**Teme cursul 4**

**Tema 8** – De gestionat exceptiile too\_many\_rows si no\_data\_found pe un exemplu pe tabela employees/departments cu delete si update.

Pentru rezolvarea acestui exercitiu am folosit copia mea a tabelei employees.

DECLARE

v\_salariu NUMBER;

v\_id\_departament NUMBER;

v\_email CHAR(25);

v\_cod emp\_pam.employee\_id%TYPE := &cod;

BEGIN

DELETE FROM emp\_pam WHERE employee\_id=v\_cod

RETURNING salary, department\_id, email

INTO v\_salariu, v\_id\_departament, v\_email;

IF SQL%NOTFOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nu exista niciun angajat cu acest id!');

ELSE

INSERT INTO emp\_pam

VALUES (1, 'Ana', 'Panait', 'email@yahoo.com', '0712312312', '10-JUN-22', 'id\_job', 100000, null, null, 90);

UPDATE emp\_pam

SET email = v\_email || '\_updated'

WHERE employee\_id = v\_cod;

COMMIT;

END IF;

EXCEPTION

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('Doi sau mai multi angajati au acelasi id, id-ul trebuie sa fie unic!');

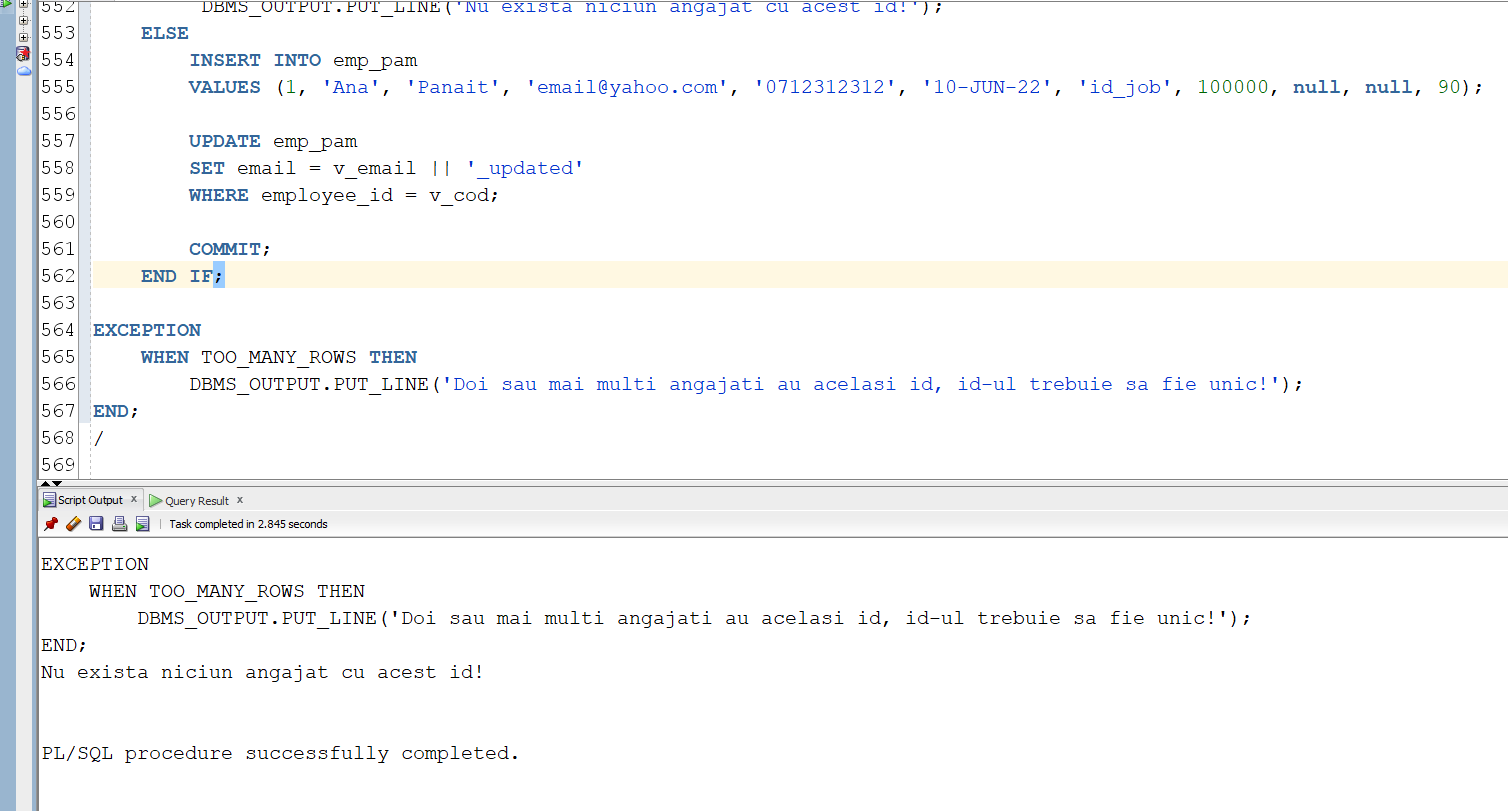
END;

/

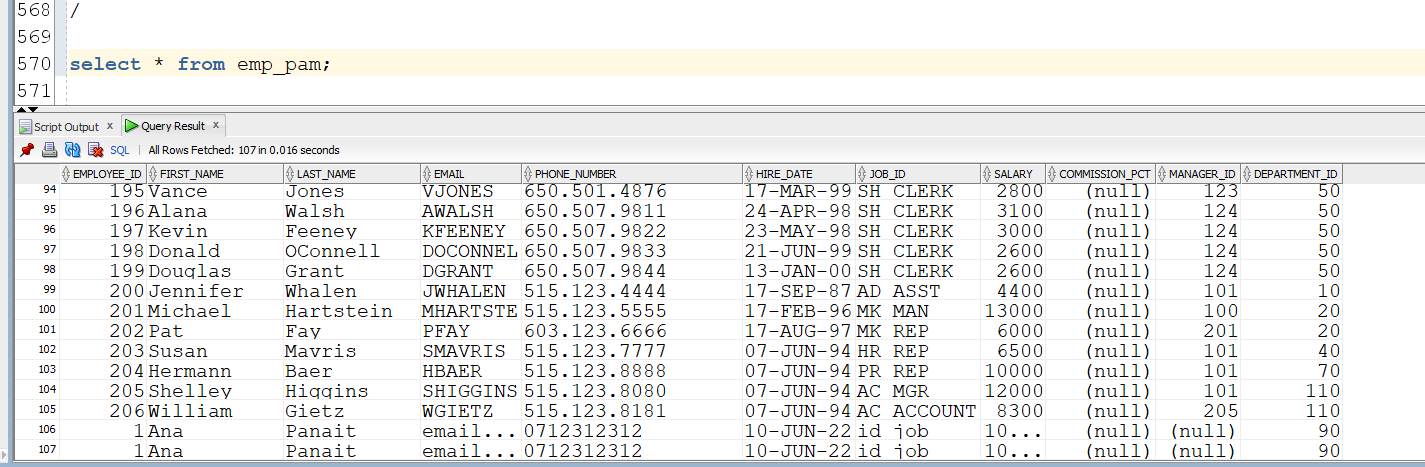
Se poate observa ca eroarea NO\_DATA\_FOUND nu a fost scrisa in blocul EXCEPTION pentru ca operatia de delete nu va arunca aceasta eroare in cazul in care nu gaseste niciun angajat cu id-ul citit de la tastatura, dar putem verifica ca nu a fost gasit si sa tratam acest caz folosind expresia SQL%NOTFOUND.

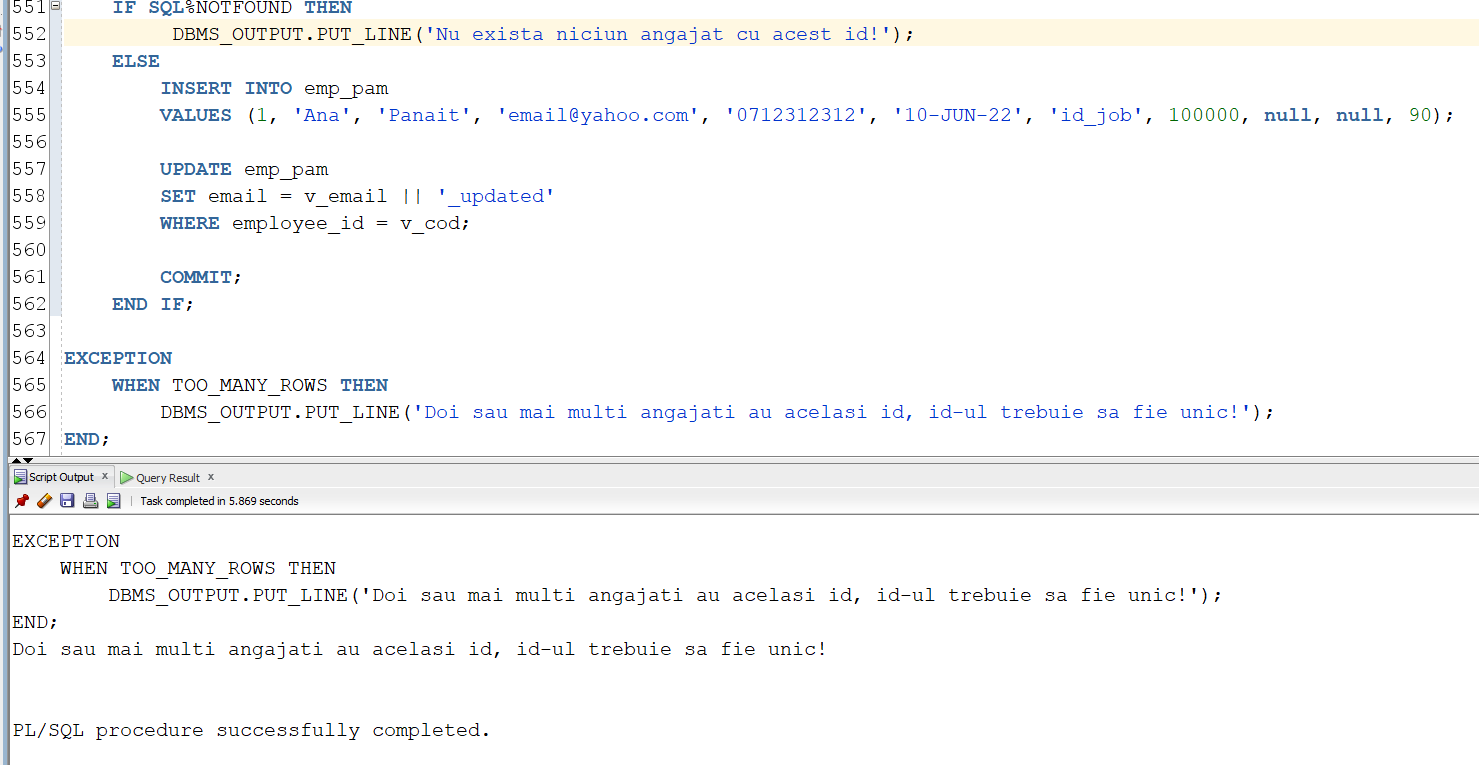
Pentru a verifica ca sunt tratate cele doua erori:

1. Am introdus de la tastatura un id inexistent (id = 3)



Se poate observa ca un mesaj corespunzator a fost afisat pentru ca operatia de delete nu a facut nimic.

1. Am rulat de doua ori codul pentru un id de angajat existent ca sa imi insereze angajatul cu id-ul 1 de doua ori.

Acum la rularea codului introducand codul 1 o sa dea eroarea too\_many\_rows la update.

**Tema 9** – Cum putem decide daca eroarea no\_data\_found a fost generate de select into sau de la accesarea unui element dintr-un tablou indexat care nu a fost atribuit/sters.

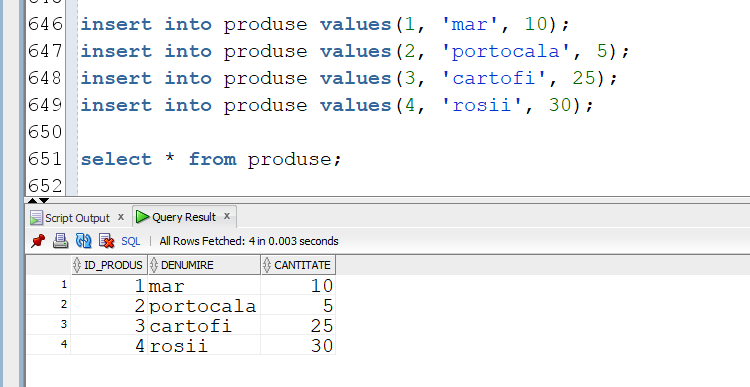
Pentru acest exercitiu am creat un tabel de test cu urmatoarea definitie si am introdus 4 inregistrari in el:

create table produse(

id\_produs INTEGER PRIMARY KEY,

denumire VARCHAR2(25),

cantitate INTEGER

);

DECLARE

TYPE prod\_table\_type IS TABLE OF produse.denumire%TYPE INDEX BY PLS\_INTEGER;

t prod\_table\_type;

nume VARCHAR2(20);

BEGIN

FOR i IN 1..5 LOOP

SELECT denumire

INTO t(i)

FROM produse

WHERE id\_produs = i;

END LOOP;

t.DELETE(t.first);

DBMS\_OUTPUT.PUT(t(0));

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

IF SQL%NOTFOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Eroare cauzata de SELECT INTO');

ELSE

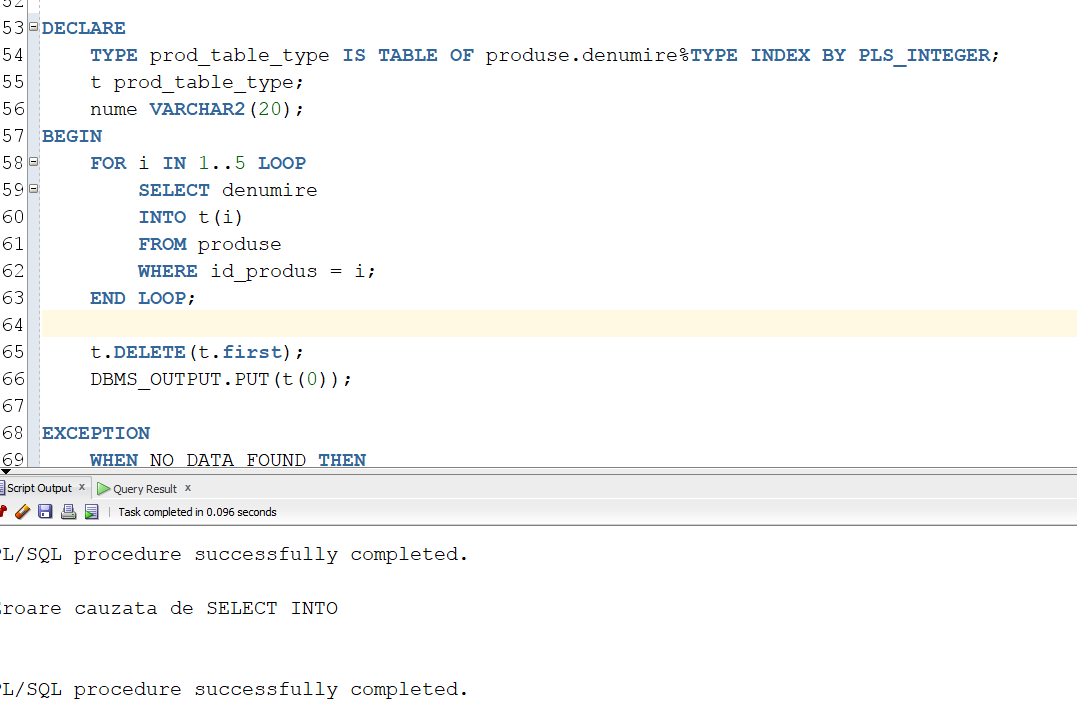
DBMS\_OUTPUT.PUT\_LINE('Eroare cauzata de accesarea unui element neatribuit din tabelul indexat');

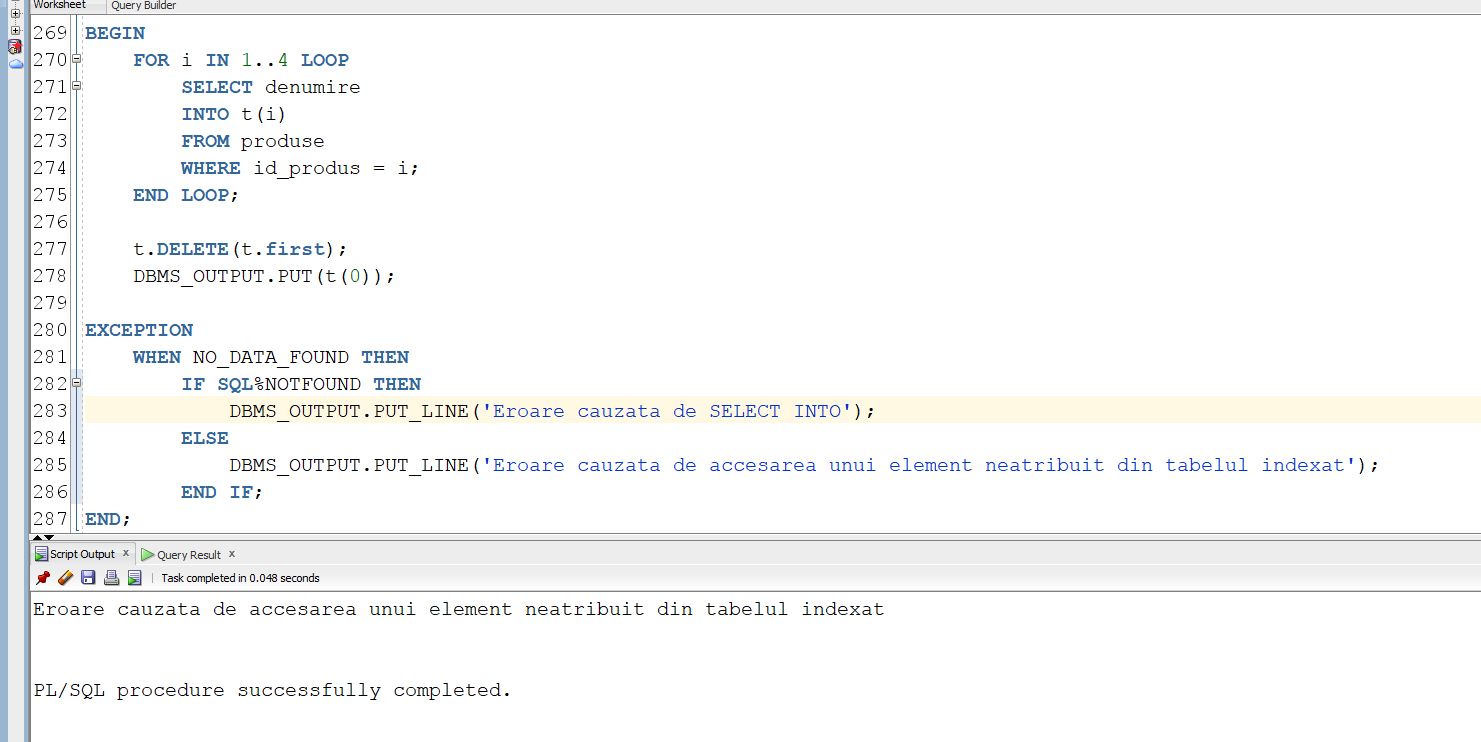
END IF;

END;

/

Pentru a testa daca eroarea este cauzata de select into sau de accesarea unui element neatribuit am folosit o conditie in blocul de exceptie (IF SQL%NOTFOUND). Putem testa ca functioneaza in felul urmator:

1. Rulez codul nemodificat (ar trebui sa dea eroare si la select into si la afisarea elementului sters daar select into fiind prima eroare aparuta doar aceea va fi vizibila). Ne asteptam sa afiseze textul “Eroare cauzata de SELECT INTO” ceea ce se si intampla.
2. Modific i-ul din for sa fie de la 1 la 4 si astfel nu va mai produca eroare la select into pentru ca avem inregistrarile in tabelul produse cu id-urile de la 1 la 4. Ne asteptam sa se afisele textul “Eroare cauzata de accesarea unui element neatribuit din tabelul indexat” ceea ce se si intampla.



**Tema 10** – De exemplificat utilizarea tablourilor imbricate cu ajutorul constructorului.

CREATE OR REPLACE TYPE tip\_persoana AS OBJECT (

id\_persoana NUMBER,

nume VARCHAR2(30),

telefon VARCHAR2(20)

);

/

CREATE OR REPLACE TYPE tip\_persoane AS TABLE OF tip\_persoana;

/

CREATE TABLE persoane (

numar\_grupa NUMBER,

col\_persoane tip\_persoane )

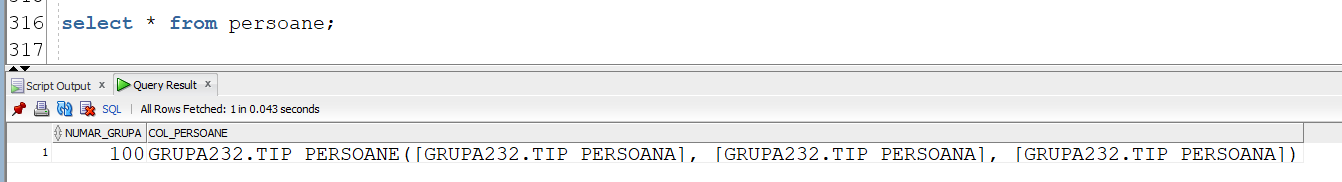
NESTED TABLE col\_persoane STORE AS col\_persoane\_nt;

/

INSERT INTO persoane VALUES (100, tip\_persoane(tip\_persoana(1, 'Panait Ana', '0712312312'),

tip\_persoana(2, 'Bob Alex', NULL),

tip\_persoana(3, 'Popescu Ion', '3212324343')

 ));

Bibliografie:

https://docs.oracle.com/cd/E11882\_01/appdev.112/e11822/adobjcol.htm#ADOBJ7245