**Tema 2 Curs**

Ex1: Sa se testeze ce exceptii apar pe comenzile UPDATE si DELETE cu clauza returning (NO\_DATA\_FOUND/TOO\_MANY\_ROWS) - Exemplu 3.8 - (un exercitiu cu DELETE...RETURNING..., alt exercitiu cu UPDATE...RETURNING... si apelati pentru valori care sa complileze fara eroare si alte 2 valori prin care sa incercati sa intrati pe cele  2 exceptii de mai sus). Ce observati? Incercati sa initializati la declarare variabilele folosite in clauza RETURNING.

* **Exemplul pentru care nu ar trebui sa arunce exceptii:**

**Cod sql:**

declare

v\_adresa member\_sma.address%type;

v\_firstname member\_sma.first\_name%type;

v\_lastname member\_sma.last\_name%type;

begin

delete from member\_sma where member\_id=101 --stergem primul membru

--(veasquez carmen 283 king street)

returning last\_name, first\_name, address

into v\_lastname, v\_firstname, v\_adresa;

dbms\_output.put\_line('Datele persoanei sterse sunt:');

dbms\_output.put\_line('Numele: ' || v\_lastname);

dbms\_output.put\_line('Preumele: ' || v\_firstname);

dbms\_output.put\_line('Adresa: ' || v\_adresa);

update member\_sma

set first\_name = 'Carmen2.0'

where member\_id = 111

returning last\_name, first\_name, address

into v\_lastname, v\_firstname, v\_adresa;

dbms\_output.put\_line('Datele persoanei actualizate sunt:');

dbms\_output.put\_line('Numele: ' || v\_lastname);

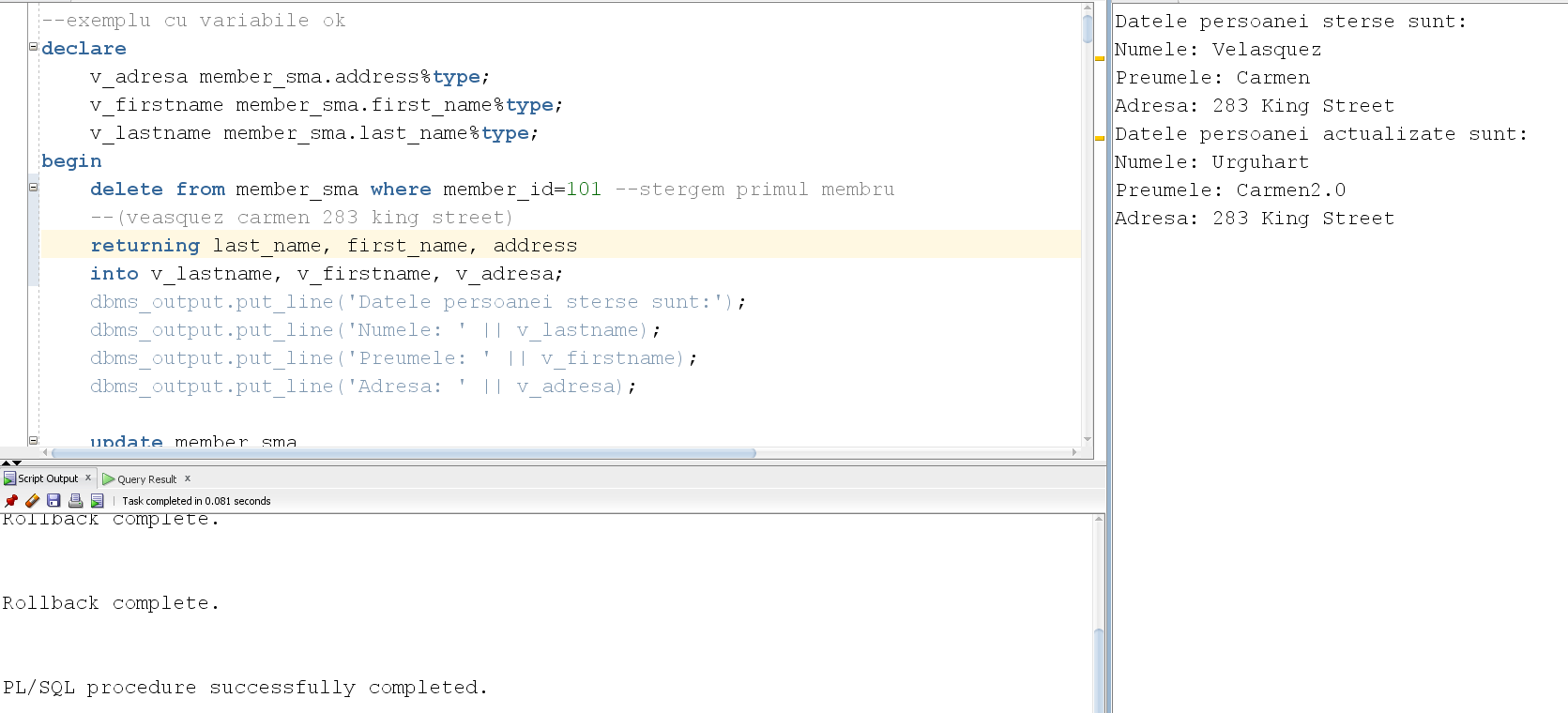
dbms\_output.put\_line('Preumele: ' || v\_firstname);

dbms\_output.put\_line('Adresa: ' || v\_adresa);

end;

/

**Print screen-uri:**



**→ cod pentru care ar trebui sa arunce exceptii (stergere om inexistent/ mai multi in acelasi timp)**

**Cod sql:**

declare --exemplu pt exceptii

v\_adresa member\_sma.address%type; -- := 'a'; initializarea ajuta la

--observarea faptului ca nu se modifica variabilele in cazul in care

--avem teoretic exceptia no\_data\_found

v\_firstname member\_sma.first\_name%type;--:= 'a';

v\_lastname member\_sma.last\_name%type;-- := 'a';

begin

delete from member\_sma where member\_id=10 --stergem un membru care nu exista, ar trb sa intre pe exceptie

--(veasquez carmen 283 king street)

returning last\_name, first\_name, address

into v\_lastname, v\_firstname, v\_adresa;

dbms\_output.put\_line('Datele persoanei sterse sunt:');

dbms\_output.put\_line('Numele: ' || v\_lastname);

dbms\_output.put\_line('Preumele: ' || v\_firstname);

dbms\_output.put\_line('Adresa: ' || v\_adresa);

update member\_sma

set first\_name = 'Carmen2.0'

--where last\_name like ('Urguharttt') --nici aici nu ar trb sa gaseasca deci la fel ca mai sus

--nu intra pe no data found ptc retine nimic si trece mai departe fara sa actualizeze campul

where last\_name like('Urguhart') --pt too\_many\_rows

returning last\_name, first\_name, address

into v\_lastname, v\_firstname, v\_adresa;

dbms\_output.put\_line('Datele persoanei actualizate sunt:');

dbms\_output.put\_line('Numele: ' || v\_lastname);

dbms\_output.put\_line('Preumele: ' || v\_firstname);

dbms\_output.put\_line('Adresa: ' || v\_adresa);

exception

when no\_data\_found then dbms\_output.put\_line('Eroare! Persoana cu acest identificator nu exista');

--nu se va intra pe exceptia no\_data\_found deoarece cand nu se va gasi persoana, valorile in variabilele

--din returning vor ramane null si se va trece mai departe

--deoarece intr-o variabila poate fi null

when too\_many\_rows then dbms\_output.put\_line('Eroare! Exista mai multe persoane cu acest identificator');

--aici se va intra pe exceptie deoarece, spre deosebire de cazul anterior

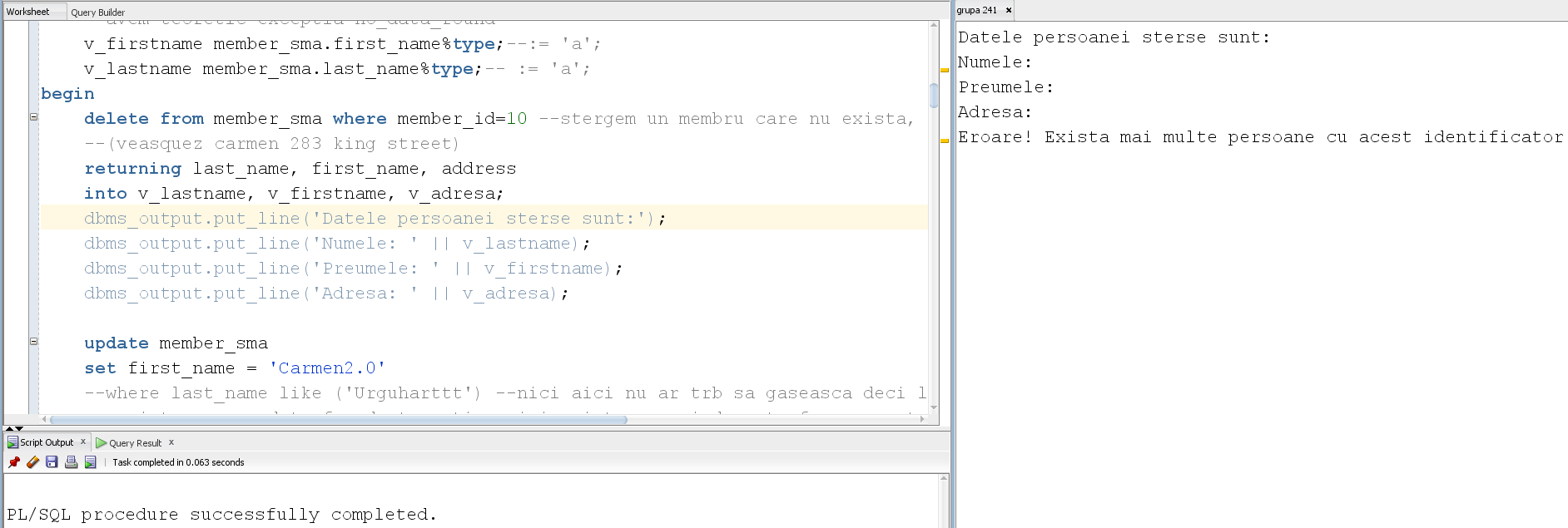
--cand se vor gasi valorile cautate, acestea vor fi mai multe si o variabila nu poate avea mai multe date

--ar fi fost nevoie de o colectie pentru asta, deci in acest caz se va intra in exceptie

end;

/

**Print screen:**

****

**Explicatii pentru exceptii:**

when no\_data\_found then dbms\_output.put\_line('Eroare! Persoana cu acest identificator nu exista');

* Nu se va intra pe exceptia no\_data\_found deoarece cand nu se va gasi persoana, valorile in variabilele din returning vor ramane null si se va trece mai departe deoarece intr-o variabila poate fi null.

when too\_many\_rows then dbms\_output.put\_line('Eroare! Exista mai multe persoane cu acest identificator');

* Aici se va intra pe exceptie deoarece, spre deosebire de cazul anterior cand se vor gasi valorile cautate, acestea vor fi mai multe si o variabila nu poate avea mai multe date ar fi fost nevoie de o colectie pentru asta, deci in acest caz se va intra in exceptie.

Ex2:  Dati 3 exemple asemanatoare cu  exemplele 3.13, 3.14, 3.15, 3.16) in care sa folositi CASE din SQL, respectiv din PL/SQL astfel incat sa afisati pentru un departament dat de la tastatura cati angajati lucreaza in acel departament. Se vor afisa mesaje corespunzatoare folosind comanda CASE din PL/SQL, dar si expresia CASE din SQL. Departamentele vor fi alese astfel incat sa lucreze in ele 1 singur angajat, mai multi angajati sau niciun angajat.

→Case ca SQL statement:

**Cod sql**

select

count(\*),

case

when count(\*) = 0 then 'Nu exista angajati in departamentul ' || department\_id

when count(\*) = 1 then 'Exista un singur angajat in departamentul ' || department\_id

else 'Exista ' || to\_char(count(\*)) || ' angajati in departamentul ' || department\_id

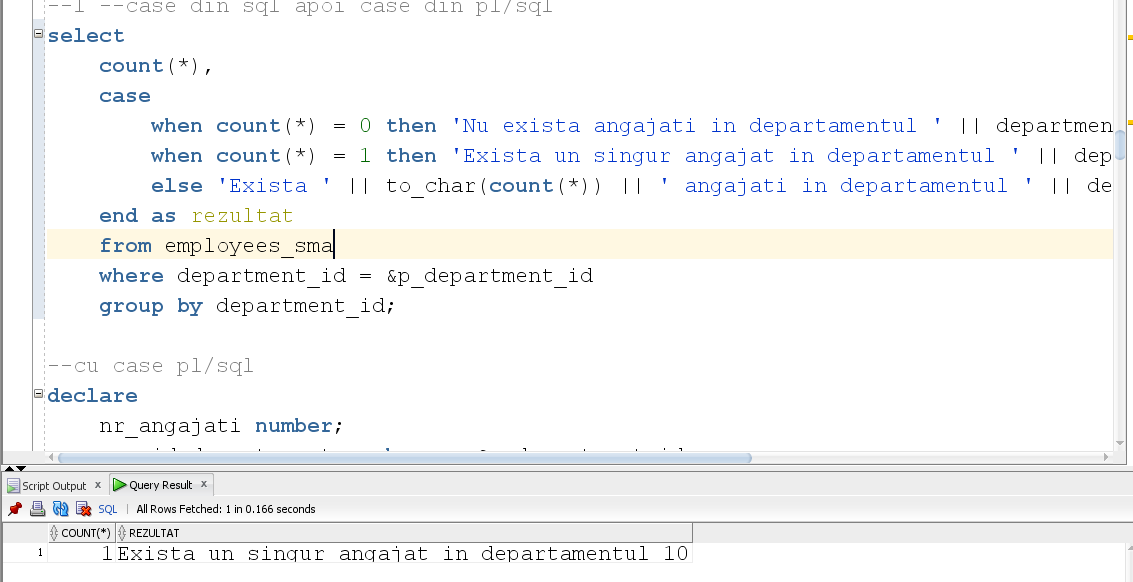
end as rezultat

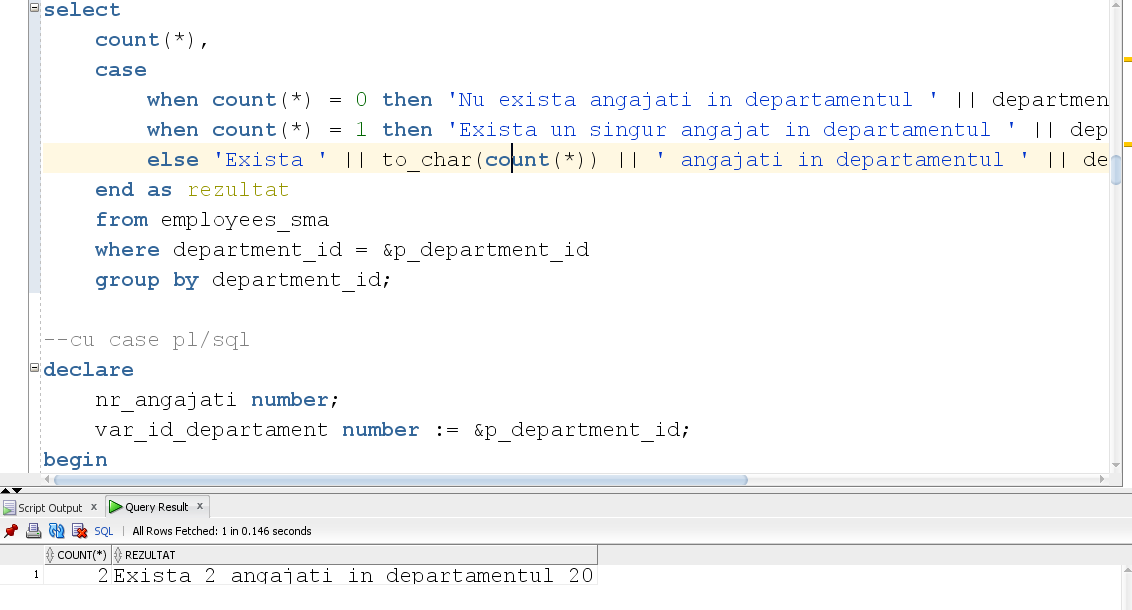
from employees\_sma

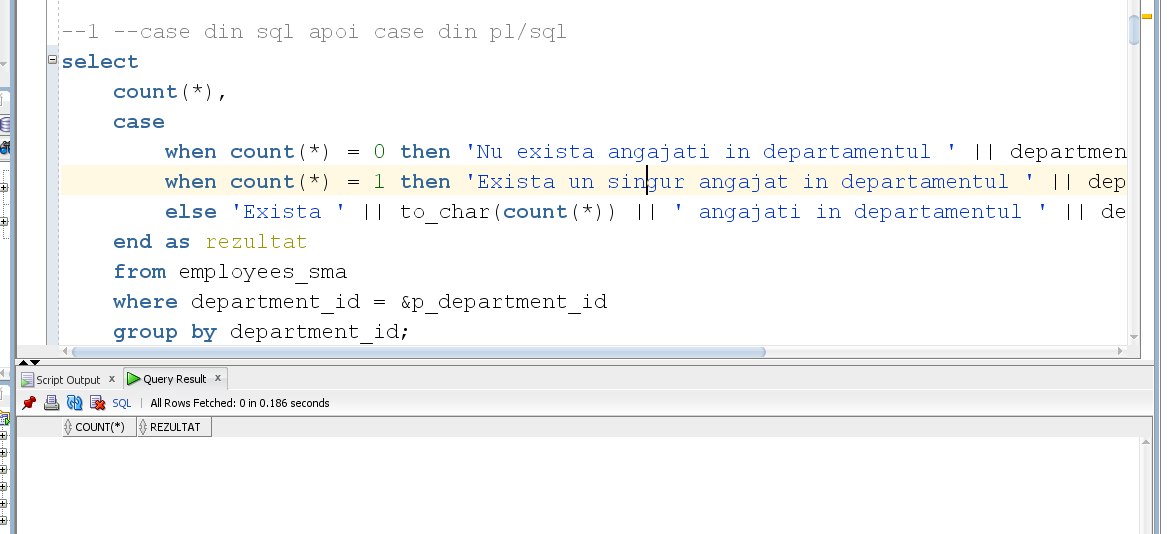
where department\_id = &p\_department\_id

group by department\_id;

**Print screen-uri:**

****

****

****

**→Case ca pl/sql statement similar cu primul exercitiu 3.14**

**Cod sql:**

declare

nr\_angajati number;

var\_id\_departament number := &p\_department\_id;

begin

select count(\*)

into nr\_angajati

from employees\_sma

where department\_id = var\_id\_departament;

case nr\_angajati

when 0 then

dbms\_output.put\_line('Nu exista angajati in departamentul ' || var\_id\_departament);

when 1 then

dbms\_output.put\_line('Exista doar un angajat in departamentul ' || var\_id\_departament);

else

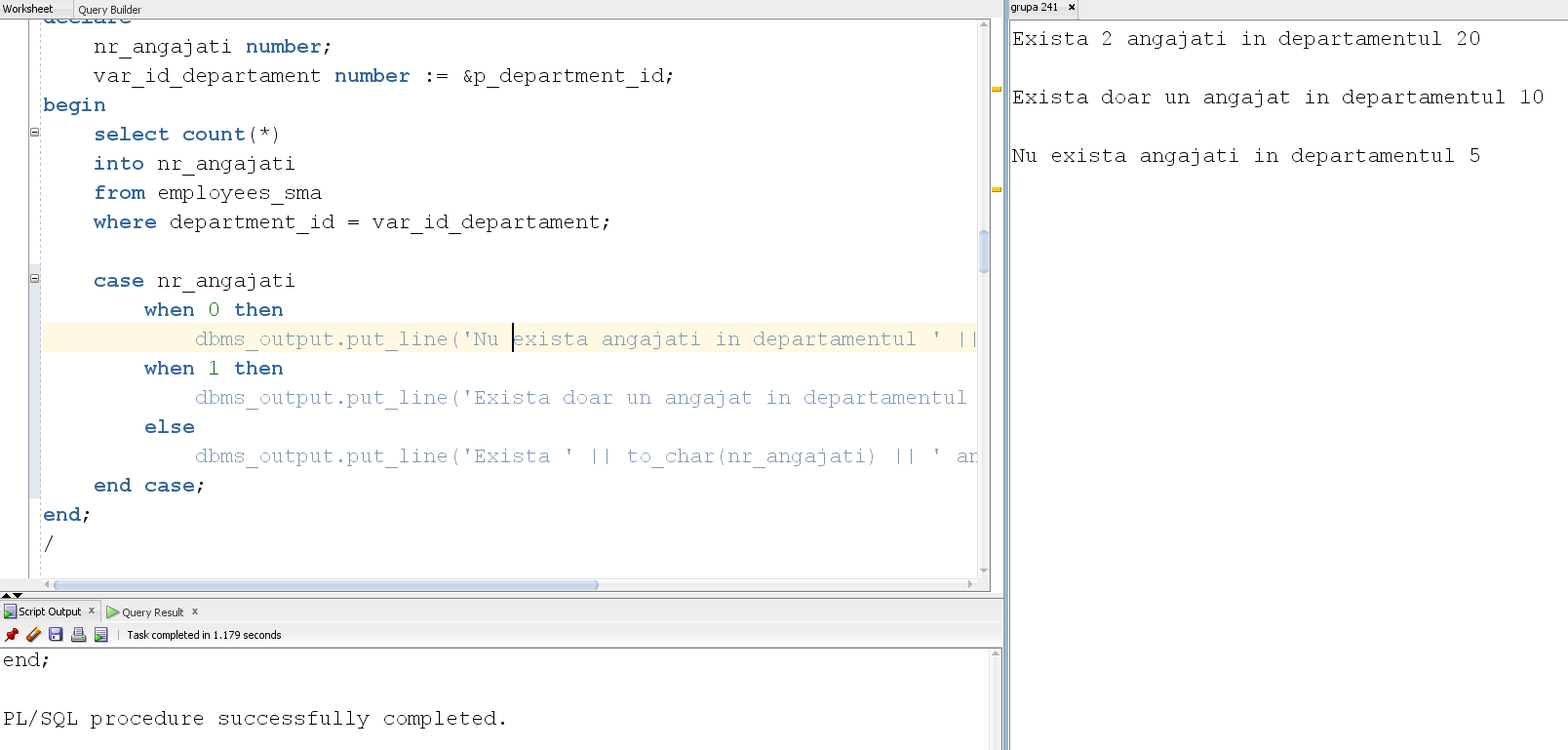
dbms\_output.put\_line('Exista ' || to\_char(nr\_angajati) || ' angajati in departamentul ' || var\_id\_departament);

end case;

end;

/

**Print screen-uri:**

****

**→similar cu primul exercitiu 3.15**

**Cod sql:**

undefine var\_departament\_id

select

case when count(\*) = 0

then 'Nu exista angajati in departamentul ' || &var\_departament\_id

when count(\*) = 1

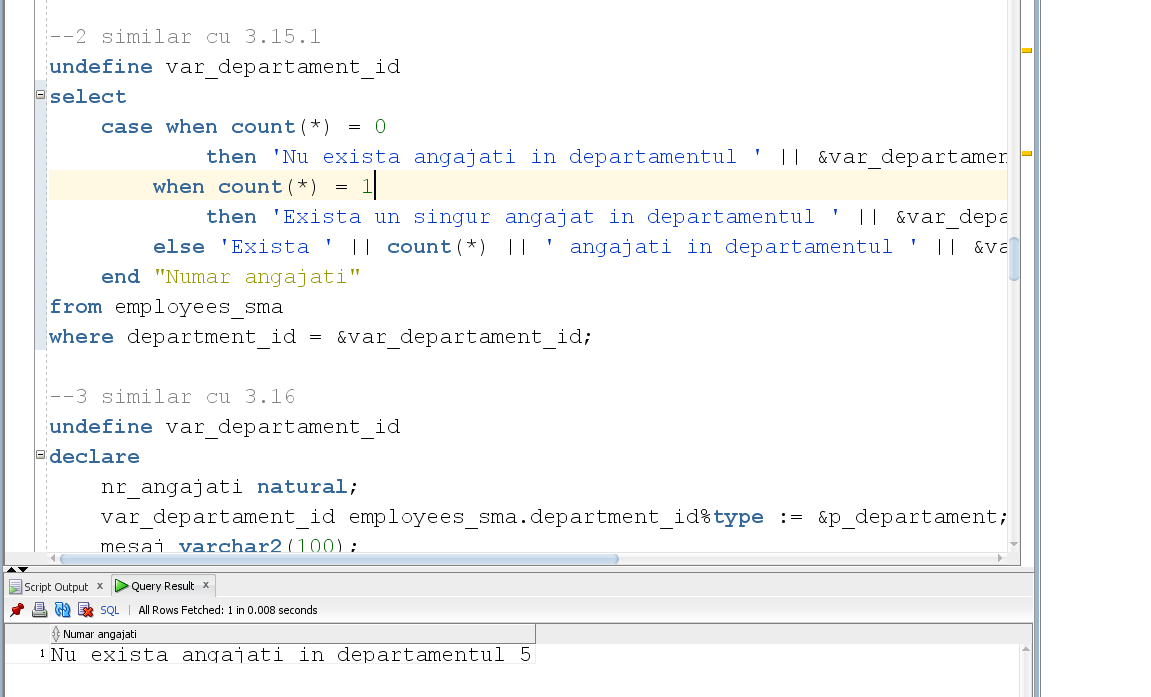
then 'Exista un singur angajat in departamentul ' || &var\_departament\_id

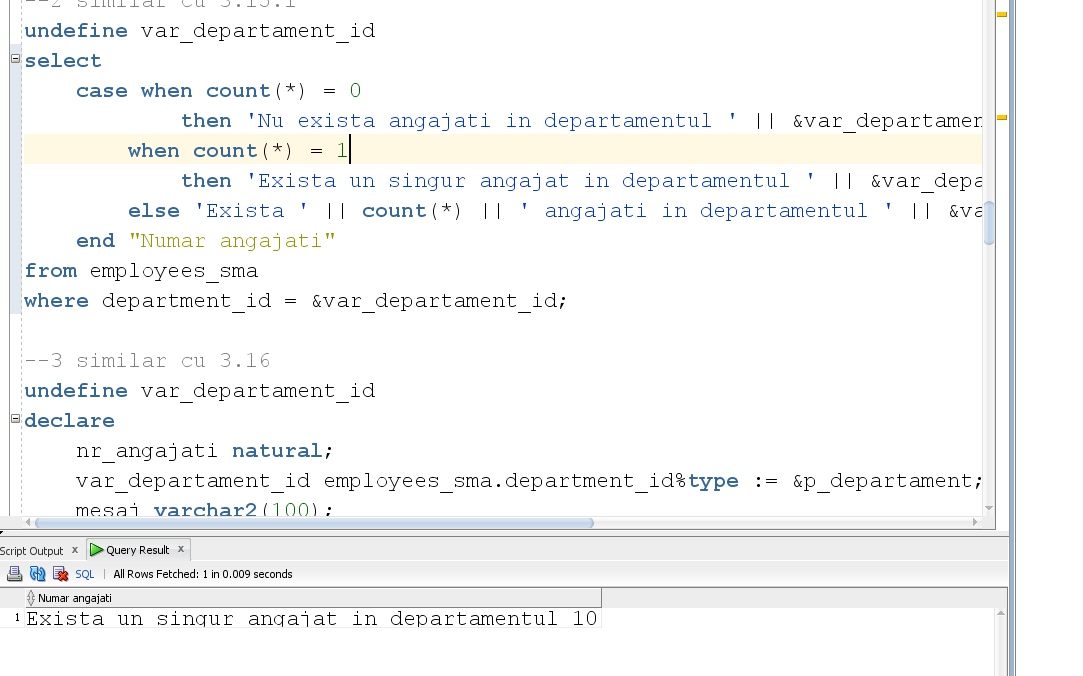
else 'Exista ' || count(\*) || ' angajati in departamentul ' || &var\_departament\_id

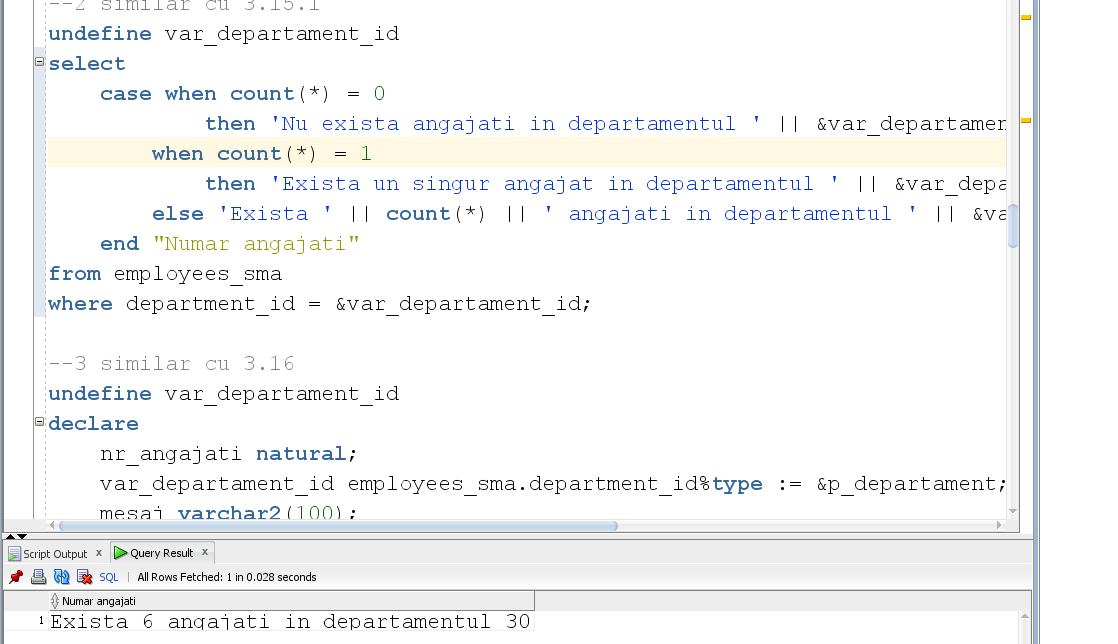
end "Numar angajati"

from employees\_sma

where department\_id = &var\_departament\_id;

**Print screen-uri: **

****

****

**→similar cu exercitiul 3.16**

**Cod sql:**

undefine var\_departament\_id

declare

nr\_angajati natural;

var\_departament\_id employees\_sma.department\_id%type := &p\_departament;

mesaj varchar2(100);

begin

select count(\*) into nr\_angajati

from employees\_sma

where department\_id=var\_departament\_id;

mesaj := case

when nr\_angajati = 0 then

'Nu exista angajati in departamentul '|| var\_departament\_id

when nr\_angajati = 1 then

'Exista doar un angajat in departamentul ' || var\_departament\_id

else

'Exista ' || nr\_angajati || ' angajati in departamentul ' || var\_departament\_id

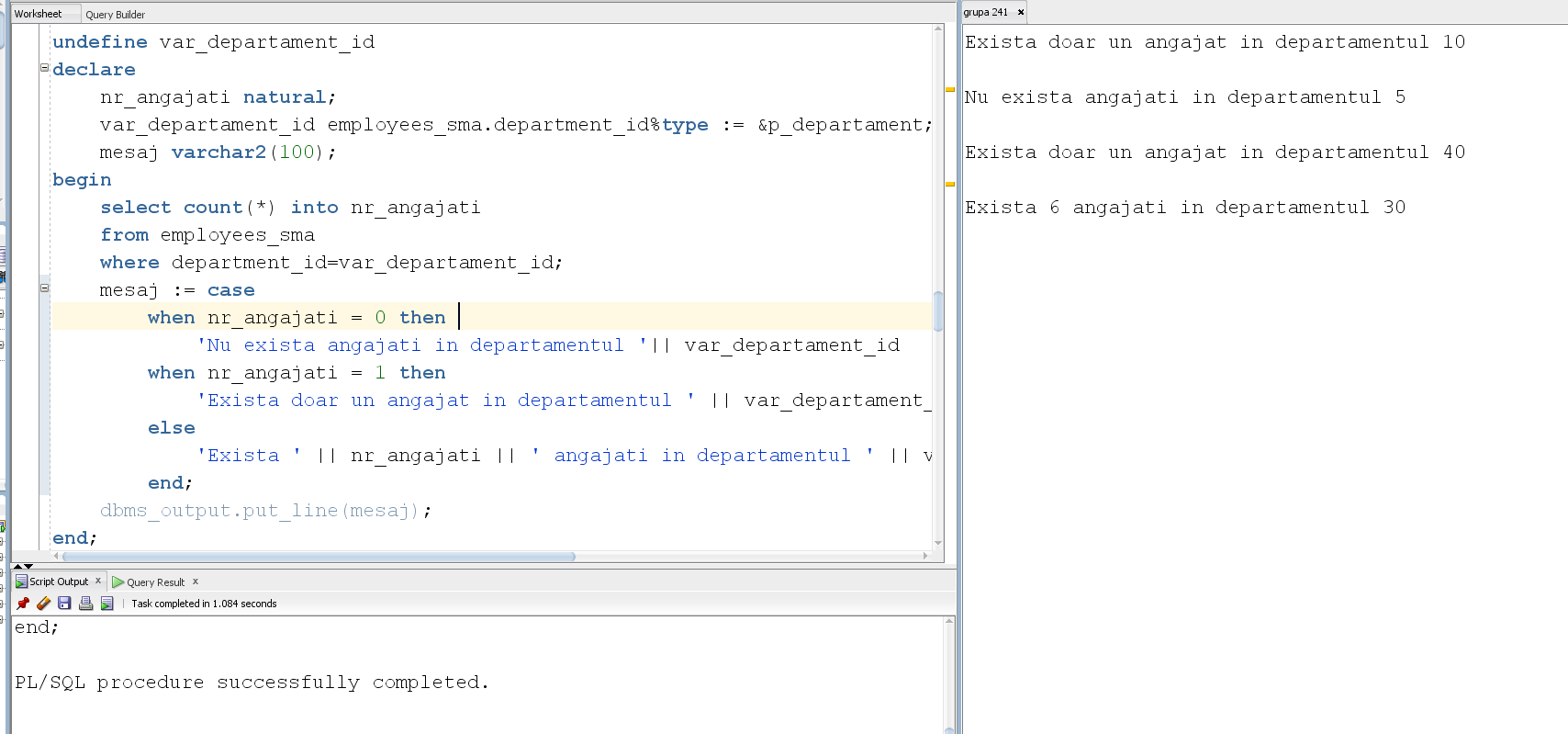
end;

dbms\_output.put\_line(mesaj);

end;

/

**Print screen-uri:**

****

Ex3:

--exemplul  4.12.b adaptat

DECLARE

TYPE tab\_imb IS TABLE OF NUMBER;

t tab\_imb := tab\_imb(1,2,3,4,5);

t\_null tab\_imb;

BEGIN

-- atribuire valori

t.EXTEND(6);

FOR i IN 6..10 LOOP

t(i):=i;

END LOOP;

--parcurgere

DBMS\_OUTPUT.PUT('Tabloul are ' || t.COUNT ||' elemente:');

FOR i IN t.FIRST..t.LAST LOOP

DBMS\_OUTPUT.PUT(t(i) || ' ');

END LOOP;

DBMS\_OUTPUT.NEW\_LINE;

--incercati sa afisati elementele de la pozitiile 11, 12 si -1. Ce observati?

--pentru ce indici se declaseaza exceptiile:

-- -ORA-06533: SUBSCRIPT\_BEYOND\_COUNT

-- -ORA-06532: SUBSCRIPT\_OUTSIDE\_LIMIT

-- -ORA-06531: COLLECTION\_IS\_NULL

--Dati voi exemple pentru a exemplifica apartia acestor trei exceptii

t := t\_null;

IF t IS NULL THEN

DBMS\_OUTPUT.PUT\_LINE('Colectie atomic null');

END IF;

END;

/

**Cod sql:**

--ex 3

DECLARE

TYPE tab\_imb IS TABLE OF NUMBER;

t tab\_imb := tab\_imb(1,2,3,4,5);

t\_null tab\_imb;

BEGIN

-- atribuire valori

t.EXTEND(6);

FOR i IN 6..10 LOOP

t(i):=i;

END LOOP;

--parcurgere

DBMS\_OUTPUT.PUT('Tabloul are ' || t.COUNT ||' elemente:');

FOR i IN t.FIRST..t.LAST LOOP

DBMS\_OUTPUT.PUT(t(i) || ' ');

END LOOP;

DBMS\_OUTPUT.NEW\_LINE;

dbms\_output.put\_line('Elementele de pe poz 11 12 si -1');

dbms\_output.put\_line(t(11)); --nu ridica exceptie ptc am extins cu 6 deci are si pozitia 11

--dbms\_output.put\_line(t(12)); -- subscript beyond count ridica exceptie ptc nu avem pozitia 12

--dbms\_output.put\_line(t(-1)); --subsript outside of limit --ridica exceptie ptc nu avem pozitia -1

--un alt exemplu pentru a ridica exceptia subscript beyond count este t(15) sau orice indice mai mare decat 11,

--colectia ar fi trebuit extinsa pentru a putea accesa un indice cu acest numar

--un alt exemplu pentru a ridica exceptia subscript outside of limit ar fi orice indice < 1 deoarece

--exceptia este ridicata in momentul in care se foloseste un index care nu poate fi in intervalul valid de pozitii

--pt o colectie. Cum ele sunt indexate de la 1 in pl/sql, orice valoarea < 1 va ridica aceasta exceptie

--exceptia colection is null este ridicata atunci cand incercam sa accesam in orice fel date dinauntrul unei

--colectii care nu a fost initializata in prealabil

t := t\_null;

IF t IS NULL THEN

DBMS\_OUTPUT.PUT\_LINE('Colectie atomic null');

END IF;

END;

/

**Explicatii**: --nu ridica exceptie ptc am extins cu 6 deci are si pozitia 11

--dbms\_output.put\_line(t(12)); -- subscript beyond count ridica exceptie ptc nu avem pozitia 12

--dbms\_output.put\_line(t(-1)); --subsript outside of limit --ridica exceptie ptc nu avem pozitia -1

--un alt exemplu pentru a ridica exceptia subscript beyond count este t(15) sau orice indice mai mare decat 11,

--colectia ar fi trebuit extinsa pentru a putea accesa un indice cu acest numar

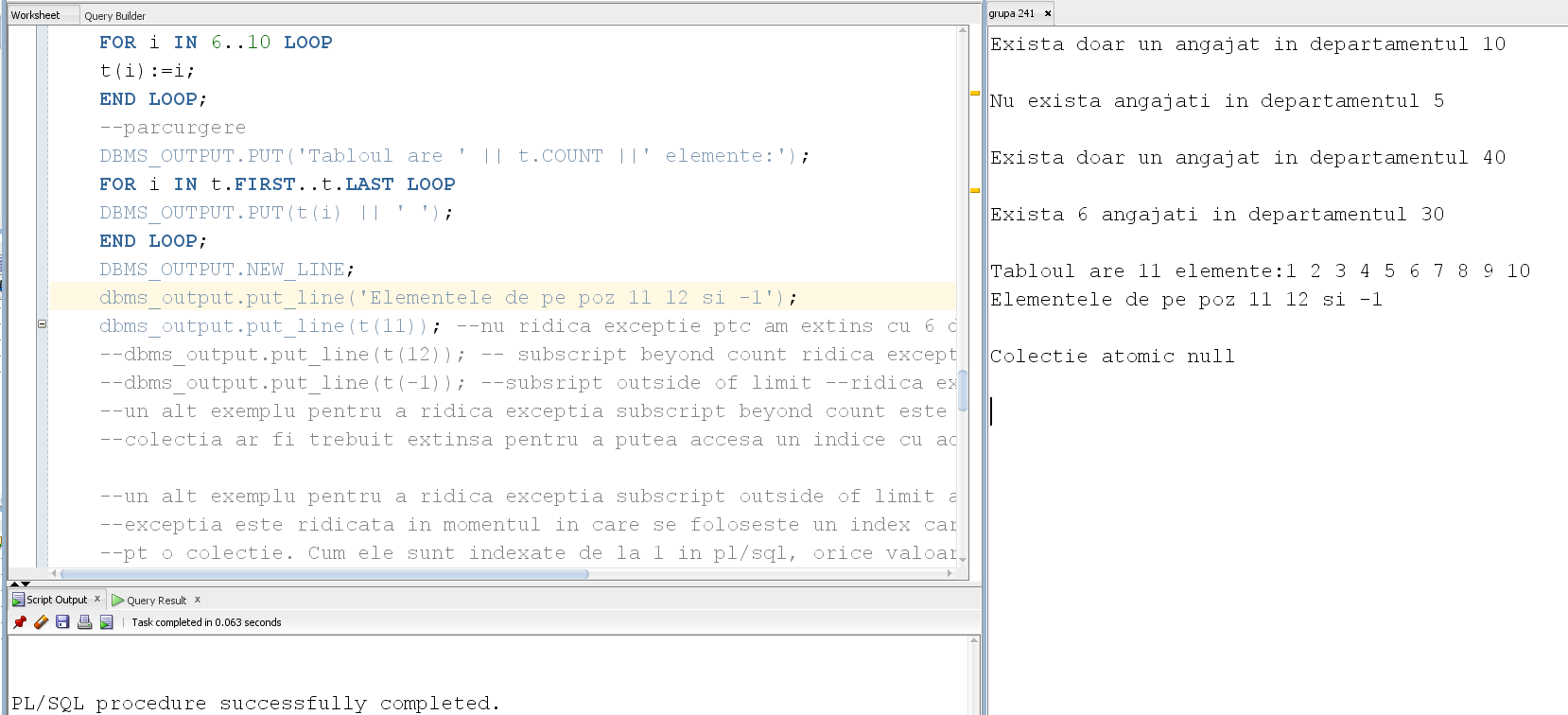
--un alt exemplu pentru a ridica exceptia subscript outside of limit ar fi orice indice < 1 deoarece

--exceptia este ridicata in momentul in care se foloseste un index care nu poate fi in intervalul valid de pozitii

--pt o colectie. Cum ele sunt indexate de la 1 in pl/sql, orice valoarea < 1 va ridica aceasta exceptie

--exceptia colection is null este ridicata atunci cand incercam sa accesam in orice fel date dinauntrul unei

--colectii care nu a fost initializata in prealabil

**Print screen-uri: **

Ex4:

--Creati cate un exercitiu asemanator cu EX3 folosind tablou indexat si altul folosind vectori  in care sa incercati sa stergeti primul si ultimul element si 2 elemente din interior. Afisati numarul de elemente din fiecare colectie, cat si fiecare element ramas. Discutati ce exceptii pot aparea.

**Cod sql (tabolu):**

declare

type tab\_imb is table of number;

t tab\_imb := tab\_imb(13, 23, 33, 44, 52, 6, 73, 8);

begin

--stergere primul si ultimul element

t.delete(1);

t.delete(t.last);

--stergere doua elemente din interiorul colectiei

t.delete(2);

t.delete(t.last);

--afisare elemente ramase

dbms\_output.put\_line('Tabloul are ' || t.count || ' elemente');

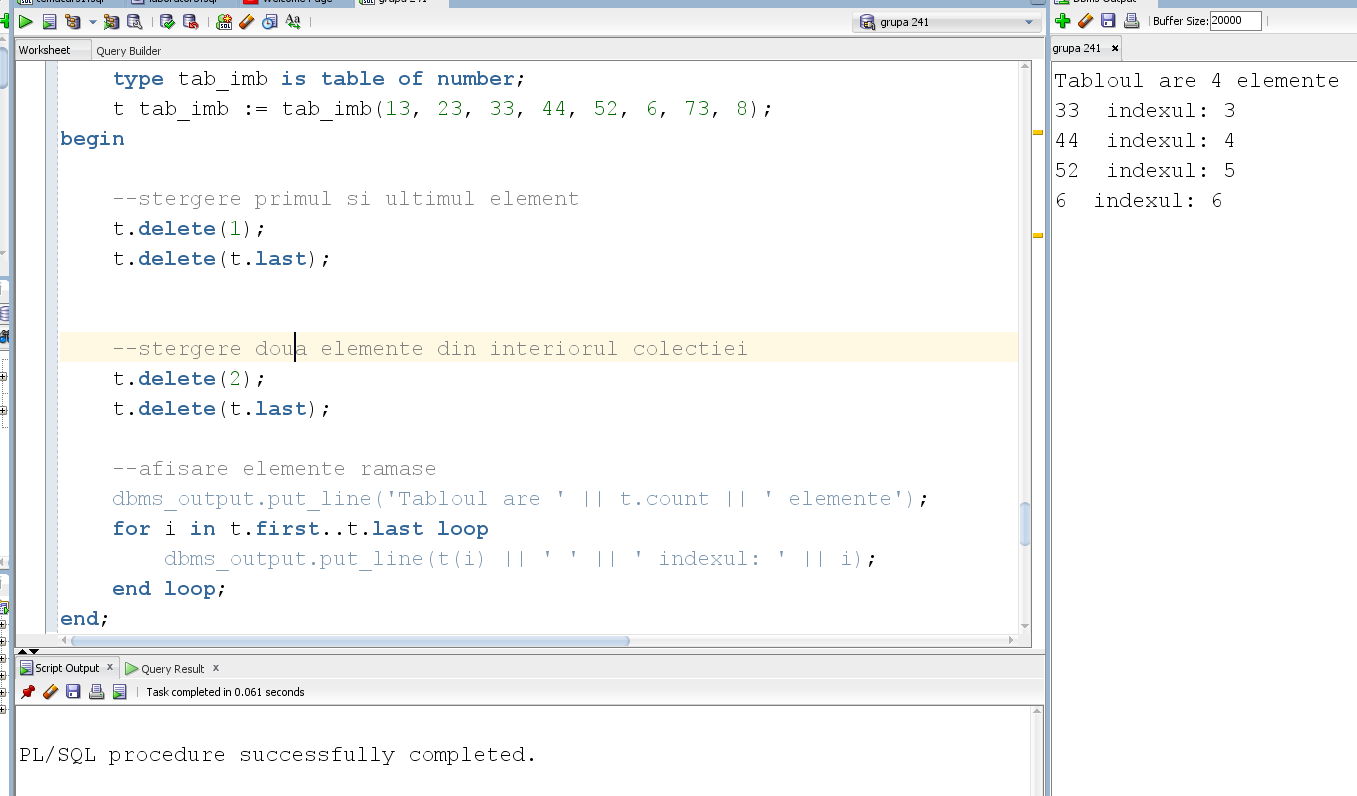
for i in t.first..t.last loop

dbms\_output.put\_line(t(i) || ' ' || ' indexul: ' || i);

end loop;

end;

/

**Print screen-uri: **

**Cod sql(vector):**

declare

type vector\_type is varray(10) of number;

v vector\_type := vector\_type(1, 2, 3, 4, 5, 6, 7, 8, 9);

new\_v vector\_type := vector\_type();

begin

--copiez elementele din v in noul v, excluzand primul si ultimul si doua elemente din interior

for i in 3..v.last - 2 loop

new\_v.extend;

new\_v(new\_v.count) := v(i);

end loop;

--afisare nr de elemente si elementele ramase din noul vector(adica vectorul modificat)

dbms\_output.put\_line('Vectorul are ' || new\_v.count || ' elemente ramase: ');

for i in 1..new\_v.count loop

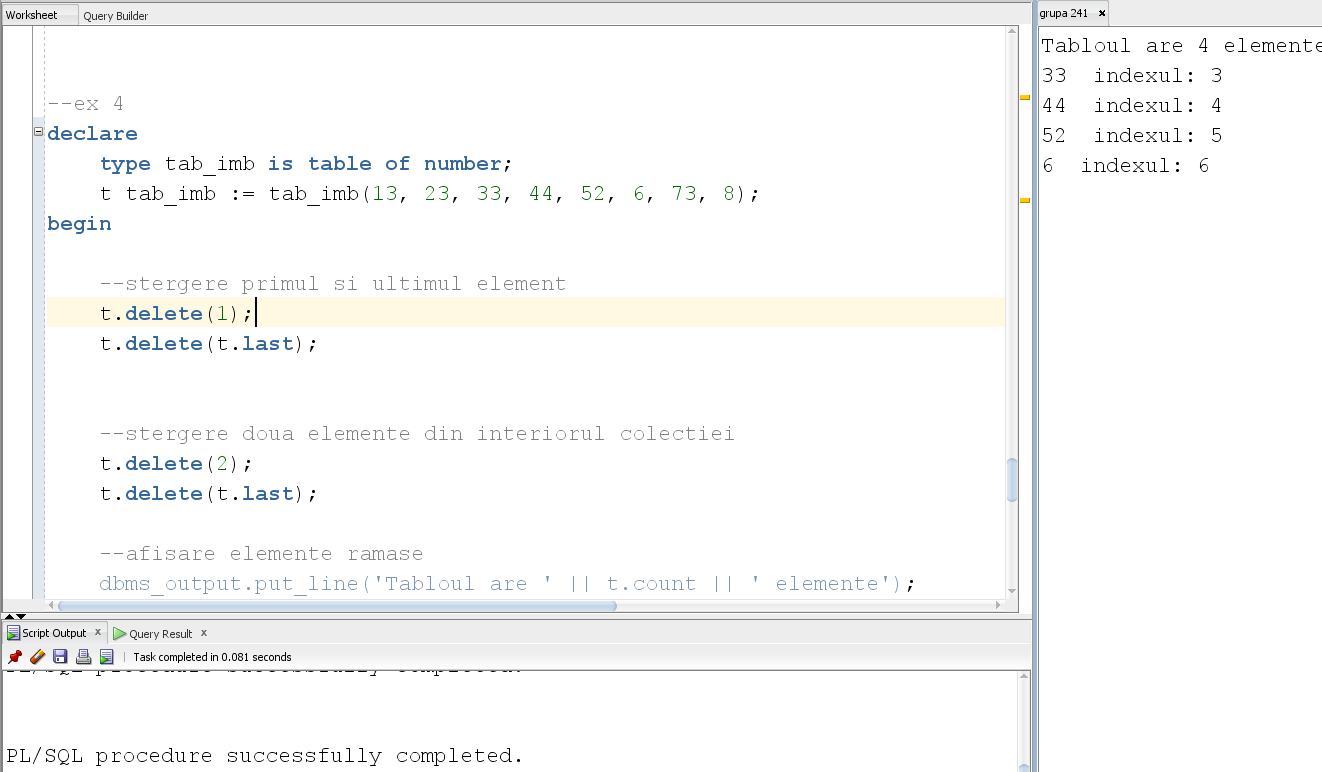
dbms\_output.put\_line(new\_v(i) || ' indexul: ' || i);

end loop;

end;

/

**Print screen-uri:**



Exceptii probabile sa apara:

- no\_data\_found: atunci cand incercam sa stergem un element care nu (mai) exista

- subscript\_beyond\_count: atunci cand incercam sa stergem un element folosind un index care depaseste numarul curent de elemente din colectie

- subscript\_outside\_limit: atunci cand incercam sa stergem elemente folosind un index care este in afara range-ului posibil de idecsi (t(0))