

数据价值 创新驱动

# SQL PROFILE介绍

郑林松

WOQU TECH  
.com

WOQU TECH 沃趣

# 个人介绍

- 1、来自沃趣科技
- 2、从事ORACLE相关工作11年
- 3、从事的行业包括通讯行业，电力能源行业，金融行业
- 4、从事DBA工作包括，开发，开发DBA，数据库管理员，架构师等。
- 5、擅长ORACLE优化和故障处理

# 目录

---

- 1 什么是SQL PROFILE
- 2 SQL PROFILE控制范围
- 3 哪些SQL可以应用SQL PROFILE
- 4 SQL PROFILE 的管理
- 5 案例分享

# 什么是SQLPROFILE

- 1、ORACLE10G开始引入的新特性，默认是开启的
- 2、SQL PROFILE是一个信息集合，存放在数据字典中
- 3、SQL PROFILE不会完全冻结一个SQL的执行计划
- 4、SQL PROFILE可以纠正查询优化器产生的错误执行计划

# SQL PROFILE控制范围

通过PROFILE的属性CATEGORY控制

1、PROFILE创建默认是DEFAULT category，可修改

2、通过如下SQL 查看

```
SELECT CATEGORY,NAME FROM DBA_SQL_PROFILES;
```

3、通过查看初始化参数

```
SHOW PARAMETER SQLTUNE_CATEGORY
```

```
ALTER SESSION,ALTER SYSTEM
```

# SQL PROFILE控制范围

```
zls@RAC>execute dbms_sqltune.accept_sql_profile(task_name => 'my_sql_tuning_task_1', task_owner => 'ZLS', category => 'ZLS', replace => TRUE, force_match => false);
PL/SQL procedure successfully completed.

zls@RAC>select name,sql_text,status,category,to_char(signature) from dba_sql_profiles;

SYS_SQLPROF_01595fad68a20000      select /*+ no_index(test test_idx) */ * from test where id1=:1
      ENABLED      DEFAULT      1287398787885333399
SYS_SQLPROF_01595fb981410001      select /*+ no_index(test test_idx) */ * from test where id1='1'
      ENABLED      DEFAULT      7163397840036348297
SYS_SQLPROF_01595fc74c9f0003      select /*+ no_index(test test_idx) */ * from test where id1='1'
      ENABLED      ZLS      16089847742493147600

zls@RAC>alter session set sqltune_category='ZLS';
Session altered.

zls@RAC>explain plan for
  2      select /*+ no_index(test test_idx) */ * from test where id1='1';

Explained.

zls@RAC>select * from table(dbms_xplan.display);

Plan hash value: 2882402178

-----
| Id | Operation          | Name      | Rows  | Bytes | Cost (%CPU)| Time     |
-----+-----+-----+-----+-----+-----+-----+
|  0 | SELECT STATEMENT    |           |      1 |      5 |      1 (0) | 00:00:01 |
|*  1 |  INDEX RANGE SCAN   | TEST_IDX  |      1 |      5 |      1 (0) | 00:00:01 |
-----

Predicate Information (identified by operation id):
-----
   1 - access("ID1"='1')

Note
-----
   - SQL profile "SYS_SQLPROF_01595fc74c9f0003" used for this statement
```

# SQL PROFILE控制范围

```
zls@RAC>alter session set sqltune_category=default;
```

```
Session altered.
```

```
zls@RAC>explain plan for  
2      select /*+ no_index(test test_idx) */ * from test where id1='1';
```

```
Explained.
```

```
zls@RAC>select      * from table(dbms_xplan.display);
```

```
Plan hash value: 1357081020
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	5	6 (0)	00:00:01
* 1	TABLE ACCESS FULL	TEST	1	5	6 (0)	00:00:01

```
Predicate Information (identified by operation id):
```

```
1 - filter("ID1"='1')
```

```
13 rows selected.
```

# 哪些SQL可以应用SQL PROFILE

- 1、SELECT 语句
- 2、UPDATE 语句
- 3、INSERT INTO SELECT语句
- 4、DELETE 语句
- 5、CREATE TABLE AS SELECT语句
- 6、MERGE 语句 (包括update 或者 insert操作)



# SQL PROFILE 的管理

需要的权限

- 1、CREATE ANY SQL PROFILE
- 2、DROP ANY SQL PROFILE
- 3、ALTER ANY SQL PROFILE

# SQL PROFILE 的管理(利用 SQL Tuning Advisor)

```
declare sk_name VARCHAR2(30);
my_task_name VARCHAR2(30);
my_sqltext CLOB;
begin
my_sqltext := 'select /*+ no_index(test test_idx) */ * from test where id1=1';
my_task_name := DBMS_SQLTUNE.CREATE_TUNING_TASK(sql_text => my_sqltext,
7      8      user_name => 'ZLS',
9      scope => 'COMPREHENSIVE',
time_limit => 60,
task_name => 'my_sql_tuning_task_4',
description => 'Task to tune a query on a specified table');
11
12 end;
13 /
```

PL/SQL procedure successfully completed.

```
zls@RAC>begin
2  DBMS_SQLTUNE.EXECUTE_TUNING_TASK( task_name => 'my_sql_tuning_task_4');
end;
4  /
```

PL/SQL procedure successfully completed.

```
zls@RAC>set long 10000
zls@RAC>set longchunksize 1000
set linesize 100 DBMS_SQLTUNE.REPORT_TUNING_TASK( 'my_sql_tuning_task_4') from DUAL;
set heading off
SELECT DBMS_SQLTUNE.REPORT_TUNING_TASK( 'my_sql_tuning_task_4') from DUAL;
zls@RAC>zls@RAC>
```

GENERAL INFORMATION SECTION

```
-----
Tuning Task Name      : my_sql_tuning_task_4
Tuning Task Owner     : ZLS
Workload Type        : Single SQL Statement
Scope                : COMPREHENSIVE
Time Limit(seconds)  : 60
Completion Status     : COMPLETED
Started at           : 01/02/2017 23:54:06
Completed at         : 01/02/2017 23:54:06
-----
```

```
Schema Name: ZLS
SQL ID      : 8qwf7247k4v2k
SQL Text    : select /*+ no_index(test test_idx) */ * from test where id1=1
```

# SQL PROFILE 的管理(利用 SQL Tuning Advisor)

```
FINDINGS SECTION (1 finding)
-----
1- SQL Profile Finding (see explain plans section below)
-----
A potentially better execution plan was found for this statement.

Recommendation (estimated benefit: 90.86%)
-----
- Consider accepting the recommended SQL profile.
  execute dbms_sqltune.accept_sql_profile(task_name =>
    'my_sql_tuning_task_4', task_owner => 'ZLS', replace => TRUE);

Validation results
-----
The SQL profile was tested by executing both its plan and the original plan
and measuring their respective execution statistics. A plan may have been
only partially executed if the other could be run to completion in less time.

      Original Plan  With SQL Profile  % Improved
      -----
Completion Status:      COMPLETE      COMPLETE
Elapsed Time (s):      .00014      .000017      87.85 %
CPU Time (s):          .0001      .0001      0 %
User I/O Time (s):      0      0
Buffer Gets:           22      2      90.9 %
Physical Read Requests: 0      0
Physical Write Requests: 0      0
Physical Read Bytes:    0      0
Physical Write Bytes:   0      0
Rows Processed:         1      1
Fetches:                1      1
Executions:             1      1

Notes
-----
1. Statistics for the original plan were averaged over 10 executions.
2. Statistics for the SQL profile plan were averaged over 10 executions.
```

```
EXPLAIN PLANS SECTION
-----
1- Original with Adjusted Cost
-----
Plan hash value: 1357081020

-----
| Id | Operation          | Name | Rows | Bytes | Cost (%CPU)| Time     |
-----
|  0 | SELECT STATEMENT   |      |    1 |    4 |    6  (0)| 00:00:01 |
|*  1 |  TABLE ACCESS FULL| TEST |    1 |    4 |    6  (0)| 00:00:01 |
-----

Predicate Information (identified by operation id):
-----

   1 - filter("ID1"=1)

2- Using SQL Profile
-----
Plan hash value: 2882402178

-----
| Id | Operation          | Name      | Rows | Bytes | Cost (%CPU)| Time     |
-----
|  0 | SELECT STATEMENT   |           |    1 |    4 |    1  (0)| 00:00:01 |
|*  1 |  INDEX RANGE SCAN  | TEST_IDX  |    1 |    4 |    1  (0)| 00:00:01 |
-----

Predicate Information (identified by operation id):
-----

   1 - access("ID1"=1)
```



# SQL PROFILE 的管理(利用 SQL Tuning Advisor)

```
SQL>
SQL> exec DBMS_SQLTUNE.ACCEPT_SQL_PROFILE(task_name=> 'my_sql_tuning_task_1',name=> 'my_sql_profile',replace=>true);
PL/SQL procedure successfully completed.

SQL> col created for a30
SQL> col last_modified for a30
SQL> set linesize 200
SQL> select name,created,last_modified,status from dba_sql_profiles;

my_sql_profile          07-JAN-17 11.48.31.000000 AM    07-JAN-17 11.48.31.000000 AM    ENABLED

Execution Plan
```

```
SQL> set autotrace on
SQL> select /*+ no_index(test test_idx) */ * from test where id1=1;
```

1

Execution Plan

Plan hash value: 2882402178

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	4	1 (0)	00:00:01
* 1	INDEX RANGE SCAN	TEST_IDX	1	4	1 (0)	00:00:01

Predicate Information (identified by operation id):

1 - access("ID1"=1)

Note

- SQL profile "my\_sql\_profile" used for this statement

# SQL PROFILE 的管理(HINTS和OUTLINE区别)

- 1、HINTS是ORACLE7引入的
- 2、HINTS是特殊的注释，传递指令给ORACLE优化器
- 3、HINTS使用在UPDATE,INSERT,SELECT,DELETE,MERGE语句后面
- 4、HINTS具体来说就是注释里面添加如下关键字LEADING,FULL,INDEX如右图

```
zls@rac>explain plan set statement id='myprofile~'  
2      for select /*+ full(test)*/ * from test where id1='1';  
Explained.
```

OUTLINE是存放在ORACLE内部的特定格式的HINTS，可以在plan\_table,v\$sql\_plan,dba\_hist\_sql\_plan视图的OTHER\_XML字段查看,具体格式化可以通过dbms\_xplan.display,或者是dbms\_xplan.display\_cursor查看OUTLINE信息，如下图

```
zls@rac>Select * From Table(dbms_xplan.display(Null,Null,'outline',Null));  
PLAN_TABLE_OUTPUT  
-----  
-----  
Plan hash value: 1357081020  
  
-----  
| Id | Operation          | Name | Rows  | Bytes | Cost (%CPU)| Time     |  
-----  
|*  0 | SELECT STATEMENT    |      |    1  |    5  |     6   (0)| 00:00:01 |  
|*  1 |   TABLE ACCESS FULL| TEST |    1  |    5  |     6   (0)| 00:00:01 |  
-----  
  
Outline Data  
-----  
/*+  
  BEGIN_OUTLINE_DATA  
  FULL(@"SEL$1" "TEST"@"SEL$1")  
  OUTLINE_LEAF(@"SEL$1")  
  ALL_ROWS  
  DB_VERSION('11.2.0.4')  
  OPTIMIZER_FEATURES_ENABLE('11.2.0.4')  
  IGNORE_OPTIM_EMBEDDED_HINTS  
  END_OUTLINE_DATA  
*/  
  
Predicate Information (identified by operation id):  
-----
```

# SQL PROFILE 的管理(利用已存在的正确执行计划)

```
zls@RAC>select sql_fulltext,sql_id,child_number,buffer_gets/executions from v$sql t where t.SQL_FULLTEXT like 'select * from test where%';
```

SQL_FULLTEXT	SQL_ID	CHILD_NUMBER	BUFFER_GETS/EXECUTIONS
select * from test where id1='1'	dt7bmn8qhc6vc	0	3
select * from test where id1='1'	dt7bmn8qhc6vc	1	41

```
zls@RAC>select * from table(dbms_xplan.display_cursor('dt7bmn8qhc6vc',0));
```

```
PLAN_TABLE_OUTPUT
```

```
SQL_ID dt7bmn8qhc6vc, child number 0
```

```
select * from test where id1='1'
```

```
Plan hash value: 2882402178
```

Id	Operation	Name
0	SELECT STATEMENT	
* 1	INDEX RANGE SCAN	TEST_IDX

```
Predicate Information (identified by operation id):
```

```
1 - access("ID1"='1')
```



# SQL PROFILE 的管理(利用已存在的正确执行计划)

```
zls@RAC>select * from table(dbms_xplan.display_cursor('dt7bmn8qhc6vc',1));
```

PLAN\_TABLE\_OUTPUT

SQL\_ID dt7bmn8qhc6vc, child number 1

select \* from test where id1='1'

Plan hash value: 1357081020

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT				6 (100)	
* 1	TABLE ACCESS FULL	TEST	1	5	6 (0)	00:00:01

Predicate Information (identified by operation id):

1 - filter("ID1"='1')

18 rows selected.

18 rows selected.

```
zls@RAC>declare
2   ar_profile_hints sys.sqlprof_attr;
3   cl_sql_text      clob;
4   l_profile_name    varchar2(30) := 'P_dt7bmn8qhc6vc'; ---定义profile名字
5   sqlid             varchar2(30) := 'dt7bmn8qhc6vc'; --sql_id
6   childno           Number(1) := 0; --正确的执行计划的child_number
7 begin
8   select extractvalue(value(d), '/hint') as outline_hints
9     bulk collect
10    into ar_profile_hints
11    from xmltable('/*outline_data/hint' passing
12                (select xmltype(other_xml) as xmlval
13                 from v$sql_plan
14                 where sql_id = sqlid
15                  and child_number = childno
16                  and other_xml is not null)) d;
17
18   select sql_fulltext
19     into cl_sql_text
20    from v$sql
21   where sql_id = sqlid
22         and child_number = childno;
23
24   dbms_sqltune.import_sql_profile(sql_text => cl_sql_text,
25                                   profile => ar_profile_hints,
26                                   --category => 'DEFAULT',
27                                   name      => l_profile_name,
28                                   force_match => true,
29                                   replace  => true);
30
31
32 end;
33 /
```

PL/SQL procedure successfully completed.

# SQL PROFILE 的管理(利用已存在的正确执行计划)

```
PL/SQL procedure successfully completed.
```

```
zls@RAC>  
zls@RAC>explain plan for  
2 select * from test where id1='1';
```

```
Explained.
```

```
zls@RAC>select * from table(dbms_xplan.display);
```

```
PLAN_TABLE_OUTPUT
```

```
-----  
-----  
-----  
Plan hash value: 2882402178
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	5	1 (0)	00:00:01
* 1	INDEX RANGE SCAN	TEST_IDX	1	5	1 (0)	00:00:01

```
-----  
Predicate Information (identified by operation id):
```

```
-----  
1 - access("ID1"='1')
```

```
Note
```

```
-----  
- SQL profile "P_dt7bmn8qhc6vc" used for this statement
```

```
17 rows selected.
```

```
zls@RAC>
```



# SQL PROFILE 的管理(利用HINTS手动生存正确执行计划)

```
zls@RAC>Delete Delete From plan_table;
```

```
2 rows deleted.
```

```
zls@RAC>commit;
```

```
Commit complete.
```

```
zls@RAC>explain plan set statement_id='myprofile~'  
2      for select /*+ full(test)*/ * from test where id1='1';
```

```
Explained.
```

```
zls@RAC>Select * From Table(dbms_xplan.display(Null,Null,'outline',Null));
```

```
PLAN_TABLE OUTPUT
```

```
-----  
Plan hash value: 1357081020
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	5	6 (0)	00:00:01
* 1	TABLE ACCESS FULL	TEST	1	5	6 (0)	00:00:01

# SQL PROFILE 的管理(利用HINTS手动生存正确执行计划)

```
zls@RAC>select plan_id,statement_id From plan_table where statement_id='myprofile~' and rownum=1;
```

```
PLAN_ID STATEMENT_ID
-----
42 myprofile~
```

```
zls@RAC>
```

```
zls@RAC>declare
2  ar_profile_hints sys.sqlprof_attr;
3  cl_sql_text      clob;
4  l_profile_name   varchar2(30) := 'P_dt7bmn8qhc6vc';
5  sqlid            varchar2(30) := 'dt7bmn8qhc6vc';
6  planid           Number(38, 0) := 42;
7  begin
8  select extractvalue(value(d), '/hint') as outline_hints
9  bulk collect
10 into ar_profile_hints
11 from xmltable('/*outline_data/hint' passing
12              (select xmltype(other_xml) as xmlval
13               from plan_table
14                where plan_id = planid
15                 and other_xml is not null)) d;
16
17 select sql_fulltext
18 into cl_sql_text
19 from v$sql
20 where sql_id = sqlid
21        and Rownum = 1;
22
23 dbms_sqltune.import_sql_profile(sql_text => cl_sql_text,
24                                profile => ar_profile_hints,
25                                --category => 'DEFAULT',
26                                name      => l_profile_name,
27                                force_match => true,
28                                replace  => true);
29
30 end;
31 /
```

```
PL/SQL procedure successfully completed.
```

```
zls@RAC>explain plan for
2  select  * from test where id1='1';
```

```
Explained.
```

```
zls@RAC>select * from table(dbms_xplan.display);
```

```
PLAN_TABLE_OUTPUT
```

```
Plan hash value: 1357081020
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	5	6 (0)	00:00:01
* 1	TABLE ACCESS FULL	TEST	1	5	6 (0)	00:00:01

```
Predicate Information (identified by operation id):
```

```
1 - filter("ID1"='1')
```

```
Note
```

```
SQL profile "P_dt7bmn8qhc6vc" used for this statement
```

```
17 rows selected.
```

```
17 rows selected.
```

```
zls@RAC>
```

# SQL PROFILE 的管理（直接利用Optimizer OutLine Data）

```
SQL> select sql_id,sql_fulltext,child_number from v$sql t where t.sql_fulltext like 'select * from test where id1=1';
```

SQL_ID	SQL_FULLTEXT	CHILD_NUMBER
8x04a31mrw52h	select * from test where id1=1	0

  

```
SQL> select * from table(dbms_xplan.display_cursor('8x04a31mrw52h',0,'outline'));
```

PLAN\_TABLE\_OUTPUT

SQL\_ID 8x04a31mrw52h, child number 0

select \* from test where id1=1

Plan hash value: 2882402178

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT				1 (100)	
* 1	INDEX RANGE SCAN	TEST_IDX	1	4	1 (0)	00:00:01

PLAN\_TABLE\_OUTPUT

Outline Data

```
/*+
  BEGIN_OUTLINE_DATA
  IGNORE_OPTIM_EMBEDDED_HINTS
  OPTIMIZER_FEATURES_ENABLE('11.2.0.4')
  DB_VERSION('11.2.0.4')
  ALL_ROWS
*/
```

PLAN\_TABLE\_OUTPUT

```
OUTLINE_LEAF(@"SEL$1")
INDEX(@"SEL$1" "TEST"@"SEL$1" ("TEST"."ID1"))
END_OUTLINE_DATA
*/
```

```
SQL> declare
2  v_hints sys.sqlprof_attr;
  v_sql_text clob;
begin
  select sql_fulltext
    into v_sql_text
    from v$sqlm = 1;x04a31mrw52h'1
   where sql_id = '8x04a31mrw52h'
     and rownum = 1;
  8  9  10  v_hints := sys.sqlprof_attr(q'[BEGIN_OUTLINE_DATA]',
11                                     q'[IGNORE_OPTIM_EMBEDDED_HINTS]',
                                     q'[OPTIMIZER_FEATURES_ENABLE('11.2.0.4')]','),'),
                                     q'[DB_VERSION('11.2.0.4')]',
13  14                                     q'[ALL_ROWS]',
                                     q'[OUTLINE_LEAF(@"SEL$1")]',
                                     q'[FULL(@"SEL$1" "TEST"@"SEL$1")]',
                                     q'[END_OUTLINE_DATA]');
  dbms_sqltune.import_sql_profile(v_sql_text,rw52h',q1_text,
                                v_hints,
                                'P_8x04a31mrw52h',
19  20  21                                force_match => true);
end;
23 /
```

PL/SQL procedure successfully completed.

```
SQL> select name,created,sql_text from dba_sql_profiles;
```

NAME	CREATED	SQL_TEXT
P_8x04a31mrw52h	07-JAN-17 12.06.42.000000 PM	select * from test where id1=1
my_sql_profile	07-JAN-17 11.48.31.000000 AM	select /*+ no_index(test test_idx) */ * from test where id1=1

SQL>



# SQL PROFILE 的管理（直接利用Optimizer OutLine Data）

```
SQL> explain plan for
2 select * from test where id1=1;
Explained.

SQL> select * from table(dbms_xplan.display(null,null,'outline'));

PLAN_TABLE_OUTPUT
-----
Plan hash value: 1357081020

| Id | Operation          | Name | Rows  | Bytes | Cost (%CPU)| Time     |
|----|-----|-----|-----|-----|-----|-----|
| 0  | SELECT STATEMENT    |      |      |      |      |      |
|* 1 | TABLE ACCESS FULL | TEST |    1  |    4  |    7   (0) | 00:00:01 |

Outline Data
-----

PLAN_TABLE_OUTPUT
-----

/*+
  BEGIN_OUTLINE_DATA
  FULL(@"SEL$1" "TEST"@"SEL$1")
  OUTLINE_LEAF(@"SEL$1")
  ALL_ROWS
  DB_VERSION('11.2.0.4')
  OPTIMIZER_FEATURES_ENABLE('11.2.0.4')
  IGNORE_OPTIM_EMBEDDED_HINTS
  END_OUTLINE_DATA
*/

PLAN_TABLE_OUTPUT
-----

Predicate Information (identified by operation id):
-----

   1 - filter("ID1"=1)

Note
-----
   - SQL profile 'P_8x04a31mrw52h' used for this statement
```

# SQL PROFILE 的管理

删除

```
begin  
DBMS_SQLTUNE.DROP_SQL_PROFILE(name => 'my_sql_profile');  
end;  
/
```

# SQL PROFILE 的管理

修改

```
BEGIN
DBMS_SQLTUNE.ALTER_SQL_PROFILE(
name => 'my_sql_profile',
attribute_name => 'STATUS',
value => 'DISABLED');
END;
/
```

# SQL PROFILE 的管理

## SQL PROFILE迁移

### 1、创建STAGE表

```
begin  
DBMS_SQLTUNE.CREATE_STGTAB_SQLPROF(table_name=>'STAGE',schema_name=>'SGPM');  
end;
```

### 2、打包

```
Begin for i in (select t.NAME from dba_sql_profiles t) loop  
DBMS_SQLTUNE.PACK_STGTAB_SQLPROF (staging_table_name  
=>'STAGE',profile_name=>i.name);  
end loop;  
end;
```

# SQL PROFILE 的管理

SQL PROFILE 迁移

3、exp sgpm/xxx tables=STAGE

4、imp sgpm/xxx tables=STAGE

5、解包

```
begin
DBMS_SQLTUNE.UNPACK_STGTAB_SQLPROF(replace => TRUE,staging_table_name => 'STAGE');
end;
```

6、验证SQL PROFILE



# 案例分享 ( PROFILE的使用 )

- 1、创建表记录数10W条
- 2、在STATUS列上面创建普通索引
- 3、收集统计信息，收集列的柱状图
- 4、STATUS列值分布不均且有2个唯一值

```
zls@RAC>CREATE TABLE test
2 AS
3 SELECT ROWNUM id,
4        DBMS_RANDOM.STRING('A', 12) name,
5        DECODE(MOD(ROWNUM, 500), 0, 'Inactive', 'Active') status
6        FROM all_objects a,dba_objects b
7        WHERE ROWNUM <= 100000;

Table created.

zls@RAC>zls@RAC>create index t_ind on test(status);

Index created.

zls@RAC>begin
2  dbms_stats.gather_table_stats(ownname      => 'ZLS',
3                                tabname       => 'TEST',
4                                no_invalidate => FALSE,
5                                estimate_percent => 100,
6                                force         => true,
7                                degree        => 5,
8                                method_opt    => 'for all columns size 2',--收集柱状图
9                                cascade       => true);
10 end;
11 /

PL/SQL procedure successfully completed.

zls@RAC>select status,count(*) from test group by status;

STATUS          COUNT(*)
-----
Active          99800
Inactive        200
```

# 案例分享 ( PROFILE的使用 ) 11G环境

```
zls@ORCL>var a varchar2(100);
zls@ORCL>exec :a := 'Inactive';

PL/SQL procedure successfully completed.

zls@ORCL>select count(name) from test where status=:a;

COUNT(NAME)
-----
          200

zls@ORCL>select sql_id,
2         child_number,
3         sql_fulltext,
4         executions,
5         is_bind_aware,
6         is_shareable
7         from v$sql t
8         where t.sql_fulltext like 'select count(name) from test where status=:a%';

SQL_ID          CHILD_NUMBER  SQL_FULLTEXT                                EXECUTIONS  I  I
-----
faubs3h6pubdv          0 select count(name) from test where status=:a              1  N  Y

zls@ORCL>select * from table(dbms_xplan.display_cursor('faubs3h6pubdv',0))
2 ;

PLAN_TABLE_OUTPUT
-----
SQL_ID faubs3h6pubdv, child number 0
select count(name) from test where status=:a
Plan hash value: 4130896540

| Id | Operation                                | Name | Rows | Bytes | Cost (%CPU)| Time |
|----|-----|-----|-----|-----|-----|-----|
| 0  | SELECT STATEMENT                        |      |      |      | 3 (100)|      |
| 1  |   SORT AGGREGATE                        |      |    1 |    21 |          |      |
| 2  |    TABLE ACCESS BY INDEX ROWID        | TEST |   199 |  4179 |  3 (0)| 00:00:01 |
|* 3  |      INDEX RANGE SCAN                   | T_IND |   199 |      |  1 (0)| 00:00:01 |

Predicate Information (identified by operation id):
-----
3 - access("STATUS"=:A)
```

```
zls@ORCL>exec :a := 'Active';

PL/SQL procedure successfully completed.

zls@ORCL>select count(name) from test where status=:a;

COUNT(NAME)
-----
       99800

zls@ORCL>select sql_id,
2         child_number,
3         sql_fulltext,
4         executions,
5         is_bind_aware,
6         is_shareable
7         from v$sql t
8         where t.sql_fulltext like 'select count(name) from test where status=:a%';

SQL_ID          CHILD_NUMBER  SQL_FULLTEXT                                EXECUTIONS  I  I
-----
faubs3h6pubdv          0 select count(name) from test where status=:a              2  N  Y

zls@ORCL>select count(name) from test where status=:a;

COUNT(NAME)
-----
       99800

zls@ORCL>select sql_id,
2         child_number,
3         sql_fulltext,
4         executions,
5         is_bind_aware,
6         is_shareable
7         from v$sql t
8         where t.sql_fulltext like 'select count(name) from test where status=:a%';

SQL_ID          CHILD_NUMBER  SQL_FULLTEXT                                EXECUTIONS  I  I
-----
faubs3h6pubdv          0 select count(name) from test where status=:a              2  N  N
faubs3h6pubdv          1 select count(name) from test where status=:a              1  Y  Y

zls@ORCL>select * from table(dbms_xplan.display_cursor('faubs3h6pubdv',1))
2 ;
```

# 案例分享 ( PROFILE的使用 ) 11G环境

```
zls@ORCL>select * from table(dbms_xplan.display_cursor('faubs3h6pubdv',1))  
2 ;
```

PLAN\_TABLE\_OUTPUT

SQL\_ID faubs3h6pubdv, child number 1

select count(name) from test where status=:a

Plan hash value: 1950795681

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT				119 (100)	
1	<del>SORT AGGREGATE</del>		1	21		
* 2	TABLE ACCESS FULL	TEST	99800	2046K	119 (1)	00:00:02

Predicate Information (identified by operation id):

2 - filter("STATUS"=:A)



# 案例分享 ( PROFILE的使用 ) bind\_aware

```
zls@RAC>var a varchar2(100)
zls@RAC>exec :a := 'Active';

PL/SQL procedure successfully completed.

zls@RAC>select /*+ bind_aware */count(name) from test where status=:a;

COUNT(NAME)
-----
99800

zls@RAC>select * from table(dbms_xplan.display_cursor('frmtct00s6xyp'));

PLAN_TABLE_OUTPUT
-----
SQL_ID frmtct00s6xyp, child number 0
-----
select /*+ bind_aware */count(name) from test where status=:a
Plan hash value: 1950795681

| Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time |
|----|-----|-----|-----|-----|-----|-----|
| 0 | SELECT STATEMENT | | | | | |
| 1 | SORT AGGREGATE | | 1 | 21 | 97 (100) | |
|* 2 | TABLE ACCESS FULL | TEST | 99800 | 2046K | 97 (2) | 00:00:02 |

Predicate Information (identified by operation id):
-----
2 - filter("STATUS"=:A)

19 rows selected.

zls@RAC>select sql_id,child_number from v$sql t where sql_id='frmtct00s6xyp';

SQL_ID CHILD_NUMBER
-----
frmtct00s6xyp 0
```

```
zls@RAC>exec :a := 'Inactive';

PL/SQL procedure successfully completed.

zls@RAC>select /*+ bind_aware */count(name) from test where status=:a;

COUNT(NAME)
-----
200

zls@RAC>select sql_id,child_number from v$sql t where sql_id='frmtct00s6xyp';

SQL_ID CHILD_NUMBER
-----
frmtct00s6xyp 0
frmtct00s6xyp 1

zls@RAC>select * from table(dbms_xplan.display_cursor('frmtct00s6xyp',1));

PLAN_TABLE_OUTPUT
-----
SQL_ID frmtct00s6xyp, child number 1
-----
select /*+ bind_aware */count(name) from test where status=:a
Plan hash value: 4130896540

| Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time |
|----|-----|-----|-----|-----|-----|-----|
| 0 | SELECT STATEMENT | | | | | |
| 1 | SORT AGGREGATE | | 1 | 21 | 3 (100) | |
| 2 | TABLE ACCESS BY INDEX ROWID | TEST | 199 | 4179 | 3 (0) | 00:00:01 |
|* 3 | INDEX RANGE SCAN | T_IND | 199 | | 1 (0) | 00:00:01 |

Predicate Information (identified by operation id):
-----
3 - access("STATUS"=:A)

20 rows selected.
```

# 案例分享（ PROFILE的使用 ） profile使用

```
zls@RAC>exec :a := 'Active';
PL/SQL procedure successfully completed.
zls@RAC>select count(name) from test where status=:a;
COUNT(NAME)
-----
99800
1 row selected.
zls@RAC>select * from table(dbms_xplan.display_cursor('faubs3h6pubdv'));
PLAN_TABLE_OUTPUT
-----
SQL_ID faubs3h6pubdv, child number 0
-----
select count(name) from test where status=:a
Plan hash value: 1950795681

| Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time |
|----|-----|-----|-----|-----|-----|-----|
| 0 | SELECT STATEMENT | | | | | |
| 1 | SORT AGGREGATE | | 1 | 21 | 97 (100) | |
|* 2 | TABLE ACCESS FULL | TEST | 99800 | 2046K | 97 (2) | 00:00:02 |

Predicate Information (identified by operation id):
-----
2 - filter("STATUS"=:A)
Note
-----
- SQL profile profile_faubs3h6pubdv_dwrose used for this statement
23 rows selected.
```

```
zls@RAC>exec :a := 'Inactive';
PL/SQL procedure successfully completed.
zls@RAC>select count(name) from test where status=:a;
COUNT(NAME)
-----
200
1 row selected.
zls@RAC>select * from table(dbms_xplan.display_cursor('faubs3h6pubdv',1));
PLAN_TABLE_OUTPUT
-----
SQL_ID faubs3h6pubdv, child number 1
-----
select count(name) from test where status=:a
Plan hash value: 4130896540

| Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time |
|----|-----|-----|-----|-----|-----|-----|
| 0 | SELECT STATEMENT | | | | | |
| 1 | SORT AGGREGATE | | 1 | 21 | 3 (100) | |
| 2 | TABLE ACCESS BY INDEX ROWID | TEST | 199 | 4179 | 3 (0) | 00:00:01 |
|* 3 | INDEX RANGE SCAN | T_IND | 199 | | 1 (0) | 00:00:01 |

Predicate Information (identified by operation id):
-----
3 - access("STATUS"=:A)
Note
-----
- SQL profile profile_faubs3h6pubdv_dwrose used for this statement
24 rows selected.
```



# Let Data Drive!

WOQU TECH all rights reserved.



微信二维码



微博二维码