



LogMiner et Flashback dans Oracle 11g R2

CREATION : 2011/05/13

MISE A JOUR : 2011/05/13

Ce tutoriel est une traduction adaptée et incomplète de l'article d'ORACLE-BASE :

http://www.oracle-base.com/articles/11g/FlashbackAndLogminerEnhancements_11gR1.php

La page officielle de LogMiner est :

http://download.oracle.com/docs/cd/E11882_01/server.112/e16536/logminer.htm

La possibilité de faire une requête flashback a été introduite avec Oracle 9i. Oracle 10g 11g ont amélioré et enrichi cette fonctionnalité. Notons également que le LogMiner a vu son ergonomie améliorée.

Améliorations de LogMiner

Dans les versions précédentes de la base de données Oracle, LogMiner était une console Java distincte. Depuis Oracle 11g, LogMiner a été intégré à l'OEM et rattaché à la fonctionnalité de flashback, ce qui rend très simple d'annuler des transactions qui ont eu des effets indésirables

LogMiner est accessible par l'OEM depuis Disponibilité > Sauvegarde/récupération > Visualiser et gérer les transactions. Voilà ce que cela donne

Instance de base de données: 1Z2 > Connecté en tant que SYS

Résultats LogMiner

Précédent Terminé

▼ Récapitulatif

Transactions correspondantes 6
Enregistrements de journalisation correspondants 6
Filtre d'interrogation where seg_owner = 'LOGTEST' and table_name = 'TEST_TAB'
Temps total 3 secondes

Les résultats affichent les transactions qui contiennent des enregistrements de journalisation correspondant au filtre d'interrogation. Les transactions peuvent contenir d'autres enregistrements de journalisation. Cliquez sur un ID pour visualiser tous les enregistrements de journalisation de la transaction correspondante. Vous pouvez filtrer davantage les résultats en effectuant des recherches dans les infos de journalisation SQL.

Résultats de la transaction

Effectuer une recherche dans les infos de journalisation SQL Exécuter Visualiser Récapitulatif de transaction

ID de transaction	Utilisateur de base de données	Horodatage de validation (commit)	Enregistrements de journalisation (sup), autres (aut)	Récapitulatif de transaction - Mises à jour (maj), Insertions (ins), journalisation suppressions (sup), autres (aut)
01000001E030000	LOGTEST			1 create table test_tab (id number, description varchar2(50), constraint test_tab_pk primary key(id));
07000A04D030000	LOGTEST	13 mai 2011 13:23:53		1 LOGTEST TEST_TAB (1 ms)
04001A002C030000	LOGTEST	13 mai 2011 13:23:53		1 LOGTEST TEST_TAB (1 ms)
0A00070024030000	LOGTEST	13 mai 2011 13:23:53		1 LOGTEST TEST_TAB (1 ms)
01000001E030000	LOGTEST	13 mai 2011 13:23:53		1 LOGTEST TEST_TAB (1 ms)
05000301D060000	LOGTEST	13 mai 2011 13:23:58		1 LOGTEST TEST_TAB (1 svp)

CONSEIL Le récapitulatif de la transaction affiche les premières tables modifiées par la transaction, avec le nombre d'instructions INSERT, DELETE et UPDATE correspondant au filtre d'interrogation.

Précédent Terminé

Comme pour les versions précédentes d'Oracle, notons que le LogMiner reste disponible par le package PL/SQL DBMS_LOGMNR (il est un peu moins gourmand en ressources que l'OEM)



Transaction flashback

Comme son nom l'indique, une transaction flashback permet de défaire les changements d'une transaction, éventuellement en incluant également les changements opérés par des transactions dépendantes. Voyons ce que cela peut donner ...

LogMiner a besoin que la base soit en mode **archivelog** et en mode **flashback**, la **journalisation supplémentaire minimale** doit également être activée. Ces options peuvent être activées par l'OEM. Passer en archivelog et/ou flashback requiert de redémarrer la base.

```
startup mount ;
alter database archivelog;
alter database flashback on;
alter database open;

alter database add supplemental log data;
% pour vérifier si c'est déjà le cas :
% select supplemental_log_data_min from v$database renvoie yes
alter database add supplemental log data (primary key) columns;
```

ORACLE Enterprise Manager 11g Database Control

Instance de base de données: 1Z2 > Paramètres de récupération

Récupération de l'instance

Pour activer la fonctionnalité de points de reprise Fast-Start, indiquez une valeur différente de zéro pour la durée de récupération souhaitée, qui sera utilisée pour définir le paramètre d'initialisation FAST_START_MTR_TARGET. Ce paramètre régit le temps nécessaire à la base de données pour effectuer la récupération après une panne pour une seule instance. Lorsque la fonctionnalité de points de reprise Fast-Start est activée, Oracle adapte automatiquement la vitesse des points de reprise de façon à attendre la durée de récupération souhaitée. Avec la valeur 0, cette fonctionnalité est désactivée.

Durée de récupération estimée en cours (secondes) 47
Durée de récupération souhaitée 0 Minutes

Récupération après défaillance matérielle

La base de données est actuellement en mode NOARCHIVELOG. En mode ARCHIVELOG, vous pouvez effectuer des sauvegardes à chaud et récupérer les données les plus récentes. Vous devez fournir de l'espace pour les fichiers de journalisation archivés. Si vous tenez à passer la base de données en mode ARCHIVELOG, vous devez effectuer immédiatement une sauvegarde. En mode NOARCHIVELOG, vous ne pouvez effectuer que des sauvegardes à froid et des données peuvent être perdues si la base de données est endommagée.

Mode ARCHIVELOG

Format de nom du fichier d'archivage des journaux: ARC%S_%R.%T

Número	Destination des fichiers de journalisation archivés	Statut	Type
1	USE_DB_RECOVERY_FILE_DEST	VALID	Local

CONSEIL Il est recommandé d'écrire les fichiers de journalisation archivés à plusieurs emplacements, sur différents disques.
 CONSEIL Vous pouvez spécifier jusqu'à 10 destinations de fichier de journalisation archivés.

Activer la journalisation supplémentaire minimale

Récupération rapide

Cette base de données utilise une zone de récupération rapide. Le graphique indique l'espace utilisé par chaque type de

Si vous oubliez une des différentes options, vous aurez des erreurs plus tard ! Ce n'est pas grave, il suffit de corriger et de recommencer.

Voici quelques commandes à taper pour créer un nouvel utilisateur et faire quelques manipulations dans son schéma :

```
create user logtest identified by logtest quota unlimited on users;
grant connect, create table to logtest;
```



Regarder l'heure avant de lancer les commandes suivantes (c'est juste pour avoir une idée, hein !!)

```
% connexion utilisateur
connect logtest/logtest
```

```
create table test_tab (
  id      number,
  description varchar2(50),
  constraint test_tab_pk primary key(id)
);
insert into test_tab (id, description) values ('1', 'Description de 1');
commit;
insert into test_tab (id, description) values ('2', 'Description de 2');
commit;
update test_tab set description = 'nouvelle description de 1' where id=1;
commit;
update test_tab set description = 'nouvelle description de 2' where id=2;
commit;
delete from test_tab where id=1;
commit;
```

Se rendre dans le LogMiner et faire une recherche sur logtest.test_tab et cliquer sur Continuer.

Cela peut prendre un peu de temps, le temps que les fichiers de journalisation en ligne et archivés soient parcourus. L'écran suivant montre le résultat correspondant à ce que l'on vient de faire :



Instance de base de données : IZZI >

Connecté en tant que SYS

Résultats LogMiner

Précédent Terminé

▼ Récapitulatif

Transactions correspondantes 6 Filtre d'interrogation where seg_owner = 'LOGTEST' and table_name = 'TEST_TAB'
Enregistrements de journalisation correspondants 6 Temps total 3 secondes

Les résultats affichent les transactions qui contiennent des enregistrements de journalisation correspondant au filtre d'interrogation. Les transactions peuvent contenir d'autres enregistrements de journalisation. Cliquez sur un ID pour visualiser tous les enregistrements de journalisation de la transaction correspondante. Vous pouvez filtrer davantage les résultats en effectuant des recherches dans les infos de journalisation SQL.

Résultats de la transaction

Effectuer une recherche dans les infos de journalisation SQL Exécuter Visualiser par Récapitulatif de transaction

ID de transaction	Utilisateur de base de données	Horodatage de validation (commit)	Enregistrements de journalisation	Récapitulatif de transaction - Mises à jour (maj), insertions (ins), suppressions (sup), autres (aut)
01000001E030000	LOGTEST			1 create table test_tab (id number, description varchar2(50), constraint test_tab_pk primary key(id));
07000A004D030000	LOGTEST	13 mai 2011 13:23:53		1 LOGTEST.TEST_TAB (1 ms)
04001A0020030000	LOGTEST	13 mai 2011 13:23:53		1 LOGTEST.TEST_TAB (1 ms)
0A00070024030000	LOGTEST	13 mai 2011 13:23:53		1 LOGTEST.TEST_TAB (1 maj)
01000001E030000	LOGTEST	13 mai 2011 13:23:53		1 LOGTEST.TEST_TAB (1 maj)
050003001D050000	LOGTEST	13 mai 2011 13:23:58		1 LOGTEST.TEST_TAB (1 sup)

CONSEIL Le récapitulatif de la transaction affiche les premières tables modifiées par la transaction, avec le nombre d'instructions INSERT, DELETE et UPDATE correspondant au filtre d'interrogation.

Précédent Terminé

On peut alors choisir une transaction en cliquant dessus. Voici ce que l'on obtient :

Détails de la transaction

Faire un Flashback de la transaction Transaction précédente Transaction suivante OK

ID de transaction 07000A004D030000 Numéro SCN de début 1358306 Heure de début 13 mai 2011 13:23:53
Utilisateur de base de données UNIKKOVN SCN de validation (commit) 1358307 Heure de validation (commit) 13 mai 2011 13:23:53
Utilisateur OS Nom de l'ordinateur

SCN	Opération	Schéma	Table	Infos de journalisation SQL
1358306	START			set transaction read write;
1358306	INSERT	LOGTEST	TEST_TAB	insert into "LOGTEST"."TEST_TAB" values ('ID' = '1', 'DESCRIPTION' = 'Description de 1');
1358307	COMMIT			commit;

Faire un Flashback de la transaction Transaction précédente Transaction suivante OK

On peut alors revenir en arrière par rapport à cette transaction, mais cela peut avoir des répercussions sur d'autres transactions alors après confirmation, l'écran de dépendances apparaît :

Vous ne verrez pas forcément cet écran. En effet, une erreur **ORA-01291: missing logfile** peut être levée.



Si le choix par défaut NONCONFLICT_ONLY ne nous satisfait pas, il est possible d'en changer :

Exécuter l'interrogation Sélectionner une transaction Afficher les dépendances Récapitulatif

Afficher les dépendances : Changer d'option de récupération OK

Sans conflit uniquement
Seules les lignes non conflictuelles de la transaction cible seront annulées (back out).

Pas de cascade - Forcé
La transaction cible sera annulée (back out) de force sans tenir compte des transactions dépendantes.

Cascade
La transaction cible et toutes ses transactions dépendantes seront annulées (back out).

OK

Cliquer sur OK donne sur un ultime récapitulatif.

Exécuter l'interrogation Sélectionner une transaction Afficher les dépendances Récapitulatif

Transaction Flashback : Récapitulatif

Base de données IZZI Afficher le script SQL d'annulation Annuler Précédent Etape 4 sur 4 Fin

Type d'opération Transaction Flashback

Informations sur la transaction Flashback

ID de transaction 07000A004D030000
 Numéro SCN de début 1358306
 Heure de début
 Utilisateur de base de données UNKNOWN
 SCN de validation (commit) 1358307
 Heure de validation (commit)
 Utilisateur OS
 Nom de l'ordinateur
 Options de récupération Sans conflit uniquement

Exécuter le code SQL

Vous pouvez exécuter une interrogation SQL pour visualiser l'effet des modifications avant de valider (commit) la transaction Flashback.

Exécuter le code SQL

Afficher le script SQL d'annulation Annuler Précédent Etape 4 sur 4 Fin

Le bouton FIN revient effectivement en arrière.

Résultat de la transaction Flashback OK

La transaction sélectionnée, 07000A004D030000, a été annulée (back out).
 L'ID de la transaction de compensation est 06001B0066040000

Pour s'en convaincre, il suffit de refaire une recherche sur les logs ;-).

Là s'arrête la traduction de l'article qui est beaucoup plus long !!!!



On aurait pu tout à fait faire la même chose avec les packages DBMS_LOGMNR et DBMS_FLASHBACK. Voilà ce que cela aurait pu donner pour la consultation des logs :

```
ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MON-YYYY HH24:MI:SS';
```

```
EXECUTE DBMS_LOGMNR.START_LOGMNR( -
  STARTTIME => '13-MAI-2011 13:00:00', -
  ENDTIME => '13-MAI-2011 14:44:00', -
  OPTIONS => DBMS_LOGMNR.DICT_FROM_ONLINE_CATALOG + -
  DBMS_LOGMNR.CONTINUOUS_MINE -
);
```

```
SELECT OPERATION, SQL_REDO, SQL_UNDO FROM V$LOGMNR_CONTENTS
  WHERE SEG_OWNER = 'LOGTEST' AND USERNAME = 'LOGTEST';
```

```
EXECUTE DBMS_LOGMNR.END_LOGMNR();
```