

【ASH】如何导出视图 DBA_HIST_ACTIVE_SESS_HISTORY 的查询结果数据

1.1 BLOG 文档结构图



1.2 前言部分

1.2.1 导读和注意事项

各位技术爱好者，看完本文后，你可以掌握如下的技能，也可以学到一些其它你所不知道的知识，~o(∩_∩)o~:

- ① 如何导出 ASH 数据--利用 exp 导出基表的数据（重点）
- ② 12c 的 expdp 参数 VIEWS_AS_TABLES 选项
- ③ expdp 工具不能导出哪些对象？

Tips:

- ① 本文在 itpub (<http://blog.itpub.net/26736162>)、博客园 (<http://www.cnblogs.com/lhrbest/>) 和微信公众号 (xiaomaimiaolhr) 上有同步更新。
- ② 文章中用到的所有代码、相关软件、相关资料及本文的 pdf 版本都请前往小麦苗的云盘下载，小麦苗的云盘地址见: <http://blog.itpub.net/26736162/viewspace-1624453/>。
- ③ 若网页文章代码格式有错乱，请下载 pdf 格式的文档来阅读。
- ④ 在本篇 BLOG 中，代码输出部分一般放在一行一列的表格中。

本文若有错误或不完善的地方请大家多多指正，您的批评指正是我写作的最大动力。

1.2.2 相关文章链接

12c 的 dmp 文件导入 11g 中参考：【故障处理】IMP-00010 错误 12C 的 dmp 文件导入 11G，地址为：
<http://blog.itpub.net/26736162/viewspace-2128197/>

1.2.3 本文简介

众所周知，视图只是一个查询数据的窗口，其不存储数据，所以在使用 exp 等工具导出的时候只能导出其定义，而不能导出视图的查询结果数据。在 Oracle 12c 中，可以采用 expdp 中的一个新增参数 VIEWS_AS_TABLES 来将视图作为表来导出，非常实用，不过对于一些特殊的表仍然不能采用 expdp 导出，例如 SYS 和 SYSTEM 下的一些表，AUD\$ 表不能使用 expdp 来导出。

另外，对于一些安全类很高的系统是不允许随意创建表，也不允许使用 PLSQL Developer 等客户端的工具，那么若是查询 DBA_HIST_ACTIVE_SESS_HISTORY 等视图的时候就非常不方便了，这个时候我们可以将该视图的内容导出来，然后导入到我们自己的测试库中就可以随意的进行分析了。那么，如何来导出这些数据的内容呢？本文将详细介绍这些内容。

1.3 如何导出 ash 数据？

根据前边的分析，我们知道视图的查询结果数据不能直接导出，那么我们可以导出这个视图的基表数据：

```
SELECT D.NAME, D.TYPE, D.REFERENCED_NAME, D.REFERENCED_TYPE
FROM DBA_DEPENDENCIES D
WHERE D.NAME IN ('DBA_HIST_ACTIVE_SESS_HISTORY',
                'DBA_HIST_PLAN_OPERATION_NAME',
                'DBA_HIST_PLAN_OPTION_NAME',
                'DBA_HIST_SQLCOMMAND_NAME',
                'DBA_HIST_TOPLEVELCALL_NAME')
AND D.TYPE = 'VIEW'
ORDER BY D.NAME, D.REFERENCED_NAME;
```

	NAME	TYPE	REFERENCED_NAME	REFERENCED_TYPE
1	DBA_HIST_ACTIVE_SESS_HISTORY	VIEW	DBA_HIST_PLAN_OPERATION_NAME	VIEW
2	DBA_HIST_ACTIVE_SESS_HISTORY	VIEW	DBA_HIST_PLAN_OPTION_NAME	VIEW
3	DBA_HIST_ACTIVE_SESS_HISTORY	VIEW	DBA_HIST_SQLCOMMAND_NAME	VIEW
4	DBA_HIST_ACTIVE_SESS_HISTORY	VIEW	DBA_HIST_TOPLEVELCALL_NAME	VIEW
5	DBA_HIST_ACTIVE_SESS_HISTORY	VIEW	WRH\$_ACTIVE_SESSION_HISTORY	TABLE
6	DBA_HIST_ACTIVE_SESS_HISTORY	VIEW	WRH\$_EVENT_NAME	TABLE
7	DBA_HIST_ACTIVE_SESS_HISTORY	VIEW	WRM\$_SNAPSHOT	TABLE
8	DBA_HIST_ACTIVE_SESS_HISTORY	VIEW	X\$MODACT_LENGTH	TABLE
9	DBA_HIST_PLAN_OPERATION_NAME	VIEW	WRH\$_PLAN_OPERATION_NAME	TABLE
10	DBA_HIST_PLAN_OPTION_NAME	VIEW	WRH\$_PLAN_OPTION_NAME	TABLE
11	DBA_HIST_SQLCOMMAND_NAME	VIEW	WRH\$_SQLCOMMAND_NAME	TABLE
12	DBA_HIST_TOPLEVELCALL_NAME	VIEW	WRH\$_TOPLEVELCALL_NAME	TABLE

主要涉及的表是图中方框里的去掉 x\$ 表后的 7 个表，其中最主要的还是 WRH\$_ACTIVE_SESSION_HISTORY 表，该表是一个分区表，导出的时候可以按照时间进行导出。其它表都是很小的表，可以全量导出。

下面尝试使用 exp 和 expdp 来导出。

1.3.1 expdp 导出 sys 用户下的表报错 ORA-39165 和 ORA-39166

```
[oracle@orcltest ~]$ more /tmp/expdp_ash_lhr_01.par
query=SYS.WRH$_ACTIVE_SESSION_HISTORY:"WHERE SAMPLE_TIME BETWEEN TO_DATE('2016-12-02 08:30:00',
'YYYY-MM-DD HH24:MI:SS') AND TO_DATE('2016-12-08 23:38:00', 'YYYY-MM-DD HH24:MI:SS')"
[oracle@orcltest ~]$
[oracle@orcltest ~]$ expdp \'/ AS SYSDBA\' directory=DATA_PUMP_DIR
tables='SYS.WRH$_ACTIVE_SESSION_HISTORY','SYS.WRM$_SNAPSHOT','SYS.WRH$_EVENT_NAME','SYS.WRH$_SQLCOMMAND
_NAME','SYS.WRH$_PLAN_OPERATION_NAME','SYS.WRH$_PLAN_OPTION_NAME','SYS.WRH$_TOPLEVELCALL_NAME'
dumpfile=expdp_ash_lhr_01.dmp parfile=/tmp/expdp_ash_lhr_01.par EXCLUDE=STATISTICS VERSION=11.2.0.1
REUSE_DUMPFILES=Y

Export: Release 11.2.0.3.0 - Production on Fri Dec 16 16:49:52 2016

Copyright (c) 1982, 2011, Oracle and/or its affiliates. All rights reserved.

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, Automatic Storage Management, OLAP, Data Mining
and Real Application Testing options
Starting "SYS"."SYS_EXPORT_TABLE_03": "/***** AS SYSDBA" directory=DATA_PUMP_DIR
tables=SYS.WRH$_ACTIVE_SESSION_HISTORY,SYS.WRM$_SNAPSHOT,SYS.WRH$_EVENT_NAME,SYS.WRH$_SQLCOMMAND_NAME,S
YS.WRH$_PLAN_OPERATION_NAME,SYS.WRH$_PLAN_OPTION_NAME,SYS.WRH$_TOPLEVELCALL_NAME
dumpfile=expdp_ash_lhr_01.dmp parfile=/tmp/expdp_ash_lhr_01.par EXCLUDE=STATISTICS VERSION=11.2.0.1
REUSE_DUMPFILES=Y
Estimate in progress using BLOCKS method...
Total estimation using BLOCKS method: 0 KB
ORA-39166: Object SYS.WRH$_ACTIVE_SESSION_HISTORY was not found.
ORA-39166: Object SYS.WRM$_SNAPSHOT was not found.
ORA-39166: Object SYS.WRH$_EVENT_NAME was not found.
ORA-39166: Object SYS.WRH$_SQLCOMMAND_NAME was not found.
ORA-39166: Object SYS.WRH$_PLAN_OPERATION_NAME was not found.
ORA-39166: Object SYS.WRH$_PLAN_OPTION_NAME was not found.
ORA-39166: Object SYS.WRH$_TOPLEVELCALL_NAME was not found.
ORA-31655: no data or metadata objects selected for job
Job "SYS"."SYS_EXPORT_TABLE_03" completed with 8 error(s) at 16:49:53
```

查询 MOS:

DataPump Export (EXPDP) Fails With Error ORA-39165: Schema SYS Was Not Found (文档 ID 553402.1)

该文章给出了如下答案:

1. There is a restriction on dataPump export. It cannot export schemas like SYS, ORDSYS, EXFSYS, MDSYS, DMSYS, CTXSYS, ORDPLUGINS, LBACSYS, XDB, SI_INFORMTN_SCHEMA, DIP, DBSNMP and WMSYS in any mode.
2. The Utilities Guide indicates the restriction only on full export mode, but the restriction actually applies to all modes.



DataPump Export (EXPDP) Fails With Error ORA-39165 Schema SYS Was Not Found (文档 ID 553402.1).mhtml

而: MOS: Why Can an Object Not Be Exported? Expdp of SYSTEM User's Table Returns ORA-39166 or ORA-31655 (文档 ID 2114233.1) 列出来了哪些对象不能导出:

Objects (tables, views, schemas, etc) which fall under either of below conditions are not exported with expdp because they are regarded as system maintained objects.

Object is listed in ku_noexp_view.

This view is a union of ku_noexp_tab and noexp\$ tables.

Objects that are listed in this view are not exported.

Object is ORACLE_MAINTAINED='Y' in ALL_OBJECTS (and DBA_OBJECTS).----针对 12c

在视图 sys.Ku_Noexp_View 中或 DBA_OBJECTS 的 ORACLE_MAINTAINED 列为 Y 的对象不能导出。

```
SELECT * FROM sys.Ku_Noexp_View d WHERE d.name LIKE '%WRH%';
```

```
SELECT * FROM DBA_OBJECTS d WHERE d.ORACLE_MAINTAINED='Y' AND D.object_name LIKE 'WR%';
```

解决该报错的方法是：

1. 使用 exp 导出

2.ctas 的方法在不受限制的 schema 下创建表，然后导出该新建的表

3. use the DBMS_AUDIT_MGMT package of Audit Vault to manage and purge audit data (see Note 731908.1). This allows for the facility to move the AUD\$ table out of the SYSTEM tablespace, which can negate the need to export the table.

注意：This issue also applies to other SYS owned auditing tables such as FGA_LOG\$

1.3.2 采用 exp 导出 ASH 数据

1.3.2.1 方法 1: ctas 建表导出，有的客户不让建表

```
CREATE TABLE ASH_TEMP_20161219 NOLOGGING AS
SELECT *
  FROM DBA_HIST_ACTIVE_SESS_HISTORY D
 WHERE D.SAMPLE_TIME BETWEEN
        TO_DATE('2016-12-01 02:00:00', 'YYYY-MM-DD HH24:MI:SS') AND
        TO_DATE('2016-12-17 06:00:00', 'YYYY-MM-DD HH24:MI:SS')
;

exp  \'/ AS SYSDBA\' tables=ASH_TEMP_20161219 file=/tmp/exp_ASH_TEMP_20161219.dmp
log=/tmp/ASH_TEMP_20161117.log buffer=41943040
imp lhr/lhr tables=ASH_TEMP_20161219 file=/tmp/ASH_TEMP_20161219.dmp
log=/tmp/imp_ASH_TEMP_20161117.log buffer=41943040
```

实验过程：

```
[oracle@orcltest ~]$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.3.0 Production on Mon Dec 19 09:51:09 2016

Copyright (c) 1982, 2011, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, Automatic Storage Management, OLAP, Data Mining
and Real Application Testing options

SYS@lhrdb> CREATE TABLE ASH_TEMP_20161219 NOLOGGING AS
2  SELECT *
3    FROM DBA_HIST_ACTIVE_SESS_HISTORY D
4   WHERE D.SAMPLE_TIME BETWEEN
5          TO_DATE('2016-12-01 02:00:00', 'YYYY-MM-DD HH24:MI:SS') AND
6          TO_DATE('2016-12-17 06:00:00', 'YYYY-MM-DD HH24:MI:SS')
7  ;
```

Table created.

```
SYS@lhrdb> exit
```

Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, Automatic Storage Management, OLAP, Data Mining
and Real Application Testing options

```
[oracle@orcltest ~]$
```

```
[oracle@orcltest ~]$
```

```
[oracle@orcltest ~]$ exp \'/ AS SYSDBA\' tables=ASH_TEMP_20161219 file=/tmp/ASH_TEMP_20161219.dmp  
log=/tmp/ASH_TEMP_20161219.log buffer=41943040
```

Export: Release 11.2.0.3.0 - Production on Mon Dec 19 09:51:44 2016

Copyright (c) 1982, 2011, Oracle and/or its affiliates. All rights reserved.

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, Automatic Storage Management, OLAP, Data Mining
and Real Application Testing options
Export done in ZHS16GBK character set and AL16UTF16 NCHAR character set

About to export specified tables via Conventional Path ...

```
. . exporting table          ASH_TEMP_20161219          102 rows exported
```

Export terminated successfully without warnings.

```
[oracle@orcltest ~]$ imp lhr/lhr tables=ASH_TEMP_20161219 file=/tmp/ASH_TEMP_20161219.dmp  
log=/tmp/ASH_TEMP_20161117.log buffer=41943040
```

Import: Release 11.2.0.3.0 - Production on Mon Dec 19 09:52:20 2016

Copyright (c) 1982, 2011, Oracle and/or its affiliates. All rights reserved.

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, Automatic Storage Management, OLAP, Data Mining
and Real Application Testing options

Export file created by EXPORT:V11.02.00 via conventional path

Warning: the objects were exported by SYS, not by you

import done in ZHS16GBK character set and AL16UTF16 NCHAR character set

```
. importing SYS's objects into LHR
```

```
. importing SYS's objects into LHR
```

```
. . importing table          "ASH_TEMP_20161219"          102 rows imported
```

Import terminated successfully without warnings.

```
[oracle@orcltest ~]$
```

1.3.2.2 方法 2: 导出基表的数据

导出基表数据:

```
---more /tmp/exp_ash_lhr_01.par
```

```
query="WHERE SAMPLE_TIME BETWEEN TO_DATE('2016-12-02 08:30:00', 'YYYY-MM-DD HH24:MI:SS') AND  
TO_DATE('2016-12-08 23:38:00', 'YYYY-MM-DD HH24:MI:SS')"
```

```
exp \'/ AS SYSDBA\' tables='WRH$_ACTIVE_SESSION_HISTORY' file=/tmp/exp_ash_lhr_01.dmp
```

```
parfile=/tmp/exp_ash_lhr_01.par log=/tmp/exp_ash_lhr_01.log GRANTS=N CONSTRAINTS=N STATISTICS=NONE
```

```
exp \'/ AS SYSDBA\'
```

```
tables='WRM$_SNAPSHOT','WRH$_EVENT_NAME','WRH$_SQLCOMMAND_NAME','WRH$_PLAN_OPERATION_NAME','WRH$_PLAN_O  
PTION_NAME','WRH$_TOPLEVELCALL_NAME' file=/tmp/exp_ash_lhr_02.dmp log=/tmp/exp_ash_lhr_02.log GRANTS=N
```



```
CONSTRAINTS=N STATISTICS=NONE
```

导入到测试用户:

```
imp lhr/lhr file=/tmp/exp_ash_lhr_01.dmp tables='WRH$_ACTIVE_SESSION_HISTORY' log=/tmp/imp_ash_lhr_01.log
FROMUSER=SYS TOUSER=LHR
imp lhr/lhr file=/tmp/exp_ash_lhr_02.dmp
tables='WRM$_SNAPSHOT','WRH$_EVENT_NAME','WRH$_SQLCOMMAND_NAME','WRH$_PLAN_OPERATION_NAME','WRH$_PLAN_O
PTION_NAME','WRH$_TOPLEVELCALL_NAME' log=/tmp/imp_ash_lhr_02.log FROMUSER=SYS TOUSER=LHR
```

```
DROP TABLE LHR.WRH$_ACTIVE_SESSION_HISTORY PURGE;
DROP TABLE LHR.WRM$_SNAPSHOT PURGE;
DROP TABLE LHR.WRH$_EVENT_NAME PURGE;
DROP TABLE LHR.WRH$_SQLCOMMAND_NAME PURGE;
DROP TABLE LHR.WRH$_PLAN_OPERATION_NAME PURGE;
DROP TABLE LHR.WRH$_PLAN_OPTION_NAME PURGE;
DROP TABLE LHR.WRH$_TOPLEVELCALL_NAME PURGE;
```

实验:

```
[oracle@orcltest ~]$ more /tmp/exp_ash_lhr_01.par
query="WHERE SAMPLE TIME BETWEEN TO_DATE('2016-12-02 08:30:00', 'YYYY-MM-DD HH24:MI:SS') AND
TO_DATE('2016-12-08 23:38:00', 'YYYY-MM-DD HH24:MI:SS')"
[oracle@orcltest ~]$ exp \'/ AS SYSDBA\' tables='WRH$_ACTIVE_SESSION_HISTORY' file=/tmp/exp_ash_lhr_01.dmp
parfile=/tmp/exp_ash_lhr_01.par log=/tmp/exp_ash_lhr_01.log GRANTS=N CONSTRAINTS=N STATISTICS=NONE
```

Export: Release 11.2.0.3.0 - Production on Mon Dec 19 10:25:05 2016

Copyright (c) 1982, 2011, Oracle and/or its affiliates. All rights reserved.

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, Automatic Storage Management, OLAP, Data Mining
and Real Application Testing options
Export done in ZHS16GBK character set and AL16UTF16 NCHAR character set
Note: grants on tables/views/sequences/roles will not be exported
Note: constraints on tables will not be exported

About to export specified tables via Conventional Path ...

```
. . exporting table      WRH$_ACTIVE_SESSION_HISTORY
. . exporting partition      WRH$_ACTIVE_971836524_0          55 rows exported
. . exporting partition      WRH$_ACTIVE_971836524_35          0 rows exported
. . exporting partition      WRH$_ACTIVE_SES_MXDB_MXSN          0 rows exported
```

Export terminated successfully without warnings.

```
[oracle@orcltest ~]$ exp \'/ AS SYSDBA\'
tables='WRM$_SNAPSHOT','WRH$_EVENT_NAME','WRH$_SQLCOMMAND_NAME','WRH$_PLAN_OPERATION_NAME','WRH$_PLAN_O
PTION_NAME','WRH$_TOPLEVELCALL_NAME' file=/tmp/exp_ash_lhr_02.dmp log=/tmp/exp_ash_lhr_02.log GRANTS=N
CONSTRAINTS=N STATISTICS=NONE
```

Export: Release 11.2.0.3.0 - Production on Mon Dec 19 10:25:12 2016

Copyright (c) 1982, 2011, Oracle and/or its affiliates. All rights reserved.

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, Automatic Storage Management, OLAP, Data Mining
and Real Application Testing options
Export done in ZHS16GBK character set and AL16UTF16 NCHAR character set

Note: grants on tables/views/sequences/roles will not be exported

Note: constraints on tables will not be exported

About to export specified tables via Conventional Path ...

. . exporting table	WRM\$ SNAPSHOT	23 rows exported
. . exporting table	WRH\$ EVENT NAME	1152 rows exported
. . exporting table	WRH\$ SQLCOMMAND NAME	165 rows exported
. . exporting table	WRH\$ PLAN OPERATION NAME	130 rows exported
. . exporting table	WRH\$ PLAN OPTION NAME	165 rows exported
. . exporting table	WRH\$ TOPLEVELCALL NAME	151 rows exported

Export terminated successfully without warnings.

```
[oracle@orcltest ~]$ imp lhr/lhr file=/tmp/exp_ash_lhr_01.dmp tables='WRH$ ACTIVE SESSION HISTORY'
log=/tmp/imp_ash_lhr_01.log FROMUSER=SYS TOUSER=LHR
```

Import: Release 11.2.0.3.0 - Production on Mon Dec 19 10:28:37 2016

Copyright (c) 1982, 2011, Oracle and/or its affiliates. All rights reserved.

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, Automatic Storage Management, OLAP, Data Mining
and Real Application Testing options

Export file created by EXPORT:V11.02.00 via conventional path

Warning: the objects were exported by SYS, not by you

import done in ZHS16GBK character set and AL16UTF16 NCHAR character set

. importing SYS's objects into LHR	
. . importing partition "WRH\$ ACTIVE SESSION HISTORY":"WRH\$ ACTIVE_971836524_0"	55 rows imported
. . importing partition "WRH\$ ACTIVE SESSION HISTORY":"WRH\$ ACTIVE_971836524_35"	0 rows imported
. . importing partition "WRH\$ ACTIVE SESSION HISTORY":"WRH\$ ACTIVE_SES_MXDB_MXSN"	0 rows imported

Import terminated successfully without warnings.

```
[oracle@orcltest ~]$
```

```
[oracle@orcltest ~]$
```

```
[oracle@orcltest ~]$ imp lhr/lhr file=/tmp/exp_ash_lhr_02.dmp
tables='WRM$ SNAPSHOT','WRH$ EVENT_NAME','WRH$ SQLCOMMAND_NAME','WRH$ PLAN_OPERATION_NAME','WRH$ PLAN_O
PTION_NAME','WRH$ TOPLEVELCALL_NAME' log=/tmp/imp_ash_lhr_02.log FROMUSER=SYS TOUSER=LHR
```

Import: Release 11.2.0.3.0 - Production on Mon Dec 19 10:27:26 2016

Copyright (c) 1982, 2011, Oracle and/or its affiliates. All rights reserved.

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, Automatic Storage Management, OLAP, Data Mining
and Real Application Testing options

Export file created by EXPORT:V11.02.00 via conventional path

Warning: the objects were exported by SYS, not by you

import done in ZHS16GBK character set and AL16UTF16 NCHAR character set

. importing SYS's objects into LHR	
. . importing table "WRM\$ SNAPSHOT"	23 rows imported
. . importing table "WRH\$ EVENT_NAME"	1152 rows imported
. . importing table "WRH\$ SQLCOMMAND_NAME"	165 rows imported
. . importing table "WRH\$ PLAN_OPERATION_NAME"	130 rows imported
. . importing table "WRH\$ PLAN_OPTION_NAME"	165 rows imported
. . importing table "WRH\$ TOPLEVELCALL_NAME"	151 rows imported

Import terminated successfully without warnings.

```
[oracle@orcltest ~]$
```

接下来就是根据这些基表来创建自己的视图了，不再详述。

1.3.3 12c expdp VIEWS_AS_TABLES 选项

expdp VIEWS_AS_TABLES 选项可以将视图看做表并将其数据导出。

```
expdp system/lhr DIRECTORY=data_pump_dir DUMPFILE=expdp_vw.dmp LOGFILE=expdp_vw.log
VIEWS_AS_TABLES=lhr.my_view
```

表数据准备：

```
create table lhr.my_tab1 (nr number, txt varchar2(10));
insert into lhr.my_tab1 values (1,'Line 1');
insert into lhr.my_tab1 values (2,'Line 2');

create table lhr.my_tab2 (nr number, col2 number, col3 varchar2(10));
insert into lhr.my_tab2 values (1,1,'c3_1');
insert into lhr.my_tab2 values (2,2,'c3_2');
commit;
create view lhr.my_view (nr, txt, col3) as
  select t1.nr, t1.txt, t2.col3
  from lhr.my_tab1 t1, lhr.my_tab2 t2
 where t1.nr=t2.nr;
```

开始导出：

```
C:\Users\xiaomaimiao>expdp system/lhr DIRECTORY=data_pump_dir DUMPFILE=expdp_vw.dmp LOGFILE=expdp_vw.log
VIEWS_AS_TABLES=lhr.my_view
```

Export: Release 12.1.0.2.0 - Production on 星期五 12月 16 16:31:49 2016

Copyright (c) 1982, 2014, Oracle and/or its affiliates. All rights reserved.

连接到: Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production

With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

启动 "SYSTEM"."SYS_EXPORT_TABLE_01": system/***** DIRECTORY=data_pump_dir DUMPFILE=expdp_vw.dmp
LOGFILE=expdp_vw.log VIEWS_AS_TABLES=lhr.my_view

正在使用 BLOCKS 方法进行估计...

处理对象类型 TABLE_EXPORT/VIEWS_AS_TABLES/TABLE_DATA

使用 BLOCKS 方法的总估计: 16 KB

处理对象类型 TABLE_EXPORT/VIEWS_AS_TABLES/TABLE

. . 导出了 "LHR"."MY_VIEW" 5.929 KB 2 行

已成功加载/卸载了主表 "SYSTEM"."SYS_EXPORT_TABLE_01"

SYSTEM.SYS_EXPORT_TABLE_01 的转储文件集为:

E:\APP\ORACLE\ADMIN\LHRDB12C\DPDUMP\EXPDP_VW.DMP

作业 "SYSTEM"."SYS_EXPORT_TABLE_01" 已于 星期五 12月 16 16:32:36 2016 elapsed 0 00:00:31 成功完成

查看其 DDL 语句：

```
C:\Users\xiaomaimiao>impdp system/lhr DIRECTORY=data_pump_dir DUMPFILE=expdp_vw.dmp LOGFILE=impdp_vw.log
sqlfile=a.txt
```


Import: Release 12.1.0.2.0 - Production on 星期五 12月 16 16:35:14 2016

Copyright (c) 1982, 2014, Oracle and/or its affiliates. All rights reserved.

连接到: Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options
已成功加载/卸载了主表 "SYSTEM"."SYS SQL FILE FULL 01"

启动 "SYSTEM"."SYS SQL FILE FULL 01": system/***** DIRECTORY=data pump dir DUMPFILE=expdp vw.dmp
LOGFILE=impdp vw.log sqlfile=a.txt

处理对象类型 TABLE EXPORT/VIEWS AS TABLES/TABLE

作业 "SYSTEM"."SYS SQL FILE FULL 01" 已于 星期五 12月 16 16:35:26 2016 elapsed 0 00:00:10 成功完成

DDL 语句内容:

```
-- CONNECT SYSTEM
ALTER SESSION SET EVENTS '10150 TRACE NAME CONTEXT FOREVER, LEVEL 1';
ALTER SESSION SET EVENTS '10904 TRACE NAME CONTEXT FOREVER, LEVEL 1';
ALTER SESSION SET EVENTS '25475 TRACE NAME CONTEXT FOREVER, LEVEL 1';
ALTER SESSION SET EVENTS '10407 TRACE NAME CONTEXT FOREVER, LEVEL 1';
ALTER SESSION SET EVENTS '10851 TRACE NAME CONTEXT FOREVER, LEVEL 1';
ALTER SESSION SET EVENTS '22830 TRACE NAME CONTEXT FOREVER, LEVEL 192 ';
-- new object type path: TABLE_EXPORT/VIEWS_AS_TABLES/TABLE
CREATE TABLE "LHR"."MY_VIEW"
  ("NR" NUMBER,
   "TXT" VARCHAR2(10 BYTE),
   "COL3" VARCHAR2(10 BYTE)
  ) SEGMENT CREATION DEFERRED
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
  NOCOMPRESS LOGGING
  TABLESPACE "USERS" ;
```

进行导入:

C:\Users\xiaomaimiao>sqlplus / as sysdba

SQL*Plus: Release 12.1.0.2.0 Production on 星期五 12月 16 16:37:03 2016

Copyright (c) 1982, 2014, Oracle. All rights reserved.

连接到:

Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

SQL> create user lhr01 identified by lhr;

用户已创建。

SQL> grant dba to lhr01;

授权成功。

SQL> exit

从 Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options 断开

C:\Users\xiaomaimiao>impdp system/lhr DIRECTORY=data_pump_dir DUMPFILE=expdp_vw.dmp LOGFILE=impdp_vw.log
remap_schema=lhr:lhr01

Import: Release 12.1.0.2.0 - Production on 星期五 12月 16 16:39:49 2016

Copyright (c) 1982, 2014, Oracle and/or its affiliates. All rights reserved.

连接到: Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options
已成功加载/卸载了主表 "SYSTEM"."SYS_IMPORT_FULL_02"

启动 "SYSTEM"."SYS_IMPORT_FULL_02": system/***** DIRECTORY=data pump dir DUMPFILE=expdp vw.dmp
LOGFILE=impdp vw.log remap schema=lhr:lhr01

处理对象类型 TABLE EXPORT/VIEWS AS TABLES/TABLE

处理对象类型 TABLE EXPORT/VIEWS AS TABLES/TABLE DATA

. . 导入了 "LHR01"."MY VIEW" 5.929 KB 2 行

作业 "SYSTEM"."SYS_IMPORT_FULL_02" 已于 星期五 12 月 16 16:39:57 2016 elapsed 0 00:00:06 成功完成

1.4 小结

--- 方法 1: ctas 建表导出 有的客户不让建表

```
CREATE TABLE ASH_TEMP_20161117 NOLOGGING AS
SELECT *
  FROM DBA_HIST_ACTIVE_SESS_HISTORY D
 WHERE D.SAMPLE_TIME BETWEEN
        TO_DATE('2016-11-10 02:00:00', 'YYYY-MM-DD HH24:MI:SS') AND
        TO_DATE('2016-11-17 06:00:00', 'YYYY-MM-DD HH24:MI:SS')
;
```

```
exp \'/ AS SYSDBA\' tables=ASH_TEMP_20161117 file=/tmp/ASH_TEMP_20161117.dmp
log=/tmp/ASH_TEMP_20161117.log buffer=41943040
```

--- 方法 2: 导出基表的数据

采用 exp 导出 ASH 数据的命令:

```
---more /tmp/exp_ash_lhr_01.par
query="WHERE SAMPLE_TIME BETWEEN TO_DATE('2016-12-02 08:30:00', 'YYYY-MM-DD HH24:MI:SS') AND
TO_DATE('2016-12-08 23:38:00', 'YYYY-MM-DD HH24:MI:SS')"
```

```
exp \'/ AS SYSDBA\' tables='WRH$_ACTIVE_SESSION_HISTORY' file=/tmp/exp_ash_lhr_01.dmp
parfile=/tmp/exp_ash_lhr_01.par log=/tmp/exp_ash_lhr_01.log GRANTS=N CONSTRAINTS=N STATISTICS=NONE
exp \'/ AS SYSDBA\'
tables='WRM$_SNAPSHOT','WRH$_EVENT_NAME','WRH$_SQLCOMMAND_NAME','WRH$_PLAN_OPERATION_NAME','WRH$_PLAN_O
PTION_NAME','WRH$_TOPLEVELCALL_NAME' file=/tmp/exp_ash_lhr_02.dmp log=/tmp/exp_ash_lhr_02.log GRANTS=N
CONSTRAINTS=N STATISTICS=NONE
```

```
imp lhr/lhr file=/tmp/exp_ash_lhr_01.dmp tables='WRH$_ACTIVE_SESSION_HISTORY' log=/tmp/imp_ash_lhr_01.log
FROMUSER=SYS TOUSER=LHR
imp lhr/lhr file=/tmp/exp_ash_lhr_02.dmp
tables='WRM$_SNAPSHOT','WRH$_EVENT_NAME','WRH$_SQLCOMMAND_NAME','WRH$_PLAN_OPERATION_NAME','WRH$_PLAN_O
PTION_NAME','WRH$_TOPLEVELCALL_NAME' log=/tmp/imp_ash_lhr_02.log FROMUSER=SYS TOUSER=LHR
```

```
DROP TABLE LHR.WRH$_ACTIVE_SESSION_HISTORY PURGE;
DROP TABLE LHR.WRM$_SNAPSHOT PURGE;
DROP TABLE LHR.WRH$_EVENT_NAME PURGE;
DROP TABLE LHR.WRH$_SQLCOMMAND_NAME PURGE;
DROP TABLE LHR.WRH$_PLAN_OPERATION_NAME PURGE;
```

```
DROP TABLE LHR.WRH$_PLAN_OPTION_NAME PURGE;
DROP TABLE LHR.WRH$_TOPLEVELCALL_NAME PURGE;
```

创建自己的 ASH 视图:

```
---- 11.2.0.3
create or replace view dh_ash_11g_lhr
(snap_id, dbid, instance_number, sample_id, sample_time, session_id, session_serial#, session_type, flags,
user_id, sql_id, is_sqlid_current, sql_child_number, sql_opcode, sql_opname, force_matching_signature,
top_level_sql_id, top_level_sql_opcode, sql_plan_hash_value, sql_plan_line_id, sql_plan_operation,
sql_plan_options, sql_exec_id, sql_exec_start, plsql_entry_object_id, plsql_entry_subprogram_id,
plsql_object_id, plsql_subprogram_id, qc_instance_id, qc_session_id, qc_session_serial#, px_flags, event,
event_id, seq#, p1text, p1, p2text, p2, p3text, p3, wait_class, wait_class_id, wait_time, session_state,
time_waited, blocking_session_status, blocking_session, blocking_session_serial#, blocking_inst_id,
blocking_hangchain_info, current_obj#, current_file#, current_block#, current_row#, top_level_call#,
top_level_call_name, consumer_group_id, xid, remote_instance#, time_model, in_connection_mgmt, in_parse,
in_hard_parse, in_sql_execution, in_plsql_execution, in_plsql_rpc, in_plsql_compilation,
in_java_execution, in_bind, in_cursor_close, in_sequence_load, capture_overhead, replay_overhead,
is_captured, is_replayed, service_hash, program, module, action, client_id, machine, port, ecid,
dbreplay_file_id, dbreplay_call_counter, tm_delta_time, tm_delta_cpu_time, tm_delta_db_time, delta_time,
delta_read_io_requests, delta_write_io_requests, delta_read_io_bytes, delta_write_io_bytes,
delta_interconnect_io_bytes, pga_allocated, temp_space_allocated)
as
select /* ASH/AWR meta attributes */
    ash.snap_id, ash.dbid, ash.instance_number,
    ash.sample_id, ash.sample_time,
    /* Session/User attributes */
    ash.session_id, ash.session_serial#,
    decode(ash.session_type, 1, 'BACKGROUND', 'BACKGROUND'),
    ash.flags,
    ash.user_id,
    /* SQL attributes */
    ash.sql_id,
    decode(bitand(ash.flags, power(2, 4)), NULL, 'N', 0, 'N', 'Y'),
    ash.sql_child_number, ash.sql_opcode,
    (select command_name from WRH$_SQLCOMMAND_NAME
     where command_type = ash.sql_opcode
     and dbid = ash.dbid) as sql_opname,
    ash.force_matching_signature,
    decode(ash.top_level_sql_id, NULL, ash.sql_id, ash.top_level_sql_id),
    decode(ash.top_level_sql_id, NULL, ash.sql_opcode,
           ash.top_level_sql_opcode),
    /* SQL Plan/Execution attributes */
    ash.sql_plan_hash_value,
    decode(ash.sql_plan_line_id, 0, to_number(NULL), ash.sql_plan_line_id),
    (select operation_name from WRH$_PLAN_OPERATION_NAME
     where operation_id = ash.sql_plan_operation#
     and dbid = ash.dbid) as sql_plan_operation,
    (select option_name from WRH$_PLAN_OPTION_NAME
     where option_id = ash.sql_plan_options#
     and dbid = ash.dbid) as sql_plan_options,
    decode(ash.sql_exec_id, 0, to_number(NULL), ash.sql_exec_id),
    ash.sql_exec_start,
    /* PL/SQL attributes */
    decode(ash.plsql_entry_object_id, 0, to_number(NULL),
           ash.plsql_entry_object_id),
    decode(ash.plsql_entry_object_id, 0, to_number(NULL),
           ash.plsql_entry_subprogram_id),
    decode(ash.plsql_object_id, 0, to_number(NULL),
           ash.plsql_object_id),
```

```
decode(ash.plsql_object_id,0,to_number(NULL),
      ash.plsql_subprogram_id),
/* PQ attributes */
decode(ash.qc_session_id, 0, to_number(NULL), ash.qc_instance_id),
decode(ash.qc_session_id, 0, to_number(NULL), ash.qc_session_id),
decode(ash.qc_session_id, 0, to_number(NULL), ash.qc_session_serial#),
decode(ash.px_flags,      0, to_number(NULL), ash.px_flags),
/* Wait event attributes */
decode(ash.wait_time, 0, evt.event_name, NULL),
decode(ash.wait_time, 0, evt.event_id,  NULL),
ash.seq#,
evt.parameter1, ash.p1,
evt.parameter2, ash.p2,
evt.parameter3, ash.p3,
decode(ash.wait_time, 0, evt.wait_class,  NULL),
decode(ash.wait_time, 0, evt.wait_class_id, NULL),
ash.wait_time,
decode(ash.wait_time, 0, 'WAITING', 'ON CPU'),
ash.time_waited,
(case when ash.blocking_session = 4294967295
      then 'UNKNOWN'
      when ash.blocking_session = 4294967294
      then 'GLOBAL'
      when ash.blocking_session = 4294967293
      then 'UNKNOWN'
      when ash.blocking_session = 4294967292
      then 'NO HOLDER'
      when ash.blocking_session = 4294967291
      then 'NOT IN WAIT'
      else 'VALID'
end),
(case when ash.blocking_session between 4294967291 and 4294967295
      then to_number(NULL)
      else ash.blocking_session
end),
(case when ash.blocking_session between 4294967291 and 4294967295
      then to_number(NULL)
      else ash.blocking_session_serial#
end),
(case when ash.blocking_session between 4294967291 and 4294967295
      then to_number(NULL)
      else ash.blocking_inst_id
end),
(case when ash.blocking_session between 4294967291 and 4294967295
      then NULL
      else decode(bitand(ash.flags, power(2, 3)), NULL, 'N',
                  0, 'N', 'Y')
end),
/* Session's working context */
ash.current_obj#, ash.current_file#, ash.current_block#,
ash.current_row#, ash.top_level_call#,
(select top_level_call_name from WRH$_TOPLEVELCALL_NAME
 where top_level_call# = ash.top_level_call#
 and dbid = ash.dbid) as top_level_call_name,
decode(ash.consumer_group_id, 0, to_number(NULL),
      ash.consumer_group_id),
ash.xid,
decode(ash.remote_instance#, 0, to_number(NULL), ash.remote_instance#),
ash.time_model,
decode(bitand(ash.time_model,power(2, 3)),0,'N','Y')
      as in_connection_mgmt,
decode(bitand(ash.time_model,power(2, 4)),0,'N','Y')as in_parse,
decode(bitand(ash.time_model,power(2, 7)),0,'N','Y')as in_hard_parse,
decode(bitand(ash.time_model,power(2,10)),0,'N','Y')as in_sql_execution,
```

```

decode(bitand(ash.time_model,power(2,11)),0,'N','Y')
                                as in_plsql_execution,
decode(bitand(ash.time_model,power(2,12)),0,'N','Y')as in_plsql_rpc,
decode(bitand(ash.time_model,power(2,13)),0,'N','Y')
                                as in_plsql_compilation,
decode(bitand(ash.time_model,power(2,14)),0,'N','Y')
                                as in_java_execution,
decode(bitand(ash.time_model,power(2,15)),0,'N','Y')as in_bind,
decode(bitand(ash.time_model,power(2,16)),0,'N','Y')as in_cursor_close,
decode(bitand(ash.time_model,power(2,17)),0,'N','Y')as in_sequence_load,
decode(bitand(ash.flags,power(2,5)),NULL,'N',0,'N','Y')
                                as capture_overhead,
decode(bitand(ash.flags,power(2,6)), NULL,'N',0,'N','Y' )
                                as replay_overhead,
decode(bitand(ash.flags,power(2,0)),NULL,'N',0,'N','Y') as is_captured,
decode(bitand(ash.flags,power(2,2)), NULL,'N',0,'N','Y' )as is_replayed,
/* Application attributes */
ash.service_hash, ash.program,
ash.module module,
ash.action action,
ash.client_id,
ash.machine, ash.port, ash.ecid,
/* DB Replay info */
ash.dbreplay_file_id, ash.dbreplay_call_counter,
/* stash columns */
ash.tm_delta_time,
ash.tm_delta_cpu_time,
ash.tm_delta_db_time,
ash.delta_time,
ash.delta_read_io_requests,
ash.delta_write_io_requests,
ash.delta_read_io_bytes,
ash.delta_write_io_bytes,
ash.delta_interconnect_io_bytes,
ash.pga_allocated,
ash.temp_space_allocated
from WRM$_SNAPSHOT sn, WRH$_ACTIVE_SESSION_HISTORY ash, WRH$_EVENT_NAME evt
where      ash.snap_id          = sn.snap_id(+)
and ash.dbid                    = sn.dbid(+)
and ash.instance_number = sn.instance_number(+)
and ash.dbid                = evt.dbid
and ash.event_id             = evt.event_id;

----- 12c
create or replace view dh_ash_12c_lhr
(snap_id, dbid, instance_number, sample_id, sample_time, session_id, session_serial#, session_type, flags,
user_id, sql_id, is_sqlid_current, sql_child_number, sql_opcode, sql_opname, force_matching_signature,
top_level_sql_id, top_level_sql_opcode, sql_plan_hash_value, sql_full_plan_hash_value,
sql_adaptive_plan_resolved, sql_plan_line_id, sql_plan_operation, sql_plan_options, sql_exec_id,
sql_exec_start, plsql_entry_object_id, plsql_entry_subprogram_id, plsql_object_id, plsql_subprogram_id,
qc_instance_id, qc_session_id, qc_session_serial#, px_flags, event, event_id, seq#, pltext, p1, p2text, p2,
p3text, p3, wait_class, wait_class_id, wait_time, session_state, time_waited, blocking_session_status,
blocking_session, blocking_session_serial#, blocking_inst_id, blocking_hangchain_info, current_obj#,
current_file#, current_block#, current_row#, top_level_call#, top_level_call_name, consumer_group_id, xid,
remote_instance#, time_model, in_connection_mgmt, in_parse, in_hard_parse, in_sql_execution,
in_plsql_execution, in_plsql_rpc, in_plsql_compilation, in_java_execution, in_bind, in_cursor_close,
in_sequence_load, in_inmemory_query, in_inmemory_populate, in_inmemory_prepopulate,
in_inmemory_repopulate, in_inmemory_trepopulate, capture_overhead, replay_overhead, is_captured,
is_replayed, service_hash, program, module, action, client_id, machine, port, ecid, dbreplay_file_id,
dbreplay_call_counter, tm_delta_time, tm_delta_cpu_time, tm_delta_db_time, delta_time,
delta_read_io_requests, delta_write_io_requests, delta_read_io_bytes, delta_write_io_bytes,
delta_interconnect_io_bytes, pga_allocated, temp_space_allocated, dbop_name, dbop_exec_id, con_dbid/*,
con_id*/)
as

```

```
select /* ASH/AWR meta attributes */
      ash.snap_id, ash.dbid, ash.instance_number,
      ash.sample_id, ash.sample_time,
      /* Session/User attributes */
      ash.session_id, ash.session_serial#,
      decode(ash.session_type, 1, 'BACKGROUND', 'BACKGROUND'),
      ash.flags,
      ash.user_id,
      /* SQL attributes */
      ash.sql_id,
      decode(bitand(ash.flags, power(2, 4)), NULL, 'N', 0, 'N', 'Y'),
      ash.sql_child_number, ash.sql_opcode,
      (select command_name
        from WRH$_SQLCOMMAND_NAME s
       where s.command_type = ash.sql_opcode
             and s.dbid = ash.dbid
             and s.con_dbid = ash.dbid) as sql_opname,
      ash.force_matching_signature,
      decode(ash.top_level_sql_id, NULL, ash.sql_id, ash.top_level_sql_id),
      decode(ash.top_level_sql_id, NULL, ash.sql_opcode,
             ash.top_level_sql_opcode),
      /* SQL Plan/Execution attributes */
      ash.sql_plan_hash_value,
      ash.sql_full_plan_hash_value,
      ash.sql_adaptive_plan_resolved,
      decode(ash.sql_plan_line_id, 0, to_number(NULL), ash.sql_plan_line_id),
      (select operation_name
        from WRH$_PLAN_OPERATION_NAME pn
       where pn.operation_id = ash.sql_plan_operation#
             and pn.dbid = ash.dbid
             and pn.con_dbid = ash.dbid) as sql_plan_operation,
      (select option_name
        from WRH$_PLAN_OPTION_NAME po
       where po.option_id = ash.sql_plan_options#
             and po.dbid = ash.dbid
             and po.con_dbid = ash.dbid) as sql_plan_options,
      decode(ash.sql_exec_id, 0, to_number(NULL), ash.sql_exec_id),
      ash.sql_exec_start,
      /* PL/SQL attributes */
      decode(ash.plsql_entry_object_id, 0, to_number(NULL),
             ash.plsql_entry_object_id),
      decode(ash.plsql_entry_object_id, 0, to_number(NULL),
             ash.plsql_entry_subprogram_id),
      decode(ash.plsql_object_id, 0, to_number(NULL),
             ash.plsql_object_id),
      decode(ash.plsql_object_id, 0, to_number(NULL),
             ash.plsql_subprogram_id),
      /* PQ attributes */
      decode(ash.qc_session_id, 0, to_number(NULL), ash.qc_instance_id),
      decode(ash.qc_session_id, 0, to_number(NULL), ash.qc_session_id),
      decode(ash.qc_session_id, 0, to_number(NULL), ash.qc_session_serial#),
      decode(ash.px_flags, 0, to_number(NULL), ash.px_flags),
      /* Wait event attributes */
      decode(ash.wait_time, 0, evt.event_name, NULL),
      decode(ash.wait_time, 0, evt.event_id, NULL),
      ash.seq#,
      evt.parameter1, ash.p1,
      evt.parameter2, ash.p2,
      evt.parameter3, ash.p3,
      decode(ash.wait_time, 0, evt.wait_class, NULL),
      decode(ash.wait_time, 0, evt.wait_class_id, NULL),
      ash.wait_time,
      decode(ash.wait_time, 0, 'WAITING', 'ON CPU'),
      ash.time_waited,
```



```
(case when ash.blocking_session = 4294967295
      then 'UNKNOWN'
      when ash.blocking_session = 4294967294
      then 'GLOBAL'
      when ash.blocking_session = 4294967293
      then 'UNKNOWN'
      when ash.blocking_session = 4294967292
      then 'NO HOLDER'
      when ash.blocking_session = 4294967291
      then 'NOT IN WAIT'
      else 'VALID'
end),
(case when ash.blocking_session between 4294967291 and 4294967295
      then to_number(NULL)
      else ash.blocking_session
end),
(case when ash.blocking_session between 4294967291 and 4294967295
      then to_number(NULL)
      else ash.blocking_session_serial#
end),
(case when ash.blocking_session between 4294967291 and 4294967295
      then to_number(NULL)
      else ash.blocking_inst_id
end),
(case when ash.blocking_session between 4294967291 and 4294967295
      then NULL
      else decode(bitand(ash.flags, power(2, 3)), NULL, 'N',
                  0, 'N', 'Y')
end),
/* Session's working context */
ash.current_obj#, ash.current_file#, ash.current_block#,
ash.current_row#, ash.top_level_call#,
(select top_level_call_name
 from WRH$_TOPLEVELCALL_NAME t
 where top_level_call# = ash.top_level_call#
       and t.dbid = ash.dbid
       and t.con_dbid = ash.dbid) as top_level_call_name,
decode(ash.consumer_group_id, 0, to_number(NULL),
       ash.consumer_group_id),
ash.xid,
decode(ash.remote_instance#, 0, to_number(NULL), ash.remote_instance#),
ash.time_model,
decode(bitand(ash.time_model, power(2, 3)), 0, 'N', 'Y')
      as in_connection_mgmt,
decode(bitand(ash.time_model, power(2, 4)), 0, 'N', 'Y') as in_parse,
decode(bitand(ash.time_model, power(2, 7)), 0, 'N', 'Y') as in_hard_parse,
decode(bitand(ash.time_model, power(2, 10)), 0, 'N', 'Y') as in_sql_execution,
decode(bitand(ash.time_model, power(2, 11)), 0, 'N', 'Y')
      as in_plsql_execution,
decode(bitand(ash.time_model, power(2, 12)), 0, 'N', 'Y') as in_plsql_rpc,
decode(bitand(ash.time_model, power(2, 13)), 0, 'N', 'Y')
      as in_plsql_compilation,
decode(bitand(ash.time_model, power(2, 14)), 0, 'N', 'Y')
      as in_java_execution,
decode(bitand(ash.time_model, power(2, 15)), 0, 'N', 'Y') as in_bind,
decode(bitand(ash.time_model, power(2, 16)), 0, 'N', 'Y') as in_cursor_close,
decode(bitand(ash.time_model, power(2, 17)), 0, 'N', 'Y') as in_sequence_load,
decode(bitand(ash.time_model, power(2, 18)), 0, 'N', 'Y') as in_inmemory_query,
decode(bitand(ash.time_model, power(2, 19)), 0, 'N', 'Y')
      as in_inmemory_populate,
decode(bitand(ash.time_model, power(2, 20)), 0, 'N', 'Y')
      as in_inmemory_prepopulate,
decode(bitand(ash.time_model, power(2, 21)), 0, 'N', 'Y')
      as in_inmemory_repopulate,
```

```
decode(bitand(ash.time_model,power(2,22)),0,'N','Y')
                                as in_inmemory_trepopulate,
decode(bitand(ash.flags,power(2,5)),NULL,'N',0,'N','Y')
                                as capture_overhead,
decode(bitand(ash.flags,power(2,6)), NULL,'N',0,'N','Y' )
                                as replay_overhead,
decode(bitand(ash.flags,power(2,0)),NULL,'N',0,'N','Y') as is_captured,
decode(bitand(ash.flags,power(2,2)), NULL,'N',0,'N','Y' )as is_replayed,
/* Application attributes */
ash.service_hash, ash.program,
ash.module module,
ash.action action,
ash.client_id,
ash.machine, ash.port, ash.ecid,
/* DB Replay info */
ash.dbreplay_file_id, ash.dbreplay_call_counter,
/* stash columns */
ash.tm_delta_time,
ash.tm_delta_cpu_time,
ash.tm_delta_db_time,
ash.delta_time,
ash.delta_read_io_requests,
ash.delta_write_io_requests,
ash.delta_read_io_bytes,
ash.delta_write_io_bytes,
ash.delta_interconnect_io_bytes,
ash.pga_allocated,
ash.temp_space_allocated,
ash.dbop_name,
ash.dbop_exec_id,
decode(ash.con_dbid, 0, ash.dbid, ash.con_dbid)/*,
con_dbid_to_id(decode(ash.con_dbid, 0, ash.dbid, ash.con_dbid)) con_id*/
from WRM$_SNAPSHOT sn, WRH$_ACTIVE_SESSION_HISTORY ash, WRH$_EVENT_NAME evt
where      ash.snap_id      = sn.snap_id(+)
and ash.dbid                = sn.dbid(+)
and ash.instance_number    = sn.instance_number(+)
and ash.dbid                = evt.dbid
and ash.event_id            = evt.event_id;
```

About Me

- 本文作者：小麦苗，只专注于数据库的技术，更注重技术的运用
- 本文在 itpub (<http://blog.itpub.net/26736162>)、博客园 (<http://www.cnblogs.com/lhrbest>) 和个人微信公众号 ([xiaomaimiao1hr](#))
- 本文 itpub 地址： <http://blog.itpub.net/26736162/viewspace-2130980/>
- 本文博客园地址： <http://www.cnblogs.com/lhrbest/p/6200899.html>
- 本文 pdf 版及小麦苗云盘地址： <http://blog.itpub.net/26736162/viewspace-1624453/>
- QQ 群：230161599 微信群：私聊
- 联系我请加 QQ 好友 (642808185)，注明添加缘由
- 于 2016-12-16 09:00 ~ 2016-12-19 23:00 在农行完成
- 文章内容来源于小麦苗的学习笔记，部分整理自网络，若有侵权或不当之处还请谅解
- 版权所有，欢迎分享本文，转载请保留出处

手机长按下图识别二维码或微信客户端扫描下边的二维码来关注小麦苗的微信公众号：xiaomaimiao1hr，免费学习最实用的数据库技术。



小猿图

