PLSQL Tuning: Bulk Collect with Dynamic SQL

How to use **Bulk Collect with Dynamic SQL** or Can I use execute immediate with Bulk Collect? With this blog post I am trying to answer this question with very simple example. Yes we can use Bulk Collect with Dynamic SQLs and can improves performance by **minimizing the number of context switches** between the PL/SQL and SQL engines.

I have a "EMP" table with "11246872" records.

```
SQL> select count(*) from emp;
  COUNT(*)
----------
  11246872
```

Lets run a **sample code for Dynamic SQL** in EMP table without using bulk-collect.

```
SQL> declare
  2     l_sql varchar2(4000);
  3     l_row emp%rowtype;
  4     c sys_refcursor;
  5  begin
  6     l_sql := 'select * from emp';
  7     open c for l_sql;
  8
  9     loop
 10             fetch c into l_row;
 11             exit when c%notfound;
 12             -- some operation
 13     end loop;
 14     close c;
 15  end;
 16  /
PL/SQL procedure successfully completed.
Elapsed: 00:03:12.84
```

Above code is nice and simple, which we want to rewrite so that we can use performance benefits of bulk collect with dynamic sql. Lets modify above code of **dynamic sql with bulk collect** and execute it to check the performance.

```
SQL> declare
  2     l_sql varchar2(4000);
  3     type t_tab is table of emp%rowtype index by binary_integer;
  4     l_tab t_tab;
  5     c sys_refcursor;
  6  begin
  7     l_sql := 'select * from emp';
  8     open c for l_sql;
  9     loop
 10             fetch c bulk collect into l_tab limit 1000;
 11
 12             for i in 1..l_tab.count
 13             loop
 14                     null;
 15                     -- some operation
 16             end loop;
 17             exit when c%notfound;
 18     end loop;
 19     close c;
 20  end;
 21  /
PL/SQL procedure successfully completed.
Elapsed: 00:00:28.45
```

We have saved a lot of context switching between the PL/SQL and SQL engines. As we can see, this simple change has given us **more than 85% performance benefit**. Always use Bulk Collect in your code, wherever you are planning to have simple cursor loop fetch.