



DOAG 2005

ORACLE® LogMiner

Einleitung

- ◆ **Oracle's LogMiner bietet seit der Version 8i die Möglichkeit, die bereits ausgeführten und in den Redologdateien gespeicherten Transaktionen wieder sichtbar zu machen.**
- ◆ **Dazu liest er die Redo-Informationen aus den Online Redologs oder Archivelogs aus und stellt die Befehle wieder in lesbarer Form zur Verfügung**

Anwendungsfälle

- ◆ Fehlerhafte Transaktionen anzeigen bzw. korrigieren
- ◆ Zeitpunkt der fehlerhaften Transaktion ermitteln
- ◆ Person/Clientrechner identifizieren, die den Fehler begangen hat
- ◆ Audit/Dokumentation von Veränderungen in wichtigen Tabellen
- ◆ Möglichkeit der Auswertung von fremden Redologs (auf Test-Instanz die Redologs der Produktiv-Instanz analysieren)

Vorbereitungen

- ◆ **Der LogMiner speichert nicht die Originalnamen der Tabellen und Spalten in den Redologs, sondern nur deren Referenznummer aus dem Data Dictionary.**
- ◆ **Daraus ergibt sich das Problem, dass eine Mapping Tabelle benutzt werden muss, die die Nummern wieder in Objektnamen umsetzen kann.**
- ◆ **Drei verschiedene Möglichkeiten die Mapping Tabelle zu verwenden stehen zur Verfügung:**
 - ▶ **Extraktion in externe Datei (Flatfile)**
 - ▶ **Speichern der Daten in die Redologs**
 - ▶ **Online-Analyse mit Hilfe des Data Dictionary**

Anwendungsmöglichkeiten

	DD Infos in Datei (ab 8i)	DD Infos in Redologs (ab 9i)	DD Infos aus Online DD (ab 9i)
Vorteile:	Belegt keinen Speicherplatz in den Redologs	Ist schneller als Extraktion in Datei	Erfordert keinen Export der Daten (schnellste Variante)
		Zeitbezug durch Redolog-Zeitstempel sichergestellt	
		Kein Extra-Backup notwendig	Kein Extra Backup notwendig
Nachteile:	Zeitpunkt der Erstellung muss gemerkt werden	Zusätzlicher Aufwand in den Redologs	Gelöschte Objekte sind nicht nachvollziehbar
	Backup der Dateien notwendig		
	Langsam		

Externe Data Dictionary Mapping Tabelle

- ◆ **Setzen Sie dazu den Parameter utl_file_dir auf ein Verzeichnis z.B.:**
- ◆ **utl_file_dir=c:\temp**
- ◆ **oder bei vorhandenen SPFILE:**
 - ▶ **ALTER SYSTEM SET utl_file_dir='c:\temp'
SCOPE=SPFILE;**
- ◆ **Hinweis:**
 - ▶ **Das utl_file_dir Verzeichnis sollte ein unbedenkliches Verzeichnis sein, da mittels des Package utl_file aus diesem Verzeichnis gelesen bzw. auch darin geschrieben werden kann!**
 - ▶ **Die Datenbank muss nach einer Änderung des Parameters neu durchgestartet werden.**

Externe Data Dictionary Mapping Tabelle

- ◆ Erzeugen Sie anschließend eine Data Dictionary Datei (Version 8i):

- ◆

```
BEGIN
DBMS_LOGMNR_D.BUILD (
DICTIONARY_FILENAME=>'dict.ora',
DICTIONARY_LOCATION => 'c:\temp');
END;
```

- ◆ Oder für die Version 9i:

- ◆

```
BEGIN
DBMS_LOGMNR_D.BUILD (
DICTIONARY_FILENAME=>'dict.ora',
DICTIONARY_LOCATION => 'c:\temp',
OPTIONS=>dbms_logmnr_d.store_in_flat_file);
END;
```

Externe Data Dictionary Mapping Tabelle

◆ Hinweis:

- ▶ Der Defaultwert für **OPTIONS** ist **dbms_logmnr_d.store_in_flat_file** und könnte im obigen Beispiel weggelassen werden.

◆ Jetzt werden die gewünschten Redolog- oder Archivelog-Dateien angegeben, in denen sich die gesuchten Informationen befinden:

◆ BEGIN

```
DBMS_LOGMNR.ADD_LOGFILE  
(LOGFILENAME=>'c:\ora\oradata\o10g\redo10g\redo10g01.log',  
OPTIONS=>dbms_logmnr.NEW);  
END;
```

Externe Data Dictionary Mapping Tabelle

◆ Danach Start des LogMiners:

- ▶ `EXECUTE DBMS_LOGMNR.START_LOGMNR(
DICTFILENAME => 'c:\temp\dict.ora');`

◆ Abfrage der gewünschten Daten aus v\$logmnr_contents:

- ▶ `SELECT FROM v$logmnr_contents;`

◆ LogMiner Session beenden:

- ▶ `EXEC dbms_logmnr.end_logmnr;`

Extrahieren der Mapping-Infos in Redologs

- ◆ Extrahieren der Mapping-Informationen in die Redologdateien

- ◆ BEGIN

```
dbms_logmnr_d.build(  
options => dbms_logmnr_d.store_in_redo_logs);  
END;
```

- ◆ Voraussetzungen beim Starten des LogMiners mit der Option **DICT_FROM_REDO_LOGS**:

- ◆ Datenbank muss **OPEN** + im **Archive Log Modus** sein

- ◆ + **Supplemental Logging** wurde aktiviert:

- ◆ **ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;**

Extrahieren der Mapping-Infos in Redologs

- ◆ Wenn man den LogMiner ausführt, muss man hierbei außer den zu analysierenden Logs, auch die Logs mit angeben, die das Dictionary beinhalten.
- ◆ Die erste und letzte Redolog-Datei mit Dictionary anzeigen:

```
▶ SELECT
dictionary_begin,dictionary_end,sequence#,name
FROM V$ARCHIVED_LOG
WHERE DICTIONARY_BEGIN='YES'
OR DICTIONARY_END='YES' ;
```

Extrahieren der Mapping-Infos in Redologs

◆ Hinweis:

- ▶ Alle Redologdateien zwischen Anfang und Ende müssen übergeben werden.

◆ BEGIN

```
DBMS_LOGMNR.ADD_LOGFILE  
(LOGFILENAME=>'c:\ora\oradata\o10g\redo3.log'  
,OPTIONS=>dbms_logmnr.NEW) ;  
END ;
```

Zeitbereiche angeben (ab 10g)

- ◆ Ab Version 10g kann man einen Zeitbereich angeben, der analysiert werden soll:

- ◆ BEGIN

```
dbms_logmnr.start_logmnr(  
STARTTIME => '10-NOV-2005 10:15:00',  
ENDTIME => '10-NOV-2005 12:45:00',  
options=>dbms_logmnr.dict_from_redo_logs  
+DBMS_LOGMNR.CONTINUOUS_MINE);  
END;
```

- ◆ Oracle sucht sich hier die passenden Redologdateien selbst zusammen

Extrahieren der Mapping-Infos in Redologs

- ◆ **Abfrage der gewünschten Daten aus v\$logmnr_contents:**
- ◆ **SELECT FROM v\$logmnr_contents;**
- ◆ **LogMiner Session beenden:**
- ◆ **EXEC dbms_logmnr.end_logmnr;**

Verwenden eines Online Data Dictionary

◆ Übergeben der Namen der Redolog-Dateien:

```
▶ BEGIN
  DBMS_LOGMNR.ADD_LOGFILE
  (LOGFILENAME=>'c:\oracle\oradata\o10g2\redolog
  03.log',
  OPTIONS=>dbms_logmnr.NEW);
END;
```

◆ Start des LogMiners:

```
▶ BEGIN
  dbms_logmnr.start_logmnr
  (options=>dbms_logmnr.dict_from_online_catalog
  );
END;
```

Verwenden eines Online Data Dictionary

◆ Abfrage der gewünschten Daten aus v\$logmnr_contents:

▶ `SELECT FROM v$logmnr_contents;`

◆ LogMiner Session beenden:

▶ `EXEC dbms_logmnr.end_logmnr;`

Einsatz des LogMiners

- ◆ Sie können mehrere Redologdateien bzw. archivierte Redologs angeben:

- ◆ BEGIN

```
DBMS_LOGMNR.ADD_LOGFILE
```

```
(LOGFILENAME=>'c:\oracle\oradata\o10g2\redo1.log',  
OPTIONS=>dbms_logmnr.NEW);
```

```
DBMS_LOGMNR.ADD_LOGFILE
```

```
(LOGFILENAME=>'c:\oracle\oradata\o10g2\redo2.log',  
OPTIONS=>dbms_logmnr.ADD);
```

```
DBMS_LOGMNR.ADD_LOGFILE
```

```
(LOGFILENAME=>'c:\oracle\oradata\o10g2\redo3.log',  
OPTIONS=>dbms_logmnr.ADD);
```

```
END;
```

Entfernen von Redologs aus der Session

- ◆ Redologdateien können wieder aus der LogMiner-Session entfernt werden:

- ◆

```
BEGIN
DBMS_LOGMNR.REMOVE_LOGFILE (
LOGFILENAME => 'c:\oracle\logs\log2.f' );
END;
/
```

Beispiele

- ◆ `CREATE TABLE scott.emp2
AS
SELECT empno,ename,deptno FROM scott.emp
WHERE empno IN (7369,7782,7499);`
- ◆ `DELETE FROM scott.emp2 WHERE deptno=10;`
- ◆ `UPDATE scott.emp2 SET ename='Hr. ' ||ename
WHERE deptno=20;`
- ◆ `INSERT INTO scott.emp2
VALUES (8000, 'PATZWahl', 30);`
- ◆ `COMMIT;`

Beispiele

- ◆ `connect sys/sys as sysdba`
- ◆ `BEGIN`
- ◆ `dbms_logmnr.ADD_LOGFILE (`
`'D:\ORACLE\ORADATA\O10g\REDO04.LOG' ,`
`options=>dbms_logmnr.new) ;`
- ◆ `dbms_logmnr.start_logmnr (`
`options=>dbms_logmnr.dict_from_online_catalog`
`+dbms_logmnr.PRINT_PRETTY_SQL) ;`
- ◆ `END ;`
- ◆ `/`

Beispiele

◆ `SELECT scn,sql_redo FROM sys.v$logmnr_contents
WHERE seg_owner='SCOTT'
AND timestamp between sysdate-1/96 and sysdate;`

◆ `SCN SQL_REDO`

◆ -----

◆ `3588137 CREATE TABLE scott.emp2 AS SELECT empno,ename,deptno
FROM scott.emp WHERE empno IN (7369,7782,7499);`

◆ `3588139 insert into "SCOTT"."EMP2" ("EMPNO", "ENAME", "DEPTNO")
values ('7369', 'SMITH', '20');`

◆ `3588139 insert into "SCOTT"."EMP2" ("EMPNO", "ENAME", "DEPTNO")
values ('7499', 'ALLEN', '30');`

◆ `3588139 insert into "SCOTT"."EMP2" ("EMPNO", "ENAME", "DEPTNO")
values ('7782', 'CLARK', '10');`

◆ `3588149 delete from "SCOTT"."EMP2" where "EMPNO" = '7782'
and "ENAME" = 'CLARK' and "DEPTNO" = '10'
and ROWID = 'AAAHhvAAGAAAAXMAAC';`

Beispiele

◆ `SELECT scn,sql_redo FROM sys.v$logmnr_contents
WHERE seg_owner='SCOTT'
AND timestamp between sysdate-1/96 and sysdate;`

◆ `SCN SQL_UNDO`

3588139 delete from "SCOTT"."EMP2" where "EMPNO" =
'7369' and "ENAME" = 'SMITH' and "DEPTNO" = '20' and
ROWID = 'AAAHhvAAGAAAAXMAAA';

◆ 3588139 delete from "SCOTT"."EMP2" where "EMPNO" =
'7499' and "ENAME" = 'ALLEN' and "DEPTNO" = '30' and
ROWID = 'AAAHhvAAGAAAAXMAAB';

◆ 3588139 delete from "SCOTT"."EMP2" where "EMPNO" =
'7782' and "ENAME" = 'CLARK' and "DEPTNO" = '10' and
ROWID = 'AAAHhvAAGAAAAXMAAC';

◆ ...

DDL Kommandos im Logminer

- ◆ **TRUNCATE TABLE scott.emp2 ;**
 - ▶ **SQL_REDO** wird angezeigt, **SQL_UNDO** nicht !
- ◆ **ALTER TABLE scott.emp2 ADD CONSTRAINT emp2_pk PRIMARY KEY (empno) ;**
 - ▶ **Darstellung in der SQL_Redo Spalte:**
 - ▶ **CREATE UNIQUE INDEX "SCOTT"."EMP2_PK" on "SCOTT"."EMP2" ("EMPNO") NOPARALLEL ;**
 - ▶ **ALTER TABLE scott.emp2 ADD CONSTRAINT emp2_pk PRIMARY KEY (empno) ;**

DDL Kommandos im Logminer

◆ **ALTER TABLE scott.emp2 DROP CONSTRAINT emp2_pk;**

▶ **SQL_Redo Spalte:**

▶ **drop index "SCOTT"."EMP2_PK";**

▶ **ALTER TABLE scott.emp2 DROP CONSTRAINT emp2_pk;**

◆ **DROP USER SCOTT cascade;**

▶ **SQL_Redo Spalte:**

▶ **drop table "SCOTT"."BONUS" cascade constraints force;**

▶ **drop table "SCOTT"."DEPT" cascade constraints force;**

▶ **drop table "SCOTT"."EMP" cascade constraints force;**

▶ **DROP USER SCOTT cascade;**

Optionen des LogMiners

◆ COMMITTED_DATA_ONLY

- ▶ Es werden bei der Analyse nur abgeschlossene Transaktionen betrachtet
- ▶ Offene oder zurückgerollte Transaktionen werden nicht angezeigt

◆ Beispiel:

- ▶

```
BEGIN
dbms_logmnr.start_logmnr
(options=>dbms_logmnr.committed_data_only);
END;
```

Optionen des LogMiners

◆ SKIP_CORRUPTION

- ▶ Überspringen von Log-Korruption:
- ▶ Nur möglich, wenn ein Block in der Redologdatei defekt ist und nicht der Header des Redologs.
- ▶

```
BEGIN dbms_logmnr.start_logmnr  
  (option=>dbms_logmnr.skip_corruption);  
END;
```
- ▶ Korrupte Zeilen werden in der View `V$LOGMNR_CONTENTS` Spalte `INFO` markiert

Optionen des LogMiners

◆ DDL_DICT_TRACKING

- ▶ Nicht bei Online Data Dictionary möglich
- ▶ Ab Oracle9i kann LogMiner sein Dictionary aktualisieren. Dies wird beim Start des LogMiners aktiviert.
- ▶ DDL-Anweisungen werden in den Redologs aufgezeichnet und können mit dem LogMiner veranschaulicht werden:
- ▶

```
SELECT username, operation, sql_redo
FROM v$logmnr_contents WHERE username = 'SCOTT'
AND operation = 'DDL';
```
- ▶

```
USERNAME OPERATION          SQL_REDO
-----
SCOTT      DDL              alter table emp add bonus number
```
- ▶

```
BEGIN dbms_logmnr.start_logmnr -
(options=>dbms_logmnr.dict_from_redo_logs +
dbms_logmnr.ddl_dict_tracking); END;
```

Optionen des LogMiners

◆ NO_SQL_DELIMITER

- ▶ Es wird kein Semikolon am Ende des SQL Befehls angezeigt. Damit kann in einem PL/SQL Cursor (EXECUTE IMMEDIATE) der Befehl leichter abgesetzt werden.

◆ NO_ROWID_IN_STMT

- ▶ Es wird keine ROWID im SQL-Statement angezeigt. Damit wird ein Absetzen des SQL-Befehls auch auf einer Remote Datenbank erleichtert.

◆ PRINT_PRETTY_SQL

- ▶ Bessere Formatierung der SQL-Befehle. Diese können jedoch nicht abgesetzt werden.

◆ CONTINUOUS_MINE (ab 10g)

- ▶ LogMiner nimmt sich automatisch die zusätzlichen archivierten Redologs, wenn ein Auswertungszeitraum angegeben wurde.

Optionen des LogMiners

◆ MINE_VALUE (ab 10g)

- ▶ Kann Zeilen noch genauer aus den Redologs herausfiltern, die z.B. folgende Fragestellung erfüllen:
- ▶ Welche Transaktionen haben das Gehalt um mehr als 50 % erhöht?

```
◆ SELECT sql_redo FROM v$logmnr_contents
WHERE operation = 'UPDATE"
AND owner_name = 'SCOTT' AND seg_name = 'EMP'
AND
dbms_logmnr.mine_value(redo_value, 'SCOTT.EMP.SAL')
>1.5*dbms_logmnr.mine_value(undo_value, 'SCOTT.EMP.
SAL' );
```

Tipps & Tricks

- ◆ **Lob/Longs Spalten werden ab einer Textlänge von ca. 3800 Bytes nicht mehr dargestellt.**
- ◆ **Bei großen Redologdateien kann ein Full Tablespace auf v\$logmnr_contents Stunden benötigen. Ideal ist hier ein Standby oder Testsystem zur Auswertung.**
- ◆ **In der Version 8i werden Drop Table Befehle als ein DELETE auf TAB\$ dargestellt.**
- ◆ **Wenn Sie mit dem Dictionary File arbeiten, sollten Sie regelmäßig eine neue Version davon erzeugen**

VerwaltungsvIEWS

- ◆ **V\$LOGMNR_CONTENTS** Inhalt der Auswertung der Redologs
- ◆ **V\$LOGMNR_DICTIONARY** Die benutzte Dictionary Datei
- ◆ **V\$LOGMNR_LOG** Übersicht der analysierten Redologdateien
- ◆ **V\$LOGMNR_PARAMETERS** Aktuelle Parametereinstellungen des Logminers

Spalten des Logminers

◆ SQL> desc v\$logmnr_contents

◆ Name

SCN
CSCN
TIMESTAMP
COMMIT_TIMESTAMP
THREAD#
LOG_ID
XIDUSN
XIDSLT
XIDSQN
PXIDUSN
PXIDSLT
PXIDSQN
RBASQN
RBABLK
RBABYTE
UBAFIL
UBABLK
UBAREC
UBASQN
ABS_FILE#
REL_FILE#
DATA_BLK#
DATA_OBJ#
DATA_OBJD#
SEG_OWNER
SEG_NAME
TABLE_NAME
SEG_TYPE

◆ SEG_TYPE_NAME
TABLE_SPACE
ROW_ID
SESSION#
SERIAL#
USERNAME
SESSION_INFO
TX_NAME
ROLLBACK
OPERATION
OPERATION_CODE
SQL_REDO
SQL_UNDO
RS_ID
SEQUENCE#
SSN
CSF
INFO
STATUS
REDO_VALUE
UNDO_VALUE
SQL_COLUMN_TYPE
SQL_COLUMN_NAME
REDO_LENGTH
REDO_OFFSET
UNDO_LENGTH
UNDO_OFFSET
DATA_OBJV#
SAFE_RESUME_SCN
XID
PXID
AUDIT_SESSIONID

Impressum

- ◆ **Oracle Schulung (SQL, DBA, PL/SQL, HTMLDB,...)**
- ◆ **Oracle Consulting & Support**
- ◆ **Oracle Entwicklung & Lizenzvertrieb**
- ◆ **Marco Patzwahl**
MuniQSoft GmbH
Grünwalder Weg 13 a
D-82008 Unterhaching b. München
- ◆ **Telefon: +49(0)89-67909040**
Fax: +49(0)89-67909050
E-Mail m.patzwahl@muniqsoft.de
Internet: www.muniqsoft.de

