Facultatea de Cibernetica, Statistica si Informatica Economica

***SISTEM DE GESTIUNE A UNEI FARMACII***

Student : Boboc Ioana

Grupa : 1052

Seria : C

Anul : II

**A.**

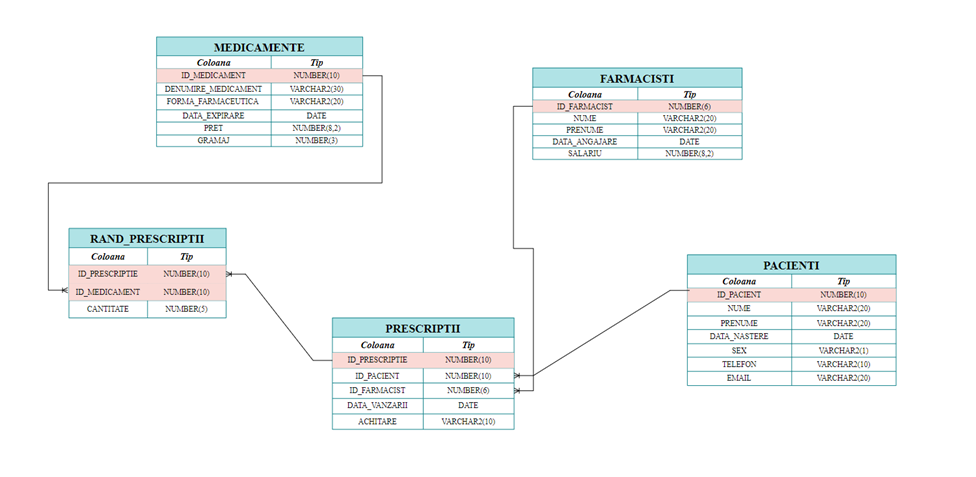
1. *Descrierea temei*

Proiectul realizat pe tema aleasa, Sistemul de gestiune a unei farmacii, prezinta o baza de date proiectata pentru a eficientiza si monitoriza operatiunile desfasurate in cadrul unei farmacii, atat din punct de vedere medical, cat si economic. In tabele sunt pastrate informatii relevante, referitoare la medicamentele ce se gasesc in farmacie, farmacistii aflati la dispozitia pacientilor ce au nevoie sa cumpere produsele notate in prescriptiile primite de acestia, preturi cantitati si altele.

Astfel, personalul farmaciei este gestionat in tabela “Farmacisti”, unde se gasesc date referitoare la id, nume, prenume, data angajarii, salariu. Despre pacienti sunt pastrate informatii precum id, nume, prenume, data nasterii, sex si alte date personale ale acestora (telefon si email). In ceea ce priveste medicamentele, suntem informati despre id, denumire, forma farmaceutica in care se gasesc, data expirarii, pretul si , binenteles, gramajul. Prescriptiile primite de catre pacienti inregistreaza cate un id propriu, pe cel al pacientului, cel al farmacistului care a ajutat clientul, data vanzarii produselor farmaceutice si modalitatea de achitare aleasa, iar in tabela Rand\_Prescriptii includem cantitatea fiecarui medicament necesar cumparat de fiecare pacient.

Din perspectiva economica, baza de date contine informatii legate de preturi, cantitati, salarii si alte aspecte financiare. Pot fi apelate comenzi pentru determinarea venitul obtinut din vanzarea fiecarui medicament cerut, ceea ce ar putea permite ulterior realizarea unor rapoarte financiare, analizarea performantei vanzarilor, observarea produselor cele mai bine vandute, monitorizarea eficientei personalului si altele. Prin intermediul acestei baze de date, farmacia va contribui la furnizarea unor servicii medicale de inalta calitate intr-un mediu economic sustenabil.

1. *Schema conceptuala pentru modelarea temei alese*



Schema conceptuala de pe semestrul I rămâne neschimbata în semestrul II.

**B.**

Interacțiunea cu serverul **Oracle** prin intermediul comenzilor SQL (LDD şi LMD): folosirea lui *execute immediate*, particularități cu privire la folosirea comenzii *select* precum și a funților la nivel de rând respectiv de grup.

1. Sa se afiseze numele studentului.

declare

v\_nume varchar2(30);

v\_prenume varchar2(30);

v\_data\_angajare farmacisti.data\_angajare%type;

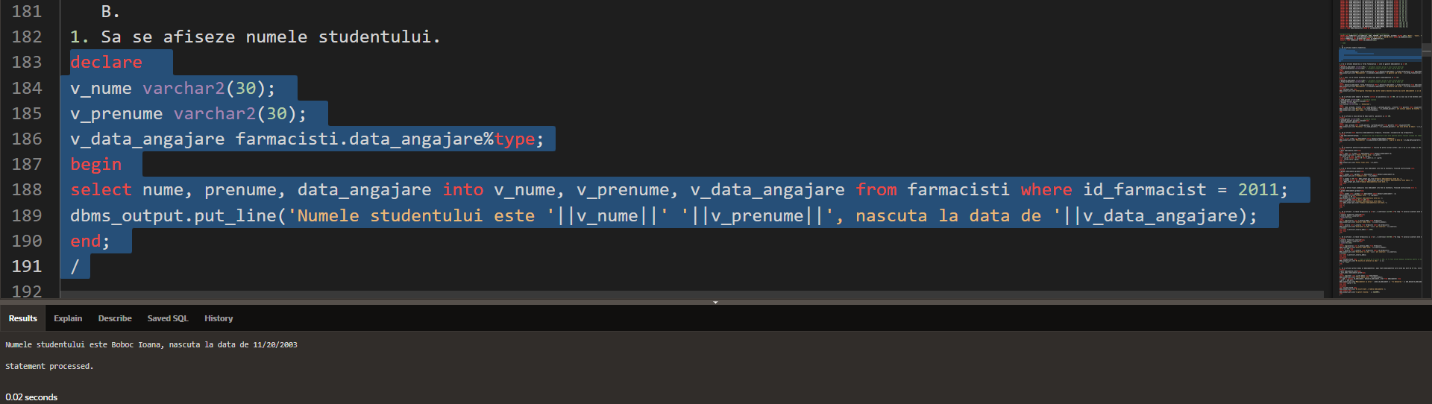
begin

select nume, prenume, data\_angajare into v\_nume, v\_prenume, v\_data\_angajare from farmacisti where id\_farmacist = 2011;

dbms\_output.put\_line('Numele studentului este '||v\_nume||' '||v\_prenume||', nascuta la data de '||v\_data\_angajare);

end;

/



2.1 Sa se afiseze denumirea si forma farmaceutica in care se gaseste medicamentul cu id 20.

declare

v\_denumire\_medicament varchar2(30);

v\_forma\_farmaceutica varchar2(20);

begin

select denumire\_medicament, forma\_farmaceutica INTO v\_denumire\_medicament, v\_forma\_farmaceutica from medicamente where id\_medicament=20;

dbms\_output.put\_line('Medicamentul '||v\_denumire\_medicament||' se gaseste sub forma: '||v\_forma\_farmaceutica);

end;

/

2.2 In plus, sa se arunce exceptia too\_many\_rows pentru medicamentele cu id >20.

declare

v\_denumire\_medicament varchar2(30); -- variabila scalara pl/sql a unui tip de date sql

v\_forma\_farmaceutica varchar2(20); -- variabila scalara pl/sql a unui tip de date sql

begin

select denumire\_medicament, forma\_farmaceutica INTO v\_denumire\_medicament, v\_forma\_farmaceutica from medicamente where id\_medicament>20;

dbms\_output.put\_line('Medicamentul '||v\_denumire\_medicament||' se gaseste sub forma: '||v\_forma\_farmaceutica);

exception

when too\_many\_rows then

dbms\_output.put\_line('Interogarea returneaza mai multe randuri deoarece exista mai multe medicamente cu un id mai mare de 20.');

end;

/

3. Sa se afiseze atat numarul de telefon initial al pacientului cu id= 400, cat si noul sau nr de telefon introdus (constant). declare

v\_nume\_pacient varchar2(20); -- variabila scalara

v\_prenume\_pacient pacienti.prenume%type;

v\_telefon varchar2(10);

v\_nr\_telefon constant varchar2(10) := '0744217567';

begin

select nume, prenume, telefon INTO v\_nume\_pacient, v\_prenume\_pacient, v\_telefon from pacienti where id\_pacient=400;

dbms\_output.put\_line('Pacientul '||v\_nume\_pacient||' '||v\_prenume\_pacient||' are initial numarul de telefon '||v\_telefon);

dbms\_output.put\_line('Noul numar: '||v\_nr\_telefon);

end;

/

4. Sa se afiseze o noua adresa de email pentru pacientul cu id= 300.

declare

v\_nume\_pacient varchar2(20); -- variabila scalara

v\_prenume\_pacient pacienti.prenume%type;

v\_email pacienti.email%type;

begin

select nume, prenume INTO v\_nume\_pacient, v\_prenume\_pacient from pacienti where id\_pacient=300;

dbms\_output.put\_line('Pacientul '||v\_nume\_pacient||' '||v\_prenume\_pacient||' are noua adresa de email: '||:v\_email);

end;

/

5. Sa se afiseze data expirarii medicamentului strepsils, folosind o variabila de tip inregistrare.

declare

v\_emp medicamente%rowtype; -- variabila de tip inregistrare care este apelata pentru fiecare coloana din tabela medicamente

begin

select \* into v\_emp from medicamente where denumire\_medicament='STREPSILS';

dbms\_output.put\_line('Medicamentul '||v\_emp.denumire\_medicament||' expira la data de '||v\_emp.data\_expirare);

end;

/

**C.**

Structuri alternative și repetitive.

1. Sa se modifice preturile medicamentelor in functie de pretul actual astfel: cele < 25 se vor scumpi cu 20%, cele > 25 si < 30 se vor scumpi cu 10%, iar cele de peste 30 se vor ieftini cu 10%, folosind instructiunea if.

declare

v\_pret medicamente.pret%type;

begin

select pret into v\_pret from medicamente where denumire\_medicament=:m;

dbms\_output.put\_line ('Pretul initial este: '||v\_pret);

if v\_pret < 25 then v\_pret:=1.2\* v\_pret;

elsif v\_pret between 25 and 30 then v\_pret:=1.1 \* v\_pret;

else v\_pret:=0.9\* v\_pret;

end if;

dbms\_output.put\_line('Pretul final este: '||v\_pret);

end;

/

2.1 Sa se afisze tipul gramajului unui medicament citit de la tastatura, folosind instructiunea case.

declare

v\_gramaj medicamente.gramaj%type;

begin

select gramaj into v\_gramaj from medicamente where denumire\_medicament=:m;

dbms\_output.put\_line ('Gramajul este: '||v\_gramaj);

case

when v\_gramaj <= 50 then dbms\_output.put\_line('Gramajul medicamentului este mic.');

when v\_gramaj between 51 and 150 then dbms\_output.put\_line('Gramajul medicamentului este mediu.');

else dbms\_output.put\_line('Gramajul medicamentului este mare.');

end case;

end;

/

2.2 Sa se afisze tipul gramajului unui medicament citit de la tastatura, folosind instructiunea else if.

declare

v\_gramaj medicamente.gramaj%type;

begin

select gramaj into v\_gramaj from medicamente where denumire\_medicament = :m;

dbms\_output.put\_line('Gramajul este: ' || v\_gramaj);

if v\_gramaj <= 50 then

dbms\_output.put\_line('Gramajul medicamentului este mic.');

else if v\_gramaj between 51 and 150 then

dbms\_output.put\_line('Gramajul medicamentului este mediu.');

else dbms\_output.put\_line('Gramajul medicamentului este mare.');

end if;

end if;

end;

/

3. Sa se afiseze în ordine farmacistii cu id-uri în intervalul 111-444 atâta timp cât salariul acestora este mai mic decât media, folosindu-se loop.

declare

v\_salariu farmacisti.salariu%type;

v\_salariu\_mediu v\_salariu%type;

i number(4):=111;

begin

select avg(salariu) into v\_salariu\_mediu from farmacisti;

dbms\_output.put\_line('Salariul mediu este: '||v\_salariu\_mediu);

loop

select salariu into v\_salariu from farmacisti where id\_farmacist=i;

dbms\_output.put\_line('Salariatul cu idul '||i||' are salariul: '||v\_salariu);

i:=i+111;

exit when v\_salariu<v\_salariu\_mediu or i>444;

end loop;

end;

/

4. Sa se afiseze în ordine farmacistii cu id-uri în intervalul 333-666 atâta timp cât salariul acestora este mai mic decât media, folosindu-se while.

declare

v\_salariu farmacisti.salariu%type;

v\_salariu\_mediu v\_salariu%type;

i number(4):=333;

begin

select avg(salariu) into v\_salariu\_mediu from farmacisti;

dbms\_output.put\_line('Salariul mediu este: '||v\_salariu\_mediu);

while i < 777 loop

select salariu into v\_salariu from farmacisti where id\_farmacist=i;

dbms\_output.put\_line('Salariatul cu idul '||i||' are salariul: '||v\_salariu);

i:=i+111;

exit when v\_salariu<v\_salariu\_mediu;

end loop;

exception

when no\_data\_found then -- daca in while as fi scris i< 888, ar fi fost afisat mesajul exceptiei pentru ca nu exista un farmacist cu id= 777

dbms\_output.put\_line('Nu exista un salariat cu idul ' || i);

end;

/

5. Sa se afiseze pretul mediu al medicamentelor, apoi toate medicamentele care costa mai mult de 10 lei, utilizand structura repetitiva for.

declare

v\_pret medicamente.pret%type;

v\_pret\_mediu medicamente.pret%type;

begin

select avg(pret) into v\_pret\_mediu from medicamente;

dbms\_output.put\_line('Pretul mediu este: '||v\_pret\_mediu);

for med in (select id\_medicament, denumire\_medicament, pret from medicamente) loop

v\_pret := med.pret;

dbms\_output.put\_line('Medicamentul cu id-ul ' ||med.id\_medicament || ' și denumirea ' || med.denumire\_medicament || ' are pretul: ' || v\_pret);

exit when v\_pret < 10;

end loop;

exception

when no\_data\_found then

dbms\_output.put\_line('Nu exista date in tabela medicamente.');

when others then

dbms\_output.put\_line('A aparut eroarea: ' || SQLERRM);

end;

/

**D.**

Colectii de date (index by table, nested table, varray).

1. Sa se utilizeze intr-un bloc pl/sql o inregistrare pentru tabela farmacisti.

declare

type t\_emp is record( -- user defined record type

prenume varchar2(25),

nume varchar2(25),

data\_angajare date

);

v\_emp t\_emp;

begin

select prenume, nume, data\_angajare into v\_emp from farmacisti where id\_farmacist=888;

dbms\_output.put\_line('Farmacistul '||v\_emp.prenume||' '||v\_emp.nume||' este angajat in data de '||v\_emp.data\_angajare);

end;

/

2. Sa se afiseze numarul de medicamente, apoi id-ul, denumirea si pretul fiecaruia, folosindu-se colectii de tip index by table.

declare

type t\_preturi is table of medicamente.pret%type index by pls\_integer;

type t\_denumiri is table of medicamente.denumire\_medicament%type index by pls\_integer;

v\_preturi t\_preturi;

v\_denumiri t\_denumiri;

v\_contor pls\_integer;

begin

for med in (select id\_medicament, denumire\_medicament, pret from medicamente) loop

v\_preturi(med.id\_medicament) := med.pret;

v\_denumiri(med.id\_medicament) := med.denumire\_medicament;

end loop;

v\_contor := v\_preturi.count;

dbms\_output.put\_line('Numarul de elemente din colectie: ' || v\_contor);

for i in v\_preturi.first .. v\_preturi.last

loop

if v\_preturi.exists(i) then

dbms\_output.put\_line('Id Medicament: '|| i ||', Denumire: '||v\_denumiri(i)||', Pret: '|| v\_preturi(i));

end if;

end loop;

exception

when others then

dbms\_output.put\_line('A aparut o eroare: ' || SQLERRM);

end;

/

3. Sa se afiseze suma salariilor tuturor farmacistilor, apoi primii 4 farmacisti atat cu for, cat si cu while, folosindu-se o colectie de tip nested table.

declare

type t\_lista\_farmacisti is table of varchar2(40);

lista\_farmacisti t\_lista\_farmacisti;

suma\_salariala number := 0;

begin

select nume || ' ' || prenume bulk collect into lista\_farmacisti from farmacisti

where rownum <= 4;

select sum(salariu) into suma\_salariala from farmacisti;

dbms\_output.put\_line('Suma salariilor tuturor farmacistilor: ' || suma\_salariala);

dbms\_output.put\_line('Parcurgere folosind FOR:');

for i in 1..lista\_farmacisti.count loop

dbms\_output.put\_line('Farmacist ' || i\*111 || ': ' || lista\_farmacisti(i));

end loop;

dbms\_output.put\_line('Parcurgere folosind WHILE:');

declare

j number := lista\_farmacisti.first;

begin

while j is not null loop

dbms\_output.put\_line('Farmacist ' || j\*111 || ': ' || lista\_farmacisti(j));

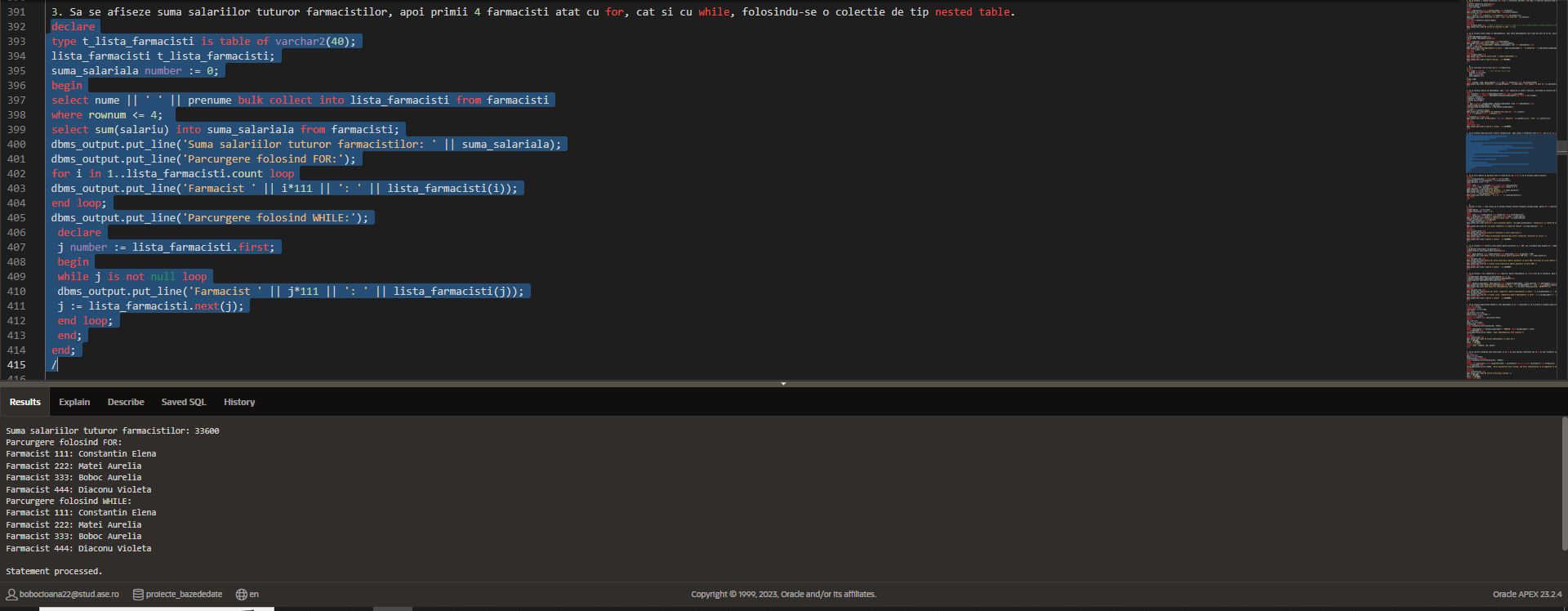
j := lista\_farmacisti.next(j);

end loop;

end;

end;

/



4. Sa se afle numarul de pacienti dintr-o colectie de tip varray si sa se afiseze numele acestora.

declare

type t\_lista\_pacienti is varray(10) of varchar2(50);

lista\_pacienti t\_lista\_pacienti := t\_lista\_pacienti();

numar\_pacienti number := 0;

begin

select nume || ' ' || prenume bulk collect into lista\_pacienti

from ( select nume, prenume from pacienti where rownum <= 6 );

numar\_pacienti := lista\_pacienti.count;

dbms\_output.put\_line('Numarul de pacienti: ' || numar\_pacienti);

dbms\_output.put\_line('Parcurgere folosind for:');

for i in 1 .. lista\_pacienti.count loop

dbms\_output.put\_line('Pacient ' || i || ': ' || lista\_pacienti(i));

end loop;

end;

/

**E.**

Tratarea excepțiilor. (minim 3 implicite, 2 explicite).

1. Scrieti un bloc in urma caruia sa se afiseze mesajul aferent exceptiei no\_data\_found, pentru un id inexistent in baza de date.

declare

v\_nume\_familie varchar2(20);

v\_numar\_actualizati number := 0;

begin

select nume into v\_nume\_familie from farmacisti where id\_farmacist=54;

update farmacisti set salariu = salariu\*1.5 where nume = v\_nume\_familie;

dbms\_output.put\_line('Numele de familie cautat este: '||v\_nume\_familie);

v\_numar\_actualizati := sql%rowcount;

if v\_numar\_actualizati > 0 then

dbms\_output.put\_line('Salariul a fost actualizat pentru '||v\_numar\_actualizati||' farmacist(i) cu numele de familie '||v\_nume\_familie|| '.');

else

dbms\_output.put\_line('Nu s-au gasit farmacisti cu numele de familie '||v\_nume\_familie|| '.');

end if;

exception

when no\_data\_found then

dbms\_output.put\_line('Nu exista un farmacist cu id-ul specificat');

when too\_many\_rows then

dbms\_output.put\_line('Trebuie actualizate salariile mai multor farmacisti. Utilizati un cursor.');

when others then

dbms\_output.put\_line('A aparut o eroare: ' || SQLERRM);

end;

/

2. Sa se afiseze data vanzarii prescriptiei pentru pacientul cu id 800, sau o exceptie daca aceasta nu a cumparat de la farmacie, sau a cumparat mai multe medicamente in zile diferite.

declare

v\_id\_pacient prescriptii.id\_pacient%type;

v\_data\_vanzarii prescriptii.data\_vanzarii%type;

begin

select data\_vanzarii into v\_data\_vanzarii from prescriptii where id\_pacient = 800;

dbms\_output.put\_line('Data vanzării prescriptiei pentru pacientul 800 este: ' || v\_data\_vanzarii);

exception

when too\_many\_rows then

dbms\_output.put\_line('Exista mai multe prescriptii pentru pacientul cu Id-ul 800. Utilizati un cursor pentru selectie.');

when no\_data\_found then

dbms\_output.put\_line('Nu s-a gasit nicio prescriptie pentru pacientul cu Id-ul 800.');

when others then

dbms\_output.put\_line('A aparut o eroare: ' || SQLERRM);

end;

/

3. Sa se afiseze id-ul, denumirea si data expirarii pentru medicamentul cu id-ul citit de la tastatura, daca expira inainte de 5 mai 2026. Sa se afiseze o exceptie daca este necesar.

declare

v\_id\_medicament medicamente.id\_medicament%type := :d;

v\_denumire\_medicament medicamente.denumire\_medicament%type;

v\_data\_expirare medicamente.data\_expirare%type;

begin

select denumire\_medicament, data\_expirare into v\_denumire\_medicament, v\_data\_expirare from medicamente where id\_medicament = v\_id\_medicament and data\_expirare < to\_date('05/05/2026', 'DD/MM/YYYY');

dbms\_output.put\_line('Denumirea medicamentului cu ID-ul ' || v\_id\_medicament || ' este: ' || v\_denumire\_medicament);

dbms\_output.put\_line('Data expirarii medicamentului este: ' || to\_char(v\_data\_expirare, 'DD/MM/YYYY'));

exception

when too\_many\_rows then

dbms\_output.put\_line('Exista mai multe inregistrari pentru medicamentul cu Id-ul ' || v\_id\_medicament || '. Utilizati un cursor pentru selectie.');

when no\_data\_found then

dbms\_output.put\_line('Nu s-a gasit nicio inregistrare pentru medicamentul cu Id-ul ' || v\_id\_medicament || ' sau data de expirare este mai mare sau egala cu 05/05/2026.');

when others then

dbms\_output.put\_line('A aparut o eroare: ' || SQLERRM);

end;

/

4. Sa se incerce modificarea denumirii unui medicament cu un id inexistent si sa se arunce o exceptie explicita.

drop table erori;

create table erori

(utilizator varchar2(40),

data date,

cod\_eroare number(10),

mesaj\_eroare varchar2(255) );

select \* from erori;

delete from erori where cod\_eroare=-20111;

declare

cod number(7);

mesaj varchar2(255);

invalid\_med exception;

pragma exception\_init(invalid\_med, -20222);

begin

update medicamente set denumire\_medicament = 'NUROFEN' where id\_medicament = 33333;

if sql%notfound then

raise\_application\_error(-20222, 'Codul medicamentului este invalid!');

end if;

exception

when invalid\_med then

dbms\_output.put\_line('Nu exista medicamentul cu acest ID');

cod := SQLCODE;

mesaj := SQLERRM;

insert into erori

values (user, sysdate, cod, mesaj);

end;

/

5. Sa se incerce stergerea unei prescriptii cu un id al unui pacient inexistent sau un id al unui farmacist care nu se regaseste in tabela si sa se arunce o exceptie explicita.

declare

cod number(7);

mesaj varchar2(255);

invalid\_presc exception;

pragma exception\_init(invalid\_presc, -20202);

begin

delete from prescriptii where id\_pacient=1919 or id\_farmacist not in ( select id\_farmacist from farmacisti);

if sql%notfound then

raise\_application\_error(-20202, 'Id-ul pacientului este invalid, sau id-ul farmacistului nu se regaseste in tabela Farmacisti.');

end if;

exception

when invalid\_presc then

dbms\_output.put\_line('Nu exista prescriptia cautata.');

cod := SQLCODE;

mesaj := SQLERRM;

insert into erori

values (user, sysdate, cod, mesaj);

end;

/

6. Sa se dubleze cantitatea unui medicament dintr-o prescriptie si sa se arunce o exceptie daca acel medicament nu se gaseste.

declare

cod number(7);

mesaj varchar2(255);

invalid\_rd\_presc exception;

pragma exception\_init(invalid\_rd\_presc, -20111);

begin

update rand\_prescriptii set cantitate= cantitate\*2 where id\_medicament= 100;

if sql%notfound then

raise\_application\_error(-20111, 'Codul medicamentului cautat nu se gaseste in nicio prescriptie.');

end if;

exception

when invalid\_rd\_presc then

dbms\_output.put\_line('Nu exista medicamentul cautat in aceasta prescriptie.');

cod := SQLCODE;

mesaj := SQLERRM;

insert into erori

values (user, sysdate, cod, mesaj);

end;

/

**F.**

Gestionarea cursorilor: impliciți și expliciți (cu și fără parametrii)

1. Sa se afiseze utilizand un cursor implicit toate medicamentele cumparate de catre pacientul cu id 800, si data vanzarii acestora.

declare

v\_id\_pacient prescriptii.id\_pacient%type := 800;

v\_data\_vanzarii prescriptii.data\_vanzarii%type;

v\_denumire\_medicament medicamente.denumire\_medicament%type;

begin

for c\_prescriptie in (select P.data\_vanzarii, M.denumire\_medicament

from prescriptii P

join rand\_prescriptii RP on P.id\_prescriptie = RP.id\_prescriptie

join medicamente M ON RP.id\_medicament = M.id\_medicament

where P.id\_pacient = v\_id\_pacient) loop

v\_data\_vanzarii := c\_prescriptie.data\_vanzarii;

v\_denumire\_medicament := c\_prescriptie.denumire\_medicament;

dbms\_output.put\_line('Data vânzării prescriptiei pentru pacientul ' || v\_id\_pacient || ' este: ' || v\_data\_vanzarii);

dbms\_output.put\_line('Denumirea medicamentului: ' || v\_denumire\_medicament);

end loop;

exception

when no\_data\_found then

dbms\_output.put\_line('Nu s-a gasit nicio prescriptie pentru pacientul cu ID-ul ' || v\_id\_pacient || '.');

when others then

dbms\_output.put\_line('A aparut o eroare: ' || SQLERRM);

end;

/

2. Sa se utilizeze un cursor implicit pentru a afisa farmacistii cu salariul de 4200 lei.

declare

v\_id\_farmacist number;

v\_nume\_farmacist varchar2(30);

begin

for c\_farmacist in (select id\_farmacist, nume from farmacisti where salariu= 4200) loop

v\_id\_farmacist := c\_farmacist.id\_farmacist;

v\_nume\_farmacist := c\_farmacist.nume;

dbms\_output.put\_line('Id-ul farmacistului: ' || v\_id\_farmacist);

dbms\_output.put\_line('Numele farmacistului: ' || v\_nume\_farmacist);

end loop;

exception

when no\_data\_found then

dbms\_output.put\_line('Nu s-a gasit niciun farmacist.');

when others then

dbms\_output.put\_line('A aparut o eroare: ' || SQLERRM);

end;

/

3. Folosind un cursor implicit, sa se dubleze preturile medicamentelor proba1, proba2, proba3, apoi acestea sa fie sterse din tabela medicamente.

declare

n number:=0;

begin

update medicamente set pret=pret\*2 where id\_medicament>650;

if sql%rowcount>0 then dbms\_output.put\_line('S-au marit preturile');

else dbms\_output.put\_line('Nu s-au marit preturile');

end if;

delete from medicamente M where M.id\_medicament not in (select id\_medicament from rand\_prescriptii);

if sql%rowcount>0 then dbms\_output.put\_line('S-au sters medicamente');

else dbms\_output.put\_line('Nu s-au sters medicamente');

end if;

exception

when others then dbms\_output.put\_line('Nu s-a sters niciun medicament->'||SQLERRM);

end;

/

4. Sa se afiseze pentru toate medicamentele id, denumire, pret si forma farmaceutica, ordonate descrescator in functie de pret, folosind un cursor explicit.

declare

cursor c is select id\_medicament, denumire\_medicament, pret, forma\_farmaceutica from medicamente order by pret desc;

r c%rowtype;

begin

if not c%isopen then

open c;

end if;

loop

fetch c into r;

exit when c%notfound;

dbms\_output.put('Id-ul medicamentului: ' || r.id\_medicament);

dbms\_output.put(', Denumirea medicamentului: ' || r.denumire\_medicament);

dbms\_output.put(', Pretul medicamentului: ' || r.pret);

dbms\_output.put(', Forma farmaceutica: ' || r.forma\_farmaceutica);

dbms\_output.put\_line(' ');

end loop;

close c;

exception

/\*when cursor\_already\_open then

dbms\_output.put\_line('Cursorul este deja deschis');\*/

when others then

dbms\_output.put\_line('A aparut o eroare: ' || SQLERRM);

end;

/

5. Folosind un cursor explicit, sa se afiseze pentru fiecare prescriptie, valoarea totala a fiecarui medicament cumparat.

declare

cursor c is

select RP.id\_prescriptie, M.denumire\_medicament, sum(M.pret \* RP.cantitate) as valoare\_prescriptie from rand\_prescriptii RP join medicamente M on RP.id\_medicament = M.id\_medicament

group by RP.id\_prescriptie, M.denumire\_medicament

order by RP.id\_prescriptie;

begin

--open c;

for r in c loop

dbms\_output.put\_line('Prescriptia Id: ' || r.id\_prescriptie || ' Medicament: ' || r.denumire\_medicament || ' Valoarea totala: ' || r.valoare\_prescriptie);

end loop;

exception

when cursor\_already\_open then

dbms\_output.put\_line('Cursorul este deja deschis.');

when others then

dbms\_output.put\_line('A aparut o eroare: ' || SQLERRM);

end;

/

6. Sa se afiseze (in ordine descrescatoare in functie de total\_vanzari) medicamentele al caror pret total vandut este mai mare decat valoarea totala a vanzarilor pentru un anumit medicament. ( id 17= 'PARACETAMOL')

declare

cursor c\_med (p\_val number) is

select m.id\_medicament, m.denumire\_medicament, sum(m.pret \* rp.cantitate) as total\_vanzari from medicamente m join rand\_prescriptii rp on m.id\_medicament = rp.id\_medicament

group by m.id\_medicament, m.denumire\_medicament

having sum(m.pret \* rp.cantitate) > p\_val

order by total\_vanzari desc;

v\_val number;

rec\_med c\_med%rowtype;

begin

select sum(m.pret \* rp.cantitate) into v\_val from rand\_prescriptii rp, medicamente m where m.id\_medicament=rp.id\_medicament and m.id\_medicament = 17;

dbms\_output.put\_line('Medicamentele a caror valoare totala vanduta este mai mare decat '||v\_val||' sunt:');

if not c\_med%isopen then

open c\_med(v\_val);

end if;

loop

fetch c\_med into rec\_med;

exit when c\_med%notfound;

dbms\_output.put\_line('Medicamentul Id: ' || rec\_med.id\_medicament || ', Denumire: ' || rec\_med.denumire\_medicament || ', Valoare totala vanduta: ' || rec\_med.total\_vanzari || ' lei.');

end loop;

close c\_med;

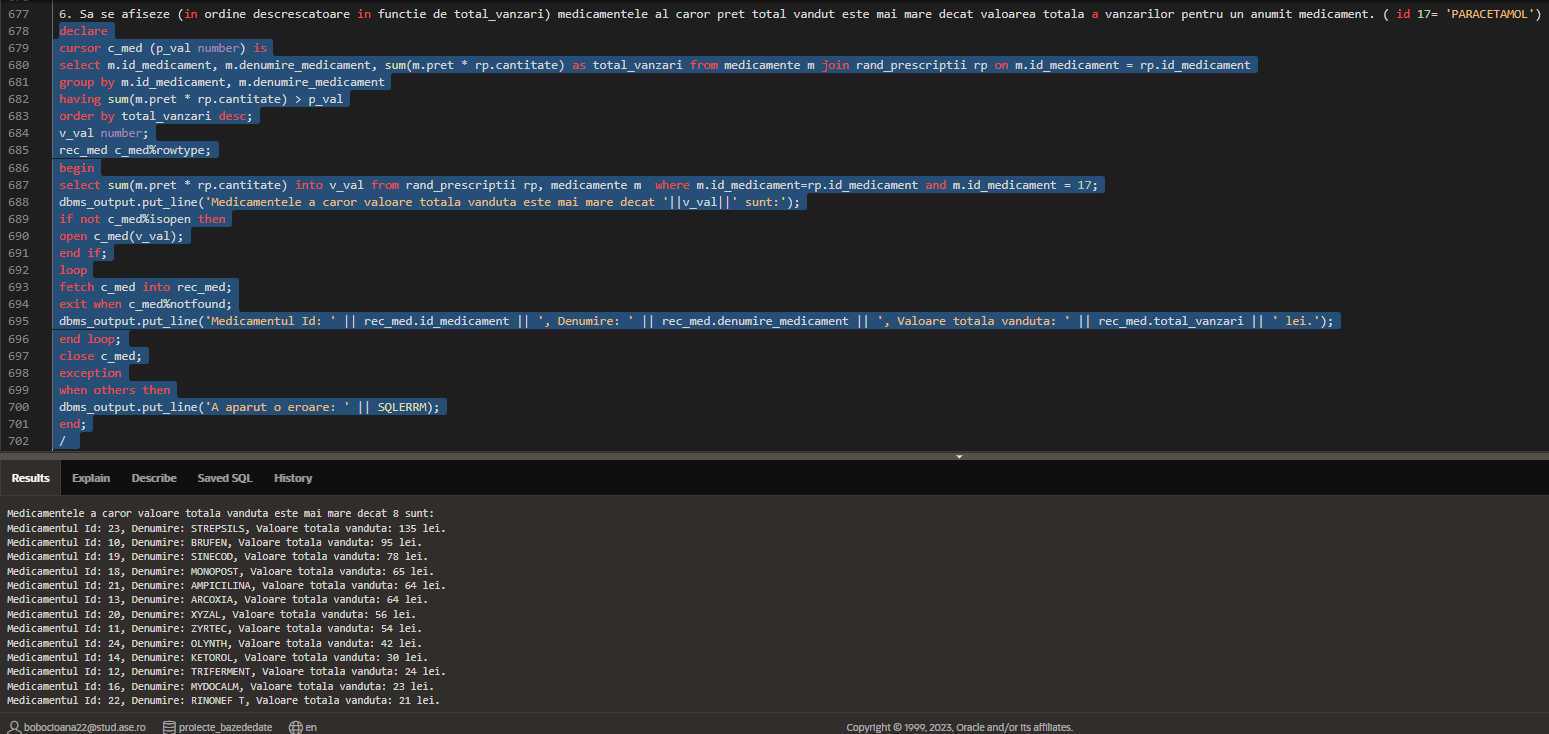
exception

when others then

dbms\_output.put\_line('A aparut o eroare: ' || SQLERRM);

end;

/



**G.**

Funcții, proceduri, includerea acestora în pachete. (minim 3 funcții, 3 proceduri și un pachet care să includă alte funcții și proceduri)

1. Sa se scrie o procedura care modifica salariul unui farmacist trimis prin id, cu un procent dat. (doar parametri IN)

create or replace procedure modifica\_salariul\_farmacist (

p\_id\_farmacist in farmacisti.id\_farmacist%type,

procent in number)

is v\_salariu farmacisti.salariu%type;

begin

select salariu into v\_salariu from farmacisti where id\_farmacist = p\_id\_farmacist;

dbms\_output.put\_line('Farmacistul are salariul de ' || v\_salariu);

update farmacisti set salariu = salariu + (salariu \* procent / 100) where id\_farmacist = p\_id\_farmacist;

select salariu into v\_salariu from farmacisti where id\_farmacist = p\_id\_farmacist;

dbms\_output.put\_line('Farmacistul are acum salariul de ' || v\_salariu);

end;

/

begin modifica\_salariul\_farmacist(2011, 15); end; /

2. Sa se afiseze toate datele despre un pacient ales prin id, folosind o procedura(parametri IN si OUT)

create or replace procedure cauta\_pacient (

p\_id\_pacient IN pacienti.id\_pacient%type,

p\_nume OUT pacienti.nume%type,

p\_prenume OUT pacienti.prenume%type,

p\_data\_nastere OUT pacienti.data\_nastere%type,

p\_sex out pacienti.sex%type,

p\_telefon out pacienti.telefon%type,

p\_email out pacienti.email%type)

is

begin

select nume, prenume, data\_nastere, sex, telefon, email into p\_nume, p\_prenume, p\_data\_nastere, p\_sex, p\_telefon, p\_email from pacienti where id\_pacient=p\_id\_pacient;

dbms\_output.put\_line(' Pacientul/Pacienta '||p\_nume||' '||p\_prenume||' este nascut(a) la data de '||p\_data\_nastere||', este de sex '||p\_sex||', are nr de telefon '||p\_telefon||' si adresa de email '||p\_email);

end;

/

declare

v\_nume pacienti.nume%type;

v\_prenume pacienti.prenume%type;

v\_data\_nastere pacienti.data\_nastere%type;

v\_sex pacienti.sex%type;

v\_telefon pacienti.telefon%type;

v\_email pacienti.email%type;

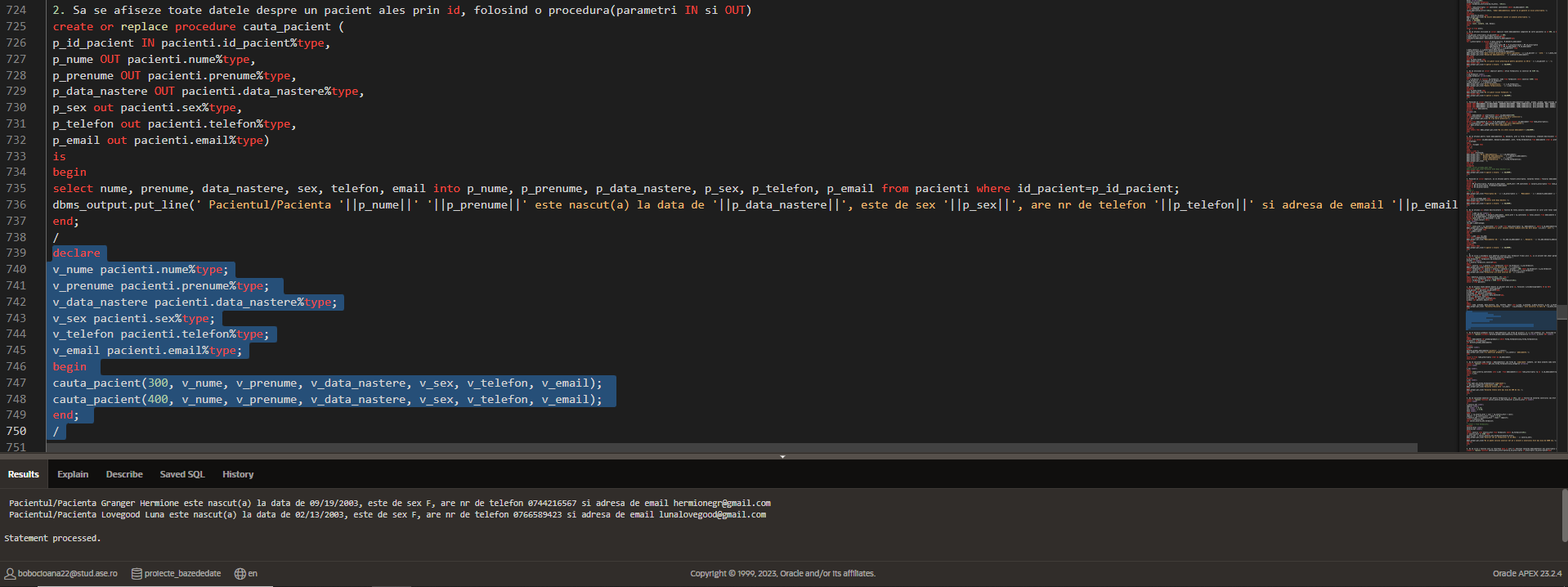
begin

cauta\_pacient(300, v\_nume, v\_prenume, v\_data\_nastere, v\_sex, v\_telefon, v\_email);

cauta\_pacient(400, v\_nume, v\_prenume, v\_data\_nastere, v\_sex, v\_telefon, v\_email);

end;

/



3. Sa se mareasca gramajul tuturor medicamentelor sub forma de picaturi cu 1,2 din gramajul lor, folosindu-se o procedura.

create or replace procedure mareste\_gramaj\_medicamente(p\_forma\_farmaceutica varchar2, p\_contor OUT number)

is

begin

update medicamente set gramaj=gramaj\*1.2 where forma\_farmaceutica=p\_forma\_farmaceutica;

p\_contor:= sql%rowcount;

end mareste\_gramaj\_medicamente;

/

declare

v\_contor number;

begin

mareste\_gramaj\_medicamente('picaturi', v\_contor);

dbms\_output.put\_line('S-a modificat gramajul a '||v\_contor||' medicamente.');

end;

/

4. Sa se calculeze suma totala a medicamentelor sub forma de 'comprimate' vandute, iar daca aceasta suma este sub 200 de lei, sa se afiseze un mesaj corespunzator.

create or replace function get\_val\_forma\_farmaceutica(p\_categorie varchar2)

return number

is

v\_val number;

begin

select sum(m.pret\*rp.cantitate) into v\_val from medicamente m join rand\_prescriptii rp on m.id\_medicament=rp.id\_medicament where m.forma\_farmaceutica='comprimate';

return v\_val;

end;

/

declare

v\_val number;

begin

v\_val:=get\_val\_forma\_farmaceutica('comprimate');

if get\_val\_categorie('comprimate')>200 then

dbms\_output.put\_line('Valoarea totala este '||v\_val);

else

dbms\_output.put\_line('Valoarea totala este mai mica de 200 de lei.');

end if;

end;

/

5. Sa se calculeze salariul net pentru farmacistul cu id 2011, iar in functie de valoarea salariului sau brut (<> 4500) sa se afiseze un mesaj corespunzator.

create or replace function calcul\_salariu\_net\_farmacist( p\_salariu\_brut in number)

return number

is

v\_salariu\_net number;

impozit number;

cas number := 0.15;

cass number := 0.08;

taxe number;

begin

taxe := (p\_salariu\_brut \* cas) + (p\_salariu\_brut \* cass);

impozit := (p\_salariu\_brut - taxe) \* 0.10;

v\_salariu\_net := p\_salariu\_brut - (taxe + impozit);

return v\_salariu\_net;

end calcul\_salariu\_net\_farmacist;

/

declare

salariu\_brut number;

salariu\_net number;

begin

select salariu into salariu\_brut from farmacisti where id\_farmacist=2011;

if salariu\_brut >= 4500 then

salariu\_net := calcul\_salariu\_net\_farmacist(salariu\_brut);

dbms\_output.put\_line('Salariul net pt farmacistul cu id 2011: ' || salariu\_net);

else

dbms\_output.put\_line('Nu se poate calcula salariul net pt o valoare a salariului brut mai mica de 4500 lei.');

end if;

end;

/

6. Sa se scrie o functie care sa returneze data in care s-a efectuat vanzarea medicamentelor din prescriptia cu id 5.

create or replace function obtine\_data\_prescriptiei(p\_id\_prescriptie in prescriptii.id\_prescriptie%type)

return date

is

v\_data\_vanzarii prescriptii.data\_vanzarii%type;

begin

select data\_vanzarii into v\_data\_vanzarii from prescriptii where id\_prescriptie = p\_id\_prescriptie;

return v\_data\_vanzarii;

exception

when no\_data\_found then

dbms\_output.put\_line('Nu exista datele cautate.');

end obtine\_data\_prescriptiei;

/

declare

v\_data\_v date;

begin

v\_data\_v:=obtine\_data\_prescriptiei(5);

dbms\_output.put\_line('Data in care au fost vandute medicamentele din prescriptia 5 este: '||TO\_DATE(v\_data\_v, 'DD.MM.YYYY'));

end;

/

7. Sa se creeze un pachet care sa actualizeze tabela de medicamente prin procedurile adauga\_medicament, modifica\_medicament, sterge\_medicament, utilizand functia exista\_id care verifica existenta unui medicament din tabela.

create or replace package actualizare\_medicamente is

procedure adauga\_medicament (p\_codm medicamente.id\_medicament%type, p\_denm medicamente.denumire\_medicament%type, p\_ff medicamente.forma\_farmaceutica%type, p\_pret medicamente.pret%type);

procedure modifica\_medicament (p\_codm medicamente.id\_medicament%type, p\_pret medicamente.pret%type);

procedure sterge\_medicament (p\_codm medicamente.id\_medicament%type);

function exista\_id (p\_codm medicamente.id\_medicament%type) return boolean;

exceptie exception;

end;

/

create or replace package body actualizare\_medicamente is

procedure adauga\_medicament (p\_codm medicamente.id\_medicament%type, p\_denm medicamente.denumire\_medicament%type, p\_ff medicamente.forma\_farmaceutica%type, p\_pret medicamente.pret%type)

is

begin

if exista\_id(p\_codm) then

raise exceptie;

else

insert into medicamente(id\_medicament,denumire\_medicament,forma\_farmaceutica,pret) values (p\_codm, p\_denm, p\_ff, p\_pret);

end if;

exception when exceptie then

dbms\_output.put\_line('Medicament existent!');

end;

procedure modifica\_medicament (p\_codm medicamente.id\_medicament%type, p\_pret medicamente.pret%type)

is

begin

if exista\_id(p\_codm) then

update medicamente set pret=p\_pret where id\_medicament=p\_codm;

else raise exceptie;

end if;

exception when exceptie then

dbms\_output.put\_line('Medicamentul cu aceast id nu exista!');

end;

procedure sterge\_medicament (p\_codm medicamente.id\_medicament%type)

is

begin

if exista\_id(p\_codm) then

delete from medicamente where id\_medicament=p\_codm;

dbms\_output.put\_line('Medicamentul cu idul '||p\_codm||' a fost sters!');

else

raise exceptie;

end if;

exception when exceptie then

dbms\_output.put\_line('Medicamentul cu aceast id nu exista!');

end;

function exista\_id (p\_codm medicamente.id\_medicament%type) return boolean

is

v\_unu number;

begin

select 1 into v\_unu from medicamente where id\_medicament=p\_codm;

return true;

exception

when no\_data\_found then

return false;

end;

end;

/

begin

actualizare\_medicamente.adauga\_medicament(1000,'proba', 'picaturi', 14);

end;

/

select \* from medicamente where id\_medicament=1000;

begin

actualizare\_medicamente.modifica\_medicament(1000,20);

end;

/

select \* from medicamente where id\_medicament=1000;

begin

actualizare\_medicamente.sterge\_medicament(1000);

end;

/

select \* from medicamente where id\_medicament=1000;

**H.**

Declanșatori la nivel de instrucțiune și de rând (minim 2 din fiecare).

1. Sa se creeze un trigger pentru comenzile insert, update si delete efectuate asupra tabelei medicamente.

create table temp\_log\_2

(tip varchar2(10), utilizator varchar2(50),

data date default sysdate);

create or replace trigger medicamente\_trigger

before insert or update or delete on medicamente

declare v\_tip temp\_log\_2.tip%type;

begin

case

when inserting then v\_tip:='inserting';

when updating then v\_tip:='updating';

when deleting then v\_tip:='deleting';

end case;

insert into temp\_log\_2(tip, utilizator, data) values (v\_tip, user, sysdate);

end;

/

select \* from temp\_log\_2;

2. Sa se construiasca un trigger care nu permite realizarea comenzii update intre orele 12:00 si 12:59, a comenzii delete intre orele 13:00 si 13:59 si a comenzii insert intre orele 14:00 si 14:59.

create or replace trigger verifica\_ora

after insert or update of denumire\_medicament, forma\_farmaceutica, gramaj or delete on medicamente

begin

if to\_char(sysdate, 'hh24:mi') between '12:00' and '12:59' and updating then

raise\_application\_error(-20001, 'Nu este permisa actualizarea la aceasta ora.');

elsif to\_char(sysdate, 'hh24:mi') between '13:00' and '13:59' and deleting then

raise\_application\_error(-20002, 'Nu este permisa stergerea la aceasta ora.');

elsif to\_char(sysdate, 'hh24:mi') between '14:00' and '14:59' and inserting then

raise\_application\_error(-20003, 'Nu este permisa adaugarea la aceasta ora.');

end if;

exception

--when others then

--dbms\_output.put\_line('Triggerul s-a declansat la '||sysdate);

end;

/

/\*update medicamente set forma\_farmaceutica='comprimate' where id\_medicament=12;

update medicamente set denumire\_medicament='proba3' where ID\_medicament=654;

delete from medicamente where denumire\_medicament='proba3';\*/

select \* from medicamente;

3. Sa se construiasca un trigger care nu permite scaderea salariului pentru farmacistii angajati la data de 01.01.1996.

create or replace trigger schimba\_salariu

before update of salariu on farmacisti

for each row

when (new.salariu < old.salariu and old.data\_angajare = to\_date('01.01.1996', 'DD.MM.YYYY'))

begin

raise\_application\_error(-20001, 'Nu este permisa scaderea salariului pentru farmacistii angajati la data de 01.01.1996');

end;

/

update farmacisti set salariu = 2500 where id\_farmacist = 444;

select \* from farmacisti;

4. Sa se construiasca un trigger care nu permite scaderea salariului pentru farmacistii care au vandut cel putin o prescriptie.

create or replace trigger schimba\_salariu\_2

before update of salariu on farmacisti

for each row

when (new.salariu<old.salariu)

declare

contor number;

begin

select count(id\_prescriptie) into contor from prescriptii p

where p.id\_farmacist=:new.id\_farmacist;

if contor>0 then

raise\_application\_error(-20001,'Nu este permisa scaderea salariului pentru farmacistii care au vandut prescriptii.');

end if;

end;

/

update farmacisti set salariu = 2500 where id\_farmacist = 111;

select \* from farmacisti;