

# Récupération d'une table détruite

par fadace ([Fabien Celaia](#))

Date de publication : 2.1.2007

Dernière mise à jour : 21.10.2007


Oups ! Je (ou un de mes stupides développeurs) viens de supprimer une table importante. Que faire sans passer par une récupération monstrueuse de base de données ? Pas de panique : Oracle flashback est là pour ça!





Débutant   Avancé   Expert



Complexité

 *Pas de récupération possible dans les schémas dépendant du tablespace système*

 *Fonctionnalité uniquement disponible depuis Oracle 10g*

 *Il n'est pas nécessaire que le flashback sont démarré pour que cette fonctionnalité fonctionne en 10g*

Créons une petite table de test avec ses indexes et ses données

```
create table test.t ( i int primary key, v varchar2(20) NULL) ;

create index test.i on test.t(v) ;

insert into test.t values(1,'Test 1', to_blob('ABC')) ;
insert into test.t (i, v) values(2,'Test 2') ;
```

Voici l'incidence que cette création à sur le métamodèle

```
select INDEX_NAME, INDEX_TYPE from dba_indexes where owner = 'CEI' and table_name='T' ;
```

INDEX_NAME	INDEX_TYPE
I	NORMAL
SYS_IL0000045586C00003\$\$	LOB
SYS_C006653	NORMAL

3 rows selected

Suppression de la table

```
drop table test.t ;
```

Incidence dans la poubelle

```
select OBJECT_NAME, ORIGINAL_NAME, TYPE, DROPSCN, CAN_UNDROP, CAN_PURGE from dba_recyclebin
where owner = 'TEST' and operation = 'DROP' ;
```

OBJECT_NAME	ORIGINAL_NAME	TYPE	DROPSCN
BIN\$JiEhYYA9YCTgQ6wQFkxgJA==\$0 I		INDEX	4075329
NO	YES		
SYS_IL0000045581C00003\$\$	SYS_IL0000045581C00003\$\$	LOB INDEX	4075332
NO	NO		
BIN\$JiEhYYA+YCTgQ6wQFkxgJA==\$0 SYS_C006652		INDEX	4075332
NO	YES		
SYS_LOB0000045581C00003\$\$	SYS_LOB0000045581C00003\$\$	LOB	4075335
NO	NO		
BIN\$JiEhYYA/YCTgQ6wQFkxgJA==\$0 T		TABLE	4075335
YES	YES		

5 rows selected

### Incidence dans le métamodèle

```
select INDEX_NAME, INDEX_TYPE from dba_indexes where owner = 'CEI' and table_name='T';
```

INDEX_NAME	INDEX_TYPE
-----	-----

0 rows selected

### Récupération des noms d'index pour usage postérieur :

```
set pagesize 0
set linesize 200

spool renameindex.sql

select 'ALTER INDEX '||OWNER||'.'||ORIGINAL_NAME||' " RENAME TO OBJECT_NAME ;' FROM dba_recyclebin
where TYPE='INDEX' ;

spool off
```

On ne peut donc restaurer que la table (ce qui semble normal) via une commande flashback.

```
FLASHBACK TABLE TEST.T TO BEFORE DROP ;
```

La table et ses indexes sont recréés, et son entrée n'existe plus dans la poubelle:

```
select OBJECT_NAME, ORIGINAL_NAME, TYPE, DROPSCN, CAN_UNDROP, CAN_PURGE from dba_recyclebin
where owner = 'TEST'
```

OBJECT_NAME	ORIGINAL_NAME	TYPE	DROPSCN
CAN_UNDROP	CAN_PURGE		
-----	-----	-----	-----

0 rows selected

Ce faisant, les indexes sont restaurés aussi, mais pas tous sous leur nom d'origine. Certains ont malheureusement gardé leur nom "poubellisé":

```
select INDEX_NAME, INDEX_TYPE from dba_indexes where owner = 'TEST' and table_name='T';
```

INDEX_NAME	INDEX_TYPE
-----	-----
BIN\$JiEhYYBFYCTgQ6wQFkxgJA==\$0	NORMAL
SYS_IL0000045586C00003\$\$	LOB
BIN\$JiEhYYBGYCTgQ6wQFkxgJA==\$0	NORMAL

On peut ensuite rejouer la commande créée précédemment pour renommer les indexes

```
@renameindex.sql
```

Au cas où la table n'as pas été supprimée, mais un ordre update/delete ou insert a été joué par erreur, il est possible de retrouver une image de la table à une heure donnée en "démontant" les données issues de l'undo

tablespace. Il va sans dire que ce quick flashback au niveau des données n'est possible que si la valeur du paramètre `undo_retention_time` n'est pas dépassé.

```
create table MonSchema.AncienneImage
as select * from MonSchema.MaTableMalMenee
AS OF TIMESTAMP TO_TIMESTAMP('09312007 1455', 'MMDDYYYY HH24MI');
```

