Tema Lab 1 PL/SQL – Ungureanu Matei

E1. Se dă următorul bloc:

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';

END;

numar:=numar+1;

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

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

END;

a) Valoarea variabilei numar în subbloc este: 2

b) Valoarea variabilei mesaj1 în subbloc este: text 2

c) Valoarea variabilei mesaj2 în subbloc este: text 3 adaugat in sub-bloc

d) Valoarea variabilei numar în bloc este: 101

e) Valoarea variabilei mesaj1 în bloc este: text 1 adaugat in blocul principal

f) Valoarea variabilei mesaj2 în bloc este: text 2 adaugat in blocul principal

Verificați răspunsul.

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;

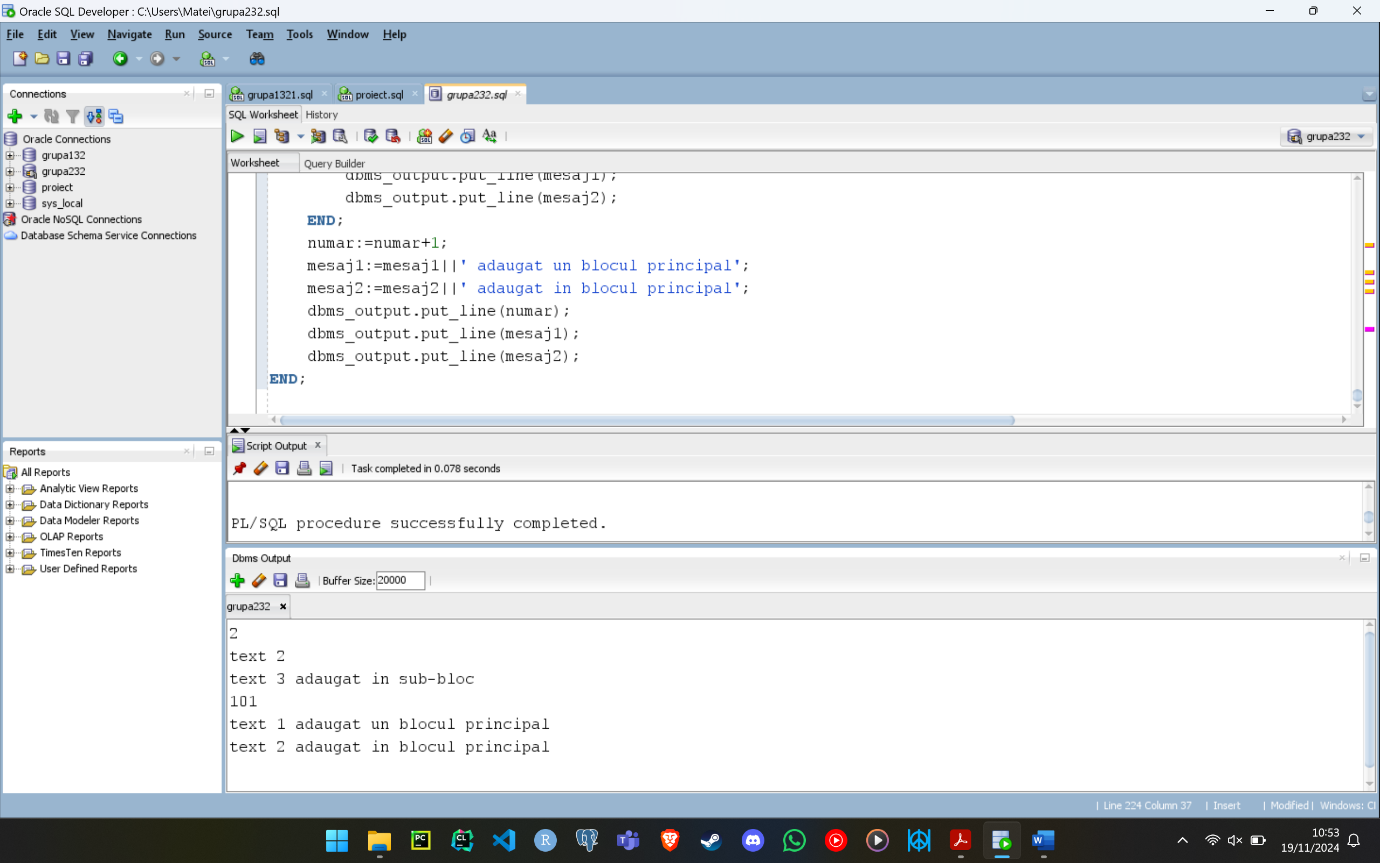
mesaj1:=mesaj1||' adaugat un blocul principal';

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

dbms\_output.put\_line(numar);

dbms\_output.put\_line(mesaj1);

dbms\_output.put\_line(mesaj2);

END;

E2. Se dă următorul enunț: Pentru fiecare zi a lunii octombrie (se vor lua în considerare și zilele din lună în care nu au fost realizate împrumuturi) obțineți numărul de împrumuturi efectuate.

a. Încercați să rezolvați problema în SQL fără a folosi structuri ajutătoare.

b. Definiți tabelul octombrie\_\*\*\* (id, data). Folosind PL/SQL populați cu date acest tabel. Rezolvați în SQL problema dată.

a.

select to\_date(rownum || '-NOV-2024', 'DD-MON-YY') as ziua, (

select count(\*)

from rental

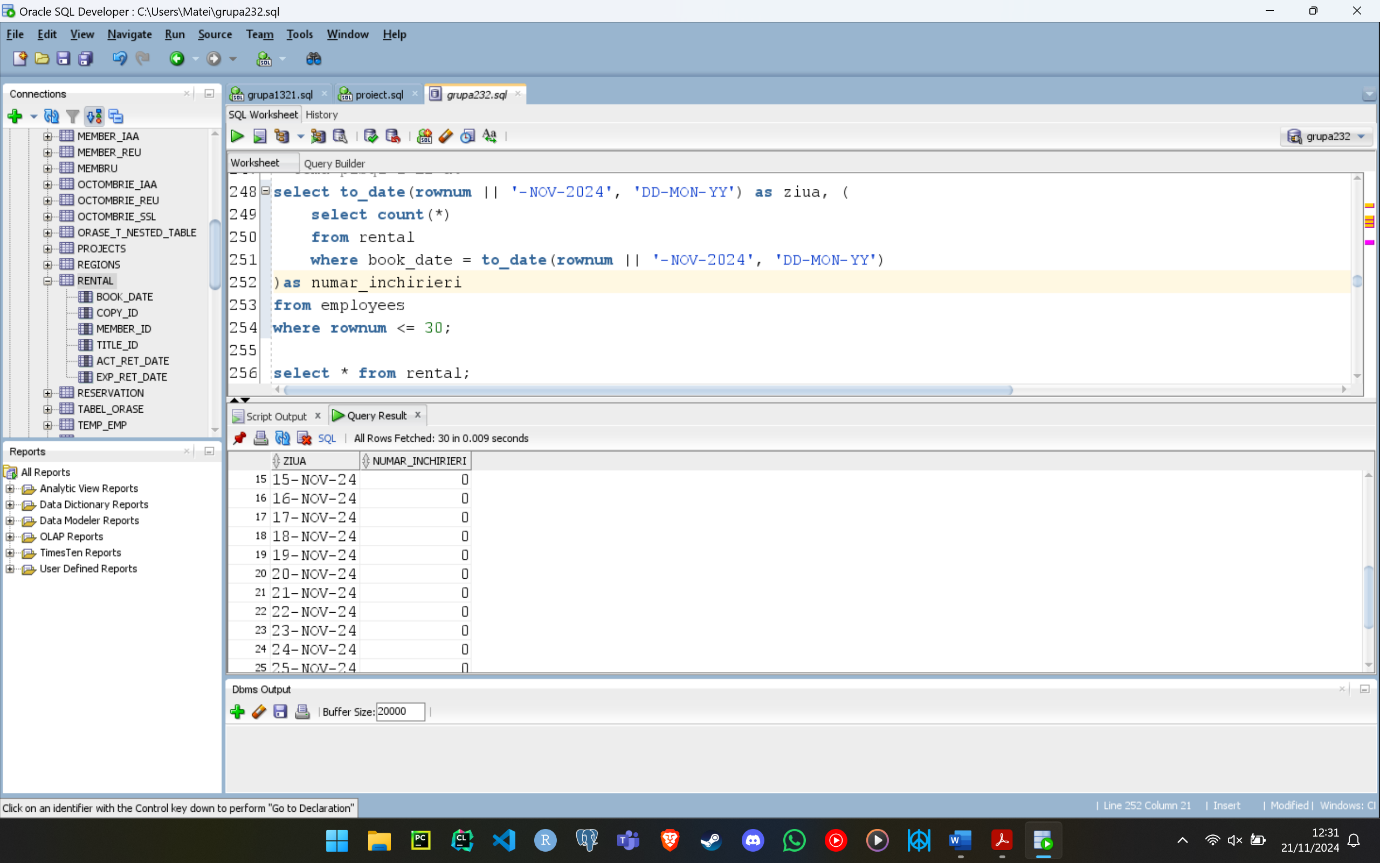
where book\_date = to\_date(rownum || '-NOV-2024', 'DD-MON-YY')

)as numar\_inchirieri

from employees

where rownum <= 30;

Am rulat un select \* from rental si nu apare nimic din octombrie, toate datele sunt din noiembrie, asa ca am rulat pentru noiembrie. Totusi, apare 0 in toate zilele, pentru ca rownum din query e inaccesibil din subquery (cred).



b.

create table octombrie\_ums (

zi number

);

begin

for i in 1..31 loop

insert into octombrie\_ums (zi) values (i);

end loop;

commit;

end;

/

select o.zi || '.10.2024' as data, count(r.book\_date) as inchirieri

from octombrie\_ums o left join rental r

on extract(day from r.book\_date) = o.zi

and extract(month from r.book\_date) = 11

group by o.zi

order by o.zi;

La fel ca la punctul a, am rulat pentru noiembrie.

Rezultat:

1.10.2024 0

2.10.2024 0

3.10.2024 0

4.10.2024 0

5.10.2024 0

6.10.2024 0

7.10.2024 0

8.10.2024 0

9.10.2024 0

10.10.2024 0

11.10.2024 0

12.10.2024 1

13.10.2024 2

14.10.2024 3

15.10.2024 4

16.10.2024 3

17.10.2024 1

18.10.2024 0

19.10.2024 0

20.10.2024 0

21.10.2024 0

22.10.2024 0

23.10.2024 0

24.10.2024 0

25.10.2024 0

26.10.2024 0

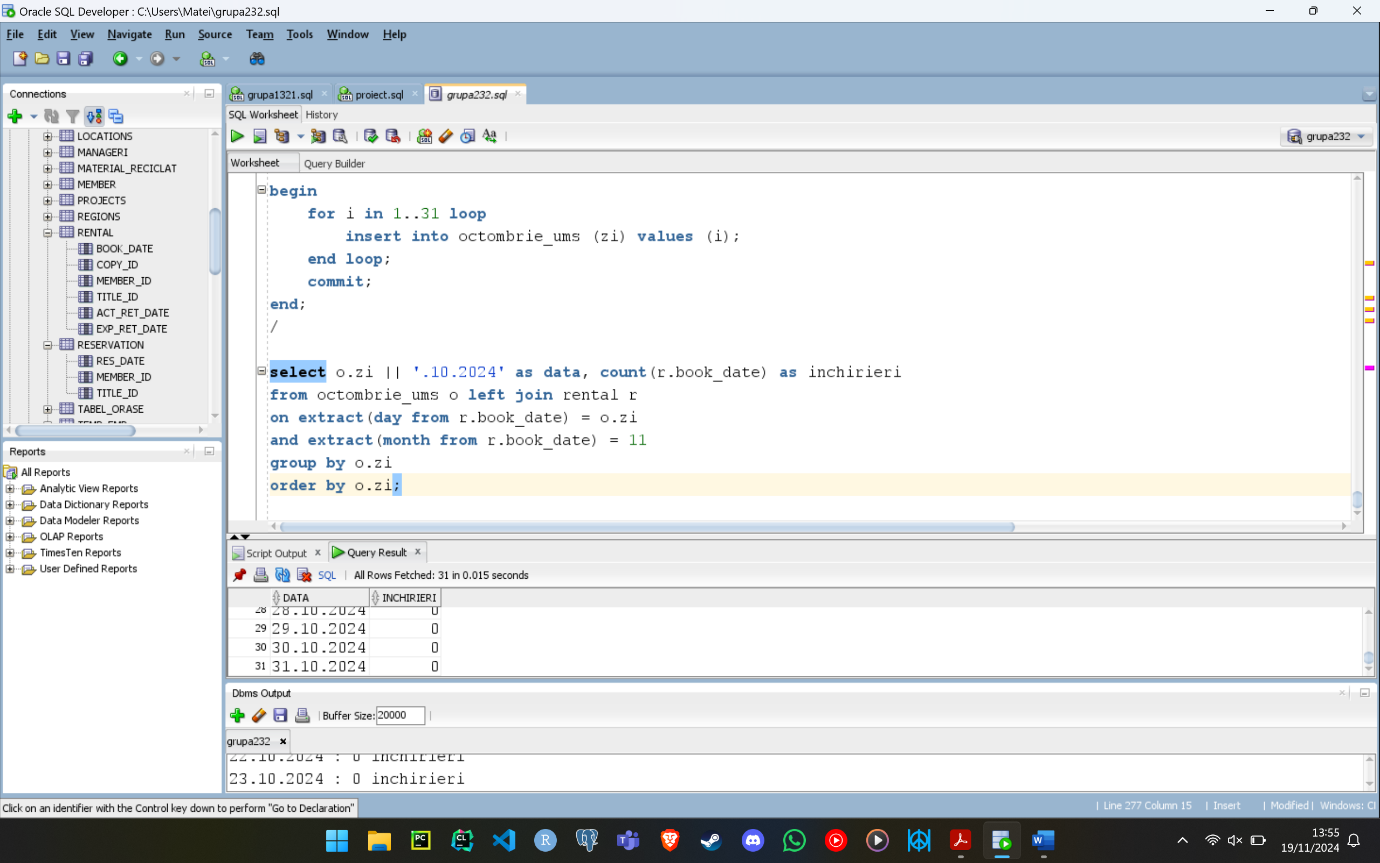
27.10.2024 0

28.10.2024 0

29.10.2024 0

30.10.2024 0

31.10.2024 0



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.

declare

nume\_membru member.last\_name%type := &p\_nume;

numar number;

numar\_filme number;

begin

select count(\*)

into numar

from member

where last\_name = nume\_membru;

if numar = 0 then

dbms\_output.put\_line('Nu exista niciun membru cu numele ' || nume\_membru);

elsif numar > 1 then

dbms\_output.put\_line('Exista cel putin 2 membri cu numele ' || nume\_membru

|| ', introduceti si prenumele: ');

declare

prenume\_membru member.first\_name%type := &p\_prenume;

numar number;

begin

select count(\*)

into numar

from member

where last\_name = nume\_membru

and first\_name = prenume\_membru;

if numar = 0 then

dbms\_output.put\_line('Nu exista membrul cautat');

elsif numar > 0 then

dbms\_output.put\_line('Exista duplicate in baza de date');

else

select count(\*)

into numar\_filme

from rental join member using(member\_id)

where last\_name = nume\_membru

and first\_name = prenume\_membru;

dbms\_output.put\_line('Membrul ' || nume\_membru || ' ' || prenume\_membru

|| ' a inchiriat ' || numar\_filme || ' filme');

end if;

end;

else

select count(\*)

into numar\_filme

from rental join member using(member\_id)

where last\_name = nume\_membru;

dbms\_output.put\_line('Membrul ' || nume\_membru || ' a inchiriat ' || numar\_filme || ' filme');

end if;

end;

/

Aici nu ruleaza bine. In primul rand imi cere si numele si prenumele imediat ce rulez, iar eu voiam sa imi ceara prenumele doar daca sunt cel putin 2 cu acelasi nume. M-am uitat in tabela si am bagat Ngao LaDoris. In al doilea rand, da eroare.

Error report -

ORA-06550: line 2, column 42:

PLS-00201: identifier 'NGAO' must be declared

ORA-06550: line 2, column 17:

PL/SQL: Item ignored

ORA-06550: line 9, column 23:

PLS-00320: the declaration of the type of this expression is incomplete or malformed

ORA-06550: line 9, column 23:

PL/SQL: ORA-00904: "NUME\_MEMBRU": invalid identifier

ORA-06550: line 6, column 5:

PL/SQL: SQL Statement ignored

ORA-06550: line 12, column 70:

PLS-00320: the declaration of the type of this expression is incomplete or malformed

ORA-06550: line 12, column 9:

PL/SQL: Statement ignored

ORA-06550: line 14, column 72:

PLS-00320: the declaration of the type of this expression is incomplete or malformed

ORA-06550: line 14, column 9:

PL/SQL: Statement ignored

ORA-06550: line 17, column 54:

PLS-00201: identifier 'LADORIS' must be declared

ORA-06550: line 17, column 28:

PL/SQL: Item ignored

ORA-06550: line 24, column 30:

PLS-00320: the declaration of the type of this expression is incomplete or malformed

ORA-06550: line 24, column 30:

PL/SQL: ORA-00904: "PRENUME\_MEMBRU": invalid identifier

ORA-06550: line 20, column 13:

PL/SQL: SQL Statement ignored

ORA-06550: line 35, column 34:

PLS-00320: the declaration of the type of this expression is incomplete or malformed

ORA-06550: line 35, column 34:

PL/SQL: ORA-00904: "PRENUME\_MEMBRU": invalid identifier

ORA-06550: line 31, column 17:

PL/SQL: SQL Statement ignored

ORA-06550: line 36, column 52:

PLS-00320: the declaration of the type of this expression is incomplete or malformed

ORA-06550: line 36, column 17:

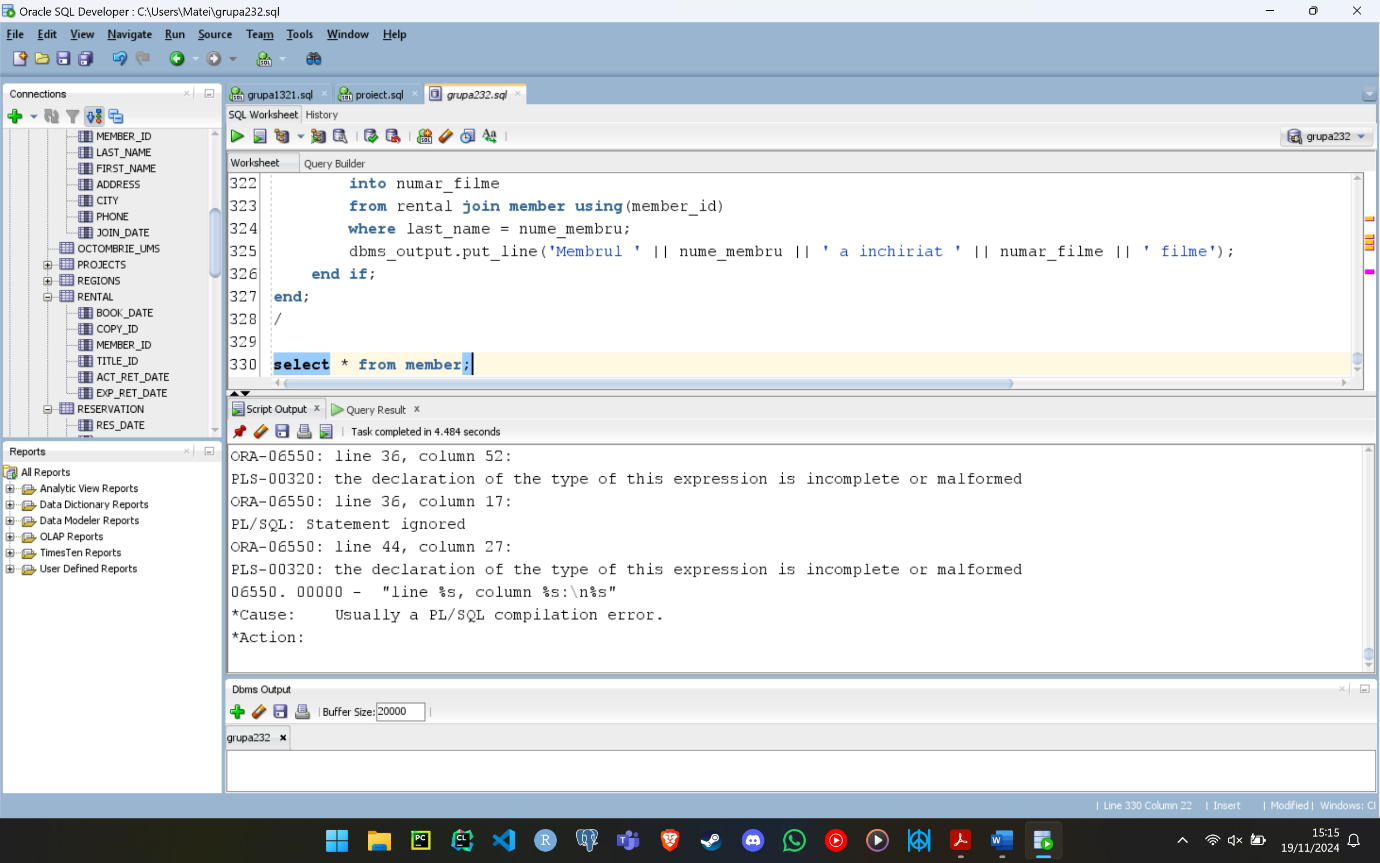
PL/SQL: Statement ignored

ORA-06550: line 44, column 27:

PLS-00320: the declaration of the type of this expression is incomplete or malformed

06550. 00000 - "line %s, column %s:\n%s"

\*Cause: Usually a PL/SQL compilation error.

\*Action:

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)

declare

nume\_membru member.last\_name%type := &p\_nume;

numar number;

numar\_filme number;

total\_filme number;

begin

select count(\*)

into total\_filme

from title;

select count(\*)

into numar

from member

where last\_name = nume\_membru;

if numar = 0 then

dbms\_output.put\_line('Nu exista niciun membru cu numele ' || nume\_membru);

elsif numar > 1 then

dbms\_output.put\_line('Exista cel putin 2 membri cu numele ' || nume\_membru

|| ', introduceti si prenumele: ');

declare

prenume\_membru member.first\_name%type := &p\_prenume;

numar number;

begin

select count(\*)

into numar

from member

where last\_name = nume\_membru

and first\_name = prenume\_membru;

if numar = 0 then

dbms\_output.put\_line('Nu exista membrul cautat');

elsif numar > 0 then

dbms\_output.put\_line('Exista duplicate in baza de date');

else

select count(\*)

into numar\_filme

from rental join member using(member\_id)

where last\_name = nume\_membru

and first\_name = prenume\_membru;

case

when numar\_filme > total\_filme \*3/4

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

when numar\_filme > total\_filme /2 and numar\_filme < total\_filme \*3/4

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

when numar\_filme > total\_filme /4 and numar\_filme < total\_filme /2

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

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

dbms\_output.put\_line('Membrul ' || nume\_membru || ' ' || prenume\_membru

|| ' a inchiriat ' || numar\_filme || ' filme');

end if;

end;

else

select count(\*)

into numar\_filme

from rental join member using(member\_id)

where last\_name = nume\_membru;

case

when numar\_filme > total\_filme \*3/4

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

when numar\_filme > total\_filme /2 and numar\_filme < total\_filme \*3/4

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

when numar\_filme > total\_filme /4 and numar\_filme < total\_filme /2

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

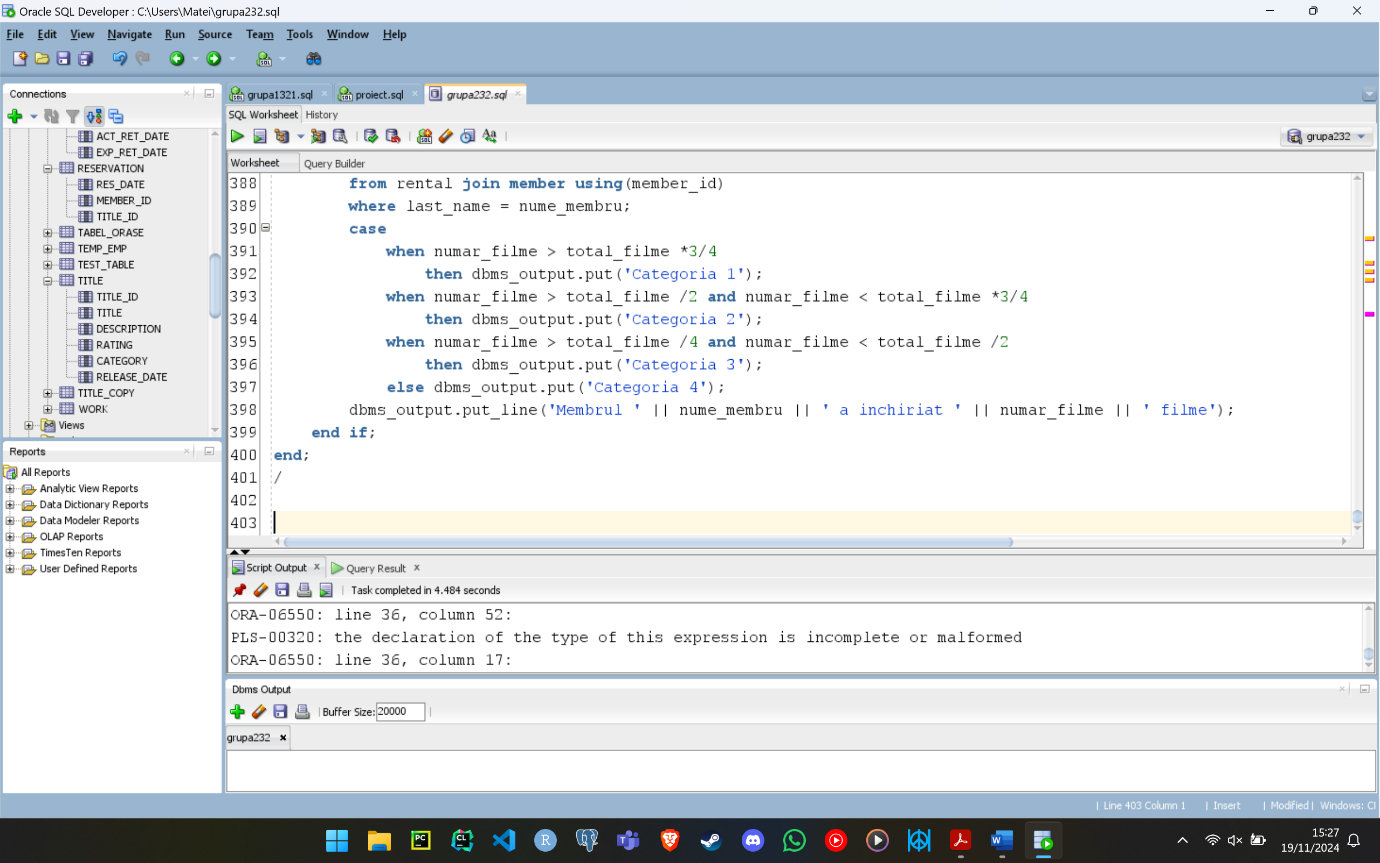
else dbms\_output.put('Categoria 4');

dbms\_output.put\_line('Membrul ' || nume\_membru || ' a inchiriat ' || numar\_filme || ' filme');

end if;

end;

/

Pe asta nu l am mai rulat pentru ca depinde de E3 (care da eroare)

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.

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

alter table member\_ums add constraint pk\_member\_ums primary key (member\_id);

alter table member\_ums add (discount number(2));

declare

numar\_filme number;

total\_filme number;

v\_discount number;

begin

select count(\*)

into total\_filme

from title;

for i in (select member\_id from member\_ums) loop

select count(\*)

into numar\_filme

from rental

where member\_id = i.member\_id;

case

when numar\_filme > total\_filme \*3/4

then v\_discount := 10;

when numar\_filme > total\_filme /2 and numar\_filme < total\_filme \*3/4

then v\_discount := 5;

when numar\_filme > total\_filme /4 and numar\_filme < total\_filme /2

then v\_discount := 3;

else v\_discount := 0;

end case;

update member\_ums

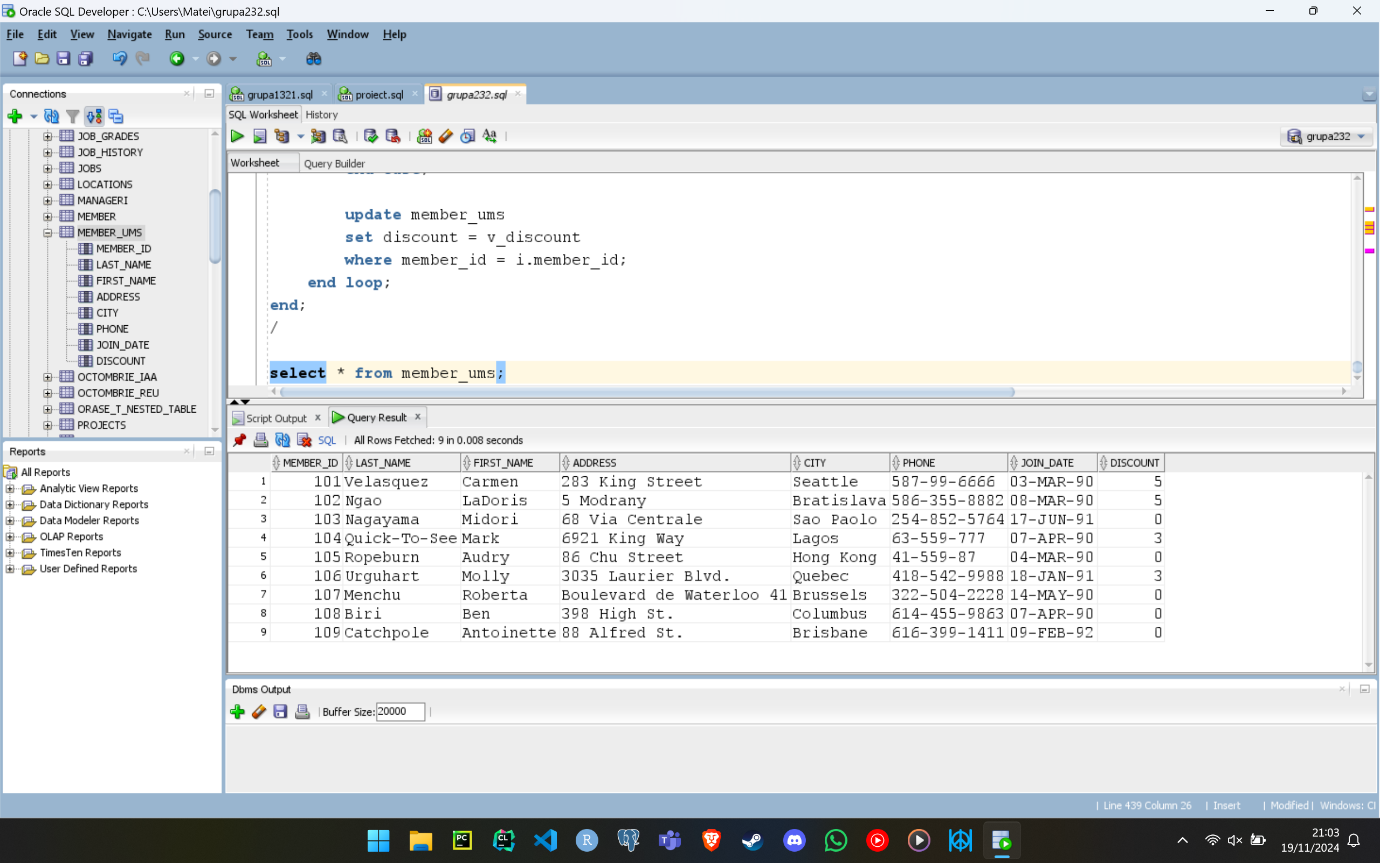
set discount = v\_discount

where member\_id = i.member\_id;

end loop;

end;

/



E6. Enunturile originale:

5. (ENUNT ORIGINAL) Definiți un bloc anonim în care să se afle numele departamentului cu cei mai mulți angajați. Comentați cazul în care există cel puțin două departamente cu număr maxim de angajați. Rezolvați problema utilizând variabile de legătură. Afișați rezultatul atât din bloc, cât și din exteriorul acestuia.

(ENUNT ADAPTAT) Definiti un bloc anonim in care sa se afle numele filmului cu cele mai multe difuzari. Comentati cazul in care exista cel putin doua filme cu numar maxim de difuzari. Rezolvati problema utilizand variabile de legatura. Afisati rezultatul atat din bloc, cat si din exteriorul acestuia.

variable rezultat filme.nume\_film%type;

begin

select count(\*)

into :rezultat

from (

select nume\_film

from filme f, difuzari d

where f.id\_film = d.id\_film

group by nume\_film

having count(\*) = (

select max(count(\*))

from difuzari

group by id\_film));

if :rezultat > 1 then

for i in (

select nume\_film

from filme f, difuzari d

where f.id\_film = d.id\_film

group by nume\_film

having count(\*) = (

select max(count(\*))

from difuzari

group by id\_film)) loop

dbms\_output.put\_line('Filmul ' || i.nume\_film);

end loop;

else

select nume\_film

into :rezultat

from filme f, difuzari d

where f.id\_film = d.id\_film

group by nume\_film

having count(\*) = (

select max(count(\*))

from difuzari

group by id\_film);

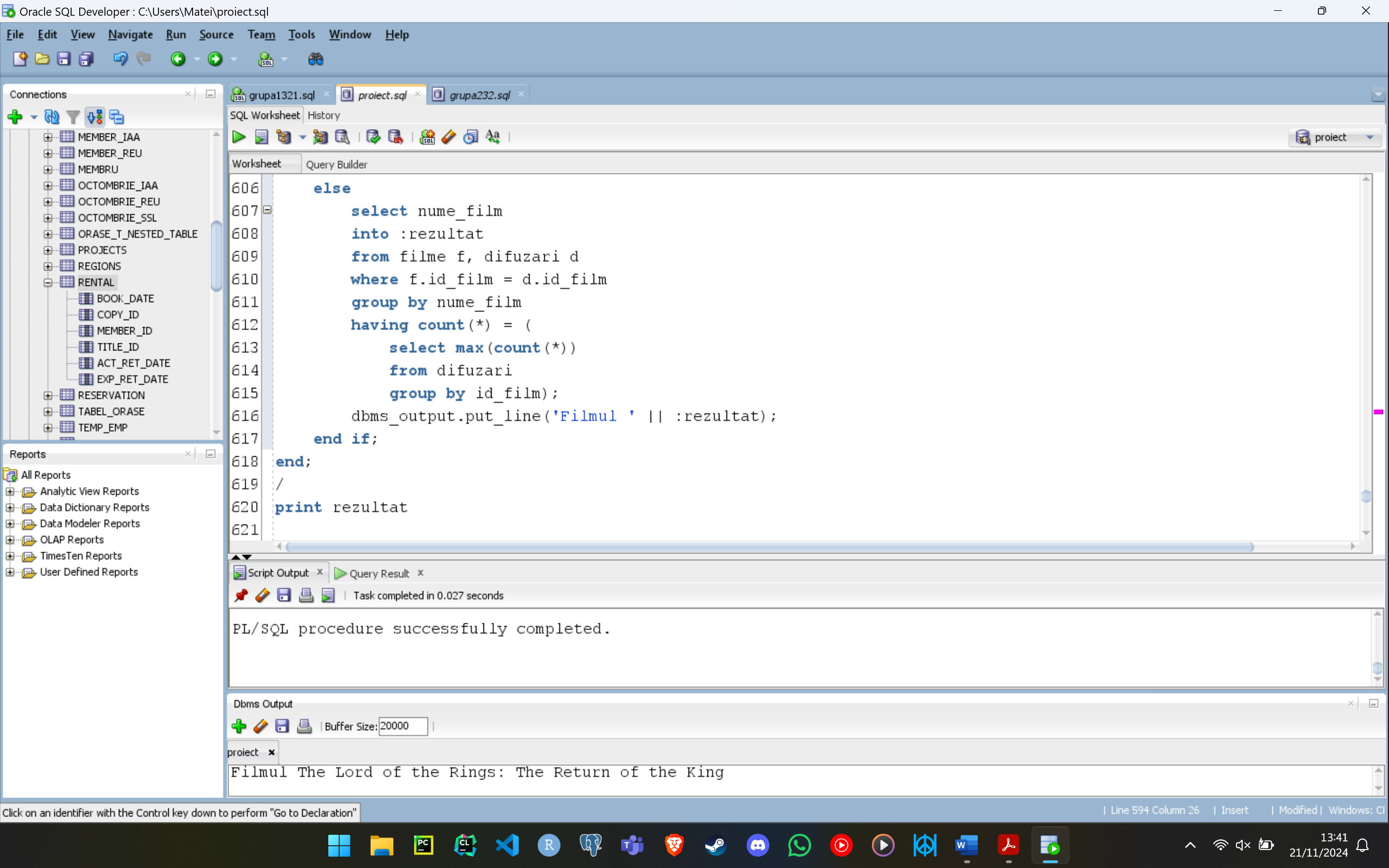
dbms\_output.put\_line('Filmul ' || :rezultat);

end if;

end;

/

print rezultat



7. (ENUNT ORIGINAL) Determinați salariul anual și bonusul pe care îl primește un salariat al cărui cod este dat de la tastatură. Bonusul este determinat astfel: dacă salariul anual este cel puțin 200001, atunci bonusul este 20000; dacă salariul anual este cel puțin 100001 și cel mult 200000, atunci bonusul este 10000, iar dacă salariul anual este cel mult 100000, atunci bonusul este 5000. Afișați bonusul obținut. Comentați cazul în care nu există niciun angajat cu codul introdus.

(ENUNT ADAPTAT) Determinati salariul anual si bonusul pe care il primeste un salariat al carui cod este dat de la tastatura. Bonusul este determinat astfel: daca salariul anual este de cel putin 80001, atunci bonusul este de 10000; daca salariul anual este cel putin 50001 si cel mult 80001, atunci bonusul este 8000, iar daca salariul anual este de cel mult 50000, atunci bonusul este 5000. Afisati bonusul obtinut. Comentati cazul in care nu exista niciun angajat cu codul introdus.

declare

v\_cod angajati.id\_angajat%type := &p\_cod;

v\_bonus number(8);

v\_salariu\_anual number(8);

begin

select count(\*)

into v\_bonus

from angajati

where id\_angajat = v\_cod;

if v\_bonus = 0

then dbms\_output.put\_line('Angajatul ' || v\_cod || ' nu exista.');

else

select salariu\*12

into v\_salariu\_anual

from angajati

where id\_angajat = v\_cod;

if v\_salariu\_anual >= 80001

then v\_bonus := 10000;

elsif v\_salariu\_anual between 50001 and 80000

then v\_bonus := 8000;

else v\_bonus := 5000;

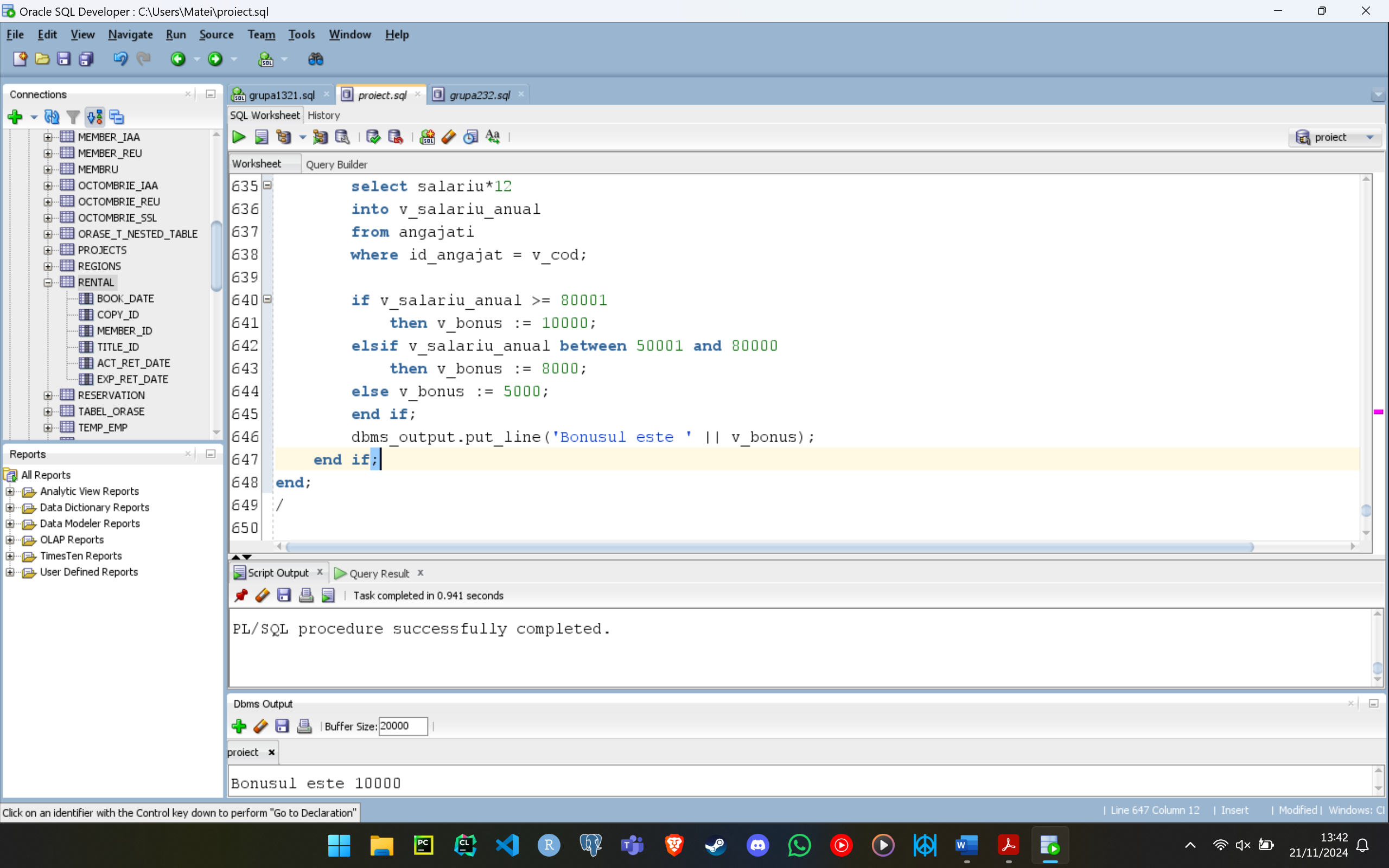
end if;

dbms\_output.put\_line('Bonusul este ' || v\_bonus);

end if;

end;

/



9. (ENUNT ORIGINAL) Scrieți un bloc PL/SQL în care stocați prin variabile de substituție un cod de angajat, un cod de departament și procentul cu care se mărește salariul acestuia. Să se mute salariatul în noul departament și să i se crească salariul în mod corespunzător. Dacă modificarea s-a putut realiza (există în tabelul emp\_\*\*\* un salariat având codul respectiv) să se afișeze mesajul “Actualizare realizata”, iar în caz contrar mesajul “Nu exista un angajat cu acest cod”. Anulați modificările realizate.

(ENUNT ADAPTAT) Scrieti un bloc PL/SQL in care sa stocati prin variabile de substitutie un cod de angajat, un cod de cinema si procentul cu care se mareste salariul acestuia. Sa se mute salariatul in noul cinema si sa i se creasca salariul in mod corespunzator. Daca modificarea s-a putut realiza (exista in tabelul angajati un salariat avand codul respectiv) sa se afiseze mesajul “Actualizare realizata”, iar in caz contrar mesajul “Nu exista un angajat cu acest cod”. Anulati modificarile realizate.

define p\_cod\_sal = 101

define p\_cod\_cinema = 102

define p\_procent = 20

declare

v\_cod\_sal angajati.id\_angajat%type := &p\_cod\_sal;

v\_cod\_cinema angajati.id\_cinema%type := &p\_cod\_cinema;

v\_procent number(8) := &p\_procent;

begin

update angajati

set id\_cinema = v\_cod\_cinema,

salariu = salariu + (salariu \* v\_procent / 100)

where id\_angajat = v\_cod\_sal;

if sql%rowcount = 0

then dbms\_output.put\_line('Nu exista un angajat cu acest cod');

else

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

end if;

end;

/

rollback;

