

## Profiler パッケージによる

### 【プロシージャのステップ別実行時間の計測 とボトルネックの調査】

**Profiler パッケージ**は、ファンクションやプロシージャで実行する各ステップの実行所要時間を計測するツールである

このことによって、プロシージャ等のボトルネックとなっいていす SQL 文がどれなのかが判明する

dbms\_profiler パッケージのインストール (**sysdba 権限**ユーザーで実行)

```
@$ORACLE_HOME/rdbms/admin/profload.sql
```

Windows 版は、

```
@%ORACLE_HOME%\rdbms\admin\profload.sql
```

```
@D:\Oracle_Client\product\11.2.0\client_1\rdbms\admin\profload.sql
```

プロファイラの時間計測保存表の作成

UNIX 版

```
connect scott/tiger ← 検証の実行ユーザーで接続する
```

```
@$ORACLE_HOME/rdbms/admin/proftab.sql ← 表の作成
```

Windows 版

```
connect scott/tiger ← 検証の実行ユーザーで接続する
```

```
@%ORACLE_HOME%\rdbms\admin\proftab.sql
```

作成される表と保存データ

表名	保存されるデータ
PLSQL_PROFILER_RUNS	PL/SQL を実行した再の実行時間や経過時間
PLSQL_PROFILER_UNITS	PL/SQL 内の各ユニットに関する情報
PLSQL_PROFILER_DATA	PL/SQL 内の各処理に関する実行回数や実行時間

## dbms\_profile を実行しての、対象プロシージャの実行

```
declare
  err number ;
begin
  ● err := dbms_profiler.start_profiler ('コメント記述');
                                ※ ここでのコメントが、RUN_COMMENT 列に反映される
  調査対象プロシージャ名() ;    例) put_report();
  ● err := dbms_profiler.stop_profiler ;
end ;
/
```

この間に実行された SQL 文について、その SQL ごとに所要時間が計測され記録される

## 取得した情報の中から「実行したプロシージャ」に対するシーケンス番号調査

```
col run_comment format a50
col run_total_time format 999999999999999999
select runid, to_char(run_date, 'yyyy/mm/dd hh24:mi:ss') RUN_DATE,
       run_comment, run_total_time
from   plsql_profiler_runs
order by runid ;
```

RUNID	RUN_DATE	RUN_COMMENT	RUN_TOTAL_TIME
2	2010/01/13 21:35:48		29.63
3	2010/01/13 21:48:29		45.25
4	2010/01/14 09:13:00		193.10
5	2018/06/18 15:15:56	テスト	899.23

※ 調査したい RUNID については、時刻とコメントを参考にして推測する

プロファイラーが取得したステップごとの経過時間を調査

```

column unit_name format a10
column occured format 999999
column line format 9999
column tot_time format 999.999999
column text format a80
select p.unit_name , p.occured , p.tot_time , p.line# line ,
       substr(s.text , 1 , 75) text
from
  (select u.unit_name , d.total_occur occured ,
         (d.total_time / 1000000000) tot_time , d.line#
   from plsql_profiler_units u , plsql_profiler_data d
   where d.runid = u.runid
        and d.unit_number = u.unit_number
        and d.total_occur > 0
        and u.unit_name = '検証対象プロシージャ名'
        and u.runid = 上で調査した番号 ) p ,
     user_source s
where p.unit_name = s.name(+) and p.line# = s.line(+)
order by p.unit_name , p.line# ;

```

全文字、大文字で指定

検証対象プロシージャ名

上で調査した番号

列名	意	味
UNIT_NAME	ユニット名 (調査プロシージャ名)	
OCCURED	この行が実行された回数	
TOT_TIME	実行にかかった時間 (全実行回数分)	
LINE	ユニット (調査プロシージャ) における行番号	
TEXT	プロシージャに記述した行のテキスト	

UNIT_NAME	OCCURED	TOT_TIME	LINE	TEXT
PUT_REPORT	1	.000021	1	procedure put_report is
PUT_REPORT	1	.034606	2	select count(*) cnt from
PUT_REPORT	1	.068046	5	select sum(e.sal) ttl_sal,
PUT_REPORT	3	1.215681	15	for rclRec in c1 loop
PUT_REPORT	1	.000021	16	strWork := 'total Head_cou
.				
.				
.				
.				

## 調査実行例)

- ・調査対象プロシージャ

```
create or replace PROCEDURE Proc_AAA(
  i_param IN NUMBER ,
  o_param OUT VARCHAR2
) IS
  wk_data VARCHAR2(20);
BEGIN
  wk_data := 'テスト開始';
  DBMS_OUTPUT.PUT_LINE( wk_data );
  o_param := '呼出し成功!';
  wk_data := 'テスト終了';
  DBMS_OUTPUT.PUT_LINE( wk_data );
  return;
END Proc_AAA;
/
```

- ・ dbms\_profile を実行しての対象プロシージャの調査

```
declare
  err number ;
begin
  err := dbms_profiler.start_profiler ( 'テスト' );
  Proc_AAA( 5 , :parameter_1 );
  err := dbms_profiler.stop_profiler ;
end ;
/
```

- ・「実行したプロシージャ」に対するシーケンス番号調査

```
col run_comment format a50
col run_total_time format 999999999999999999
select runid, to_char( run_date , 'yyyy/mm/dd hh24:mi:ss' ) RUN_DATE ,
       run_comment , run_total_time
from plsql_profiler_runs
order by runid ;
```

- ・プロファイラーが取得したステップごとの経過時間を調査

```
column unit_name format a10
column occured format 999999
column line format 9999
column tot_time format 999.999999
column text format a80
select p.unit_name , p.occured , p.tot_time , p.line# line ,
       substr(s.text, 1, 75) text
from
  (select u.unit_name , d.total_occur  occured ,
         (d.total_time / 1000000000) tot_time, d.line#
   from plsql_profiler_units u , plsql_profiler_data d
   where d.runid = u.runid
        and d.unit_number = u.unit_number
        and d.total_occur > 0
        and u.unit_name = 'PROC_AAA'
        and u.runid = 1 ) p ,
user_source s
where p.unit_name = s.name(+) and p.line# = s.line(+)
order by p.unit_name , p.line# ;
```

すべて大文字  
で指定のこと

上で調査した番号