

## DUAL 系列

### 1.1 BLOG 文档结构图

└ DUAL 系列
└ 1.1 BLOG 文档结构图
└ 1.2 前言部分
└ 1.2.1 导读和注意事项
└ 1.2.2 相关参考文章链接
└ 1.2.3 本文简介
└ 1.3 dual 官方文档说明
└ 1.4 DUAL 总结
└ 1.5 DUAL 表破坏后的恢复
└ 1.5.1 不重启数据库
└ 1.5.2 重启数据库
└ 1.6 关于 replication_dependency_tracking 参数
└ 第 2 章实验中用到的 SQL 总结
└ About Me

### 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(n\_n)o~：

① DUAL 遭到破坏后的重建（重点）

② 关于参数 replication\_dependency\_tracking 简介

③ DUAL 简介

**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 ( xiaomaimiao1hr ) 有同步更新

② 文章中用到的所有代码，相关软件，相关资料请前往小麦苗的云盘下载

( <http://blog.itpub.net/26736162/viewspace-1624453/> )

③ 若文章代码格式有错乱，推荐使用搜狗、360 或 QQ 浏览器，也可以下载 pdf 格式的文档来查看，pdf 文档

下载地址：<http://blog.itpub.net/26736162/viewspace-1624453/>，另外 itpub 格式显示有问题，可以去博客园地址

阅读

④ 本篇 BLOG 中命令的输出部分需要特别关注的地方我都用灰色背景和粉红色字体来表示，比如下边的例子中，

thread 1 的最大归档日志号为 33，thread 2 的最大归档日志号为 43 是需要特别关注的地方；而命令一般使用黄色背景和红色字体标注；对代码或代码输出部分的注释一般采用蓝色字体表示。

色背景和红色字体标注；对代码或代码输出部分的注释一般采用蓝色字体表示。

```
List of Archived Logs in backup set 11
Thrd Seq      Low SCN      Low Time      Next SCN      Next Time
-----
1      32          1621589      2015-05-29 11:09:52 1625242      2015-05-29 11:15:48
1      33          1625242      2015-05-29 11:15:48 1625293      2015-05-29 11:15:58
2      42          1613951      2015-05-29 10:41:18 1625245      2015-05-29 11:15:49
2      43          1625245      2015-05-29 11:15:49 1625253      2015-05-29 11:15:53

[ZHLHRDB1:root]:/>ls -o
T_XDESK_APP1_vg
rootvg
[ZHLHRDB1:root]:/>
00:27:22 SQL> alter tablespace idxtbs read write;

====> 2097152*512/1024/1024/1024=1G
```

本文如有错误或不完善的地方请大家多多指正，ITPUB 留言或 QQ 皆可，您的批评指正是我写作的最大动力。

## 1.2.2 相关参考文章链接

## 1.2.3 本文简介

DUAL 系列很早之前是看杨建荣的 blog 了解到的，最近又看他的书又看见了这块内容，于是决定自己整理一下。

在此，小麦苗还是想说的是看书和自己做实验是完全不同的 2 回事，不多说了。

### 1.3 dual 官方文档说明

DUAL is a table automatically created by Oracle Database along with the data dictionary. DUAL is in the schema of the user SYS but is accessible by the name DUAL to all users. It has one column, DUMMY, defined to be VARCHAR2(1), and contains one row with a value X. Selecting from the DUAL table is useful for computing a constant expression with the SELECT statement. Because DUAL has only one row, the constant is returned only once. Alternatively, you can select a constant, pseudocolumn, or expression from any table, but the value will be returned as many times as there are rows in the table. Refer to "About SQL Functions" for many examples of selecting a constant value from DUAL.

Note:

Beginning with Oracle Database 10g Release 1, logical I/O is not performed on the DUAL table when computing an expression that does not include the DUMMY column. This optimization is listed as FAST DUAL in the execution plan. If you SELECT the DUMMY column from DUAL, then this optimization does not take place and logical I/O occurs.

### 1.4 DUAL 总结

有关 Oracle 的 DUAL 表，我们需要掌握以下几点：

① DUAL 是系统提供的一个虚表，也常称为 DUMMY 表，临时显示结果的表，里边的内容没有特定的意义，就是为了存在而存在

② DUAL 是属于 SYS SCHEMA 的一个表，然后以 PUBLIC SYNONYM 的方式供其他数据库 USER 使用，这一点可以由如下 SQL 验证：

```
SYS@lhrdb> col owner format a15
SYS@lhrdb> col object_name format a15
SYS@lhrdb> SELECT D.OWNER,D.OBJECT_NAME,D.OBJECT_ID,D.OBJECT_TYPE FROM DBA_OBJECTS D WHERE
D.OBJECT_NAME='DUAL';
```

OWNER	OBJECT_NAME	OBJECT_ID	OBJECT_TYPE
SYS	DUAL	91680	TABLE
PUBLIC	DUAL	117	SYNONYM

③ ORACLE 对 DUAL 表的操作做了一些内部处理，尽量保证 DUAL 表中只返回一条记录，当然这些内部操作是不

可见的。我们尝试往 DUAL 表中插入多条记录，但始终显示的是一条记录。

```
SYS@lhrdb> INSERT INTO DUAL VALUES('X');

1 row created.

SYS@lhrdb> COMMIT;

Commit complete.

SYS@lhrdb> INSERT INTO DUAL VALUES('X');

1 row created.

SYS@lhrdb> COMMIT;

Commit complete.

SYS@lhrdb> select * from dual;

D
-
X
```

④ DUAL 表出现问题，所有相关的基础操作都会受到影响；若是 DUAL 表被删除的话，数据库将不能启动，报 ORA-01775 的错误。

⑤ DUAL 表在数据库启动的不同阶段，里边的字段还是会有一些不同。在数据库 OPEN 状态下是一行一列的表，在 mount 和 nomount 状态下是一行四列的表，验证如下：

```
SYS@lhrdb> startup force nomount
ORACLE instance started.

Total System Global Area 1720328192 bytes
Fixed Size                  2247072 bytes
Variable Size               486540896 bytes
Database Buffers           1224736768 bytes
Redo Buffers                 6803456 bytes
SYS@lhrdb> select * from dual;

ADDR                INDX    INST_ID DU
-----
0000000110299728      0          1 X

SYS@lhrdb> alter database mount;

Database altered.

SYS@lhrdb> select * from dual;

ADDR                INDX    INST_ID D
-----
0000000110299728      0          1 X

SYS@lhrdb> alter database open;

Database altered.

SYS@lhrdb> select * from dual;

D
-
X
```

## ⑥ DUAL 的常用方式

- a. 查看当前连接用户      `SELECT USER FROM DUAL ;`
- b. 查看当前日期、时间    `SELECT TO_CHAR(SYSDATE,'YYYY-MM-DD HH24:MI:SS') FROM DUAL ;`
- c. 当作计算器用        `SELECT 8*9 FROM DUAL ;`
- d. 查看序列值            `SELECT MYSEQ.NEXTVAL FROM DUAL ;`

1.5 DUAL 表破坏后的恢复

项目	db
db 类型	single db
db version	11.2.0.4.0
db 存储	ASM
OS 版本及 kernel 版本	AIX 64 位 7.1.0.0

我们首先删除 SYS 用户下的 DUAL 表：

```
SYS@lhrdb> SHOW USER
USER is "SYS"
SYS@lhrdb> SELECT COUNT(*) FROM DBA_OBJECTS D WHERE D.status='INVALID';

COUNT(*)
-----
         0
SYS@lhrdb> DROP TABLE DUAL;

Table dropped.

SYS@lhrdb> SELECT * FROM DUAL;
SELECT * FROM DUAL
          *
ERROR at line 1:
ORA-01775: looping chain of synonyms

SYS@lhrdb> SELECT COUNT(*) FROM DBA_OBJECTS D WHERE D.status='INVALID';

COUNT(*)
-----
       1042
SYS@lhrdb> SET PAGESIZE 9999
SYS@lhrdb> SELECT D.OWNER, COUNT(*) FROM DBA_OBJECTS D WHERE D.status='INVALID' GROUP BY D.OWNER;

OWNER                                COUNT(*)
-----
PUBLIC                                222
XDB                                    7
OLAPSYS                               43
APEX_030200                           157
SYS                                    403
MDSYS                                  44
SYSMAN                                 66
EXFSYS                                 23
```

```

ORACLE_OCM          1
WMSYS                42
ORDSYS               11
IX                   6
CTXSYS               15
DBSNMP               1
OE                   1

```

```
15 rows selected.
```

```
SYS@lhrdb>
```

可以看到删除 DUAL 表的时候并没有报错,但是查询的时候报错了,并且系统中大约有 1000 多个对象变为了无效的状态,业务系统中的 PL/SQL 代码中含有大量的 DUAL 查询,若是该表被删除,对系统的影响是非常大的。

查看告警日志,也有一些错误报出:

```

Errors in file /oracle/app/oracle/diag/rdbms/lhrdb/lhrdb/trace/lhrdb_m001_41877662.trc:
ORA-00604: error occurred at recursive SQL level 1
ORA-04063: package body "SYS.PRVT_HDM" has errors
ORA-06508: PL/SQL: could not find program unit being called: "SYS.PRVT_HDM"
ORA-06512: at line 1
Wed Aug 10 10:02:40 2016
Errors in file /oracle/app/oracle/diag/rdbms/lhrdb/lhrdb/trace/lhrdb_j000_11075780.trc:
ORA-12012: error on auto execute of job 4002
ORA-01775: looping chain of synonyms

```

DUAL 的恢复分为 2 种情况,一种是数据库在没有重启的情况下,第二种是数据库在重启的情况下,但是总的恢复原则都是在数据库 OPEN 状态下重建 DUAL 表,下边我们分别测试。

### 1.5.1 不重启数据库

不重启数据库的情况下我们可以采用重建 DUAL 表,插入 DUMMY 的数据。

```

SYS@lhrdb> CREATE TABLE SYS.DUAL(DUMMY VARCHAR2(1)) STORAGE(INITIAL 1) TABLESPACE SYSTEM;
Table created.

SYS@lhrdb> INSERT INTO DUAL VALUES('X');
1 row created.

SYS@lhrdb> COMMIT;
Commit complete.

SYS@lhrdb> CREATE OR REPLACE PUBLIC SYNONYM DUAL FOR DUAL;
Synonym created.

SYS@lhrdb> GRANT SELECT ON DUAL TO PUBLIC WITH GRANT OPTION;
Grant succeeded.

SYS@lhrdb> SELECT * FROM DUAL;

D
-
X

SYS@lhrdb> SELECT SYSDATE FROM DUAL;

SYSDATE
-----
2016-08-10 10:22:37

```

```

SYS@lhrdb> SELECT COUNT(*) FROM DBA_OBJECTS D WHERE D.status='INVALID';

COUNT(*)
-----
1041

SYS@lhrdb> SELECT D.OWNER, COUNT(*) FROM DBA_OBJECTS D WHERE D.status='INVALID' GROUP BY D.OWNER;

OWNER                                COUNT(*)
-----
PUBLIC                                222
XDB                                    7
OLAPSYS                               43
APEX_030200                           157
SYS                                    402
MDSYS                                  44
SYSMAN                                 66
EXFSYS                                 23
ORACLE_OCM                             1
WMSYS                                  42
ORDSYS                                 11
IX                                      6
CTXSYS                                 15
DBSNMP                                 1
OE                                      1

15 rows selected.

SYS@lhrdb> col owner format a15
SYS@lhrdb> col object_name format a15
SYS@lhrdb> SELECT D.OWNER,D.OBJECT_NAME,D.OBJECT_ID,D.OBJECT_TYPE FROM DBA_OBJECTS D WHERE
D.OBJECT_NAME='DUAL';

OWNER      OBJECT_NAME      OBJECT_ID OBJECT_TYPE
-----
SYS        DUAL              91680 TABLE
PUBLIC     DUAL              117 SYNONYM

```

可以看到 DUAL 表已经重建成功了，但是还有很多的无效对象，下边我们利用脚本

@?/rdbms/admin/utlrp.sql 来重新编译这些对象。

```

SYS@lhrdb> @?/rdbms/admin/utlrp.sql

TIMESTAMP
-----
COMP_TIMESTAMP UTLRP_BGN 2016-08-10 10:25:13

DOC> The following PL/SQL block invokes UTL_RECOMP to recompile invalid
DOC> objects in the database. Recompilation time is proportional to the
DOC> number of invalid objects in the database, so this command may take
DOC> a long time to execute on a database with a large number of invalid
DOC> objects.
DOC>
DOC> Use the following queries to track recompilation progress:
DOC>
DOC> 1. Query returning the number of invalid objects remaining. This
DOC> number should decrease with time.
DOC> SELECT COUNT(*) FROM obj$ WHERE status IN (4, 5, 6);
DOC>
DOC> 2. Query returning the number of objects compiled so far. This number
DOC> should increase with time.
DOC> SELECT COUNT(*) FROM UTL_RECOMP_COMPILED;
DOC>
DOC> This script automatically chooses serial or parallel recompilation
DOC> based on the number of CPUs available (parameter cpu_count) multiplied
DOC> by the number of threads per CPU (parameter parallel_threads_per_cpu).
DOC> On RAC, this number is added across all RAC nodes.
DOC>
DOC> UTL_RECOMP uses DBMS_SCHEDULER to create jobs for parallel
DOC> recompilation. Jobs are created without instance affinity so that they
DOC> can migrate across RAC nodes. Use the following queries to verify

```

```
DOC> whether UTL_RECOMP jobs are being created and run correctly:
DOC>
DOC> 1. Query showing jobs created by UTL_RECOMP
DOC>      SELECT job_name FROM dba_scheduler_jobs
DOC>           WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>
DOC> 2. Query showing UTL_RECOMP jobs that are running
DOC>      SELECT job_name FROM dba_scheduler_running_jobs
DOC>           WHERE job_name like 'UTL_RECOMP_SLAVE_%';
DOC>#

PL/SQL procedure successfully completed.

TIMESTAMP
-----
COMP_TIMESTAMP UTLRP_END 2016-08-10 10:27:08

DOC> The following query reports the number of objects that have compiled
DOC> with errors.
DOC>
DOC> If the number is higher than expected, please examine the error
DOC> messages reported with each object (using SHOW ERRORS) to see if they
DOC> point to system misconfiguration or resource constraints that must be
DOC> fixed before attempting to recompile these objects.
DOC>#

OBJECTS WITH ERRORS
-----
0

DOC> The following query reports the number of errors caught during
DOC> recompilation. If this number is non-zero, please query the error
DOC> messages in the table UTL_RECOMP_ERRORS to see if any of these errors
DOC> are due to misconfiguration or resource constraints that must be
DOC> fixed before objects can compile successfully.
DOC>#

ERRORS DURING RECOMPILATION
-----
0
```

Function created.

PL/SQL procedure successfully completed.

Function dropped.

PL/SQL procedure successfully completed.

```
SYS@lhrdb> SELECT COUNT(*) FROM DBA_OBJECTS D WHERE D.status='INVALID';

COUNT(*)
-----
0
```

无效对象消失，数据库正常。

## 1.5.2 重启数据库

若在删除了 DUAL 表后我们重启了数据库，就会发现数据库不能正常启动。

```
SYS@lhrdb> DROP TABLE DUAL;

Table dropped.

SYS@lhrdb> SELECT * FROM DUAL;
SELECT * FROM DUAL
```



```
*
ERROR at line 1:
ORA-01775: looping chain of synonyms

SYS@lhrdb> SHUTDOWN IMMEDIATE;
Database closed.
Database dismounted.
ORACLE instance shut down.
SYS@lhrdb>
SYS@lhrdb> STARTUP
ORACLE instance started.

Total System Global Area 1720328192 bytes
Fixed Size                2247072 bytes
Variable Size             486540896 bytes
Database Buffers          1224736768 bytes
Redo Buffers              6803456 bytes
Database mounted.
ORA-01092: ORACLE instance terminated. Disconnection forced
ORA-01775: looping chain of synonyms
Process ID: 42074156
Session ID: 156 Serial number: 5
```

#### ALERT 告警日志：

```
Errors in file /oracle/app/oracle/diag/rdbms/lhrdb/lhrdb/trace/lhrdb_ora_42074156.trc:
ORA-01775: looping chain of synonyms
Errors in file /oracle/app/oracle/diag/rdbms/lhrdb/lhrdb/trace/lhrdb_ora_42074156.trc:
ORA-01775: looping chain of synonyms
Error 1775 happened during db open, shutting down database
USER (ospid: 42074156): terminating the instance due to error 1775
Instance terminated by USER, pid = 42074156
ORA-1092 signalled during: ALTER DATABASE OPEN...
opiodr aborting process unknown ospid (42074156) as a result of ORA-1092
Wed Aug 10 10:36:13 2016
ORA-1092 : opitsk aborting process
```

trace 文件/oracle/app/oracle/diag/rdbms/lhrdb/lhrdb/trace/lhrdb\_ora\_42074156.trc 的

#### 内容：

```
[ZFXDESKDB1:root]:>more /oracle/app/oracle/diag/rdbms/lhrdb/lhrdb/trace/lhrdb_ora_42074156.trc
Trace file /oracle/app/oracle/diag/rdbms/lhrdb/lhrdb/trace/lhrdb_ora_42074156.trc
Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
With the Partitioning, Real Application Clusters, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
ORACLE_HOME = /oracle/app/oracle/product/11.2.0/db
System name:     AIX
Node name:      ZFXDESKDB1
Release:        1
Version:        7
Machine:        00F63A614C00
Instance name:  lhrdb
Redo thread mounted by this instance: 1
Oracle process number: 29
Unix process pid: 42074156, image: oracle@ZFXDESKDB1 (TNS V1-V3)

*** 2016-08-10 10:36:12.598
*** SESSION ID:(156.5) 2016-08-10 10:36:12.598
*** CLIENT ID:() 2016-08-10 10:36:12.598
*** SERVICE NAME:(SYS$USERS) 2016-08-10 10:36:12.598
*** MODULE NAME:(sqlplus@ZFXDESKDB1 (TNS V1-V3)) 2016-08-10 10:36:12.598
*** ACTION NAME:() 2016-08-10 10:36:12.598

ORA-01775: looping chain of synonyms
ORA-01775: looping chain of synonyms

*** 2016-08-10 10:36:12.598
USER (ospid: 42074156): terminating the instance due to error 1775
```

这个错误和数据库参数 replication\_dependency\_tracking 有关,该参数默认为 TRUE,我们需要暂时设置

为 FALSE 来绕过这个检查，把数据库启动起来，然后才可以重建 DUAL 表，在修复数据库后，重启数据库设置

replication\_dependency\_tracking 为 TRUE.

```
SYS@lhrdb> show parameter replication_dependency_tracking
ERROR:
ORA-03114: not connected to ORACLE
```

```
SYS@lhrdb> conn / as sysdba
Connected to an idle instance.
SYS@lhrdb> show parameter replication_dependency_tracking
ORA-01034: ORACLE not available
Process ID: 0
Session ID: 0 Serial number: 0
```

```
SYS@lhrdb> startup nomount
ORACLE instance started.
```

```
Total System Global Area 1720328192 bytes
Fixed Size                2247072 bytes
Variable Size             486540896 bytes
Database Buffers         1224736768 bytes
Redo Buffers              6803456 bytes
```

```
SYS@lhrdb> set line 9999
```

```
SYS@lhrdb> show parameter replication_dependency_tracking
```

NAME	TYPE	VALUE
replication_dependency_tracking	boolean	TRUE

```
SYS@lhrdb> alter system set replication_dependency_tracking=false scope=spfile;
```

```
System altered.
```

```
SYS@lhrdb> startup force
ORACLE instance started.
```

```
Total System Global Area 1720328192 bytes
Fixed Size                2247072 bytes
Variable Size             486540896 bytes
Database Buffers         1224736768 bytes
Redo Buffers              6803456 bytes
```

```
Database mounted.
Database opened.
```

```
SYS@lhrdb> SELECT * FROM DUAL;
SELECT * FROM DUAL
*
```

```
ERROR at line 1:
ORA-01775: looping chain of synonyms
```

接下来就是重建 DUAL 表和编译失效的对象，过程和前边的一样，具体执行省略，代码如下：

```
CREATE TABLE SYS.DUAL(DUMMY VARCHAR2(1)) STORAGE(INITIAL 1) TABLESPACE
SYSTEM;
INSERT INTO DUAL VALUES('X');
COMMIT;
CREATE OR REPLACE PUBLIC SYNONYM DUAL FOR DUAL;
GRANT SELECT ON DUAL TO PUBLIC WITH GRANT OPTION;
@?/rdbms/admin/utlrlp.sql
```

最后我们需要做的事就是设置 replication\_dependency\_tracking 为 true

```
SYS@lhrdb> alter system set replication_dependency_tracking=true scope=spfile;
```

```
System altered.
```

```
SYS@lhrdb> conn / as sysdba
Connected.
SYS@lhrdb> shutdown immediate;
```

```
Database closed.
Database dismounted.
ORACLE instance shut down.
SYS@lhrdb> startup
ORACLE instance started.

Total System Global Area 1720328192 bytes
Fixed Size                2247072 bytes
Variable Size             486540896 bytes
Database Buffers         1224736768 bytes
Redo Buffers              6803456 bytes
Database mounted.
Database opened.
```

1.6 关于 replication\_dependency\_tracking 参数

我们查询官方文档可以了解到参数 replication\_dependency\_tracking 的一些内容：

oracle 10g+11g 官方文档 小麦苗制作(http://blog.itpub.net/26736162/abstract/1/)

隐藏 上一步 前进 主页 打印 选项

目录 索引 搜索 收藏夹

键入要搜索的单词:

列出主题 显示

选择主题:

标题	位置	名次
AWR SQL Report for...	oracle 10gt...	1
REPLICATION DEPEND...	oracle 10gt...	2
REPLICATION DEPEND...	oracle 10gt...	3
ALTER SYSTEM	oracle 10gt...	4
Index	oracle 10gt...	5
Index	oracle 10gt...	6
Index	oracle 10gt...	7
Index	oracle 10gt...	8
Initialization Par...	oracle 10gt...	9
Oracle Database Ma...	oracle 10gt...	10
Oracle Database Ma...	oracle 10gt...	11
Contents	oracle 10gt...	12
Oracle Database Ma...	oracle 10gt...	13
Contents	oracle 10gt...	14
Oracle Database Ma...	oracle 10gt...	15
Planning Your Repl...	oracle 10gt...	16
Initialization Par...	oracle 10gt...	17
Changing Parameter...	oracle 10gt...	18
Changing Parameter...	oracle 10gt...	19
Planning Your Repl...	oracle 10gt...	20
Index	oracle 10gt...	21

Hide Navigation Search

☐ This Book ☒ Entire Library

Reference

Database Library · Master Index · Master Glossary · Book List · Data Dictionary · SQL Keywords · Acronyms · Initialization Parameters · Advanced Search · Error Messages

Categories

- Installation
- Getting Started
- Administration
- Application Development
- Grid Computing
- High Availability
- Data Warehousing
- Content Management and Unstructured Data
- Information Integration
- Security
- Videos

New and changed documents

Previous Next

REPLICATION\_DEPENDENCY\_TRACKING

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	No
Range of values	true   false

REPLICATION\_DEPENDENCY\_TRACKING enables or disables dependency tracking for read/write operations to the database. Dependency tracking is essential for propagating changes in a replicated environment in parallel.

Values:

- TRUE
- FALSE

Enables dependency tracking.

Allows read/write operations to the database to run faster, but does not produce dependency information for Oracle to perform parallel propagation.

**Note:**

Do not specify this value unless you are sure that your application will not perform any read/write operations to the replicated tables.

Property	Description
Parameter type	Boolean
Default value	true
Modifiable	No
Range of values	true   false

REPLICATION DEPENDENCY TRACKING enables or disables dependency tracking for read/write operations to the database. Dependency tracking is essential for propagating changes in a replicated environment in parallel.

Values:

TRUE : Enables dependency tracking.

FALSE : Allows read/write operations to the database to run faster, but does not produce dependency information for Oracle to perform parallel propagation.

Note:

Do not specify this value unless you are sure that your application will not perform any read/write operations to the replicated tables.

**See Also:**

[Oracle Database Advanced Replication](#) for more information on parallel propagation dependency tracking

## 第 2 章 实验中用到的 SQL 总结

### 1、重建 DUAL 表：

```
CREATE TABLE SYS.DUAL(DUMMY VARCHAR2(1)) STORAGE(INITIAL 1) TABLESPACE SYSTEM;
INSERT INTO DUAL VALUES('X');
COMMIT;
CREATE OR REPLACE PUBLIC SYNONYM DUAL FOR DUAL;
GRANT SELECT ON DUAL TO PUBLIC WITH GRANT OPTION;
@?/rdbms/admin/utlrp.sql
SELECT COUNT(*) FROM DBA_OBJECTS D WHERE D.status='INVALID';
SELECT D.OWNER, COUNT(*) FROM DBA_OBJECTS D WHERE D.status='INVALID' GROUP BY D.OWNER;
alter system set replication_dependency_tracking=false scope=spfile;
```

### 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 (xiaomaimiaolhr) 上有同步更新，推荐 pdf 文件阅读或博客园地址阅读
- ❖ QQ 群：230161599 微信群：私聊
- ❖ 本文 itpub 地址：<http://blog.itpub.net/26736162/viewspace-2123291/> 博客园地址：<http://www.cnblogs.com/lhrbest/articles/5758162.html>
- ❖ 本文 pdf 版：<http://yunpan.cn/cdEQedhCs2kFz> (提取码：ed9b)
- ❖ 小麦苗分享的其它资料：<http://blog.itpub.net/26736162/viewspace-1624453/>
- ❖ 联系我请加 QQ 好友(642808185)，注明添加缘由

❖ 于 2016-08-10 09:00~2016-08-10 19:00 在中行完成

❖ 【版权所有，文章允许转载，但须以链接方式注明源地址，否则追究法律责任】

---

长按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：xiaomaimiaolhr, 学习最实用的数据库技术。



小麦苗