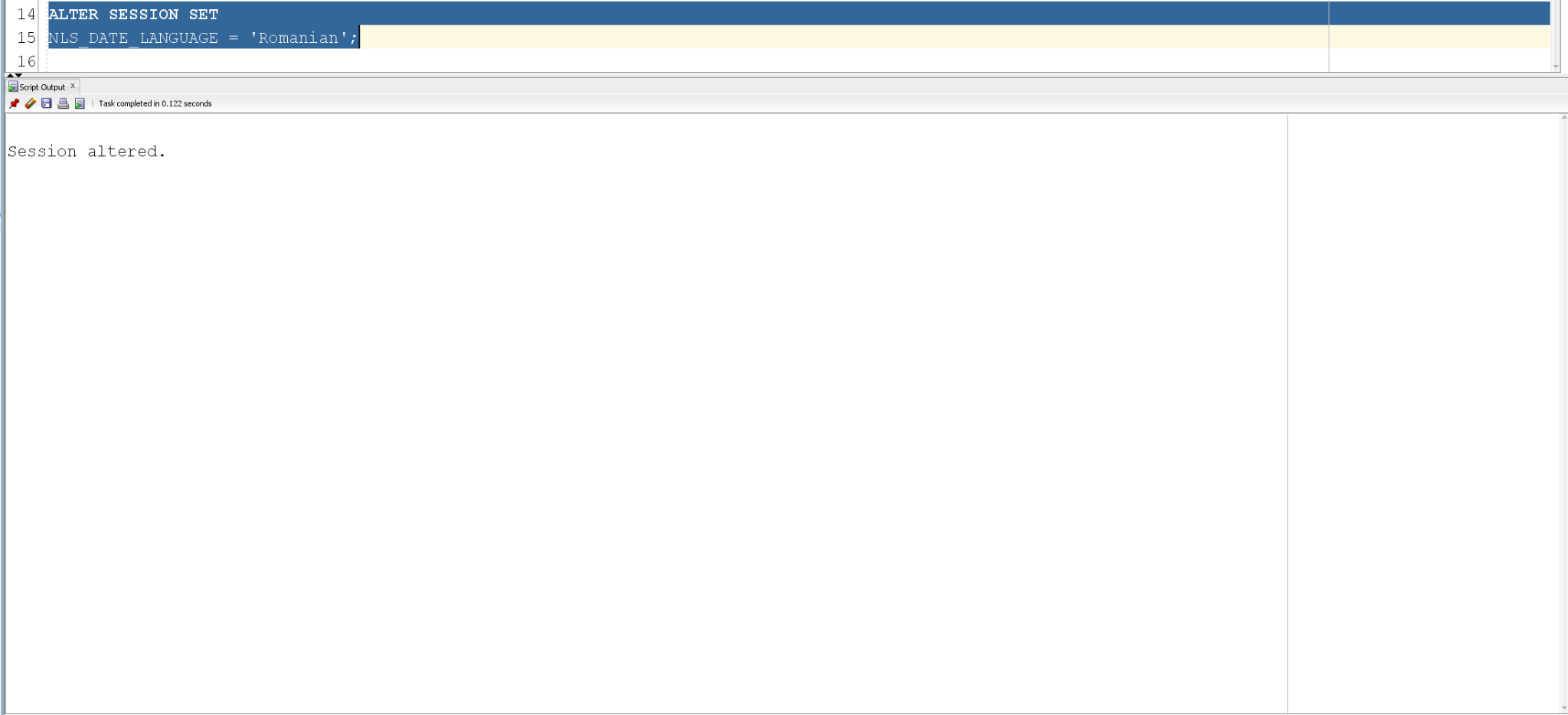
**Exemple de utilizare a tipurilor de date char cu stocare de diacritice si caractere speciale**

Am setat mai intai pentru sesiunea curenta limba romana folosind:  
  
ALTER SESSION SET

NLS\_DATE\_LANGUAGE = 'Romanian';



Din pacate se pare ca nu recunoaste caracterele ‘ățș’. Lui ‘ă’ ii asigneaza ‘a’, iar lui ‘ț’ si ‘ș’ le asigneaza ‘?’. In schimb caracterele ‘â’ si ‘î’ sunt recunoscute.

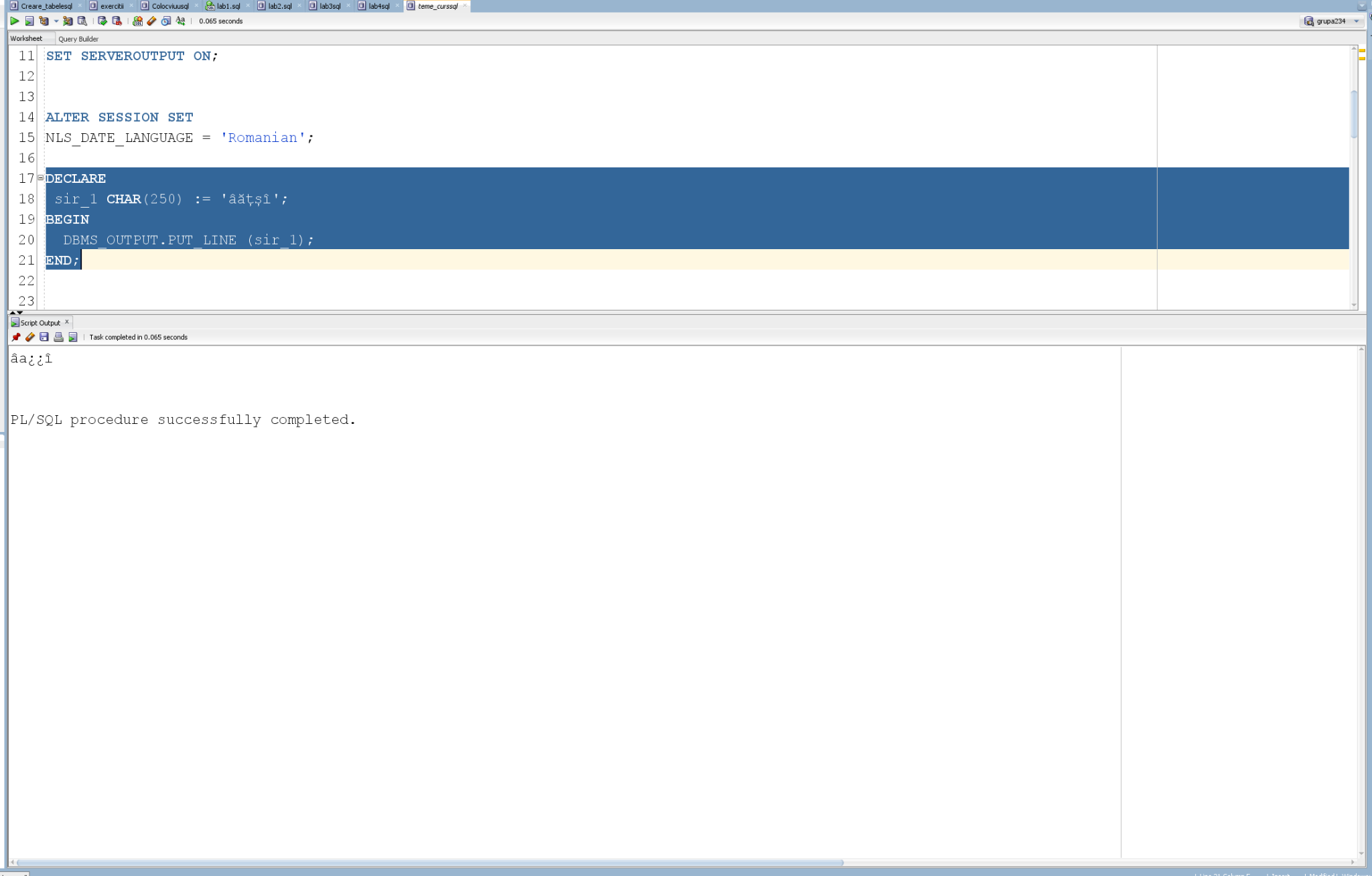
DECLARE

sir\_1 CHAR(250) := 'âățșî';

BEGIN

DBMS\_OUTPUT.PUT\_LINE (sir\_1);

END;



Am luat ca referinta sirul “âățșî`\>/?@^ egrtfhefaa” care are 25 de caractere ce cuprinde cele 5 diacritice, caractere speciale, spatii si litere mici din alfabetul englez. Sa testam acum diferentele dintre cele 4 metode de declarare pentru siruri de caractere:

DECLARE

sir\_1 CHAR(25) := 'âățșî`\>/?@^ egrtfhefaa'; --25 de caractere

sir\_2 VARCHAR2(25) := 'âățșî`\>/?@^ egrtfhefaa';

sir\_3 NCHAR(25) := 'âățșî`\>/?@^ egrtfhefaa';

sir\_4 NVARCHAR2(25) := 'âățșî`\>/?@^ egrtfhefaa';

BEGIN

DBMS\_OUTPUT.PUT\_LINE (sir\_1);

DBMS\_OUTPUT.PUT\_LINE (sir\_2);

DBMS\_OUTPUT.PUT\_LINE (sir\_3);

DBMS\_OUTPUT.PUT\_LINE (sir\_4);

IF sir\_1 = sir\_2 THEN

DBMS\_OUTPUT.PUT\_LINE ('sir\_1 = sir\_2');

ELSE

DBMS\_OUTPUT.PUT\_LINE ('sir\_1 != sir\_2' );

END IF;

IF sir\_1 = sir\_3 THEN

DBMS\_OUTPUT.PUT\_LINE ('sir\_1 = sir\_3');

ELSE

DBMS\_OUTPUT.PUT\_LINE ('sir\_1 != sir\_3' );

END IF;

IF sir\_1 = sir\_4 THEN

DBMS\_OUTPUT.PUT\_LINE ('sir\_1 = sir\_4');

ELSE

DBMS\_OUTPUT.PUT\_LINE ('sir\_1 != sir\_4' );

END IF;

IF sir\_2 = sir\_4 THEN

DBMS\_OUTPUT.PUT\_LINE ('sir\_2 = sir\_4');

ELSE

DBMS\_OUTPUT.PUT\_LINE ('sir\_2 != sir\_4' );

END IF;

IF sir\_3 = sir\_4 THEN

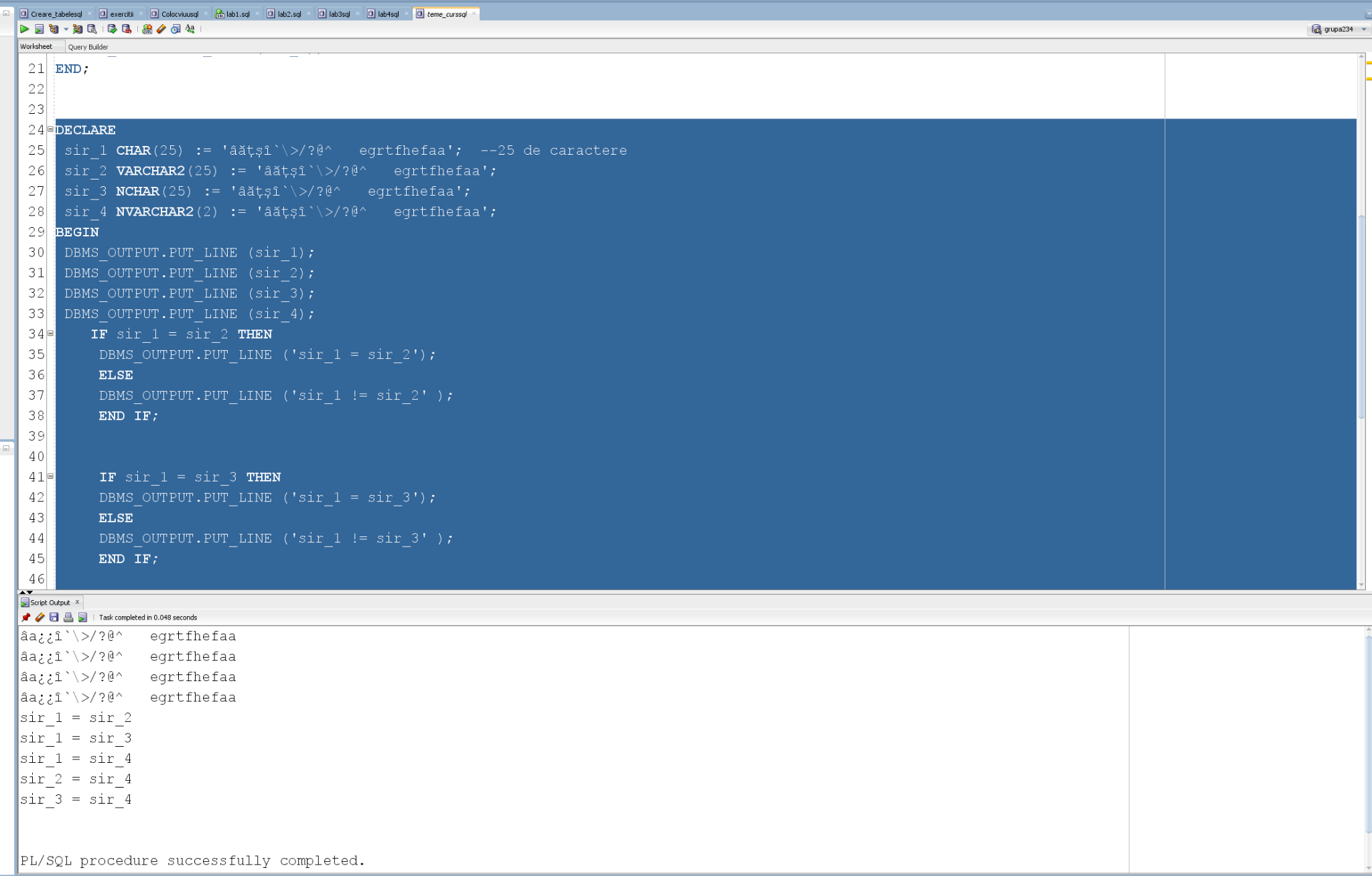
DBMS\_OUTPUT.PUT\_LINE ('sir\_3 = sir\_4');

ELSE

DBMS\_OUTPUT.PUT\_LINE ('sir\_3 != sir\_4' );

END IF;

END;



Din cate se poate vedea, fiecare tip de declarare recunoaste exact la fel sirul nostru declarat, si toate cele 4 siruri vor fi egale daca au aceeasi lungime setata la limita. Dar daca dam spatiu mare pentru fiecare din ele?  
  
DECLARE

sir\_1 CHAR(250) := 'âățșî`\>/?@^ egrtfhefaa'; --25 de caractere

sir\_2 VARCHAR2(250) := 'âățșî`\>/?@^ egrtfhefaa';

sir\_3 NCHAR(250) := 'âățșî`\>/?@^ egrtfhefaa';

sir\_4 NVARCHAR2(250) := 'âățșî`\>/?@^ egrtfhefaa';

BEGIN

DBMS\_OUTPUT.PUT\_LINE (sir\_1);

DBMS\_OUTPUT.PUT\_LINE (sir\_2);

DBMS\_OUTPUT.PUT\_LINE (sir\_3);

DBMS\_OUTPUT.PUT\_LINE (sir\_4);

IF sir\_1 = sir\_2 THEN

DBMS\_OUTPUT.PUT\_LINE ('sir\_1 = sir\_2');

ELSE

DBMS\_OUTPUT.PUT\_LINE ('sir\_1 != sir\_2' );

END IF;

IF sir\_1 = sir\_3 THEN

DBMS\_OUTPUT.PUT\_LINE ('sir\_1 = sir\_3');

ELSE

DBMS\_OUTPUT.PUT\_LINE ('sir\_1 != sir\_3' );

END IF;

IF sir\_1 = sir\_4 THEN

DBMS\_OUTPUT.PUT\_LINE ('sir\_1 = sir\_4');

ELSE

DBMS\_OUTPUT.PUT\_LINE ('sir\_1 != sir\_4' );

END IF;

IF sir\_2 = sir\_4 THEN

DBMS\_OUTPUT.PUT\_LINE ('sir\_2 = sir\_4');

ELSE

DBMS\_OUTPUT.PUT\_LINE ('sir\_2 != sir\_4' );

END IF;

IF sir\_3 = sir\_4 THEN

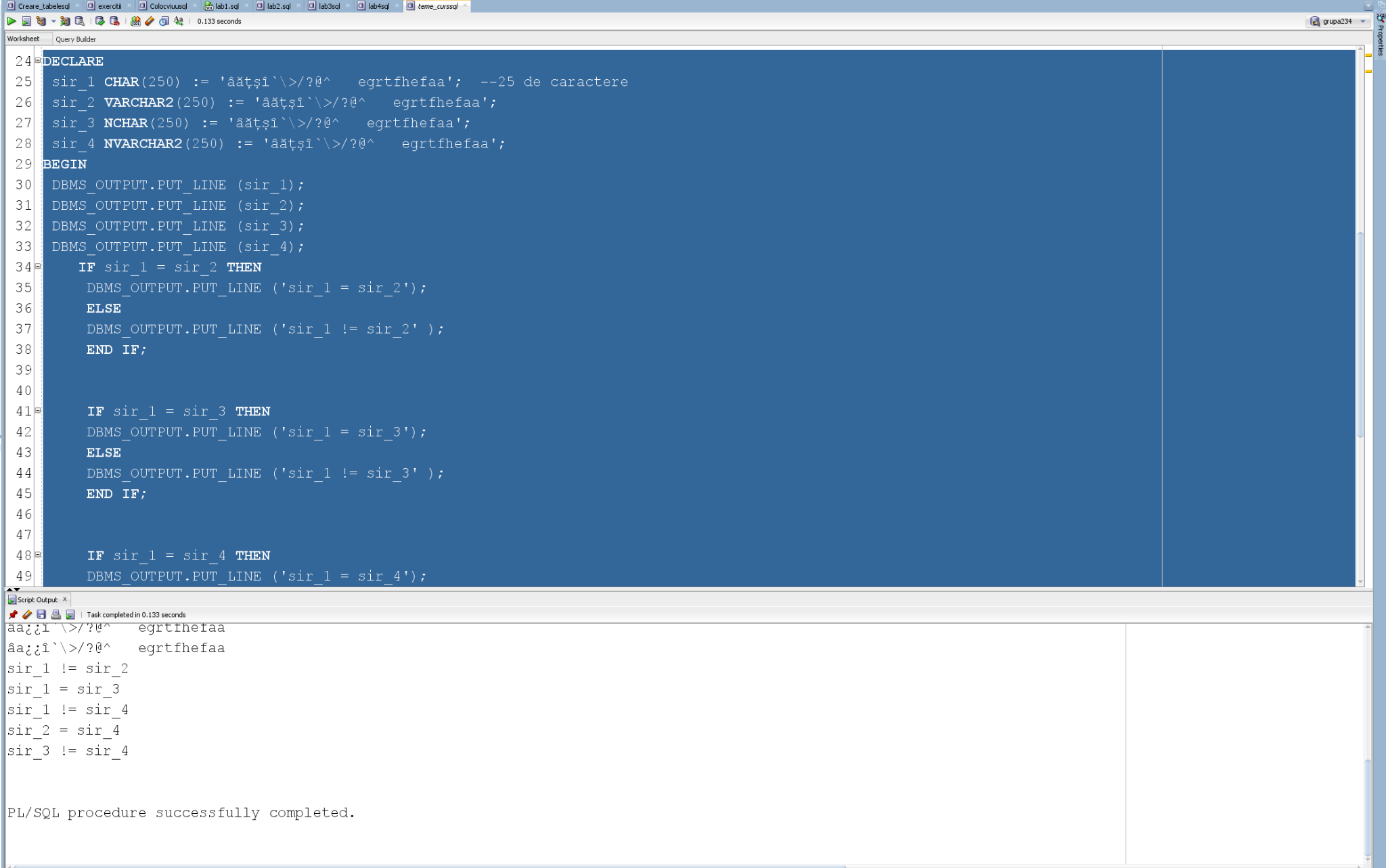
DBMS\_OUTPUT.PUT\_LINE ('sir\_3 = sir\_4');

ELSE

DBMS\_OUTPUT.PUT\_LINE ('sir\_3 != sir\_4' );

END IF;

END;



Acum se paote observa clar ca sirurile declarate char si nchar vor lua multe caracte ‘empty’ ca sa poate sa umple tot spatiul alocat.

Dar daca toate lungimile sunt diferite?

DECLARE

sir\_1 CHAR(250) := 'âățșî`\>/?@^ egrtfhefaa'; --25 de caractere

sir\_2 VARCHAR2(200) := 'âățșî`\>/?@^ egrtfhefaa';

sir\_3 NCHAR(50) := 'âățșî`\>/?@^ egrtfhefaa';

sir\_4 NVARCHAR2(25) := 'âățșî`\>/?@^ egrtfhefaa';

BEGIN

DBMS\_OUTPUT.PUT\_LINE (sir\_1);

DBMS\_OUTPUT.PUT\_LINE (sir\_2);

DBMS\_OUTPUT.PUT\_LINE (sir\_3);

DBMS\_OUTPUT.PUT\_LINE (sir\_4);

IF sir\_1 = sir\_2 THEN

DBMS\_OUTPUT.PUT\_LINE ('sir\_1 = sir\_2');

ELSE

DBMS\_OUTPUT.PUT\_LINE ('sir\_1 != sir\_2' );

END IF;

IF sir\_1 = sir\_3 THEN

DBMS\_OUTPUT.PUT\_LINE ('sir\_1 = sir\_3');

ELSE

DBMS\_OUTPUT.PUT\_LINE ('sir\_1 != sir\_3' );

END IF;

IF sir\_1 = sir\_4 THEN

DBMS\_OUTPUT.PUT\_LINE ('sir\_1 = sir\_4');

ELSE

DBMS\_OUTPUT.PUT\_LINE ('sir\_1 != sir\_4' );

END IF;

IF sir\_2 = sir\_4 THEN

DBMS\_OUTPUT.PUT\_LINE ('sir\_2 = sir\_4');

ELSE

DBMS\_OUTPUT.PUT\_LINE ('sir\_2 != sir\_4' );

END IF;

IF sir\_3 = sir\_4 THEN

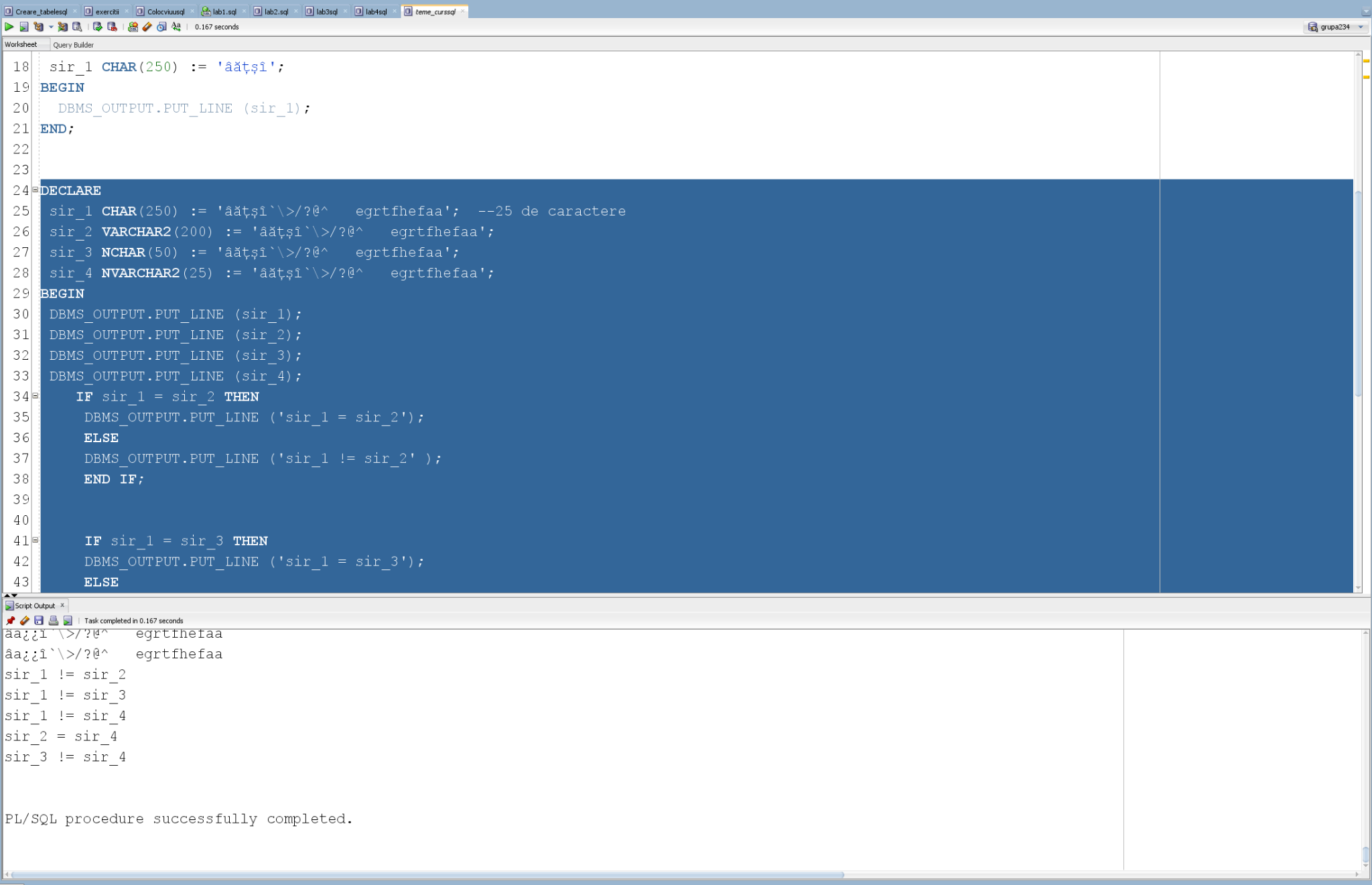
DBMS\_OUTPUT.PUT\_LINE ('sir\_3 = sir\_4');

ELSE

DBMS\_OUTPUT.PUT\_LINE ('sir\_3 != sir\_4' );

END IF;

END;



Acum se poate observa ca si sirul 1 este diferit de sirul 3 deoarce sirul 3 umple cu mult mai putine caractere ‘empty’ decat sirul1. Sirurile 2 si 4 au ramas in continuare egale deoarece acestea aloca exact cat spatiu e nevoie pentru sirul dat.

Toate sirurile dau eroare cand dimenisunea declarata e < lungimea sirului:

DECLARE

sir\_1 CHAR(24) := 'âățșî`\>/?@^ egrtfhefaa';

BEGIN

DBMS\_OUTPUT.PUT\_LINE (sir\_1);

END;



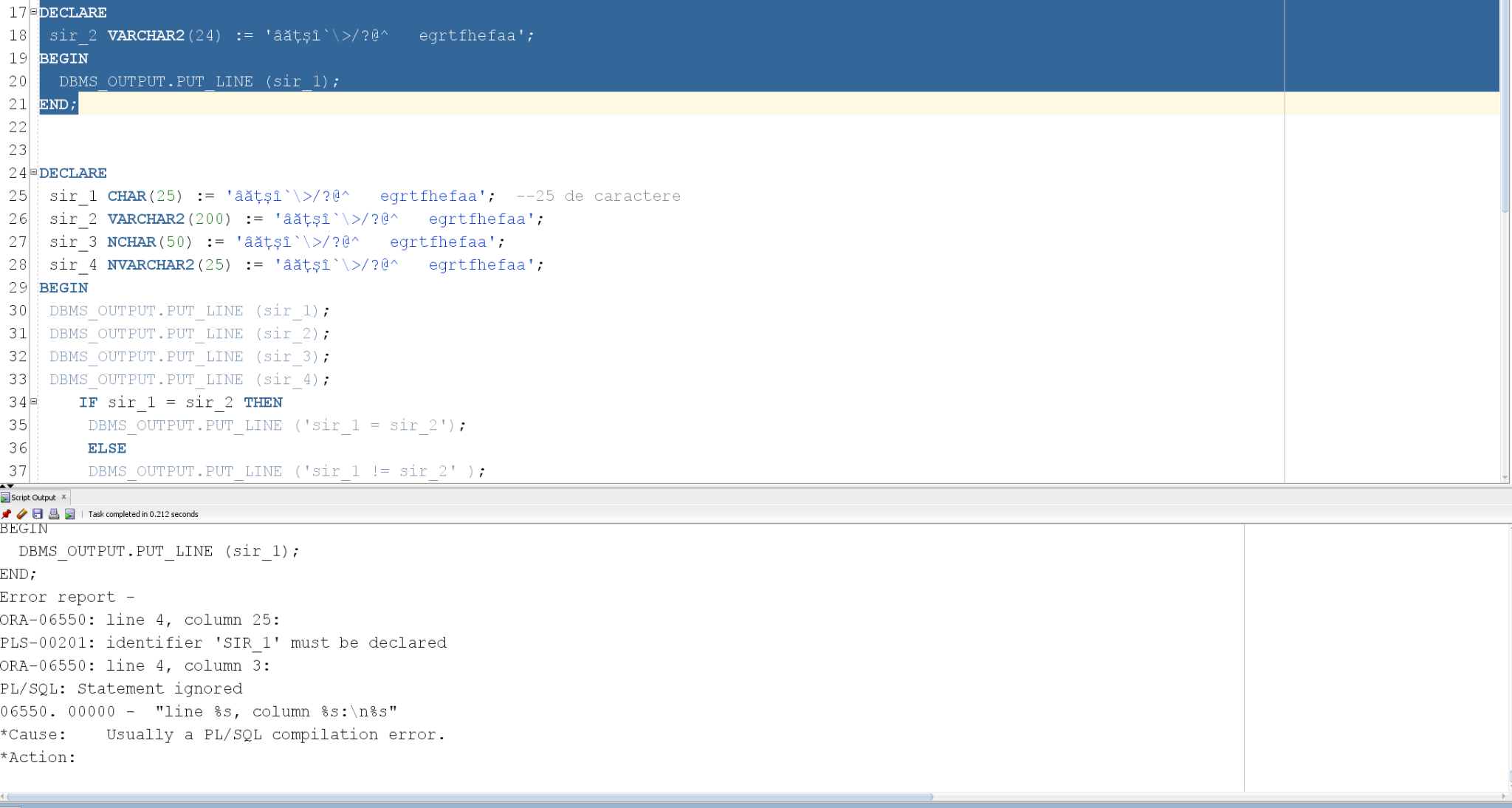
DECLARE

sir\_2 VARCHAR2(24) := 'âățșî`\>/?@^ egrtfhefaa';

BEGIN

DBMS\_OUTPUT.PUT\_LINE (sir\_2);

END;



DECLARE

sir\_3 NCHAR(24) := 'âățșî`\>/?@^ egrtfhefaa';

BEGIN

DBMS\_OUTPUT.PUT\_LINE (sir\_3);

END;



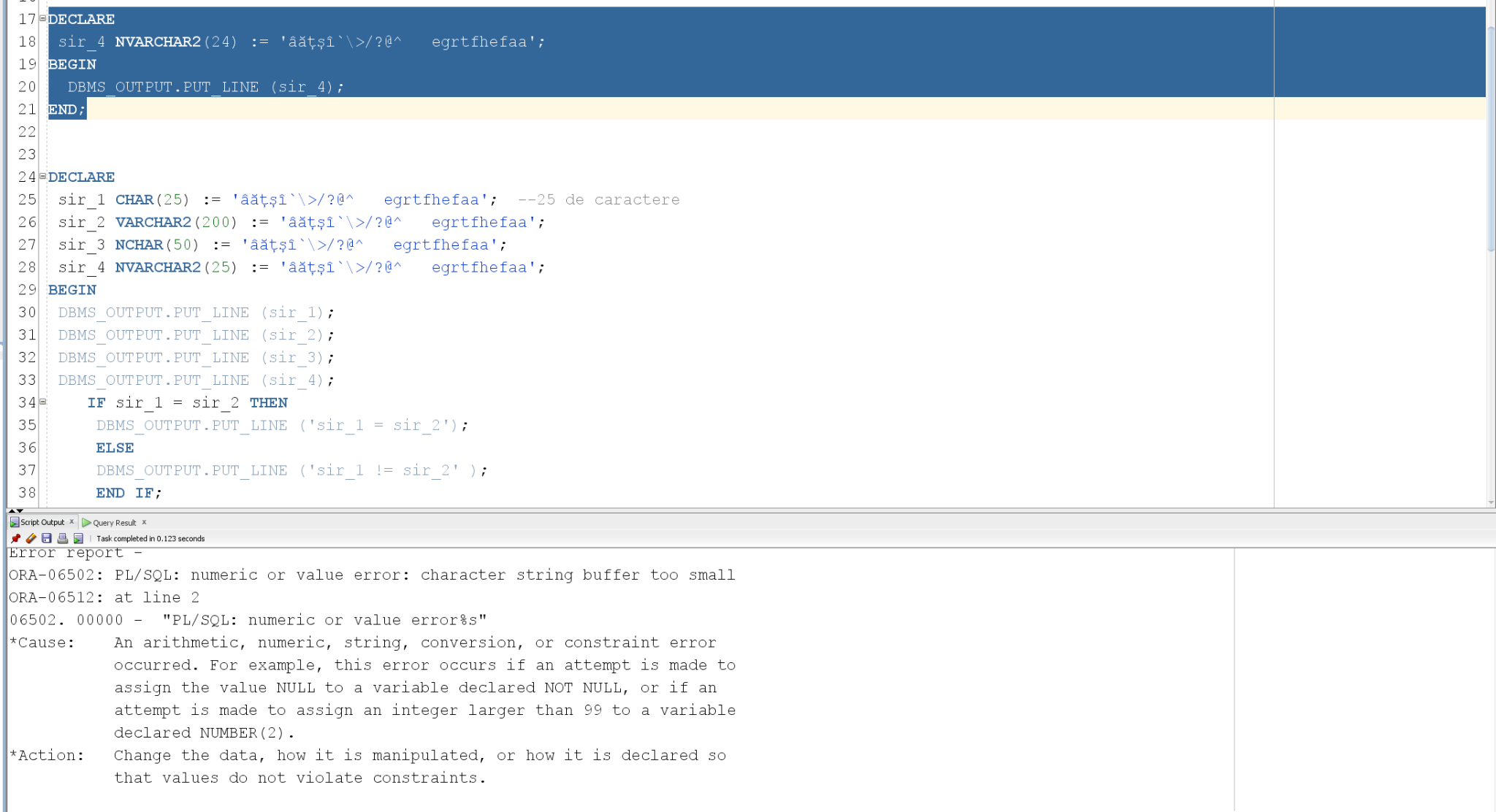
DECLARE

sir\_4 NVARCHAR2(24) := 'âățșî`\>/?@^ egrtfhefaa';

BEGIN

DBMS\_OUTPUT.PUT\_LINE (sir\_4);

END;



**CONCLUZIE:**

Asadar, se pare ca toate tipurile de declarare pentru siruri de caractere suporta la fel diacriticele, si se comporta similar si pentru caractere speciale cat si pentru cele obisnuite. Ar trebui utilizate CHAR si NCHAR doar cand stim dimensiunea fixa a sirului nostru, deoarce pot exista caractere invizibile ce ne pot incurca. VARCHAR si NVARCHAR sunt cele pe care le vom dori sa le utilizam cand nu stim dimensiunea sirului sau cand dimensiunea acestuia e variabila.