

开发技术:关于变量、移动以及含义(3) PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/143/2021_2022__E5_BC_80_E5_8F_91_E6_8A_80_E6_c102_143243.htm here is an explanation of the interesting parts of the anonymous block in listing 3. lines 2 and 3. declare variables to retrieve the values returned by the `get_title_or_text` procedure. lines 5-7. declare a nested table to hold the data identified by the cursor variable. lines 8-13. call the stored procedure to return the cursor variable and the count of rows in the result set. lines 16-19. use `bulk collect` to fetch all the rows in the result set into the nested table. then close the cursor variable. note that i use `bulk collect` with an implicit `0select` into to retrieve all the rows identified by the cursor. if you are querying a large volume of data, this approach can consume an unacceptable amount of memory. under such circumstances, you may want to switch to using `bulk collect` with a `limit` clause. lines 21-23. use a `forall` statement to push the data into the joke archive. listing 4 contains a modified version of the previous anonymous block in listing 3, showing how to use the `limit` clause to restrict the number of rows fetched with a `bulk collect` query, thus reducing the amount of memory needed to populate the collection. code listing 4: from results to collection to forall plus limit

```
declare l_count pls_integer. l_jokes sys_refcursor.
type jokes_tt is table of jokes.text%type. l_joke_array jokes_tt :=
jokes_tt ().begin get_title_or_text (title_like_in => %insect%
,return_title_in => false ,joke_count_out => l_count ,jokes_out =>
l_jokes ). dbms_output.put_line (number of jokes found = ||
```

```
l_count). loop -- fetch the next 100 rows. fetch l_jokes bulk collect
into l_joke_array limit 100. exit when l_joke_array.count = 0. --
push them into the archive. forall indx in l_joke_array.first ..
l_joke_array.last insert into joke_archive values (sysdate,
l_joke_array (indx)). end loop. close l_jokes.end.
```

this answer demonstrates how you can take advantage of the flexibility of cursor variables to construct your queries at runtime without losing the performance benefits of the bulk collect and forall features. moving back and forth i need to be able to move back and forth within a cursor result set, comparing the contents of distinct rows. i also would like to be able to read from the end of the result set to the beginning. how can i do that inside a pl/sql program?

100Test 下载
频道开通 , 各类考试题目直接下载。 详细请访问
www.100test.com