**Gestionarea Bazei de Date a unui spital**

**Descrierea problemei:**

Am ales sa realizez baza de date a unui spital cu scopul de a eficientiza organizarea membrilor spitalului si pentru a facilita stocarea informatiilor despre pacienti si personal.

**Relatiile dintre tabele sunt:**

-fiecare paciet poate avea unul sau mai multe diagnostice

-fiecare pacient poate avea una sau mai multe alergii

-fiecare afectiune poate avea unul sau mai multe diagnostice

-fiecare diagnostic poate avea un tratament

-fiecare doctor poate atribui unul sau mai multe diagnostice

-un diagnostic poate fi asociat uneia sau mai multor internari

**Diagram

Description automatically generatedSchema bazei de date**

**Crearea tabelelor**

DROP TABLE Pacient CASCADE CONSTRAINTS;

DROP TABLE Doctor CASCADE CONSTRAINTS;

DROP TABLE Alergie CASCADE CONSTRAINTS;

DROP TABLE Afectiune CASCADE CONSTRAINTS;

DROP TABLE Diagnostic CASCADE CONSTRAINTS;

DROP TABLE Tratament CASCADE CONSTRAINTS;

DROP TABLE Internari CASCADE CONSTRAINTS;

CREATE TABLE Pacient (

CNP CHAR(13) CONSTRAINT pk\_pacient PRIMARY KEY,

nume VARCHAR(30) CONSTRAINT nn\_nume NOT NULL,

prenume VARCHAR(30) NOT NULL,

sex CHAR(2),

data\_nastere Date

);

CREATE TABLE Doctor (

cod\_doctor CHAR(10) CONSTRAINT pk\_doctor PRIMARY KEY,

nume\_doctor VARCHAR(30) NOT NULL,

prenume\_doctor VARCHAR(30) NOT NULL,

data\_angajare Date,

specializare VARCHAR(30),

sectia VARCHAR(30)

);

CREATE TABLE Afectiune (

cod\_boala CHAR(10) CONSTRAINT pk\_afectiune PRIMARY KEY,

denumire\_boala VARCHAR(30),

cod\_tratament CHAR(10)

);

CREATE TABLE Diagnostic (

cod\_diagnostic CHAR(10) CONSTRAINT pk\_diagnostic PRIMARY KEY,

cod\_boala CHAR(10) CONSTRAINT fk\_diagnostic\_1 REFERENCES Afectiune(cod\_boala),

CNP CHAR(13) CONSTRAINT fk\_diagnostic\_2 REFERENCES Pacient(CNP),

data\_diagnosticului Date,

cod\_doctor CHAR(10) CONSTRAINT fk\_diagnostic\_3 REFERENCES Doctor(cod\_doctor)

);

CREATE TABLE Tratament (

cod\_tratament CHAR(10) CONSTRAINT pk\_tratament PRIMARY KEY,

data\_incepere Date NOT NULL,

data\_terminare Date NOT NULL,

pret\_medicament NUMBER NOT NULL,

cod\_diagnostic CHAR(10) CONSTRAINT fk\_tratament REFERENCES Diagnostic(cod\_diagnostic)

);

CREATE TABLE Internari (

cod\_internare CHAR(10) CONSTRAINT pk\_internari PRIMARY KEY,

data\_internare Date NOT NULL,

data\_externare Date NOT NULL,

cod\_diagnostic CHAR(10) CONSTRAINT fk\_internari REFERENCES Diagnostic(cod\_diagnostic),

salon char(5)

);

CREATE TABLE Alergie (

cod\_alergie CHAR(10) CONSTRAINT pk\_alergie PRIMARY KEY,

nume\_alergie VARCHAR(20) NOT NULL,

severitate VARCHAR(10) NOT NULL,

CNP CHAR(13) CONSTRAINT fk\_alergie REFERENCES Pacient(CNP)

);

**Alergie**

INSERT INTO Alergie(cod\_alergie, nume\_alergie, severitate, CNP) values ('123' ,'ambrozie', 'usoara','5263484920214');

INSERT INTO Alergie(cod\_alergie, nume\_alergie, severitate, CNP) values ('124' ,'aspirina', 'severa','6854776390251');

INSERT INTO Alergie(cod\_alergie, nume\_alergie, severitate, CNP) values ('125' ,'alimentara', 'moderata','2656666778958');

Text

Description automatically generated

Graphical user interface, application

Description automatically generated

**Pacient**

INSERT INTO Pacient(CNP, nume, prenume, sex, data\_nastere) values ('5263484920214', 'Catana', 'Ioan-Alexandru', 'm', to\_date('13-03-2001', 'dd-mm-yyyy'));

INSERT INTO Pacient(CNP, nume, prenume, sex, data\_nastere) values ('6889923847201', 'Ion', 'Mihaela', 'f', to\_date('04-04-1995', 'dd-mm-yyyy'));

INSERT INTO Pacient(CNP, nume, prenume, sex, data\_nastere) values ('2658742153958', 'Catana', 'Florin', 'm', to\_date('09-01-1947', 'dd-mm-yyyy'));

INSERT INTO Pacient(CNP, nume, prenume, sex, data\_nastere) values ('5263408315869', 'Popescu', 'Marcel', 'm', to\_date('21-06-1998', 'dd-mm-yyyy'));

INSERT INTO Pacient(CNP, nume, prenume, sex, data\_nastere) values ('6854776390251', 'Ionescu', 'Mihai', 'm', to\_date('15-12-1995', 'dd-mm-yyyy'));

INSERT INTO Pacient(CNP, nume, prenume, sex, data\_nastere) values ('5439860973658', 'Popa', 'Miruna', 'f', to\_date('05-11-1997', 'dd-mm-yyyy'));

INSERT INTO Pacient(CNP, nume, prenume, sex, data\_nastere) values ('5266969690214', 'Titeche', 'Anamaria-Florentina', 'f', to\_date('25-04-2000', 'dd-mm-yyyy'));

INSERT INTO Pacient(CNP, nume, prenume, sex, data\_nastere) values ('6854544477551', 'Rosca', 'Laura', 'f', to\_date('22-04-1977', 'dd-mm-yyyy'));

INSERT INTO Pacient(CNP, nume, prenume, sex, data\_nastere) values ('2656666778958', 'Geany', 'Felix', 'm', to\_date('28-10-1980', 'dd-mm-yyyy'));

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

**Afectiune**

INSERT INTO Afectiune(cod\_boala, denumire\_boala, cod\_tratament) values ('120', 'apendicita' , '125');

INSERT INTO Afectiune(cod\_boala, denumire\_boala, cod\_tratament) values ('140', 'tensiune' , '145');

INSERT INTO Afectiune(cod\_boala, denumire\_boala, cod\_tratament) values ('220', 'scleroza' , '215');

INSERT INTO Afectiune(cod\_boala, denumire\_boala, cod\_tratament) values ('320', 'bronsita' , '665');

INSERT INTO Afectiune(cod\_boala, denumire\_boala, cod\_tratament) values ('420', 'depresie' , '695');

INSERT INTO Afectiune(cod\_boala, denumire\_boala, cod\_tratament) values ('540', 'otita' , '705');

INSERT INTO Afectiune(cod\_boala, denumire\_boala, cod\_tratament) values ('710', 'COVID-19' , '815');

INSERT INTO Afectiune(cod\_boala, denumire\_boala, cod\_tratament) values ('720', 'toxiinfectie alimentara' , '825');

INSERT INTO Afectiune(cod\_boala, denumire\_boala, cod\_tratament) values ('730', 'sinuzita' , '835');

Graphical user interface

Description automatically generated with medium confidence

Table

Description automatically generated

**Diagnostic**

INSERT INTO Diagnostic(cod\_diagnostic, cod\_boala, CNP, data\_diagnosticului, cod\_doctor) values ('223', '120', '5263484920214', to\_date('12-01-2023', 'dd-mm-yyyy'), '2781');

INSERT INTO Diagnostic(cod\_diagnostic, cod\_boala, CNP, data\_diagnosticului, cod\_doctor) values ('333', '120', '6889923847201', to\_date('05-01-2023', 'dd-mm-yyyy'), '4689');

INSERT INTO Diagnostic(cod\_diagnostic, cod\_boala, CNP, data\_diagnosticului, cod\_doctor) values ('413', '140', '2658742153958', to\_date('27-02-2023', 'dd-mm-yyyy'), '4689');

INSERT INTO Diagnostic(cod\_diagnostic, cod\_boala, CNP, data\_diagnosticului, cod\_doctor) values ('423', '140', '5263408315869', to\_date('14-02-2023', 'dd-mm-yyyy'), '9075');

INSERT INTO Diagnostic(cod\_diagnostic, cod\_boala, CNP, data\_diagnosticului, cod\_doctor) values ('453', '220', '6854776390251', to\_date('15-03-2023', 'dd-mm-yyyy'), '6666');

INSERT INTO Diagnostic(cod\_diagnostic, cod\_boala, CNP, data\_diagnosticului, cod\_doctor) values ('603', '320', '5439860973658', to\_date('17-09-2022', 'dd-mm-yyyy'), '6964');

INSERT INTO Diagnostic(cod\_diagnostic, cod\_boala, CNP, data\_diagnosticului, cod\_doctor) values ('803', '420', '5266969690214', to\_date('28-10-2022', 'dd-mm-yyyy'), '4200');

INSERT INTO Diagnostic(cod\_diagnostic, cod\_boala, CNP, data\_diagnosticului, cod\_doctor) values ('183', '420', '6854544477551', to\_date('24-11-2021', 'dd-mm-yyyy'), '8945');

INSERT INTO Diagnostic(cod\_diagnostic, cod\_boala, CNP, data\_diagnosticului, cod\_doctor) values ('993', '540', '2656666778958', to\_date('12-01-2021', 'dd-mm-yyyy'), '7887');

Text

Description automatically generated with low confidence

Graphical user interface, text

Description automatically generated

**Doctor**

INSERT INTO Doctor(cod\_doctor, nume\_doctor, prenume\_doctor, data\_angajare, specializare, sectia) values ('2781' , 'Popescu', 'Alex' , to\_date('15-12-2014','dd-mm-yyyy'), 'neurolog', 'neurologie');

INSERT INTO Doctor(cod\_doctor, nume\_doctor, prenume\_doctor, data\_angajare, specializare, sectia) values ('4689' , 'Ioanas' , 'Ioana', to\_date('12-07-2013','dd-mm-yyyy'), 'chirurg', 'chirurgie');

INSERT INTO Doctor(cod\_doctor, nume\_doctor, prenume\_doctor, data\_angajare, specializare, sectia) values ('9075' , 'Tudor' , 'Rares', to\_date('18-05-2015','dd-mm-yyyy'), 'cardiolog', 'cardiologie');

INSERT INTO Doctor(cod\_doctor, nume\_doctor, prenume\_doctor, data\_angajare, specializare, sectia) values ('6666' , 'Mihalcea' , 'Maria', to\_date('25-03-2014','dd-mm-yyyy'), 'ginecolog', 'ginecologie');

INSERT INTO Doctor(cod\_doctor, nume\_doctor, prenume\_doctor, data\_angajare, specializare, sectia) values ('6964' , 'Mocanu' , 'Stefan', to\_date('14-02-2013','dd-mm-yyyy'), 'pediatru', 'pediatrie');

INSERT INTO Doctor(cod\_doctor, nume\_doctor, prenume\_doctor, data\_angajare, specializare, sectia) values ('4200' , 'Tudorici' , 'Matei', to\_date('28-10-2015','dd-mm-yyyy'), 'endocrinolog', 'endocrinologie');

INSERT INTO Doctor(cod\_doctor, nume\_doctor, prenume\_doctor, data\_angajare, specializare, sectia) values ('8945' , 'Matache' , 'Luca', to\_date('07-11-2016','dd-mm-yyyy'), 'specialist ORL', 'ORL');

INSERT INTO Doctor(cod\_doctor, nume\_doctor, prenume\_doctor, data\_angajare, specializare, sectia) values ('7887' , 'Savu' , 'Elvira', to\_date('12-12-2017','dd-mm-yyyy'), 'psihiatru', 'psihiatrie');

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

**Tratament**

INSERT INTO Tratament(cod\_tratament, data\_incepere, data\_terminare, pret\_medicament, cod\_diagnostic) values ('125' , to\_date('13-01-2023', 'dd-mm-yyyy'), to\_date('14-01-2023', 'dd-mm-yyyy'), 150, '223');

INSERT INTO Tratament(cod\_tratament, data\_incepere, data\_terminare, pret\_medicament, cod\_diagnostic) values ('145' , to\_date('06-01-2023', 'dd-mm-yyyy'), to\_date('07-01-2023', 'dd-mm-yyyy'), 190, '333');

INSERT INTO Tratament(cod\_tratament, data\_incepere, data\_terminare, pret\_medicament, cod\_diagnostic) values ('215' , to\_date('28-02-2023', 'dd-mm-yyyy'), to\_date('30-03-2023', 'dd-mm-yyyy'), 300, '413');

INSERT INTO Tratament(cod\_tratament, data\_incepere, data\_terminare, pret\_medicament, cod\_diagnostic) values ('665' , to\_date('15-02-2023', 'dd-mm-yyyy'), to\_date('16-02-2023', 'dd-mm-yyyy'), 140, '423');

INSERT INTO Tratament(cod\_tratament, data\_incepere, data\_terminare, pret\_medicament, cod\_diagnostic) values ('695' , to\_date('16-03-2023', 'dd-mm-yyyy'), to\_date('17-03-2023', 'dd-mm-yyyy'), 870, '453');

INSERT INTO Tratament(cod\_tratament, data\_incepere, data\_terminare, pret\_medicament, cod\_diagnostic) values ('705' , to\_date('18-09-2022', 'dd-mm-yyyy'), to\_date('19-09-2022', 'dd-mm-yyyy'), 90, '603');

INSERT INTO Tratament(cod\_tratament, data\_incepere, data\_terminare, pret\_medicament, cod\_diagnostic) values ('815' , to\_date('29-10-2022', 'dd-mm-yyyy'), to\_date('30-11-2022', 'dd-mm-yyyy'), 5400, '803');

INSERT INTO Tratament(cod\_tratament, data\_incepere, data\_terminare, pret\_medicament, cod\_diagnostic) values ('825' , to\_date('25-11-2021', 'dd-mm-yyyy'), to\_date('26-12-2021', 'dd-mm-yyyy'), 179, '183');

INSERT INTO Tratament(cod\_tratament, data\_incepere, data\_terminare, pret\_medicament, cod\_diagnostic) values ('835' , to\_date('13-01-2021', 'dd-mm-yyyy'), to\_date('13-03-2021', 'dd-mm-yyyy'), 185, '993');Table

Description automatically generated

Table

Description automatically generated

**Internari**

INSERT INTO Internari(cod\_internare, data\_internare, data\_externare, cod\_diagnostic, salon) values ('1001' , to\_date('16-03-2023', 'dd-mm-yyyy'), to\_date('17-03-2023', 'dd-mm-yyyy'),'803','2203');

INSERT INTO Internari(cod\_internare, data\_internare, data\_externare, cod\_diagnostic, salon) values ('1002' , to\_date('18-09-2022', 'dd-mm-yyyy'), to\_date('19-09-2022', 'dd-mm-yyyy'),'603','2706');

INSERT INTO Internari(cod\_internare, data\_internare, data\_externare, cod\_diagnostic, salon) values ('1008' , to\_date('29-10-2022', 'dd-mm-yyyy'), to\_date('30-11-2022', 'dd-mm-yyyy'),'223', '2016');

INSERT INTO Internari(cod\_internare, data\_internare, data\_externare, cod\_diagnostic, salon) values ('1230' , to\_date('25-11-2021', 'dd-mm-yyyy'), to\_date('26-12-2021', 'dd-mm-yyyy'),'333','2104');

INSERT INTO Internari(cod\_internare, data\_internare, data\_externare, cod\_diagnostic, salon) values ('2001' , to\_date('13-01-2021', 'dd-mm-yyyy'), to\_date('13-03-2021', 'dd-mm-yyyy'),'993', '2203');

Table

Description automatically generated

Text

Description automatically generated

**Structuri de control**

--1. Sa se scrie un bloc PL/SQL prin care să se creasca pretul medicamentelor pentru tratamentul al carui cod este introdus de la tastaura, astfel:

* De doua ori daca pretul initial e mai mic decat 150
* Cu 50% daca pretul initial este intre 150 si 300
* Cu 25% in alt caz

SET SERVEROUTPUT ON

DECLARE

v\_pret tratament.pret\_medicament%TYPE;

v\_cod tratament.cod\_tratament%TYPE:=&cod;

BEGIN

SELECT pret\_medicament

INTO v\_pret

FROM Tratament

WHERE cod\_tratament = v\_cod;

DBMS\_OUTPUT.PUT\_LINE('Pretul initial al medicamentelor pentru tratamentul cu codul ' || v\_cod || ' este: ' || v\_pret);

IF v\_pret < 150 THEN

v\_pret:=2\* v\_pret;

ELSIF v\_pret between 150 and 300 THEN

v\_pret:=1.5 \* v\_pret;

ELSE

v\_pret:=1.25\* v\_pret;

END IF;

DBMS\_OUTPUT.PUT\_LINE('Pretul final al medicamentelor pentru tratamentul cu codul ' || v\_cod ||' este: '||v\_pret);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nu exista tratamentul');

END;

/

Pentru codul 705, pretul este 90 si ar trebui sa se dubleze:

Graphical user interface, text, application, Word

Description automatically generated

Pentru codul 825 pretul este 179 si ar trebui sa creasca cu 50%:

Graphical user interface, text, application, email

Description automatically generated

Pentru codul 695 pretul este 870 si ar trebui sa creasca cu 25%:

Graphical user interface, text, application, email

Description automatically generated

Pentru coduri care nu exista se arunca exceptie

Graphical user interface, text, application, email

Description automatically generated

--2. Pentru doctorii ce au codul mai mic de 6667 sa se adauge o luna de vechime.

BEGIN

UPDATE doctor

SET data\_angajare = data\_angajare + 30

WHERE cod\_doctor < 6667;

IF SQL%NOTFOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nu exista doctori ce au codul mai mic de 6667');

ELSE

DBMS\_OUTPUT.PUT\_LINE('S-a modificat data de angajare pentru ' || SQL%ROWCOUNT || ' doctori');

END IF;

COMMIT;

END;

/

Graphical user interface, application, email

Description automatically generated

In cazul in care nu exista niciun doctor cu codul mai mic decat 6667 se va afisa:

Graphical user interface, text, application

Description automatically generated

--3. Sa se afiseze toti pacientii spitalului si doctorii lor

DECLARE

CURSOR c IS

SELECT nume, prenume, sex, data\_nastere, cnp

FROM pacient;

r c%ROWTYPE;

v\_nume\_doctor doctor.nume\_doctor%TYPE;

v\_prenume\_doctor doctor.prenume\_doctor%TYPE;

BEGIN

OPEN c;

LOOP

FETCH c INTO r;

EXIT WHEN c%NOTFOUND;

SELECT dr.nume\_doctor, dr.prenume\_doctor INTO v\_nume\_doctor, v\_prenume\_doctor

FROM doctor dr, pacient p, diagnostic d

WHERE p.cnp=d.cnp AND

d.cod\_doctor = dr.cod\_doctor AND

r.cnp=p.cnp;

DBMS\_OUTPUT.PUT\_LINE('Pacientul '|| r.nume ||' ' || r.prenume ||' de sex ' || r.sex ||' nascut in data de ' ||r.data\_nastere || ' are doctorul ' || v\_nume\_doctor ||' '|| v\_prenume\_doctor);

END LOOP;

CLOSE c;

END;

/

Graphical user interface, text, application

Description automatically generated

--4. --Utilizati un cursor pentru a prelua numele, codul si vechimea doctorului al carui cod este introdus de la tastatura

DECLARE

CURSOR c(p\_cod\_doctor doctor.cod\_doctor%TYPE) IS

SELECT nume\_doctor,prenume\_doctor, cod\_doctor, ROUND((SYSDATE-data\_angajare)/365) vechime

FROM doctor

WHERE cod\_doctor = p\_cod\_doctor;

r c%ROWTYPE;

v\_cod\_doctor doctor.cod\_doctor%TYPE :=&cod;

BEGIN

OPEN c (v\_cod\_doctor);

LOOP

FETCH c INTO r;

EXIT WHEN c%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Dr. '|| r.nume\_doctor ||' '||r.prenume\_doctor||' cu codul ' || r.cod\_doctor ||' are vechimea de '|| r.vechime||' ani');

END LOOP;

CLOSE c;

END;

/

Graphical user interface, text, application, email

Description automatically generated

--5.Sa se afiseze pacientii si numarul de alergii pe care le au fiecare

DECLARE

CURSOR c IS SELECT a.cnp, a.nume,COUNT(p.cod\_alergie) nr\_alergii

FROM pacient a JOIN alergie p

ON a.cnp = p.cnp

-- WHERE id\_departament IS NOT NULL

GROUP BY a.cnp, a.nume

ORDER BY COUNT(p.cod\_alergie) DESC;

BEGIN

FOR v IN c LOOP

DBMS\_OUTPUT.PUT\_LINE(v.nume||' '|| v.nr\_alergii);

END LOOP;

END;

/

Graphical user interface, text, application, email

Description automatically generated

--6. Sa se insereze un doctor nou fara a se specifica toate detaliile necesare

DECLARE

INSERT\_EXCEPT EXCEPTION;

PRAGMA EXCEPTION\_INIT(INSERT\_EXCEPT, -01400);

BEGIN

INSERT INTO doctor(cod\_doctor,specializare) values (6, 'Chirurg plastician');

EXCEPTION

WHEN insert\_except THEN

DBMS\_OUTPUT.PUT\_LINE('Nu s-au precizat toate informatiile pentru detalii doctorului');

DBMS\_OUTPUT.PUT\_LINE(SQLERRM);

END;

/

Graphical user interface, text, application, email

Description automatically generated

7--Se modifică denumirea pacientului cu CNP-ul 2656666778958. Dacă nu se produce nicio actualizare sau dacă apare o altă excepție, atunci se declanşează o excepţie prin care să fie avertizat utilizatorul:

DECLARE

invalid\_pacient EXCEPTION;

BEGIN

UPDATE pacient

SET nume='Moraru'

WHERE CNP='265666678958';

IF SQL%NOTFOUND THEN

RAISE invalid\_pacient;

END IF;

EXCEPTION

WHEN invalid\_pacient THEN

DBMS\_OUTPUT.PUT\_LINE('Nu exista pacientul cu acest CNP');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Nu se poate actualiza numele pacientului!');

END;

/

Graphical user interface, table

Description automatically generated

Daca nu exista CNP-ul:

Graphical user interface, text, application, email

Description automatically generated

**Gestionarea Cursorilor și a Excepțiilor**

SET SERVEROUTPUT ON;

--1 Într-un bloc PL/SQL utilizați un cursor parametrizat pentru a prelua numele,prenumele și vârsta pacientilor ce au un anumit sex (m/f/n)

--parcurgeți cursorul și afișați informațiile solicitate pentru acei pacienti care au sexul egal cu valoarea citită de la tastatură si afisati numarul de pacienti parcursi;

--în cazul în care nu există sexul indicat, se va invoca o excepție, care se va trata corespunzător;

--în cazul în care nu există pacienti cu acel sex, se va invoca o excepție, care se va trata corespunzător.

DECLARE

CURSOR c (p\_sex pacient.sex%TYPE) IS

SELECT nume, prenume, ROUND((SYSDATE-data\_nastere)/365) varsta

FROM pacient

WHERE sex = p\_sex;

r c%ROWTYPE;

v\_sex pacient.sex%TYPE;

nu\_exista\_pacient EXCEPTION;

sexul\_nu\_exista EXCEPTION;

prea\_multe\_caractere EXCEPTION;

PRAGMA EXCEPTION\_INIT(prea\_multe\_caractere, -06502);

BEGIN

v\_sex := '&sexul';

OPEN c (v\_sex);

LOOP

FETCH c INTO r;

EXIT WHEN c%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(r.nume||' ' ||r.prenume||' ' || r.varsta);

END LOOP;

IF c%ROWCOUNT=0 AND (v\_sex = 'm' OR v\_sex = 'f' OR v\_sex = 'n') THEN

RAISE nu\_exista\_pacient;

ELSE DBMS\_OUTPUT.PUT\_LINE('S-au afisat '|| c%ROWCOUNT || ' pacienti.');

END IF;

IF v\_sex != 'm' AND v\_sex != 'f' AND v\_sex != 'n' THEN

RAISE sexul\_nu\_exista;

END IF;

CLOSE c;

EXCEPTION

WHEN nu\_exista\_pacient THEN DBMS\_OUTPUT.PUT\_LINE('Exista sexul, dar niciun pacient nu il are');

WHEN sexul\_nu\_exista THEN DBMS\_OUTPUT.PUT\_LINE('Sexul introdus de la tastatura nu exista');

WHEN prea\_multe\_caractere THEN DBMS\_OUTPUT.PUT\_LINE('Input-ul trebuie sa fie de un caracter. Ceea ce ati introdus depaseste limita.');

WHEN OTHERS THEN DBMS\_OUTPUT.PUT\_LINE('A aparut o eroare');

END;

/

Pentru sexul feminin

Graphical user interface, text, application

Description automatically generated

Pentru sexul masculin:

Text

Description automatically generated

Pentru sexul neutru:

Graphical user interface, text, application, email

Description automatically generated

Pentru un sex care cu exista (de ex. t):

Text

Description automatically generated

Daca inputul depaseste limita maxima de caractere:

Text

Description automatically generated

--2. Într-un bloc PL/SQL citiți de la tastatură identificatorul unei boli. Afișați denumirea bolii care are acel cod.

--dacă boala nu există, tratați excepția cu o rutină de tratare corespunzătoare;

--daca boala exista, dar niciun pacient nu o are, tratati exceptia

--tratați orice altă excepție cu o rutină de tratare corespunzătoare.

DECLARE

CURSOR c(p\_cod\_boala afectiune.cod\_boala%TYPE) IS

SELECT denumire\_boala

FROM afectiune

WHERE cod\_boala = p\_cod\_boala;

r c%ROWTYPE;

v\_denumire\_boala afectiune.denumire\_boala%TYPE;

v\_test NUMBER;

v\_cod afectiune.cod\_boala%TYPE;

boala\_nu\_exista EXCEPTION;

boala\_nu\_este\_inregistrata EXCEPTION;

prea\_multe\_caractere EXCEPTION;

input\_nu\_numar EXCEPTION;

PRAGMA EXCEPTION\_INIT(prea\_multe\_caractere, -06502);

PRAGMA EXCEPTION\_INIT (input\_nu\_numar, -06550);

BEGIN

v\_cod := &cod;

SELECT denumire\_boala INTO v\_denumire\_boala

FROM afectiune

WHERE cod\_boala = v\_cod;

OPEN c(v\_cod);

LOOP

FETCH c INTO r;

EXIT WHEN c%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Boala: '|| v\_denumire\_boala );

END LOOP;

SELECT COUNT( cnp) INTO v\_test

FROM diagnostic

WHERE cod\_boala=v\_cod;

IF v\_test=0 AND c%NOTFOUND THEN

RAISE boala\_nu\_este\_inregistrata;

ELSE

DBMS\_OUTPUT.PUT\_LINE(v\_test || ' pacienti au aceasta boala');

END IF;

CLOSE c;

EXCEPTION

WHEN TOO\_MANY\_ROWS THEN DBMS\_OUTPUT.PUT\_LINE('Sunt prea multe boli cu acest cod');

WHEN NO\_DATA\_FOUND THEN DBMS\_OUTPUT.PUT\_LINE('Nu exista boala');

WHEN boala\_nu\_este\_inregistrata THEN DBMS\_OUTPUT.PUT\_LINE('Boala exista dar niciun pacient nu o are');

WHEN prea\_multe\_caractere THEN DBMS\_OUTPUT.PUT\_LINE('Input-ul trebuie sa fie de 10 caractere. Ceea ce ati introdus depaseste limita.');

WHEN OTHERS THEN DBMS\_OUTPUT.PUT\_LINE('A aparut o eroare');

END;

/

Pentru codul 140: exista boala, 2 pacienti o au:

Graphical user interface, text

Description automatically generated

Pentru codul 720: exista boala, nu o are niciun pacient:

Text

Description automatically generated with low confidence

Pentru codul 97: boala nu exista:

Logo

Description automatically generated with medium confidence

Daca lungimea codului depaseste limita de 10 caractere:



--3 Sa se modifice data de terminare a tratamentului al carui cod este introdus de la tastatura prin incrementarea sa cu 28 zile

DECLARE

v\_data\_terminare tratament.data\_terminare%TYPE;

v\_cod tratament.cod\_tratament%TYPE:= &cod;

nu\_exista\_tratament EXCEPTION;

PRAGMA EXCEPTION\_INIT (nu\_exista\_tratament,+100);

BEGIN

SELECT data\_terminare INTO v\_data\_terminare

FROM tratament

WHERE cod\_tratament = v\_cod;

DBMS\_OUTPUT.PUT\_LINE('Data veche a terminarii tratamentului este: '|| v\_data\_terminare);

UPDATE tratament

SET data\_terminare = data\_terminare +28

WHERE cod\_tratament = v\_cod;

IF SQL%ROWCOUNT=0 THEN

RAISE nu\_exista\_tratament;

END IF;

DBMS\_OUTPUT.PUT\_LINE('Data noua a terminarii tratamentului este: '|| TO\_CHAR(TO\_DATE(v\_data\_terminare, 'DD-MON-YY') + 28, 'DD-MON-YYYY'));

EXCEPTION

WHEN nu\_exista\_tratament THEN DBMS\_OUTPUT.PUT\_LINE('Nu exista tratament ce are codul introdus de dumneavoastra');

WHEN TOO\_MANY\_ROWS THEN DBMS\_OUTPUT.PUT\_LINE('Exista mai mult de un tratament care are codul introdus de dumneavoastra');

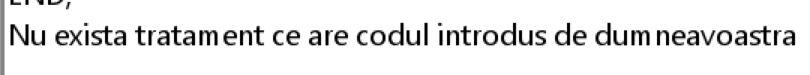
END;

Pentru codul 125:

Graphical user interface, text, application, email

Description automatically generated

Pentru codul 9:



--4. Sa se afiseze toti pacientii care au medicamentele mai scumpe decat o valoare introdusa de la tastatura.

--Daca nu exista niciun pacient sa se trateze exceptia cu o rutina corespunzatoare.

SET SERVEROUTPUT ON;

DECLARE

v\_numar NUMBER :=0;

v\_pret\_med tratament.pret\_medicament%TYPE;

pretul\_nu\_number EXCEPTION;

nicio\_valoare EXCEPTION;

CURSOR c\_tratamente IS

SELECT p.nume, t.cod\_tratament, t.pret\_medicament

FROM pacient p

JOIN diagnostic d ON p.CNP = d.CNP

JOIN tratament t ON d.cod\_diagnostic = t.cod\_diagnostic

WHERE t.pret\_medicament > v\_pret\_med;

BEGIN

v\_pret\_med := &pret;

BEGIN

v\_pret\_med := TO\_NUMBER(v\_pret\_med);

EXCEPTION

WHEN OTHERS THEN

RAISE pretul\_nu\_number;

END;

FOR r\_tratament IN c\_tratamente LOOP

v\_numar := v\_numar +1;

DBMS\_OUTPUT.PUT\_LINE(r\_tratament.nume || ' are un tratament cu codul ' || r\_tratament.cod\_tratament || ' si pretul de ' || r\_tratament.pret\_medicament);

END LOOP;

IF v\_numar = 0 THEN

RAISE nicio\_valoare;

ELSE

DBMS\_OUTPUT.PUT\_LINE('S-au afisat: '||v\_numar||' valori');

END IF;

EXCEPTION

WHEN nicio\_valoare THEN DBMS\_OUTPUT.PUT\_LINE('Nu exista niciun pacient care sa indeplineasca conditia.');

WHEN pretul\_nu\_number THEN

DBMS\_OUTPUT.PUT\_LINE('Pretul introdus de dumneavoastra nu este un numar.');

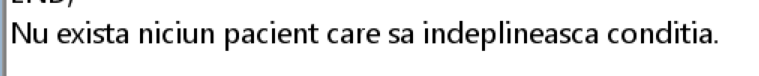
END;

Pentru pretul de 89:

Table

Description automatically generated

Pentru pretul de 6000:



--5.Sa se actualizeze severitatea alergiilor a caror denumire este introdusa de la tastatura.

--Daca severitatea nu este usoara, moderata sau severa sa se arunce o exceptie cu un mesaj in care utilizatorul sa fie informat de aceasta.

--Daca nu exista nicio alergie cu acea denumire sa se trateze exceptia cu o rutina corespunzatoare

DECLARE

severitate\_invalida EXCEPTION;

niciun\_update EXCEPTION;

severitate\_alergie alergie.severitate%TYPE;

denumire alergie.nume\_alergie%TYPE;

nr\_updates INTEGER := 0;

BEGIN

severitate\_alergie := '&severitate';

denumire := '&numele\_alergiei';

IF severitate\_alergie != 'severa' AND severitate\_alergie !='moderata' AND severitate\_alergie !='usoara'

THEN RAISE severitate\_invalida;

ELSE

UPDATE alergie

SET severitate = severitate\_alergie

WHERE nume\_alergie = denumire;

nr\_updates := SQL%ROWCOUNT;

END IF;

IF nr\_updates = 0

THEN

RAISE niciun\_update;

ELSE

DBMS\_OUTPUT.PUT\_LINE('Numarul de actualizari facute: ' || nr\_updates);

END IF;

EXCEPTION

WHEN severitate\_invalida THEN

DBMS\_OUTPUT.PUT\_LINE('Severitatea introdusa nu este una dintre: usoara, moderata sau severa ');

WHEN niciun\_update THEN

DBMS\_OUTPUT.PUT\_LINE('Nu s-a facut niciun update din cauza ca denumirea alergiei nu exista in baza de date. ');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Nu se poate actualiza severitatea alergiei!');

END;

Inainte de modificare:

Table

Description automatically generated

Dupa ce am modificat alergiile de arahide sa fie severe:

Graphical user interface, table

Description automatically generated

Daca severitatea nu este una din cele 3 predefinite:

Graphical user interface, text, application, email

Description automatically generated

Daca denumirea nu se gaseste in baza de date:

Graphical user interface, text, application, email

Description automatically generated

**Functii**

--1. Scrie functia max\_durata\_internare care calculeaza si afiseaza internarea cu durata cea mai mare si pacientul asociat ei

CREATE OR REPLACE FUNCTION max\_durata\_internare RETURN VARCHAR2 IS

v\_internare VARCHAR2(100);

v\_pacient VARCHAR2(100);

BEGIN

SELECT 'Pacientul ' || p.nume || ' ' || p.prenume || ' a fost internat timp de ' || TRUNC((i.data\_externare- i.data\_internare)) || ' zile.', p.nume || ' ' || p.prenume INTO v\_internare, v\_pacient

FROM internari i

JOIN diagnostic d ON i.cod\_diagnostic = d.cod\_diagnostic

JOIN pacient p ON d.cnp = p.cnp

WHERE i.data\_externare IS NOT NULL

ORDER BY (i.data\_externare- i.data\_internare) DESC

FETCH FIRST ROW ONLY;

DBMS\_OUTPUT.PUT\_LINE(v\_internare);

RETURN v\_pacient;

EXCEPTION

WHEN TOO\_MANY\_ROWS THEN DBMS\_OUTPUT.PUT\_LINE('Sunt mai multe internari cu durata maxima');

RETURN NULL;

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nu există internări în baza de date.');

RETURN NULL;

END;

/

DECLARE

v\_pacient pacient.nume%TYPE;

BEGIN

v\_pacient := max\_durata\_internare;

IF v\_pacient IS NOT NULL THEN

DBMS\_OUTPUT.PUT\_LINE('Pacientul cu cea mai mare durata de internare este ' || v\_pacient || '.');

END IF;

END;

/

Graphical user interface, text, application, email

Description automatically generated

--2. Sa se afiseze durata angajarii ununi anumit doctor, al carui cod este dat de la tastatura

CREATE OR REPLACE FUNCTION Calcul\_vechime\_munca(p\_cod\_doctor IN doctor.cod\_doctor%TYPE)

RETURN number

AS

v\_vechime\_doctor number;

BEGIN

SELECT ROUND((SYSDATE-data\_angajare)/365) INTO v\_vechime\_doctor

FROM doctor

WHERE cod\_doctor = p\_cod\_doctor;

RETURN v\_vechime\_doctor;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN RETURN NULL;

END;

/

DECLARE

v\_vechime\_doctor number;

v\_cod\_tastat doctor.cod\_doctor%TYPE := &cod\_doctor;

BEGIN

v\_vechime\_doctor:=Calcul\_vechime\_munca(v\_cod\_tastat);

IF v\_vechime\_doctor IS NULL THEN

DBMS\_OUTPUT.PUT\_LINE('Nu exista doctor ce are codul introdus de dumneavoastra');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Doctorul cu codul '||v\_cod\_tastat ||' are vechimea de ' ||v\_vechime\_doctor || ' ani');

END IF;

END;

/

Graphical user interface, text, application, email

Description automatically generated

In cazul in care nu exista doctoral (ex codul 760)

Graphical user interface, text, application, email

Description automatically generated

--3.Construiti functia Calcul\_varsta care sa returneze varsta pacientului al carui cod este dat ca parametru de intrare. Tratati cazul în care angajatul indicat nu exista.

CREATE OR REPLACE FUNCTION Calcul\_varsta(p\_cnp IN pacient.cnp%TYPE)

RETURN number

AS

v\_varsta number;

BEGIN

SELECT ROUND((SYSDATE-data\_nastere)/365) INTO v\_varsta

FROM pacient

WHERE cnp = p\_cnp;

RETURN v\_varsta;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN RETURN NULL;

END;

/

DECLARE

v\_varsta number;

v\_cnp\_tastat pacient.cnp%type;

BEGIN

v\_cnp\_tastat := '&cnp';

v\_varsta:=Calcul\_varsta(v\_cnp\_tastat);

IF v\_varsta IS NULL THEN

DBMS\_OUTPUT.PUT\_LINE('Nu exista pacientul');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Pacientul ' ||v\_cnp\_tastat|| ' are varsta de '||v\_varsta || ' ani');

END IF;

EXCEPTION

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('Sunt prea multi pacienti cu acest cnp.');

END;

/Graphical user interface, text

Description automatically generated

In cazul in care pacientul nu exista:

Graphical user interface, text, application, email

Description automatically generated

In cazul in care sunt mai multi pacienti cu cnp identic:

Graphical user interface, text

Description automatically generated

--4. Sa se creeze o functe prin care sa se afiseze numele complet si doctorul asociat pacientului al carui cnp este introdus de la tastatura

SET SERVEROUTPUT ON SIZE 1000000;

CREATE OR REPLACE FUNCTION get\_numele\_complet(p\_cnp pacient.cnp%TYPE)

RETURN VARCHAR2

AS

v\_nume VARCHAR2(50);

v\_doctor VARCHAR2(50);

BEGIN

SELECT CONCAT(CONCAT(p.nume, ' '), p.prenume), dr.nume\_doctor INTO v\_nume, v\_doctor

FROM pacient p, doctor dr, diagnostic d

WHERE p.cnp = d.cnp

AND d.cod\_doctor = dr.cod\_doctor

AND p.cnp = p\_cnp;

RETURN v\_nume || ' | ' || v\_doctor;

EXCEPTION

WHEN TOO\_MANY\_ROWS THEN DBMS\_OUTPUT.PUT\_LINE('Exista mai mult de un pacient cu acest cnp.');

RETURN NULL;

WHEN NO\_DATA\_FOUND THEN DBMS\_OUTPUT.PUT\_LINE('Pacientul cu cnp-ul specificat nu exista.');

RETURN NULL;

END;

/

DECLARE

cnp\_tastat pacient.cnp%TYPE:='&cnp';

BEGIN

DBMS\_OUTPUT.PUT\_LINE(get\_numele\_complet(cnp\_tastat));

END;

Ex cnp existent 6854776390251

A screenshot of a computer

Description automatically generated with medium confidence

In cazul in care pacientul nu exista:

A screenshot of a computer

Description automatically generated with medium confidence

--5.Sa se afiseze salonul in care se afla pacientul al carui nume este introdus de la tastatura

CREATE OR REPLACE FUNCTION get\_salon(p\_nume IN pacient.nume%TYPE)

RETURN NUMBER

AS

v\_salon NUMBER;

BEGIN

SELECT i.salon INTO v\_salon

FROM internari i,diagnostic d, pacient p

WHERE i.cod\_diagnostic=d.cod\_diagnostic

AND

p.cnp=d.cnp

AND

p.nume=p\_nume;

RETURN v\_salon;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN RETURN NULL;

WHEN TOO\_MANY\_ROWS THEN RETURN NULL;

END;

/

DECLARE

v\_salon number;

v\_nume\_tastat pacient.nume%type;

BEGIN

v\_nume\_tastat := '&nume';

v\_salon:=get\_salon(v\_nume\_tastat);

IF v\_salon IS NULL THEN

DBMS\_OUTPUT.PUT\_LINE('Nu exista pacient internat cu numele specificat de dumneavoastra');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Pacientul '||v\_nume\_tastat||' este internat in salonul '||v\_salon);

END IF;

END;

/

Graphical user interface, text, application, email

Description automatically generated

In cazul in care pacientul nu exista:

Graphical user interface, text, application, email

Description automatically generated

**Proceduri**

--1. Scrieti procedura specializare\_doctor prin care sa se afiseze specializarea si numarul pacientilor doctorului al carui cod\_doctor este introdus de la tastatura.

CREATE OR REPLACE PROCEDURE specializare\_doctor(p\_cod\_doctor IN doctor.cod\_doctor%TYPE) IS

v\_specializare doctor.specializare%TYPE;

v\_numar\_pacienti NUMBER;

BEGIN

SELECT specializare INTO v\_specializare

FROM doctor

WHERE cod\_doctor = p\_cod\_doctor;

SELECT COUNT(\*) INTO v\_numar\_pacienti

FROM diagnostic

WHERE cod\_doctor = p\_cod\_doctor;

DBMS\_OUTPUT.PUT\_LINE('Doctorul cu codul ' || p\_cod\_doctor || ' are specializarea ' || v\_specializare || ' si are ' || v\_numar\_pacienti || ' pacienti.');

EXCEPTION

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('Sunt mai multi doctori cu codul ' || p\_cod\_doctor || '.');

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nu exista un doctor cu codul ' || p\_cod\_doctor || '.');

END;

/

DECLARE

v\_cod\_doc doctor.cod\_doctor%type:=&cod;

BEGIN

specializare\_doctor(v\_cod\_doc);

END;

/Graphical user interface, text, application

Description automatically generated

In cazul in care nu exista codul:

Graphical user interface, text, application

Description automatically generated

--2. Scrieti procedura Termen\_tratament care sa calculeze si sa afiseze durata fiecarui tratament într-un an indicat ca parametru de intrare. Apelati procedura.

SET SERVEROUTPUT ON;

CREATE OR REPLACE PROCEDURE Termen\_tratament(p\_an IN NUMBER)

AS

CURSOR c(p\_an NUMBER) IS

SELECT data\_terminare-data\_incepere durata

FROM tratament

WHERE EXTRACT(YEAR FROM data\_terminare)=p\_an;

r c%ROWTYPE;

anul\_nu\_exista EXCEPTION;

BEGIN

OPEN c (p\_an);

LOOP

FETCH c INTO r;

EXIT WHEN c%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Tratamentul a avut durata de '||r.durata||' zile.' );

END LOOP;

IF c%rowcount = 0 THEN

RAISE anul\_nu\_exista;

END IF;

CLOSE C;

EXCEPTION

WHEN anul\_nu\_exista THEN DBMS\_OUTPUT.PUT\_LINE('Nu s-au realizat tratamente in acel an!');

END;

/

EXECUTE Termen\_tratament(2021);

EXECUTE Termen\_tratament(1900);

Pentru anul 2021:

Graphical user interface, text, application

Description automatically generated

Pentru anul 1900:

Graphical user interface, text, application, email

Description automatically generated

--3. Sa se creeze o procedura prin care sa se modifice numele pacientului dat ca parametru si sa se insereze un nou pacient ale carui date sunt introduse de la tastatura

SET SERVEROUTPUT ON;

CREATE OR REPLACE PROCEDURE set\_pacient(p\_cnp IN pacient.cnp%TYPE)

AS

v\_cnp pacient.cnp%TYPE:= &cnp;

v\_nume pacient.nume%TYPE:='&nume';

v\_prenume pacient.prenume%TYPE:='&prenume';

v\_sex pacient.sex%TYPE:='&sex';

v\_data\_nastere pacient.data\_nastere%TYPE:=to\_date('&data\_nastere','dd-mm-yyyy');

v\_nume\_nou pacient.nume%TYPE:='&nume\_nou';

nu\_exista EXCEPTION;

BEGIN

INSERT INTO Pacient(CNP, nume, prenume, sex, data\_nastere) values (v\_cnp, v\_nume, v\_prenume, v\_sex, v\_data\_nastere);

DBMS\_OUTPUT.PUT\_LINE('S-a adaugat pacientul cu cnp: ' || v\_cnp||' numele ' || v\_nume || ' ' || v\_prenume ||' de sexul '|| v\_sex ||' nascut in data de '|| v\_data\_nastere);

UPDATE pacient

SET nume= v\_nume\_nou

WHERE cnp=p\_cnp;

IF SQL%ROWCOUNT = 0 THEN

RAISE nu\_exista;

ELSE

DBMS\_OUTPUT.PUT\_LINE('Numele pacientului a fost schimbat cu succes in '||v\_nume\_nou||'!');

END IF;

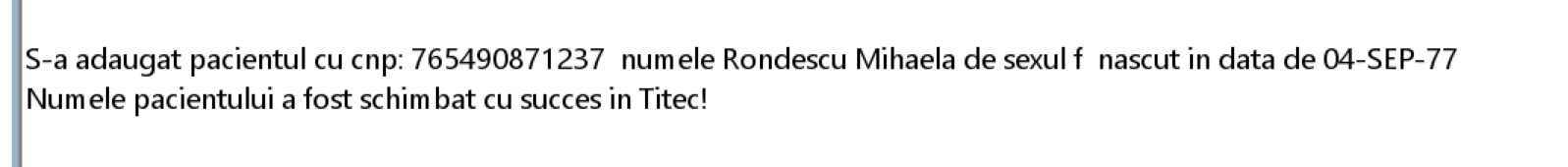
EXCEPTION

WHEN nu\_exista THEN DBMS\_OUTPUT.PUT\_LINE('Nu exista pacientul cu acest CNP');

END;

/

EXECUTE set\_pacient('1234567890123');



In cazul in care nu exista pacientul

Graphical user interface, text, application, email

Description automatically generated

--4. Apelati functia Calcul\_varsta în cadrul unei proceduri, Varsta\_pacienti, prin care se vor parcurge toti pacientii, în scopul afisarii varstei fiecaruia.

CREATE OR REPLACE PROCEDURE Varsta\_pacienti

AS

CURSOR c IS SELECT CONCAT(CONCAT(nume,' '), prenume) nume\_complet ,cnp, Calcul\_varsta(cnp) varsta FROM pacient;

r c%ROWTYPE;

atentie EXCEPTION;

BEGIN

OPEN c;

LOOP

FETCH c INTO r;

EXIT WHEN c%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(r.nume\_complet || ', ' || r.varsta);

END LOOP;

IF c%ROWCOUNT=0 THEN

RAISE atentie;

END IF;

CLOSE c;

EXCEPTION

WHEN atentie THEN DBMS\_OUTPUT.PUT\_LINE('Nu exista pacienti internati');

WHEN TOO\_MANY\_ROWS THEN DBMS\_OUTPUT.PUT\_LINE('Exista prea multi oameni cu acest cnp');

END;

/

EXECUTE Varsta\_pacienti;

Text

Description automatically generated

**--2 Pachete**

--pachetul 1

--Sa se creeze un pachet care sa contina:

-- -o procedura care sa afiseze numar de pacienti internati intr-un anumit an dat ca parametru

-- -o proceudra prin care pentru un anumit pacient sa se afiseze durata tratamentului pe care il are

-- -sa se trateze exceptiile posibile

SET SERVEROUTPUT ON;

CREATE OR REPLACE PACKAGE pachet1 AS

PROCEDURE get\_nr\_pacienti(p\_nr number);

PROCEDURE get\_zile\_tratament(p\_id IN pacient.cnp%TYPE);

END;

CREATE OR REPLACE PACKAGE BODY pachet1 AS

PROCEDURE get\_nr\_pacienti(p\_nr number)

AS

v\_nr\_pacienti NUMBER;

BEGIN

SELECT COUNT(p.cnp) INTO v\_nr\_pacienti

FROM pacient p, tratament t, diagnostic d

WHERE EXTRACT(YEAR FROM t.data\_terminare) = p\_nr

AND

p.cnp = d.cnp

AND

d.cod\_diagnostic = t.cod\_diagnostic;

DBMS\_OUTPUT.PUT\_LINE('In anul ales s-au internat'||v\_nr\_pacienti||'pacienti');

EXCEPTION

WHEN NO\_DATA\_FOUND THEN DBMS\_OUTPUT.PUT\_LINE('Nu s-au gasit date');

END;

PROCEDURE get\_zile\_tratament(p\_id IN pacient.cnp%TYPE)

IS

v\_zile NUMBER;

BEGIN

SELECT (t.data\_terminare-t.data\_incepere) INTO v\_zile

FROM tratament t, diagnostic d, pacient p

WHERE p.cnp=p\_id

AND

p.cnp = d.cnp

AND

d.cod\_diagnostic = t.cod\_diagnostic;

DBMS\_OUTPUT.PUT\_LINE(v\_zile);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN DBMS\_OUTPUT.PUT\_LINE('Nu exista pacientul');

END;

END pachet1;

/

BEGIN

pachet1.get\_nr\_pacienti(2021);

END;

BEGIN

pachet1.get\_zile\_tratament(5263485975214);

END;

A screenshot of a computer

Description automatically generated with medium confidence

Pentru anul 2021

A screenshot of a computer

Description automatically generated

Pentru anul 2024

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

SET SERVEROUTPUT ON;

CREATE OR REPLACE PACKAGE pachet2 AS

FUNCTION get\_nr\_pacienti\_afectiune(p\_cod IN afectiune.cod\_boala%TYPE) RETURN NUMBER;

PROCEDURE get\_nume\_pacient(p\_cod\_doctor IN doctor.cod\_doctor%TYPE);

END;

/

CREATE OR REPLACE PACKAGE BODY pachet2 AS

FUNCTION get\_nr\_pacienti\_afectiune(p\_cod IN afectiune.cod\_boala%TYPE) RETURN NUMBER AS

v\_nr\_pacienti NUMBER;

no\_pacienti exception;

BEGIN

SELECT COUNT(p.cnp)

INTO v\_nr\_pacienti

FROM pacient p

JOIN diagnostic d ON p.cnp = d.cnp

JOIN afectiune a ON a.cod\_boala = d.cod\_boala

WHERE a.cod\_boala = p\_cod;

if v\_nr\_pacienti > 0 then

RETURN v\_nr\_pacienti;

else

raise no\_pacienti;

end if;

EXCEPTION

WHEN no\_pacienti THEN

DBMS\_OUTPUT.PUT\_LINE('Nu s-au gasit pacienti.');

RETURN NULL;

END;

PROCEDURE get\_nume\_pacient(p\_cod\_doctor IN doctor.cod\_doctor%TYPE) AS

v\_nume\_pacient VARCHAR2(40);

BEGIN

SELECT CONCAT(CONCAT(p.nume, ' '), p.prenume)

INTO v\_nume\_pacient

FROM pacient p

JOIN diagnostic d ON p.cnp = d.cnp

JOIN doctor dr ON d.cod\_doctor = dr.cod\_doctor

WHERE dr.cod\_doctor = p\_cod\_doctor;

DBMS\_OUTPUT.PUT\_LINE('Numele pacientului este ' || v\_nume\_pacient);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN DBMS\_OUTPUT.PUT\_LINE('Nu s-a găsit pacientul');

END;

END pachet2;

/

DECLARE

RESULT NUMBER;

BEGIN

RESULT := pachet2.get\_nr\_pacienti\_afectiune(120);

DBMS\_OUTPUT.PUT\_LINE(RESULT || ' pacienti');

END;

BEGIN

pachet2.get\_nume\_pacient(6964);

END;

A screenshot of a computer program

Description automatically generated with low confidence

In caz de eroare:

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer error

Description automatically generated with medium confidence

**-3 declansatori**

--1. Să se creeze un declanșator care previne introducerea pacienților mai mici de un an.

CREATE OR REPLACE TRIGGER check\_patient\_age

BEFORE INSERT ON PACIENT

FOR EACH ROW

DECLARE

v\_age NUMBER;

BEGIN

v\_age := TRUNC(MONTHS\_BETWEEN(SYSDATE, :NEW.DATA\_NASTERE) / 12);

IF v\_age < 1 THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Nu pot fi introdusi pacienti cu varsta mai mica de un an');

END IF;

END;

/

A close-up of a logo

Description automatically generated with medium confidence

INSERT INTO Pacient(CNP, nume, prenume, sex, data\_nastere) values ('5263485970005', 'Petrescu', 'Marcel', 'm', to\_date('21-02-2023', 'dd-mm-yyyy'));

A picture containing text, font, screenshot

Description automatically generated

--2. Sa se creeze un declansator care sa nu permita introducerea unui cnp cu o lungime diferita de 13 cifre

CREATE OR REPLACE TRIGGER nr\_cnp

BEFORE INSERT OR UPDATE OF cnp ON pacient

FOR EACH ROW

BEGIN

IF ( LENGTH(:new.cnp) != 4) THEN

RAISE\_APPLICATION\_ERROR(-20000,'CNP invalid! Va rugam sa verificati ca lungimea este corecta.');

END IF;

END;

/

A picture containing text, font, line, screenshot

Description automatically generated

INSERT INTO Pacient(CNP, nume, prenume, sex, data\_nastere) values ('5263485975', 'Petrescu', 'Marcel', 'm', to\_date('21-09-1968', 'dd-mm-yyyy'));

A picture containing text, font, screenshot, white

Description automatically generated

--3. Să se creeze un declanșator care să nu permită introducerea unei date de terminare a tratamentului care este înaintea celei de începere.

set serveroutput on;

CREATE OR REPLACE TRIGGER diferenta\_date

BEFORE UPDATE OR INSERT ON Tratament

FOR EACH ROW

WHEN (NEW.data\_terminare <NEW.data\_incepere)

BEGIN

RAISE\_APPLICATION\_ERROR(-20001, 'Data de terminare a tratamentului nu poate sa fie mai devreme de data inceperii!');

END;

/

A black text on a white background

Description automatically generated with medium confidence

INSERT INTO Tratament values ('14283' , to\_date('20-12-2023', 'dd-mm-yyyy'), to\_date('25-12-2021', 'dd-mm-yyyy'),140 ,'2568');

A picture containing text, font, screenshot

Description automatically generated