE
  );
  TYPE TABLOU IS TABLE OF DATE_CONSULTATIE INDEX BY PLS_INTEGER;

```

Procedure CONSULTATII_MEDIC compiled

Proiect Sisteme de Gestiune a Bazelor de Date, anul 2, semestrul 1
 Berca Teodora

```

        ISTORIC_CONSULTATII(I).PRENUME_PACIENT || ''
        'DATA CONSULTATIE: ' ||
        ISTORIC_CONSULTATII(I).DATA_CONSULTATIE || ''
        'NUMAR CABINET ' ||
        ISTORIC_CONSULTATII(I).NUMAR_CABINET || '' ||
        'DIAGNOSTIC ' ||
        ISTORIC_CONSULTATII(I).DENUMIRE_DIAGNOSTIC
    );
END LOOP;

EXCEPTION
    WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20000, 'NU EXISTA MED');
    WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR(-20001, 'EXISTA MAI MULT DE 1 MEDIC');
    WHEN OTHERS THEN RAISE_APPLICATION_ERROR(-20002, 'ALTA EROARE');
END;

EXECUTE CONSULTATII_MEDIC('ALECA');
EXECUTE CONSULTATII_MEDIC('BDESCU');
EXECUTE CONSULTATII_MEDIC('OCTAVIAN');

```

Script Output X | Task completed in 0.054 seconds

Error starting at line : 248 in command -
 BEGIN CONSULTATII_MEDIC('ALECA'); END;
 Error report -
 ORA-20001: EXISTA MAI MULTII MEDICI CU NUMELE DAT
 ORA-06512: at "UTILIZATOR.CONSULTATII_MEDIC", line 49
 ORA-06512: at line 1

Worksheet | Query Builder | project *

```

        'DIAGNOSTIC ' ||
        ISTORIC_CO
    );
END LOOP;

EXCEPTION
    WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20000, 'NU EXISTA MED');
    WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR(-20001, 'EXISTA MAI MULT DE 1 MEDIC');
    WHEN OTHERS THEN RAISE_APPLICATION_ERROR(-20002, 'ALTA EROARE');
END;

EXECUTE CONSULTATII_MEDIC('ALECA');
EXECUTE CONSULTATII_MEDIC('BDESCU');
EXECUTE CONSULTATII_MEDIC('OCTAVIAN');

```

Script Output X | Query Result X | Task completed in 0.03 seconds

PL/SQL procedure successfully completed.

project *

NUME PACIENT: Lazar PRENUME PACIENT: Horatiu DATA CONSULTATIE: 06-JAN-23 NUMAR CABINET 2 DIAGNOSTIC Acne
 NUME PACIENT: Stanciu PRENUME PACIENT: Silviu DATA CONSULTATIE: 06-JAN-23 NUMAR CABINET 2 DIAGNOSTIC Dermatita
 NUME PACIENT: Anghel PRENUME PACIENT: Mihnea DATA CONSULTATIE: 06-JAN-23 NUMAR CABINET 2 DIAGNOSTIC Dermatita
 NUME PACIENT: Anghel PRENUME PACIENT: Mihnea DATA CONSULTATIE: 07-JAN-23 NUMAR CABINET 2 DIAGNOSTIC Dermatita

The screenshot shows a PL/SQL script being run in Oracle SQL Developer. The script includes a loop, exception handling for specific errors, and three execute statements. The third execute statement, which attempts to call a procedure named 'CONSULTATII_MEDIC' with the argument 'OCTAVIAN', results in an ORA-20000 error because the procedure does not exist. The error message is displayed in the 'Script Output' tab.

```
'DIAGNOSTIC'||  
ISTORIC_CONSULTATII(I).DENUMIRE_DIAGNOSTIC  
);  
END LOOP;  
  
EXCEPTION  
WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20000, 'NU EXISTA MEDICI CU N  
WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR(-20001, 'EXISTA MAI MULTI MED  
WHEN OTHERS THEN RAISE_APPLICATION_ERROR(-20002,'ALTA EROARE');  
END;  
  
EXECUTE CONSULTATII_MEDIC('ALECA');  
EXECUTE CONSULTATII_MEDIC('BADESCU');  
EXECUTE CONSULTATII_MEDIC('OCTAVIAN');
```

Script Output | Query Result | Task completed in 0.024 seconds

```
BEGIN CONSULTATII_MEDIC('OCTAVIAN'); END;  
Error report -  
ORA-20000: NU EXISTA MEDICI CU NUMELE DAT  
ORA-06512: at "UTILIZATOR.CONSULTATII_MEDIC", line 48  
ORA-06512: at line 1  
20000. 00000 - "%s"  
*Cause: The stored procedure 'raise_application_error'  
was called which causes this error to be generated.  
*Action: Correct the problem as described in the error message or contact  
the application administrator or DBA for more information.
```

CREATE OR REPLACE PROCEDURE CONSULTATII_MEDIC (V_NUME_MEDICI.NUME%TYPE)

AS

```
TYPE DATE_CONSULTATIE IS RECORD(  
    NUME_PACIENT PACIENTI.NUME%TYPE,  
    PRENUME_PACIENT PACIENTI.PRENUME%TYPE,  
    DATA_CONSULTATIE CONSULTATII.DATA_CONSULTATIE%TYPE,  
    NUMAR_CABINET CABINETE.NUMAR%TYPE,  
    DENUMIRE_DIAGNOSTIC DIAGNOSTICE.DENUMIRE%TYPE  
);
```

TYPE TABLOU IS TABLE OF DATE_CONSULTATIE INDEX BY PLS_INTEGER;

ISTORIC_CONSULTATII TABLOU;

V_EROARE PACIENTI.NUME%TYPE;

BEGIN

```
SELECT NUME  
INTO V_EROARE  
FROM MEDICI
```

Proiect Sisteme de Gestiune a Bazelor de Date, anul 2, semestrul 1
Berca Teodora

```
WHERE UPPER(NUME)=UPPER(V_NUME);

SELECT P.NUME, P.PRENUME, C.DATA_CONSULTATIE, CAB.NUMAR, D.DENUMIRE

BULK COLLECT INTO ISTORIC_CONSULTATII

FROM MEDICI M

JOIN CONSULTATII C ON C.ID_MEDIC = M.ID_MEDIC

JOIN PACIENTI P ON P.ID_PACIENT = C.ID_PACIENT

JOIN DIAGNOSTICE D ON D.ID_DIAGNOSTIC = C.ID_DIAGNOSTIC

JOIN CABINETE CAB ON CAB.ID_CABINET = C.ID_CABINET

WHERE UPPER(V_NUME)=UPPER(M.NUME) AND EXTRACT(YEAR FROM SYSDATE)-EXTRACT(YEAR FROM
P.DATA_NASTERE)<=50;

FOR I IN ISTORIC_CONSULTATII.FIRST..ISTORIC_CONSULTATII.LAST LOOP

DBMS_OUTPUT.PUT_LINE(

'NUME PACIENT: ' ||

ISTORIC_CONSULTATII(I).NUME_PACIENT || '' ||

'PRENUME PACIENT: ' ||

ISTORIC_CONSULTATII(I).PRENUME_PACIENT || '' ||

'DATA CONSULTATIE:' ||

ISTORIC_CONSULTATII(I).DATA_CONSULTATIE || '' ||

'NUMAR CABINET'||

ISTORIC_CONSULTATII(I).NUMAR_CABINET || '' ||

'DIAGNOSTIC '| |

ISTORIC_CONSULTATII(I).DENUMIRE_DIAGNOSTIC

);

END LOOP;

EXCEPTION

WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20000, 'NU EXISTA MEDICI CU NUMELE DAT');

WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR(-20001, 'EXISTA MAI MULTI MEDICI CU NUMELE DAT');

WHEN OTHERS THEN RAISE_APPLICATION_ERROR(-20002,'ALTA EROARE');

END;

EXECUTE CONSULTATII_MEDIC('ALECA');

EXECUTE CONSULTATII_MEDIC('BADESCU');

EXECUTE CONSULTATII_MEDIC('OCTAVIAN');
```

EXERCIȚIU 10:

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

CERINȚĂ: Programările la clinică se pot înregistra doar în timpul programului de lucru, adică în intervalul 8:00-17:00.

The screenshot shows two code editors in Oracle SQL Developer. The top editor contains the trigger definition:

```
--EXERCITIU 10
CREATE OR REPLACE TRIGGER INTERVAL_PROGRAMARI
BEFORE INSERT OR DELETE OR UPDATE ON CONSULTATII
BEGIN
    IF (TO_CHAR(SYSDATE,'HH24') NOT BETWEEN 8 AND 17)
    THEN
        RAISE_APPLICATION_ERROR(-20001,'PROGRAMARILE SE FAC DOAR IN INTERVALUL 8-17');
    END IF;
END;
```

The bottom editor contains the function definition and the trigger definition:

```
Function NUMAR_CONSULTATII compiled

Trigger INTERVAL_PROGRAMARI compiled

CREATE OR REPLACE TRIGGER INTERVAL_PROGRAMARI
BEFORE INSERT OR DELETE OR UPDATE ON CONSULTATII
BEGIN
    IF (TO_CHAR(SYSDATE,'HH24') NOT BETWEEN 8 AND 17)
    THEN
        RAISE_APPLICATION_ERROR(-20001,'PROGRAMARILE SE FAC DOAR IN INTERVALUL 8-17');
    END IF;
END;

INSERT INTO CONSULTATII(ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL, DATA_CONSULTATIE, ID_TIP_CONSULTATIE) VALUES (40, 1, 1, 1, 9, 1, TO_DATE('2024/01/08 08:30','YYYY/MM/DD HH:MI'));
```

The output panes show:

- Script Output: Task completed in 0.048 seconds
- Query Result: Function NUMAR_CONSULTATII compiled
- Script Output: Task completed in 0.075 seconds
- Query Result: Error starting at line : 131 in command -
INSERT INTO CONSULTATII(ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL, DATA_CONSULTATIE, ID_TIP_CONSULTATIE) VALUES (40, 1, 1, 1, 9, 1, TO_DATE('2024/01/08 08:30','YYYY/MM/DD HH:MI'))
Error report -
ORA-20001: PROGRAMARILE SE FAC DOAR IN INTERVALUL 8-17
ORA-06512: at "UTILIZATOR.INTERVAL_PROGRAMARI", line 4
ORA-04088: error during execution of trigger 'UTILIZATOR.INTERVAL_PROGRAMARI'

```
CREATE OR REPLACE TRIGGER INTERVAL_PROGRAMARI
BEFORE INSERT OR DELETE OR UPDATE ON CONSULTATII
BEGIN
    IF (TO_CHAR(SYSDATE,'HH24') NOT BETWEEN 8 AND 17)
    THEN
        RAISE_APPLICATION_ERROR(-20001,'PROGRAMARILE SE FAC DOAR IN INTERVALUL 8-17');
    END IF;
```

END;

```
INSERT INTO CONSULTATII(ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL,  
DATA_CONSULTATIE)
```

```
VALUES (40, 1, 1, 1, 9, 1, TO_DATE('2024/01/08 08:30','YYYY/MM/DD HH:MI'));
```

EXERCIȚIU 11:

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

CERINȚĂ: În momentul în care se adaugă sau se actualizează o consultație, trebuie să se verifice dacă medicul și cabinetul aparțin de același departament.

```
--EXERCITIU 11  
CREATE OR REPLACE TRIGGER VERIFICA_PROGRAMARE  
BEFORE INSERT OR UPDATE ON CONSULTATII  
FOR EACH ROW  
DECLARE  
V_AUX1 MEDICI.ID_DEPARTAMENT%TYPE;  
V_AUX2 CABINETE.ID_DEPARTAMENT%TYPE;  
  
BEGIN  
  
SELECT ID_DEPARTAMENT  
INTO V_AUX1  
FROM MEDICI  
WHERE ID_MEDIC IN (:NEW.ID_MEDIC);  
  
SELECT ID_DEPARTAMENT  
INTO V_AUX2  
FROM CABINETE  
WHERE ID_CABINET IN (:NEW.ID_CABINET);  
  
IF V_AUX1 <> V_AUX2 THEN  
RAISE APPLICATION ERROR(-20000, 'MEDICUL SI CABINETUL TREBUIE SA APARTINA DE ACELASI DEPARTAMENT');  
END IF;  
END;
```

Trigger VERIFICA_PROGRAMARE compiled

```
BEGIN  
SELECT ID_DEPARTAMENT  
INTO V_AUX1  
FROM MEDICI  
WHERE ID_MEDIC IN (:NEW.ID_MEDIC);  
  
SELECT ID_DEPARTAMENT  
INTO V_AUX2  
FROM CABINETE  
WHERE ID_CABINET IN (:NEW.ID_CABINET);  
  
IF V_AUX1 <> V_AUX2 THEN  
RAISE_APPLICATION_ERROR(-20000, 'MEDICUL SI CABINETUL TREBUIE SA APARTINA DE ACELASI DEPARTAMENT');  
END IF;  
  
END;  
  
--AICI MEDICUL SI CABINETUL NU APARTIN DE ACELASI DEPARTAMENT  
INSERT INTO CONSULTATII(ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL, DATA_CONSULTATIE)  
VALUES(39, 1, 5, 2, 10, 2, TO_DATE('2023/01/05 08:30','YYYY/MM/DD HH:MI'));
```

Query Result | Script Output | Task completed in 0.03 seconds

```
ORA-20000: MEDICUL SI CABINETUL TREBUIE SA APARTINA DE ACELASI DEPARTAMENT  
ORA-06512: at "UTILIZATOR.VERIFICA_PROGRAMARE", line 18  
ORA-04088: error during execution of trigger 'UTILIZATOR.VERIFICA_PROGRAMARE'
```

Proiect Sisteme de Gestiune a Bazelor de Date, anul 2, semestrul 1
Berca Teodora

```
CREATE OR REPLACE TRIGGER VERIFICA_PROGRAMARE
BEFORE INSERT OR UPDATE ON CONSULTATII
FOR EACH ROW
DECLARE
V_AUX1 MEDICI.ID_DEPARTAMENT%TYPE;
V_AUX2 CABINETE.ID_DEPARTAMENT%TYPE;
BEGIN
SELECT ID_DEPARTAMENT
INTO V_AUX1
FROM MEDICI
WHERE ID_MEDIC IN (:NEW.ID_MEDIC);
SELECT ID_DEPARTAMENT
INTO V_AUX2
FROM CABINETE
WHERE ID_CABINET IN (:NEW.ID_CABINET);
IF V_AUX1 <> V_AUX2 THEN
RAISE_APPLICATION_ERROR(-20000, 'MEDICUL SI CABINETUL TREBUIE SA APARTINA DE ACELASI DEPARTAMENT');
END IF;
END;
```

INSERT INTO CONSULTATII(ID_CONSULTATIE, ID_CABINET, ID_MEDIC, ID_DIAGNOSTIC, ID_PACIENT, ID_ASISTENT_MEDICAL, DATA_CONSULTATIE)

VALUES(39, 1, 5, 2, 10, 2, TO_DATE('2023/01/05 08:30','YYYY/MM/DD HH:MI'));

EXERCIȚIU 12:

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

CERINȚĂ: Să se creeze o tabelă istoric bază de date în care să se înregistreze informații cu privire la acțiunile ce se fac asupra bazei de date : cine face acțiunea (utilizatorul), ce acțiune face (CREATE, ALTER, DROP), ce tip are, ce nume îi dă și la ce dată a fost făcută acțiunea.

OBSERVAȚII: Pentru rezolvarea acestei cerințe am creat următoarea tabelă:

The screenshot shows three separate code snippets in the SQL Developer interface:

- Top Snippet:** A CREATE TABLE statement for "ISTORIC_BAZA_DE_DATE" with columns: UTILIZATOR (Varchar2(50)), EVENIMENT (Varchar2(20)), TIP (Varchar2(30)), NUME (Varchar2(30)), and DATA_MODIFICARE (Timestamp(3)). It ends with a semicolon.
- Middle Snippet:** A CREATE OR REPLACE TRIGGER statement named "ISTORIC". It triggers after CREATE, DROP, or ALTER on any schema object. The trigger body begins with BEGIN, followed by an INSERT INTO statement for the "ISTORIC_BAZA_DE_DATE" table, specifying columns: SYS.LOGIN_USER, SYS.SYSEVENT, SYS.DICTIONARY_OBJ_TYPE, SYS.DICTIONARY_OBJ_NAME, and SYSTIMESTAMP(3). It ends with END;.
- Bottom Snippet:** A CREATE TABLE statement for "EXEMPLU" with columns: EXEMPLU1 (Varchar2(10)) and EXEMPLU2 (Varchar2(10)). It ends with a semicolon.

Below each snippet, the "Script Output" tab shows the results of the execution:

- For the first snippet: "Table ISTORIC_BAZA_DE_DATE created."
- For the second snippet: "Trigger ISTORIC compiled"
- For the third snippet: "Table EXEMPLU created."

```
DROP TABLE EXEMPLU;

Table ISTORIC_BAZA_DE_DATE created.

Trigger ISTORIC compiled

Table EXEMPLU created.

Table EXEMPLU dropped.
```

```
CREATE TABLE EXEMPLU2
(
    EXEMPLU1 VARCHAR2(10),
    EXEMPLU2 VARCHAR2(10)
);

ALTER TABLE EXEMPLU2
ADD EXEMPLU3 VARCHAR2(10);

Table EXEMPLU dropped.

Table EXEMPLU2 created.
```

```
ALTER TABLE EXEMPLU2
ADD EXEMPLU3 VARCHAR2(10);

Table EXEMPLU2 created.

Table EXEMPLU2 altered.
```

UTILIZATOR	EVENIMENT	TIP	NUME	DATA_MODIFICARE
1 UTILIZATOR	CREATE	TABLE EXEMPLU	09-JAN-24 11.40.07.021000000 AM	
2 UTILIZATOR	DROP	TABLE EXEMPLU	09-JAN-24 11.40.46.095000000 AM	
3 UTILIZATOR	CREATE	TABLE EXEMPLU2	09-JAN-24 11.46.33.977000000 AM	
4 UTILIZATOR	ALTER	TABLE EXEMPLU2	09-JAN-24 11.47.00.136000000 AM	

CREATE TABLE ISTORIC_BAZA_DE_DATE

```
(  
    UTILIZATOR VARCHAR2(50),  
    EVENIMENT VARCHAR2(20),  
    TIP VARCHAR2(30),  
    NUME VARCHAR2(30),  
    DATA_MODIFICARE TIMESTAMP(3)  
);  
  
CREATE OR REPLACE TRIGGER ISTORIC  
AFTER CREATE OR DROP OR ALTER ON SCHEMA  
BEGIN  
    INSERT INTO ISTORIC_BAZA_DE_DATE  
    VALUES(SYS.LOGIN_USER, SYS.SYSEVENT, SYS.DICTIONARY_OBJ_TYPE,SYS.DICTIONARY_OBJ_NAME, SYSTIMESTAMP(3));  
END;  
  
CREATE TABLE EXEMPLU  
(  
    EXEMPLU1 VARCHAR2(10),  
    EXEMPLU2 VARCHAR2(10)  
);  
DROP TABLE EXEMPLU;  
  
CREATE TABLE EXEMPLU2  
(  
    EXEMPLU1 VARCHAR2(10),  
    EXEMPLU2 VARCHAR2(10)  
);  
ALTER TABLE EXEMPLU2  
ADD EXEMPLU3 VARCHAR2(10);
```

EXERCITIUL 13:

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

The screenshot shows two panels from Oracle SQL Developer. The top panel displays the package specification (CREATE OR REPLACE PACKAGE PACHET AS ... END PACHET;). The bottom panel shows the package body (CREATE OR REPLACE PACKAGE BODY PACHET AS ... END PACHET;). Both panels include a 'Script Output' tab at the bottom with a message indicating the task completed in approximately 0.04 seconds.

```
CREATE OR REPLACE PACKAGE PACHET AS
  PROCEDURE ISTORIC_PACIENT(V_ID PACIENTI.ID_PACIENT%TYPE);
  PROCEDURE TOP_MEDICI_DEPARTAMENT(V_DEP DEPARTAMENTE.ID_DEPARTAMENT%TYPE);
  FUNCTION NUMAR_CONSULTATII(V_ID DEPARTAMENTE.ID_DEPARTAMENT%TYPE,MINI NUMBER,MAXI NUMBER) RETURN DEPARTAMENTE.SPECIALIZARE%TYPE;
  PROCEDURE CONSULTATII_MEDIC(V_NUME MEDICI.NUME%TYPE);
END PACHET;

CREATE OR REPLACE PACKAGE BODY PACHET AS
  PROCEDURE ISTORIC_PACIENT (V_ID PACIENTI.ID_PACIENT%TYPE)
  AS
    TYPE DATE_ANALIZE IS RECORD(
      TIP_ANALIZA TIPURI_ANALIZE.DENUMIRE%TYPE,
      REZULTAT REZULTATE.VALOARE%TYPE,
      INTERPRETARE REZULTATE.INTERPRETARE%TYPE
    );
    TYPE SET_REZULTATE IS VARRAY(6) OF DATE_ANALIZE;
    TYPE COLECTIE_SET_REZULTATE IS TABLE OF SET_REZULTATE;
    V_COLECTIE_SET_REZULTATE COLECTIE_SET_REZULTATE := COLECTIE_SET_REZULTATE();
```

```
CREATE OR REPLACE PACKAGE PACHET AS

  PROCEDURE ISTORIC_PACIENT(V_ID PACIENTI.ID_PACIENT%TYPE);

  PROCEDURE TOP_MEDICI_DEPARTAMENT(V_DEP DEPARTAMENTE.ID_DEPARTAMENT%TYPE);

  FUNCTION NUMAR_CONSULTATII(V_ID DEPARTAMENTE.ID_DEPARTAMENT%TYPE,MINI NUMBER,MAXI NUMBER)
  RETURN DEPARTAMENTE.SPECIALIZARE%TYPE;

  PROCEDURE CONSULTATII_MEDIC(V_NUME MEDICI.NUME%TYPE);

END PACHET;
```

```
CREATE OR REPLACE PACKAGE BODY PACHET AS

  PROCEDURE ISTORIC_PACIENT (V_ID PACIENTI.ID_PACIENT%TYPE)
  AS
    TYPE DATE_ANALIZE IS RECORD(
      TIP_ANALIZA TIPURI_ANALIZE.DENUMIRE%TYPE,
      REZULTAT REZULTATE.VALOARE%TYPE,
      INTERPRETARE REZULTATE.INTERPRETARE%TYPE
    );
    TYPE SET_REZULTATE IS VARRAY(6) OF DATE_ANALIZE;
```

```
TYPE COLECTIE_SET_RESULTATE IS TABLE OF SET_RESULTATE;
V_COLECTIE_SET_RESULTATE COLECTIE_SET_RESULTATE := COLECTIE_SET_RESULTATE();
TYPE ISTORIC_DIAGNOSTICE IS TABLE OF DIAGNOSTICE%ROWTYPE;
V_ISTORIC_DIAGNOSTICE ISTORIC_DIAGNOSTICE;
TYPE ISTORIC_RECOLTARI IS TABLE OF RECOLTARI%ROWTYPE INDEX BY PLS_INTEGER;
V_ISTORIC_RECOLTARI ISTORIC_RECOLTARI;
V_ULTIMA_RECOLTARE RECOLTARI.ID_RECOLTARE%TYPE;
V_NUME PACIENTI.NUME%TYPE;
V_PRENUME PACIENTI.PRENUME%TYPE;
BEGIN
SELECT NUME
INTO V_NUME
FROM PACIENTI
WHERE ID_PACIENT = V_ID;
SELECT PRENUME
INTO V_PRENUME
FROM PACIENTI
WHERE ID_PACIENT = V_ID;
DBMS_OUTPUT.PUT_LINE('NUME: '||V_NUME||' PRENUME: '||V_PRENUME);
SELECT D.ID_DIAGNOSTIC, D.DENUMIRE, D.TRATAMENT, D.SIMPTOME
BULK COLLECT INTO V_ISTORIC_DIAGNOSTICE
FROM DIAGNOSTICE D
JOIN CONSULTATII C ON C.ID_DIAGNOSTIC = D.ID_DIAGNOSTIC
JOIN PACIENTI P ON C.ID_PACIENT = P.ID_PACIENT
WHERE P.ID_PACIENT = V_ID;
SELECT R.ID_RECOLTARE, R.ID_SALA_RECOLTARE, R.ID_ASISTENT_MEDICAL, R.ID_PACIENT, R.DATA_RECOLTARE, R.STATUS
BULK COLLECT INTO V_ISTORIC_RECOLTARI
FROM RECOLTARI R
JOIN PACIENTI P ON P.ID_PACIENT = R.ID_PACIENT
WHERE P.ID_PACIENT = V_ID
ORDER BY R.DATA_RECOLTARE ASC;
IF V_ISTORIC_RECOLTARI.COUNT = 0 THEN
    DBMS_OUTPUT.PUT_LINE ('PACIENTUL NU A FACUT ANALIZE MEDICALE');
```

```
ELSE

DBMS_OUTPUT.PUT_LINE('ISTORIC ANALIZE MEDICALE:');

FOR I IN V_ISTORIC_RECOLTARI.FIRST..V_ISTORIC_RECOLTARI.LAST LOOP

    V_COLECTIE_SET_RESULTATE.EXTEND;

    V_COLECTIE_SET_RESULTATE(V_COLECTIE_SET_RESULTATE.LAST):=SET_RESULTATE();

    V_COLECTIE_SET_RESULTATE(V_COLECTIE_SET_RESULTATE.LAST).EXTEND(5);

    SELECT TA.DENUMIRE, REZ.VALOARE, REZ.INTERPRETARE

    BULK COLLECT INTO V_COLECTIE_SET_RESULTATE(V_COLECTIE_SET_RESULTATE.LAST)

    FROM RECOLTARI R

    JOIN PACIENT P ON P.ID_PACIENT = R.ID_PACIENT

    JOIN RESULTATE REZ ON REZ.ID_RECOLTARE = R.ID_RECOLTARE

    JOIN TIPURI_ANALIZE TA ON TA.ID_TIP_ANALIZA = REZ.ID_TIP_ANALIZA

    WHERE P.ID_PACIENT = V_ID AND R.ID_RECOLTARE = V_ISTORIC_RECOLTARI(I).ID_RECOLTARE;

END LOOP;

FOR I IN V_COLECTIE_SET_RESULTATE.FIRST..V_COLECTIE_SET_RESULTATE.LAST LOOP

    DBMS_OUTPUT.PUT_LINE('RECOLTAREA CU ID-UL: '||V_ISTORIC_RECOLTARI(I).ID_RECOLTARE||' DE LA DATA:
    '||V_ISTORIC_RECOLTARI(I).DATA_RECOLTARE);

    FOR J IN V_COLECTIE_SET_RESULTATE(I).FIRST..V_COLECTIE_SET_RESULTATE(I).LAST LOOP

        DBMS_OUTPUT.PUT_LINE(
            V_COLECTIE_SET_RESULTATE(I)(J).TIP_ANALIZA || ' ' ||
            V_COLECTIE_SET_RESULTATE(I)(J).RESULTAT || ' ' ||
            V_COLECTIE_SET_RESULTATE(I)(J).INTERPRETARE
        );
    END LOOP;
END LOOP;

END IF;

IF V_ISTORIC_DIAGNOSTICE.COUNT = 0 THEN

    DBMS_OUTPUT.PUT_LINE('PACIENTUL NU A PRIMIT DIAGNOSTICE');

ELSE

    DBMS_OUTPUT.PUT_LINE('ISTORIC DIAGNOSTICE:');

    FOR I IN V_ISTORIC_DIAGNOSTICE.FIRST..V_ISTORIC_DIAGNOSTICE.LAST LOOP

        DBMS_OUTPUT.PUT_LINE(

```

```
V_ISTORIC_DIAGNOSTICE(I).DENUMIRE
);
END LOOP;
END IF;
END;

PROCEDURE TOP_MEDICI_DEPARTAMENT(V_DEP DEPARTAMENTE.ID_DEPARTAMENT%TYPE)
AS
TYPE ID_MEDICI IS TABLE OF MEDICI.ID_MEDIC%TYPE INDEX BY PLS_INTEGER;
V_MEDICI ID_MEDICI;
V_NUMAR INT;
V_CURSOR INT :=0;
V_TEST DEPARTAMENTE.ID_DEPARTAMENT%TYPE;
CURSOR C1 IS
SELECT M.ID_MEDIC ID_MEDIC,M.NUME NUME , M.PRENUME PRENUME, D.DENUMIRE DEPARTAMENT, M.SALARIU
SALARIU
FROM MEDICI M
JOIN DEPARTAMENTE D ON M.ID_DEPARTAMENT = D.ID_DEPARTAMENT
WHERE M.SALARIU IN (
SELECT MAX(SALARIU)
FROM MEDICI
WHERE ID_DEPARTAMENT = V_DEP
)
AND M.ID_DEPARTAMENT=V_DEP;

CURSOR C2 (V_ID MEDICI.ID_MEDIC%TYPE) IS
SELECT COUNT(C.ID_CONSULTATIE) NUMAR_CONSULTATII
FROM MEDICI M
LEFT JOIN CONSULTATII C ON C.ID_MEDIC = M.ID_MEDIC
HAVING M.ID_MEDIC = V_ID
GROUP BY M.ID_MEDIC;
BEGIN
SELECT ID_DEPARTAMENT
INTO V_TEST
```

```
FROM DEPARTAMENTE

WHERE ID_DEPARTAMENT = V_DEP;

FOR I IN C1 LOOP

    V_CURSOR:=V_CURSOR+1;

    OPEN C2(I.ID_MEDIC);

    FETCH C2 INTO V_NUMAR;

    EXIT WHEN C2%NOTFOUND;

    DBMS_OUTPUT.PUT_LINE('MEDICUL'||I.NUME||' '||I.PRENUME||' CU SALARIUL IN VALOARE DE'||I.SALARIU||' DIN
DEPARTAMENTUL'||I.DEPARTAMENT||' ARE UN NUMAR DE'||V_NUMAR||' CONSULTATII');

    CLOSE C2;

END LOOP;

IF V_CURSOR = 0 THEN

    DBMS_OUTPUT.PUT_LINE('NU EXISTA MEDICI IN ACEST DEPARTAMENT');

END IF;

EXCEPTION

    WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20000, 'NU EXISTA DEPARTAMENT CU ID-UL DAT');

END;

FUNCTION NUMAR_CONSULTATII(V_ID DEPARTAMENTE.ID_DEPARTAMENT%TYPE, MINI NUMBER, MAXI NUMBER)
RETURN DEPARTAMENTE.SPECIALIZARE%TYPE

IS

    NUMAR_NEGATIV EXCEPTION;
    NU_APARTINE_INTERVAL EXCEPTION;
    ORDINE_GRESITA_CAPETE EXCEPTION;

    V_DEPT DEPARTAMENTE.SPECIALIZARE%TYPE;
    V_NUMAR NUMBER;
    V_SPECIALIZARE DEPARTAMENTE.SPECIALIZARE%TYPE;

BEGIN

    IF MINI<0 OR MAXI<0 THEN

        RAISE NUMAR_NEGATIV;

    END IF;

    SELECT SPECIALIZARE
    INTO V_DEPT
```

Proiect Sisteme de Gestiune a Bazelor de Date, anul 2, semestrul 1
Berca Teodora

```
FROM DEPARTAMENTE
WHERE ID_DEPARTAMENT = V_ID;
SELECT COUNT(C.ID_CONSULTATIE)
INTO V_NUMAR
FROM DEPARTAMENTE D
JOIN MEDICI M ON M.ID_DEPARTAMENT = D.ID_DEPARTAMENT
JOIN CONSULTATII C ON C.ID_MEDIC = M.ID_MEDIC
WHERE D.ID_DEPARTAMENT = V_ID;
IF MINI > MAXI THEN
    RAISE ORDINE_GRESITA_CAPETE;
END IF;
IF V_NUMAR<MINI OR V_NUMAR>MAXI THEN
    RAISE NU_APARTINE_INTERVAL;
END IF;
RETURN V_DEPT;
EXCEPTION
    WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20000,'NU EXISTA DEPARTAMENT CU ID-UL DAT');
    WHEN NUMAR_NEGATIV THEN RAISE_APPLICATION_ERROR(-20001,'CAPELELE DE INTERVAL TREBUIE SA FIE NUMERE POZITIVE');
    WHEN NU_APARTINE_INTERVAL THEN RAISE_APPLICATION_ERROR(-20002,'NUMARUL DE CONSULTATII DIN ACEST DEPARTAMENT NU SE INCADREAZA IN INTERVALUL INTRODUS');
    WHEN ORDINE_GRESITA_CAPETE THEN RAISE_APPLICATION_ERROR(-20003,'CAPELELE DE INTERVAL NU SUNT ASEZATE IN ORDINEA CORECTA, MINIMUL TREBUIE SA FIE MAI MIC SAU EGAL CU MAXIMUL');
END NUMAR_CONSULTATII;

PROCEDURE CONSULTATII_MEDIC (V_NUME MEDICI.NUME%TYPE)
AS
TYPE DATE_CONSULTATIE IS RECORD(
    NUME_PACIENT PACIENTI.NUME%TYPE,
    PRENUME_PACIENT PACIENTI.PRENUME%TYPE,
    DATA_CONSULTATIE CONSULTATII.DATA_CONSULTATIE%TYPE,
    NUMAR_CABINET CABINETE.NUMAR%TYPE,
    DENUMIRE_DIAGNOSTIC DIAGNOSTICE.DENUMIRE%TYPE
);
```

```
TYPE TABLOU IS TABLE OF DATE_CONSULTATIE INDEX BY PLS_INTEGER;
ISTORIC_CONSULTATII TABLOU;
V_EROARE PACIENTI.NUME%TYPE;

BEGIN
    SELECT NUME
    INTO V_EROARE
    FROM MEDICI
    WHERE UPPER(NUME)=UPPER(V_NUME);
    SELECT P.NUME, P.PRENUME, C.DATA_CONSULTATIE, CAB.NUMAR, D.DENUMIRE
    BULK COLLECT INTO ISTORIC_CONSULTATII
    FROM MEDICI M
    JOIN CONSULTATII C ON C.ID_MEDIC = M.ID_MEDIC
    JOIN PACIENTI P ON P.ID_PACIENT = C.ID_PACIENT
    JOIN DIAGNOSTICE D ON D.ID_DIAGNOSTIC = C.ID_DIAGNOSTIC
    JOIN CABINETE CAB ON CAB.ID_CABINET = C.ID_CABINET
    WHERE UPPER(V_NUME)=UPPER(M.NUME) AND EXTRACT(YEAR FROM SYSDATE)-EXTRACT(YEAR FROM
P.DATA_NASTERE)<=50;
    FOR I IN ISTORIC_CONSULTATII.FIRST..ISTORIC_CONSULTATII.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(
            'NUME PACIENT: ' ||
            ISTORIC_CONSULTATII(I).NUME_PACIENT || '' ||
            'PRENUME PACIENT: ' ||
            ISTORIC_CONSULTATII(I).PRENUME_PACIENT || '' ||
            'DATA CONSULTATIE: ' ||
            ISTORIC_CONSULTATII(I).DATA_CONSULTATIE || '' ||
            'NUMAR CABINET ' ||
            ISTORIC_CONSULTATII(I).NUMAR_CABINET || '' ||
            'DIAGNOSTIC ' ||
            ISTORIC_CONSULTATII(I).DENUMIRE_DIAGNOSTIC
        );
    END LOOP;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20000, 'NU EXISTA MEDICI CU NUMELE DAT');
```

Proiect Sisteme de Gestiune a Bazelor de Date, anul 2, semestrul 1
 Berca Teodora

```

        WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR(-20001, 'EXISTA MAI MULTI MEDICI CU NUMELE DAT');

        WHEN OTHERS THEN RAISE_APPLICATION_ERROR(-20002, 'ALTA EROARE');

END;

```

END PACHET;

```

        WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20000, 'NU EXISTA MEDICI CU NUMELE DAT');
        WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR(-20001, 'EXISTA MAI MULTI MEDICI CU NUMELE DAT');
        WHEN OTHERS THEN RAISE_APPLICATION_ERROR(-20002, 'ALTA EROARE');

END;

END PACHET;

EXECUTE PACHET.ISTORIC_PACIENT(1);
EXECUTE PACHET.TOP_MEDICI_DEPARTAMENT(1);

```

Script Output | Task completed in 0.088 seconds

PL/SQL procedure successfully completed.

Dbms Output | Messages - Log | Buffer Size:20000

project x

```

NUME: Dumitrescu PRENUME: Mara
ISTORIC ANALIZE MEDICALE:
RECOLTAREA CU ID-UL: 1 DE LA DATA: 21-AUG-23
Glicemie 90 NORMAL
Magneziu 2 NORMAL
Vitamina D 70 NORMAL
RECOLTAREA CU ID-UL: 4 DE LA DATA: 21-OCT-23
Calciu Ionic 4 NORMAL
Hemoglobina Glicata 3 NORMAL
RECOLTAREA CU ID-UL: 5 DE LA DATA: 21-DEC-23
Glicemie 120 CRESCAT
TERAPIE ANTICRIZANTICA

```

```

        WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20000, 'NU EXISTA MEDICI CU NUMELE DAT');
        WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR(-20001, 'EXISTA MAI MULTI MEDICI CU NUMELE DAT');
        WHEN OTHERS THEN RAISE_APPLICATION_ERROR(-20002, 'ALTA EROARE');

END;

END PACHET;

EXECUTE PACHET.ISTORIC_PACIENT(1);
EXECUTE PACHET.TOP_MEDICI_DEPARTAMENT(1);
EXECUTE PACHET.TOP_MEDICI_DEPARTAMENT(5);

```

Script Output | Task completed in 0.058 seconds

PL/SQL procedure successfully completed.

Dbms Output | Messages - Log | Buffer Size:20000

project x

```

MEDICUL Bagu Cristian CU SALARIUL IN VALOARE DE 7000 DIN DEPARTAMENTUL DEP_ORL ARE UN NUMAR DE 2 CONSULTATII
MEDICUL Sandu Ramona CU SALARIUL IN VALOARE DE 7000 DIN DEPARTAMENTUL DEP_ORL ARE UN NUMAR DE 2 CONSULTATII

```

```

        WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20000, 'NU EXISTA MEDICI CU NUMELE DAT');
        WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR(-20001, 'EXISTA MAI MULTI MEDICI CU NUMELE DAT');
        WHEN OTHERS THEN RAISE_APPLICATION_ERROR(-20002, 'ALTA EROARE');

END;

END PACHET;

EXECUTE PACHET.ISTORIC_PACIENT(1);
EXECUTE PACHET.TOP_MEDICI_DEPARTAMENT(1);
EXECUTE PACHET.TOP_MEDICI_DEPARTAMENT(5);
SELECT PACHET.NUMAR_CONSULTATII(1,0,15)
FROM DUAL;
SELECT PACHET.NUMAR_CONSULTATII(1,0,5)
FROM DUAL;

```

Script Output | Query Result | All Rows Fetched: 1 in 0.003 seconds

PACHET.NUMAR_CONSULTATII(1,0,15)
1 DERMATOLOGIE

The screenshot shows the Oracle SQL Developer interface. The top window displays a portion of a PL/SQL package body:

```
ISTORIC_CONSULTATII(I).DENUMIRE_DIAGNOSTIC
);
END LOOP;

EXCEPTION
  WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20000, 'NU EXISTA MEDICI CU NUMELE DAT');
  WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR(-20001, 'EXISTA MAI MULTII MEDICI CU NUMELE DAT');
  WHEN OTHERS THEN RAISE_APPLICATION_ERROR(-20002,'ALTA EROARE');

END;
END PACHET;
EXECUTE PACHET.CONSULTATII_MEDIC('BADESCU');

EXECUTE PACHET.ISTORIC_PACIENT(1);
4
```

The bottom window, titled "Script Output", shows the results of the executed code:

```
Script Output x | Task completed in 0.073 seconds
Dbms Output x | Messages - Log | Buffer Size: 20000 | project x
NUME PACIENT: Lazar PRENUME PACIENT: Horatiu DATA CONSULTATIE: 06-JAN-23 NUMAR CABINET 2 DIAGNOSTIC Acnee
NUME PACIENT: Stanciu PRENUME PACIENT: Silviu DATA CONSULTATIE: 06-JAN-23 NUMAR CABINET 2 DIAGNOSTIC Dermatita
NUME PACIENT: Anghel PRENUME PACIENT: Mihnea DATA CONSULTATIE: 06-JAN-23 NUMAR CABINET 2 DIAGNOSTIC Dermatita
NUME PACIENT: Anghel PRENUME PACIENT: Mihnea DATA CONSULTATIE: 07-JAN-23 NUMAR CABINET 2 DIAGNOSTIC Dermatita
```

```
EXECUTE PACHET.ISTORIC_PACIENT(1);

EXECUTE PACHET.TOP_MEDICI_DEPARTAMENT(5);

SELECT PACHET.NUMAR_CONSULTATII(1,0,15)

FROM DUAL;

EXECUTE PACHET.CONSULTATII_MEDIC('BADESCU');
```