E3. Definiți un bloc anonim în care să se determine numărul de filme (titluri) împrumutate de un

membru al cărui nume este introdus de la tastatură. Tratați următoarele două situații: nu există nici

un membru cu nume dat; există mai mulți membrii cu același nume.

E4. Modificați problema anterioară astfel încât să afișați și următorul text:

- Categoria 1 (a împrumutat mai mult de 75% din titlurile existente)

- Categoria 2 (a împrumutat mai mult de 50% din titlurile existente)

- Categoria 3 (a împrumutat mai mult de 25% din titlurile existente)

- Categoria 4 (altfel)

**Cod sql:**

declare

nume\_membru member.last\_name%type;

nume member.last\_name%type;

numar\_aparitii\_nume number(5):=0;

numar\_titluri\_distincte number(5) := 0;

numar\_titluri number(5):=0;

numar\_total\_titluri number(5):=0;

begin

nume := '&nume';

nume\_membru := nume;

--vedem daca aruncam sau nu exceptii

--nr aparitii nune

select count(\*)

into numar\_aparitii\_nume

from member

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

--aruncare exceptii

if numar\_aparitii\_nume = 0 then raise no\_data\_found;

elsif numar\_aparitii\_nume > 1 then raise too\_many\_rows;

end if;

--daca nu aruncam exceptii continuam logica programului

--nr total titluri

select count(distinct(title\_id))

into numar\_total\_titluri

from title;

-- doar de verificare:dbms\_output.put\_line('NUMAR TOTAL TITLURI ' || numar\_total\_titluri);

--calculare titluri(nu distincte neaparat) ptc nu mentioneaza cerinta

select count(r.title\_id)

into numar\_titluri

from rental r

join member m on m.member\_id = r.member\_id

where upper(m.last\_name)=upper(nume\_membru);

--afisare nr titluri

dbms\_output.put\_line('Membrul cu numele ' || nume\_membru || ' a inchiriat '|| numar\_titluri || ' titluri');

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

into numar\_titluri\_distincte

from rental r

join member m on m.member\_id = r.member\_id

where upper(m.last\_name)=upper(nume\_membru);

--afisare nr titluri distincte

dbms\_output.put\_line('dintre care '|| numar\_titluri\_distincte || ' titluri distincte');

--verificare din ce categorie face parte

if numar\_titluri\_distincte > 0.75\*(numar\_total\_titluri) then --categoria 1 75%

dbms\_output.put\_line('Membrul face parte din Categoria 1');

elsif numar\_titluri\_distincte > 0.5\*(numar\_total\_titluri) then --categoria 2 50%

dbms\_output.put\_line('Membrul face parte din Categoria 2');

elsif numar\_titluri\_distincte > 0.25\*(numar\_total\_titluri) then --categoria 3 25%

dbms\_output.put\_line('Membrul face parte din Categoria 3');

else dbms\_output.put\_line('Membrul face parte din Categoria 4'); --categoria 4 altfel

end if;

--tratare exceptii

exception

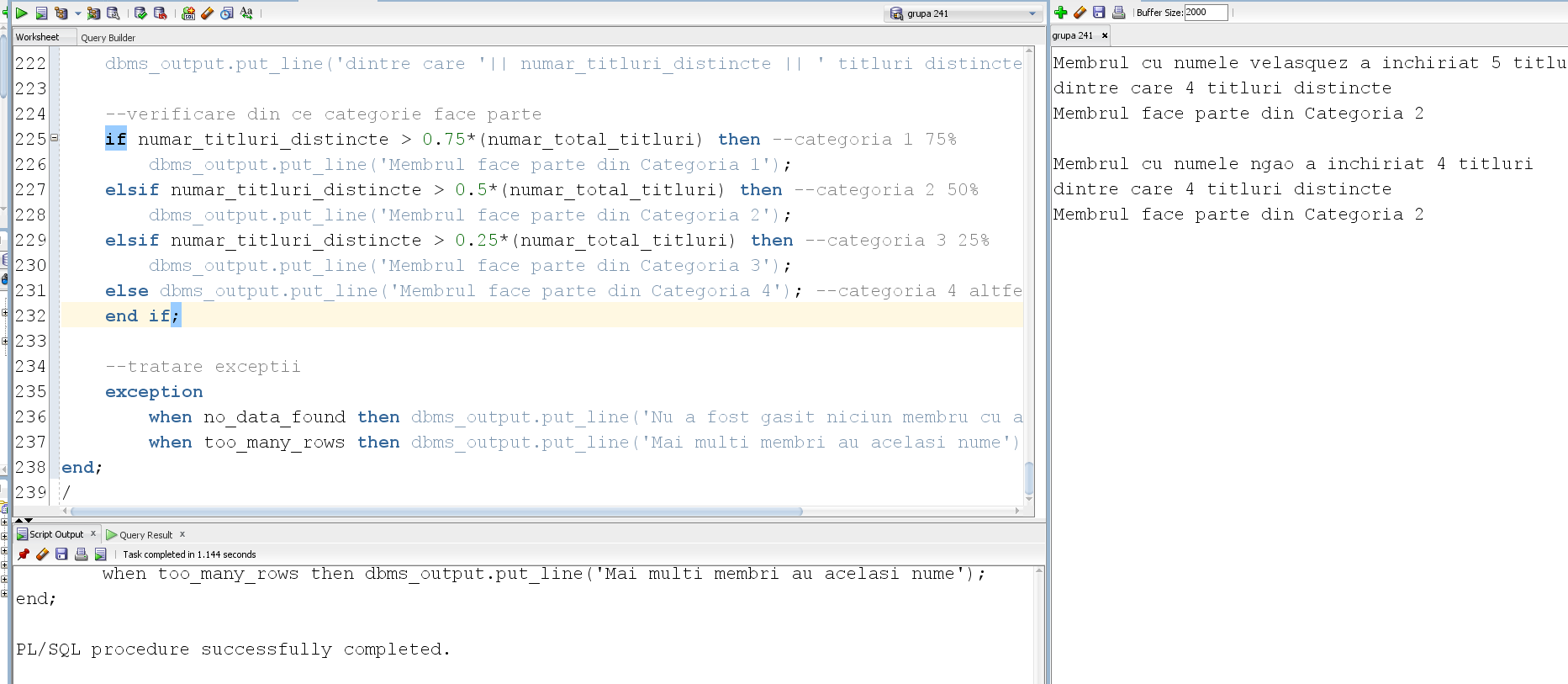
when no\_data\_found then dbms\_output.put\_line('Nu a fost gasit niciun membru cu acest nume');

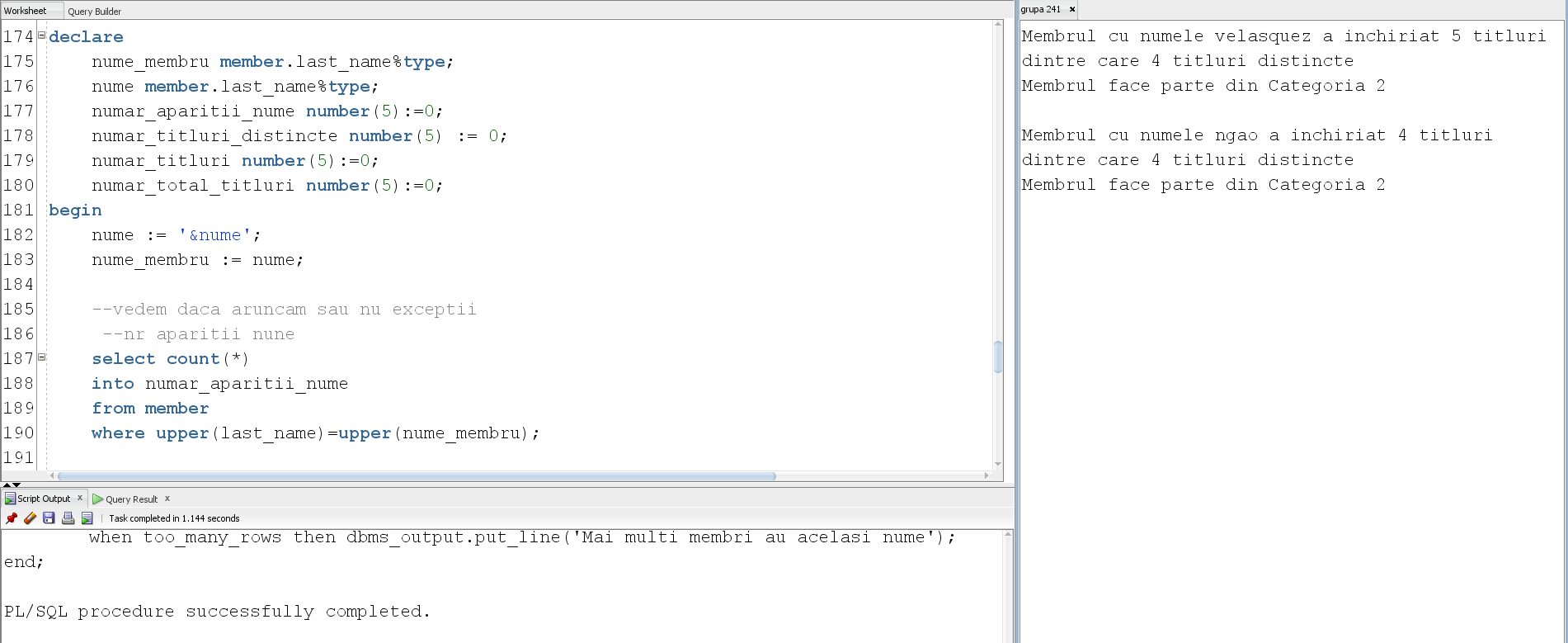
when too\_many\_rows then dbms\_output.put\_line('Mai multi membri au acelasi nume');

end**;**

**/**

**Print screen-uri:**

****

****

**E5. Creați tabelul member\_\*\*\* (o copie a tabelului member). Adăugați în acest tabel coloana**

**discount, care va reprezenta procentul de reducere aplicat pentru membrii, în funcție de categoria**

**din care fac parte aceștia:**

**- 10% pentru membrii din Categoria 1**

**- 5% pentru membrii din Categoria 2**

**- 3% pentru membrii din Categoria 3**

**- nimic**

**Actualizați coloana discount pentru un membru al cărui cod este dat de la tastatură. Afișați un**

**mesaj din care să reiasă dacă actualizarea s-a produs sau nu.**

**Cod sql:**

declare

categorie number(2):=0;

numar\_titluri\_distincte number(4) :=0;

numar\_total\_titluri number (4) :=0;

v\_id member\_sma.member\_id%type;

begin

v\_id := '&v\_id';

--nr total titluri ca sa stabilim categoria

select count(distinct(title\_id))

into numar\_total\_titluri

from title;

--vedem care e categoria folosind codul de la ex 3

--vedem cate titluri distincte a inchiriat

select count(distinct r.title\_id)

into numar\_titluri\_distincte

from rental r

join member m on m.member\_id = r.member\_id

where m.member\_id = v\_id;

--verificare din ce categorie face parte

if numar\_titluri\_distincte > 0.75\*(numar\_total\_titluri) then --categoria 1 75%

categorie := 1;

elsif numar\_titluri\_distincte > 0.5\*(numar\_total\_titluri) then --categoria 2 50%

categorie :=2;

elsif numar\_titluri\_distincte > 0.25\*(numar\_total\_titluri) then --categoria 3 25%

categorie := 3;

else categorie := 4; --categoria 4 altfel

end if;

--actualizare tabel cu discountul aferent

update member\_sma

set discount =

case

when categorie = 1 then 0.10

when categorie = 2 then 0.05

when categorie = 3 then 0.03

else 0

end

where member\_sma.member\_id = v\_id;

--verificare update

if sql%rowcount = 0 then

dbms\_output.put\_line('Atentie! Update-ul nu s-a realizat.');

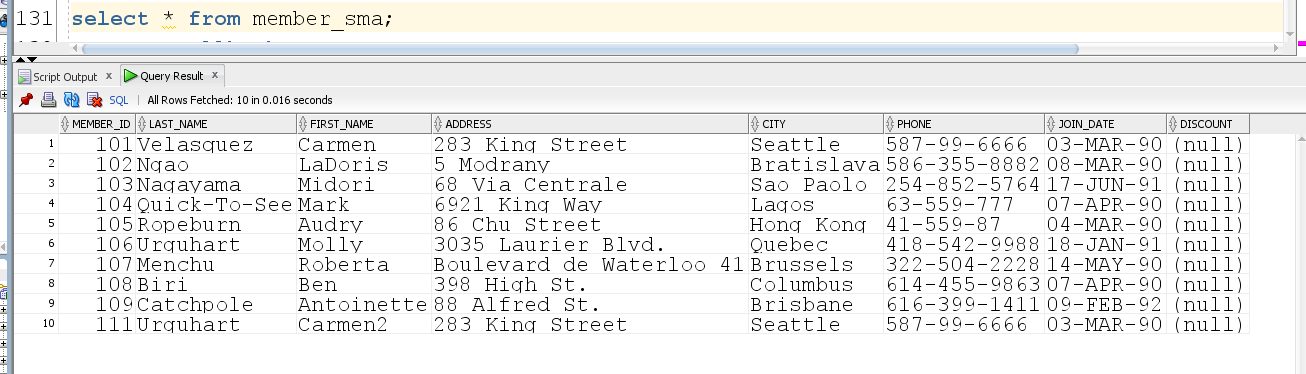
else dbms\_output.put\_line('Update realizat cu succes pentru membrul cu id-ul ' || v\_id);

end if;

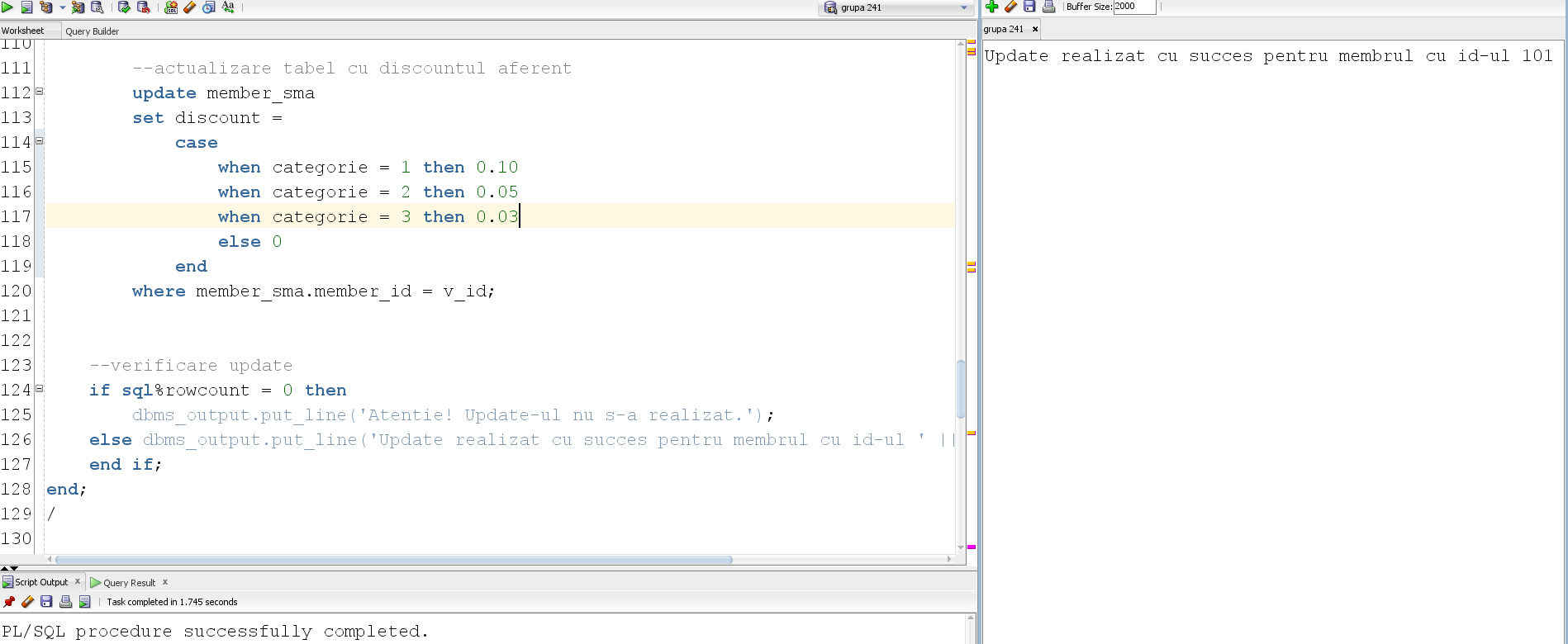
end;

/

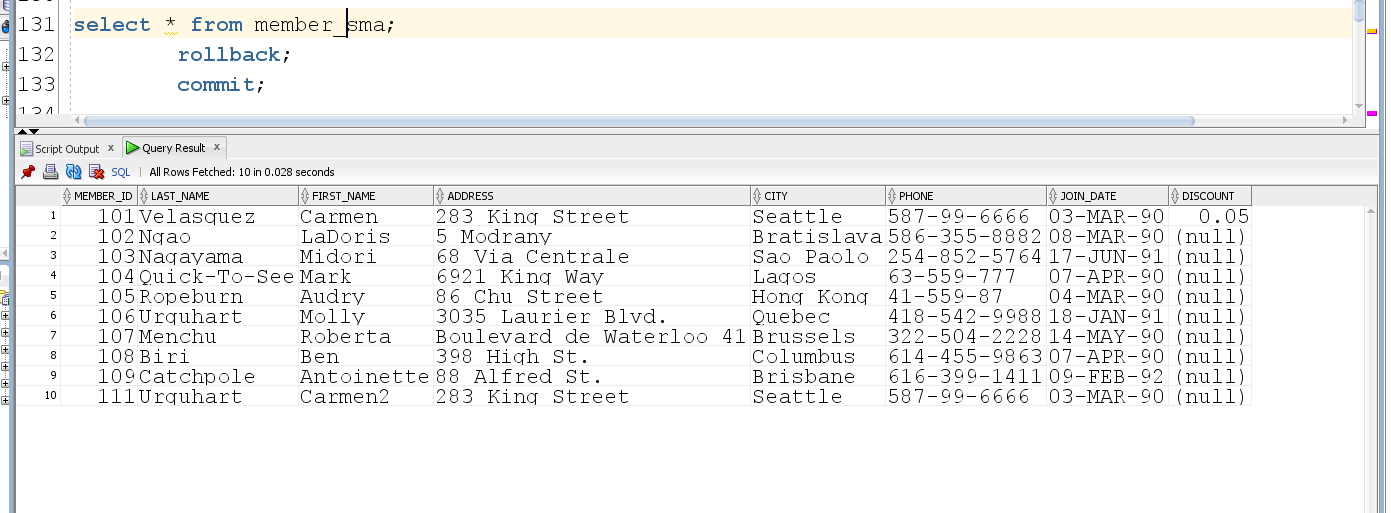
**Print screen-uri:**

Tabelul member\_sma + coloana discount****

Actualizare pentru id=101 (Velasquez) (are imprumutate 4 titluri distincte din cele 7 deci se incadreaza la categoria 2)

****

Tabelul member\_sma dupa actualizare:



Incercare actualizare pentru member\_id = 5 (nu exista)

