

TEMĂ

În această temă voi prezenta compararea timpilor de execuție pentru toate cele 3 tipuri de colecții (vector, tabel indexat, respectiv tabel imbricat) precizând după, de ce unul este mai rapid decât celălalt. De asemenea voi prezenta rezolvarea exercițiului 3 din laboratorul 2 PL/SQL folosind doar operatori multiset în SQL.

Vector vs Tabel Indexat vs Tabel imbricat (Timp de execuție).

Să se inițializeze primele 3.000.000 elemente din fiecare colecție folosind formula $((i*2)+2)*10$.

Rezolvare:

DECLARE

TYPE tab_vec IS VARRAY(3000000) OF NUMBER;

t tab_vec := tab_vec();

TYPE tab_ind IS TABLE OF NUMBER INDEX BY PLS_INTEGER;

t2 tab_ind;

TYPE tab_imb IS TABLE OF NUMBER;

t3 tab_imb := tab_imb();

timp_executie number(10);

start_time number(10);

stop_time number(10);

BEGIN

start_time := dbms_utility.get_time();

for i in 1..3000000 loop

t.extend;

t(i) := ((i*2)+2)*10;

end loop;

stop_time := dbms_utility.get_time();

```

timp_executie := stop_time - start_time;

DBMS_OUTPUT.PUT_LINE('Initializarea in vector a durat ' || timp_executie);


start_time := dbms_utility.get_time();

for i in 1..3000000 loop

    t2(i) := ((i*2)+2)*10;

end loop;

stop_time := dbms_utility.get_time();

timp_executie := stop_time - start_time;

DBMS_OUTPUT.PUT_LINE('Initializarea in tabel indexat a durat ' || timp_executie);


start_time := dbms_utility.get_time();

for i in 1..3000000 loop

    t3.extend;

    t3(i) := ((i*2)+2)*10;

end loop;

stop_time := dbms_utility.get_time();

timp_executie := stop_time - start_time;

DBMS_OUTPUT.PUT_LINE('Initializarea in tabel imbricat a durat ' || timp_executie);

END;

```

Print-Screen:

The screenshot displays the Oracle SQL Developer interface. The main window shows a PL/SQL script with three loops, each measuring the time taken to initialize data in a vector, an indexed table, and a nested table. The script is as follows:

```
13 start_time := dbms_utility.get_time();
14 for i in 1..3000000 loop
15   t.extend;
16   t(i) := ((i*2)+2)*10;
17 end loop;
18 stop_time := dbms_utility.get_time();
19 timp_executie := stop_time - start_time;
20 DBMS_OUTPUT.PUT_LINE('Initializarea in vector a durat ' || timp_executie);
21
22 start_time := dbms_utility.get_time();
23 for i in 1..3000000 loop
24   t2(i) := ((i*2)+2)*10;
25 end loop;
26 stop_time := dbms_utility.get_time();
27 timp_executie := stop_time - start_time;
28 DBMS_OUTPUT.PUT_LINE('Initializarea in tabel indexat a durat ' || timp_executie);
29
30 start_time := dbms_utility.get_time();
31 for i in 1..3000000 loop
32   t3.extend;
33   t3(i) := ((i*2)+2)*10;
34 end loop;
35 stop_time := dbms_utility.get_time();
36 timp_executie := stop_time - start_time;
37 DBMS_OUTPUT.PUT_LINE('Initializarea in tabel imbricat a durat ' || timp_executie);
38 END;
```

The output window on the right shows the results of the execution:

```
Initializarea in vector a durat 114
Initializarea in tabel indexat a durat 64
Initializarea in tabel imbricat a durat 115
```

The status bar at the bottom indicates that the PL/SQL procedure was successfully completed.

Observăm că inițializarea elementelor folosind formula din cerință a durat 114 unități de timp pentru vector, 64 pentru tabelul indexat și 115 pentru tabelul imbricat.

Inițializarea elementelor în tabelul indexat este mult mai rapidă decât în tabelul imbricat sau în vector deoarece, spre deosebire de acestea nu execută instrucțiunea **extend** (o instrucțiune în plus care costă și ea timp pentru a fi executată). Tabelul indexat își alocă dinamic memorie, spre deosebire de vector/tabelul imbricat (necesită comanda **extend**). De asemenea folosește **PLS_INTEGER** ca tip de date care este mult mai optim, dar și recomandat în practică decât number.

Rezolvarea exercițiului 3 din laboratorul 2 PL/SQL folosind operatori MULTISET la nivel de SQL.

2. 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. Inșerați 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.
3. Rezolvați problema anterioară folosind un alt tip de colecție studiat.

Rezolvare:

-- Ex3

```
CREATE OR REPLACE TYPE tip_orase_Roberto IS TABLE OF VARCHAR2(100);
```

```
/
```

```
CREATE TABLE excursie_Roberto (cod_excursie  NUMBER(4) PRIMARY KEY,  
                                denumire      VARCHAR2(20),  
                                orase         tip_orase_Roberto,  
                                status        VARCHAR2(20)  
                                ) NESTED TABLE orase STORE AS tabb_imb_orase;
```

```
select * from excursie_Roberto;
```

--a) Inșerați 5 înregistrări în tabel.

```
INSERT INTO excursie_Roberto VALUES (  
    1,  
    'La arabi',  
    tip_orase_Roberto('Dubai', 'Abu Dhabi', 'Schardscha'),  
    'OPEN'
```

);

INSERT INTO excursie_Robertto VALUES (

2,

'Pe o insula Pustie',

tip_orase_robertto('Hawai', 'Ceva frumos', 'Takimi'),

'CLOSED'

);

INSERT INTO excursie_Robertto VALUES (

3,

'La munte',

tip_orase_Robertto('Brasov', 'Tusnad', 'Sinaia'),

'OPEN'

);

INSERT INTO excursie_Robertto VALUES (

4,

'In Suedia',

tip_orase_Robertto('Umea', 'Stockholm', 'Ystad'),

'CLOSED'

);

INSERT INTO excursie_Robertto VALUES (

5,

'In Italia',

tip_orase_Robertto('Milano', 'Roma', 'Bergamo'),

'OPEN'

);

```
select * from excursie_Roberto;
```

--b) Actualizați coloana orase pentru o excursie specificată:

-- b1) adăugați un oraș nou în listă, ce va fi ultimul vizitat în excursia respectivă;

```
INSERT INTO TABLE (SELECT orase
FROM excursie_Roberto
WHERE cod_excursie = 1)
VALUES('Habibi');
```

```
select * from excursie_Robertto;
```

-- b2) adăugați un oraș nou în listă, ce va fi al doilea oraș vizitat în excursia respectivă;

```
CREATE TABLE aux_imb (ID NUMBER(10), nume_oras VARCHAR2(100));
```

```
INSERT INTO aux_imb VALUES (1,(SELECT * FROM TABLE (SELECT orase
                                FROM excursie_Roberto
                                WHERE cod_excursie = 2)
                                WHERE ROWNUM = 1));
```

```
INSERT INTO aux_imb VALUES (2, 'Vulcanos');
```

```
UPDATE excursie_Robertto  
SET orase = (SELECT CAST (COLLECT(nume_oras) AS tip_orase_Robertto)  
FROM aux_imb) MULTISET UNION DISTINCT (SELECT orase  
FROM excursie_Robertto  
WHERE cod_excursie = 2)
```

```
WHERE cod_excursie = 2;
```

```
/
```

```
select * from excursie_Robertto;
```

--b3) - inversați ordinea de vizitare a două dintre orașe al căror nume este specificat;

```
UPDATE TABLE (SELECT orase
```

```
    from excursie_Robertto
```

```
    where cod_excursie = 3) a
```

```
SET VALUE(a) = CASE
```

```
    WHEN COLUMN_VALUE = 'Brasov' THEN 'Tusnad'
```

```
    WHEN COLUMN_VALUE = 'Tusnad' THEN 'Brasov'
```

```
    WHEN COLUMN_VALUE <> 'Brasov' or COLUMN_VALUE <> 'Tusnad' THEN
```

```
COLUMN_VALUE
```

```
END;
```

```
select * from excursie_Robertto;
```

--b4) - eliminați din listă un oraș al cărui nume este specificat.

```
DELETE FROM TABLE (SELECT orase
```

```
    FROM excursie_Robertto
```

```
    WHERE cod_excursie = 4) a
```

```
WHERE COLUMN_VALUE = 'Umea';
```

```
select * from excursie_Robertto;
```

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

```
SELECT cardinality(orse) Numar_orse_vizitate, orse  
FROM excursie_Robertto  
WHERE cod_excursie = 4;
```

--d) Pentru fiecare excursie afișați lista orașelor vizitate.

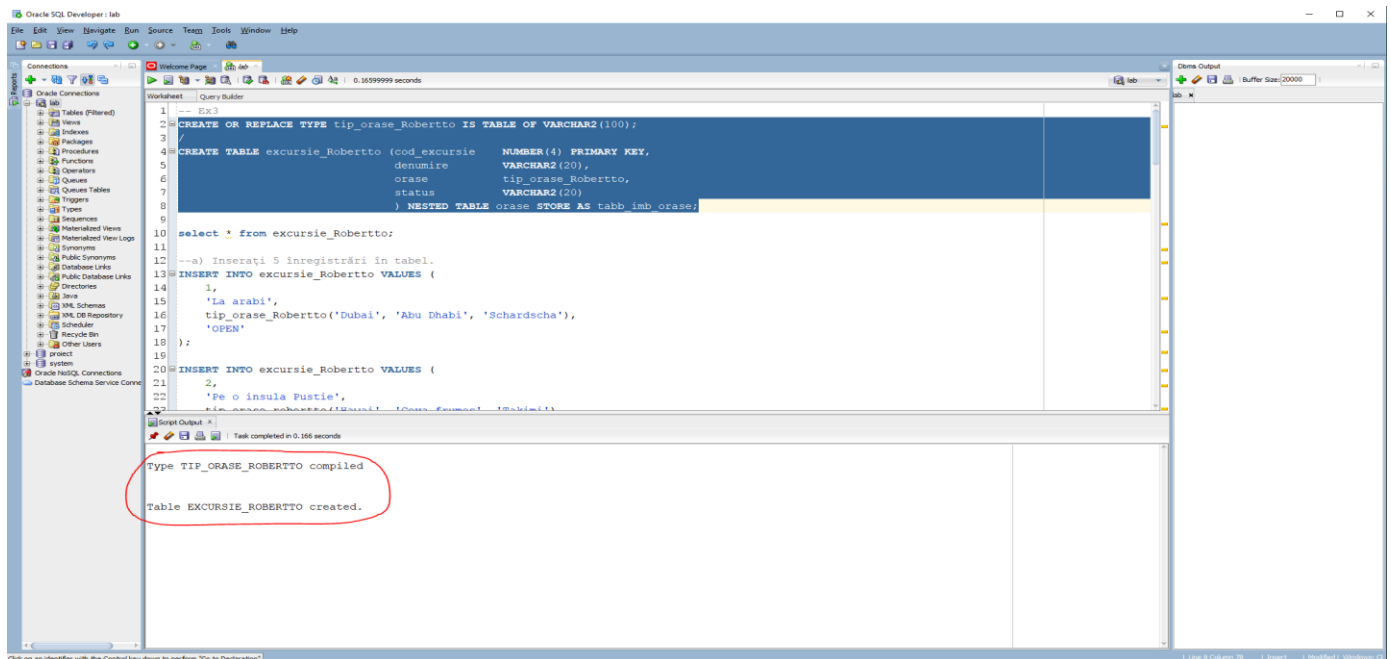
```
SELECT denumire, orse  
FROM excursie_Robertto;
```

--e) Anulați excursiile cu cele mai puține orașe vizitate.

```
UPDATE excursie_Robertto  
SET status = 'CLOSED'  
WHERE cardinality(orse) = (SELECT MIN(Numar)  
                           FROM (SELECT cardinality(orse) Numar, orse  
                                FROM excursie_Robertto));
```

```
select * from excursie_Robertto;
```

Print-Screen:



Verificăm dacă tabelul nostru excursie_Robertto s-a creat cu succes.

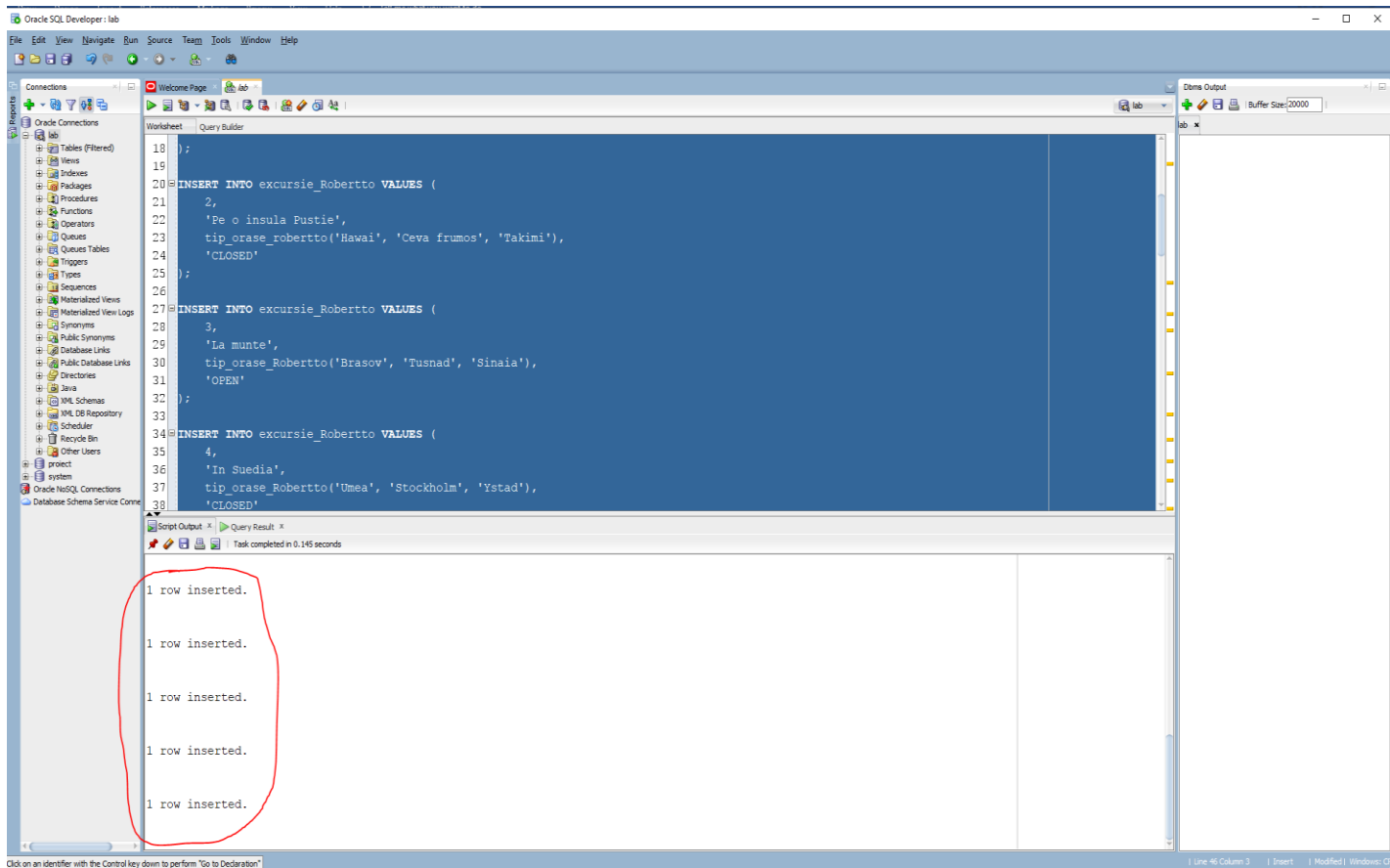
```
10 select * from excursie_Robertto;
11
12 --a) Inerați 5 înregistrări în tabel.
13 INSERT INTO excursie_Robertto VALUES (
14     1,
15     'La arabi',
16     tip_orase_Robertto('Dubai', 'Abu Dhabi'),
17     'OPEN'
18 );
19
20 INSERT INTO excursie_Robertto VALUES (
21     2,
22     'Pe o insula Pustie',
23     tip_orase_Robertto('Hawaii', 'Ceva frumos'),
24     'CLOSED'
25 );
26
27 INSERT INTO excursie_Robertto VALUES (
28     3,
29     'La munte',
30     tip_orase_Robertto('Brasov', 'Tusnad', 'Sinaia'),
31     'OPEN'
32 );
33
34 INSERT INTO excursie_Robertto VALUES (
35     4,
36     'In Suedia',
37     tip_orase_Robertto('Umea', 'Stockholm', 'Ystad'),
38     'CLOSED'
39 );
40
```

Script Output x Query Result x

SQL All Rows Fetched: 0 in 0.006 seconds

COD_EXC... DENUMIRE ORASE STATUS

a) Inerați 5 înregistrări în tabel.

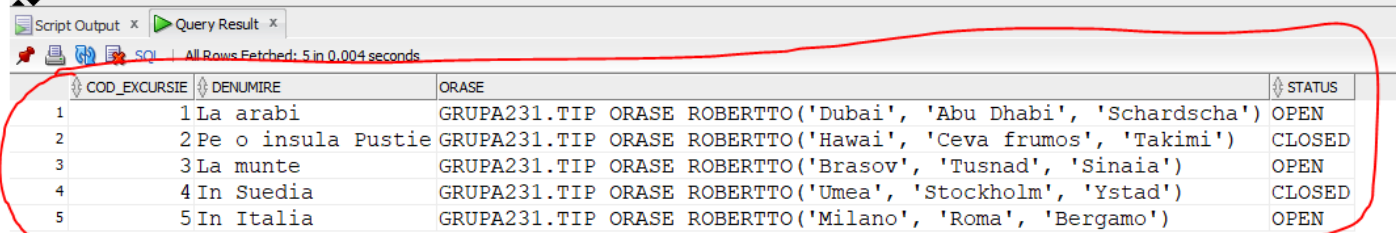


Verificăm dacă datele noastre au fost inserate cu succes in tabelul nostru excursie_Robertto.

```
48 select * from excursie_Robertto;
```

```
49
```

```
50 --b) Actualizați coloana orase pentru o excursie specificată:
```



COD_EXCURSIE	DENUMIRE	ORASE	STATUS
1	La arabi	GRUPA231.TIP ORASE ROBERTTO('Dubai', 'Abu Dhabi', 'Schardscha')	OPEN
2	Pe o insula Pustie	GRUPA231.TIP ORASE ROBERTTO('Hawai', 'Ceva frumos', 'Takimi')	CLOSED
3	La munte	GRUPA231.TIP ORASE ROBERTTO('Brasov', 'Tusnad', 'Sinaia')	OPEN
4	In Suedia	GRUPA231.TIP ORASE ROBERTTO('Umea', 'Stockholm', 'Ystad')	CLOSED
5	In Italia	GRUPA231.TIP ORASE ROBERTTO('Milano', 'Roma', 'Bergamo')	OPEN

b) Actualizați coloana orase pentru o excursie specificată:

b1) adăugați un oraș nou în listă, ce va fi ultimul vizitat în excursia respectivă;

```
53 INSERT INTO TABLE (SELECT orase
54 FROM excursie_Robertto
55 WHERE cod_excursie = 1)
56 VALUES('Habibi');
57
58 select * from excursie_Robertto;
59
```

Script Output x Query Result x
Task completed in 0.049 seconds

1 row inserted.

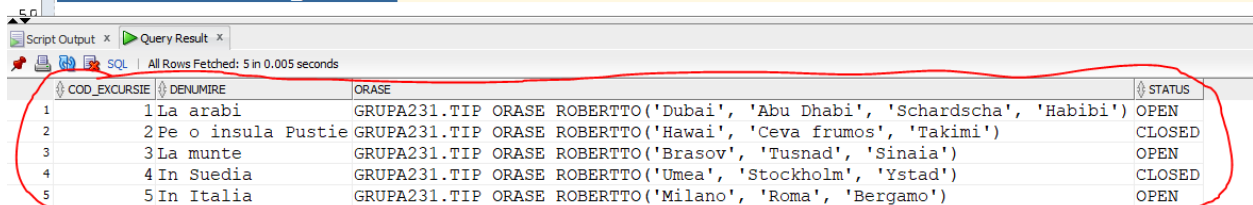
1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

```
58 select * from excursie_Robertto;
```



COD_EXCURSIE	DENUMIRE	ORASE	STATUS
1	La arabi	GRUPA231.TIP ORASE ROBERTTO('Dubai', 'Abu Dhabi', 'Schardscha', 'Habibi')	OPEN
2	Pe o insula Pustie	GRUPA231.TIP ORASE ROBERTTO('Hawai', 'Ceva frumos', 'Takimi')	CLOSED
3	La munte	GRUPA231.TIP ORASE ROBERTTO('Brasov', 'Tusnad', 'Sinaia')	OPEN
4	In Suedia	GRUPA231.TIP ORASE ROBERTTO('Umea', 'Stockholm', 'Ystad')	CLOSED
5	In Italia	GRUPA231.TIP ORASE ROBERTTO('Milano', 'Roma', 'Bergamo')	OPEN

b2) adăugați un oraș nou în listă, ce va fi al doilea oraș vizitat în excursia respectivă;

```
60 -- b2) adăugați un oraș nou în listă, ce va fi al doilea oraș vizitat în excursia respectivă;
61 CREATE TABLE aux_imb (ID NUMBER(10), nume_oras VARCHAR2(100));
62
63 INSERT INTO aux_imb VALUES (1, (SELECT * FROM TABLE (SELECT orase
64                                     FROM excursie_Robertto
65                                     WHERE cod_excursie = 2)
66                                     WHERE ROWNUM = 1));
67 INSERT INTO aux_imb VALUES (2, 'Vulcanos');
68
69 UPDATE excursie_Robertto
70 SET orase = (SELECT CAST (COLLECT(nume_oras) AS tip_orase_Robertto)
71             FROM aux_imb) MULTISET UNION DISTINCT (SELECT orase
72                                                     FROM excursie_Robertto
73                                                     WHERE cod_excursie = 2)
74 WHERE cod_excursie = 2;
75 /
76
77 select * from excursie_Robertto;
78
```

Script Output x Query Result x
Task completed in 0.059 seconds

1 row inserted.

Table AUX_IMB created.

1 row inserted.

1 row inserted.

1 row updated.

```
77 select * from excursie_Robertto;
```

78

```
79 --b3) - inversați ordinea de vizitare a două dintre orașe al căror nume este specificat;
```

Script Output x Query Result x
All Rows Fetched: 5 in 0.007 seconds

COD_EXCURSIE	DENUMIRE	ORASE	STATUS
1	1La arabi	GRUPA231.TIP ORASE ROBERTTO('Dubai', 'Abu Dhabi', 'Schardscha', 'Habibi')	OPEN
2	2Pe o insula Pustie	GRUPA231.TIP ORASE ROBERTTO('Hawai', 'Vulcanos', 'Ceva frumos', 'Takimi')	CLOSED
3	3La munte	GRUPA231.TIP ORASE ROBERTTO('Brasov', 'Tusnad', 'Sinaia')	OPEN
4	4In Suedia	GRUPA231.TIP ORASE ROBERTTO('Umea', 'Stockholm', 'Ystad')	CLOSED
5	5In Italia	GRUPA231.TIP ORASE ROBERTTO('Milano', 'Roma', 'Bergamo')	OPEN

b3) inversați ordinea de vizitare a două dintre orașe al căror nume este specificat;

```
79 --b3) - inversați ordinea de vizitare a două dintre orașe al căror nume este specificat;
80
81 UPDATE TABLE (SELECT orase
82                 from excursie_Robertto
83                 where cod_excursie = 3) a
84 SET VALUE(a) = CASE
85                 WHEN COLUMN_VALUE = 'Brasov' THEN 'Tusnad'
86                 WHEN COLUMN_VALUE = 'Tusnad' THEN 'Brasov'
87                 WHEN COLUMN_VALUE <> 'Brasov' or COLUMN_VALUE <> 'Tusnad' THEN COLUMN_VALUE
88                 END;
89
90 select * from excursie_Robertto;
91
```

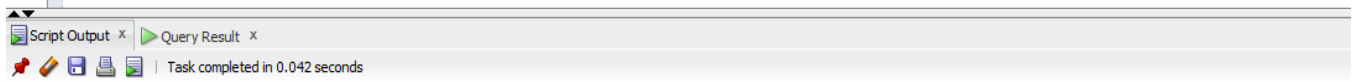


Table AUX_IMB created.

1 row inserted.

1 row inserted.

1 row updated.

3 rows updated.

```
90 select * from excursie_Robertto;
91
92 --b4) - eliminați din listă un oraș al cărui nume este specificat.
93 DELETE FROM TABLE (SELECT orase
94                        FROM excursie_Robertto
```

Script Output x Query Result x				
All Rows Fetched: 5 in 0.008 seconds				
COD_EXCURSIE	DENUMIRE	ORASE	STATUS	
1	1La arabi	GRUPA231.TIP ORASE ROBERTTO('Dubai', 'Abu Dhabi', 'Schardscha', 'Habibi')	OPEN	
2	2Pe o insula Pustie	GRUPA231.TIP ORASE ROBERTTO('Hawai', 'Vulcanos', 'Ceva frumos', 'Takimi')	CLOSED	
3	3La munte	GRUPA231.TIP ORASE ROBERTTO('Tusnad', 'Brasov', 'Sinaia')	OPEN	
4	4In Suedia	GRUPA231.TIP ORASE ROBERTTO('Umea', 'Stockholm', 'Ystad')	CLOSED	
5	5In Italia	GRUPA231.TIP ORASE ROBERTTO('Milano', 'Roma', 'Bergamo')	OPEN	

b4) eliminați din listă un oraș al cărui nume este specificat.

```
92 --b4) - eliminați din listă un oraș al cărui nume este specificat.
93 DELETE FROM TABLE (SELECT orase
94                        FROM excursie_Robertto
95                        WHERE cod_excursie = 4) a
96 WHERE COLUMN_VALUE = 'Umea';
97
98 select * from excursie_Robertto;
99
100 --c) Pentru o excursie al cărui cod este dat, afișați numărul de orașe
```

Script Output x Query Result x
Task completed in 0.047 seconds

1 row deleted.

```
98 select * from excursie_Robertto;
99
100 --c) Pentru o excursie al cărui cod este dat, afișați numărul de orașe vizitate, respectiv numele orașelor.
```

Script Output x Query Result x
All Rows Fetched: 5 in 0.004 seconds

COD_EXCURSIE	DENUMIRE	ORASE	STATUS
1	1 La arabi	GRUPA231.TIP ORASE ROBERTTO('Dubai', 'Abu Dhabi', 'Schardscha', 'Habibi')	OPEN
2	2 Pe o insula Pustie	GRUPA231.TIP ORASE ROBERTTO('Hawai', 'Vulcanos', 'Ceva frumos', 'Takimi')	CLOSED
3	3 La munte	GRUPA231.TIP ORASE ROBERTTO('Tusnad', 'Brasov', 'Sinaia')	OPEN
4	4 In Suedia	GRUPA231.TIP ORASE ROBERTTO('Stockholm', 'Ystad')	CLOSED
5	5 In Italia	GRUPA231.TIP ORASE ROBERTTO('Milano', 'Roma', 'Bergamo')	OPEN

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

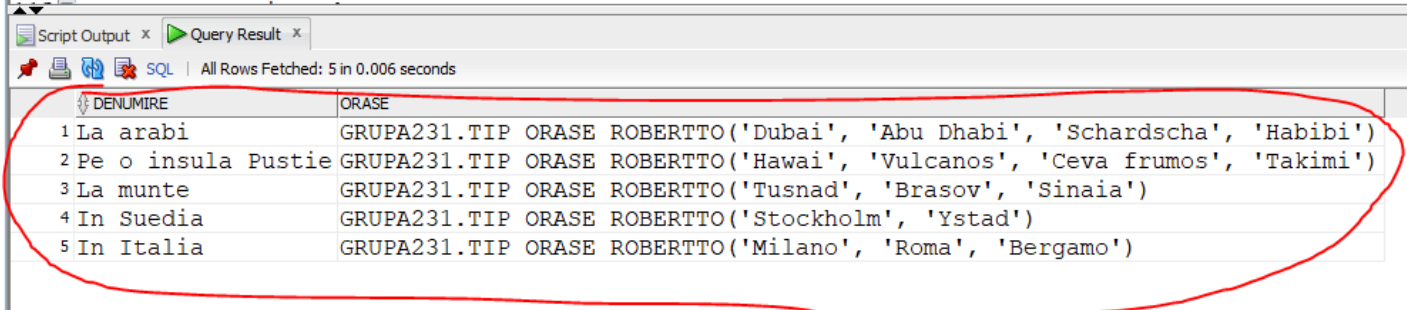
```
100 --c) Pentru o excursie al cărui cod este dat, afișați numărul de orașe vizitate, respectiv numele orașelor.
101 SELECT cardinality(orase) Numar_orase_vizitate, orase
102 FROM excursie_Robertto
103 WHERE cod_excursie = 4;
```

Script Output x Query Result x
All Rows Fetched: 1 in 0.006 seconds

NUMAR_ORASE_VIZITATE	ORASE
1	2 GRUPA231.TIP ORASE ROBERTTO('Stockholm', 'Ystad')

d) Pentru fiecare excursie afișați lista orașelor vizitate.

```
105 --d) Pentru fiecare excursie afișați lista orașelor vizitate.
106 SELECT denumire, orase
107 FROM excursie_Robertto;
108
109 --e) Anulați excursiile cu cele mai puține orașe vizitate.
```



Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.006 seconds

DENUMIRE	ORASE
1 La arabi	GRUPA231.TIP ORASE ROBERTTO('Dubai', 'Abu Dhabi', 'Schardscha', 'Habibi')
2 Pe o insula Pustie	GRUPA231.TIP ORASE ROBERTTO('Hawai', 'Vulcanos', 'Ceva frumos', 'Takimi')
3 La munte	GRUPA231.TIP ORASE ROBERTTO('Tusnad', 'Brasov', 'Sinaia')
4 In Suedia	GRUPA231.TIP ORASE ROBERTTO('Stockholm', 'Ystad')
5 In Italia	GRUPA231.TIP ORASE ROBERTTO('Milano', 'Roma', 'Bergamo')

e) Anulați excursiile cu cele mai puține orașe vizitate.

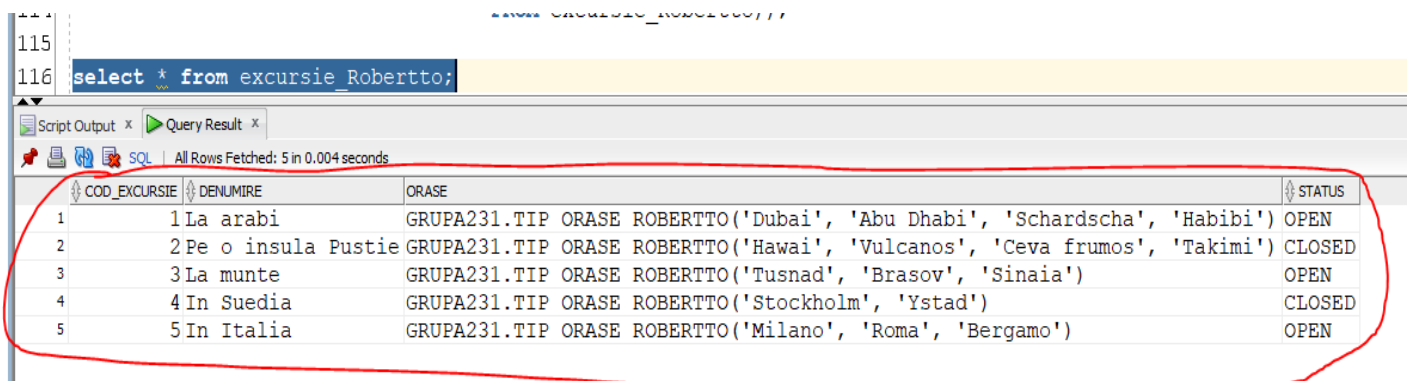
```
109 --e) Anulați excursiile cu cele mai puține orașe vizitate.
110 UPDATE excursie_Robertto
111 SET status = 'CLOSED'
112 WHERE cardinality(orase) = (SELECT MIN(Numar)
113                             FROM (SELECT cardinality(orase) Numar, orase
114                                   FROM excursie_Robertto));
115
```

Script Output x Query Result x

Task completed in 0.052 seconds

1 row deleted.

1 row updated.



Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.004 seconds

COD_EXCURSIE	DENUMIRE	ORASE	STATUS
1	1 La arabi	GRUPA231.TIP ORASE ROBERTTO('Dubai', 'Abu Dhabi', 'Schardscha', 'Habibi')	OPEN
2	2 Pe o insula Pustie	GRUPA231.TIP ORASE ROBERTTO('Hawai', 'Vulcanos', 'Ceva frumos', 'Takimi')	CLOSED
3	3 La munte	GRUPA231.TIP ORASE ROBERTTO('Tusnad', 'Brasov', 'Sinaia')	OPEN
4	4 In Suedia	GRUPA231.TIP ORASE ROBERTTO('Stockholm', 'Ystad')	CLOSED
5	5 In Italia	GRUPA231.TIP ORASE ROBERTTO('Milano', 'Roma', 'Bergamo')	OPEN

Popescu Paullo Robertto Karloss

Grupa 231

Temă SGBD #6