

LogMiner 配置使用手册

1 Logminer 简介

1.1 LogMiner 介绍

Oracle LogMiner 是 Oracle 公司从产品 8i 以后提供的一个实际非常有用的分析工具，使用该工具可以轻松获得 Oracle 在线/归档日志文件中的具体内容，特别是该工具可以分析出所有对于数据库操作的 DML 和 DDL 语句。该工具特别适用于调试、审计或者回退某个特定的事务。

LogMiner 分析工具实际上是由一组 PL/SQL 包和一些动态视图(Oracle8i 内置包的一部分) 组成，它作为 Oracle 数据库的一部分来发布是 8i 产品提供的一个完全免费的工具。但该工具和其他 Oracle 内建工具相比使用起来显得有些复杂，主要原因是该工具没有提供任何图形用户界面 (GUI)。

1.2 LogMiner 作用

在 Oracle 8i 之前，Oracle 没有提供任何协助数据库管理员来读取和解释重作日志文件内容的工具。系统出现问题，对于一个普通的数据管理员来讲，唯一可以作的工作就是将所有的 log 文件打包，然后发给 Oracle 公司的技术支持，然后静静地等待 Oracle 公司技术支持给我们最后的答案。然而从 8i 以后，Oracle 提供了这样一个强有力的工具--LogMiner。

LogMiner 工具即可以用来分析在线，也可以用来分析离线日志文件，即可以分析本身自己数据库的重作日志文件，也可以用来分析其他数据库的重作日志文件。

总的说来，LogMiner 工具的主要用途有：

1. 跟踪数据库的变化：可以离线的跟踪数据库的变化，而不会影响在线系统的性能。
2. 回退数据库的变化：回退特定的变化数据，减少 point-in-time recovery 的执行。
3. 优化和扩容计划：可通过分析日志文件中的数据以分析数据增长模式

1.3 使用详解

1.3.1 安装 LogMiner

在使用 LogMiner 之前需要确认 Oracle 是否带有进行 LogMiner 分析包，一般来说

Windows 操作系统 Oracle10g 以上都默认包含。如果不能确认，可以 DBA 身份登录系统，查看系统中是否存在运行 LogMiner 所需要的 dbms_logmnr、dbms_logmnr_d 包，如果没有需要安装 LogMiner 工具，必须首先要运行下面这样两个脚本：

- 1、\$ORACLE_HOME/rdbms/admin/dbmslm.sql
- 2、\$ORACLE_HOME/rdbms/admin/dbmslmd.sql.

这两个脚本必须均以 DBA 用户身份运行。其中第一个脚本用来创建 DBMS_LOGMNR 包，该包用来分析日志文件。第二个脚本用来创建 DBMS_LOGMNR_D 包，该包用来创建数据字典文件。

创建完毕后将包括如下过程和视图：

类型	过程名	用途
过程	Dbms_logmnr_d.build	创建一个数据字典文件
过程	Dbms_logmnr.add_logfile	在类表中增加日志文件以供分析
过程	Dbms_logmnr.start_logmnr	使用一个可选的字典文件和前面确定要分析日志文件来启动 LogMiner
过程	Dbms_logmnr.end_logmnr	停止 LogMiner 分析
视图	V\$logmnr_dictionary	显示用来决定对象 ID 名称的字典文件的信息
视图	V\$logmnr_logs	在 LogMiner 启动时显示分析的日志列表
视图	V\$logmnr_contents	LogMiner 启动后，可以使用该视图在 SQL 提示符下输入 SQL 语句来查询重做日志的内容

1.3.2 创建数据字典文件

LogMiner 工具实际上是由两个新的 PL/SQL 内建包 (DBMS_LOGMNR 和 DBMS_LOGMNR_D)和四个 V\$动态性能视图(视图是在利用过程 DBMS_LOGMNR.START_LOGMNR 启动 LogMiner 时创建) 组成。在使用 LogMiner 工具分析 redo log 文件之前，可以使用 DBMS_LOGMNR_D 包将数据字典导出为一个文本文件。该字典文件是可选的，但是如果没有它，LogMiner 解释出来的语句中关于数据字典中的部分 (如表名、列名等) 和数值都将是 16 进制的形式，我们是无法直接理解的。例如，下面的 sql 语句：

```
INSERT INTO dm_dj_swry (rydm, rymc) VALUES (00005, '张三');
```

LogMiner 解释出来的结果将是下面这个样子，

```
insert into Object#308(col#1, col#2) values (hextoraw('c30rte567e436'),
hextoraw('4a6f686e20446f65'));
```

创建数据字典的目的就是让 LogMiner 引用涉及到内部数据字典中的部分时为他们实际的名字，而不是系统内部的 16 进制。数据字典文件是一个文本文件，使用包 DBMS_LOGMNR_D

来创建。如果我们要分析的数据库中的表有变化，影响到库的数据字典也发生变化，这时就需要重新创建该字典文件。另外一种情况是在分析另外一个数据库文件的重作日志时，也必须要重新生成一遍被分析数据库的数据字典文件。

创建数据字典文件之前需要配置 LogMiner 文件夹：

```
CREATE DIRECTORY utlfile AS 'D:\oracle\oradata\practice\LOGMNR';  
alter system set utl_file_dir='D:\oracle\oradata\practice\LOGMNR' scope=spfile;
```

创建字典文件需要以 DBA 用户登录，创建到上面配置好的 LogMiner 文件夹中：

```
CONN LOGMINER/ LOGMINER@PRACTICE AS SYSDBA  
EXECUTE dbms_logmnr_d.build(dictionary_filename => 'dictionary.ora', dictionary_  
location => 'D:\oracle\oradata\practice\LOGMNR');
```

1.3.3 加入需分析的日志文件

Oracle 的 LogMiner 可以分析在线 (online) 和归档 (offline) 两种日志文件，加入分析日志文件使用 dbms_logmnr.add_logfile 过程，第一个文件使用 dbms_logmnr.NEW 参数，后面文件使用 dbms_logmnr.ADDFILE 参数。

1、创建列表

```
BEGIN  
    dbms_logmnr.add_logfile(logfilename=>'D:\oracle\oradata\practice\REDO03.LOG',options=>dbms_logmnr.NEW);  
END;  
/
```

2、添加其他日志文件到列表

```
BEGIN  
    dbms_logmnr.add_logfile(logfilename=>'D:\oracle\oradata\practice\ARCHIVE\ARC00002_0817639922.001',options=>dbms_logmnr.ADDFILE);  
    dbms_logmnr.add_logfile(logfilename=>'D:\oracle\oradata\practice\ARCHIVE\ARC00003_0817639922.001',options=>dbms_logmnr.ADDFILE);  
END;  
/
```

1.3.4 使用 LogMiner 进行日志分析

Oracle 的 LogMiner 分析时分为无限制条件和限制条件两种，无限制条件中分析所有加入到分析列表日志文件，限制条件根据限制条件分析指定范围日志文件。

1、无限制条件

EXECUTE

```
dbms_logmnr.start_logmnr(dictfilename=>'D:\oracle\oradata\practice\LOGMNR\dictionary.ora');
```

2、有限制条件

通过对过程 DBMS_LOGMNR.START_LOGMNR 中几个不同参数的设置(参数含义见表 1), 可以缩小要分析日志文件的范围。通过设置起始时间和终止时间参数我们可以限制只分析某一时间范围的日志。

参数	参数类型	默认值	含义
StartScn	数字型	0	分析重作日志中 SCN \geq StartScn 日志文件部分
EndScn	数字型	0	分析重作日志中 SCN \leq EndScn 日志文件部分
StartTime	日期型	1998-01-01	分析重作日志中时间戳 \geq StartTime 的日志文件部分
EndTime	日期型	2988-01-01	分析重作日志中时间戳 \leq EndTime 的日志文件部分
DictFileName	字符型		字典文件该文件包含一个数据库目录的快照。

如下面的例子，我们仅仅分析 2013 年 6 月 8 日的日志，：

```
EXECUTE dbms_logmnr.start_logmnr(
    DictFileName => dictfilename=>'D:\..\practice\LOGMNR\dictionary.ora',
    StartTime => to_date('2013-6-8 00:00:00','YYYY-MM-DD HH24:MI:SS')
    EndTime => to_date('2013-6-8 23:59:59','YYYY-MM-DD HH24:MI:SS '));
```

也可以通过设置起始 SCN 和截至 SCN 来限制要分析日志的范围：

```
EXECUTE dbms_logmnr.start_logmnr(
    DictFileName => 'D:\..\practice\LOGMNR\dictionary.ora',
    StartScn => 20,
    EndScn => 50);
```

1.3.5 观察分析结果 (v\$logmnr_contents)

到现在为止，我们已经分析得到了重作日志文件中的内容。动态性能视图 v\$logmnr_contents 包含 LogMiner 分析得到的所有的信息。

```
SELECT sql_redo FROM v$logmnr_contents;
```

如果我们仅仅想知道某个用户对于某张表的操作，可以通过下面的 SQL 查询得到，该查询可以得到用户 LOGMINER 对表 EMP 所作的一切工作。

```
SELECT sql_redo FROM v$logmnr_contents WHERE username='LOGMINER' AND
```

tablename='EMP';

序号	名称	含义
1	SCN	特定数据变化的系统更改号
2	TIMESTAM	数据改变发生的时间
3	COMMIT_TIMESTAMP	数据改变提交的时间
4	SEG_OWNER	数据发生改变的段名称
5	SEG_NAME	段的所有者名称
6	SEG_TYPE	数据发生改变的段类型
7	SEG_TYPE_NAME	数据发生改变的段类型名称
8	TABLE_SPACE	变化段的表空间
9	ROW_ID	特定数据变化行的 ID
10	SESSION_INFO	数据发生变化时用户进程信息
11	OPERATION	重做记录中记录的操作(如 INSERT)
12	SQL_REDO	可以为重做记录重做指定行变化的 SQL 语句(正向操作)
13	SQL_UNDO	可以为重做记录回退或恢复指定行变化的 SQL 语句(反向操作)

需要强调一点的是，视图 v\$logmnr_contents 中的分析结果仅在我们运行过程 'dbms_logmnr.start_logmnr' 这个会话的生命期中存在。这是因为所有的 LogMiner 存储都在 PGA 内存中，所有其他的进程是看不到它的，同时随着进程的结束，分析结果也随之消失。

最后，使用过程 DBMS_LOGMNR.END_LOGMNR 终止日志分析事务，此时 PGA 内存区域被清除，分析结果也随之不再存在。

2 数据同步 Oracle 数据库设置

Oracle 数据库使用 LogMiner 查看执行 SQL 语句，其中需要进行如下四步骤是指：

1. 设置数据库为归档模式；
2. 设置 LogMiner 字典文件路径等；
3. 创建数据同步用户(如用户名为 LOGMINER，该用户拥有 DBA 权限)；
4. 验证配置是否成功；

2.1 设置数据库为归档模式

2.1.1 查看数据库是否为归档模式

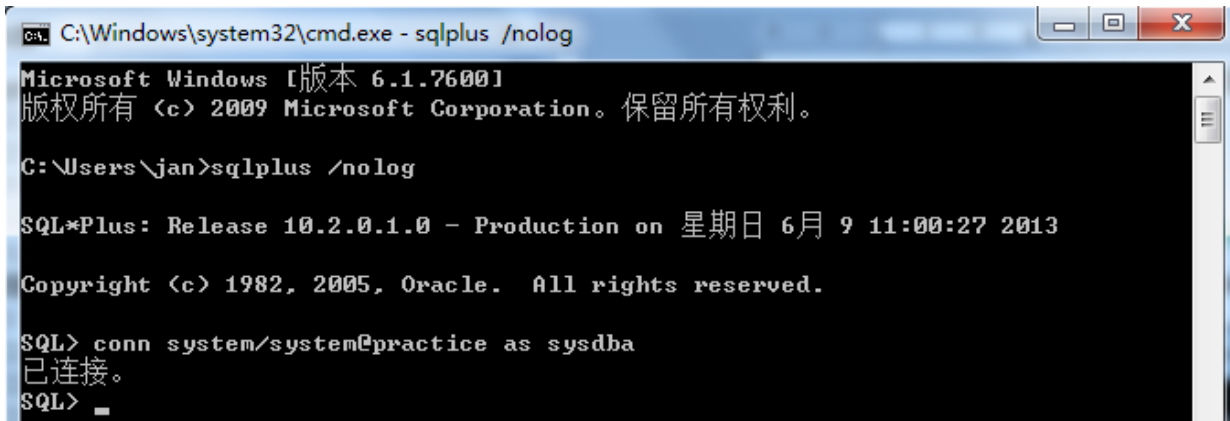
使用 SqlPlus 或者命令行界面连接数据库（以下以命令行界面操作）

--进入 SqlPlus 程序

```
sqlplus /nolog
```

--使用 DBA 用户登录到源数据库中

```
conn system/system@practic as sysdba
```



```
C:\Windows\system32\cmd.exe - sqlplus /nolog
Microsoft Windows [版本 6.1.7600]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\jan>sqlplus /nolog

SQL*Plus: Release 10.2.0.1.0 - Production on 星期日 6月 9 11:00:27 2013

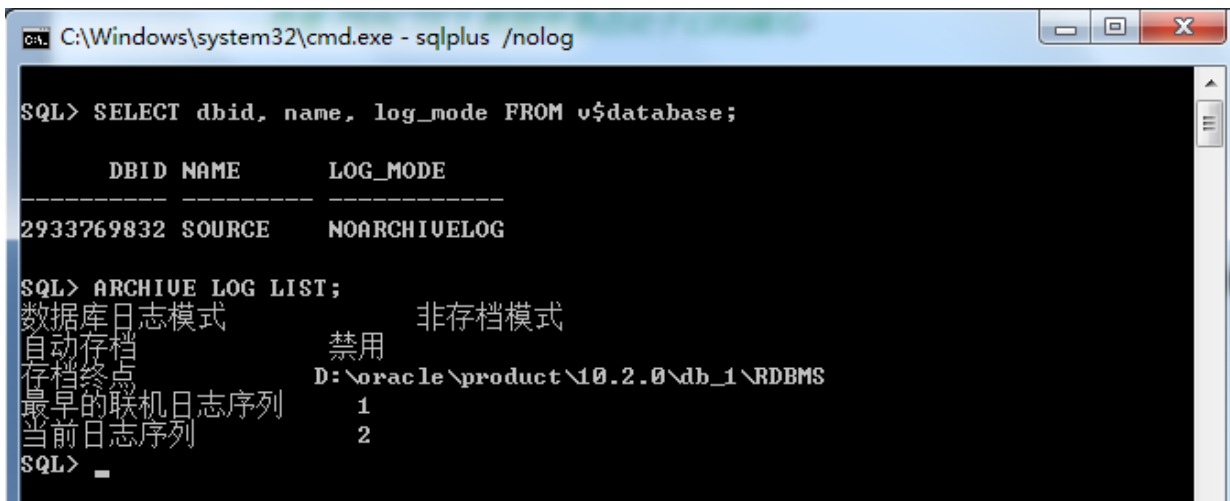
Copyright (c) 1982, 2005, Oracle. All rights reserved.

SQL> conn system/system@practice as sysdba
已连接。
SQL>
```

--查看 PRACTICE 数据库是否处于归档模式

```
SELECT dbid, name, log_mode FROM v$database;
```

或者 ARCHIVE LOG LIST;



```
C:\Windows\system32\cmd.exe - sqlplus /nolog

SQL> SELECT dbid, name, log_mode FROM v$database;

   DBID NAME          LOG_MODE
-----
2933769832 SOURCE          NOARCHIVELOG

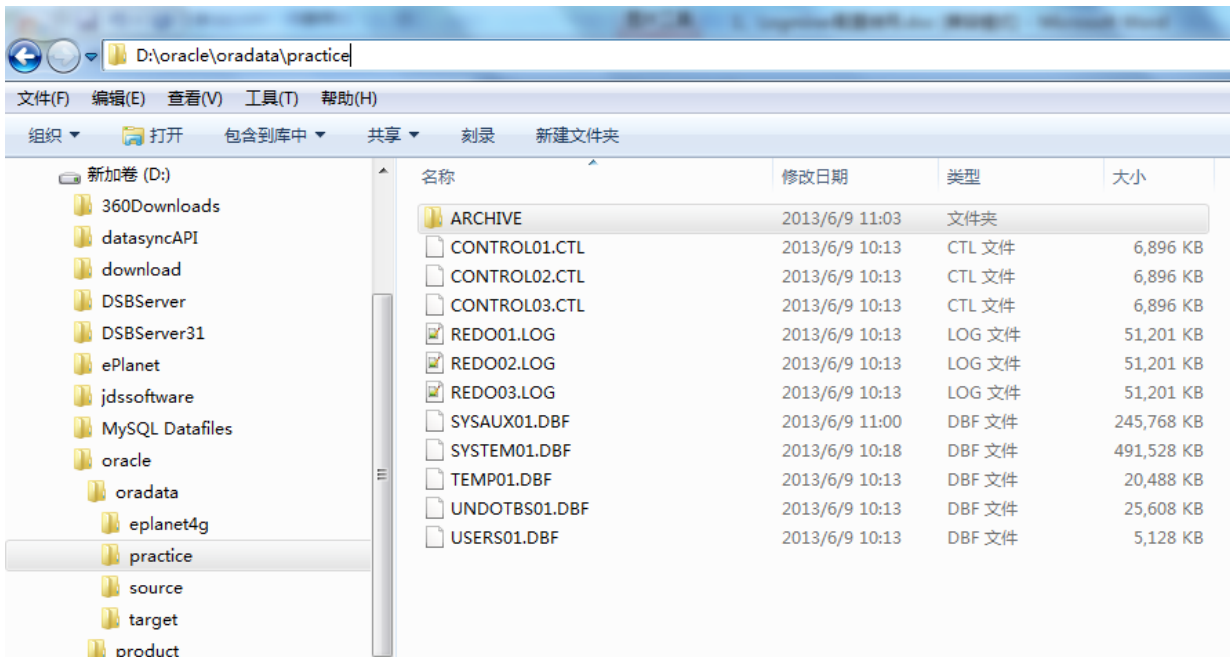
SQL> ARCHIVE LOG LIST;
数据库日志模式          非存档模式
自动存档                禁用
存档终点                D:\oracle\product\10.2.0\db_1\RDBMS
最早的联机日志序列      1
当前日志序列            2
SQL>
```

如果显示数据库显示为归档模式，则设置数据库为归档模式可跳过；如果显示数据库为非归档模式则需要进行以下设置。

上图显示数据库未进行归档，需要进行归档设置。

2.1.2 设置归档模式

创建 ARCHIVE 文件夹，ARCHIVE 文件夹路径根据所在服务器进行设置，在下面操作中设置为"D:\oracle\oradata\practice\ARCHIVE"

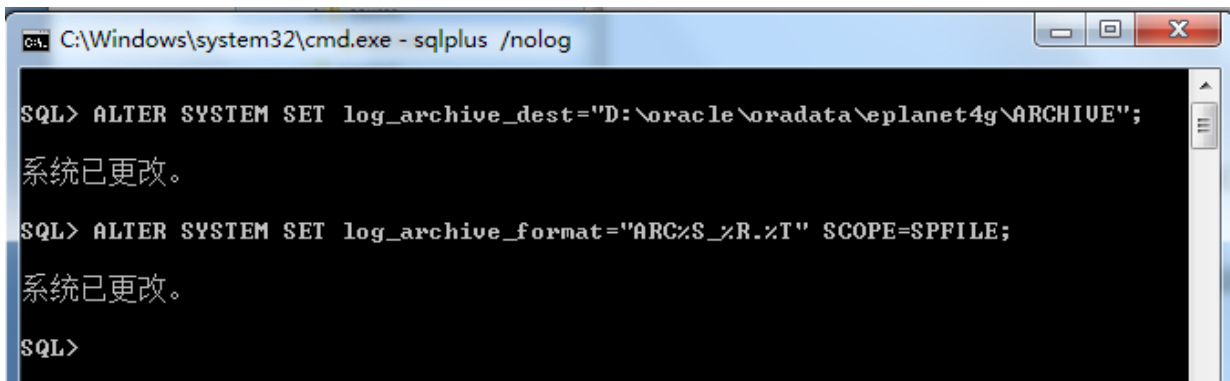


--设置归档日志文件路径

```
ALTER SYSTEM SET log_archive_dest="D:\oracle\oradata\practice\ARCHIVE";
```

--日志文件名称格式：

```
ALTER SYSTEM SET log_archive_format="ARC%S_%R.%T" SCOPE=SPFILE;
```



--修改完毕后，关闭数据库，以 MOUNT 方式启动

```
SHUTDOWN IMMEDIATE;
```

```
STARTUP MOUNT;
```

--设置数据库为归档模式

```
ALTER DATABASE ARCHIVELOG;
```

```

C:\Windows\system32\cmd.exe - sqlplus /nolog

SQL> SHUTDOWN IMMEDIATE;
数据库已经关闭。
已经卸载数据库。
ORACLE 例程已经关闭。
SQL> STARTUP MOUNT;
ORACLE 例程已经启动。

Total System Global Area 1258291200 bytes
Fixed Size          1250548 bytes
Variable Size       301992716 bytes
Database Buffers    947912704 bytes
Redo Buffers        7135232 bytes
数据库装载完毕。
SQL> ALTER DATABASE ARCHIVELOG;
数据库已更改。

```

(注意：如果重启数据失败，请参考第 4 章节异常问题处理)

2.1.3 验证归档是否设置成功

--查看 PRACTICE 数据库是否处于归档模式

```

SELECT dbid, name, log_mode FROM v$database;
或者 ARCHIVE LOG LIST;

```

```

C:\Windows\system32\cmd.exe - sqlplus /nolog

SQL> SELECT dbid, name, log_mode FROM v$database;

   DBID NAME          LOG_MODE
-----
2933769832 SOURCE      ARCHIVELOG

SQL> ARCHIVE LOG LIST;
数据库日志模式          存档模式
自动存档                  启用
存档终点                  D:\oracle\oradata\eplanet4g\ARCHIVE
最早的联机日志序列        1
下一个存档日志序列        2
当前日志序列                2
SQL>

```

--验证参数设置是否起作用

```

SELECT dest_id, status, destination FROM v$archive_dest WHERE dest_id =1;

```

```

C:\Windows\system32\cmd.exe - sqlplus /nolog

SQL> SELECT dest_id, status, destination FROM v$archive_dest WHERE dest_id =1;

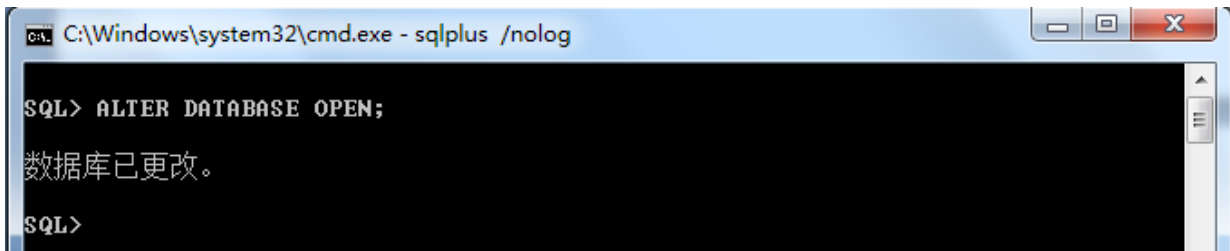
   DEST_ID STATUS      DESTINATION
-----
1  VALID      D:\oracle\oradata\eplanet4g\ARCHIVE

SQL>

```

--在参数文件设置已经起作用，打开数据库

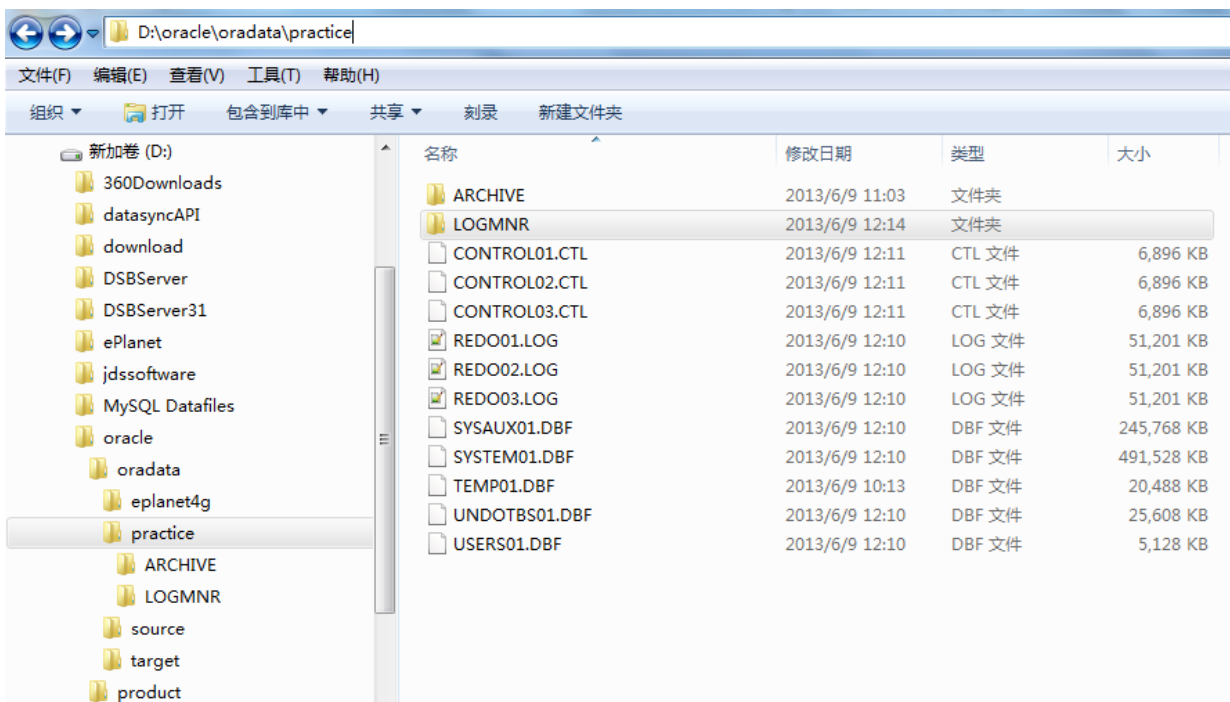

```
ALTER DATABASE OPEN;
```



2.2 LogMiner 设置

2.2.1 创建 LogMiner 文件夹

创建 LOGMNR 文件夹，路径为"D:\oracle\oradata\practice\LOGMNR"



2.2.2 设置 LogMiner 字典文件路径

--创建数据字典文件

```
CREATE DIRECTORY utlfile AS 'D:\oracle\oradata\practice\LOGMNR';
```

```
alter system set utl_file_dir='D:\oracle\oradata\practice\LOGMNR' scope=spfile;
```

```
C:\Windows\system32\cmd.exe - sqlplus /nolog

SQL> CREATE DIRECTORY utlfile AS 'D:\oracle\oradata\practice\LOGMNR';
目录已创建。

SQL> alter system set utl_file_dir='D:\oracle\oradata\practice\LOGMNR' scope=spfile;
系统已更改。

SQL>
```

2.2.3 开启 LogMiner 日志补充模式

--创建数据字典文件

```
alter database add supplemental log data;
```

```
C:\Windows\system32\cmd.exe - sqlplus /nolog

SQL> alter database drop supplemental log data (all) columns;
数据库已更改。

SQL> alter database add supplemental log data;
数据库已更改。

SQL>
```

2.2.4 重启数据库验证

--修改完毕后，关闭数据库，以 MOUNT 方式启动

```
SHUTDOWN IMMEDIATE;
```

```
STARTUP;
```

--查看 Logminer 文件夹是否设置

```
SHOW PARAMETER utl_file_dir;
```

```
C:\Windows\system32\cmd.exe - sqlplus /nolog
SQL> SHUTDOWN IMMEDIATE;
数据库已经关闭。
已经卸载数据库。
ORACLE 例程已经关闭。
SQL> STARTUP;
ORACLE 例程已经启动。

Total System Global Area 1258291200 bytes
Fixed Size 1250548 bytes
Variable Size 310381324 bytes
Database Buffers 939524096 bytes
Redo Buffers 7135232 bytes
数据库装载完毕。
数据库已经打开。
SQL> SHOW PARAMETER utl_file_dir;

NAME                                TYPE                                VALUE
-----                                -                                -
utl_file_dir                          string                              D:\oracle\oradata\practice\LOG
MNR
SQL>
```

2.3 创建数据同步用户

在数据库创建 LOGMINER 用户，该用户需要具有 DBA 权限

--在源数据库创建 LOGMINER 用户，并赋予 DBA 权限

```
CREATE USER LOGMINER IDENTIFIED BY LOGMINER;
GRANT CONNECT, RESOURCE, DBA TO LOGMINER;
```

```
C:\Windows\system32\cmd.exe - sqlplus /nolog
SQL> CREATE USER LOGMINER IDENTIFIED BY LOGMINER;
用户已创建。
SQL> GRANT CONNECT, RESOURCE, DBA TO LOGMINER;
授权成功。
```

3 使用 LogMiner 读取日志例子

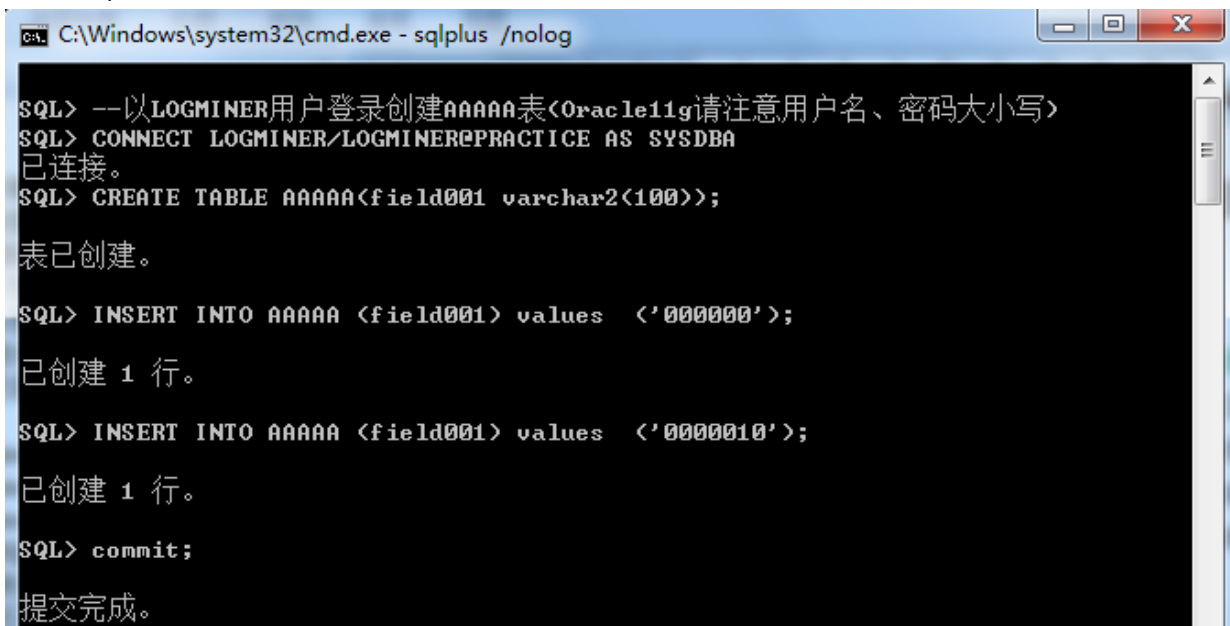
在使用 LogMiner 读取归档/在线日志需要按照第 2 章节进行设置，设置完毕后可以对归档和在线日志进行分析。特别是需要开启 LogMiner 日志补充模式，如果没有开始 LogMiner 补充模式将无法查看 DDL 语句，按照测试结果看，只有开始 LogMiner 日志补充模式后，才能查看 DDL 语句，在此之前进行 DDL 将无法进行查看。

3.1 使用 LogMiner 读取在线日志

3.1.1 测试数据准备

--以 LOGMINER 用户登录 (非 DBA 登录) 创建 AAAAA 表(Oracle11g 请注意用户名、密码大小写)

```
CONNECT LOGMINER/LOGMINER@PRACTICE
CREATE TABLE AAAAA(field001 varchar2(100));
INSERT INTO AAAAA (field001) values ('000000');
INSERT INTO AAAAA (field001) values ('0000010');
commit;
```



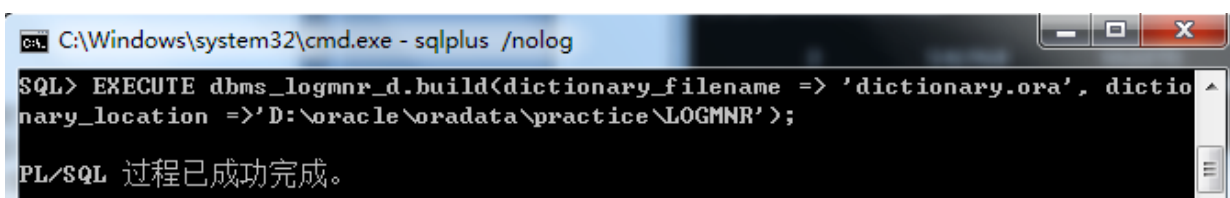
```
ca. C:\Windows\system32\cmd.exe - sqlplus /nolog
SQL> --以LOGMINER用户登录创建AAAAA表<Oracle11g请注意用户名、密码大小写>
SQL> CONNECT LOGMINER/LOGMINER@PRACTICE AS SYSDBA
已连接。
SQL> CREATE TABLE AAAAA(field001 varchar2(100));
表已创建。
SQL> INSERT INTO AAAAA (field001) values ('000000');
已创建 1 行。
SQL> INSERT INTO AAAAA (field001) values ('0000010');
已创建 1 行。
SQL> commit;
提交完成。
```

3.1.2 创建数据字典文件

数据库对象发生变化，需要重新创建数据字典文件

--以 LOGMINER 用户 (DBA 权限) 登录，生成字典文件

```
CONN LOGMINER/LOGMINER@PRACTICE AS SYSDBA
EXECUTE dbms_logmnr_d.build(dictionary_filename => 'dictionary.ora', dictionary_location => 'D:\oracle\oradata\practice\LOGMNR');
```



```
ca. C:\Windows\system32\cmd.exe - sqlplus /nolog
SQL> EXECUTE dbms_logmnr_d.build(dictionary_filename => 'dictionary.ora', dictionary_location => 'D:\oracle\oradata\practice\LOGMNR');
PL/SQL 过程已成功完成。
```

3.1.3 确认当前处于联机状态的日志文件

--需要确认当前处于联机状态的日志文件

```
SELECT group#, sequence#, status, first_change#, first_time FROM V$log ORDER BY first_change#;
```

```
C:\Windows\system32\cmd.exe - sqlplus /nolog

SQL> SELECT group#, sequence#, status, first_change#, first_time FROM U$log ORDER BY first_change#;

  GROUP# SEQUENCE# STATUS          FIRST_CHANGE# FIRST_TIME
-----
         1         2 INACTIVE          541968 09-6月 -13
         2         3 ACTIVE            552272 09-6月 -13
         3         4 CURRENT           555526 09-6月 -13
```

从上图可以看出在线日志 REDO03 处于 ACTIVE 状态中

3.1.4 加入需分析的日志文件

--加入解析在线日志文件

```
BEGIN
dbms_logmnr.add_logfile(logfilename=>'D:\oracle\oradata\practice\REDO03.LOG',options=>dbms_logmnr.NEW);
END;
/
```

```
C:\Windows\system32\cmd.exe - sqlplus /nolog

SQL> BEGIN
  2  dbms_logmnr.add_logfile(logfilename=>'D:\oracle\oradata\practice\REDO03.LOG',options=>dbms_logmnr.NEW);
  3  END;
  4  /

PL/SQL 过程已成功完成。

SQL> _
```

3.1.5 使用 LogMiner 进行分析

--启动 LogMiner 进行分析

```
EXECUTE
dbms_logmnr.start_logmnr(dictfilename=>'D:\oracle\oradata\practice\LOGMNR\dictionary.ora');
```

```
C:\Windows\system32\cmd.exe - sqlplus /nolog

SQL> EXECUTE dbms_logmnr.start_logmnr(dictfilename=>'D:\oracle\oradata\practice\LOGMNR\dictionary.ora');

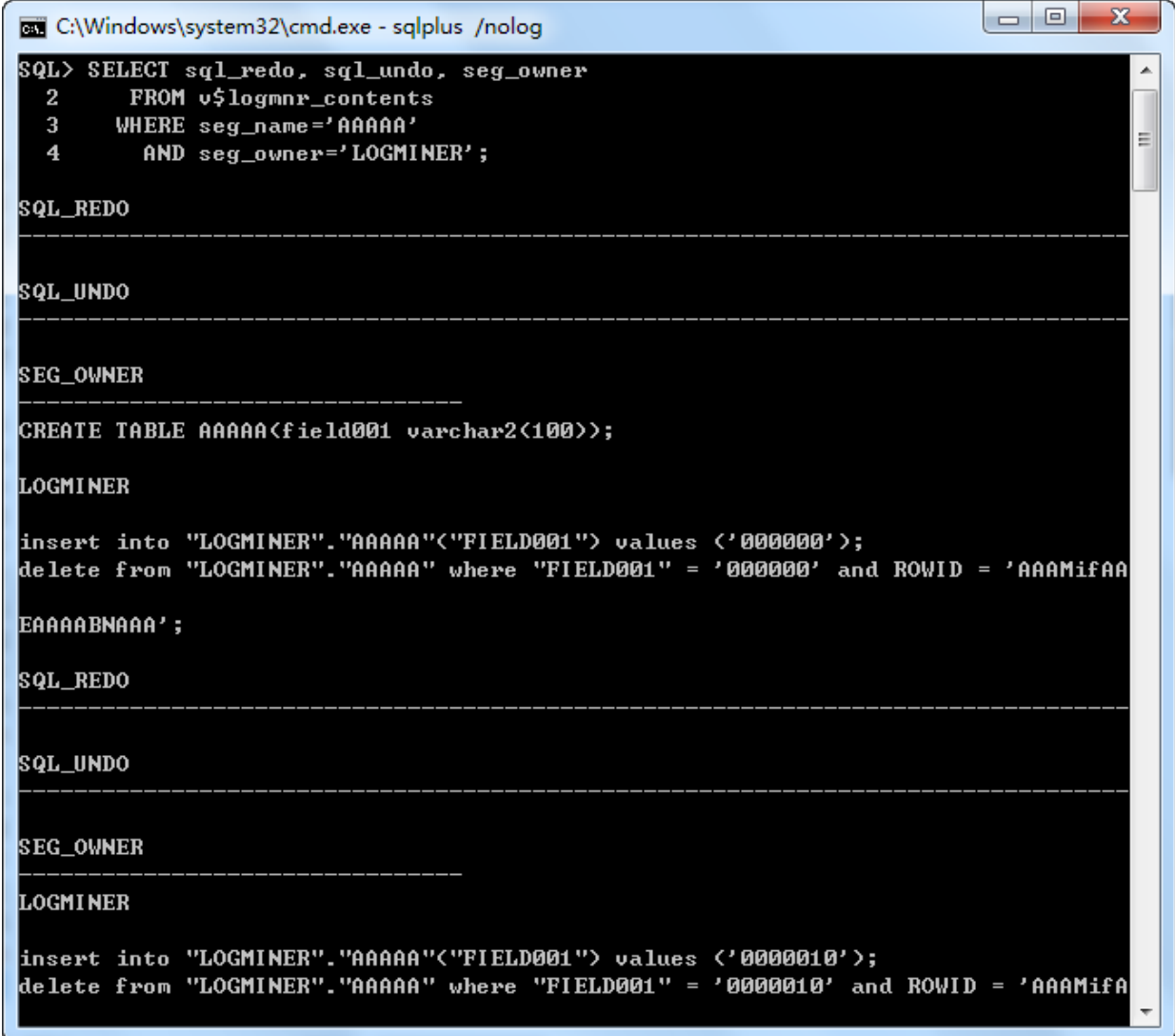
PL/SQL 过程已成功完成。

SQL> _
```

3.1.6 观察分析结果

--查询相关操作日志

```
SELECT sql_redo, sql_undo, seg_owner
FROM v$logmnr_contents
WHERE seg_name='AAAAA'
AND seg_owner='LOGMINER';
```



```
C:\Windows\system32\cmd.exe - sqlplus /nolog
SQL> SELECT sql_redo, sql_undo, seg_owner
2     FROM v$logmnr_contents
3     WHERE seg_name='AAAAA'
4     AND seg_owner='LOGMINER';

SQL_REDO
-----
SQL_UNDO
-----
SEG_OWNER
-----
CREATE TABLE AAAAA(field001 varchar2(100));
LOGMINER
insert into "LOGMINER"."AAAAA"("FIELD001") values ('000000');
delete from "LOGMINER"."AAAAA" where "FIELD001" = '000000' and ROWID = 'AAAMifAA
EAAAABNAAA';
SQL_REDO
-----
SQL_UNDO
-----
SEG_OWNER
-----
LOGMINER
insert into "LOGMINER"."AAAAA"("FIELD001") values ('0000010');
delete from "LOGMINER"."AAAAA" where "FIELD001" = '0000010' and ROWID = 'AAAMifa
```

3.2 使用 LogMiner 读取归档日志

3.2.1 测试数据准备

--以 LOGMINER 用户登录 (非 DBA 权限) 创建 EMP 表(Oracle11g 请注意用户名、密码大小写)

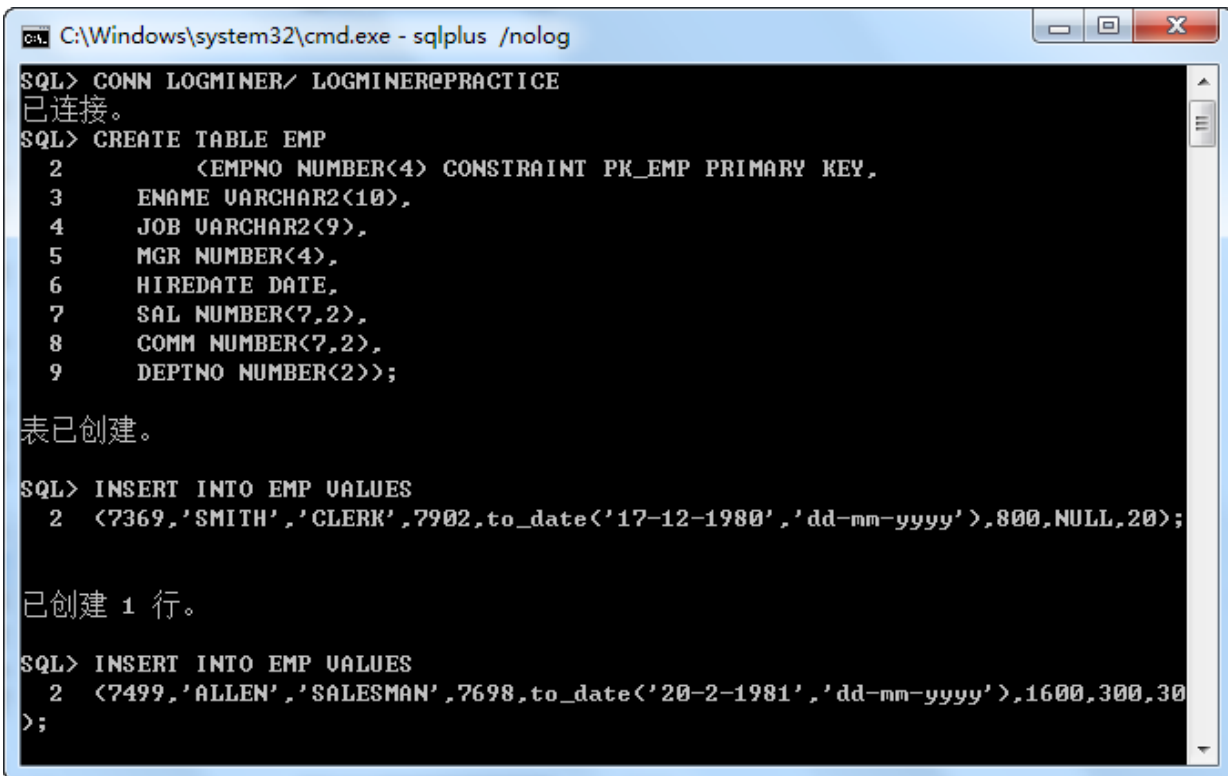
```
CONN LOGMINER/ LOGMINER@PRACTICE
CREATE TABLE EMP
```

```
(EMPNO NUMBER(4) CONSTRAINT PK_EMP PRIMARY KEY,
```

```
ENAME VARCHAR2(10),
JOB VARCHAR2(9),
MGR NUMBER(4),
HIREDATE DATE,
SAL NUMBER(7,2),
COMM NUMBER(7,2),
DEPTNO NUMBER(2));
```

--插入 EMP 数据

```
INSERT INTO EMP VALUES
(7369,'SMITH','CLERK',7902,to_date('17-12-1980','dd-mm-yyyy'),800,NULL,20);
INSERT INTO EMP VALUES
(7499,'ALLEN','SALESMAN',7698,to_date('20-2-1981','dd-mm-yyyy'),1600,300,30);
INSERT INTO EMP VALUES
(7521,'WARD','SALESMAN',7698,to_date('22-2-1981','dd-mm-yyyy'),1250,500,30);
INSERT INTO EMP VALUES
(7566,'JONES','MANAGER',7839,to_date('2-4-1981','dd-mm-yyyy'),2975,NULL,20);
COMMIT;
```



```
C:\Windows\system32\cmd.exe - sqlplus /nolog
SQL> CONN LOGMINER/ LOGMINER@PRACTICE
已连接。
SQL> CREATE TABLE EMP
 2      (EMPNO NUMBER(4) CONSTRAINT PK_EMP PRIMARY KEY,
 3      ENAME VARCHAR2(10),
 4      JOB VARCHAR2(9),
 5      MGR NUMBER(4),
 6      HIREDATE DATE,
 7      SAL NUMBER(7,2),
 8      COMM NUMBER(7,2),
 9      DEPTNO NUMBER(2));
表已创建。
SQL> INSERT INTO EMP VALUES
 2  (7369,'SMITH','CLERK',7902,to_date('17-12-1980','dd-mm-yyyy'),800,NULL,20);
已创建 1 行。
SQL> INSERT INTO EMP VALUES
 2  (7499,'ALLEN','SALESMAN',7698,to_date('20-2-1981','dd-mm-yyyy'),1600,300,30);
>;
```

--从 v\$log 视图中找出日志文件的序号

```
CONNECT system/system@practice as sysdba
ALTER SYSTEM SWITCH LOGFILE;
select sequence#, FIRST_CHANGE#, NEXT_CHANGE#,name from v$archived_log order
by sequence# desc;
```

```
C:\Windows\system32\cmd.exe - sqlplus /nolog

SQL> CONNECT system/system@practice as sysdba
已连接。
SQL> ALTER SYSTEM SWITCH LOGFILE;
系统已更改。

SQL> select sequence#, FIRST_CHANGE#, NEXT_CHANGE#,name from v$archived_log order
by sequence# desc;

SEQUENCE# FIRST_CHANGE# NEXT_CHANGE#
-----
NAME
-----
2          541968          552272
D:\ORACLE\PRODUCT\10.2.0\FLASH_RECOVERY_AREA\PRACTICE\ARCHIVELOG\2013_06_09\01_M
F_1_2_8U812GVH_.ARC
2          541968          552272
D:\ORACLE\ORADATA\PRACTICE\ARCHIVE\ARC00002_0817639922.001
```

3.2.2 创建数据字典文件

确保按照 2.2 进行 logMiner 设置

--以 LOGMINER 用户 (DBA 权限) 登录, 生成字典文件

```
CONN LOGMINER/ LOGMINER@PRACTICE AS SYSDBA
```

```
EXECUTE dbms_logmnr_d.build(dictionary_filename => 'dictionary.ora', dictionary_loca
tion => 'D:\oracle\oradata\practice\LOGMNR');
```

```
C:\Windows\system32\cmd.exe - sqlplus /nolog

SQL> EXECUTE dbms_logmnr_d.build(dictionary_filename => 'dictionary.ora', dictio
nary_location => 'D:\oracle\oradata\practice\LOGMNR');

PL/SQL 过程已成功完成。
```

3.2.3 加入需分析的日志文件

--加入解析日志文件

```
BEGIN
```

```
dbms_logmnr.add_logfile(logfilename=> 'D:\oracle\oradata\practice\ARCHIVE\ARC000
02_0817639922.001',options=> dbms_logmnr.NEW);
```

```
END;
```

```
/
```



```
C:\Windows\system32\cmd.exe - sqlplus /nolog

SQL> --加入解析日志文件
SQL> BEGIN
  2  dbms_logmnr.add_logfile(logfilename=>'D:\oracle\oradata\practice\ARCHIVE\AR
C000002_0817639922.001',options=>dbms_logmnr.NEW);
  3  END;
  4  /

PL/SQL 过程已成功完成。
```

3.2.4 使用 LogMiner 进行分析

--启动 LogMiner 进行分析

EXECUTE

```
dbms_logmnr.start_logmnr(dictfilename=>'D:\oracle\oradata\practice\LOGMNR\dictio
nary.ora');
```

```
C:\Windows\system32\cmd.exe - sqlplus /nolog

SQL> EXECUTE dbms_logmnr.start_logmnr(dictfilename=>'D:\oracle\oradata\practice\
LOGMNR\dictionary.ora');

PL/SQL 过程已成功完成。

SQL> _
```

3.2.5 观察分析结果

--查询相关操作日志

```
SELECT sql_redo, sql_undo
       FROM v$logmnr_contents
       WHERE seg_name='EMP'
              AND seg_owner='LOGMINER';
```

```
C:\Windows\system32\cmd.exe - sqlplus /nolog
SQL> SELECT sql_redo, sql_undo
2     FROM v$logmnr_contents
3     WHERE seg_name='EMP'
4     AND seg_owner='LOGMINER';

SQL_REDO
-----
SQL_UNDO
-----

CREATE TABLE EMP
  (EMPNO NUMBER(4) CONSTRAINT PK_EMP PRIMARY KEY,
   ENAME VARCHAR2(10),
   JOB VARCHAR2(9),
   MGR NUMBER(4),
   HIREDATE DATE,
   SAL NUMBER(7,2),
   COMM NUMBER(7,2),
   DEPTNO NUMBER(2));

SQL_REDO
-----
SQL_UNDO
-----

insert into "LOGMINER"."EMP"("EMPNO","ENAME","JOB","MGR","HIREDATE","SAL","COMM"
,"DEPTNO") values ('7369','SMITH','CLERK','7902',TO_DATE('17-12月-80','DD-MON-R
R'),'800',NULL,'20');
delete from "LOGMINER"."EMP" where "EMPNO" = '7369' and "ENAME" = 'SMITH' and "J
```

4 其他

4.1 异常问题处理

4.1.1 出现 ORA-12514 错误

如果出现 ORA-12514 错误时，如下图所示：

```
C:\Windows\system32\cmd.exe - sqlplus /nolog
SQL> SHUTDOWN IMMEDIATE;
数据库已经关闭。
已经卸载数据库。
ORACLE 例程已经关闭。
SQL> STARTUP MOUNT;
ORA-12514: TNS: 监听程序当前无法识别连接描述符中请求的服务
SQL> ALTER DATABASE ARCHIVELOG;
ERROR:
ORA-03114: 未连接到 ORALCE
```

需要修改 listener.ora 文件，具体在 Oracle 安装目录\NETWORK\ADMIN 下，当前操作为"D:\oracle\product\10.2.0\db_1\NETWORK\ADMIN\listener.ora"加入如下设置

```
(SID_DESC =
  (GLOBAL_DBNAME = practice)
  (ORACLE_HOME = D:\oracle\product\10.2.0\db_1)
  (SID_NAME = practice)
)
```

设置后需要重新启动 TNSListener，即可生效

4.1.2 出现 ORA-16018 错误

如果出现 ORA-16018 错误时，如下图所示：

```
C:\Windows\system32\cmd.exe - sqlplus /nolog
SQL> ALTER SYSTEM SET log_archive_dest="D:\oracle\oradata\practice\ARCHIVE";
ALTER SYSTEM SET log_archive_dest="D:\oracle\oradata\practice\ARCHIVE"
*
第 1 行出现错误:
ORA-02097: 无法修改参数，因为指定的值无效
ORA-16018: 无法将 LOG_ARCHIVE_DEST 与 LOG_ARCHIVE_DEST_n 或
DB_RECOVERY_FILE_DEST 一起使用
```

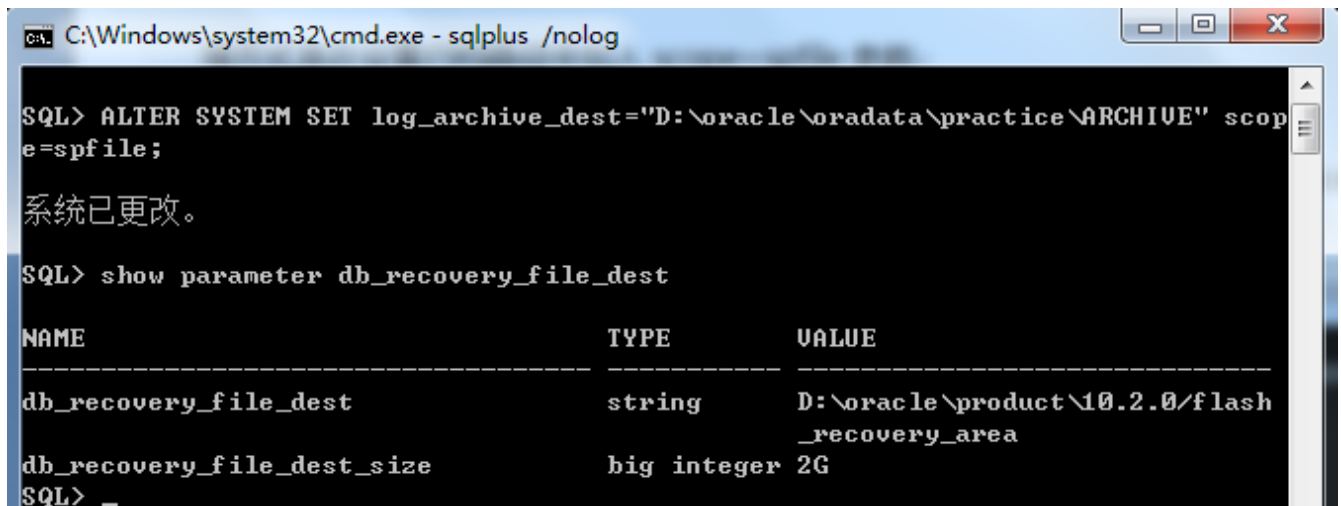
该问题是数据库开启了闪回功能，归档文件默认情况下是保存到闪回路径中，简单的处理方式是在设置归档路径中加入 scope=spfile 参数

--设置归档日志文件路径

```
ALTER SYSTEM SET log_archive_dest="D:\oracle\oradata\practice\ARCHIVE"
scope=spfile;
```

```
C:\Windows\system32\cmd.exe - sqlplus /nolog
SQL> ALTER SYSTEM SET log_archive_dest="D:\oracle\oradata\practice\ARCHIVE" scop
e=spfile;
系统已更改。
SQL>
```

此时查看闪回路径，该路径并未影响，只不过闪回文件和归档文件保存到各自文件夹中



```
C:\Windows\system32\cmd.exe - sqlplus /nolog

SQL> ALTER SYSTEM SET log_archive_dest='D:\oracle\oradata\practice\ARCHIVE' scope=spfile;

系统已更改。

SQL> show parameter db_recovery_file_dest

NAME                                TYPE                                VALUE
-----                                -
db_recovery_file_dest                string                              D:\oracle\product\10.2.0\flash
_recovery_area
db_recovery_file_dest_size           big integer                          2G
SQL>
```

4.2 LogMiner 相关资料

参见 Oracle 官方站点对 LogMiner 介绍，地址如下：

http://docs.oracle.com/cd/E11882_01/server.112/e22490/logminer.htm