

LogMiner – Além do FlashBack Query

O Oracle LogMiner é uma ferramenta que permite consultas a arquivos de redo log's online e archives através de SQL. Os arquivos de redo contém o histórico da atividades em um banco de dados. Isso já existia no Oracle8i

O Oracle também possui a tecnologia de *Flashback Query* (introduzida no 9i) que permite ver o valor de uma coluna em um ponto no passado, antes que este valor tenha sido alterado ou gravado. No entanto isso só vai funcionar se a alteração ocorreu dentro do período determinado por **UNDO_RETENTION**, cujo default de 15 minutos. Muitas vezes esta informação não pode ser recuperada, ou por que o tempo expirou, ou por que o segmentos de undo foram re-utilizados. Abaixo tem um pequeno exemplo de como utilizar o logminer quando não temos a possibilidade de usar o recurso de flashback

```
SQL> SELECT to_char(sysdate, 'hh24:mi:ss') from dual;
```

```
TO_CHAR(  
-----  
13:53:41
```

```
SQL> select * from v$log;
```

GROUP#	THREAD#	SEQUENCE#	BYTES	MEMBERS	ARC	STATUS	FIRST_CHANGE#	FIRST_TI
1	1	83	52428800	1	YES	INACTIVE	952754	30/08/09
2	1	84	52428800	1	NO	CURRENT	972768	30/08/09
3	1	82	52428800	1	YES	INACTIVE	950683	30/08/09

```
SQL> select * from dept;
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resources	203	2400
50	Shipping	121	1500
60	IT	103	1400

6 linhas selecionadas

```
SQL> delete from dept;
```

6 linhas deletadas.

```
SQL> commit;
```

Commit concluído.

```
SQL> select * from dept;
```

não há linhas selecionadas

Como podemos ver as linhas não existem mais. No entanto as imagens anteriores da DML ainda estão na tablespace de undo e podem ser utilizadas com a **Flashback Query** ou com **Flashback Table To Timestamp**.

```
SQL> select * from dept as of timestamp systimestamp - interval '1' minute;
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resources	203	2400
50	Shipping	121	1500
60	IT	103	1400

6 linhas selecionadas.

Como podemos notar as “before images” ainda estão na **TS de Undo**. Mas e se um transação as sobrescrevesse ? .

Vou executar uma DML paralela que vai usar todos os segmento de rollback da TS de Undo

```
SQL> conn sh/sh
```

Conectado.

```
SQL> create table big as select * from sales;
```

Tabela criada.

```
SQL>
```

```
SQL> alter session enable parallel dml;
```

Sessão alterada.

```
SQL> update /*+ parallel (big,10) */ big set amount_sold=amount_sold;
```

```
update /*+ parallel (big,10) */ big set amount_sold=amount_sold
```

```
*
```

ERRO na linha 1:

ORA-12801: erro assinalado no servidor de consulta paralela P009

ORA-30036: não foi possível estender o segmento em 8 no tablespace de undo 'UNDOTBS1'

Sem ter como acessar as imagens anteriores ao delete, tanto **Flashback Query** como **Flashback Table** não funcionam mais.

```
SQL> conn geddy/geddy
```

Conectado.

```
SQL> select * from dept as of timestamp to_timestamp('2009-08-30:13:21:11', 'yyyy-mm-dd:hh24:mi:ss')
```

```
);
```

```
select * from dept as of timestamp to_timestamp('2009-08-30:13:21:11', 'yyyy-mm-dd:hh24:mi:ss')
```

```
*
```

ERRO na linha 1:

ORA-01555: instantâneo muito antigo: número de segmento de rollback 18 com nome "_SYSSMU18\$" muito

```
SQL> alter table dept enable row movement;
```

Tabela alterada.

```
SQL> flashback table dept to timestamp to_timestamp('2009-08-30:13:21:00', 'yyyy-mm-dd:hh24:mi:ss');
```

```
flashback table dept to timestamp to_timestamp('2009-08-30:13:21:00', 'yyyy-mm-dd:hh24:mi:ss')
```

```
*
```

ERRO na linha 1:

ORA-00604: ocorreu um erro no nível 1 SQL recursivo

ORA-12801: erro assinalado no servidor de consulta paralela P000
ORA-01555: instantâneo muito antigo: número de segmento de rollback 18 com nome "_SYSSMU18\$" muito

Utilizar Flashback ficou impossível. Também não tenho flashback logs, portanto não posso utilizar o Flashback Database. Poderia fazer um TSPITR agora, mas não tenho tempo ☺. Agora é a vez do LogMiner entrar em ação.

```
SQL> select * from v$log;
```

GROUP#	THREAD#	SEQUENCE#	BYTES	MEMBERS	ARC	STATUS	FIRST_CHANGE#	FIRST_TI
1	1	86	52428800	1	NO	CURRENT	979188	30/08/09
2	1	84	52428800	1	YES	ACTIVE	972768	30/08/09
3	1	85	52428800	1	YES	ACTIVE	976378	30/08/09

Neste momento estou na sequence# 86. O caso é que eu sei exatamente qual sequence# de archive eu preciso investigar. Achar quando exatamente o erro lógico aconteceu é a parte mais difícil do processo

```
SQL> select name from v$archived_log where sequence#=84;
```

NAME

```
-----  
C:\ORACLE\FLASH_RECOVERY_AREA\DB10G\ARCHIVELOG\2009_08_30\01_MF_1_84_59OCS2PG_.ARC
```

```
SQL> begin
```

```
2 dbms_logmnr.add_logfile  
3 ('C:\ORACLE\FLASH_RECOVERY_AREA\DB10G\ARCHIVELOG\2009_08_30\01_MF_1_84_59OCS2PG_.ARC',  
4 dbms_logmnr.new);  
5 end;  
6 /
```

Procedimento PL/SQL concluído com sucesso.

```
SQL> execute dbms_logmnr.start_logmnr(options=>DBMS_LOGMNR.DICT_FROM_ONLINE_CATALOG);
```

Procedimento PL/SQL concluído com sucesso.

```
SQL> select sql_undo , ' -- ', sql_redo from v$logmnr_contents where seg_name='DEPT';
```

```
SQL_UNDO          ' -- ' SQL_REDO  
-----  
insert into "GEDDY"."DEPT"("DEPA -- delete from "GEDDY"."DEPT" where  
RTMENT_ID","DEPARTMENT_NAME","MA "DEPARTMENT_ID" = '10' and "DEP  
NAGER_ID","LOCATION_ID") values  ARTMENT_NAME" = 'Administration'  
( '10','Administration','200','17 and "MANAGER_ID" = '200' and "L  
00'); OCATION_ID" = '1700' and ROWID =  
'AAAM1QAEEAAAD48AAG';  
  
insert into "GEDDY"."DEPT"("DEPA -- delete from "GEDDY"."DEPT" where  
RTMENT_ID","DEPARTMENT_NAME","MA "DEPARTMENT_ID" = '20' and "DEP  
NAGER_ID","LOCATION_ID") values  ARTMENT_NAME" = 'Marketing' and  
( '20','Marketing','201','1800'); "MANAGER_ID" = '201' and "LOCATI  
ON_ID" = '1800' and ROWID = 'AAA  
M1QAEEAAAD48AAH';  
  
insert into "GEDDY"."DEPT"("DEPA -- delete from "GEDDY"."DEPT" where  
RTMENT_ID","DEPARTMENT_NAME","MA "DEPARTMENT_ID" = '30' and "DEP  
NAGER_ID","LOCATION_ID") values  ARTMENT_NAME" = 'Purchasing' and  
( '30','Purchasing','114','1700') "MANAGER_ID" = '114' and "LOCAT  
ION_ID" = '1700' and ROWID = 'AA  
AM1QAEEAAAD48AAI';  
  
insert into "GEDDY"."DEPT"("DEPA -- delete from "GEDDY"."DEPT" where  
RTMENT_ID","DEPARTMENT_NAME","MA "DEPARTMENT_ID" = '40' and "DEP
```

```

NAGER_ID", "LOCATION_ID") values          ARTMENT_NAME" = 'Human Resources
('40', 'Human Resources', '203', '2    ' and "MANAGER_ID" = '203' and "
400');                                  LOCATION_ID" = '2400' and ROWID
                                        = 'AAAM1QAEEAAD48AAJ';

insert into "GEDDY"."DEPT"("DEPA      -- delete from "GEDDY"."DEPT" where
RTMENT_ID", "DEPARTMENT_NAME", "MA    "DEPARTMENT_ID" = '50' and "DEP
NAGER_ID", "LOCATION_ID") values       ARTMENT_NAME" = 'Shipping' and "
('50', 'Shipping', '121', '1500');    MANAGER_ID" = '121' and "LOCATIO
                                        N_ID" = '1500' and ROWID = 'AAAM
                                        1QAEEAAD48AAK';

insert into "GEDDY"."DEPT"("DEPA      -- delete from "GEDDY"."DEPT" where
RTMENT_ID", "DEPARTMENT_NAME", "MA    "DEPARTMENT_ID" = '60' and "DEP
NAGER_ID", "LOCATION_ID") values       ARTMENT_NAME" = 'IT' and "MANAGE
('60', 'IT', '103', '1400');         R_ID" = '103' and "LOCATION_ID"
                                        = '1400' and ROWID = 'AAAM1QAEEA
                                        AAD48AAL';

```

6 linhas selecionadas.

Poiso ainda tornar a coluna sql_undo um pouco mais legível com a opção print_pretty_sql. Além disso quero executar os comandos em um loop PL/SQL, utilizando sql dinâmico e essa abordagem requer que ponto e vírgula no final de cada comando seja removido. A opção no_sql_delimiter resolve isso pra mim :

```

SQL> begin
2     dbms_logmnr.start_logmnr(
3         options =>
4         dbms_logmnr.
5         dict_from_online_catalog +
6         dbms_logmnr.print_pretty_sql +
7         dbms_logmnr.no_sql_delimiter
8     );
9 end;
10 /

```

Procedimento PL/SQL concluído com sucesso.

Após este comando a sql_undo pode ser lido assim :

```

SQL> select sql_undo from v$logmnr_contents where seg_name='DEPT';

```

```

SQL_UNDO
-----
insert into "GEDDY"."DEPT"
values
  "DEPARTMENT_ID" = 10,
  "DEPARTMENT_NAME" = 'Adminis
tration',
  "MANAGER_ID" = 200,
  "LOCATION_ID" = 1700

insert into "GEDDY"."DEPT"
values
  "DEPARTMENT_ID" = 20,
  "DEPARTMENT_NAME" = 'Marketi
ng',
  "MANAGER_ID" = 201,
  "LOCATION_ID" = 1800

insert into "GEDDY"."DEPT"
values

```

```

        "DEPARTMENT_ID" = 30,
        "DEPARTMENT_NAME" = 'Purchasing',
        "MANAGER_ID" = 114,
        "LOCATION_ID" = 1700

insert into "GEDDY"."DEPT"
values
    "DEPARTMENT_ID" = 40,
    "DEPARTMENT_NAME" = 'Human Resources',
    "MANAGER_ID" = 203,
    "LOCATION_ID" = 2400

insert into "GEDDY"."DEPT"
values
    "DEPARTMENT_ID" = 50,
    "DEPARTMENT_NAME" = 'Shipping',
    "MANAGER_ID" = 121,
    "LOCATION_ID" = 1500

insert into "GEDDY"."DEPT"
values
    "DEPARTMENT_ID" = 60,
    "DEPARTMENT_NAME" = 'IT',
    "MANAGER_ID" = 103,
    "LOCATION_ID" = 1400

```

6 linhas selecionadas.

```
SQL> select * from dept;
```

não há linhas selecionadas

Para finalizar basta apenas executar o bloco abaixo para que tenhamos as linhas de volta na tabela dept

```

SQL> begin
2   for undo_rec in (
3       select sql_undo, row_id
4       from v$logmnr_contents
5       where table_name='DEPT' and seg_owner='GEDDY')
6       loop
7           begin
8               execute immediate undo_rec.sql_undo;
9           end;
10      end loop;
11  end;
12  /

```

Procedimento PL/SQL concluído com sucesso.

```
SQL> SELECT * FROM DEPT;
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resources	203	2400
50	Shipping	121	1500
60	IT	103	1400

6 linhas selecionadas.