

TP ABD

Vues dynamiques adéquates, permettant de vérifier l'état (open) de la base, s'assurez que les fichiers de la base ont été créés et vérifiez la taille de l'instance (Total System Global Area)

```
select * from v$instance
select * from v$datafile
select * from v$sga
```

La liste des utilisateurs de la base

```
select * distinct username from v$session
select * distinct username from v$process
```

Créer un fichier d'initialisation des paramètres serveur :

```
create spfile from pfile='...'
```

Deux techniques d'observation des paramètres d'initialisation.

```
v$instance
v$database
```

Affichez le nom de la base, le nom de l'instance et la taille des blocs de données.

```
select * from v$parameter where name='db_block_size'
```

Affichez les noms et taille des fichiers de données, les fichiers redo-log en ligne et le nom des fichiers de contrôle.

```
select name,bytes from v$datafile
select name,bytes from v$controlfile
```

Modifiez un paramètre (PROCESSES par exemple) de manière définitive (dans le spfile).

```
alter system set processes=45 scope spfile
alter system set processes=45 scope memory (session courante)
alter system set processes=45 scope both (session courante + spfile)
```

PS: en consultant v\$parameter avant et après modification (la colonne ISSES_MODIFIABLE indique que le paramètre peut être modifié par alter session, la colonne ISSYS_MODIFIABLE qu'il peut être modifié par alter system, et la colonne ISMODIFIED que le paramètre a été modifié dans cette session).

Créer un utilisateur :

```
create user bernard identified by morin;
grant create session to bernard;
```

Tuez la session de bernard :

```
select username,sid,serial# from v$session
alter system kill (sid,serial#)
```

Restreignez l'accès à la base :

```
alter system enable restricted session
```

Activez l'extension automatique de vos deux fichiers de données,

```
alter database datafile '/opt/oradata/DBOBH/system01.dbf autoextend on
alter database datafile '/opt/oradata/DBOBH/sysaux01.dbf autoextend on
```

Les segments de rollback :

dba_rollback_segs

Le tablespace SYSTEM :

```
select file_name from dba_data_files where tablespace_name="system"
```

Combien d'espace est déjà utilisé dans la base ?

```
select tablespace_name,bytes from dba_free_space
```

Quelles tables du dictionnaire sont utilisées pour définir la vue dba_users

```
select * from DICT where table_name='DBA_USERS'  
select text from dba_views where view_name='DBA_USERS'
```

Le script à exécuter pour établir les fonctionnalités PL/SQL : catproc.sql
Testez si tous les packages sont valides

```
select count(*) as nb_package_non_valide from user_objects where object_type='PACKAGE' and  
status='VALID'
```

Quel est le nom des fichiers de contrôle ? Où sont-ils localisés ?

```
select * from v$controlfile
```

Ajouter un fichier de contrôle à votre base.

```
alter system set controlFiles=('/opt/oradata/DBOBH/ctrlDBOBH.ctl,...) SCOPE ...
```

Quels sont les fichiers redo-log de votre base ?

```
select member from v$logfile
```

Combien y a-t-il de groupes et de membres ?

```
select groups from v$thread ; select group#,members from v$log
```

Le mode archivage est-il activé ?

```
select log_mode from v$database
```

Ajoutez un membre à chaque groupe et vérifiez leur existence.

```
alter database add logfile member '/opt/oradata/DBOBH/log1c.rdo' to GROUP 1  
alter database add logfile member '/opt/oradata/DBOBH/log2c.rdo' to GROUP 2
```

Créez un nouveau groupe et vérifiez son existence.

```
select groups from v$thread  
alter database add logfile member '/opt/oradata/DBOBH/log3a.rdo' size 10M;  
alter database rename file '/opt/oradata/DBOBH/log3a.rdo' to '/opt/oradata/DBOBH/Temp/log3a.rdo'  
alter database drop logfile group 3;
```

Modifier la taille des fichiers de redo-log (impossible directement):

```
Créer un nouveau redo-log      alter system switch logfile supprime ancien fichier  
vérifier le status dans v$log
```

ABD-Cuenot_tp_21-11-2006

@/opt/oracle/product/10.2.0/rdbms/admin/catproc.sql

Vérification que tous les packets sont valides:

```
select count(*) as nb_package_non_valide from user_objects where object_type='PACKAGE' and status='VALID';
```

NB_PACKAGE_NON_VALIDE

```
-----  
445
```

Après quelques vérifications, malgré les erreurs, ça a l'air de fonctionner ...

Nom des fichiers de contrôle et leur localisation:

```
select * from v$controlfile;
```

/opt/oradata/DBCUG/control01.ctl

Un fichier de contrôle.

Connaitre les fichiers redo-log de ma base:

```
SELECT * FROM v$logfile;
```

```
1 ONLINE  
/opt/oradata/DBCUG/log1a.rdo  
NO
```

```
2 ONLINE  
/opt/oradata/DBCUG/log2a.rdo  
NO
```

2 fichiers redo log (noms ci dessus).

Le nombre de groupes et de membres:

```
SELECT group#, sequence#, bytes, members, status FROM v$log;
```

```
GROUP# SEQUENCE# BYTES MEMBERS STATUS  
-----  
1 201 6291456 1 INACTIVE  
2 202 6291456 1 CURRENT
```

Mode archivage désactivé. Informations sur les archives log:

```
ARCHIVE LOG LIST;
```

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

```
SELECT archiver FROM v$instance;
```

```
Database log mode          No Archive Mode  
Automatic archival        Disabled
```

Ajout d'un membre au groupe 1 (l'opération peut être répétée pour tous les groupes):

```
alter database add logfile member '/opt/oradata/DBCUG/log3a.rdo' to group 1;
```

Database altered.

Vérification:

```
SQL> SELECT * FROM v$logfile;
```

```
GROUP# STATUS TYPE  
-----  
MEMBER
```

```

-----
IS_
---
      1      ONLINE
/opt/oradata/DBCUG/log1a.rdo
NO

      2      ONLINE
/opt/oradata/DBCUG/log2a.rdo
NO

      1 INVALID ONLINE
/opt/oradata/DBCUG/log3a.rdo
NO

```

3 rows selected.

SQL> SELECT group#, sequence#, bytes, members, status FROM v\$log;

```

GROUP# SEQUENCE#  BYTES  MEMBERS STATUS
-----
      1      201  6291456    2 INACTIVE
      2      202  6291456    1 CURRENT

```

2 rows selected.

Ajout d'un nouveau groupe:

SQL> alter database add logfile group 3 ('/opt/oradata/DBCUG/log4a.rdo') size 5M;

Database altered.

Vérification de son existence.

SQL> SELECT group#, sequence#, bytes, members, status FROM v\$log;

```

GROUP# SEQUENCE#  BYTES  MEMBERS STATUS
-----
      1      201  6291456    2 INACTIVE
      2      202  6291456    1 CURRENT
      3         0   5242880    1 UNUSED

```

3 rows selected.

Suppression du membre ajouté au groupe 1:

SQL> alter database drop logfile member '/opt/oradata/DBCUG/log3a.rdo';

Database altered.

Vérification de la suppression:

SQL> SELECT * FROM v\$logfile;

```

GROUP# STATUS TYPE
-----
MEMBER

```

```

-----
IS_
---
      1      ONLINE
/opt/oradata/DBCUG/log1a.rdo
NO

```

```
2 ONLINE
/opt/oradata/DBCUG/log2a.rdo
NO
```

```
3 ONLINE
/opt/oradata/DBCUG/log4a.rdo
NO
```

3 rows selected.

Suppression du groupe crée:

```
SQL> alter database drop logfile group 3;
```

Database altered.

Vérification de la suppression:

```
SQL> SELECT group#, sequence#, bytes, members, status FROM v$log;
```

GROUP#	SEQUENCE#	BYTES	MEMBERS	STATUS
1	201	6291456	1	INACTIVE
2	202	6291456	1	CURRENT

2 rows selected.

ABD-Cuenot_tp_29-11-2006

8. Gestion des tablespaces et des fichiers de données

* Création des tablespaces

```
SQL> CREATE TABLESPACE RBS DATAFILE '/opt/oradata/DBCUG/rbs.dbf' size 10M;
SQL> CREATE TABLESPACE DATA01 DATAFILE '/opt/oradata/DBCUG/data01.dbf' size 2M;
SQL> CREATE TABLESPACE DATA02 DATAFILE '/opt/oradata/DBCUG/data02.dbf' size 1M;
SQL> CREATE TEMPORARY TABLESPACE TEMP TEMPFILE '/opt/oradata/DBCUG/temp.dbf' size 2M
EXTENT MANAGEMENT LOCAL;
SQL> CREATE TABLESPACE INDX01 DATAFILE '/opt/oradata/DBCUG/indx01.dbf' size 1M
AUTOEXTEND ON;
SQL> CREATE TABLESPACE RONLY DATAFILE '/opt/oradata/DBCUG/ronly.dbf' size 1M;
```

* Vérification qu'ils ont bien été créés

```
SQL> select * from dba_data_files;
```

* Affichage des informations du tablespace DATA02 puis extension de sa taille de 500k

```
SQL> select * from dba_data_files where tablespace_name='DATA02';
SQL> ALTER DATABASE DATAFILE '/opt/oradata/DBCUG/data02.dbf' RESIZE 1500K;
```

Database altered.

* Déplacement du tablespace INDX01:

- Mise OFFLINE du tablespace

```
SQL> alter tablespace INDX01 OFFLINE;
```

- Renommage du fichier physiquement

```
mv /opt/oradata/DBCUG/indx01.dbf /opt/oradata/DBCUG/indx1.dbf
```

- Renommage du fichier dans oracle

```
SQL> alter database rename file '/opt/oradata/DBCUG/indx01.dbf' to  
'/opt/oradata/DBCUG/indx1.dbf';
```

- Remise ONLINE du tablespace

```
SQL> alter tablespace INDX01 ONLINE;
```

* Opérations sur RONLY

-

ABD-Cuenot_tp_06-12-2006

8. Gestion des tablespace et des fichiers de données

* Opérations sur RONLY

- Création d'un segment de rollback ONLINE dans SYSTEM

```
SQL> create rollback segment TMP_RBS tablespace SYSTEM;
```

```
SQL> alter rollback segment TMP_RBS online;
```

```
SQL> select segment_name,status from dba_rollback_segs where segment_name='TMP_RBS';
```

- Création de la table t1 sur RONLY

```
SQL> CREATE TABLE t1(ID NUMBER(8) UNIQUE) TABLESPACE RONLY;
```

```
SQL> ALTER TABLESPACE RONLY READ ONLY;
```

```
SQL> select tablespace_name,status from dba_tablespaces where tablespace_name='RONLY';
```

- Tentative de création de t2

```
SQL> CREATE TABLE t2(ID NUMBER(8) UNIQUE) TABLESPACE RONLY;
```

```
CREATE TABLE t2(ID NUMBER(8) UNIQUE) TABLESPACE RONLY
```

*

ERROR at line 1:

ORA-01647: tablespace 'RONLY' is read only, cannot allocate space in it

- Suppression de t1

```
SQL> DROP TABLE t1;
```

* Suppression du tablespace RONLY

```
SQL> DROP TABLESPACE RONLY;
```

Tablespace dropped.

9. Structures de stockage

* Connexion en tant que SYSTEM, augmentation du tablespace et exécution du script

```
SQL> ALTER DATABASE DATAFILE '/opt/oradata/DBCUG/data01.dbf' RESIZE 2300K;
```

```
SQL> select tablespace_name,bytes from dba_free_space;
```

TABLESPACE_NAME	BYTES
SYSTEM	16384
SYSTEM	2768896
SYSTEM	57344
SYSTEM	49152
SYSTEM	188416

SYSTEM	2498560
RBS	10420224
DATA01	2293760
DATA02	1441792
INDX01	983040

10 rows selected.

SQL> @/users/prof/brandel/labs/cr_segs.sql

* Différents types de segments de la base

SQL> select segment_name,segment_type from dba_segments where tablespace_name='DATA01';

* Segments qui peuvent avoir un maximum de 5 extend supplémentaires

SQL> select segment_name,segment_type,extents from dba_segments where tablespace_name='DATA01';

* File Number and Block Number de l'entête de la table EMP

SQL> select FILE_ID, BLOCKS from dba_data_files where TABLESPACE_NAME='DATA01';

* Nombre de fragments, espace libre et plus grand extend libre pour chaque tablespace

SQL> select tablespace_name,count(*) as Nom, sum(bytes) as block,max(bytes) as max from dba_free_space group by tablespace_name;

* Exécution du script cr_frgs.sql

SQL> @/users/prof/brandel/labs/cr_frgs.sql

ABD-Cuenot_tp_13-12-2006

10. Gestion des segments de rollback

* Insertion dans la table EMP d'un enregistrement avec tmp_rbs OFFLINE.

**SQL> insert into EMP (EMPNO,ENAME)VALUES('25','test');
insert into EMP (EMPNO,ENAME)VALUES('25','test')**

ERROR at line 1:

ORA-01552: cannot use system rollback segment for non-system tablespace
'DATA01'

Ca ne marche pas car le rollback segment ne peut pas être utilisé puisqu'il est offline.

* Création de 4 segments de rollback dans le tablespace RBS

**SQL> alter rollback segment tmp_rbs online;
SQL> create rollback segment rollback1 tablespace RBS;
SQL> create rollback segment rollback2 tablespace RBS;
SQL> create rollback segment rollback3 tablespace RBS;
SQL> create rollback segment rollback4 tablespace RBS;**

**SQL> alter rollback segment tmp_rbs offline
SQL> drop rollback segment tmp_rbs;**

* Pour que la question A fonctionne il faut utiliser un rollback segment qui est online

**SQL> alter rollback segment rollback1 online;
SQL> insert into EMP (EMPNO,ENAME)VALUES('25','test');**

* Ajout des nouveaux segments de rollback disponibles au démarrage de l'instance

SQL> alter rollback segment rollback1 online;

Ajout de la ligne suivante dans initDBCUG.ora :
rollback_segments=(rollback1,rollback2,rollback3,rollback4)

* Création du segment de rollback DEMO_RBS

SQL> create rollback segment DEMO_RBS STORAGE (INITIAL 10K NEXT 10K OPTIMAL 128K);

* Nombre d'extends dans DEMO_RBS

SQL> select segment_name,extents from dba_segments where segment_name='DEMO_RBS';
SEGMENT_NAME

```
-----  
EXTENTS  
-----  
DEMO_RBS  
      2
```

Il y a donc 2 extends.

* Nombre d'extends apres execution du script

SQL> select segment_name,extents from dba_segments where segment_name='DEMO_RBS';

SEGMENT_NAME

```
-----  
EXTENTS  
-----  
DEMO_RBS  
      2
```

On ne constate donc aucun changement.