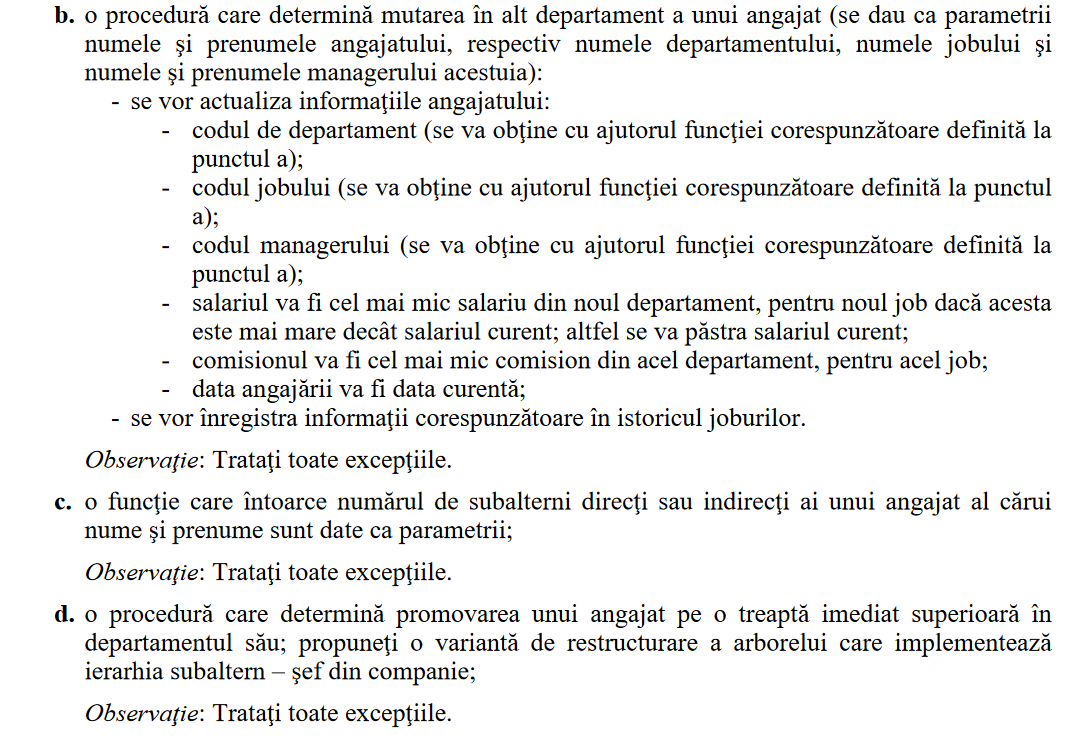
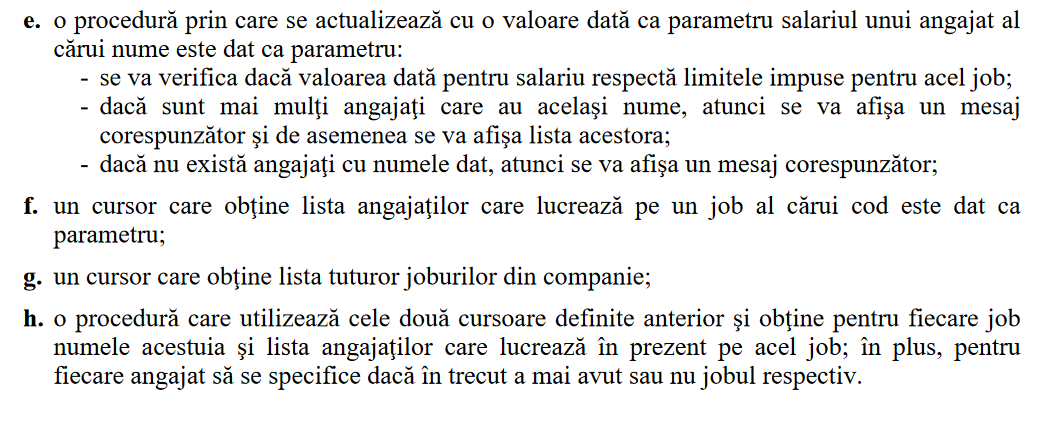
**Tema 6**

Nume: Pop Maria

Grupa: 232





create or replace package pachet1\_pm is

-- b)

procedure mutare\_angajat(nume employees.last\_name%type, prenume employees.last\_name%type,

nume\_manager employees.last\_name%type,

prenume\_manager employees.last\_name%type,

nume\_departament varchar2, nume\_job varchar2);

function get\_comision(cod\_dept employees.department\_id%type, cod\_job employees.job\_id%type)

return employees.commission\_pct%type;

-- c)

function numar\_subalterni(nume employees.last\_name%type, prenume employees.last\_name%type)

return number;

-- d)

procedure promovare\_angajat(cod\_ang employees.employee\_id%type);

-- e)

procedure actualizare\_salariu(nr number, nume varchar2);

-- f)

cursor lista\_angajati(cod\_job employees.job\_id%type)

return employees%rowtype;

-- g)

cursor lista\_joburi return jobs%rowtype is

select \*

from jobs;

-- h)

procedure informatii\_joburi;

end pachet1\_pm;

/

create or replace package body pachet1\_pm is

-- f)

cursor lista\_angajati(cod\_job employees.job\_id%type)

return employees%rowtype IS

select \*

from employees

where job\_id = cod\_job;

-- b)

function get\_comision(cod\_dept employees.department\_id%type, cod\_job employees.job\_id%type)

return employees.commission\_pct%type is

comision employees.commission\_pct%type;

begin

select min(commission\_pct)

into comision

from employees

where department\_id = cod\_dept and job\_id = cod\_job;

if comision is null then

comision := 0;

end if;

return comision;

end get\_comision;

-- b)

procedure mutare\_angajat(nume employees.last\_name%type, prenume employees.last\_name%type,

nume\_manager employees.last\_name%type,

prenume\_manager employees.last\_name%type,

nume\_departament varchar2, nume\_job varchar2)

is

data\_angajare employees.hire\_date%type;

comision employees.commission\_pct%type;

salariu\_nou employees.salary%type;

salariu\_curent employees.salary%type;

cod\_angajat employees.employee\_id%type;

cod\_departament employees.department\_id%type;

cod\_job employees.job\_id%type;

cod\_manager employees.manager\_id%type;

begin

cod\_departament := get\_cod\_dept(nume\_departament);

cod\_job := get\_cod\_job(nume\_job);

cod\_manager := get\_cod\_manager(nume\_manager, prenume\_manager);

salariu\_nou := get\_salariu(cod\_departament, cod\_job);

select salary, employee\_id, hire\_date

into salariu\_curent, cod\_angajat, data\_angajare

from emp\_pm

where upper(last\_name) = upper(nume) and

upper(first\_name) = upper(prenume);

if salariu\_nou < salariu\_curent then

salariu\_nou := salariu\_curent;

end if;

comision := get\_comision(cod\_departament, cod\_job);

insert into job\_history\_pm values(cod\_angajat, data\_angajare, sysdate, cod\_job, cod\_departament);

update emp\_pm

set hire\_date = sysdate, job\_id = cod\_job, salary = salariu\_nou,

commission\_pct = comision, manager\_id = cod\_manager, department\_id = cod\_departament

where employee\_id = cod\_angajat;

end mutare\_angajat;

-- c)

function numar\_subalterni(nume employees.last\_name%type, prenume employees.last\_name%type)

return number is

nr\_subalterni number := 0;

cod\_ang employees.employee\_id%type;

begin

select employee\_id

into cod\_ang

from employees

where upper(last\_name) = upper(nume) and upper(first\_name) = upper(prenume);

if cod\_ang is null then

raise\_application\_error(-20004, 'Nu exista angajatul');

end if;

select count(\*)

into nr\_subalterni

from emp\_pm

start with employee\_id = cod\_ang

connect by prior employee\_id = manager\_id;

return nr\_subalterni;

end numar\_subalterni;

-- d)

procedure promovare\_angajat(cod\_ang employees.employee\_id%type)

is

cod\_man1 employees.manager\_id%type;

cod\_man2 employees.manager\_id%type;

begin

select manager\_id

into cod\_man1

from employees

where employee\_id = cod\_ang;

if cod\_man1 is null then

dbms\_output.put\_line('Nu poate fi promovat');

else

select manager\_id

into cod\_man2

from employees

where employee\_id = cod\_man1;

if cod\_man2 is null then

update emp\_pm

set manager\_id = null

where employee\_id = cod\_ang;

else

update emp\_pm

set manager\_id = cod\_man2

where employee\_id = cod\_ang;

end if;

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

end if;

end promovare\_angajat;

-- e)

procedure actualizare\_salariu(nr number, nume varchar2)

is

prenume employees.last\_name%type;

cod\_job employees.job\_id%type;

cod\_ang employees.employee\_id%type;

nr\_ang number;

minim number;

maxim number;

begin

select count(\*)

into nr\_ang

from employees

where upper(last\_name) = upper(nume);

if nr\_ang = 1 then

select job\_id, employee\_id

into cod\_job, cod\_ang

from employees

where upper(last\_name) = upper(nume);

select min\_salary, max\_salary

into minim, maxim

from jobs

where job\_id = cod\_job;

if nr >= minim and nr <= maxim then

update emp\_pm

set salary = nr

where employee\_id = cod\_ang;

else

dbms\_output.put\_line('Salariul nou nu corespunde');

end if;

elsif nr\_ang = 0 then

dbms\_output.put\_line('Nu exista angajati cu numele dat');

else

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

for i in (select \* from employees

where upper(nume) = upper(last\_name))

loop

dbms\_output.put\_line(i.last\_name || ' ' || i.first\_name);

end loop;

end if;

end actualizare\_salariu;

-- h)

procedure informatii\_joburi

is

nr\_joburi number;

BEGIN

for i in pachet1\_pm.lista\_joburi loop

dbms\_output.put\_line(i.job\_title || ': ');

for j in pachet1\_pm.lista\_angajati(i.job\_id) loop

select count(\*)

into nr\_joburi

from job\_history\_pm

where employee\_id = j.employee\_id and job\_id = j.job\_id;

dbms\_output.put\_line(j.last\_name || j.first\_name || ': ' || nr\_joburi);

end loop;

dbms\_output.new\_line;

end loop;

end informatii\_joburi;

end pachet1\_pm;

/