

Tablouri indexate vs. Tablouri imbricate vs. Vectori

Cerință: Comparați, din punct de vedere al optimizării, cele trei tipuri de colecții: tablouri indexate, tablouri imbricate și vectori.

Soluție:

Pentru a putea realiza o comparație între cele trei, avem nevoie să implementăm același exercițiu pe fiecare colecție în parte.

Exercițiu: Să se genereze 10 000 000 de valori random cuprinse între 1 și 1 000 000 000 într-o colecție. Afișați în cât timp a fost realizată rularea implementării, alături de câte numere conține colecția (acest lucru nu va fi făcut cu funcția COUNT).

Tablou indexat

```
DECLARE
    TYPE tab_ind IS TABLE OF NUMBER INDEX BY PLS_INTEGER;
    val tab_ind;
    countByMe number := 0;
    time1 NUMBER := dbms_utility.get_time;
BEGIN
    select trunc(dbms_random.value(1,1000000000)) "RandomValues"
    BULK COLLECT INTO val
    from dual
    connect by level <= 10000000;

    FOR i IN val.FIRST .. val.LAST LOOP
        countByMe := countByMe + 1;
    END LOOP;

    dbms_output.put_line('Elapsed time ASSOCIATIVE ARRAY = ' ||
                        (dbms_utility.get_time - time1)/100 ||
                        ' for a table with ' || countByMe || ' numbers.');
```

END;

Tablou imbricat

```
DECLARE
    TYPE tab_imb IS TABLE OF NUMBER;
    val tab_imb := tab_imb();
    countByMe number := 0;
    time1 NUMBER := dbms_utility.get_time;
BEGIN
    val.extend(1000000);

    select trunc(dbms_random.value(1,1000000000)) "RandomValues"
    BULK COLLECT INTO val
    from dual
    connect by level <= 100000000;

    FOR i IN val.FIRST .. val.LAST LOOP
        countByMe := countByMe + 1;
    END LOOP;

    dbms_output.put_line('Elapsed time NESTED TABLES = ' ||
        (dbms_utility.get_time - time1)/100 ||
        ' for a table with ' || countByMe || ' numbers.');
```

END;

Vector

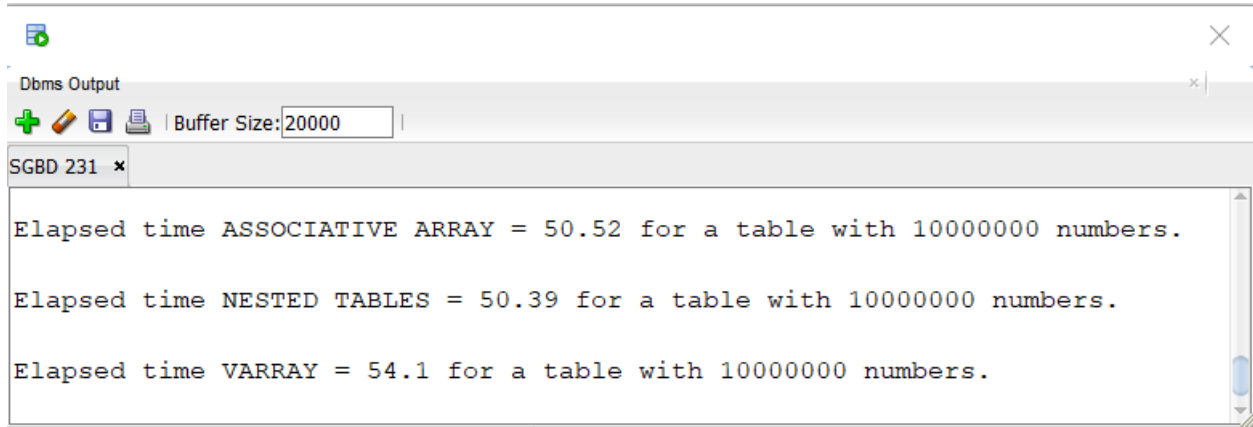
```
DECLARE
    TYPE tab_vec IS VARRAY(100000000) OF PLS_INTEGER;
    val tab_vec := tab_vec();
    countByMe number := 0;
    time1 NUMBER := dbms_utility.get_time;
BEGIN
    select trunc(dbms_random.value(1,1000000000)) "RandomValues"
    BULK COLLECT INTO val
    from dual
    connect by level <= 100000000
    order by 1;

    FOR i IN val.FIRST .. val.LAST LOOP
        countByMe := countByMe + 1;
    END LOOP;

    dbms_output.put_line('Elapsed time VARRAY = ' ||
        (dbms_utility.get_time - time1)/100 ||
        ' for a table with ' || countByMe || ' numbers.');
```

END;

OUTPUT



The screenshot shows a 'Dbms Output' window with a 'Buffer Size' of 20000. It contains three lines of text reporting elapsed times for different table structures, each with 10,000,000 numbers.

```
Elapsed time ASSOCIATIVE ARRAY = 50.52 for a table with 10000000 numbers.  
Elapsed time NESTED TABLES = 50.39 for a table with 10000000 numbers.  
Elapsed time VARRAY = 54.1 for a table with 10000000 numbers.
```

După cum se poate vedea în imagine, tablourile imbricate sunt cele mai eficiente în cazul exercițiului de mai sus, iar vectorii sunt cei mai ineficienți.

➔ TABLOURI IMBRICATE > TABLOURI INDEXATE > VECTORI ➔