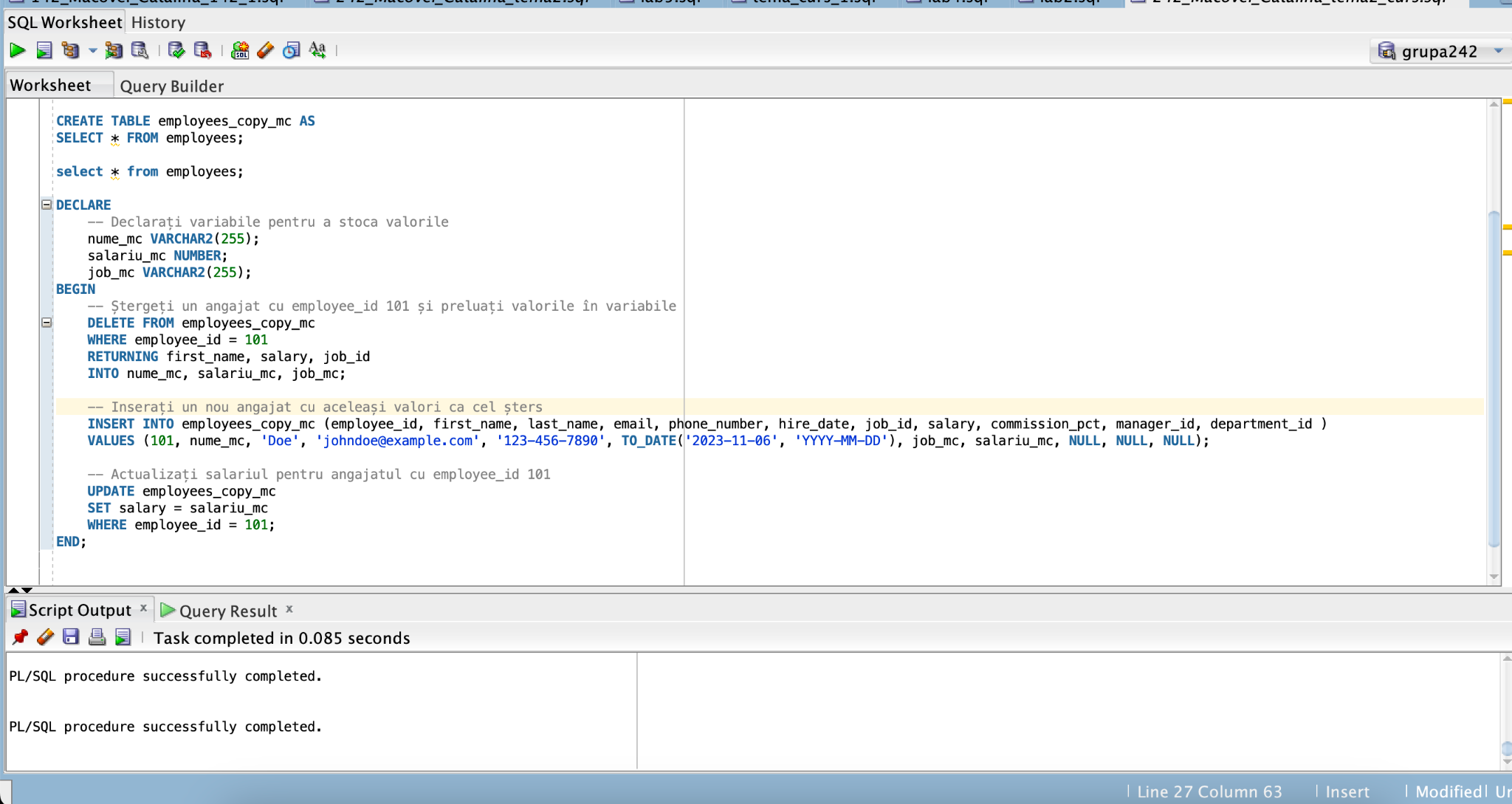
SGBD Curs **Tema 2**

short line

Macovei Catalina  
Grupa 242

Exercițiul 1



Cod sursa fără excepții

CREATE TABLE employees\_copy\_mc AS

SELECT \* FROM employees;

select \* from employees;

DECLARE

-- Declarați variabile pentru a stoca valorile

nume\_mc VARCHAR2(255);

salariu\_mc NUMBER;

job\_mc VARCHAR2(255);

BEGIN

-- Ștergeți un angajat cu employee\_id 101 și preluați valorile în variabile

DELETE FROM employees\_copy\_mc

WHERE employee\_id = 101

RETURNING first\_name, salary, job\_id

INTO nume\_mc, salariu\_mc, job\_mc;

-- Inserați un nou angajat cu aceleași valori ca cel șters

INSERT INTO employees\_copy\_mc (employee\_id, first\_name, last\_name, email, phone\_number, hire\_date, job\_id, salary, commission\_pct, manager\_id, department\_id )

VALUES (101, nume\_mc, 'Doe', 'johndoe@example.com', '123-456-7890', TO\_DATE('2023-11-06', 'YYYY-MM-DD'), job\_mc, salariu\_mc, NULL, NULL, NULL);

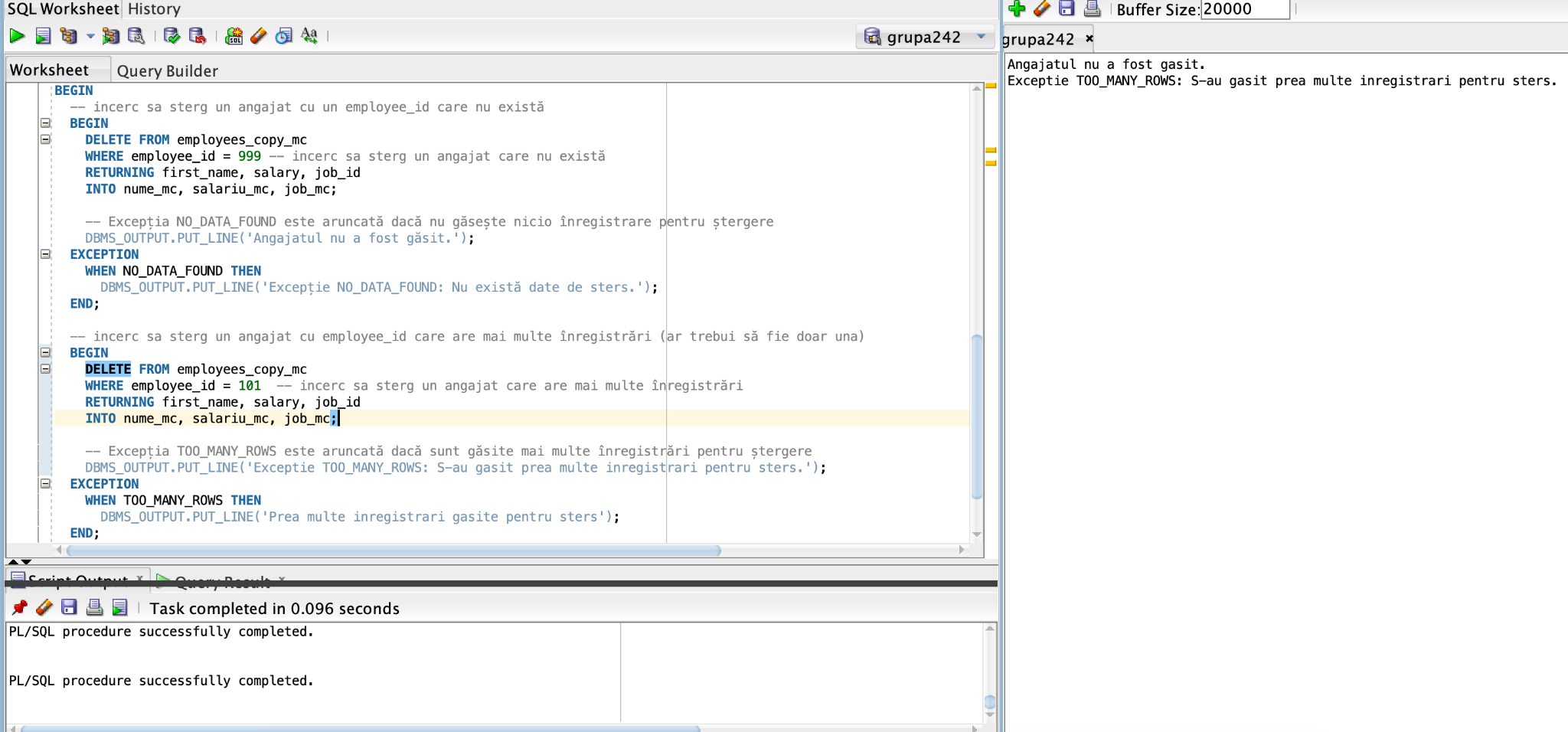
-- Actualizați salariul pentru angajatul cu employee\_id 101

UPDATE employees\_copy\_mc

SET salary = salariu\_mc

WHERE employee\_id = 101;

END;



Cod sursa cu excepții pentru DELETE

DECLARE

nume\_mc VARCHAR2(255);

salariu\_mc NUMBER;

job\_mc VARCHAR2(255);

BEGIN

-- incerc sa sterg un angajat cu un employee\_id care nu există

BEGIN

DELETE FROM employees\_copy\_mc

WHERE employee\_id = 999 -- incerc sa sterg un angajat care nu există

RETURNING first\_name, salary, job\_id

INTO nume\_mc, salariu\_mc, job\_mc;

-- Excepția NO\_DATA\_FOUND este aruncată dacă nu găsește nicio înregistrare pentru ștergere

DBMS\_OUTPUT.PUT\_LINE('Angajatul nu a fost găsit.');

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Excepție NO\_DATA\_FOUND: Nu există date de sters.');

END;

-- incerc sa sterg un angajat cu employee\_id care are mai multe înregistrări (ar trebui să fie doar una)

BEGIN

DELETE FROM employees\_copy\_mc

WHERE employee\_id = 101 -- incerc sa sterg un angajat care are mai multe înregistrări

RETURNING first\_name, salary, job\_id

INTO nume\_mc, salariu\_mc, job\_mc;

-- Excepția TOO\_MANY\_ROWS este aruncată dacă sunt găsite mai multe înregistrări pentru ștergere

DBMS\_OUTPUT.PUT\_LINE('Exceptie TOO\_MANY\_ROWS: S-au gasit prea multe inregistrari pentru sters.');

EXCEPTION

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('Prea multe inregistrari gasite pentru sters');

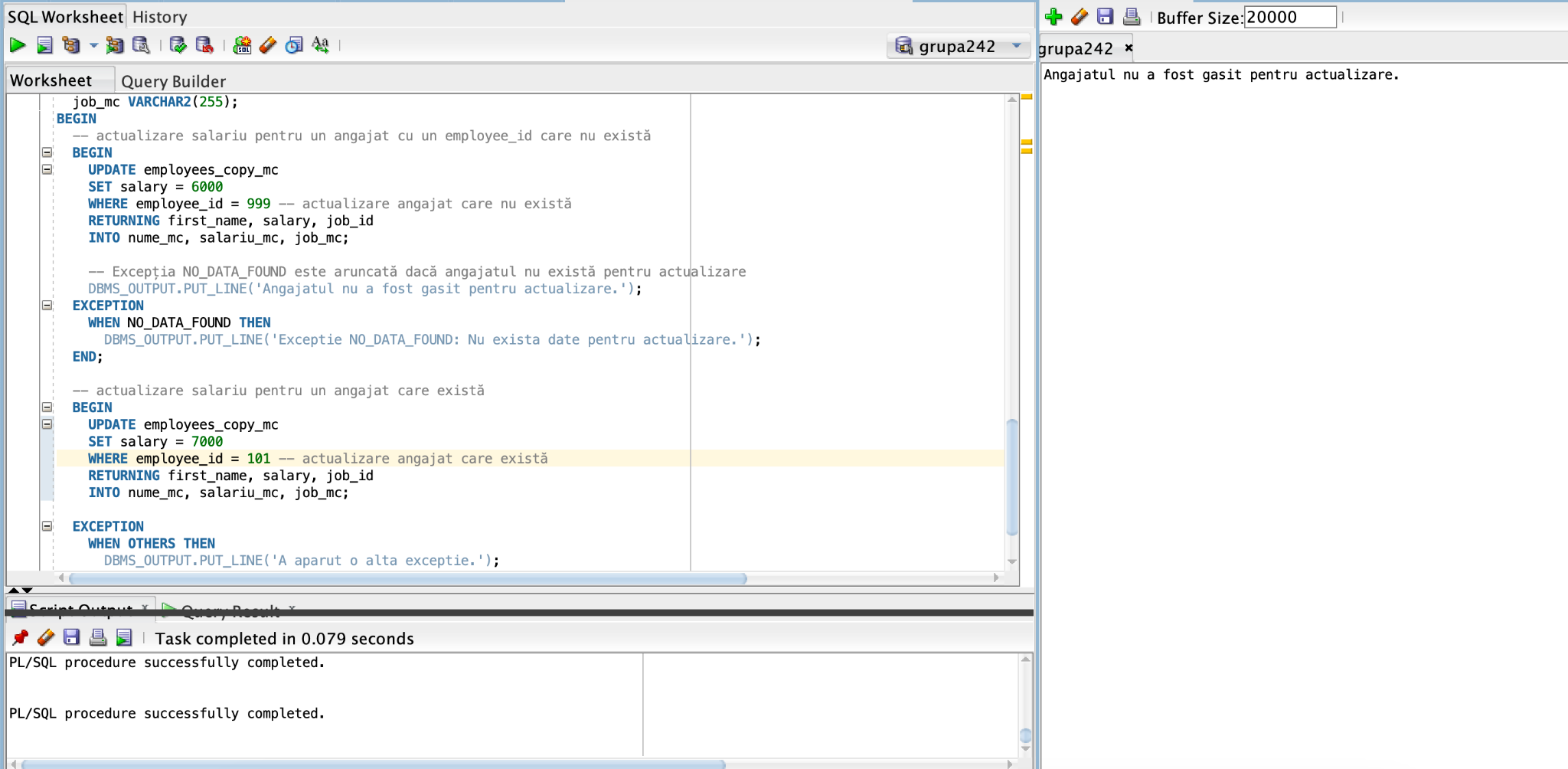
END;

END;

/

OBSERVAȚII:

* Excepția NO\_DATA\_FOUND este aruncată dacă nu găsește nicio înregistrare pentru ștergere
* Excepția TOO\_MANY\_ROWS este aruncată dacă sunt găsite mai multe înregistrări pentru ștergere



Cod sursa excepții pentru UPDATE:

DECLARE

nume\_mc VARCHAR2(255);

salariu\_mc NUMBER;

job\_mc VARCHAR2(255);

BEGIN

-- actualizare salariu pentru un angajat cu un employee\_id care nu există

BEGIN

UPDATE employees\_copy\_mc

SET salary = 6000

WHERE employee\_id = 999 -- actualizare angajat care nu există

RETURNING first\_name, salary, job\_id

INTO nume\_mc, salariu\_mc, job\_mc;

-- Excepția NO\_DATA\_FOUND este aruncată dacă angajatul nu există pentru actualizare

DBMS\_OUTPUT.PUT\_LINE('Angajatul nu a fost gasit pentru actualizare.');

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Exceptie NO\_DATA\_FOUND: Nu exista date pentru actualizare.');

END;

-- actualizare salariu pentru un angajat care există

BEGIN

UPDATE employees\_copy\_mc

SET salary = 7000

WHERE employee\_id = 101 -- actualizare angajat care există

RETURNING first\_name, salary, job\_id

INTO nume\_mc, salariu\_mc, job\_mc;

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('A aparut o alta exceptie.');

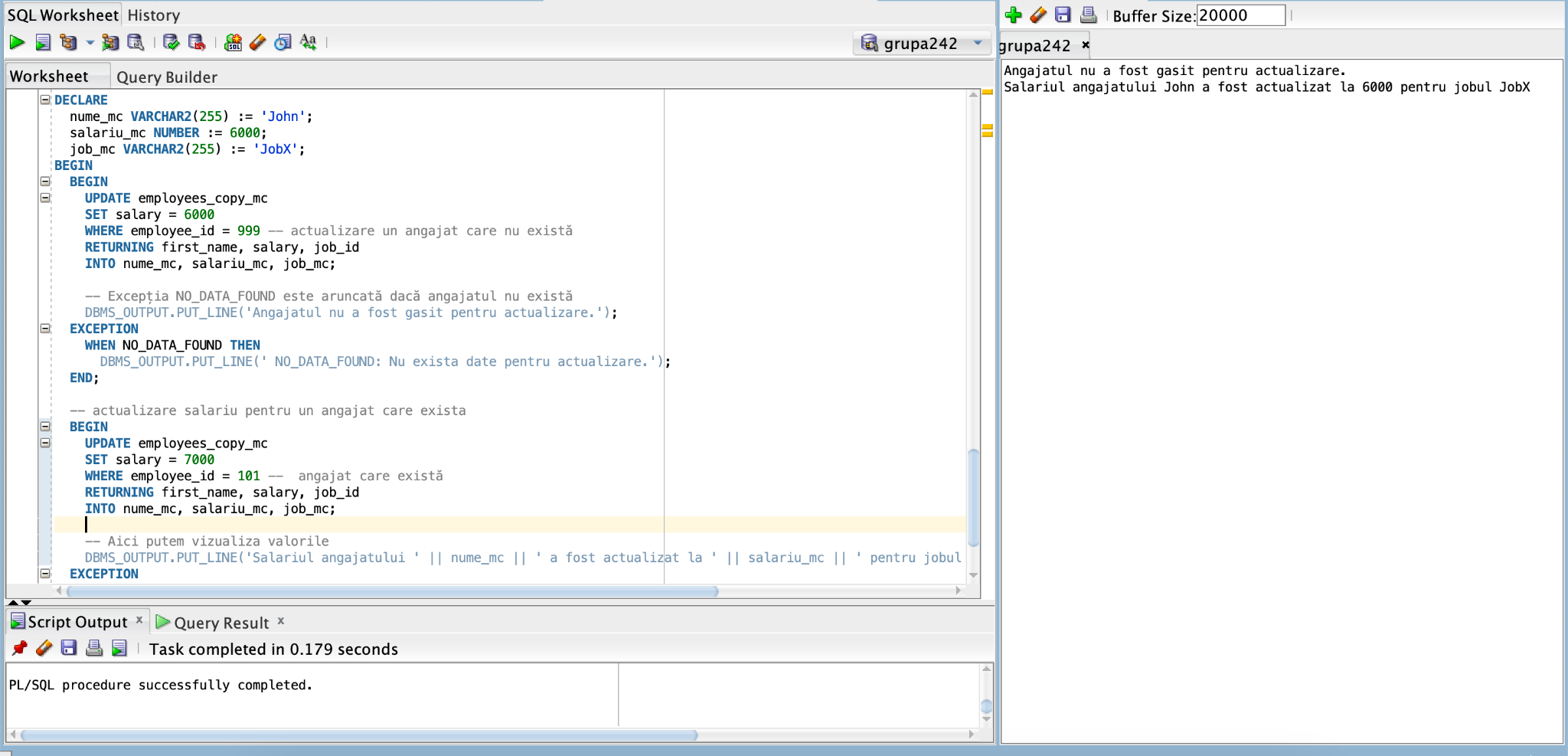
END;

END;

/

OBSERVAȚII:

* Excepția NO\_DATA\_FOUND este aruncată dacă angajatul nu există pentru actualizare
* Pentru angajat care exista nu se arunca excepții



Cod sursa inițializare variabile:

DECLARE

nume\_mc VARCHAR2(255) := 'John';

salariu\_mc NUMBER := 6000;

job\_mc VARCHAR2(255) := 'JobX';

BEGIN

BEGIN

UPDATE employees\_copy\_mc

SET salary = 6000

WHERE employee\_id = 999 -- actualizare un angajat care nu există

RETURNING first\_name, salary, job\_id

INTO nume\_mc, salariu\_mc, job\_mc;

-- Excepția NO\_DATA\_FOUND este aruncată dacă angajatul nu există

DBMS\_OUTPUT.PUT\_LINE('Angajatul nu a fost gasit pentru actualizare.');

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE(' NO\_DATA\_FOUND: Nu exista date pentru actualizare.');

END;

-- actualizare salariu pentru un angajat care exista

BEGIN

UPDATE employees\_copy\_mc

SET salary = 7000

WHERE employee\_id = 101 -- angajat care există

RETURNING first\_name, salary, job\_id

INTO nume\_mc, salariu\_mc, job\_mc;

-- Aici putem vizualiza valorile

DBMS\_OUTPUT.PUT\_LINE('Salariul angajatului ' || nume\_mc || ' a fost actualizat la ' || salariu\_mc || ' pentru jobul ' || job\_mc);

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('alta exceptie.');

END;

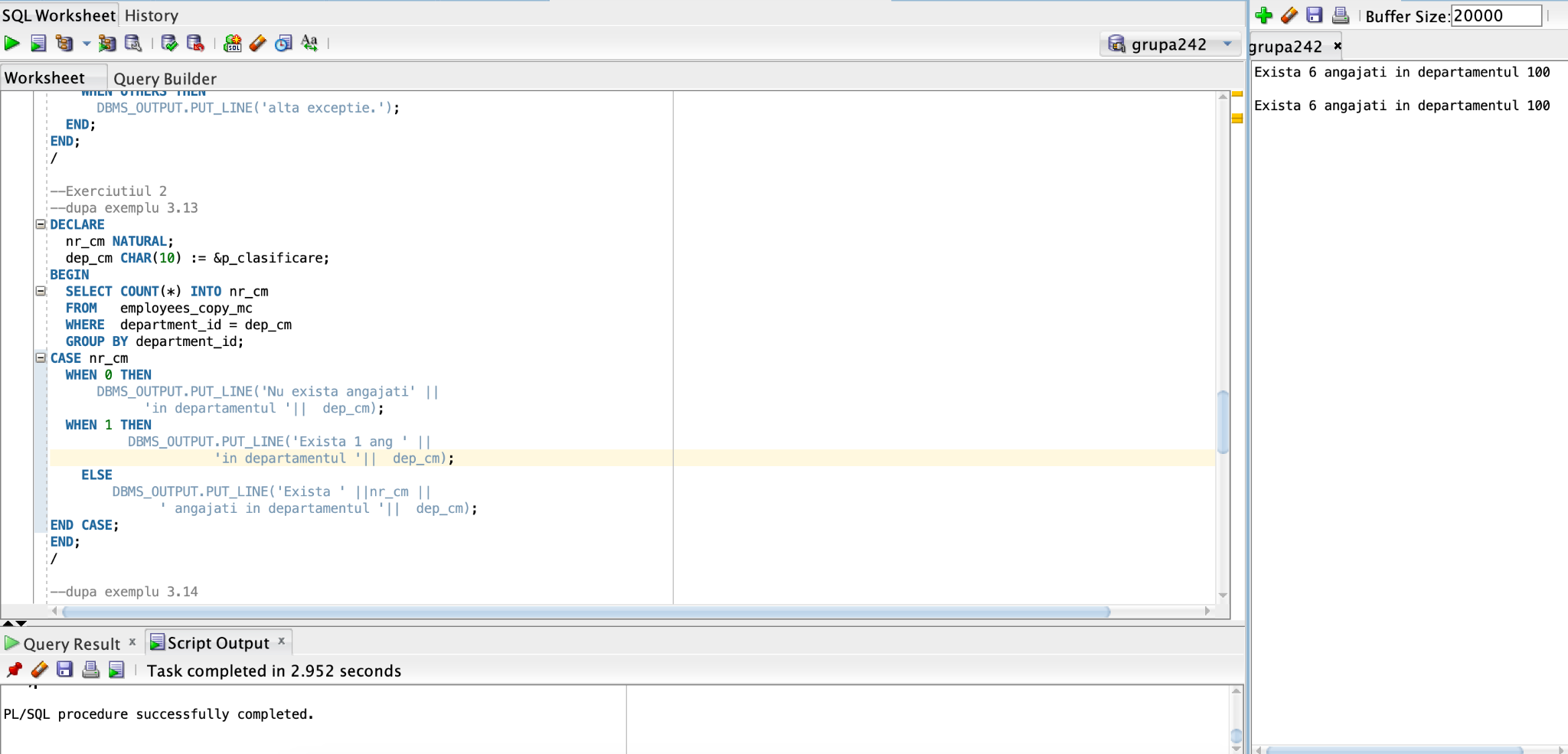
END;

/

Observație:

Dacă variabilele sunt inițializate în DECLARE, putem evita valorile neașteptate.

Exercitiul 2



Cod sursa:

--dupa exemplu 3.13

DECLARE

nr\_cm NATURAL;

dep\_cm CHAR(10) := &p\_clasificare;

BEGIN

SELECT COUNT(\*) INTO nr\_cm

FROM employees\_copy\_mc

WHERE department\_id = dep\_cm

GROUP BY department\_id;

CASE nr\_cm

WHEN 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Nu exista angajati' ||

'in departamentul '|| dep\_cm);

WHEN 1 THEN

DBMS\_OUTPUT.PUT\_LINE('Exista 1 ang ' ||

'in departamentul '|| dep\_cm);

ELSE

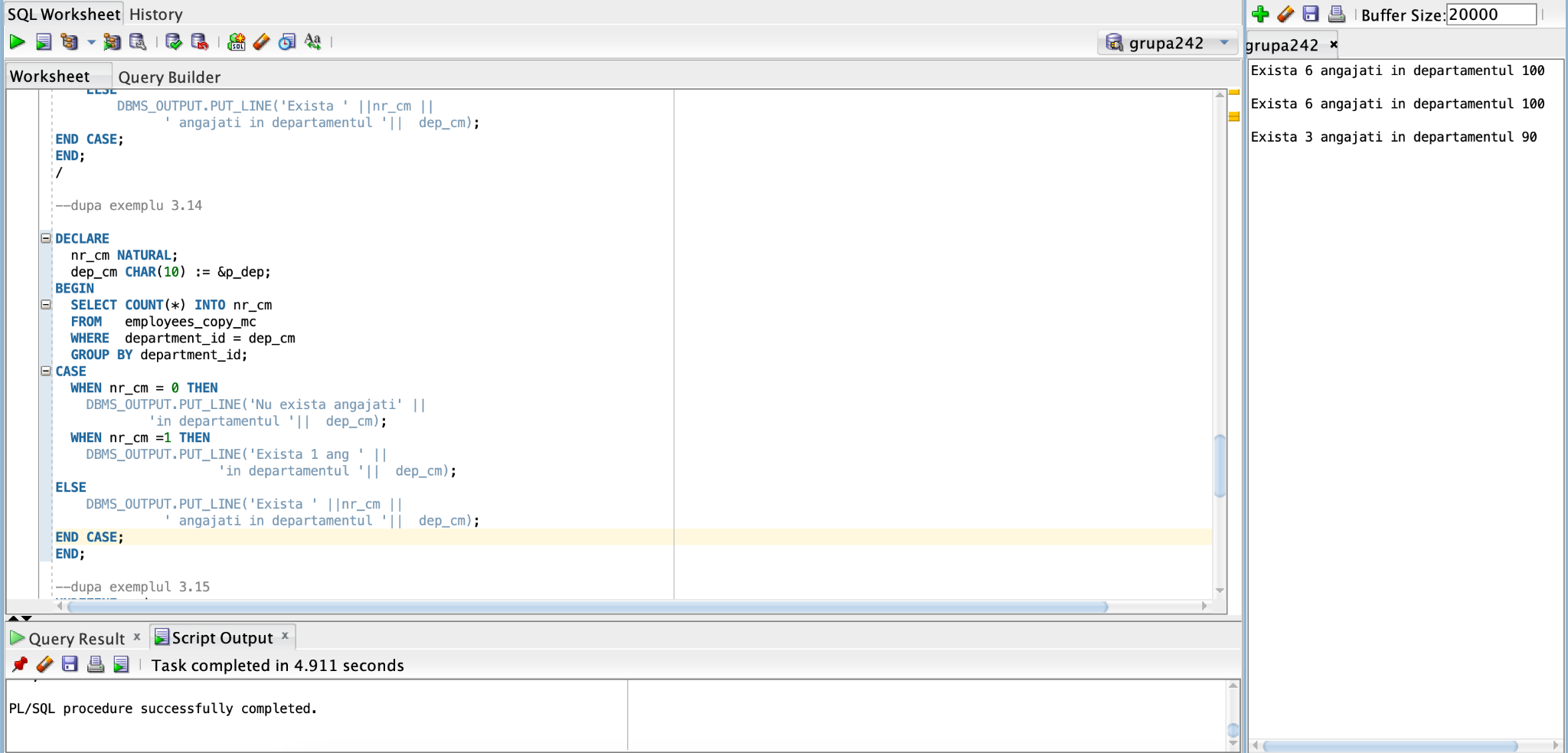
DBMS\_OUTPUT.PUT\_LINE('Exista ' ||nr\_cm ||

' angajati in departamentul '|| dep\_cm);

END CASE;

END;

/



Cod sursa:

--dupa exemplu 3.14

DECLARE

nr\_cm NATURAL;

dep\_cm CHAR(10) := &p\_dep;

BEGIN

SELECT COUNT(\*) INTO nr\_cm

FROM employees\_copy\_mc

WHERE department\_id = dep\_cm

GROUP BY department\_id;

CASE

WHEN nr\_cm = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Nu exista angajati' ||

'in departamentul '|| dep\_cm);

WHEN nr\_cm =1 THEN

DBMS\_OUTPUT.PUT\_LINE('Exista 1 ang ' ||

'in departamentul '|| dep\_cm);

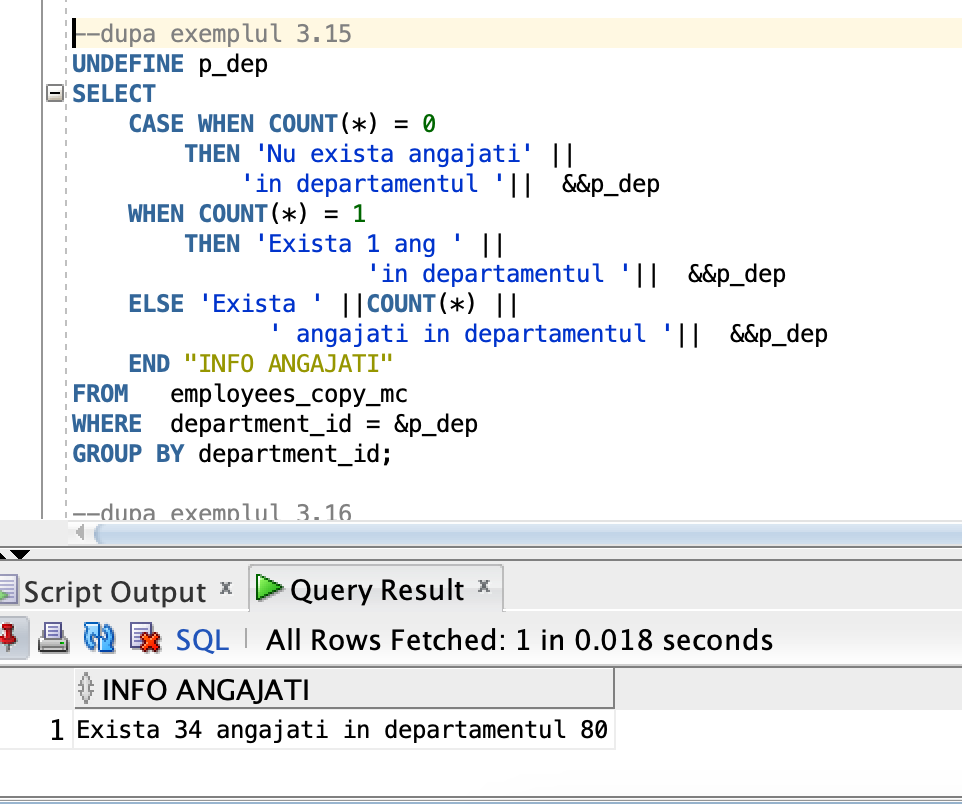
ELSE

DBMS\_OUTPUT.PUT\_LINE('Exista ' ||nr\_cm ||

' angajati in departamentul '|| dep\_cm);

END CASE;

END;



Cod sursa:

--dupa exemplul 3.15

UNDEFINE p\_dep

SELECT

CASE WHEN COUNT(\*) = 0

THEN 'Nu exista angajati' ||

'in departamentul '|| &&p\_dep

WHEN COUNT(\*) = 1

THEN 'Exista 1 ang ' ||

'in departamentul '|| &&p\_dep

ELSE 'Exista ' ||COUNT(\*) ||

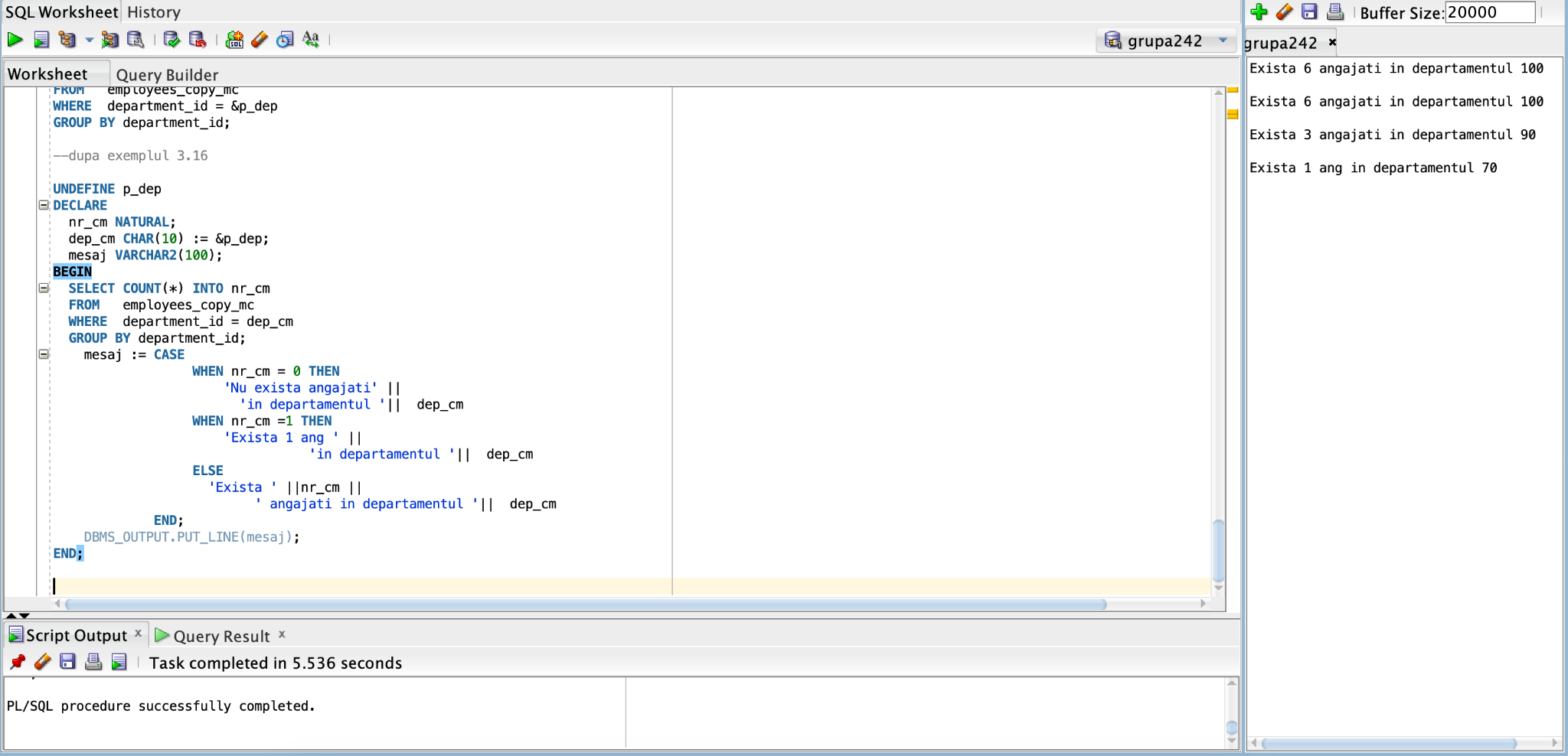
' angajati in departamentul '|| &&p\_dep

END "INFO ANGAJATI"

FROM employees\_copy\_mc

WHERE department\_id = &p\_dep

GROUP BY department\_id;



Cod sursa:

--dupa exemplul 3.16

UNDEFINE p\_dep

DECLARE

nr\_cm NATURAL;

dep\_cm CHAR(10) := &p\_dep;

mesaj VARCHAR2(100);

BEGIN

SELECT COUNT(\*) INTO nr\_cm

FROM employees\_copy\_mc

WHERE department\_id = dep\_cm

GROUP BY department\_id;

mesaj := CASE

WHEN nr\_cm = 0 THEN

'Nu exista angajati' ||

'in departamentul '|| dep\_cm

WHEN nr\_cm =1 THEN

'Exista 1 ang ' ||

'in departamentul '|| dep\_cm

ELSE

'Exista ' ||nr\_cm ||

' angajati in departamentul '|| dep\_cm

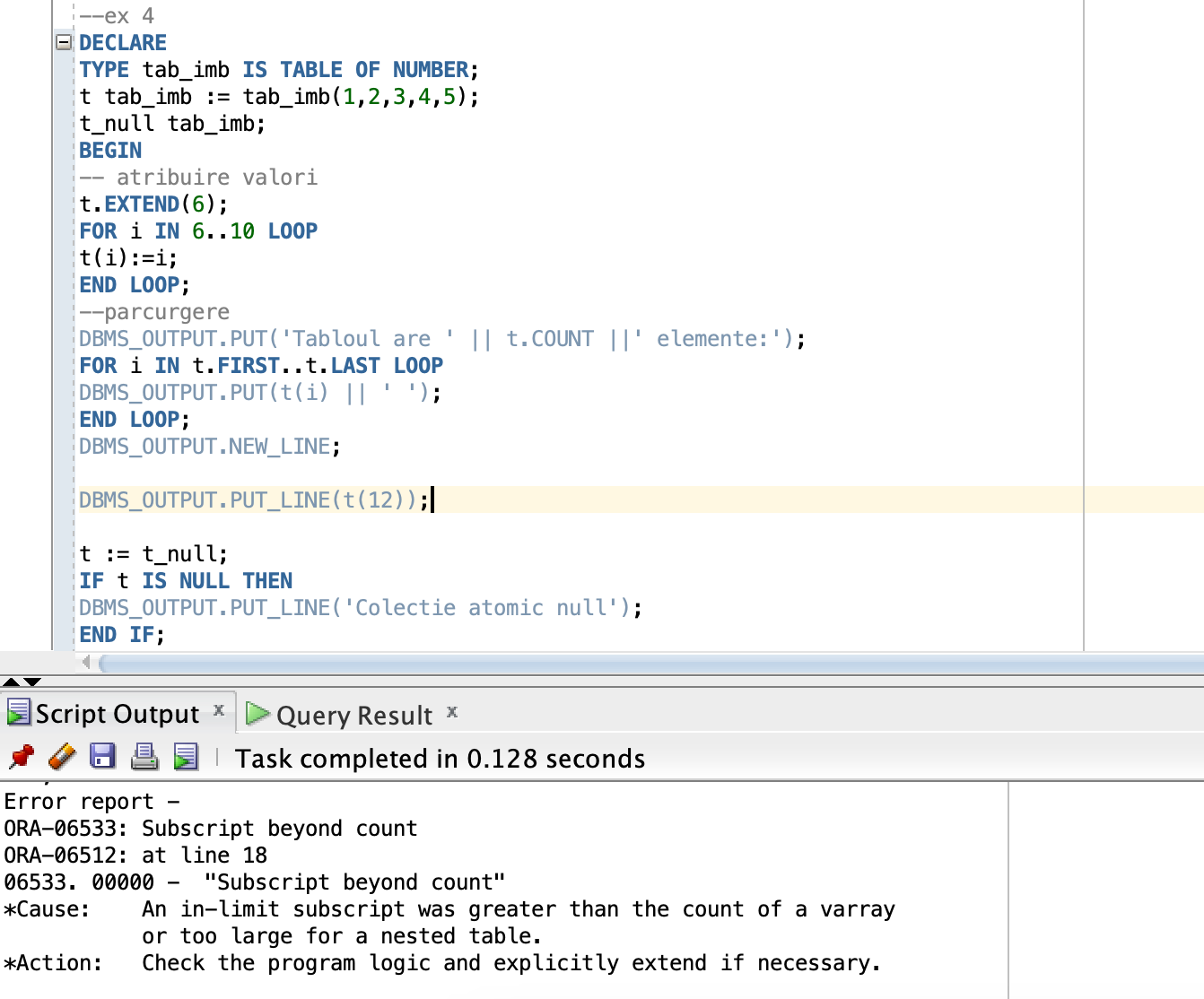
END;

DBMS\_OUTPUT.PUT\_LINE(mesaj);

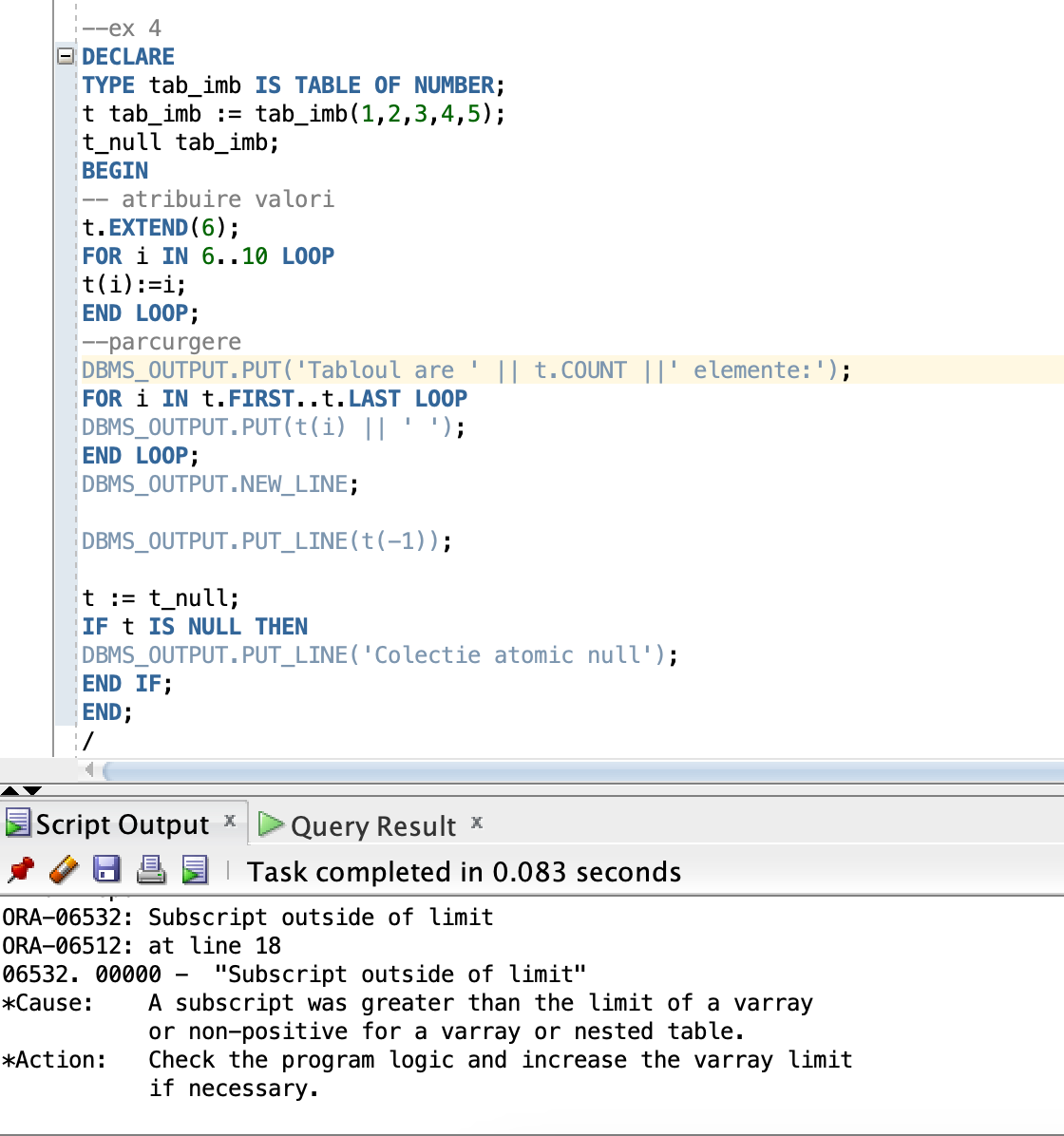
END;

Exercitiul 3:

* ORA-06533: SUBSCRIPT\_BEYOND\_COUNT este declanșată atunci când se încearcă accesarea unei poziții dincolo de limita tabelului (în acest caz, 11 sau 12).



* ORA-06532: SUBSCRIPT\_OUTSIDE\_LIMIT este declanșată atunci când se încearcă accesarea unei poziții în afara limitelor definite ale tabelului (de exemplu, -1 sau o valoare mai mare decât capacitatea declarată a tabelului).



* ORA-06531: COLLECTION\_IS\_NULL este declanșată atunci când se încearcă accesarea unei colecții care este declarată ca fiind null (nepopulată).

DECLARE

TYPE tab\_imb IS TABLE OF NUMBER;

t\_null tab\_imb;

BEGIN

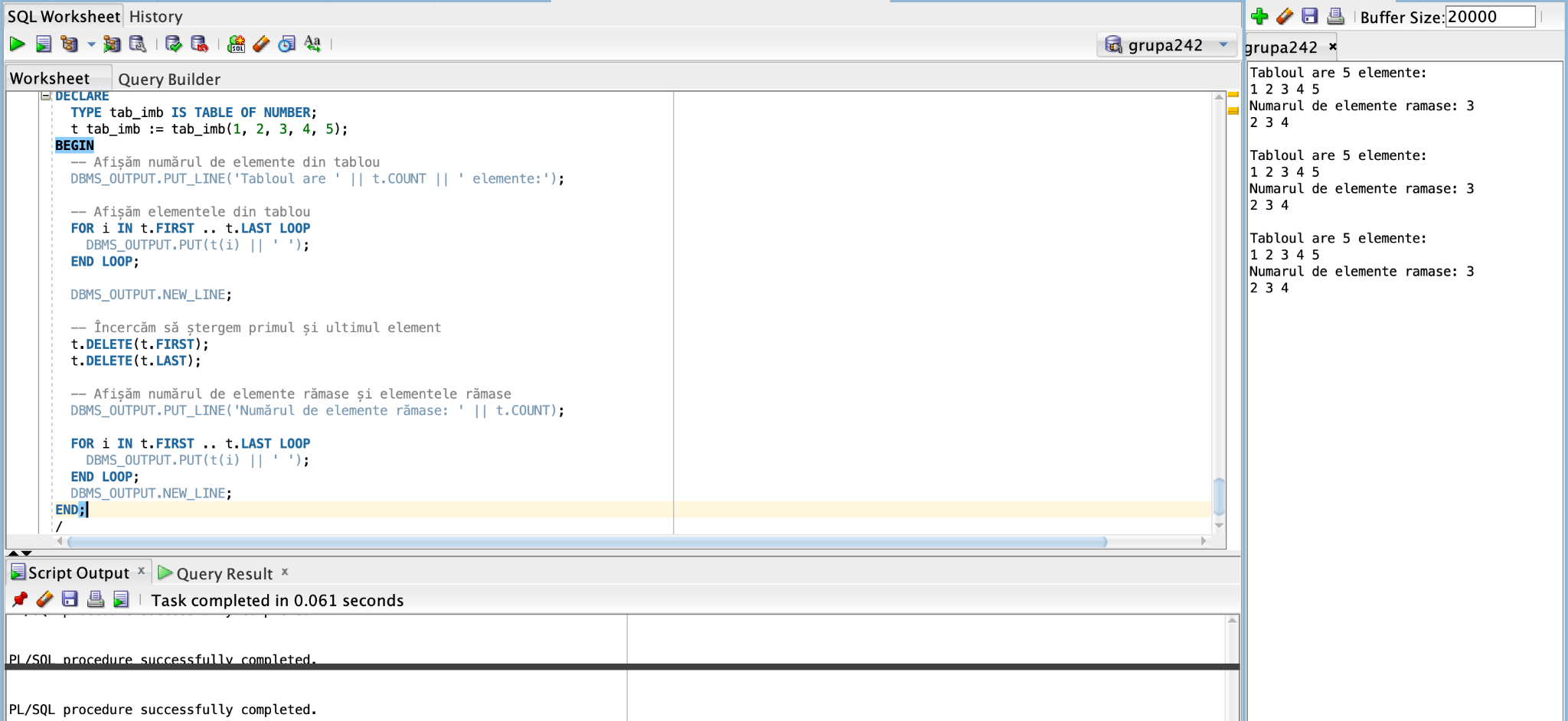
-- Încercam sa accesam o colectie null

t\_null(1) := 1; -- Acest lucru va genera excepția ORA-06531: COLLECTION\_IS\_NULL

END;

/

Exercitiul 4



Exceptii care pot aparea:

* NO\_DATA\_FOUND - Această excepție poate apărea atunci când încercați să ștergeți un element dintr-un tablou care este deja gol (adică nu există date de șters). De exemplu, dacă încercați să ștergeți primul element dintr-un tablou gol, va fi generată o excepție NO\_DATA\_FOUND.
* SUBSCRIPT\_OUTSIDE\_LIMIT - Această excepție poate apărea atunci când încercați să accesați o poziție în afara limitelor valide ale tabloului sau să ștergeți un element în afara limitelor valide ale tabloului. De exemplu, dacă încercați să ștergeți un element dintr-o poziție care nu există în tablou sau depășiți limita tabloului, va fi generată o excepție SUBSCRIPT\_OUTSIDE\_LIMIT.

Cod sursa:

DECLARE

TYPE tab\_imb IS TABLE OF NUMBER;

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

BEGIN

-- Afișăm numărul de elemente din tablou

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

-- Afișăm elementele din tablou

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

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

END LOOP;

DBMS\_OUTPUT.NEW\_LINE;

-- Încercăm să ștergem primul și ultimul element

t.DELETE(t.FIRST);

t.DELETE(t.LAST);

-- Afișăm numărul de elemente rămase și elementele rămase

DBMS\_OUTPUT.PUT\_LINE('Numărul de elemente rămase: ' || t.COUNT);

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

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

END LOOP;

DBMS\_OUTPUT.NEW\_LINE;

END;

/