**Tema 3**

Exercitiul 1

DECLARE

numar number(3):=100;

mesaj1 varchar2(255):='text 1';

mesaj2 varchar2(255):='text 2';

BEGIN

DECLARE

numar number(3):=1;

mesaj1 varchar2(255):='text 2';

mesaj2 varchar2(255):='text 3';

BEGIN

numar:=numar+1;

mesaj2:=mesaj2||' adaugat in sub-bloc';

dbms\_output.put\_line(numar);

dbms\_output.put\_line(mesaj1);

dbms\_output.put\_line(mesaj2);

END;

numar:=numar+1;

mesaj1:=mesaj1||' adaugat un blocul principal';

mesaj2:=mesaj2||' adaugat in blocul principal';

dbms\_output.put\_line(numar);

dbms\_output.put\_line(mesaj1);

dbms\_output.put\_line(mesaj2);

END;

/

a) Valoarea variabilei numar in subbloc este 2.

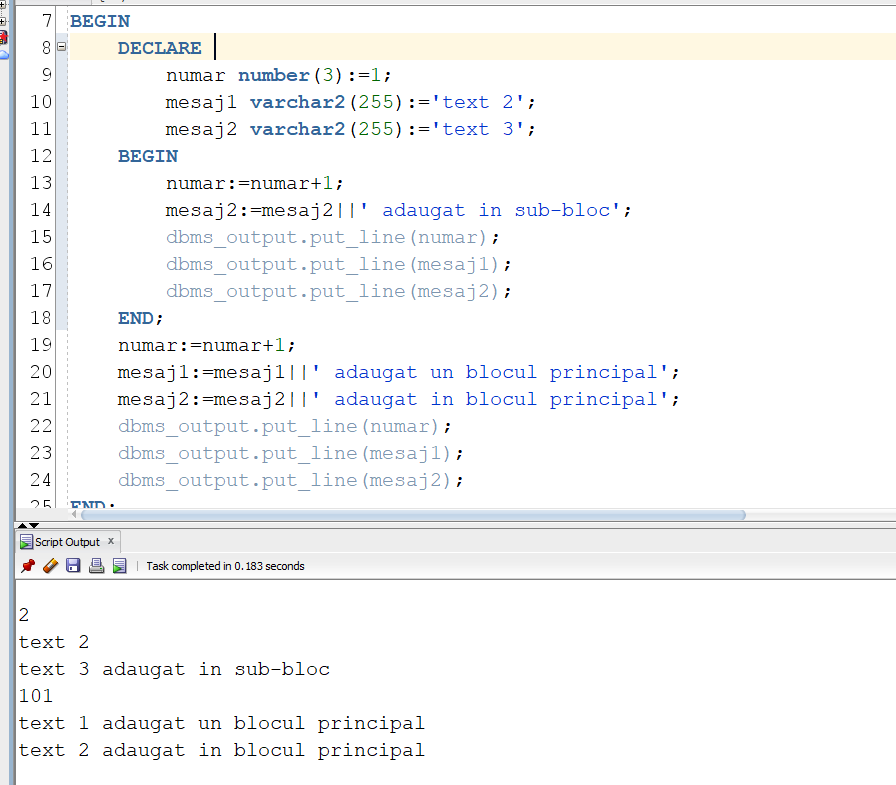
b) Valoarea variabilei mesaj1 in subbloc este "text 2".

c) Valoarea variabilei mesaj2 in subbloc este "text 3 adaugat in sub-bloc".

d) Valoarea variabilei numar in bloc este 101.

e) Valoarea variabilei mesaj1 in bloc este "text 1 adaugat un blocul principal".

f) Valoarea variabilei mesaj2 in bloc este "text 2 adaugat in blocul principal".



Exercitiul 2

a)

select dt, (

select count(\*)

from rental

where to\_char(book\_date, 'dd') = to\_char(dt, 'dd') and to\_char(book\_date, 'Mon') = 'Oct'

) as "Imprumuturi"

from (

select (last\_day('1-Oct-22') - rownum + 1) dt

from dual

connect by rownum <= extract(day from last\_day('1-Oct-22'))

)

order by dt;

b)

create table octombrie\_pam(

id number,

data date

);

DECLARE

imprumuturi NUMBER := 0;

ultima\_zi NUMBER := to\_char(last\_day('1-Oct-22'), 'dd');

BEGIN

FOR zi IN 1..ultima\_zi LOOP

select count(\*)

into imprumuturi

from rental

where to\_char(book\_date, 'dd') = zi and to\_char(book\_date, 'Mon') = 'Oct';

INSERT INTO octombrie\_pam VALUES (imprumuturi, TO\_DATE(zi ||'.10.2022', 'DD.MM.YYYY'));

END LOOP;

END;

/

Exercitiul 3

DECLARE

v\_imprumuturi NUMBER;

v\_nume VARCHAR2(25) := '&nume';

BEGIN

select count(distinct(r.title\_id))

into v\_imprumuturi

from rental r join member m on (r.member\_id(+) = m.member\_id)

where lower(last\_name) = lower(v\_nume) or lower(first\_name) = lower(v\_nume)

group by m.member\_id;

dbms\_output.put\_line('Numarul de filme imprumutate de ' || v\_nume || ' este '|| v\_imprumuturi);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

dbms\_output.put\_line('Nu exista niciun membru cu numele dat!');

WHEN TOO\_MANY\_ROWS THEN

dbms\_output.put\_line('Exista mai multi membrii cu acelasi nume!');

END;

/

Exercitiul 4

select \* from rental;

select \* from member;

DECLARE

v\_imprumuturi NUMBER;

v\_titluri NUMBER;

v\_proc NUMBER;

v\_nume VARCHAR(101) := '&nume';

BEGIN

select count(\*) into v\_titluri

from title;

select count(distinct(r.title\_id))

into v\_imprumuturi

from rental r join member m on (r.member\_id(+) = m.member\_id)

where lower(last\_name) = lower(v\_nume) or lower(first\_name) = lower(v\_nume)

group by m.member\_id;

v\_proc := (v\_imprumuturi/v\_titluri)\*100;

if v\_proc > 75 then

dbms\_output.put\_line('Categoria 1');

elsif v\_proc > 50 then

dbms\_output.put\_line('Categoria 2');

elsif v\_proc > 25 then

dbms\_output.put\_line('Categoria 3');

else

dbms\_output.put\_line('Categoria 4');

end if;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

dbms\_output.put\_line('Nu exista niciun membru cu numele dat!');

WHEN TOO\_MANY\_ROWS THEN

dbms\_output.put\_line('Exista mai multi membrii cu acelasi nume!');

END;

/

EXERCITIUL 5

create table member\_pam as select \* from member;

alter table member\_pam

add discount number;

set serveroutput on;

DECLARE

v\_cod\_membru member\_pam.member\_id%TYPE := &cod;

v\_imprumuturi NUMBER;

v\_titluri NUMBER;

v\_proc NUMBER;

BEGIN

select count(\*) into v\_titluri

from title;

select count(distinct(r.title\_id))

into v\_imprumuturi

from rental r join member m on (r.member\_id(+) = m.member\_id)

where m.member\_id = v\_cod\_membru

group by m.member\_id

having m.member\_id = v\_cod\_membru;

v\_proc := (v\_imprumuturi/v\_titluri)\*100;

if v\_proc > 75 then

update member\_pam

set discount = 10

where member\_id = v\_cod\_membru;

commit;

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

elsif v\_proc > 50 then

update member\_pam

set discount = 5

where member\_id = v\_cod\_membru;

commit;

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

elsif v\_proc > 25 then

update member\_pam

set discount = 3

where member\_id = v\_cod\_membru;

commit;

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

else

update member\_pam

set discount = 0

where member\_id = v\_cod\_membru;

commit;

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

end if;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Actualizare esuata, nu exista niciun angajat cu acest cod');

END;

/