

## Mise à jour du fichier de contrôle

## Objectifs

- ⌘ Explication sur les utilisations du fichier de contrôle
- ⌘ Examen du contenu du fichier de contrôle
- ⌘ Récupération de l'information sur le fichier de contrôle
- ⌘ Multiplexage du fichier de contrôle

## Utilisation du fichier de contrôle

## Contenu du fichier de contrôle

- ⌘ Fichier binaire
- ⌘ Sert pour la base de données
  - ☑ lors du démarrage normal
    - ☑ A chaque fois qu'une instance monte une base de données
    - ☑ Renseigne l'emplacement des fichiers de données et des fichiers de reprise
  - ☑ lors du démarrage après panne
    - ☑ Contient les informations nécessaires à la remise en état de la base de données
  - ☑ pour le bon fonctionnement
    - ☑ continuellement mis à jour
    - ☑ doit être disponible quand une base de données est montée ou ouverte
- ⌘ Si un fichier de contrôle devient indisponible => la base ne fonctionne pas correctement

- ⌘ Le nom et l'identification de la base de données
- ⌘ le nom et l'emplacement des fichiers
  - ☑ de données
  - ☑ de reprise
- ⌘ le nom des tablespaces
- ⌘ La date et l'heure de la création de la base de données
- ⌘ le numéro de séquence du fichier de reprise courant
- ⌘ informations sur le dernier point de synchronisation
- ⌘ historique du journal (v8)

## Détails du fichier de contrôle

### ⌘ Paramètres influençant sa taille :

- ☒ MAXLOGFILES
- ☒ MAXLOGMEMBERS
- ☒ MAXLOGHISTORY
- ☒ MAXDATAFILES
- ☒ MAXINSTANCES

### ⌘ Requêtes sur le fichier de contrôle

- ☒ V\$CONTROLFILE
  - ☒ NAME
- ☒ V\$PARAMETER
  - ☒ NAME
  - ☒ VALUE
- ☒ V\$CONTROLFILE\_RECORD\_SECTION (V8)
  - ☒ TYPE
  - ☒ RECORDS\_SIZE
  - ☒ RECORDS\_TOTAL
  - ☒ RECORDS\_USED

Mise à jour  
des fichiers de reprise  
(ou redo log files  
ou journaux)

## Multiplexage du fichier de contrôle

- ⌘ Vu son importance pour une base Oracle, il faut prévenir des défaillances disques ou suppressions malheureuses
- ⌘ Oracle gère alors plusieurs copies du même fichier
  - ☒ par exemple, sur des disques différents
- ⌘ paramètre CONTROL\_FILES dans le fichier d'initialisation
  - ☒ dans initSID.ora

```
control_files = (d:\data\ctl1SID.ora,
d:\data\ctl2SID.ora)
```
- ⌘ Permet d'éviter un *unique point d'échec*

## Objectifs

- ⌘ Explication sur l'utilisation des fichiers de reprise (*journaux*)
- ⌘ Récupération de l'information sur le fichier de reprise et sur l'archivage
- ⌘ Contrôle des basculements et des points de synchronisation des fichiers de reprise
- ⌘ Multiplexage et maintenance de fichiers de reprise

## Utilisation des fichiers de reprise (rappel)

- ✖ Idée de base
  - ☒ enregistrer *toutes* les modifications apportées aux données pour minimiser les problèmes liés aux pannes
- ✖ Le serveur Oracle met à jour les fichiers de reprise
  - ☒ buffer de reprise
  - ☒ processus LGWR
- ✖ Ne sont utilisés qu'en cas d'échec d'une instance pour restaurer des *données validées non écrites dans les fichiers de données*

## Initialisation des fichiers de reprise

- ✖ Générés lors de la création de la base de données
- ✖ Paramètres
  - ☒ MAXLOGFILES : nombre max de groupe dans une base de données (<255)
  - ☒ MAXLOGMEMBERS : nombre max de membre par groupe
  - ☒ LOG\_FILES (dans init<sid>.ora) : nombre max de fichiers de reprise à ouvrir :
    - ☒ doit être inférieur à  $\text{MAXLOGFILES} * \text{MAXLOGMEMBERS}$

## Groupes et membres de fichiers de reprise

- ✖ **Groupes** de fichiers de reprise :
  - ☒ ensemble de copies identiques de fichiers de reprise *online*
- ✖ Un **membre** est un élément d'un groupe, i.e. un fichier de reprise
- ✖ LGWR écrit *simultanément* sur chaque membre du groupe
  - ☒ tous les membres d'un groupe possèdent un *numéro de séquence log*
    - ☒ sert d'identifiant
    - ☒ le numéro courant est stocké dans le fichier de contrôle
- ✖ En général, les membres d'un groupe sont placés sur des disques différents
- ✖ Oracle requiert au moins 2 groupes pour le fonctionnement d'une base de données

## Rappel

- ✖ Serveur Oracle enregistre *séquentiellement* toutes les modifications dans le **buffer de reprise circulaire** de la SGA
- ✖ Transfert du buffer vers le *groupe courant* de reprise par **LGWR**
  - ☒ quand un commit est exécuté
  - ☒ quand le buffer est rempli au 1/3
  - ☒ lors d'un délai d'attente du LGWR (3s)
  - ☒ avant que DBWR n'écrive les blocs modifiés du buffer cache de données dans les fichiers de données
- ✖ LGWR écrit *séquentiellement* dans les fichiers de reprise
  - ☒ *organisation circulaire des groupes*
  - ☒ *quand un groupe est plein, on passe au suivant*

## Point de synchronisation

### ✂ Un point de synchronisation se produit:

- ☒ quand LGWR change de groupe courant
- ☒ quand une instance s'arrête (N,T ou I)
- ☒ avec les paramètres d'initialisation
  - ☒ LOG\_CHECKPOINT\_INTERVAL
  - ☒ LOG\_CHECKPOINT\_TIMEOUT
- ☒ commande de l'administrateur
  - ☒ ALTER SYSTEM SWITCH LOGFILE
  - ☒ ALTER SYSTEM CHECKPOINT GLOBAL
    - plus rapide, n'implique pas de switch logfile
- ☒ Sauvegarde d'un tablespace activé ou quand un tablespace est désactivé

### ✂ Actions faites :

- ☒ tous les blocs modifiés du cache de données *qui sont aussi traités par le cache de reprise* sont écrits par DBWR
- ☒ le processus en arrière plan CKPT met à jour les fichiers de contrôle pour signaler son aboutissement

## Archivage

### ✂ Dépend de la politique locale vis à vis des données, i.e.

- ☒ Une perte des données peut-elle être envisagée ?
  - Oui ? Non ?
- ☒ Si oui, quelle est la durée maximale tolérable ?
  - 1 jour ? 1 semaine ?

### ✂ Décision importante de l'administrateur

- ☒ mode ARCHIVELOG
  - fichiers de reprise sont archivés
    - permet de garder **toutes** les modifications de la base
- ☒ mode NOARCHIVELOG
  - fichiers de reprise sont **écrasés** (quand une boucle est faite)
    - On accepte de perdre en cas de panne des modifications

- ☒ Distinction entre les fichiers de reprise *online et offline*

*archive log les b*

## Archivage des fichiers de reprise

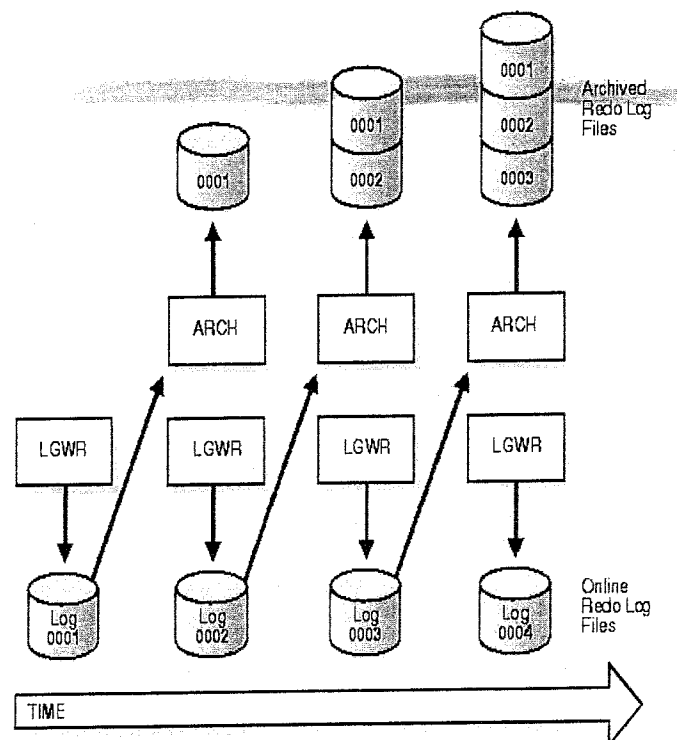
### ✂ Assure une cohérence parfaite des données en cas de panne

### ✂ Processus en arrière plan ARCH

- ☒ déclenché à chaque basculement

### ✂ Archivage

- ☒ soit manuel
  - ☒ c'est à l'administrateur de déclencher ou pas un archivage des fichiers de reprise
    - ALTER SYSTEM ARCHIVE LOG
    - pas conseillé par Oracle
- ☒ soit automatique
  - ☒ paramètre de l'instance LOG\_ARCHIVE\_START=true
  - ☒ dans init.ora, voir aussi paramètres :
    - LOG\_ARCHIVE\_DEST
    - LOG\_ARCHIVE\_FORMAT
- ☒ possibilité de les mutiplexer



## Requêtes sur les fichiers de reprise

### Sur l'archivage :

#### Server Manager :

☒ ARCHIVE LOG LIST

#### Interroger les vues

☒ V\$DATABASE

name, log\_mode

☒ V\$INSTANCE

archiver

☒ V\$LOG\_HISTORY, V\$ARCHIVED\_LOG,  
V\$ARCHIVE\_DEST

### sur les groupes

☒ V\$THREAD

groups, current\_group#, sequence#

### Sur les membres

☒ V\$LOG

group#, members, status, sequence#, bytes

## Modification de fichiers de reprise

### Ajout de groupes

☒ ALTER DATABASE ADD LOGFILE

('D:\data\DISK3\log3a.rdo',

'D:\data\DISK4\log3b.rdo') size 150 K;

taille  
du fichier  
des

### Ajout de membres

☒ ALTER DATABASE ADD LOGFILE MEMBER

'D:\data\DISK3\log1b.rdo' TO GROUP 1,

'D:\data\DISK3\log2b.rdo' TO GROUP 2;

### Déplacement des fichiers de reprise

1. Arrêter la base de données

2. Copier les fichiers de reprise au nouvel emplacement

3. Monter la base de données

4. Exécuter un ordre

ALTER DATABASE [db] RENAME FILE  
['file' [, 'file'] ... TO ['file' [, 'file'] ...

5. Ouvrir la base de données

## Suppression

### Suppression de *groupes* de fichiers de reprise

☒ ALTER DATABASE DROP LOGFILE

GROUP 3

### Suppression de *membres* fichiers de reprise

☒ ALTER DATABASE DROP LOGFILE

MEMBER 'D:\data\DISK3\log2b.rdo'

☒ Le groupe doit être *inactif*

☒ un *autre membre* du groupe doit exister

☒ Si mode ARCHIVELOG et que le groupe du membre  
n'a pas été archivé, opération non permise.

### Suppression des contenus des fichiers

☒ ALTER DATABASE CLEAR LOGFILE

'D:\data\DISK3\log2a.rdo'

## Gestion des fichiers de données et des Tablespaces

## Objectifs

- ⌘ Description de la structure logique de la base de données
- ⌘ Création des tablespaces
- ⌘ Changement de la taille des tablespaces par différentes méthodes
- ⌘ Changement du statut et stockage de l'initialisation des tablespaces
- ⌘ Déplacement des tablespaces
- ⌘ Préparation des tablespaces requis

## Généralités

### Base de données Oracle

- ⌘ structure physique
  - ⌘ fichier de contrôle
  - ⌘ fichiers de reprise
  - ⌘ fichiers de données
- ⌘ structure logique
  - ⌘ tablespace
  - ⌘ segment
  - ⌘ extent
  - ⌘ bloc

=> Nécessaire de comprendre ces 2 niveaux de représentation

⌘ Sous OEM, *Storage Manager*

## Utilisation du tablespace

- ⌘ Contrôle de l'allocation d'espace et affectation de quotas aux utilisateurs
- ⌘ Contrôle de disponibilité des données par la mise *online* ou *offline* des tablespaces
- ⌘ *Distribution du stockage des données* sur plusieurs dispositifs
  - ⌘ pour améliorer les E/S
  - ⌘ pour réduire la contention sur un seul disque
- ⌘ Exécution de *sauvegarde partielle*
- ⌘ Conservations de volumes importants de données *statiques* sur des dispositifs en lecture seule.

## Tablespaces

- ⌘ Appartient à une seule base de données
- ⌘ Chaque tablespace comprend un ou plusieurs fichiers SE
- ⌘ Les tablespaces peuvent être mis on/off line lors de l'exécution de la base de données
  - ⌘ excepté le tablespace SYSTEM
  - ⌘ et un tablespace avec un rollback segment actif
- ⌘ Possibilité de basculer entre le statut Read/Write et le statut Read seul.

## Fichiers de données

- ☞ Un ou plusieurs fichiers de données sont associés à un tablespace
  - ☑ un fichier de données ne peut appartenir qu'à un tablespace
  - ☑ un serveur Oracle crée toujours *un peu plus* d'espace disque que ce qui est spécifié

### ☞ Informations quantitatives :

- ☑ nombre maximum de fichier par tablespace = 1023
- ☑ nombre maximum de tablespace par base de données : 64 000

## Lien segments/extents/blocs

### ☞ Segments

- ☑ Espace alloué à un type spécifique de structure logique de stockage dans un tablespace
  - ☐ segments de table, segments d'index, segment temporaire, rollback segment

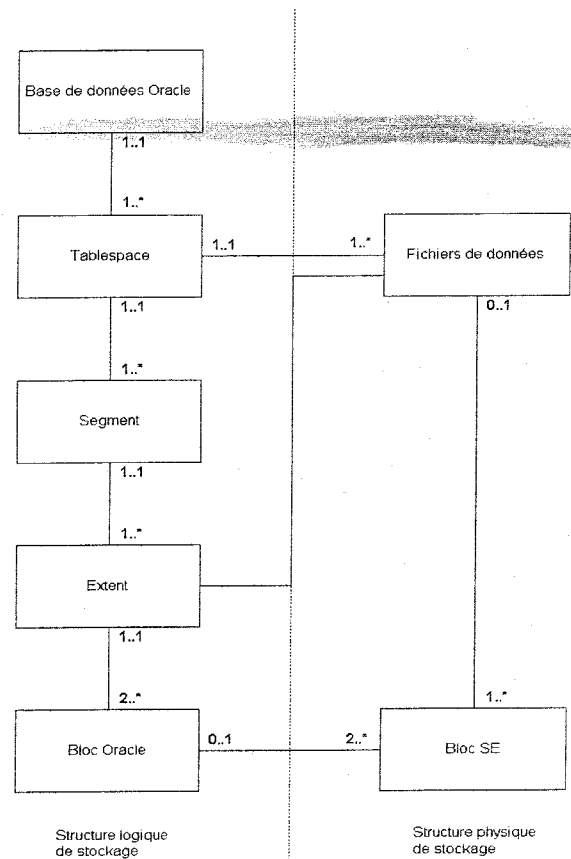
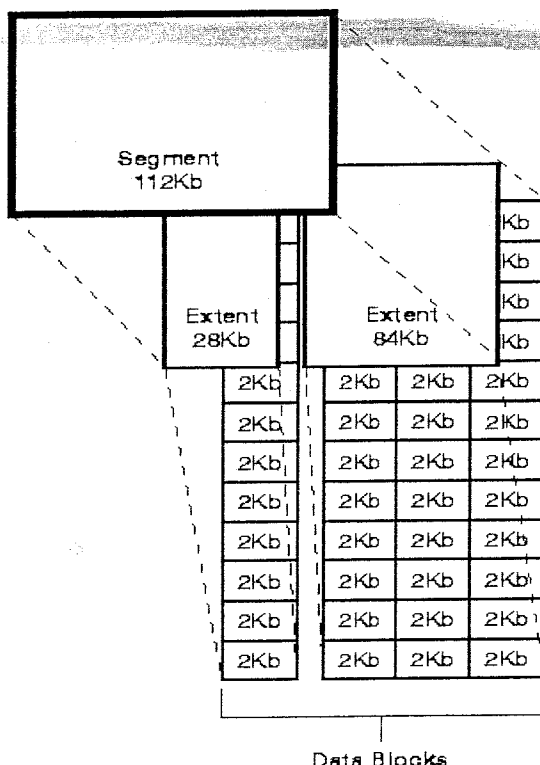
### ☞ Extents

- ☑ Ensemble de blocs contigus.
- ☑ Chaque type de segment est composé d'un ou plusieurs extents

### ☞ Blocs de données

- ☑ contient un ou plusieurs blocs de fichier physique alloué à partir d'un fichier de données existant
- ☑ plus petite unité d'E/S
- ☑ sa taille vaut DB\_BLOCK\_SIZE

## Structure logique



# Tablespace SYSTEM et non-SYSTEM

## ⌘ Tablespace SYSTEM contient:

- ☑ informations relatives au dictionnaire de données
- ☑ Un rollback segment SYSTEM

## ⌘ Tablespace non-SYSTEM contient:

- ☑ les rollback segments
- ☑ les segments temporaires
- ☑ les données applicatives
- ☑ les indexes applicatifs

## ⌘ S'il en existe un seul, c'est lui !

## ⌘ Introduisent plus de souplesse dans l'administration

## ⌘ Ne devrait pas contenir de données utilisateurs

# Syntaxe

```
⌘ CREATE TABLESPACE nom
   DATAFILE file SIZE integer [K|M]
   [autoextend_clause]
   [, file [autoextend_clause]
   [MINIMUM EXTENT integer
   [K|M]] [DEFAULT
   storage_clause]
   [PERMANENT | TEMPORARY]
   [ONLINE | OFFLINE]
```

```
⌘ storage_clause ::= STORAGE (
   [INITIAL integer [K|M]]
   [NEXT integer [K|M]]
   [MINEXTENTS integer]
   [MAXEXTENTS [integer | UNLIMITED]
   [PCTINCREASE integer] )
```

```
⌘ autoextend_clause ::= AUTOEXTEND
   [OFF | ON
   [NEXT integer [K|M] ]
   [MAXSIZE
   [UNLIMITED | integer [K|M]]]]
```

# Paramètres de stockage

## ⌘ Paramètres qui influent sur l'allocation d'espace de segment

- ☑ INITIAL : taille du 1er extent (min à 2\*DB\_BLOCK\_SIZE, par défaut à 5 blocs)
- ☑ NEXT : taille du suivant (min à 1 bloc, par défaut à 5 blocs)
- ☑ MINEXTENTS : nombre d'extents alloués lors de la création du segment (min à 1).
- ☑ MAXEXTENTS : nombre maximum d'extent d'un segment (min à 1, par défaut dépend de la taille du bloc, i.e. DB\_BLOCK\_SIZE, max =UNLIMITED)
- ☑ PCTINCREASE : % de croissance de la taille de l'extent calculée par :  
$$\text{Taille du nième extent} = \text{NEXT} * (1 + \text{PCTINCREASE}/100)^{(n-2)}$$
  - ☑ minimum à 0
  - ☑ valeur par défaut à 50
  - ☑ valeur calculée arrondie (multiple de 5 blocs).

# Tablespaces temporaires

## ⌘ Utilisé pour les opérations de tri

- ☑ par les segments de tri (= segments temporaires)

## ⌘ Ne peut pas contenir d'objets permanents

## ⌘ Exemple :

```
☑ CREATE temporary tablespace temp1
   tempfile c:\oracle\ora92\temp\temp1.dbf SIZE 50M
```



## Augmenter la capacité de stockage

### \* Deux façons de procéder :

- ☑ Ajouter des fichiers de données au tablespace

```
ALTER TABLESPACE app_data  
ADD DATAFILE  
'd:\data\DISK5\app03.dbf' SIZE 200M;
```

- ☑ Modifier la taille des fichiers de données existants

```
Automatiquement avec la clause AUTOEXTEND  
ALTER TABLESPACE app_data  
ADD DATAFILE  
'd:\data\DISK6\app04.dbf' SIZE 200M  
AUTOEXTEND ON  
NEXT 10M MAXSIZE 500M;
```

- ☑ Manuellement avec la commande

```
ALTER DATABASE  
ALTER DATABASE DATAFILE  
'd:\data\DISK5\app02.dbf' RESIZE 2M;
```

## Statut OFFLINE

- \* Les données d'un tablespace offline ne sont plus accessibles

- ☑ L'administrateur peut sauvegarder cette partie des données sans arrêter le fonctionnement du reste de la base

- \* Ne peuvent jamais être mis offline

- ☑ le tablespace SYSTEM

- ☑ tout tablespace avec des rollbacks segments actifs

- \* Exemple :

```
ALTER TABLESPACE app_data OFFLINE;
```

- ☑ Il existe 3 modes :

- ☑ NORMAL (par défaut)

- ☑ TEMPORARY (en cas de pbmes)

- ☑ IMMEDIATE (idem)

## Déplacement de fichiers de données

### 1) BD ouverte

- ☑ Le tablespace concerné doit être offline

- ☑ Les fichiers de données cibles doivent exister (commandes SE)

- ☑ Avec ALTER TABLESPACE :

```
ALTER TABLESPACE app_data  
RENAME DATAFILE  
'd:\data\ DISK4\app01.dbf' TO  
'd:\data\DISK5\app01.dbf'
```

### 2) BD fermée

- ☑ Arrêter puis monter la base de données

- ☑ Utiliser l'ordre ALTER DATABASE

```
ALTER DATABASE  
RENAME FILE  
'd:\data\DISK1\system01.dbf' TO  
'd:\data\ DISK2\system01.dbf' ;
```

## Lecture seule

- \* Empêche toute opération d'écriture sur les fichiers de données

- \* Permet de faciliter l'administration d'un sous-ensemble stable des données de l'application

- ☑ données peuvent résider sur CDROM ou WORM

- ☑ Exemple :

```
ALTER TABLESPACE app_data  
READ ONLY
```

- \* Doit être mis online dans la base de données qui l'a créée

=> pas adapté pour

- ☑ des besoins d'archivage

- ☑ de publications de données

## Lecture seule (2)

- ⌘ Le tablespace doit être online
- ⌘ Aucune transaction active n'est permise
- ⌘ Le tablespace ne doit pas contenir de rollback segments actifs
- ⌘ Le tablespace ne doit être en train de participer à une *sauvegarde online*.

## Suppression d'un tablespace

### ⌘ Syntaxe :

```
DROP TABLESPACE tablespace  
[INCLUDING CONTENTS  
[CASCADE CONSTRAINTS]] ;
```

☑ INCLUDING CONTENTS : supprime tous les segments du tablespace

☑ CASCADE CONSTRAINTS : supprime les contraintes d'intégrité référentielle des tables définies dans un autre tablespace qui référencent une table du tablespace à supprimer

⌘ Un tablespace contenant encore des données ne peut être supprimé sans l'option INCLUDING CONTENTS

⌘ Ne supprime que les pointeurs de fichiers du fichier de contrôle  
=> les fichiers de données existent toujours

*SELECT file\_name from dba-segment seg, dba.data-file*

## Informations sur les tablespaces

### ⌘ Requêtes sur la vue DBA\_TABLESPACES

- TABLESPACE\_NAME
- NEXT\_EXTENT
- MAX\_EXTENTS
- PCT\_INCREASE
- MIN\_EXTLEN
- STATUS
- CONTENTS

### ⌘ Informations sur les fichiers de données

#### Requêtes sur la vue DBA\_DATA\_FILES

- FILE\_NAME
- TABLESPACE\_NAME
- BYTES
- AUTOEXTENSIBLE
- MAXBYTES
- INCREMENT\_BY

## Informations (2)

⌘ Informations sur les fichiers de données et les tablespaces *à partir du fichier de contrôle*  
Jointure entre V\$DATAFILE et V\$TABLESPACE sur l'attribut TS#

### ⌘ A retenir :

- ☑ utiliser de nombreux tablespaces
- ☑ assigner des quotas aux utilisateurs

### ⌘ Quelques conseils

- ☑ Séparez les données utilisateurs des données du dictionnaire
- ☑ Séparez les données utilisateurs suivant les applications qui les utilisent
- ☑ Afin de diminuer la contention E/S, placer les fichiers des tablespaces différents sur des disques distincts
- ☑ Séparer les segments rollback des segments de données