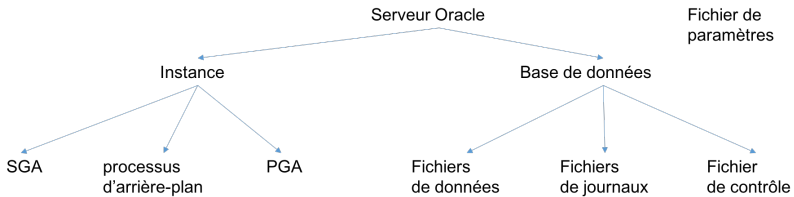


# Administration des bases de données

Mehdi Haddad  
`mehdi.haddad@u-pec.fr`

2016 - 2017

# Cours précédent



# Plan du cours

1. Démarrage et arrêt
2. Création d'une base
3. Gestion des droits des fichiers

# Démarrage et arrêt

- ▶ Pour rendre une base accessible à tous les utilisateurs, il faut démarrer une instance et ouvrir la base de données avec cette instance.
- ▶ Il y a trois grandes phases dans le processus de démarrage :
  - ▶ démarrage de l'instance ;
  - ▶ montage de la base de données ;
  - ▶ ouverture de la base de données.
- ▶ De même, il y a trois grandes phases dans le processus d'arrêt :
  - ▶ fermeture de la base de données ;
  - ▶ démontage de la base de données ;
  - ▶ arrêt de l'instance.

# Démarrage et arrêt

- ▶ Une instance peut être démarrée avec trois niveaux successifs de disponibilité de la base de données, correspondant aux trois phases du démarrage :
  - ▶ Instance démarrée (état NOMOUNT);
  - ▶ Base montée (état MOUNT);
  - ▶ Base ouverte (état OPEN).

## Démarrage et arrêt - Instance démarrée

- ▶ Lors du démarrage de l'instance, le fichier de paramètres est lu, la SGA est allouée et les processus d'arrière-plan sont démarrés.
- ▶ À ce stade, seule l'instance est lancée ; il n'y a pas de base de données associée.
- ▶ Les vues dynamiques relatives à l'instance (V\$INSTANCE, V\$SGA, V\$OPTION, V\$PARAMETER, V\$VERSION, etc.) sont interrogeables mais pas les vues dynamiques relatives à la base de données (V\$DATABASE par exemple). Cet état est principalement utilisé lors de la création d'une nouvelle base.

## Démarrage et arrêt - Base montée

- ▶ Lors du montage de la base de données, l'instance utilise le paramètre `CONTROL_FILES` du fichier de paramètres pour localiser les fichiers de contrôle et les ouvrir.
- ▶ Dans le fichier de contrôle, l'instance extrait le nom et le statut des fichiers de données et des fichiers de journalisation mais ne les ouvre pas et ne vérifie pas non plus leur présence ; si un fichier n'est pas trouvé, aucun message d'erreur n'est affiché.
- ▶ À ce stade, une base de données est associée à l'instance (`V$DATABASE` est maintenant interrogeable) mais n'est pas ouverte pour une utilisation "normale" : personne ne peut se connecter à la base de données, à l'exception d'un utilisateur ayant le privilège `SYSDBA`

## Démarrage et arrêt - Base montée

- ▶ Les vues statiques du dictionnaire ne sont notamment pas accessibles.
- ▶ Dans cet état, le DBA peut effectuer certaines tâches d'administration : renommer ou déplacer un fichier de données ou un fichier de journalisation, activer ou désactiver l'archivage des fichiers de journalisation, effectuer une récupération de la base de données.



## Démarrage et arrêt - Base ouverte

- ▶ Lors de l'ouverture de la base de données, l'instance ouvre les fichiers de journalisation et les fichiers de données qui étaient en ligne au moment de l'arrêt et vérifie la cohérence de la base de données.
- ▶ Si l'un des fichiers de données à ouvrir n'est pas trouvé ou est endommagé, l'instance signale une erreur et la base de données n'est pas ouverte.
- ▶ Si la base de données peut être ouverte mais que le dernier arrêt n'était pas un arrêt propre, SMON effectue la récupération de l'instance.
- ▶ À ce stade, la base de données est accessible pour une utilisation "normale" : les utilisateurs peuvent se connecter. Le dictionnaire de données est totalement disponible.

# Démarrage

```
STARTUP [NOMOUNT | MOUNT [nom_base] | OPEN [nom_base]]  
        [RESTRICT] [PFILE=nom_fichier]
```

- ▶ NOMOUNT | MOUNT | OPEN : niveau de disponibilité souhaité.
- ▶ nom\_base : nom de la base à monter ou à ouvrir.
- ▶ RESTRICT : restreint l'accès à la base aux utilisateurs ayant le privilège RESTRICTED SESSION.
- ▶ PFILE : nom du fichier de paramètres à utiliser.

## Arrêt

SHUTDOWN [NORMAL | IMMEDIATE | TRANSACTIONAL | ABORT]

- ▶ **NORMAL** : Oracle attend que tous les utilisateurs soient déconnectés (pas de nouvelle connexion autorisée) puis ferme proprement la base de données.
- ▶ **IMMEDIATE** : Oracle déconnecte tous les utilisateurs (en effectuant un ROLLBACK des éventuelles transactions en cours) puis ferme proprement la base de données.
- ▶ **TRANSACTIONAL** : Oracle attend que toutes les transactions en cours se terminent avant de déconnecter les utilisateurs (pas de nouvelle transaction autorisée) puis ferme proprement la base de données.
- ▶ **ABORT** : Oracle déconnecte tous les utilisateurs (sans effectuer de ROLLBACK des éventuelles transactions en cours) puis ferme "brutalement" la base de données, sans effectuer de point de synchronisation (checkpoint). Une récupération de l'instance sera nécessaire lors du prochain démarrage.

# Création d'une base de données

1. Créer les répertoires sur les disques.
2. Préparer un nouveau fichier de paramètres texte.
3. Créer le fichier de mots de passe.
4. Positionner la variable d'environnement `ORACLE_SID`
5. Créer le fichier de paramètres serveur.
6. Démarrer l'instance.
7. Créer la base de données.
8. Création du dictionnaire de données.

## Créer les répertoires sur les disques

- ▶ Un répertoire d'administration portant le nom de la base de données, situé dans le répertoire `$ORACLE_BASE/admin`.
- ▶ Un répertoire de données, portant le nom de la base de données, situé dans un répertoire `oradata` lui-même situé dans `ORACLE_BASE`.

## Fichier de paramètres texte

- ▶ Il y a plus de 250 paramètres documentés par Oracle.
- ▶ Sur la totalité des paramètres, une trentaine de paramètres sont suffisants pour la plupart des bases de données.

```
db_name = adminBD
compatible = 11.2.0
control_files = /u01/app/oracle/oradata/adminBD/control01.ctl
db_block_size = 8192
sga_target=503316480
shared_pool_size=176160768
streams_pool_size=0
```

# Fichier de mots de passe et variable d'environnement

- Il est nécessaire de créer le fichier de mots de passe pour l'authentification SYSDBA en utilisant l'outil orapwd.

```
ORAPWD FILE=fichier [PASSWORD=mot_de_passe]  
[ENTRIES=nombre] [FORCE=y|n] [IGNORECASE=y|n]  
[SYSBACKUP=y|n]
```

- Positionner la variable d'environnement ORACLE\_SID au niveau du système d'exploitation :

```
export ORACLE_SID=adminBD
```

- Se connecter a sqlplus  
sqlplus sys as sysdba

# Fichier de paramètres serveur et démarrage de l'instance

- ▶ La création d'un fichier de paramètres serveur s'effectue à partir d'un fichier de paramètres texte

```
SQL> CREATE SPFILE FROM PFILE = 'chemin_vers_pfile';
```

- ▶ L'instance peut être démarrée, en NOMOUNT puisque la base de données n'existe pas encore :

```
SQL> STARTUP NOMOUNT
```



# Créer la base de données

```
CREATE DATABASE [nom_base]
[ USER SYS IDENTIFIED BY mot_de_passe ]
[ USER SYSTEM IDENTIFIED BY mot_de_passe ]
[ CONTROLFILE REUSE ]
[ SET DEFAULT { BIGFILE | SMALLFILE } TABLESPACE ]
[ DATAFILE spécification_fichier [,...] ]
    [ EXTENT MANAGEMENT LOCAL ]
[ SYSAUX DATAFILE spécification_fichier [,...] ]
[[ BIGFILE | SMALLFILE ] UNDO TABLESPACE nom
    [ DATAFILE spécification_fichier [,...] ] ]
[[ BIGFILE | SMALLFILE ] DEFAULT TEMPORARY TABLESPACE nom
    [ TEMPFILE spécification_fichier [,...] ]
    [ EXTENT MANAGEMENT LOCAL ] [ UNIFORM [ SIZE valeur [K|M|G|T] ] ] ]
[ DEFAULT TABLESPACE nom
    [ DATAFILE spécification_fichier [,...] ]
    [ clause_extent_management ] ]
[ LOGFILE [GROUP numéro] spécification_fichier_redo [,...] ]
[ ARCHIVELOG | NOARCHIVELOG ]
[ FORCE LOGGING ]
[ CHARACTER SET jeu ]
[ NATIONAL CHARACTER SET jeu ]
[ SET TIME_ZONE = { '+'|- hh:mi' | 'region' } ]
[ MAXINSTANCES nombre ] [ MAXLOGFILES nombre ]
[ MAXLOGMEMBERS nombre ] [ MAXDATAFILES nombre ] ;
```

## Création du dictionnaire de données

- ▶ Après l'exécution de l'ordre SQL `CREATE DATABASE`, la base de données est parfaitement opérationnelle, mais les vues et les synonymes qui rendent le dictionnaire de données exploitable ne sont pas créés.
- ▶ Pour créer ces vues et ces synonymes il faut exécuter (sous SYS) des scripts fournis par Oracle (dans `$ORACLE_HOME/rdbms/admin`) :
  - `catalog.sql` : vues et synonymes de base ;
  - `catproc.sql` : compléments pour les options procédurales (création d'un grand nombre de packages fournis par Oracle).
- ▶ Les scripts `catalog.sql` et `catproc.sql` appellent d'autres scripts.

## Gestion des droits sur des fichiers

Les droits des fichiers d'un répertoire peuvent être affichés par la commande :

```
ls -l
```

- ▶ Le premier symbole « - », « d », soit « l », nous indiquant la nature du fichier :  
- : fichier d : répertoire l : lien
- ▶ les neuf caractères suivant représentent 3 groupes de 3 symboles chacun, indiquant si le fichier (ou répertoire) est autorisé en lecture, écriture ou exécution.
- ▶ Les 3 groupes correspondent, dans cet ordre, aux droits du propriétaire, du groupe puis du reste des utilisateurs.
- ▶ Ce sont ces lettres qui sont utilisées pour symboliser lesdites permissions. Si la permission n'est pas accordée, la lettre en question est remplacé par « - ».

## Gestion des droits : exemple

`drwxr-xr-x`

- ▶ `d` : c'est un répertoire.
- ▶ `rwx` pour le premier groupe de 3 symboles : son propriétaire peut lire, écrire et exécuter.
- ▶ `r-x` pour le deuxième groupe de 3 symboles : le groupe peut uniquement lire et exécuter le fichier, sans pouvoir le modifier.
- ▶ `r-x` pour le troisième groupe de 3 symboles : le reste du monde peut uniquement lire et exécuter le fichier, sans pouvoir le modifier.

## Changement des droits : chown

La commande chown (change owner) permet de changer le propriétaire du fichier. Seuls le super-utilisateur ou le propriétaire actuel d'un fichier peut utiliser chown.

```
sudo chown utilisateur nom_fichier
```

chown permet aussi de changer en une seule commande le propriétaire et le groupe du fichier :

```
sudo chown utilisateur:groupe nom_fichier
```

## Changement des droits : chmod

La commande chmod (change mode) permet de modifier les permissions sur un fichier.

```
chmod 750 nom_fichier
```

<b>rw</b> x	<b>r</b> - <b>x</b>	<b>---</b>
7(4+2+1)	5(4+0+1)	0(0+0+0)

chmod permet également de modifier récursivement les permission d'un répertoire :

```
chmod -R 750 nom_répertoire
```