**Temă 2 Lab SGBD – 17.10.2024**

**Șerba Ioana-Raluca grupa 241**

**Exercițiul E2 din Laborator PLSQL2.pdf.:**

* **Enunț:**

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).

a. Inserați 5 înregistrări în tabel.

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.

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

d. Pentru fiecare excursie afișați lista orașelor vizitate.

e. Anulați excursiile cu cele mai puține orașe vizitate.

* **Cod Sursă:**

*--E2*/  
 CREATE OR REPLACE TYPE tip\_orase\_miau IS VARRAY(10) OF VARCHAR2(20);  
/  
  
 CREATE TABLE excursie\_miau (  
 COD\_EXCURSIE NUMBER(4),  
 DENUMIRE VARCHAR2(20),  
 ORASE tip\_orase\_miau,  
 STATUS VARCHAR2(20));  
 *--de ce daca dau create or replace nu mai merge  
 --DROP TABLE EXCURSIE\_MIAU;*/  
*--a inserati 5 inregistrari in tabel* INSERT INTO excursie\_miau  
 VALUES (1,'pisica maro', tip\_orase\_miau('canapea','patul meu'), 'disponobila');  
 INSERT INTO excursie\_miau  
 VALUES (2,'pisica portocalie', tip\_orase\_miau('bucatarie','chiuveta'), 'anulata');  
 INSERT INTO excursie\_miau  
 VALUES (3,'pisica alba', tip\_orase\_miau('frigider','patul ei'), 'anulata');  
 INSERT INTO excursie\_miau  
 VALUES (4,'pisica neagra', tip\_orase\_miau('covor','dulap'), 'disponobila');  
 INSERT INTO excursie\_miau  
 VALUES (5,'pisica gri', tip\_orase\_miau('birou','masa'), 'disponobila');  
/  
*--------------------------------------------*

*A screenshot of a computer

Description automatically generated*

*--b adaugati un oras nou in lista care  
--va fi ultimul vizitat*DECLARE  
 oras\_ult tip\_orase\_miau := tip\_orase\_miau();  
BEGIN  
 SELECT ORASE  
 INTO oras\_ult  
 FROM EXCURSIE\_MIAU  
 WHERE COD\_EXCURSIE = 4;  
   
 *--il extind* oras\_ult.extend;  
 oras\_ult(oras\_ult.last) := 'fereastra';  
   
 UPDATE EXCURSIE\_MIAU   
 SET ORASE = oras\_ult  
 WHERE COD\_EXCURSIE = 4;  
   
END;  
/  
 SELECT *\** FROM EXCURSIE\_MIAU WHERE COD\_EXCURSIE=4;  
 *--a dat bine GRUPA241.TIP\_ORASE\_MIAU('covor', 'dulap', 'fereastra')*/

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

*--va fi al doilea oras vizitat in excursia respectiva*/  
DECLARE  
 oras\_2 tip\_orase\_miau := tip\_orase\_miau();  
BEGIN  
 SELECT ORASE  
 INTO oras\_2  
 FROM EXCURSIE\_MIAU  
 WHERE COD\_EXCURSIE = 5;  
   
 *--DBMS\_OUTPUT.PUT\_LINE(oras\_2(2));  
 --il extind* oras\_2.extend;  
 oras\_2(oras\_2.last) := '';  
 *--mut ce era de la 2 mai mare in capatul listei* FOR I IN 2..oras\_2.last LOOP  
 oras\_2(oras\_2.last + 2 - I) := oras\_2(oras\_2.last + 1 - I);  
 END LOOP;  
 oras\_2(2) := 'pe caiet';  
 *--DBMS\_OUTPUT.PUT\_LINE(oras\_2(2));* UPDATE EXCURSIE\_MIAU   
 SET ORASE = oras\_2  
 WHERE COD\_EXCURSIE = 5;  
   
END;  
/  
 SELECT *\** FROM EXCURSIE\_MIAU WHERE COD\_EXCURSIE=5;  
 *--a dat bine GRUPA241.TIP\_ORASE\_MIAU('birou', 'pe caiet', 'masa')*

*A screenshot of a computer

Description automatically generated* A screenshot of a computer

Description automatically generated/  
  
*--inversati ordinea de vizitare a 2 orase al caror nume este specificat*DECLARE  
 oras\_inv tip\_orase\_miau := tip\_orase\_miau();  
 oras1 VARCHAR2(20);  
 oras2 VARCHAR2(20);  
   
BEGIN  
 SELECT ORASE  
 INTO oras\_inv  
 FROM EXCURSIE\_MIAU  
 WHERE COD\_EXCURSIE = 3;  
   
 oras1 := '&oras1';  
 oras2 := '&oras2';  
 *--DBMS\_OUTPUT.PUT\_LINE(oras1 || oras2);* FOR I IN 1..ORAS\_INV.COUNT LOOP  
 IF ORAS\_INV(I) = ORAS1 THEN ORAS\_INV(I) := ORAS2;  
 ELSIF ORAS\_INV(I) = ORAS2 THEN ORAS\_INV(I) := ORAS1;  
 END IF;  
 END LOOP;  
   
 UPDATE EXCURSIE\_MIAU  
 SET ORASE = ORAS\_INV  
 WHERE COD\_EXCURSIE = 3;  
   
   
END;  
/  
 SELECT *\** FROM EXCURSIE\_MIAU WHERE COD\_EXCURSIE = 3;  
 *--a dat bine GRUPA241.TIP\_ORASE\_MIAU('patul ei', 'frigider')* UPDATE EXCURSIE\_MIAU  
 SET ORASE = tip\_orase\_miau('frigider','patul ei')  
 WHERE COD\_EXCURSIE = 3;

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

/  
  
*--eliminati din lista un oras al carui nume este specificat*DECLARE  
 oras\_del tip\_orase\_miau := tip\_orase\_miau();  
 oras\_nou tip\_orase\_miau := tip\_orase\_miau();  
 oras\_scos VARCHAR2(20);  
BEGIN  
 SELECT ORASE  
 INTO oras\_del  
 FROM EXCURSIE\_MIAU  
 WHERE COD\_EXCURSIE = 2;  
   
 oras\_scos := '&oras1';  
   
 FOR I IN 1..ORAS\_DEL.COUNT LOOP  
 IF ORAS\_DEL(I) != ORAS\_SCOS THEN  
 ORAS\_NOU.EXTEND;  
 ORAS\_NOU(ORAS\_NOU.COUNT) := ORAS\_DEL(I);  
 END IF;  
 END LOOP;  
   
 UPDATE EXCURSIE\_MIAU  
 SET ORASE = ORAS\_NOU  
 WHERE COD\_EXCURSIE = 2;  
   
   
END;  
/  
 SELECT *\** FROM EXCURSIE\_MIAU WHERE COD\_EXCURSIE = 2;  
 *--a dat bine GRUPA241.TIP\_ORASE\_MIAU('chiuveta')* UPDATE EXCURSIE\_MIAU  
 SET ORASE = tip\_orase\_miau('bucatarie','chiuveta')  
 WHERE COD\_EXCURSIE = 2;  
/

A computer screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated *A screenshot of a computer

Description automatically generated*  
*--------------------------------------------  
--c se da cod excursie afisati nr orase vizitate + numele oraselor*DECLARE  
 cod\_dat excursie\_miau.cod\_excursie%type;  
 lista\_orase tip\_orase\_miau := tip\_orase\_miau();  
BEGIN  
 cod\_dat := &cod\_excursie;  
 SELECT ORASE  
 INTO LISTA\_ORASE  
 FROM EXCURSIE\_MIAU  
 WHERE COD\_EXCURSIE = COD\_DAT;  
   
 DBMS\_OUTPUT.*PUT\_LINE*('Nr orase vizitate: '|| LISTA\_ORASE.COUNT);  
 FOR I IN 1..LISTA\_ORASE.COUNT LOOP  
 DBMS\_OUTPUT.*PUT*(LISTA\_ORASE(I) || ' ');  
 END LOOP;  
 DBMS\_OUTPUT.*NEW\_LINE*;  
  
END;  
/  
SELECT *\** FROM EXCURSIE\_MIAU;  
/

A screenshot of a computer

Description automatically generated  
*-------------------------------  
--d pt fiecare excursie afisati lista oraselor vizitate*DECLARE  
 TYPE orase\_toate IS TABLE OF tip\_orase\_miau;  
 ORASE\_ALL ORASE\_TOATE;  
BEGIN  
 SELECT ORASE  
 bulk collect INTO ORASE\_ALL  
 FROM EXCURSIE\_MIAU;  
   
 *--DBMS\_OUTPUT.PUT\_LINE(ORASE\_ALL(1)(1));* FOR I IN 1..ORASE\_ALL.LAST LOOP  
 DBMS\_OUTPUT.*PUT\_LINE*('EXCURSIA' || I);  
 FOR J IN 1..ORASE\_ALL(I).COUNT LOOP  
 DBMS\_OUTPUT.*PUT*(ORASE\_ALL(I)(J) || ' ');  
 END LOOP;  
 DBMS\_OUTPUT.*NEW\_LINE*;  
 END LOOP;  
END;  
*--------------*

*A screenshot of a computer

Description automatically generated*/  
*--e anulati excursiile cu cele mai putine orase vizitate*DECLARE  
 TYPE orase\_toate IS TABLE OF tip\_orase\_miau;  
 ORASE\_ALL ORASE\_TOATE;  
 TYPE id\_excursie IS TABLE OF NUMBER(2);  
 id\_excursii id\_excursie := id\_excursie();  
 orase\_min NUMBER(3);  
BEGIN   
 SELECT ORASE  
 BULK COLLECT INTO ORASE\_ALL  
 FROM EXCURSIE\_MIAU;  
 *--gasim minimul* orase\_min := orase\_all(1).LAST;  
 ID\_EXCURSII.EXTEND;  
 ID\_EXCURSII(1) := 1;  
   
 *--dbms\_output.put\_line(orase\_min);* FOR I IN 2..ORASE\_ALL.COUNT LOOP  
 IF ORASE\_ALL(I).COUNT < ORASE\_MIN THEN  
 ORASE\_MIN := ORASE\_ALL(I).COUNT;  
 ID\_EXCURSII.DELETE;  
 ID\_EXCURSII.EXTEND;  
 ID\_EXCURSII(1) := I;  
 ELSIF ORASE\_ALL(I).COUNT = ORASE\_MIN THEN  
 ID\_EXCURSII.EXTEND;  
 ID\_EXCURSII(ID\_EXCURSII.COUNT) := I;  
 END IF;  
 END LOOP;  
   
 *--DBMS\_OUTPUT.PUT\_LINE(ORASE\_MIN);* FOR I IN 1..ID\_EXCURSII.COUNT LOOP  
 UPDATE EXCURSIE\_MIAU  
 SET STATUS = 'anulata'  
 WHERE COD\_EXCURSIE = ID\_EXCURSII(I);  
 *--DBMS\_OUTPUT.PUT\_LINE(id\_excursii(i));* END LOOP;  
END;  
/  
SELECT *\** FROM EXCURSIE\_MIAU;

Înainte de a rula (am făcut ca 2 linii să aibă 3 orașe, care nu era disponibile, deoarece la început toate aveau aceași lungime):

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

După:

A screenshot of a computer

Description automatically generated

(A anulat-o pe cea de lungime 2, care era disponibilă).