Exercitiul E4 (PARTIAL, doar exemplul 10) din fisierul Laborator2\_PLSQL.pdf si exercitiul E2.

E2

**Definiți un tip colecție denumit tip\_orase\_\*\*\*. Creați tabelul excursie\_\*\*\* cu următoarea structură:**

**cod\_excursie NUMBER(4), denumire VARCHAR2(20), orase tip\_orase\_\*\*\* (ce va conține lista**

**orașelor care se vizitează într-o excursie, într-o ordine stabilită; de exemplu, primul oraș din listă va fi**

**primul oraș vizitat), status (disponibilă sau anulată).**

1. **Inserați 5 înregistrări în tabel.**

--facut la laborator

**b. Actualizați coloana orase pentru o excursie specificată:**

**- adăugați un oraș nou în listă, ce va fi ultimul vizitat în excursia respectivă;**

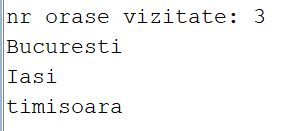
**- adăugați un oraș nou în listă, ce va fi al doilea oraș vizitat în excursia respectivă;**

**- inversați ordinea de vizitare a două dintre orașe al căror nume este specificat;**

**- eliminați din listă un oraș al cărui nume este specificat.**

--facut la laborator

**c. Pentru o excursie al cărui cod este dat, afișați numărul de orașe vizitate, respectiv numele orașelor.**



DECLARE

v\_cod NUMBER:='&p\_cod';

v\_lista tip\_orase\_cco;

BEGIN

SELECT orase

INTO v\_lista

FROM excursie\_cco

WHERE cod\_excursie=v\_cod;

DBMS\_OUTPUT.PUT\_LINE('nr orase vizitate: '||v\_lista.last);

FOR j IN v\_lista.FIRST..v\_lista.LAST loop

DBMS\_OUTPUT.PUT\_LINE (v\_lista(j));

END LOOP;

END;

/

**d. Pentru fiecare excursie afișați lista orașelor vizitate.**

DECLARE

CURSOR coduri\_excursii is SELECT cod\_excursie from excursie\_cco;

v\_lista tip\_orase\_cco;

v\_cod NUMBER(4);

BEGIN

OPEN coduri\_excursii;

LOOP

FETCH coduri\_excursii INTO v\_cod;

EXIT WHEN coduri\_excursii%notfound;

DBMS\_OUTPUT.PUT\_LINE('excursie: '||v\_cod);

SELECT orase

INTO v\_lista

FROM excursie\_cco

WHERE cod\_excursie=v\_cod;

DBMS\_OUTPUT.PUT\_LINE('nr orase vizitate: '||v\_lista.last);

FOR j IN v\_lista.FIRST..v\_lista.LAST loop

DBMS\_OUTPUT.PUT\_LINE (v\_lista(j));

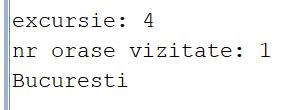
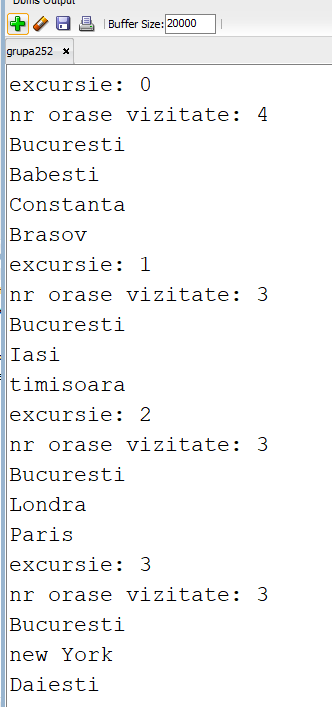
END LOOP;

END LOOP;

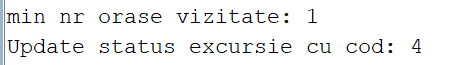
CLOSE coduri\_excursii;

END;

/

****

**e. Anulați excursiile cu cele mai puține orașe vizitate.**



--e

DECLARE

CURSOR coduri\_excursii is SELECT cod\_excursie from excursie\_cco;

v\_lista tip\_orase\_cco;

v\_cod NUMBER(4);

v\_min NUMBER(6):=10000;

BEGIN

OPEN coduri\_excursii;

LOOP

FETCH coduri\_excursii INTO v\_cod;

EXIT WHEN coduri\_excursii%notfound;

SELECT orase

INTO v\_lista

FROM excursie\_cco

WHERE cod\_excursie=v\_cod;

IF v\_lista.last<v\_min THEN v\_min:=v\_lista.last;

END IF;

END LOOP;

CLOSE coduri\_excursii;

DBMS\_OUTPUT.PUT\_LINE('min nr orase vizitate: '||v\_min);

OPEN coduri\_excursii;

LOOP

FETCH coduri\_excursii INTO v\_cod;

EXIT WHEN coduri\_excursii%notfound;

SELECT orase

INTO v\_lista

FROM excursie\_cco

WHERE cod\_excursie=v\_cod;

IF v\_lista.last=v\_min THEN

UPDATE excursie\_cco

SET status='anulat'

WHERE cod\_excursie=v\_cod;

DBMS\_OUTPUT.PUT\_LINE('Update status excursie cu cod: '||v\_cod);

END IF;

END LOOP;

CLOSE coduri\_excursii;

END;

/

E4 Adaptați cerințele exercițiilor 9 și 10 pentru diagrama proiectului prezentată la materia Baze de

Date din anul I. Rezolvați aceste exerciții în PL/SQL, folosind baza de date proprie.

10. Creați tabelul emp\_test\_\*\*\* cu coloanele employee\_id și last\_name din tabelul employees.

Adăugați în acest tabel un nou câmp numit telefon de tip tablou imbricat. Acest tablou va menține

pentru fiecare salariat toate numerele de telefon la care poate fi contactat. Inserați o linie nouă în

tabel. Actualizați o linie din tabel. Afișați informațiile din tabel. Ștergeți tabelul și tipul.

CREATE TABLE emp\_test\_\*\*\* AS

SELECT employee\_id, last\_name FROM employees

WHERE ROWNUM <= 2;

CREATE OR REPLACE TYPE tip\_telefon\_\*\*\* IS TABLE OF VARCHAR(12);

/

ALTER TABLE emp\_test\_\*\*\*

ADD (telefon tip\_telefon\_\*\*\*)

NESTED TABLE telefon STORE AS tabel\_telefon\_\*\*\*;

INSERT INTO emp\_test\_\*\*\*

VALUES (500, 'XYZ',tip\_telefon\_\*\*\*('074XXX', '0213XXX', '037XXX'));

UPDATE emp\_test\_\*\*\*

SET telefon = tip\_telefon\_\*\*\*('073XXX', '0214XXX')

WHERE employee\_id=100;

SELECT a.employee\_id, b.\*

FROM emp\_test\_\*\*\* a, TABLE (a.telefon) b;

DROP TABLE emp\_test\_\*\*\*;

DROP TYPE tip\_telefon\_\*\*\*;