

Administration des bases de données

Khaled Jouini
j.khaled@gmail.com

Institut Supérieur d'Informatique et des Technologies de Communication

La reproduction de ce document par tout moyen que ce soit, sans l'avis de l'auteur, est interdite conformément aux lois protégeant la propriété intellectuelle

Ce document peut comporter des erreurs, à utiliser donc en connaissance de cause!

Plan du cours

- 1 Introduction générale
- 2 Concepts de base sur le stockage de données
- 3 Les bases de l'architecture Oracle
- 4 Gestion de l'instance
- 5 Gestion des utilisateurs
- 6 Création manuelle d'une BD et configuration du réseau
- 7 Sauvegarde et récupération



Syllabus du cours

Pré-requis

- Modèle relationnel (structure, contraintes, SQL)
- Architecture d'un SGBDR, connaissances utiles, mais pas indispensables

Objectifs

- Connaître les tâches d'un Administrateur de bases de données
- Connaître les concepts et les points clés de l'architecture d'Oracle
- Savoir effectuer les principales tâches d'administration sous Oracle

Syllabus du cours

SQL à la maison

- Oracle Database 11g Personal Edition ou (à défaut) Oracle Database 11g Express Edition (également les versions 12, 18, 19 et 21c).
<https://www.oracle.com/fr/database/technologies/xe-downloads.html>

Pour en savoir plus...

- Documentation Oracle :
<https://docs.oracle.com/en/database/oracle/oracle-database/index.html>.
- *Oracle 19c Administration*, O. Heurtel, eni Editions, 2021

Chapitre 1 - Introduction générale

- Sécurité, confidentialité et disponibilité des données
- Rôles du DBA

Sécurisation des données : Objectifs

Sécurité des (bases de) données :

- Confidentialité des données : contrôle d'accès
- Intégrité : modifications illicites et perte
- Disponibilité : dénis de service

Maintien de la confidentialité

Confidentialité

- Détecter et empêcher les intrusions et les accès non autorisés : seuls les utilisateurs autorisés ont accès aux informations qui leur sont destinées
- Au niveau des BD : gestion des authentifications, des privilèges et des profils; chiffrement (cryptage) des données, etc.
- Englobe également la traçabilité et l'audit : quel utilisateur a exécuté quelle action sur quelles données et avec quelles privilèges
- Différents modèles, mais essentiellement le modèle RBAC (*Role Based Access Control*)

Maintien de l'intégrité

Intégrité

- Détecter et empêcher l'altération volontaire ou fortuite des données
- Altérations, essentiellement dues à des
 - pannes systèmes
 - manipulations erronées
 - sabotages
 - etc.
- Techniques vues dans ce cours
 - Systèmes RAID
 - Oracle RMAN : sauvegarde et restauration
 - Data Guard



Maintien de la disponibilité

Disponibilité

- Détecter et empêcher les dénis de service : les données doivent être accessibles aux moments voulus aux utilisateurs autorisés
 - Dénier de service : un utilisateur ne parvient pas à accéder à une donnée pour laquelle il a une autorisation d'accès
- ⇒ Le système doit fonctionner sans faille durant les plages d'utilisation prévues
- Techniques vues dans ce cours
 - Oracle RAC (*Real Application Cluster*)
 - Data Guard

Chapitre 1 - Introduction générale

- Sécurité, confidentialité et disponibilité des données
- Rôles du DBA

Rôles du DBA

- 1 Installation et configuration du SGBD et des applications clientes (rôle technique)
- 2 Planification et création des bases de données
- 3 Gestion de l'espace de stockage
- 4 Gestion des comptes des utilisateurs et des droits d'accès
- 5 Déploiement les procédures de sauvegarde et de récupération pour protéger la base de données
- 6 Optimisation des performances d'accès aux données.
- 7 Gestion des aspects réseau

Principaux outils d'administration

Oracle Universal Installer : interface graphique pour l'installation des outils Oracle

Oracle DataBase Configuration Assistant : assistant pour la création et la configuration d'une base de données à partir d'un modèle par défaut (*OLTP*, *OLAP* ou mixte)

DataBase Upgrade Assistant : assistant pour migration du serveur d'une version à une autre

Net Configuration Assistant : assistant pour la configuration des accès en réseau

Oracle Enterprise Manager : interface graphique permettant d'effectuer les tâches d'administration et de maintenance

SQL*Plus : application cliente du serveur de BD. Il s'agit d'un interpréteur de commandes SQL



Chapitre 2 - Concepts de base sur le stockage de données

■ Introduction

- I. Structures physiques de stockage
- III. Structures logiques de stockage sous Oracle
- IV. Gestion des tablespaces et des fichiers de données
- V. Systèmes RAID
- Quiz

Hiérarchie des mémoires

Mémoire vive

- Espace de travail du processeur : toute donnée ou tout programme doit être chargé en mémoire vive avant d'être traité par le processeur
- Mémoire **volatile** : perd son contenu lorsque la machine est éteinte

Disque dur

- Mémoire **persistante** : ne perd pas son contenu lorsque la machine est éteinte
- Grande capacité de stockage par rapport à la mémoire vive
- Lent : 1 accès au disque dur \approx 1 million d'accès à la mémoire vive

Objectifs d'un SGBD

Base de données : ensemble de fichiers stockés sur disque(s) dur(s) Raisons du stockage sur disque

1 Souvent : taille de la base > taille mémoire vive

2 Persistance

⇒ Lenteur du disque **Problèmes potentiels de performance**

SGBD

- Organiser les données de manière à minimiser le temps d'accès aux données
- Offrir les outils de paramétrage et de contrôle pour exploiter au mieux les ressources matérielles

Ce chapitre...

- 1 Structures physiques de stockage
- 2 Structures logiques de stockage sous Oracle

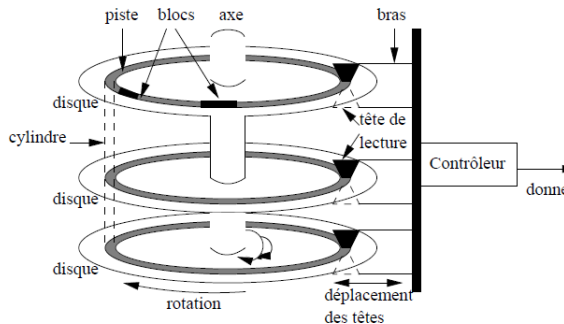
Chapitre 2 - Concepts de base sur le stockage de données

- Introduction
- I. Structures physiques de stockage
 - 1. Fonctionnement d'un disque
 - 2. Optimisation
- III. Structures logiques de stockage sous Oracle
- IV. Gestion des tablespaces et des fichiers de données
- V. Systèmes RAID
- Quiz

1. Fonctionnement d'un disque - a. Dispositif

Disque dur

- Ensemble de **plateaux** tournant à vitesse constante autour d'un axe
- +
- **têtes de L/E** se déplaçant perpendiculairement à l'axe
- Plateau : composé de **pistes** concentriques
- Piste : divisée en **blocs** contenant les données
- **Contrôleur** : reçoit les demandes de L/E et les transforme en mouvements appropriés des têtes de lecture



1. Fonctionnement d'un disque - b. Bloc

Bloc

- **Unité d'échange** entre la mémoire vive et le disque
- Taille typique : 8 Ko. octets

Accès à un bloc

- Par son **adresse** : numéro du plateau + numéro de la piste + numéro du bloc sur la piste
- **Temps de lecture**
 - 1 **Délai de positionnement** : positionnement de la tête de lecture sur la piste contenant le bloc
 - 2 **Temps de latence** : rotation du disque pour attendre que le bloc passe sous la tête de lecture
 - 3 **Temps de transfert** (négligeable) : transfert du bloc vers la mémoire vive

2. Optimisation – a. Proximité

Définition (Proximité)

Mettre dans un nombre minimal de blocs les données qui sont fréquemment utilisées en même temps

- Ex. les clients d'une même ville
- Permet de réduire le **nombre de blocs transférés** en mémoire vive

2. Optimisation - b. Regroupement

Définition (Regroupement)

Mettre dans des blocs consécutifs les données qui ont de grandes chances d'être utilisées en même temps

- Ex. les tuples d'une même relation
- Permet de réduire le **délai de positionnement** de la tête de lecture et le **temps de latence**

Proximité et regroupement \Rightarrow **principe de localité spatiale**

2. Optimisation - c. Tampon mémoire (*Buffer*)

Définition (Tampon mémoire)

Zone en mémoire vive destinée à garder des copies d'une partie des blocs se trouvant sur le disque

- Chaque fois qu'un bloc est lu, une copie en est gardée dans le tampon pour que l'on puisse l'utiliser ultérieurement
- Accès à un bloc : vérifier d'abord s'il ne se trouve pas déjà en mémoire vive avant de le lire depuis le disque
⇒ **principe de localité temporelle**
- Taille du tampon : paramétrable dans la plupart des SGBD

2. Optimisation - c. Tampon mémoire (*Buffer*)

Tampon saturé : plusieurs stratégies

- **LRU** (*Least Recently Used*) : le bloc le plus récemment lu, remplace le moins récemment lu
- **Touch Count** : le bloc le plus récemment lu remplace celui qui totalise le moins d'utilisations
- Exemple
 - Taille tampon : 3 blocs
 - Utilisation des blocs : $t_1 \rightarrow B_a$, $t_2 \rightarrow B_b$, $t_3 \rightarrow B_b$, $t_4 \rightarrow B_c$, $t_5 \rightarrow B_a$ et $t_6 \rightarrow B_d$
 - Quel bloc est remplacé à t_6 ?
 - Selon LRU :
 - Selon Touch Count :

Chapitre 2 - Concepts de base sur le stockage de données

- Introduction
- I. Structures physiques de stockage
- III. Structures logiques de stockage sous Oracle
 - 1. Bloc Oracle
 - 2. Extension
 - 3. Segment
 - 4. Tablespace
- IV. Gestion des tablespaces et des fichiers de données
- V. Systèmes RAID
- Quiz

1. Bloc Oracle - a. Définition

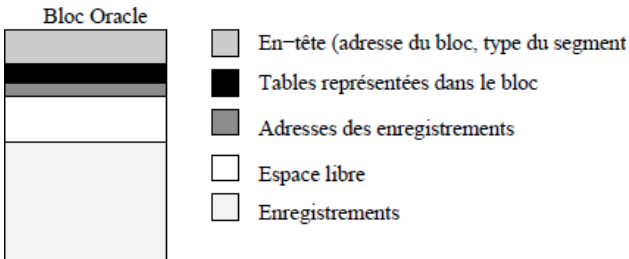
Bloc Oracle (*Block*)

- Plus petite unité de stockage et de transfert de données utilisée par Oracle
- De taille paramétrable : correspond à un multiple de bloc disque
- ⇒ **Intérêt** : Réduire le nombre d'accès au(x) disque(s) en transmettant plus d'informations à la fois à chaque accès
- ⇒ Cependant, une taille de bloc "trop" grande : risque de saturer rapidement le tampon mémoire avec des données inutiles
 - La "bonne" taille dépend de l'application
- Paramètre **DB_BLOCK_SIZE**
 - Pour consulter la taille d'un bloc Oracle à partir de SQL*PLUS :
`SQL> show parameter DB_BLOCK_SIZE`

1. Bloc Oracle - b. Structure

Structure d'un bloc Oracle, essentiellement :

- 1 Entête : contient l'adresse du bloc
- 2 Répertoire des tables : liste des tables des données stockées
- 3 Répertoire des enregistrements : adresse des enregistrements stockés
- 4 Espace libre
- 5 Espace des données : contient les enregistrements



1. Bloc Oracle - c. Contrôle du remplissage

Problème : Mise-à-jour (UPDATE) augmentant la taille d'un tuple dans un bloc saturé

- 1 Conduit au déplacement d'une partie ou de tout un enregistrement dans un autre bloc \Rightarrow **Chaînage**
- 2 Chaînage : opération coûteuse en terme de temps d'exécution, surtout si répétée pour chaque m-à-j + dégradation du temps nécessaire pour lire les tuples chaînés

Solution : réserver un espace dans un bloc pour permettre l'augmentation de la taille des enregistrements due aux m-à-j

1. Bloc Oracle - c. Contrôle du remplissage

Deux paramètres **PCTFREE** et **PCTUSED** spécifiés lors de la création d'une table

■ PCTFREE (PerCenT FREE)

- Pourcentage minimum d'espace libre réservé pour permettre l'accroissement de la taille des enregistrements dû aux mises à jour
- Si taux de remplissage $\geq 100 - \text{PCTFREE} \Rightarrow$ bloc plein : mises-à-jour et suppressions autorisées, insertions interdites

$$\text{PCTFREE} = \frac{\text{Taille moyenne d'un tuple} - \text{Taille initiale d'un tuple}}{\text{Taille moyenne d'un tuple}} \times 100$$

- Par défaut 10%

1. Bloc Oracle - c. Contrôle du remplissage

■ PCTUSED (PerCent USED)

- Pourcentage minimum d'espace devant être occupé par les enregistrements
- Si taux de remplissage \leq PCTUSED \Rightarrow bloc disponible : les insertions peuvent y reprendre
- Par défaut 40%

? Quel intérêt à considérer PCTUSED et non 100-PCTFREE, pour reprendre les insertions

\Rightarrow La gestion des blocs disponibles/indisponibles est moins intensive

1. Bloc Oracle - c. Contrôle du remplissage

Exemple 1

```
CREATE TABLE Produit (pnom VARCHAR(20), prix INTEGER) PCTFREE 30,  
PCTUSED 40;
```

- 1 Les insertions sont autorisées jusqu'à ce que le bloc soit plein à
- 2 Lorsque le bloc est plein à 70 %
 - 1 Les insertions sont interdites dans le bloc ;
 - 2 Suppressions et modifications sont autorisées ;
- 3 Les insertions sont à nouveau autorisées si le taux de remplissage devient inférieur à

⇒ Au pire, quel pourcentage du bloc reste inutilisé?

1. Bloc Oracle - c. Contrôle du remplissage

Exemple 2

Considérons une table où :

- 1 Les insertions (INSERT), les suppressions (DELETE) et les m-à-j (UPDATE) sont fréquentes;
- 2 Les mises à jour n'augmentent pas la taille des tuples significativement;
- 3 La table est volumineuse et économiser l'espace de stockage est primordial

Quelles valeurs choisir pour PCTFREE et PCTUSED?

- ⇒ Pas besoin de réserver beaucoup d'espace pour les m-à-j : petite valeur pour PCTFREE (Ex. 5)
- ⇒ Récupérer rapidement les espaces libérés par les DELETE : grande valeur pour PCTUSED (Ex. 60)

1. Bloc Oracle - c. Contrôle du remplissage

Modification de PCTFREE et de PCTUSED

- `ALTER TABLE table_name PCTFREE int;`
 - `ALTER TABLE PRODUIT PCTFREE 20;`
- `ALTER TABLE table_name PCTUSED int;`
 - `ALTER TABLE PRODUIT PCTUSED 50;`

2. Extension

Extension (*Extent*)

- Ensemble de blocs Oracle **contigus** stockant un même type de données
 - Ex. des tuples d'une même relation
- ⇒ Objectif : améliorer le **regroupement** et donc réduire le temps de nécessaire pour lire des données d'un même type

3. Segment – a. Définition

Segment (*Segment*)

- Ensemble de toutes les extensions stockant les données d'un même type (Ex. les tuples d'une même table)
- Il existe quatre types de segment
 - **Segment de table** : espace occupé par les tuples d'une table
 - **Segment d'index** : espace occupé par un index
 - **Segment d'annulation** : espace occupé par les **images** des données avant que celles-ci ne soient modifiées
 - **Segment temporaire** : espace temporaire où sont stockées les données temporaires utilisées lors d'un tri, d'une jointure, etc.
- Les segments sont groupés dans des **tablespaces**

3. Segment – b. Segment d'annulation (*RollBack Segment*)

Problème : Nécessité d'annuler des actions sur les données, suite à une panne ou autre

Exemple : virement bancaire de 100DT d'un compte A à un compte B

- 2 actions : débiter A de 100 et créditer B de 100
 - A est débité, mais une panne empêche B d'être crédité
- ⇒ la mise-à-jour de A doit être annulée!!
- ⇒ il est nécessaire de garder une **image** de A avant sa mise-à-jour
- ⇒ il est nécessaire de savoir avec quelles autres actions la mise-à-jour de A forme une séquence d'opérations indivisible (= **transaction**)

3. Segment - b. Segment d'annulation (*RollBack Segment*)

Transaction ACID (Atomicité, Cohérence, Isolation, Durabilité)

- Action ou séquence d'actions d'un utilisateur ou d'une application qui doivent être exécutées, soit toutes, soit aucune
- Commence par un `BEGIN TRANSACTION` et se termine par un `COMMIT` (validation) ou un `ROLLBACK` (annulation)
- Enchaînement type
 - 1 Transaction lancée (`BEGIN TRANSACTION`)
 - 2 Les nouvelles images des données modifiées sont stockées dans le segment de table et les anciennes dans le segment d'annulation
 - 3 En cas de `ROLLBACK`, l'ancienne image remplace la nouvelle dans le segment de table
- Vue `V$TRANSACTION` : transactions utilisant le segment d'annulation actif



4. Tablespace

Tablespace

- **Espace logique** permettant d'organiser "logiquement" une base de données, par exemple un tablespace par thème
- Contient un ou plusieurs segments (un segment appartient à un et un seul tablespace)
- Stocké physiquement dans un (au moins) ou plusieurs fichiers, appelés **fichiers de données**
- **Unité de sauvegarde/restauration**

4. Tablespace

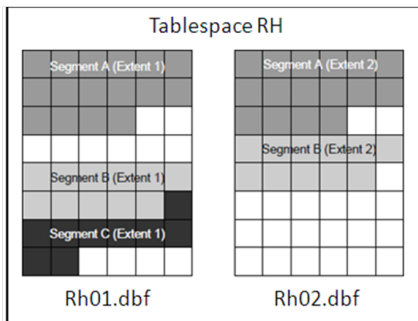
Exemple : BD gérant la facturation et les RH

1 Tablespace Facturation :

- Contient les tables (segments) facture, produit, client, ligne_fact, etc.
- Stocké dans les fichiers fact01.dbf et fact02.dbf

2 Tablespace RH :

- Contient les tables (segments) Employé, Paie, Absence, etc.
- Stocké dans les fichiers rh01.dbf et rh02.dbf



En résumé...

- Un tablespace ne peut appartenir qu'à une seule base de données à la fois.
- Un segment ne peut pas être réparti sur plusieurs tablespaces, mais peut s'étendre à plusieurs fichiers de données d'un même tablespace.
- Un extent ne peut appartenir qu'à un seul fichier.

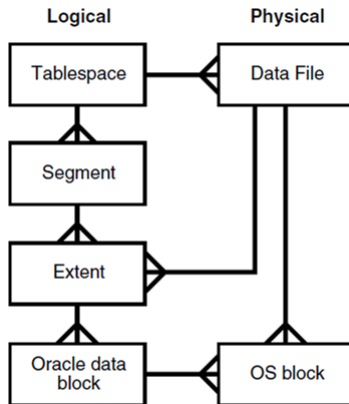


Figure: Structures logiques et physiques de stockage
[Oracle® Concepts, 11g Release 2 (11.2). E10713-05
Fév. 2010]

Chapitre 2 - Concepts de base sur le stockage de données

- Introduction
- I. Structures physiques de stockage
- III. Structures logiques de stockage sous Oracle
- **IV. Gestion des tablespaces et des fichiers de données**
 - 1. Vue d'ensemble et directives
 - 2. Tablespaces temporaires et tablespaces d'annulation
 - 3. Information sur les tablespaces et les fichiers
- V. Systèmes RAID
- Quiz

1. Vue d'ensemble et directives

- Une BD Oracle contient au minimum deux tablespaces créés en même temps que la BD : **SYSTEM** et **SYSAUX** (depuis la version 10)
 - Tablespace SYSAUX : contient les objets créés par les produits Oracle
 - Tablespace SYSTEM :
 - Contient le dictionnaire de données, y compris les programmes stockés
 - Contient le segment d'annulation SYSTEM
 - Ne doit pas contenir les données des utilisateurs (bien que ça soit possible)
- ⇒ **Nécessité de créer d'autres tablespaces**

1. Vue d'ensemble et directives

- En pratique, il est fortement recommandé de créer :
 - Un **tablespace UNDO** (pour les segments d'annulation) :
CREATE UNDO TABLESPACE...
 - Un **tablespace temporaire** (pour les segments temporaires) :
CREATE TEMPORARY TABLESPACE...
 - Au moins un **tablespace permanent** (de données) pour éviter de mettre les données utilisateurs dans SYSTEM :
CREATE TABLESPACE...
- En général, séparer les segments
 - 1 Selon leur type : annulation, temporaire, d'index ou de table
 - 2 Segments de table (de données) : selon les besoins de sauvegarde et de maintenance

1. Vue d'ensemble et directives

L'utilisation de plusieurs tablespaces permet :

- 1 Réaliser des des maintenances et des sauvegardes/restaurations partielles
- 2 Équilibrer la charge en E/S et accès parallèle : généralement, un serveur utilise plusieurs disques et les fichiers d'un même tablespace sont stockés dans des disques différents (**Striping**)
- 3 Gérer l'espace de stockage alloué à chaque utilisateur (Chapitre Gestion des utilisateurs)
- 4 Séparer les données du dictionnaire de données des données applicatives
- 5 Séparer les données de plusieurs applications stockées dans la même BD
- 6 Séparer le stockage des différents types d'objets

2. Tablespaces temporaires

Temporary Tablespace (Tablespace temporaire)

- Composé uniquement par des segments temporaires
- Création

```
CREATE TEMPORARY TABLESPACE temp_01  
    TEMPFILE 'temp.tmp' SIZE 20 M AUTOEXTEND ON;
```

- Restrictions :
 - ne peut pas être en mode lecture seule (READ ONLY)
 - ne peut pas être en mode journalisation (LOGGING)
 - ne peut contenir que des fichiers temporaires (TEMPFILE)

2. Tablespaces temporaires

Temporary Tablespace (Tablespace temporaire)

- Désignation d'un tablespace par défaut (pour tous les utilisateurs)

```
ALTER DATABASE
```

```
    DEFAULT TEMPORARY TABLESPACE nomTbsTemp;
```

- Affectation d'un tablespace temporaire aux activités d'un utilisateur

```
ALTER USER nomUser TEMPORARY TABLESPACE nomTbsTemp;
```

- Vue `DATABASE_PROPERTIES` : permet de trouver le tablespace temporaire par défaut d'une BD

```
SELECT * FROM DATABASE_PROPERTIES;
```

2. Tablespaces d'annulation (Undo tablespace)

Undo Tablespace (Tablespace d'annulation)

- A partir de la version 9i
- Contient tous les segments d'annulation (excepté le segment d'annulation du dictionnaire)
- **Intérêt** : un seul tablespace UNDO au lieu de plusieurs segments Rollback dispersés sur différents tablespaces + gestion automatique
- Il peut en exister plusieurs, mais un instant donné un seul est utilisé par le système

```
ALTER SYSTEM SET UNDO_TABLESPACE = undotbs_02;
```

2. Tablespaces d'annulation (Undo tablespace)

Undo Tablespace (Tablespace d'annulation)

■ Création

```
CREATE UNDO TABLESPACE nomTbsUndo  
    DATAFILE ...  
    [RETENTION [NON GUARANTEE|GUARANTEE]]  
    ...;
```

- Si RETENTION GUARANTEE et que Oracle ne trouve pas d'espace pour garder les données UNDO, toute opération DML est bloquée
- Si RETENTION NON GUARANTEE et qu'une partie des données d'annulation a été écrasée ⇒ la fameuse erreur ORA-01555 snapshot too old

■ Modification de la durée de rétention (par défaut 900 secondes)

```
ALTER SYSTEM SET UNDO_RETENTION = 1000;
```

3. Informations sur les tablespaces et les fichiers de données

Vues du dictionnaire contenant des informations sur les tablespaces et les fichiers de données :

■ Tablespaces

- DBA_TABLESPACES
- V\$TABLESPACE

■ Fichiers de données

- DBA_DATA_FILES
- V\$DATAFILE

■ Fichiers temporaires

- DBA_TEMP_FILES
- V\$TEMPFILE

Chapitre 2 - Concepts de base sur le stockage de données

- Introduction
- I. Structures physiques de stockage
- III. Structures logiques de stockage sous Oracle
- IV. Gestion des tablespaces et des fichiers de données
- **V. Systèmes RAID**
 - 1. Présentation
 - 2. Niveaux
- Quiz

1. Présentation

■ Technologie RAID (*Redundant Array of Independent Disks*)

- Idée : il est plus intéressant d'un point de vue, coût, fiabilité et performances de combiner plusieurs disques "grand marché" que d'avoir un seul disque dur à haute performance et à grande capacité
- Objectif 1 : **limiter les conséquences des pannes matérielles en répartissant les données sur un grand nombre de disques**
- Objectif 2 : **accélérer l'accès aux données**
- Il existe plusieurs niveaux RAID (de 0 à 6), chacun atteignant l'un ou les deux objectifs précédents.

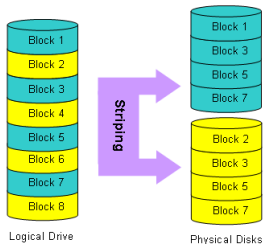
2. Niveaux - a. RAID 0

RAID 0 : volume agrégé par bandes (*Striping*)

■ Principe

- Chaque fichier est divisé en bandes (*Stripes*) de taille fixe (Ex. 64 ko)
- Chaque bande est stockée sur un disque différent
- L'accès (en L/E) aux différentes bandes d'un même fichier peut ainsi se faire en parallèle

■ Améliore les performances, mais **ne protège pas des pannes**



2. Niveaux – b. RAID 1

RAID 1 : Disques en miroir (*Mirroring*)

■ Principe :

- Au moins deux disques : un **disque principal** et des **disques miroirs**
- Les écritures s'effectuent **en parallèle** sur le principal et ses miroirs.

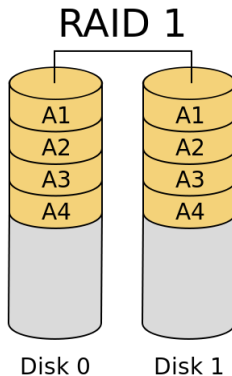
■ Excellente protection des données

- Permet certaines optimisations en lecture : Ex. la demande d'accès à un bloc peut être transmise au disque dont la tête de lecture est la plus proche de la piste contenant le bloc.

- Coûteux (de moins en moins vrai) : la **moitié de l'espace** de stockage est **perdu**

2. Niveaux - b. RAID 1

RAID 1 : Disques en miroir (*Mirroring*)



2. Niveaux - c. RAID 4

RAID 4 : Répartition et parité

■ Principe

- Considérer $n - 1$ disques de la grappe comme étant un seul
- Répartir les données du les $n - 1$ disques
- L'unité de répartition est le bloc : le premier est affecté au premier disque le deuxième au deuxième disque est ainsi de suite
- Pas de redondance, mais utilisation de la **parité** et d'un disque supplémentaire pour reconstruire l'information perdue

■ Parité

- Bit de parité : $=0$ si le nombre de 1 est paire et $=1$ sinon (OU EXclusif)

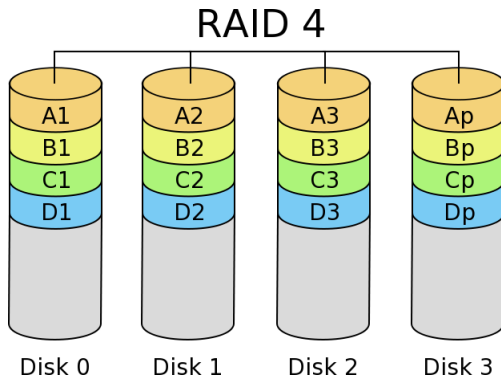
■ Exemple

disque 1	1100
disque 2	0110
disque C	1010

- Si un des disques est endommagé, il est possible de reconstruire l'information à partir des $n - 1$ autres disques et du disque de contrôle

2. Niveaux - c. RAID 4

RAID 4 : Répartition et parité



2. Niveaux - c. RAID 4

RAID 4 : Répartition et parité

■ Avantage

- Un seul disque supplémentaire est nécessaire quel que soit le nombre de disques de données

■ Limites

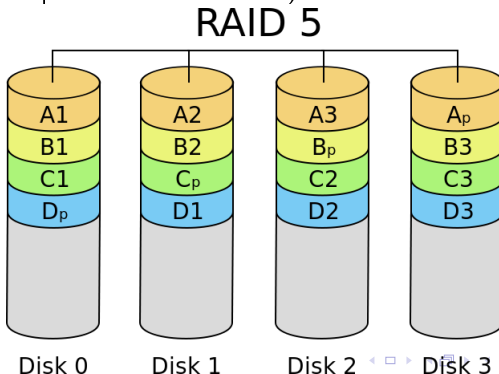
- Si plus d'un disque est endommagé à la fois (improbable), impossible de reconstruire l'information perdue
- Le disque de contrôle est un goulot d'étranglement : il subit $n - 1$ fois plus d'écriture que les autres disques (n nombre total de disques)
- Reconstruction lente pour les disques volumineux

2. Niveaux - d. RAID 5

RAID 5 : volume agrégé par bandes à parité répartie

■ Principe

- Même principe que le RAID 4
- Pour éviter d'avoir un goulot d'étranglement, les blocs de contrôle ne sont pas stockés dans un seul disque, mais répartis sur les différents disques (de la même manière que les blocs de données)

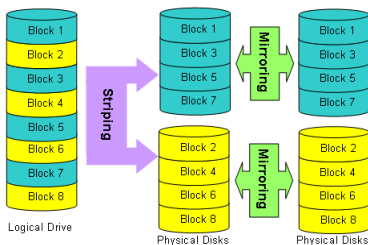


2. Niveaux - e. RAID 1+0

RAID 1+0

■ Principe

- Combine le RAID 1 et le RAID 0 utiliser le *mirroring*, puis le *striping*
- ⇒ nécessite au moins 4 disques
- Combine les avantages *mirroring* et du *striping* ⇒ niveau recommandé pour les fichiers de données s'il existe suffisamment de disques (autrement on utilise généralement le RAID 5)
- Appelé également architecture **SAME** (*Strip And Mirror Everything*)



Chapitre 2 - Concepts de base sur le stockage de données

- Introduction
- I. Structures physiques de stockage
- III. Structures logiques de stockage sous Oracle
- IV. Gestion des tablespaces et des fichiers de données
- V. Systèmes RAID
- Quiz

Quiz

- 1 Le serveur hôte d'une base de données dispose d'un disque magnétique et d'un disque SSD. Comment peut-on stocker certaines tables sur le disque magnétique et d'autres qui nécessitent un accès rapide sur le disque SSD.
- 2 Décrivez le principe de **localité spatiale**. Comment est-il appliqué dans le stockage ?
- 3 Quelle est la différence principale entre **RAID 0** et **RAID 1** ?
- 4 À quoi sert un **segment d'annulation** dans une base de données Oracle ?
- 5 Quelle est la relation entre un **tablespace**, un **segment**, et une **extension** ?
- 6 Quelles sont les deux paramètres, PCTFREE et PCTUSED, utilisés pour le contrôle du remplissage d'un bloc Oracle ? Expliquez leur utilité.

Chapitre 3 - Les bases de l'architecture Oracle

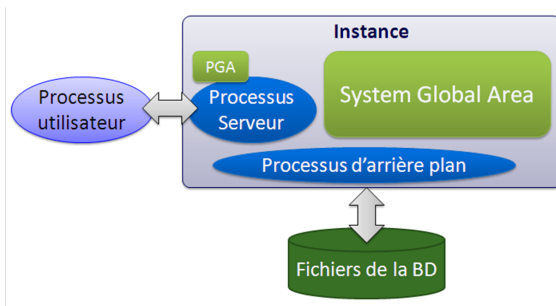
■ Introduction

- I. La base de données
- II. Instance
- III. Traitement d'un ordre SQL
- IV. Dictionnaire de données
- V. Architecture Multitenant
- Quiz

Architecture d'un serveur de BD Oracle

Serveur de BD Oracle : comporte deux éléments distincts

- 1 **La base de données** : ensemble de fichiers contenant les **données**, les informations sur les données (**le dictionnaire de données**) et les journaux de modifications sur les données (**fichiers log**)
- 2 **L'instance** : ensemble de **processus** et de **zones en mémoire vive** permettant l'interaction avec les fichiers de la BD.



Chapitre 3 - Les bases de l'architecture Oracle

■ Introduction

■ I. La base de données

- 1. Structure de la base
- 2. Fichiers de données
- 3. Fichiers de journalisation
- 4. Fichier de contrôle

■ II. Instance

■ III. Traitement d'un ordre SQL

■ IV. Dictionnaire de données

■ V. Architecture Multitenant

■ Quiz

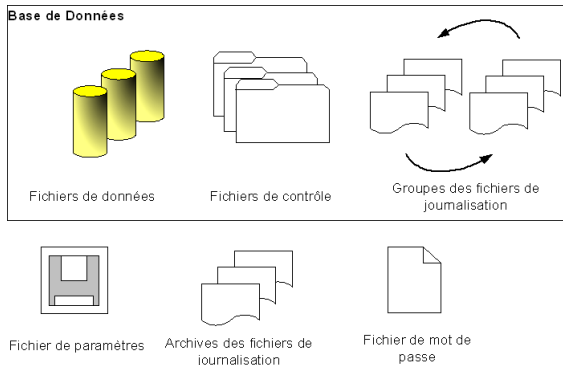
1. Structure de la base

Base de données

- Ensemble de fichiers stockés sur les disques durs du serveur hébergeant la base
- Porte un nom défini lors de sa création stocké dans le paramètre d'initialisation **DB_NAME** (Ex. DBsit)
- En complément l'emplacement de la BD sur le réseau peut être inscrit dans le paramètre **DB_DOMAIN** (Ex. infcom.rnu.tn)
- Nom global de la BD : DB_NAME.DB_DOMAIN (DBsit.infcom.rnu.tn)

1. Structure de la base

- Il existe différents types de fichiers : de données, de journalisation, etc.
- Les fichiers de paramètres, de contrôle, des mots de passe et les archives des fichiers de journalisation ne font pas partie de la BD



2. Fichiers de données (*Data Files*)

- Contiennent :
 - Les **tables** et les **index** créés par les utilisateurs
 - Les **données d'annulation** (images avant modification)
 - Le **dictionnaire de données**
- Groupés logiquement dans des tablespaces

3. Fichiers de journalisation (*Redo Log Files*)

- Contiennent un **historique** de toutes les **opérations de modification** faites sur les données (depuis une certaine période)
- Utilisés par le système pour reconstituer les fichiers de données en cas d'altération de ceux-ci ou de crash système
- Informations sur les fichiers journaux : vues **V\$LOGFILES** et **V\$LOG** du dictionnaire de données

3. Fichiers de journalisation (*Redo Log Files*)

Écriture dans les fichiers Redo Log

Multiplexée

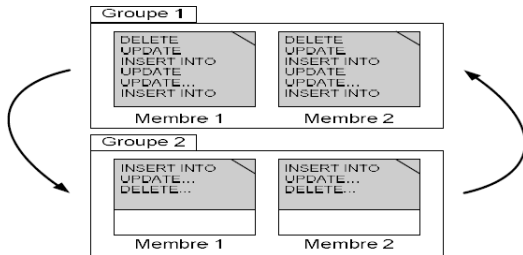
- Maintien de plusieurs **copies synchrones** d'un même fichier log
- **Groupe** : ensemble de fichiers multiplexés (identiques).
- **Membre** : un fichier appartenant à un groupe synchrone est dit membre de ce groupe (vue **V\$LOGFILE**)
- Il existe au minimum deux groupes par BD (vue **V\$LOG**).

Cyclique

- Chaque fois que les fichiers d'un groupe sont saturés, le système écrit dans les fichiers du groupe suivant (**basculement** ou **switch**)
- ⇒ Si le contenu des fichiers n'est pas archivé, il est définitivement perdu
- La BD est par défaut en mode NOARCHIVELOG.
Pour activer l'archivage : `ALTER DATABASE ARCHIVELOG`



3. Fichiers de journalisation (*Redo Log Files*)



- Ajout d'un nouveau groupe
`ALTER DATABASE ADD LOGFILE [GROUP numGroup] ('nomFichier1.LOG',
['nomFichier2.LOG',...] SIZE [K|M]`
- Ajout d'un fichier membre à un groupe (multiplexage)
`ALTER DATABASE ADD LOGFILE MEMBER 'nomFichier.LOG' TO GROUP
numGroupe`
- Basculement forcé d'un groupe à un autre
`ALTER SYSTEM SWITCH LOGFILE`

4. Fichier de contrôle (*Control File*)

- Contient des **informations sur la structure physique de la BD** :
 - Nom et date de création de la BD
 - Emplacement des fichiers de données et de journalisation
 - Position courante dans les log
 - Informations sur les points de reprise (*Checkpoint*)
 - Etc.
- Premier fichier consulté par l'instance après l'ouverture d'une BD (pour localiser les autres fichiers)
 - ⇒ **Multiplexage** fortement recommandé (pour des raisons de performances, 8 copies au maximum)
- Mis à jour automatiquement à chaque modification de la structure de la BD (Ex. ajout/suppression de fichiers de la BD, création d'un point de reprise, etc;)

4. Fichier de contrôle (*Control File*)

- Le paramètre **CONTROL_FILES** contient l'emplacement des fichiers de contrôle
- Consultation à partir de SQL*PLUS :
"SQL> show parameter control_files"
- La vue **V\$CONTROLFILE** du dictionnaire renseigne sur le fichier de contrôle

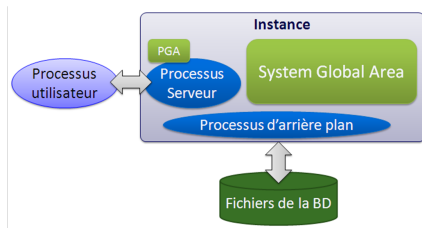
Chapitre 3 - Les bases de l'architecture Oracle

- Introduction
- I. La base de données
- **II. Instance**
 - 1. Vue d'ensemble
 - 2. La SGA
 - 3. La PGA
 - 4. Processus Serveur
 - 5. Processus d'arrière plan
- III. Traitement d'un ordre SQL
- IV. Dictionnaire de données
- V. Architecture Multitenant
- Quiz

1. Vue d'ensemble

Instance

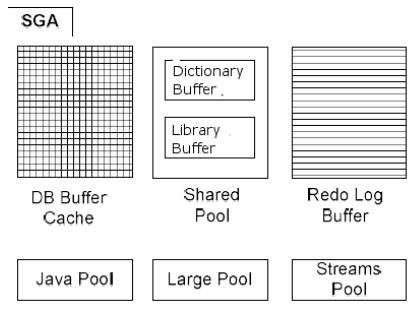
- Ouvre une seule BD à la fois
- Identifiée par un SID (*System ID*) stocké dans le paramètre **INSTANCE_NAME** (généralement, même valeur que pour DB_NAME)
- Se compose de :
 - 1 Zones en mémoire vive :
 - 1 *System Global Area* (SGA)
 - 2 *Program Global Area* (PGA)
 - 2 Processus serveur
 - 3 Processus d'arrière plan



2. La SGA

System Global Area

- Ensemble de tampons mémoire partagés par les processus de l'instance
- Allouée au démarrage de la BD et libérée à sa fermeture
- 3 zones obligatoires et 3 zones optionnelles
- Taille maximale : paramètre **SGA_MAX_SIZE**



2. La SGA - a. Le pool partagé (*shared pool*)

- Rôle : essentiellement, **accélérer l'analyse des requêtes**
- Composé de deux tampons : le **Library Buffer** et le **Dictionary Buffer**
- **Library Buffer**
 - Tampon stockant des informations sur les requêtes SQL les plus récemment exécutées
 - Informations : texte, version compilée et plan d'exécution
 - Remplacement : LRU
- **Dictionary Buffer**
 - Tampon stockant les données du dictionnaire les plus récemment utilisées (tables, droits des utilisateurs, etc.)
 - Remplacement : LRU
- Taille du pool partagé : paramètre **SHARED_POOL_SIZE**



2. La SGA - b. Le tampon de la BD (*DataBase Buffer*)

- Tampon mémoire réservé au stockage des blocs de données Oracle (blocs de tables, d'index, d'annulation)
- Rôle :
- Remplacement
 - Géré selon la stratégie LRU, appliquée en premier aux blocs dont le contenu a été modifié (**Dirty Blocks**)
 - Raison : ⇒ Les blocs modifiés sont écrits en premier sur le disque pour minimiser l'information perdue et le temps de sa restauration

2. La SGA - b. Le tampon de la BD (*DataBase Buffer*)

■ **DB_CACHE_SIZE**

- Paramètre contrôlant la taille du *DataBase Buffer*

⇒ doit être un multiple de DB_BLOCK_SIZE.

Nombre maximal de blocs dans le tampon =

N.B. si un tablespace *tbs* utilise une taille de bloc \neq DB_BLOCK_SIZE

- Nécessité d'initialiser au préalable un tampon supplémentaire dont la taille est un multiple de la taille du bloc de *tbs*

⇒ Paramètre (selon la taille de bloc utilisée dans *tbs*) : **DB_***nk***_CACHE_SIZE** ,
avec $n \in \{2, 4, 8, 16, 32\}$

```
ALTER SYSTEM SET DB_32K_CACHE_SIZE=300M
```

2. La SGA - c. Le tampon de journalisation (*Redo Log Buffer*)

- Tampon mémoire réservé au stockage de l'historique des opérations de modification faites sur les données
- Périodiquement le contenu du tampon est écrit dans les fichiers de journalisation (par exemple, suite à un commit)
- Taille définie par le paramètre **LOG_BUFFER**
- Rôle : Écriture par lot (*batch*) des entrées log au lieu d'une écriture par entrée (*i.e.* par opération de modification)

2. La SGA - d. Autres zones mémoires

■ Java Pool :

- Plusieurs routines internes d'Oracle utilise du code Java
Ex. import/export de données
- Zone de mémoire utilisée par la machine virtuelle Java intégrée. Paramètre **JAVA_POOL_SIZE**

■ Large Pool : stocke essentiellement, les données sur les sessions des utilisateurs. Paramètre **LARGE_POOL_SIZE**

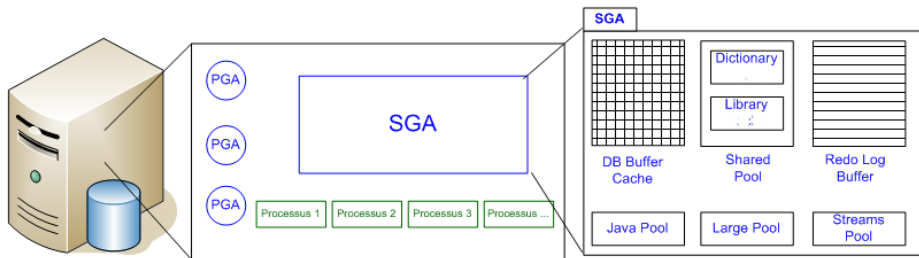
■ Streams Pool : Zone de mémoire utilisée pour l'échange d'informations entre processus. Paramètre **STREAMS_POOL_SIZE**

■ SGA Fixe : inclut les informations sur l'état de la base, l'instance, les verrous, etc.

2. La SGA - e. Gestion automatique

- Tailles des différentes zones mémoires : gérables manuellement (par le DBA) ou automatiquement (par Oracle)
- **Gestion manuelle** : donner une taille à chacune des zones. La somme des tailles doit être \leq **SGA_MAX_SIZE**
- **Gestion automatique**
 - Activée si le paramètre **SGA_TARGET** \neq 0.
 - La valeur choisie pour **SGA_TARGET** doit être \leq **SGA_MAX_SIZE**.
 - Paramètre dynamique : `ALTER SYSTEM SET SGA_TARGET = 500 M`

2. La SGA - En résumé...



SERVEUR BD =
Instance
 +
BD (fichiers)

INSTANCE =
Zones mémoires
 +
Processus

ZONES MÉMOIRES =
3 + 3

3. PGA (*Program Global Area*)

- Zone mémoire privée des différents processus, une PGA par processus
- PGA agrégée : PGA totale allouée à tous les processus
- Paramètre **PGA_AGGREGATE_TARGET** : taille de la PGA agrégée (Oracle se charge de la répartition entre les différents processus)

4. Processus serveur

■ **Processus Utilisateur** (PU ou *User Process*) :

- Programme demandant une interaction avec le Serveur de BD (Ex. SQL*PLUS)
- Ne fait pas partie du serveur de BD et n'interagit avec celui-ci qu'au travers d'un **processus serveur**

■ **Processus Serveur** (PS ou *Server Process*) :

- Prend en charge les appels générés par un ou plusieurs processus utilisateurs et leur renvoie les résultats
- Notamment : chargement des données dans le tampon de BD et écriture dans le tampon Redo Log
- Deux modes : serveur dédié et serveur partagé (spécifié au moment de la création de la BD)
 - **Dédié** : un processus serveur pour chaque processus utilisateur
 - **Partagé** : un processus serveur pour plusieurs processus utilisateur

5. Processus d'arrière plan – a. Rôle et présentation

Rôle :

- 1 Gérer la SGA et écriture à partir de la SGA vers les fichiers de la BD
- 2 Assurer le bon fonctionnement de l'instance, même en cas de crash

Déclenchement :

- Au démarrage de l'instance pour certains
- Ultérieurement suite à des événements particuliers (ex. Tampon saturé) pour d'autres

5. Processus d'arrière plan – b. Processus DBW (*DataBase Writer*)

- Rôle : écriture dans les fichiers de données (sur disque) des blocs devant sortir du tampon de la base de données
- Sur les systèmes multi-processeurs, multi-disques, il est recommandé d'en avoir plusieurs (jusqu'à 20) numérotés comme suit : DBW0, DBW1,..., DBW9, DBWa,..., DBWj
- Paramètre **DB_WRITER_PROCESSES** : détermine le nombre de processus DBW

5. Processus d'arrière plan – c. Processus LGWR (*LoG WRiter*)

- Rôle : écriture séquentielle par lot dans les fichiers de journalisation (Redo log) des entrées log devant sortir du tampon Redo Log
- Déclenchement
 - 1 Suite à une validation (COMMIT);
 - 2 Toute les trois secondes (*cf.* 5.d. Processus CKPT);
 - 3 Quand $\frac{1}{3}$ du tampon Redo Log est plein :
 - 4 Avant que DBW n'écrive sur disque les blocs modifiés (*Dirty*) : pour éviter les modifications non journalisées en cas de crash système.

5. Processus d'arrière plan – d. Processus CKPT (*CheckPoint*)

■ Rôle

- 1 Forcer l'écriture de tous les blocs modifiés "*Dirty*" (et donc du contenu du tampon Redo log) sur disque ⇔ Création d'un **point de reprise** (*CheckPoint*)
- 2 Consigner l'estampille temporelle (*System Change Number* ou **SCN**) de l'écriture :
 - dans le fichier de contrôle,
 - dans les en-têtes des fichiers de données et de journalisation

⇒ En cas crash, le SCN indique le point de départ de la restauration

■ Intérêt :

- Restauration plus rapide en cas de crash système
- ⇒ Les entrées Redo log datant d'avant le SCN ne sont pas rejouées

5. Processus d'arrière plan – d. Processus CKPT (*CheckPoint*)

■ Déclenchement de CKPT :

- A la demande : `ALTER SYSTEM CHECKPOINT`
- A la fermeture d'une BD (excepté `SHUTDOWN ABORT`)
- Lors de la mise hors ligne d'un tablespace
- Toute les 3 secondes
- Lorsque LGWR bascule d'un groupe à un autre :

5. Processus d'arrière plan – e. Le processus SMON (*System MONitor*)

Déclenchement : à l'ouverture de la BD

Rôle : récupération de la BD suite à un arrêt brutal

- Détection de l'arrêt brutal de la BD :
 - 1 Comparaison des SCN des en-têtes des fichiers de données et de journalisation avec le SCN du fichier de contrôle
 - 2 Si différents, une récupération est nécessaire
- Récupération : deux étapes
 - 1 **Roll Forward** : rejouer toutes les transactions, validées ou non, depuis le dernier point de reprise (fichiers de journalisation + fichier de contrôle)
 - 2 **Roll Back** : annuler les transactions non validées (segments d'annulation)

5. Processus d'arrière plan – e. Le processus SMON (System MONitor)

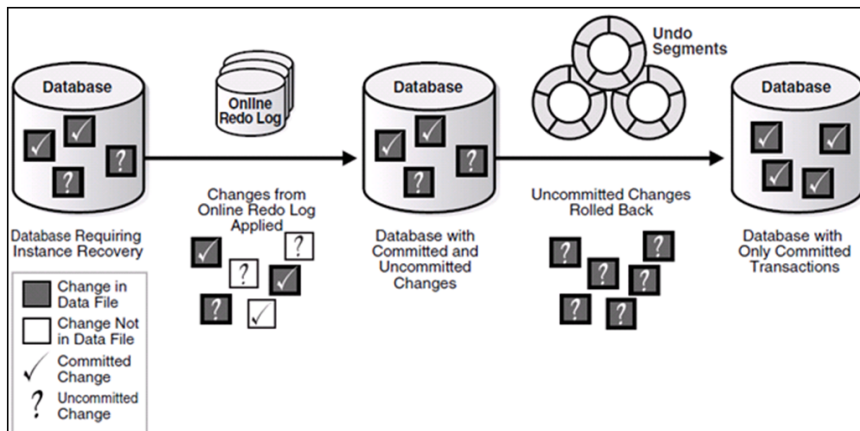


Figure: Roll Forward (Redo) suivi d'un Roll Back (Undo) [Oracle© Database Concepts 11g Release 2 (11.2) E10713-05, 2010]

5. Processus d'arrière plan – f. Le processus PMON (*Process MONitor*)

PMON, essentiellement

- Détection et nettoyage suite au plantage d'un processus utilisateur
 - Exemple :
 - 1 Un PU ouvre une session et lance une requête UPDATE
 - 2 Les données concernées sont verrouillées et le UPDATE exécuté
 - 3 Le PU n'est plus présent et n'a pas fermé la session (n'a ni validé ni annulé la transaction)!!
- ⇒ PMON détecte le blocage, annule la transaction et libère les verrous

5. Processus d'arrière plan – e. Processus ARC (ARChive)

- Rôle : archivage des fichiers de journalisation
- Déclenchement : basculement d'un groupe de fichiers à un autre
 - Suite à la saturation du groupe
 - Suite à ALTER SYSTEM SWITCH LOGFILE
- Il peut en exister plusieurs : ARC_n
- ARC n'écrit pas directement sur la bande de sauvegarde, mais sur les disques de la machine hôte
- Paramètre d'initialisation **LOG_ARCHIVE_DEST_n** : détermine l'emplacement de la n^{eme} copie d'archive
`ALTER SYSTEM SET log_archive_dest_1='location=chemin'`
`SCOPE=spfile`

En résumé...

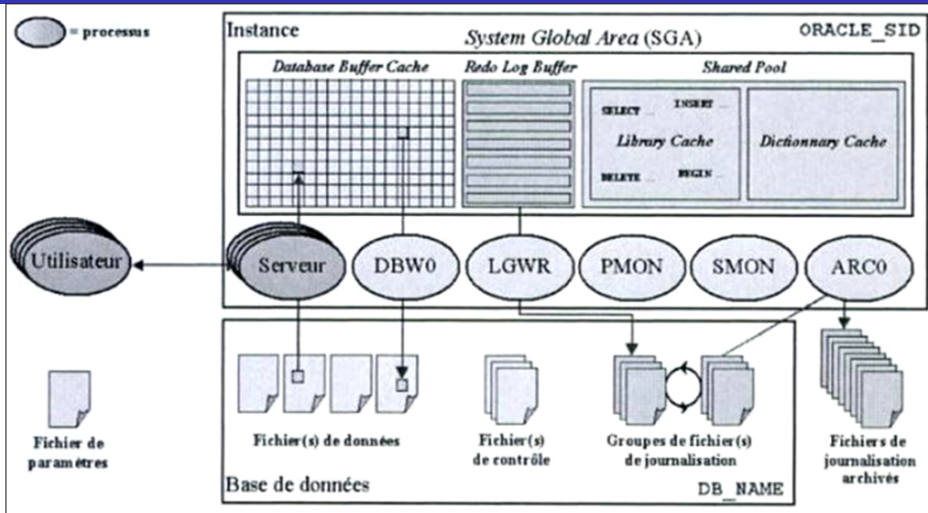


Figure: [Oracle 11g Administration, Olivier Heurtel, eni Editions, Nov. 2008]

Chapitre 3 - Les bases de l'architecture Oracle

- Introduction
- I. La base de données
- II. Instance
- **III. Traitement d'un ordre SQL**
 - 1. Traitement d'un SELECT
 - 2. Traitement d'un ordre DML (UPDATE/INSERT/DELETE)
- IV. Dictionnaire de données
- V. Architecture Multitenant
- Quiz

1. Traitement d'un SELECT

1 Analyse (*Parse*)

- 1 Le PU envoie la requête au PS
- 2 Si la requête n'a pas déjà été analysée précédemment (consultation du *Library Buffer*), le PS l'analyse
- 3 Le PS doit vérifier si les objets interrogés par la requête existent et si l'utilisateur a le droit de les consulter. Si ces informations ne sont pas présentes dans le *Dictionary Buffer*, le PS les charge
- 4 Génération et enregistrement du plan d'exécution (si non présent dans le *Library Buffer*)

2 Exécution : exécution du plan de la requête. Si les blocs des données interrogées ne sont pas dans le tampon de la BD, le PS les charge dans le tampon à partir,

- 1 du fichier de données si les données n'ont pas été modifiées par une transaction non encore validée
- 2 du tablespace d'annulation (undo), sinon

3 Extraction : le PS renvoie au PU le résultat de la requête



2. Traitement d'un ordre DML (UPDATE/INSERT/DELETE)

1 Analyse (*Parse*)

- Idem. que pour une sélection

2 Exécution

- 1 Si les blocs de données et les blocs d'annulation ne se trouvent pas déjà dans le tampon de la BD, le PS les charge.
- 2 Le PS place des verrous sur les tuples à modifier
- 3 Le PS place l'image avant des données dans le bloc d'annulation.
- 4 Les modifications sont enregistrées dans les blocs de données qui sont alors marqués comme "*Dirty*"
- 5 Les informations sur la modification sont consignées dans le tampon de journalisation

Chapitre 3 - Les bases de l'architecture Oracle

- Introduction
- I. La base de données
- II. Instance
- III. Traitement d'un ordre SQL
- **IV. Dictionnaire de données**
 - 1. Rappel
 - 2. Vues statiques
 - 3. Vues dynamiques
- V. Architecture Multitenant
- Quiz

1. Rappel

Dictionnaire de données

- 1 Ensemble de vues et de tables contenant les informations sur toutes les composantes logiques et physiques de la BD et de l'instance
- 2 Créé et mis à jour par le système lorsque des ordres DDL sont exécutées (CREATE/ALTER/DROP)
- 3 Propriété de l'utilisateur SYS \Rightarrow stocké dans le tablespace SYSTEM.
- 4 Les données des tables sont codées et modifiables uniquement par le système. Accessibles uniquement via des vues
- 5 Deux types de vues :
 - **Vues statiques** : basées sur les "vraies" tables du dictionnaire
 - **Vues dynamiques** : basées sur de l'information se trouvant en mémoire vive (Ex. SGA) ou extraite du fichier de contrôle



2. Vues statiques

Trois types :

- Vues préfixées par **USER_**
 - Donnent accès aux informations sur les objets **possédés** (créés) par l'utilisateur
 - Exemple : USER_TABLES : tables créées par l'utilisateur
- Vues préfixées par **ALL_**
 - Donnent accès aux informations sur les objets (tables, index, vues, triggers, etc.) **accessibles** par l'utilisateur
 - Exemple : ALL_TABLES : toutes les tables que l'utilisateur peut voir
- Vues préfixées par **DBA_**
 - Interrogeables uniquement par les **administrateurs**
 - Donnent accès aux informations sur **tous** les objets de la BD
 - Exemple : DBA_TABLES : toutes les tables de la BD

3. Vues dynamiques

- Contiennent l'information sur l'**activité en cours de la BD et de l'instance**
- Préfixées par **V\$**
- **V\$FIXED_TABLE** : contient la liste des vues dynamiques. Quelques vues dynamiques :

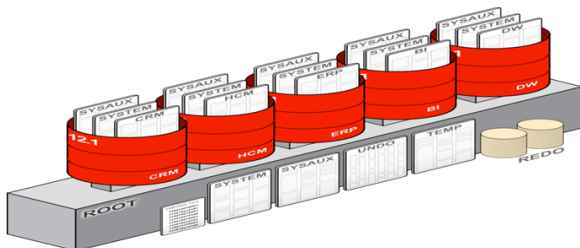
Nom	Description
V\$DATABASE	Informations sur la BD
V\$INSTANCE	Informations sur l'instance
V\$SGA	Informations résumées sur la SGA
V\$PARAMETER	Informations sur les paramètres d'initialisation
V\$SQL	Informations sur les requêtes SQL exécutées

Chapitre 3 - Les bases de l'architecture Oracle

- Introduction
- I. La base de données
- II. Instance
- III. Traitement d'un ordre SQL
- IV. Dictionnaire de données
- **V. Architecture Multitenant**
 - 1. Pluggable Databases
 - 2. Opérations usuelles
- Quiz

1. Pluggable Databases

- 1 Historiquement (du moins jusqu'à la version 12g) une instance Oracle exécute les traitements sur une seule BD et une BD ne peut être utilisée que par une seule instance.
- 2 A partir de la version 12g, il est désormais possible qu'une instance exécute des traitements sur plusieurs "Pluggable" DBs.
- 3 Les pluggables DBs sont exportables d'une instance à une autre et partagent plusieurs composants (SGA, UNDO, log, etc.)
- 4 Les composants en commun sont regroupés dans une BD dite container (Container Database)



6. Pluggable Databases

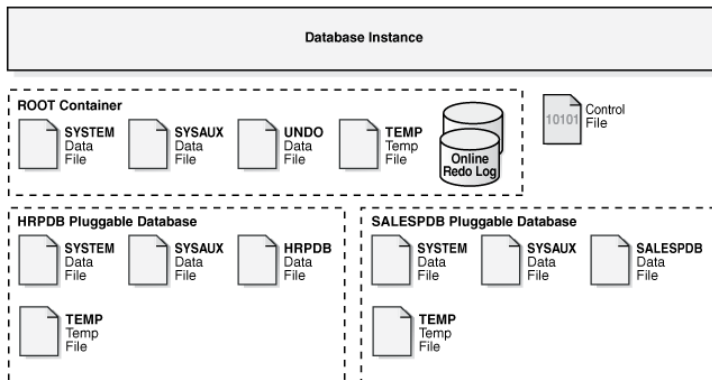


Figure:

6. Pluggable Databases

Specify Database Identification Details

19^c ORACLE Database

- Database Operation
- Creation Mode
- Deployment Type
- Database Identification**
- Storage Option
- Fast Recovery Option
- Database Options
- Configuration Options
- Management Options
- User Credentials
- Creation Option
- Summary
- Progress Page
- Finish

Provide a unique database identifier information. An Oracle database is uniquely identified by a Global database name, typically of the form "name.domain".

Global database name:

SID:

Service name:

☒ **Create as Container database**

A Container database can be used for consolidating multiple databases into a single database, and it enables database virtualization. A Container database (CDB) can have zero or more pluggable databases (PDB).

☒ Use Local Undo tablespace for PDBs

☐ Create an empty Container database

☒ Create a Container database with one or more PDBs

Number of PDBs:

PDB name:

Figure:

6. Pluggable Databases

- Création d'une pluggable database

```
CREATE PLUGGABLE DATABASE newDBtest ADMIN USER isitcom IDENTIFIED BY isitcom  
FILE_NAME_CONVERT = ('C:\app\product\18.0.0\oradata\XE\pdbseed',  
'C:\app\product\18.0.0\oradata\XE\newDBTest');
```

- Se placer sur une PDB particulière

```
ALTER SESSION SET CONTAINER = newDBTest
```

- Suppression

```
ALTER PLUGGABLE DATABASE pdb3 CLOSE; DROP PLUGGABLE  
DATABASE pdb3 INCLUDING DATAFILES;
```

- Ouverture

```
ALTER PLUGGABLE DATABASE pdb2 OPEN READ WRITE;
```

- Export/import

```
CREATE PLUGGABLE DATABASE pdb2 USING  
'/u01/app/oracle/oradata/cdb1/pdb2/pdb2.xml' NOCOPY TEMPFILE  
REUSE;
```



Chapitre 3 - Les bases de l'architecture Oracle

- Introduction
- I. La base de données
- II. Instance
- III. Traitement d'un ordre SQL
- IV. Dictionnaire de données
- V. Architecture Multitenant
- **Quiz**

Quiz

- 1 Un administrateur d'une BD constate une lenteur dans la journalisation des logs. Quelles actions peut il entreprendre?
- 2 Pourquoi est il nécessaire de consigner dans le log `Begin transaction`, `Commit` et `Rollback`?
- 3 Expliquez le rôle du `Redo Log Buffer` dans la SGA.
- 4 À quoi sert le fichier de contrôle (`Control File`) dans une base Oracle ?
- 5 Quelles sont les différences principales entre un `Shared Server` et un `Dedicated Server` ?
- 6 Quel rôle joue le SCN dans la reprise sur pannes?



Chapitre 4 - Gestion de l'instance

- I. Fichier de paramètres
 - 1. Présentation
 - 2. PFILE vs. SPFILE
 - 3. Modification des paramètres d'initialisation
- II. Démarrage et arrêt d'un serveur de BD Oracle
- III. Diagnostic et surveillance d'un serveur de BD Oracle

1. Présentation

Fichier de paramètres

- Consulté par l'instance à son démarrage et avant l'ouverture de la BD. **Ne fait pas partie de la BD**
- Contient l'ensemble des **paramètres d'initialisation de l'instance**
 - Paramètres d'emplacement : DB_CREATE_FILE_DEST, CONTROL_FILES, etc.
 - Paramètres de limites et d'allocation mémoire : DB_BLOCK_SIZE, DB_CACHE_SIZE, SGA_MAX_SIZE, etc.
- Il existe deux types de fichiers de paramètres : le PFILE (historique) et le SPFILE (depuis la version 9i)

1. Présentation

■ Exemple d'un fichier de paramètres PFILE

```
# Initialization Parameter File: initORCL.ora
db_name                = ORCL
instance_name          = ORCL
control_files          = (
                        home/dba01/ORADATA/u01/control01dba01.ctl,
                        home/dba01/ORADATA/u02/control01dba02.ctl)
db_block_size          = 4096
db_cache_size          = 4M
shared_pool_size       = 50000000
java_pool_size         = 50000000
core_dump_dest         = /home/dba01/ADMIN/CDUMP
undo_management        = AUTO
undo_tablespace        = UNDOTBS
. . .
```

2. PFILE vs. SPFILE

PFILE (Parameter File)	SPFILE (Server Parameter File)
Fichier texte	Fichier binaire
Modifiable via un éditeur texte	Modifiable uniquement via SQL
Présent sur chaque machine pouvant démarrer l'instance (problème de synchronisation)	Centralisé : un seul fichier sur la machine hôte de la BD
Nommé initSID.ora	Nommé spfileSID.ora
Modification à froid	Modifiable à chaud (pour les paramètres dynamiques)

- Possibilité de créer un PFILE à partir d'un SPFILE et inversement :
`CREATE PFILE/SPFILE [nomFichier1] FROM SPFILE/PFILE [nomFichier2]`
- Lors du démarrage d'une instance, le serveur cherche un spfile et sinon un pfile

3. Modification des paramètres d'initialisation

Instruction SQL pour modifier les paramètres d'initialisation

```
ALTER SYSTEM SET nomParam = nouvelleValeur  
[SCOPE = [BOTH | MEMORY | SPFILE]]
```

- SCOPE = SPFILE : si le paramètre est **statique** (modifiable uniquement à froid), ou bien s'il est souhaité que les changements ne prennent effet qu'au prochain démarrage de l'instance
- SCOPE = MEMORY : si le paramètre est **dynamique** (modifiable à chaud). Les changements affectent uniquement l'instance en cours
- SCOPE = BOTH : si le paramètre est **dynamique** et les modifications concernent aussi bien l'instance en cours que les prochaines

3. Modification des paramètres d'initialisation

- Si l'instance démarre à l'aide d'un PFILE
 - Seule l'option `MEMORY` est utilisable avec `ALTER SYSTEM...`
 - Le DBA doit retranscrire les modifications dans le PFILE
- Pour consulter la valeur courante d'un paramètre *nomParametre*
 - 1 A partir de SQL*PLUS: `"SQL>show parameter nomParametre"`
 - 2 SQL : `"SELECT name, value FROM V$PARAMETER WHERE name = nomParametre"`
- Pour consulter la valeur stockée dans un SPFILE d'un paramètre *nomParametre*
 - SQL : `"SELECT name, value FROM V$SPPARAMETER WHERE name = nomParametre"`

Chapitre 4 - Gestion de l'instance

- I. Fichier de paramètres
- II. Démarrage et arrêt d'un serveur de BD Oracle
 - 1. Démarrage
 - 2. Arrêt
- III. Diagnostic et surveillance d'un serveur de BD Oracle

1. Démarrage - a. Syntaxe

Démarrage - Syntaxe (non exhaustive)

```
STARTUP [FORCE] [RESTRICT] [PFILE=nomPFILE]
        [OPEN | MOUNT | NOMOUNT]
```

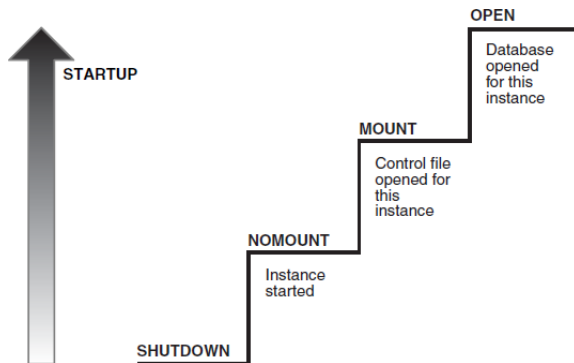


Figure: Phases de démarrage [Oracle© Database Concepts 11g Release 2 (11.2)E10713-05,2010]



1. Démarrage - b. STARTUP NOMOUNT

STARTUP NOMOUNT

- 1 Un SPFILE (spfile%.ora) est cherché. Si non trouvé un pfile (init%.ora) est cherché et utilisé à la place
 - 2 La SGA est allouée selon les paramètres d'initialisation
 - 3 Les processus d'arrière plan sont lancés
- La BD n'est ni montée ni ouverte
⇒ les vues dynamiques relatives à l'instance (V\$INSTANCE, V\$SGA, etc.) sont interrogeables, mais **pas celles relatives à la BD** (V\$DATABASE)
 - Pour monter (resp. ouvrir) la BD ultérieurement :
ALTER DATABASE MOUNT (resp. ALTER DATABASE OPEN)

1. Démarrage - c. STARTUP MOUNT

STARTUP MOUNT

- 1 Ouverture du fichier de contrôle de la BD (localisation à partir du paramètre d'initialisation `CONTROL_FILES`)
 - 2 Si le fichier de contrôle est multiplexé et que les copies ne sont pas identiques, l'instance reste à l'état `NOMOUNT`
 - 3 Depuis le fichier de contrôle, extraction des noms, des emplacements et des états des fichiers de données et de journalisation. Ces fichiers ne sont à ce stade **ni ouverts ni vérifiés**
- Connexion à la BD possible uniquement pour les administrateurs. Ils peuvent y faire certaines tâches d'administration
Ex. Renommage/déplacement de fichiers, activation de la journalisation, etc.
(Pourquoi ne sont elles pas possibles en mode `NOMOUNT`?)
 - La BD est dite associée à l'instance, les vues dynamiques relatives à la BD sont consultables (`V$DATABASE`), mais **pas les vues statiques**
 - Pour la rendre accessible à tous le monde ultérieurement :

```
ALTER DATABASE OPEN
```

1. Démarrage - d. STARTUP OPEN

STARTUP ou STARTUP OPEN

- 1 Ouverture des fichiers de journalisation et de données
- 2 SMON est lancé. Si les SCN des en-têtes des fichiers de données et de journalisation sont différents du SCN apparaissant dans le fichier de contrôle
⇒ **Roll Forward + Roll Back.**
- Après l'ouverture, la BD est disponible à une utilisation "normale" pour tous les utilisateurs

1. Démarrage - e. Autres options de démarrage

STARTUP FORCE

- 1 Interrompt l'instance en cours, puis exécute un démarrage normal

⇒ Equivalent à SHUTDOWN ABORT + STARTUP

STARTUP RESTRICT

- N'autorise l'accès à la BD qu'aux administrateurs et les utilisateurs disposant du privilège RESTRICTED SESSION
- Utilisé pour effectuer certaines opérations d'administration qui nécessite que la BD soit ouverte, mais qu'il est préférable de réaliser sans utilisateur connecté
- Ex. Import/export de données, réorganisation d'une table etc.
- Pour quitter le mode restreint

```
ALTER SYSTEM DISABLE RESTRICTED SESSION
```



2. Arrêt - a. Syntaxe

Syntaxe (non exhaustive)

SHUTDOWN [NORMAL | TRANSACTIONAL | IMMEDIATE | ABORT]

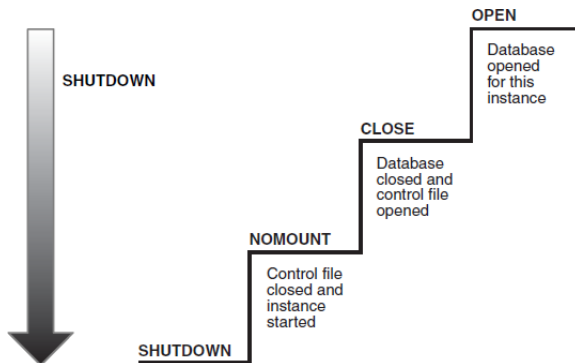


Figure: Phases d'arrêt [Oracle© Database Concepts 11g Release 2 (11.2)E10713-05,2010]

2. Arrêt - b. Options d'arrêt

SHUTDOWN ou SHUTDOWN NORMAL

- Attend la déconnexion de tous les utilisateurs et l'arrêt de toutes les transactions en cours
- Effectue un checkpoint avant l'arrêt

SHUTDOWN TRANSACTIONAL

- Attend l'arrêt de toutes les transactions (mais pas la déconnexion de tous les utilisateurs)
- Effectue un checkpoint avant l'arrêt

2. Arrêt - b. Options d'arrêt

SHUTDOWN IMMEDIATE

- Fermeture instantanée sans attendre les sessions/transactions en cours.
- Effectue un rollback des transactions non validées, puis un checkpoint avant l'arrêt

SHUTDOWN ABORT

- Fermeture instantanée sans attendre les sessions/transactions en cours.
- Pas de checkpoint avant l'arrêt \Rightarrow recouvrement au démarrage
- À utiliser uniquement si blocage des autres types d'arrêt

2. Arrêt - b. Options d'arrêt

En résumé...

Database Behavior	ABORT	IMMEDIATE	TRANSACTIONAL	NORMAL
Permits new user connections	No	No	No	No
Waits until current sessions end	No	No	No	Yes
Waits until current transactions end	No	No	Yes	Yes
Performs a checkpoint and closes open files	No	Yes	Yes	Yes

Figure: Options d'arrêt [Oracle© Database Concepts 11g Release 2 (11.2)E10713-05,2010]

Chapitre 4 - Gestion de l'instance

- I. Fichier de paramètres
- II. Démarrage et arrêt d'un serveur de BD Oracle
- III. Diagnostic et surveillance d'un serveur de BD Oracle
 - 1. Vue d'ensemble
 - 2. Le fichier d'alertes alertSID.log
 - 3. Les fichiers trace des processus d'arrière plan
 - 4. Les fichiers trace des processus utilisateur
 - 5. Outils de diagnostic

1. Fichiers de diagnostic

Problème et incident

- Problème : erreur critique. A un identificateur unique (ORA-...)
- Incident : occurrence d'un problème

Fichiers de diagnostic

- Informations relatives aux événements significatifs qui se sont produits,
- Permettent la résolution des incidents et l'amélioration de la gestion quotidienne de la base de données.

Plusieurs types

- Fichier **alertSID.log** : informations sur le fonctionnement au quotidien de la BD
- Fichiers **trace de processus d'arrière-plan** : informations sur les erreurs détectées par les processus d'arrière plan
- Fichiers **trace utilisateur** : informations sur les processus utilisateurs



2. Le fichier d'alertes alertSID.log

- Chaque instance possède un fichier d'alertes, créé au démarrage si besoin.
Exemple d'informations
 - Création, démarrages/arrêts
 - Modification de la structure (tablespaces, fichiers de données)
 - Certains types d'erreurs : erreurs du SE (ORA-07445), manque de mémoire pour le Pool partagé (ORA-04031)
 - Problèmes relatifs à l'écriture et l'archivage des fichiers de journalisation
 - Alertes (Tablespace presque plein, disque presque plein, etc.)
 - Etc.
- Contient également des pointeurs sur des fichiers trace qui contiennent des informations plus détaillées.
- En perpétuel grossissement ⇒ doit être géré par le DBA
- Format xml disponible (depuis la version 11g) : log.xml

3. Les fichiers trace des processus d'arrière plan

- Consignent les erreurs détectées par les processus d'arrière plan,
- Permettent de diagnostiquer et de corriger les erreurs.
- Créés lorsqu'un processus d'arrière plan détecte une erreur.
- Convention de nommage : SID_nomProcessus_idProcessus.trc (Ex. DBisitcom_lgwr_23004.trc)

4. Les fichiers trace des processus processus utilisateur

- Créés par le processus utilisateur,
- Peuvent être générés par un processus serveur,
- Contiennent des statistiques sur les instructions SQL (utiles pour le réglage ou *tuning*)
- Contiennent des messages sur les erreurs constatées au cours d'une session.
- Convention de nommage : SID_ORA_PID.trc (Ex. DBisitcom_ora_20045.trc)
- Activation de la collecte de statistiques (paramètre **SQL_TRACE**):
ALTER SESSION SET SQL_TRACE=TRUE

5. Outils de diagnostic

■ Référentiel de Diagnostic Automatique (*Automatic Diagnostic Repository*)

- Répertoire structuré, centralisant tous les fichiers de diagnostic (depuis la version 11g)
- Emplacement défini par le paramètre : `DIAGNOSTIC_DEST`

■ La vue `v$diag_info` renseigne sur les emplacements des fichiers d'alerte et de trace

```
SQL> SELECT name, value FROM v$diag_info;
```

NAME	VALUE
Diag Enabled	TRUE
ADR Base	d:\app\khaled
ADR Home	d:\app\khaled\diag\rdbms\dbisitcom\dbisitcom
Diag Trace	d:\app\khaled\diag\rdbms\dbisitcom\dbisitcom\trace
Diag Alert	d:\app\khaled\diag\rdbms\dbisitcom\dbisitcom\alert
Diag Incident	d:\app\khaled\diag\rdbms\dbisitcom\dbisitcom\incident
Diag Cdump	d:\app\khaled\diag\rdbms\dbisitcom\dbisitcom\cdump
Health Monitor	d:\app\khaled\diag\rdbms\dbisitcom\dbisitcom\hm
Default Trace File	d:\app\khaled\diag\rdbms\dbisitcom\dbisitcom\trace\db isitcom_ora_2600.trc

NAME	VALUE
Active Problem Count	2
Active Incident Count	4

11 ligne(s) sélectionnée(s).



5. Outils de diagnostic

■ Outils (consultation du référentiel ADR)

- *Entreprise Manager : Support WorkBench*
- L'utilitaire en ligne de commandes *adrci (ADR Command Interpreter)*.

Quelques commandes :

Commande : <code>adrci></code>	Effet
<code>SHOW ALERT</code>	Affiche les log du fichier d'alertes
<code>SHOW INCIDENT</code>	Affiche les incidents répertoriés
<code>SHOW INCIDENT -MODE DETAIL -P "incident_id=7465"</code>	Affiche les détails de l'incident 7465 et l'emplacement des fichiers de trace relatifs
<code>SHOW TRACEFILE</code>	Affiche l'emplacement des fichiers de trace

Chapitre 5 - Gestion des utilisateurs

■ Introduction

- I. Gestion basique des comptes des utilisateurs
- II. Gestion des privilèges
- III. Gestion des rôles
- IV. Gestion des profils
- V. Audit de l'activité des utilisateurs
- Quiz

Introduction

Rôle de l'administrateur de sécurité et des ressources

- 1 Définir une politique de sécurité
- 2 Gérer les **comptes des utilisateurs**
- 3 Assurer l'attribution et la révocation des droits (**privilèges** et **rôles**)
- 4 Gérer les ressources système (**profils**)
- 5 Effectuer des **audits**

Ce chapitre...

- 1 Gestion basique des comptes des utilisateurs
- 2 Gestion des privilèges
- 3 Gestion des rôles
- 4 Gestion des profils
- 5 Audit de l'activité des utilisateurs



Chapitre 5 – Gestion des utilisateurs

- Introduction
- I. Gestion basique des comptes des utilisateurs
 - 1. Notions de schéma et d'utilisateur
 - 2. Création d'un nouvel utilisateur
 - 3. Modification et suppression
- II. Gestion des privilèges
- III. Gestion des rôles
- IV. Gestion des profils
- V. Audit de l'activité des utilisateurs
- Quiz

1. Notions de schéma et d'utilisateur

■ **Utilisateur** de BD Oracle (*User*) :

- Identifié par un **login** avec lequel il se connecte au système
- Possède un ensemble de droits ou **privilèges** associés à son login.
- Dispose d'un espace de stockage par défaut (**tablespace**) dans lequel sont stockés les objets qu'il crée

■ **Schéma** (*Schema Objects*)

- collection de **tous les objets** (tables, vues, index, etc.) **créés par un utilisateur** précis (N.B. n'inclut pas les fichiers et les tablespaces)

■ Oracle assimile un schéma à son utilisateur ⇒

- Chaque utilisateur possède un seul schéma et réciproquement
- Le login sert à identifier à la fois un utilisateur et son schéma
- La création d'un utilisateur entraîne celle de son schéma
- Accès à un objet : login.nomObjet (Ex. sys.dba_tablespaces)

2. Création d'un nouvel utilisateur – a. Enchaînement typique

- 1 Choisir un nom d'utilisateur et un mécanisme d'authentification,
- 2 Décider des tablespaces dans lesquels l'utilisateur peut stocker certains de ses objets
- 3 Affecter un tablespace par défaut et un tablespace temporaire
- 4 Décider des quotas pour chaque tablespace,
- 5 Créer un utilisateur
- 6 Accorder des privilèges et des rôles a l'utilisateur

2. Création d'un nouvel utilisateur – b. Ordre SQL

CREATE USER *login*

- > Mode d'identification, Oracle ou SE ou LDAP
IDENTIFIED [by *motDePasse*|EXTERNALLY|GLOBALLY AS *nomExterne*]
- > Tablespace par défaut où seront stockés les objets créés par l'utilisateur
[DEFAULT TABLESPACE *nomTablespace*]
- > Tablespace temporaire des requêtes exécutées par l'utilisateur
[TEMPORARY TABLESPACE *nomTablespaceTemporaire*]
- > Espace maximal que l'utilisateur peut utiliser dans les tablespaces
[QUOTA [*int* [K|M] | UNLIMITED] ON *nomTablespace*]
[QUOTA [*int* [K|M] | UNLIMITED] ON *nomTablespace2*]
- > Indique si l'utilisateur doit changer son mot de passe à la première connexion
[PASSWORD EXPIRE]
- > Verrouillage du compte
[ACCOUNT[LOCK|UNLOCK]]
- > Profil
[PROFILE *nomProfil*]



2. Création d'un nouvel utilisateur – c. Exemples

```
1 CREATE USER isitcom IDENTIFIED BY i$itc0m
```

```
2 CREATE USER isitcom IDENTIFIED BY i$itc0m  
  DEFAULT TABLESPACE tbs_users  
  PASSWORD EXPIRE  
  ACCOUNT LOCK
```

```
3 CREATE USER isitcom IDENTIFIED BY i$itc0m  
  DEFAULT TABLESPACE tbs_users  
  QUOTA 10M ON tbs_users  
  TEMPORARY TABLESPACE tmp_users  
  QUOTA 5M ON tbs_hr  
  PASSWORD EXPIRE
```

2. Création d'un nouvel utilisateur – d. Utilisateur identifié par le SE

- S'authentifie au niveau du SE mais pas au niveau d'Oracle (SQL>CONNECT /)
- Oracle vérifie uniquement si le nom de l'utilisateur courant du SE correspond à un utilisateur de la BD
- Nom utilisateur de la BD : OS_AUTHENT_PREFIX & nomUserSE
- Valeur de **OS_AUTHENT_PREFIX**
 - Par défaut : OPS\$
 - Peut être modifiée : ALTER SYSTEM SET OS_AUTHENT_PREFIX='';
- Création
 - CREATE USER OPS\$myUser IDENTIFIED EXTERNALLY
 - Sous Windows il est nécessaire de rajouter le nom de domaine, ou à défaut de la machine (le tout en majuscule) : CREATE USER "OPS\$MY_DOMAIN\MY_USER" IDENTIFIED EXTERNALLY



3. Modification (`ALTER USER`) et suppression (`DROP USER`)

Modification du mot de passe

```
ALTER USER isitcom IDENTIFIED BY c0mi$it [REPLACE i$itc0m]
```

Modification des quotas

```
ALTER USER isitcom QUOTA 15M ON tbs_users QUOTA 0M ON tbs_hr
```

Modification des tablespaces par défaut

```
ALTER USER isitcom DEFAULT TABLESPACE tbs2 TEMPORARY TABLESPACE tmp2;
```

Verrouillage

```
ALTER USER isitcom ACCOUNT LOCK
```

Déverrouillage

```
ALTER USER isitcom ACCOUNT UNLOCK
```

Suppression d'un utilisateur (et de son schéma s'il n'est pas vide)

```
DROP USER isitcom [CASCADE]
```

- Les vues `DBA_USERS` et `DBA_TS_QUOTAS` contiennent les informations sur les utilisateurs

Chapitre 5 - Gestion des utilisateurs

- Introduction
- I. Gestion basique des comptes des utilisateurs
- **II. Gestion des privilèges**
 - 1. Privilèges système vs. privilèges objet
 - 2. Privilèges système
 - 3. Privilèges objet
- III. Gestion des rôles
- IV. Gestion des profils
- V. Audit de l'activité des utilisateurs
- Quiz

1. Privilèges système vs. privilèges objet

■ Privilèges accordés à un utilisateur :

- Déterminent quelles actions l'utilisateur est autorisé à effectuer sur la base
- Accordés par l'instruction **GRANT** et retirés par l'instruction **REVOKE**
- Peuvent être groupés dans des **rôles**
- Deux types : privilèges au niveau du système et privilèges au niveau des objets

■ Privilège au niveau du système

- Droit d'exécuter une action sur n'importe quel objet d'un type particulier (près de 160).
- Ex. CREATE/DROP/ALTER TABLESPACE, USER, SESSION, TABLE, VIEW, etc.

■ Privilège au niveau objet

- Droit d'exécuter une action sur un objet spécifique
- Ex. SELECT, UPDATE, INSERT, etc.

2. Privilèges système - a. Exemples

Catégorie	Exemple
TABLE	CREATE [ANY] TABLE ALTER [ANY] TABLE DROP [ANY] TABLE SELECT [ANY] TABLE UPDATE [ANY] TABLE DELETE [ANY] TABLE
INDEX	CREATE [ANY] INDEX ALTER [ANY] INDEX DROP [ANY] INDEX
SESSION	CREATE SESSION ALTER SESSION RESTRICTED SESSION
TABLESPACE	CREATE TABLESPACE ALTER TABLESPACE DROP TABLESPACE

- Mot réservé **ANY** : droit d'exécuter un ordre dans n'importe quel schéma

2. Privilèges système - b. Privilèges d'administration

SYSOPER	STARTUP SHUTDOWN ALTER DATABASE OPEN MOUNT RECOVER DATABASE ALTER DATABASE BACKUP CONTROLFILE TO ALTER DATABASE ARCHIVELOG RESTRICTED SESSION
SYSDBA	SYSOPER WITH ADMIN OPTION CREATE DATABASE

- Vue `SYSTEM_PRIVILEGE_MAP` : liste de tous les privilèges système
- Vue `DBA_SYS_PRIVS` : privilèges système accordés à un utilisateur
- **N.B.** Si un utilisateur disposant du privilège SYSDBA se connecte à la BD avec `CONNECT..AS SYSDBA` il est connecté au schéma sys

2. Privilèges système - c. Attribution

■ Syntaxe :

```
GRANT [nomPrivilege | ALL PRIVILEGES] TO [nomUtilisateur | nomRole | PUBLIC]  
[WITH ADMIN OPTION]
```

- ALL PRIVILEGES : tous les privilèges système sont accordés à l'utilisateur
- PUBLIC : le privilège est attribué à tous les utilisateurs
- WITH ADMIN OPTION : donne à l'utilisateur le droit d'attribuer et de retirer le privilège reçu
- Exemples :
 - 1 GRANT CREATE TABLE, SELECT ANY TABLE, CREATE TABLESPACE TO
isitcom
 - 2 GRANT SYSDBA TO administrateur WITH ADMIN OPTION
 - 3 GRANT CREATE SESSION TO PUBLIC

2. Privilèges système - d. Révocation

- Syntaxe :

```
REVOKE [nomPrivilege | ALL PRIVILEGES | nomRole1] FROM  
[nomUtilisateur | nomRole2 | PUBLIC]
```

- Tout utilisateur disposant d'un privilège système avec l'option WITH ADMIN OPTION peut révoquer ce privilège de n'importe quel autre utilisateur

- Exemples de révocation :

- 1 REVOKE ALTER TABLESPACE FROM isitcom

- 2 REVOKE CREATE SESSION FROM PUBLIC

2. Privilèges système - d. Révocation

- Révocation des privilèges système : entraîne un effet de cascade pour les ordres DML (SELECT/DELETE/UPDATE/INSERT), mais pas pour les autres

- Exemple 1

- 1 *user*₁ dispose du privilège CREATE TABLE WITH ADMIN OPTION
- 2 *user*₁ transmet à *user*₂ le privilège CREATE TABLE
- 3 Le privilège CREATE TABLE est retiré à *user*₁
 - ⇒ *user*₁ ne peut plus créer des nouvelles tables
 - ⇒ *user*₂ garde le droit de créer des nouvelles tables

- Exemple 2

- 1 *user*₁ dispose du privilège SELECT TABLE WITH ADMIN OPTION
- 2 *user*₁ transmet à *user*₂ le privilège SELECT TABLE
- 3 Le privilège SELECT TABLE est retiré à *user*₁
 - ⇒ *user*₁ ne peut plus faire des sélections
 - ⇒ *user*₂ ?

2. Privilèges système - e. Accès aux données du dictionnaire

- Accès aux données du dictionnaire : protégé par le paramètre d'initialisation **07_DICTIONARY_ACCESSIBILITY**
- 07_DICTIONARY_ACCESSIBILITY=FALSE (valeur par défaut) :
 - Les utilisateurs disposant du privilège SELECT ANY TABLE n'ont pas accès au dictionnaire (schéma SYS)
 - Pour qu'un utilisateur ait accès au dictionnaire il doit avoir le privilège système SELECT ANY DICTIONARY
- Pour modifier la valeur de 07_DICTIONARY_ACCESSIBILITY (déconseillé)
`ALTER SYSTEM SET 07_DICTIONARY_ACCESSIBILITY=TRUE SCOPE=SPFILE`

3. Privilèges objet - a. Exemples

Action	TABLE[.nomCol]	VIEW[.nomCol]	SEQUENCE	PROCEDURE
ALTER	✓	✓	✓	✓
DELETE	✓	✓		
EXECUTE				✓
INDEX	✓	✓		
INSERT	✓	✓		
SELECT	✓	✓	✓	
UPDATE	✓	✓		

3. Privilèges objet - b. Attribution

Attribution

- Syntaxe (non exhaustive) :

```
GRANT [nomPrivilege | ALL PRIVILEGES] ON login.nomTable[(nomColonne)] TO  
[nomUtilisateur | nomRole | PUBLIC][WITH GRANT OPTION]
```

- WITH GRANT OPTION : donne à l'utilisateur le droit d'attribuer et de retirer le privilège reçu

- Exemples

1 GRANT ALL PRIVILEGES ON isitcom.etudiant TO scolarite

2 GRANT SELECT,UPDATE (nom,adresse) ON isitcom.etudiant TO guichet

3. Privilèges objet - c. Révocation

Révocation

- Syntaxe :

```
REVOKE [nomPrivilege | ALL PRIVILEGES | nomRole1] ON  
login.nomTable[(nomColonne)] FROM [nomUtilisateur | nomRole2 | PUBLIC]
```

- Exemple :

```
REVOKE UPDATE (nom,adresse) ON isitcom.etudiant FROM guichet
```

- Seul l'utilisateur ayant attribué un privilège objet peut le révoquer

- La révocation d'un privilège objet **entraîne un effet de cascade**. Exemple

- 1 *user₁* dispose du privilège SELECT on *Emp.sal* WITH GRANT OPTION

- 2 *user₁* transmet à *user₂* le privilège SELECT on *Emp.sal*

- 3 Le privilège SELECT on *Emp.sal* est retiré à *user₁*

⇒ *user₁* comme *user₂* n'ont plus accès à la colonne *Emp.sal*

Chapitre 5 - Gestion des utilisateurs

- Introduction
- I. Gestion basique des comptes des utilisateurs
- II. Gestion des privilèges
- **III. Gestion des rôles**
 - 1. Vue d'ensemble
 - 2. Opérations de base
 - 3. Activation/désactivation d'un rôle
- IV. Gestion des profils
- V. Audit de l'activité des utilisateurs
- Quiz

1. Vue d'ensemble

Rôle (*role*) :

- Ensemble nommé de privilèges
- Peut être affecté soit à un utilisateur soit à un autre rôle
- Intérêt : et non individuellement
- Oracle fournit un ensemble de rôles pré-définis (CONNECT, RESSOURCE, DBA, MGMT_USER, etc..)
- La vue [DBA_ROLES](#) fournit l'ensemble des rôles définis

2. Opérations de base

Création (non exhaustive)

```
CREATE ROLE nomRole [NOT IDENTIFIED| IDENTIFIED BY motDePasse]
```

Attribution et révocation de privilèges système à un rôle (même syntaxe que pour un utilisateur)

```
GRANT...[WITH ADMIN OPTION]
```

```
REVOKE...
```

Attribution et révocation de privilèges objet à un rôle (même syntaxe que pour un utilisateur, mais pas de clause [WITH GRANT OPTION])

```
GRANT...
```

```
REVOKE...
```

Suppression

```
DROP ROLE nomRole
```

Attribution d'un rôle à un utilisateur

```
GRANT nomRole TO nomUser [WITH ADMIN OPTION]
```

Révocation d'un rôle

```
REVOKE nomRole FROM nomUser
```

3. Activation/désactivation d'un rôle

■ Activation d'un rôle

- Si l'utilisateur bénéficiaire n'est pas connecté au moment de l'attribution : automatique à la première connexion
- Sinon, l'utilisateur bénéficiaire doit explicitement l'activer à l'aide de :
`SET ROLE nomRole1 [IDENTIFIED BY pwdRole], nomRole2...`

■ Désactivation d'un rôle

- Désactivation de tous les rôles : `SET ROLE NONE`
- Désactivation d'un ensemble de rôles : `SET ROLE ALL EXCEPT nomRole1, nomRole2, ...`
- Il n'est pas possible de désactiver individuellement un rôle

3. Activation/désactivation d'un rôle

- Paramètre d'initialisation **MAX_ENABLED_ROLES** : nombre maximal de rôles actifs simultanément pour un utilisateur (par défaut 30)
- Intérêt activation/désactivation : un utilisateur peut exécuter une application qui active un rôle en fournissant le mot de passe sans que l'utilisateur ne puisse utiliser le rôle directement
- La vue **SESSION_ROLES** fournit l'ensemble des rôles actifs pour une session
`SELECT * FROM session_roles`

Chapitre 5 - Gestion des utilisateurs

- Introduction
- I. Gestion basique des comptes des utilisateurs
- II. Gestion des privilèges
- III. Gestion des rôles
- **IV. Gestion des profils**
 - 1. Vue d'ensemble
 - 2. Exemple de limitations
 - 3. Opérations de base
 - 4. Fonction de vérification des mots de passe
- V. Audit de l'activité des utilisateurs
- Quiz

1. Vue d'ensemble

Profil (*profile*) :

- Ensemble nommé de **limitations** :
 - 1 limitations sur les mots de passe
Ex. nombre maximal de tentatives de connexion, etc.
 - 2 limitations sur les ressources systèmes
Ex. nombre maximal de sessions qu'un même utilisateur peut ouvrir, temps CPU maximal accordé à une session, etc.
- Intérêt : contrôle de l'utilisation des ressources et renforcement de la sécurité
- Activation du contrôle des limitations :
`ALTER SYSTEM SET resource_limit = true`
- Profil DEFAULT : profil par défaut affecté à un utilisateur

2. Exemple de limitations

Limitations sur les mots de passe (renforcement de la sécurité)

failed_login_attempts	Nombre maximal de tentatives d'authentification infructueuses avant que le compte ne soit verrouillé
password_lock_time	Durée en jours du verrouillage d'un compte suite à failed_login_attempts tentatives d'authentification infructueuses
password_life_time	Durée de validité en jours d'un mot de passe
password_grace_time	Durée en jour maximale laissée à un utilisateur pour changer son mot de passe arrivé à expiration

2. Exemple de limitations

Limitations sur l'utilisation des ressources

idle_time	Durée maximale en minutes d'inactivité d'une session. Au delà l'utilisateur ne peut que valider ou annuler ses transactions
connect_time	Durée maximale en minutes d'une session. Si dépassée les transactions non validées sont annulées et l'utilisateur est déconnecté
cpu_per_session	Temps CPU maximal en centaines de secondes consommé durant une session. Au delà l'utilisateur ne peut qu'annuler ou valider ses transactions
sessions_per_user	Nombre maximale de sessions simultanées qu'un utilisateur peut ouvrir

3. Opérations de base - a. Syntaxe

Création (non exhaustive)

```
CREATE PROFILE nomProfil LIMIT  
    nomLimitation1 [valeur | UNLIMITED | DEFAULT]  
    [nomLimitation2 [valeur | UNLIMITED | DEFAULT]]  
    [...]
```

Attribution d'un profil à utilisateur

```
ALTER USER nomUser PROFILE nomProfil
```

Modification d'un profil

```
ALTER nomProfil LIMIT  
    nomLimitation [nouvelleValeur | UNLIMITED | DEFAULT]
```

Suppression

```
DROP PROFILE nomProfil [CASCADE]
```

Informations sur les profils

vue `DBA_PROFILES`

3. Opérations de base - b. Exemples

Création

```
CREATE PROFILE pAgent LIMIT  
  sessions_per_user 2  
  idle_time 600  
  failed_login_attempts 3  
  password_lock_time 15/1440  
  cpu_per_session 500
```

Attribution d'un profil à utilisateur

```
ALTER USER agent PROFILE pAgent
```

Modification d'un profil

```
ALTER DEFAULT LIMIT sessions_per_user 5  
ALTER pAgent LIMIT failed_login_attempts UNLIMITED
```

Suppression

```
DROP PROFILE pAgent [CASCADE]
```

4. Fonction de vérification des mots de passe

- SYS : a la possibilité de créer une fonction de vérification de mots de passe et de l'affecter à un profil
- Signature de la fonction :
`nomFonction(login IN VARCHAR2(30), pwd IN VARCHAR2(30), ancienPwd IN VARCHAR2(30)) RETURN BOOLEAN`
- Affectation à un profil :
`CREATE PROFILE pAgent LIMIT PASSWORD_VERIFY_FUNCTION nomFonction`
- Désactivation :
`ALTER PROFILE pAgent LIMIT PASSWORD_VERIFY_FUNCTION DEFAULT`

Chapitre 5 - Gestion des utilisateurs

- Introduction
- I. Gestion basique des comptes des utilisateurs
- II. Gestion des privilèges
- III. Gestion des rôles
- IV. Gestion des profils
- **V. Audit de l'activité des utilisateurs**
 - 1. Vue d'ensemble
 - 2. Activation/désactivation de l'audit
 - 3. Activation/désactivation de l'audit
 - 4. Catégories d'audit
 - 5. Gestion des données d'audit
- Quiz

1.a. Qu'est ce que l'audit?

Fonction d'audit

- Surveiller les actions faites par les utilisateurs sur la BD
- Détecter les activités suspectes ou anormales faites
 - sur un objet donné, ou
 - par un utilisateur donné, ou
 - par une action (ordre SQL) donnée, ou
 - avec un privilège donné

Ex. `AUDIT INSERT TABLE BY isitcom WHENEVER NOT SUCCESSFUL`

1.b. Informations collectées (trace) lors d'un audit

Informations collectées (trace) lors d'un audit

- Dépendent du type d'audit
- Essentiellement :
 - 1 Login SE de l'utilisateur ayant ouvert la session
 - 2 Login Oracle de l'utilisateur ayant ouvert la session
 - 3 Identifiant de la session
 - 4 Identifiant du terminal
 - 5 Nom de l'objet utilisé (Ex. table *Emp*)
 - 6 Code de l'opération effectuée ou essayée. Le vue **AUDIT_ACTION** liste les actions et leurs codes (1 CREATE TABLE, 2 INSERT, etc.)
 - 7 Code de terminaison de l'opération (Ex.0 si tout se passe bien).
 - 8 Estampille temporelle

2. Activation/désactivation de l'audit

- Activation/désactivation de la fonction d'audit : paramètre d'initialisation statique **AUDIT_TRAIL**
- NONE : valeur par défaut. Désactive l'audit.
- ALTER SYSTEM SET audit_trail=**DB** SCOPE=spfile
 - Trace d'audit stockée dans la BD
 - Essentiellement dans la table **sys.AUD\$**
 - Accessible via la vue **DBA_AUDIT_TRAIL**
 - Exemple : lister les actions faites par l'utilisateur isitcom

```
SELECT username, timestamp, action_name FROM dba_audit_trail
WHERE username = 'ISITCOM'
```

3. Activation/désactivation de l'audit

- `ALTER SYSTEM SET audit_trail=OS SCOPE=spfile`
 - Trace d'audit stockée dans le SE
 - Windows : Observateur d'événements
 - Unix : fichier dont l'emplacement est défini par le paramètre d'initialisation `audit_file_dest`
- `ALTER SYSTEM SET audit_trail=xml SCOPE=spfile`
 - Trace d'audit stockée dans un document XML
 - Emplacement défini par le paramètre d'initialisation `audit_file_dest`

4. Catégories d'audit - a. Types d'audit

Types d'audit

- 1 Audit par défaut
- 2 Audit des ordres SQL
- 3 Audit des privilèges
- 4 Audit des objets

4. Catégories d'audit - b. Audit par défaut

- Toujours activé, même si `AUDIT_TRAIL=NONE`
- Stockage de la trace dans le SE, même si `AUDIT_TRAIL=DB`
- Informations tracées :
 - Démarrage de l'instance
 - Arrêt de l'instance
 - Nom de l'utilisateur du système d'exploitation qui s'est connecté à Oracle avec des privilèges système

4. Catégories d'audit - c. Audit des ordres SQL

■ Syntaxe :

```
AUDIT action [BY nomUser] [WHENEVER [NOT] SUCCESSFUL] [BY  
ACCESS|SESSION]
```

- *action* : action à tracer. Ex. CREATE TABLE, DROP TABLE, TABLESPACE, ROLE, etc.
- BY *nomUser* : utilisateur dont l'action est à tracer
- WHENEVER NOT SUCCESSFUL : trace générée uniquement si l'action échoue
- BY SESSION | BY ACCESS : informations plus détaillées si BY ACCESS

■ Désactivation :

```
NOAUDIT action [BY nomUser]
```

- Liste des actions tracées : `DBA_STMT_AUDIT_OPTS`

4. Catégories d'audit - d. Audit des privilèges

■ Syntaxe :

`AUDIT privilege [BY nomUser] [BY ACCESS|SESSION]`

- *privilege* : privilège à tracer. Ex. : SELECT ANY TABLE, ALTER TABLESPACE, ALTER USER, ect.

- Exemple : `AUDIT CREATE ANY TABLE BY isitcom`

■ Désactivation :

`NOAUDIT privilege [BY nomUser]`

- Liste des privilèges tracés : `DBA_PRIV_AUDIT_OPTS`

4. Catégories d'audit - e. Audit des objets

- Audit d'une action particulière faite sur un objet particulier

- **Syntaxe :**

```
AUDIT action ON objet [BY nomUser] [WHENEVER [NOT] SUCCESSFUL] [BY  
ACCESS|SESSION]
```

- *action* : UPDATE, INSERT, DELETE, UPDATE, etc.

- *objet* : table, vue, séquence, etc.

- Exemple : AUDIT SELECT, INSERT, DELETE ON isitcom.Etudiants BY
scolarite

- **Désactivation :**

```
NOAUDIT action ON objet [BY nomUser]
```

- Liste des objets tracés : DBA_OBJ_AUDIT_OPTS



5. Gestion des données d'audit

- Si la trace d'audit est saturée, les instructions tracées ne s'exécutent plus correctement
- Le DBA doit purger par lui-même la trace (pas d'utilitaires Oracle)
- Exemple :

```
DELETE FROM sys.aud$ WHERE timestamp# < SYSDATE - 90
```

Chapitre 5 - Gestion des utilisateurs

- Introduction
- I. Gestion basique des comptes des utilisateurs
- II. Gestion des privilèges
- III. Gestion des rôles
- IV. Gestion des profils
- V. Audit de l'activité des utilisateurs
- Quiz

Quiz

- 1 Quelle est la différence entre un privilège Objet et un privilège système
- 2 Que signifie l'option WITH ADMIN OPTION lors de l'attribution d'un privilège ?
- 3 Dans quels cas la révocation d'un privilège entraîne un effet de cascade?
- 4 Comment un utilisateur peut-il activer un rôle qui lui a été attribué ?
- 5 Quel est l'objectif d'un profil et quels types de limitations peut-il imposer ?
- 6 Quelles informations sont collectées dans les traces d'audit lors d'une tentative d'accès infructueuse ?

Chapitre 6 - Création manuelle d'une BD et configuration du réseau

■ Introduction

- I. Création d'une base de données
- III. Configuration du réseau (Oracle Net)

Introduction

Mise en place d'un BD Oracle

- 1 Installation d'Oracle côté serveur et côté client
- 2 Création de la BD
- 3 Configuration du réseau côté serveur et côté client

Oracle : fournit un ensemble d'outils graphiques pour assister l'administrateur

Ce chapitre...

- 1 Création d'une base de données
- 2 Configuration du réseau (Oracle Net)

Chapitre 6 - Création manuelle d'une BD et configuration du réseau

- Introduction
- I. Création d'une base de données
 - 1. Notes sur l'architecture OFA
 - 2. Création manuelle
- III. Configuration du réseau (Oracle Net)

1. Notes sur l'architecture OFA - a. Présentation

Optimal File Architecture (OFA)

- Ensemble de recommandations sur l'arborescence et le nommage des fichiers du serveur destinées à faciliter l'administration des produits Oracle
- Principe : séparer le produit Oracle, les fichiers relatifs à l'administration et les fichiers de la BD
- *Oracle Universal Installer* : respecte les recommandations OFA
- Deux répertoires importants : **Oracle_Base** et **Oracle_Home**



1. Notes sur l'architecture OFA - b. Oracle_Base

Oracle_Base

- Racine de l'arborescence
- Conventions d'emplacement et de nommage
 - Windows : `X:\app\compte`, où *X* est un lecteur de disque et *compte* le compte utilisé pour l'installation
Ex. `d:\app\foulen`
 - Unix et Linux : `/pm/ccc/compte`, où *pm* est un point de montage, *ccc* une chaîne quelconque et *compte* le compte utilisé pour l'installation
Ex. `/u01/app/foulen`

1. Notes sur l'architecture OFA - b. Oracle_Base

Oracle_Base

- Contient, entre autres
 - `oradata` : fichiers des différentes BDs (fichiers de données, de contrôle, temporaires, d'annulation, de journalisation et d'audit)
 - `admin` : fichiers d'administration des BDs (Ex. `pfile`)
 - `cfgtoollogs` : fichiers journaux des assistants de configuration
 - `diag` : pour le Référentiel de Diagnostic Automatique (*ADR*)
 - `product` : contient un ou plusieurs répertoire *Oracle_Home*, un par version d'un produit Oracle installée
- `oradata` et `admin` :
 - Divisés chacun en des sous-répertoires, un par BD
 - Chaque sous-répertoire porte le nom de la BD correspondante (paramètre `DB_NAME`)



1. Notes sur l'architecture OFA - c.

Oracle_Home

Oracle_Home

- Sous-répertoire d'Oracle_Base\product
- Contient les fichiers du logiciel installé (Ex. serveur de BD : utilitaires, exécutables, scripts, etc.)
- Conventions d'emplacement et de nommage
 - ORACLE_BASE\product\v.v.v\type_n
 - v.v.v : version du produit
 - type : type du produit. Ex. dbhome pour serveur de BD
 - _n : un numéro d'ordre du type

Ex. d:\app\foulen\product\11.2.0\dbhome1



2. Création manuelle - a. Étapes

- 1 Choisir un nom (SID) pour identifier l'instance et la distinguer des autres instances tournant sur la machine

Windows : `set oracle_sid=dbisitcom`

Unix : `export oracle_sid=dbisitcom`

- 2 Avec l'utilitaire ORAPWD, créer un fichier de mots de passe

`ORACLE_HOME\database\PWD`*SID*.ora

`ORAPWD FILE=D:\app\Khaled\product\11.2.0\dbhome_1\database\PWDdbisitcom.ora`

- 3 Créer un fichier de paramètres PFILE `ORACLE_HOME\database\init`*SID*.ora : à partir de l'exemple Oracle disponible soit dans la documentation soit sous le répertoire

`ORACLE_HOME\dbs\init.ora`

`D:\app\Khaled\product\11.2.0\dbhome_1\database\initdbisitcom.ora`

- 4 Modifier le nouveau pfile : essentiellement le chemin du fichier de contrôle, le paramètre `db_name` et la chaîne `<ORACLE_BASE>` (avec un éditeur de texte)

`db_name=dbisitcom`

`control_files = 'D:\app\khaled\oradata\dbisitcom\control01.CTL'`

`<ORACLE_BASE>` est remplacé par `D:\app\khaled`

2. Création manuelle - a. Étapes

- 5 Créer un répertoire sous oradata et un autre sous admin ayant le nom de votre BD (pour respecter l'architecture OFA) :

```
D:\app\Khaled\oradata\DBISITCOM
```

```
D:\app\Khaled\admin\DBISITCOM
```

- 6 (Uniqu. sous Windows) Créer une instance sous forme d'un service Windows avec l'utilitaire ORADIM¹

```
ORADIM -NEW -SID dbisitcom -STARTMODE MANUAL -PFILE
```

```
D:\app\Khaled\product\11.2.0\dbhome_1\database\initdbisitcom.ora
```

- 7 Se connecter à l'instance (déterminé par le SE selon la valeur de oracle_sid)

```
SQLPLUS /nolog puis SQL> CONNECT / AS SYSDBA
```

- 8 Créer un fichier de paramètres serveur SPFILE

```
SQL> CREATE SPFILE FROM PFILE
```

- 9 Démarrer l'instance (aucune vérification de cohérence à ce niveau)

```
SQL> STARTUP NOMOUNT
```

- 10 Créer la base de données

```
SQL> CREATE DATABASE...
```

¹N.B. Pour supprimer le service : `ORADIM -DELETE -SID dbisitcom`

2. Création manuelle - a. Étapes

12 Créer des tablespaces additionnels (cf. chapitre ??)

13 Créer les vues du dictionnaire de données et les synonymes avec le script CATALOG.SQL :

```
SQL> @ ORACLE_HOME\rdbms\admin\CATALOG.SQL
```

14 Créer les packages PL/SQL avec le script CATPROC.SQL :

```
SQL> @ ORACLE_HOME\rdbms\admin\CATPROC.SQL
```

15 (Windows uniq.) Mettre l'instance en mode démarrage automatique

```
ORADIM -EDIT -SID DBISITCOM -STARTMODE AUTO -SRVSTART SYSTEM
```

16 (Unix uniq.) Mettre l'instance en mode démarrage automatique en l'enregistrant dans le fichier oratab (solaris sous /etc, linux sous /var/opt/oracle)²

```
vi /var/opt/oracle/oratab
```

```
dbisitcom:/u01/app/oracle/product/11.2.0/dbhome_1:Y
```

17 Configurer le DB control (script ORACLE_HOME\bin\emca)

```
emca -config dbcontrol db -repos create
```

²pour plus d'informations voir

2. Création manuelle - b. Exemple CREATE DATABASE

```
CREATE DATABASE DBISITCOM
```

-> Choix des mots de passe de SYS et SYSTEM

```
USER SYS IDENTIFIED BY i$1tc0m
```

```
USER SYSTEM IDENTIFIED BY oradmin
```

-> 2 groupes de journalisation de taille 100M chacun

```
LOGFILE
```

```
GROUP 1 'd:\app\khaled\oradata\DBISITCOM\redo11.log' SIZE 100M REUSE,
```

```
GROUP 2 'd:\app\khaled\oradata\DBISITCOM\redo21.log' SIZE 100M REUSE
```

-> Nombre max de groupes Redo log

```
MAXLOGFILES 5
```

-> Nombre de fichiers membre par groupe (dépend du système d'exploitation)

```
MAXLOGMEMBERS 3
```

-> Nombre max de fichiers de données

```
MAXDATAFILES 100
```

-> Nombre max d'instances pouvant être associées à la BD

```
MAXINSTANCES 1
```

2. Création manuelle - b. Exemple CREATE DATABASE

-> Encodage des caractères

```
CHARACTER SET US7ASCII
```

```
NATIONAL CHARACTER SET AL16UTF16
```

-> Fichier stockant le tablespace SYSTEM

```
DATAFILE 'd:\app\khaled\oradata\DBISITCOM\system01.dbf' SIZE 500M REUSE
```

```
AUTOEXTEND ON MAXSIZE UNLIMITED
```

-> Fichier stockant le tablespace SYSAUX

```
SYSAUX DATAFILE 'd:\app\khaled\oradata\DBISITCOM\sysaux01.dbf' SIZE 500M REUSE
```

```
AUTOEXTEND ON MAXSIZE UNLIMITED
```

-> Tablespace temporaire

```
DEFAULT TEMPORARY TABLESPACE tempts1 TEMPFILE
```

```
'd:\app\khaled\oradata\DBISITCOM\temp01.dbf' SIZE 20M
```

-> Tablespace d'annulation

```
UNDO TABLESPACE undotbs1 DATAFILE 'd:\app\khaled\oradata\DBISITCOM\undotbs01.dbf'  
SIZE 200M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;
```

Chapitre 6 - Création manuelle d'une BD et configuration du réseau

- Introduction
- I. Création d'une base de données
- III. Configuration du réseau (Oracle Net)
 - 1. Présentation d'Oracle Net
 - 2. Identification d'un serveur de BD
 - 3. Configuration côté client
 - 4. Configuration côté serveur
 - 5. L'utilitaire LiSteNeR ConTroL lsnrctl

1. Présentation d'Oracle Net

- Rôle : permettre les accès distants d'applications clientes à un serveur de base de données Oracle
- Installation et configuration : côté client et côté serveur
- Objectif : rendre le réseau "transparent" pour les applications
⇒ sans spécification de l'adresse de la machine hôte du serveur de BD, du protocole réseau, etc.

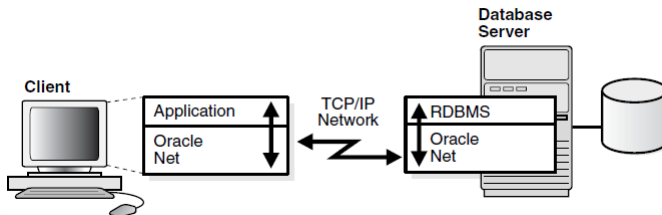


Figure: [Oracle© Net Services Administrator's Guide 11g Release 2 (11.2) E10836-06, 2010]

1. Présentation d'Oracle Net

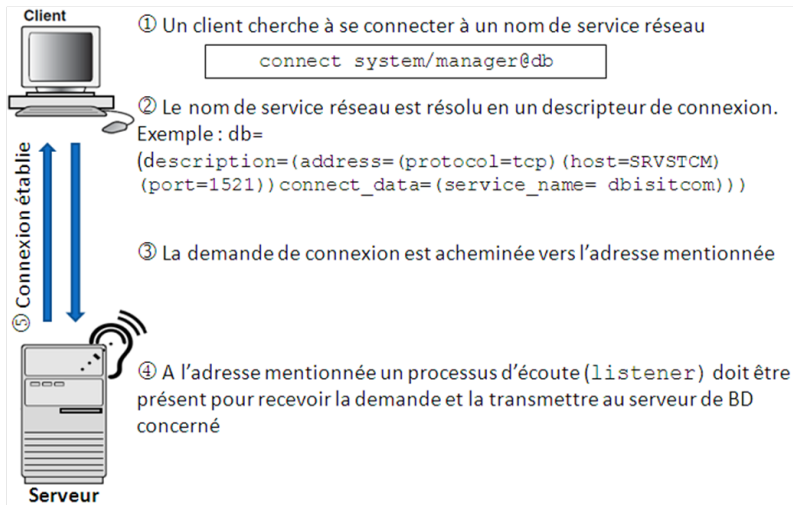


Figure: Vue d'ensemble

2. Identification d'un serveur de BD

- Serveur de BD (instance + BD) : offre le service consistant à connecter et à exécuter les requêtes des applications clientes
- Machine serveur (hôte du serveur de BD) :
 - Peut être l'hôte de plusieurs serveurs de BD
 - Chaque serveur de BD est identifié par un **nom de service** (*Service name*)
 - Au moins un **processus d'écoute** (*Listener*) doit y être présent pour orienter les requêtes vers le nom de service (serveur de BD) concerné
- **Nom de service** :
 - Valeur stockée dans le paramètre **SERVICE_NAMES** du fichier de paramètres
 - Par défaut correspond au nom global de la BD renseigné lors de sa création :
DB_NAME.DB_DOMAIN
Ex. dbisitcom (si pas de domaine), dbisitcom.infcom.rnu.tn



3. Configuration côté client - a. Descripteur de connexion

Accès à un serveur de BD à partir d'un client :

- 1 Protocole réseau (Ex. TCP/IP)
- 2 Adresse de la machine hôte. Ex. adresse IP : 10.1.1.5 ou nom réseau de la machine : SRVCSTCM (utilisation d'un DNS)
- 3 Port d'écoute (si protocole TCP/IP) : par défaut 1521 pour Oracle
- 4 Nom de service du serveur de BD

⇒ Informations représentées sous la forme d'un **descripteur de connexion** (*connect descriptor*)



3. Configuration côté client - a. Descripteur de connexion

Descripteur de connexion

- Exemple :

```
db =(description=(address=(protocol=tcp)(host=SRVCSTCM)  
(port=1521))connect_data=(service_name=dbisitcom))
```

- db :

- **nom de service réseau** (*Net Service Name*)

- Utilisé pour identifier le descripteur de connexion

- Connexion à un serveur de BD depuis un client : par le nom du service réseau.

Ex. connect system/manager@db.

(@: indique le nom du service réseau auquel l'utilisateur demande à se connecter)

- Noms de service réseau auxquels un client peut accéder : groupés dans le fichier **tnsnames.ora**



3. Configuration côté client - b. Résolution des noms de connexions réseau

- `tnsnames.ora` (*tns : transparent network substrate*)
 - Rôle : **résolution des noms de service réseau** en des descripteurs de connexion
 - Doit être présent sur toute machine désirant se connecter au(x) serveur(s) de BD
- Autres méthodes de résolution
 - 1 Méthode de connexion simplifiée :
 - Pour TCP/IP, depuis la version Oracle 10g
 - Adresse de connexion (sans configuration du client) :
[//]hôte[:port]/[service]
 - Ex. `connect system/manager@SRVCTCM:1521/dbisitcom`
 - 2 LDAP : via un annuaire électronique (LDAP) centralisé.
 - Nécessite un produit tier (Ex. Oracle Net Directory, Windows Active Directory, OpenLDAP, etc.)

3. Configuration côté client - b. Résolution des noms de connexions réseau

- `ORACLE_HOME\network\admin\sqlnet.ora` : stocke les méthodes de résolution des noms de service utilisables par le client
- La commande `tnsping nomServiceReseau` exécutée au niveau du client permet de tester si un processus d'écoute peut être contacté à l'adresse correspondante

4. Configuration côté (machine) serveur - a. Processus d'écoute (*Listener*)

Processus d'écoute :

- Rôle : orienter les demandes de connexion des applications clientes vers le serveur de BD approprié
- Se matérialise par
 - un service sous Windows (Oracle<nomOracle>_Home>TNSListener)
 - un processus sous Unix (tnslsnr)
- Gestion (démarrage/arrêt/modification) : par l'utilitaire `lsnrctl`
- Configuration (fichier `ORACLE_HOME/network/admin/listener.ora`) :
 - Comment le processus écoute (ex. selon quel protocole et pour quelles adresses)
 - Pour quels serveurs de BD
- Création : il suffit de créer une entrée dans le fichier `listener.ora`



4. Configuration côté (machine) serveur - b.

Fichier listener.ora

listener.ora (essentiellement) :

- 1 Emplacements d'écoute (ADDRESS_LIST) :** Adresses réseaux utilisées par le processus d'écoute pour recevoir les demandes de connexion
- 2 Services de BD (SID_LIST_LISTENER) :** Serveurs de BD inscrits auprès du processus d'écoute (pour lesquelles le processus écoute et accepte des demandes de connexions)

4. Configuration côté (machine) serveur - b.

Fichier listener.ora

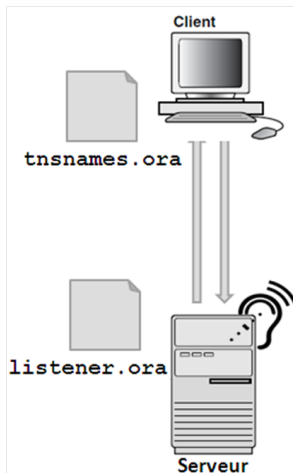
Exemple listener.ora :

```
LISTENER=
  (DESCRIPTION=
    (ADDRESS_LIST=
      (address=(protocol=tcp)(host=SRVCSTCM)(port=1521))
    )
  )
SID_LIST_LISTENER =
  (SID_DESC =
    (GLOBAL_DBNAME = dbisitcom)
    (ORACLE_HOME = d:\app\khaled\product\11.2.0\dbhome_1) 3
    (SID_NAME = dbisitcom)
  )
)
```

³Entrée requise uniquement si l'instance et le processus d'écoute n'appartiennent pas au même ORACLE_HOME



En résumé...



5. L'utilitaire LiSteNeR ConTroLsnrtctl

lsnrctl :

- S'utilise côté serveur pour gérer le(s) processus d'écoute
- Principales commandes

Commande : lsnrctl>	Effet
HELP	Affiche la liste des commandes
HELP <i>commande</i>	Affiche l'aide sur une commande
START	Démarre le processus d'écoute
STOP	Arrête le processus d'écoute
STATUS	Affiche les informations sur la configuration du processus d'écoute et les services enregistrés
SERVICES	Affiche des informations détaillées sur les services enregistrés auprès du processus d'écoute
RELOAD	Recharge la configuration du processus d'écoute à partir de <code>listener.ora</code>

Chapitre 7 - Sauvegarde et récupération

■ Introduction

- I. Concepts de base
- II. Sauvegarde avec RMAN
- III. Restauration et récupération avec RMAN
- Quiz
- Exercice

Introduction

Sécurité des données : une des tâches principales de l'administrateur

- 1 Protection des **fichiers sensibles** (fichier de contrôle, fichier de paramètres, fichiers de journalisation, etc.)
- 2 Mise en place d'une **stratégie de sauvegarde et de récupération**
 - Adaptée aux contraintes de l'entreprise
 - Testée et documentée

Introduction

- Deux grandes catégories de sauvegardes : **logiques** et **physiques**
- **Sauvegarde logique** : copie de la définition de tous les objets (fichiers trace) + export/import des données
- **Sauvegarde physique** :
 - Copie physique (d'une partie) des fichiers de la BD
 - À l'aide des commandes du SE et de scripts SQL (déconseillé) ou à l'aide de l'utilitaire **RMAN** (*Recovery MANager*)

Ce chapitre...

- 1 Concepts de base
- 2 Sauvegarde avec RMAN
- 3 Restauration et récupération avec RMAN



Chapitre 7 - Sauvegarde et récupération

■ Introduction

■ I. Concepts de base

- 1. Principe de la sauvegarde/récupération
- 2. Sauvegarde cohérente vs. sauvegarde incohérente
- 3. Quel mode de fonctionnement pour la BD?

■ II. Sauvegarde avec RMAN

■ III. Restauration et récupération avec RMAN

■ Quiz

■ Exercice

1. Principe de la sauvegarde/récupération

- Archivage des fichiers Redo Log (mode ARCHIVELOG) : permet un recouvrement complet de la BD suite à une panne
- Principe : restauration, puis récupération

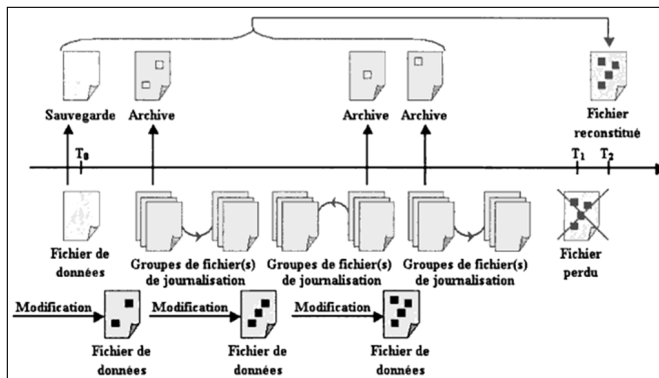
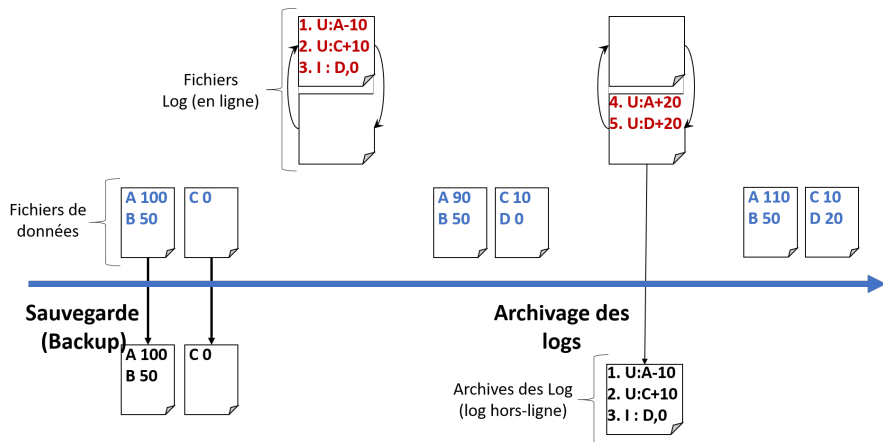


Figure: [Oracle 11g Administration, Olivier Heurtel, eni Editions, Nov. 2008]

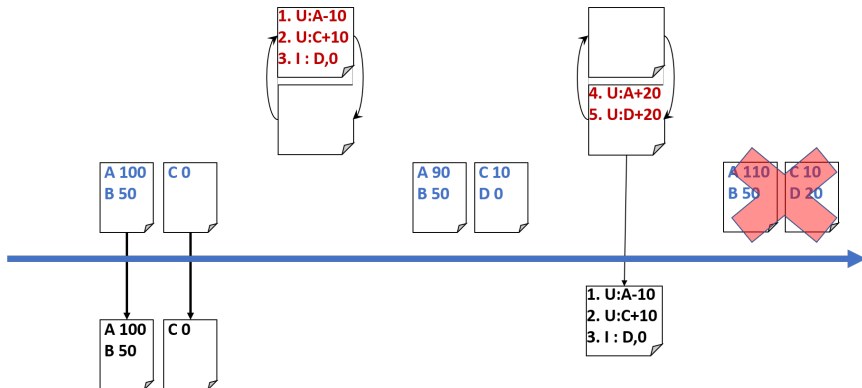
1. Principe de la sauvegarde/récupération

1. Sauvegarde



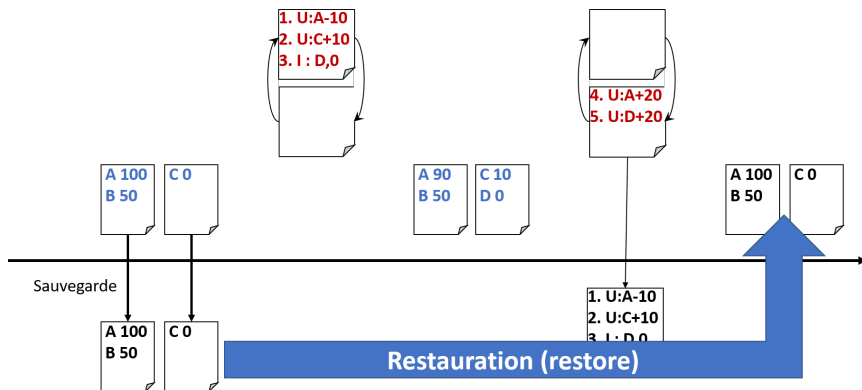
1. Principe de la sauvegarde/récupération

2. Fichiers de données endommagés



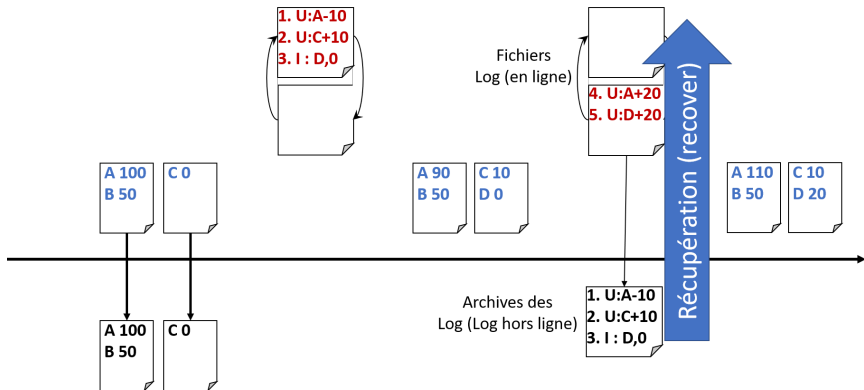
1. Principe de la sauvegarde/récupération

3. Restauration



1. Principe de la sauvegarde/récupération

4. Récupération (si ARCHIVELOG)



2. Sauvegarde cohérente vs. sauvegarde incohérente

Sauvegarde cohérente (*cold-backup*)

- Sauvegarde de la **totalité de la BD** après un **arrêt propre** : sauvegarde base fermée ou *cold backup*
 - **Cohérente** : fichiers de données et fichier de contrôle portent le même SCN (**synchrones**)
- ⇒ Utilisable sans application des fichiers de journalisation
- ⇒ Seul type de sauvegarde possible si mode NOARCHIVELOG
- ⇒ Pas possible pour les BD devant fonctionner 24/24, 7/7

2. Sauvegarde cohérente vs. sauvegarde incohérente

Sauvegarde incohérente (*hot backup*)

- Sauvegarde sans fermeture de la BD (sauvegarde BD ouverte)
 - ⇒ Les fichiers sauvegardés ne sont pas synchrones (pas sauvegardés au même instant) et peuvent subir des modifications post-sauvegarde
- BD restaurée à partir d'une copie incohérente ⇒ nécessite l'utilisation des Redo Log pour rejouer les modifications et remettre la BD à un état cohérent
 - ⇒ Possible uniquement si mode ARCHIVELOG
 - ⇒ Souhaitable car disponibilité accrue de la BD

3. Quel mode de fonctionnement pour la BD?

		Pertes de données acceptables	
		Oui	Non
Sauvegarde BD fermée possible	Oui	NOARCHIVELOG	ARCHIVELOG
	Non	ARCHIVELOG	ARCHIVELOG

- ARCHIVELOG : au moins une des contraintes ci-dessous existe
 - Si aucune perte de données n'est tolérée
 - Si la BD doit rester disponible à tout moment
- Notes sur le mode ARCHIVELOG
 - Mettre la BD en mode ARCHIVELOG : la BD doit être en mode MOUNT + ALTER DATABASE ARCHIVELOG
 - LOG_ARCHIVE_DEST_*n* : emplacement de la *n^{eme}* copie de l'archive ($n \leq 10$)
 - Vue V\$ARCHIVED_LOG : informations sur les archives log

Chapitre 7 - Sauvegarde et récupération

- Introduction
- I. Concepts de base
- II. Sauvegarde avec RMAN
 - 1. Vue d'ensemble
 - 2. Commande de sauvegarde
 - 3. Format d'une sauvegarde : image vs. backup set
 - 4. Sauvegarde complète
 - 5. Sauvegarde partielle
 - 6. Éléments de configuration de RMAN
 - 7. Zone de récupération rapide (Flash Recovery Area)
- III. Restauration et récupération avec RMAN
- Quiz
- Exercice

1. Vue d'ensemble

RMAN (*Recovery MANager*)

- Utilitaire en ligne de commande permettant la sauvegarde et la récupération d'une **BD cible** (*Target DataBase*)
- Exemples de lancement
 - `> rman target sys/i$itc0m`
 - `> rman target sys/i$itc0m@db`
 - `> rman`
RMAN> connect target /@localhost:1521/XEPDB1
 - Connection au root container (≥ 12) `rman target=`
- A besoin que la BD cible soit ouverte ou à défaut **montée** (pour accéder au fichier de contrôle)
- **Ne permet pas** la sauvegarde des fichiers de configuration du réseau et celui des mots de passe
- S'utilise également avec Enterprise Manager



2. Commande de sauvegarde - a. Syntaxe (non exhaustive)

Syntaxe générale : `BACKUP` [*comment*] *quoi* [*options*]

■ *comment*

- Format de la sauvegarde : **image** (AS COPY) ou **backup set** (AS [COMPRESSED] BACKUP SET)
- Portée de la sauvegarde : **tous les blocs** (par défaut), ou **incrémentale** (INCREMENTAL LEVEL *n* [CUMULATIVE])

■ *quoi*

- Type de la sauvegarde : **Sauvegarde complète** (DATABASE), **Sauvegarde partielle** (TABLESPACE *nomTablespace*)
- SPFILE

■ *options*

- INCLUDING CURRENT CONTROLFILE
- PLUS ARCHIVELOG
- NOT BACKEDUP SINCE TIME = *date*
- TAG = '*nomSauvegarde*'



3. Format d'une sauvegarde : image vs. backup set

- RMAN permet deux alternatives pour la copie physique des données : **image** et **backup set**
- **Image**
 - Même principe que la copie d'un fichier dans les SE
 - Duplication (copie identique) d'un **seul** fichier (de données, d'archives log, de contrôle ou de paramètres d'initialisation)
 - Inclut tous les blocks du fichier dupliqué, même ceux qui n'ont jamais été utilisés (blocs vides)
 - Ne peut être écrite que dans un disque (pas directement sur la bande de sauvegarde)

3. Format d'une sauvegarde : image vs. backup set

- **Backup set** (Jeu de sauvegarde)
 - Copie d'un ou de **plusieurs fichiers** dans un format lisible uniquement par RMAN
 - **N'inclut pas les blocs n'ayant jamais été utilisés**
 - **N'inclut pas les données d'annulation non nécessaires à la récupération** (transactions validées)
 - Stocké dans un (par défaut) ou plusieurs fichiers binaires appelés **Backup piece** (éléments de sauvegarde)
 - Possibilité de lecture multiplexée (simultanée) lors d'une sauvegarde

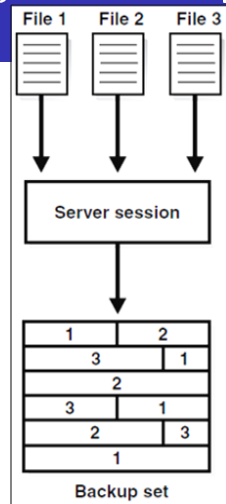


Figure: [Oracle® Database Backup and Recovery User's Guide 11g Release 2 (11.2)]

4. Sauvegarde complète - a. Présentation

Sauvegarde **complète**

- Sauvegarde de la **totalité** de la BD
- Exemples :
 - `BACKUP DATABASE;`
 - `BACKUP AS COMPRESSED BACKUPSET DATABASE;` : jeu de sauvegarde compressé (sauvegarde plus lente mais gain typique de 80% de l'espace requis)
 - `BACKUP VALIDATE DATABASE` : permet de vérifier si la BD est en "bon état" pour la sauvegarde (tous les fichiers sont accessibles)
- Peut se faire
 - BD fermée (*cold backup*) : sauvegarde cohérente, si `NOARCHIVELOG`
 - BD ouverte (*hot backup*) : sauvegarde incohérente, si `ARCHIVELOG`

4. Sauvegarde complète - b. Scénarios

■ Sauvegarde BD fermée (cohérente):

```
SHUTDOWN IMMEDIATE;           # arrêter la base
STARTUP MOUNT;                  # monter la base
BACKUP AS COMPRESSED BACKUPSET DATABASE # sauvegarder la base
SQL "ALTER DATABASE OPEN"      # ouvrir la base
```

■ Sauvegarde BD ouverte (incohérente)

```
BACKUP DATABASE PLUS ARCHIVELOG # sauvegarder la base
                                # et les archives
```

■ Sauvegarde d'une pluggable database ($\geq 12c$)

```
rman target=/  
RMAN> BACKUP PLUGGABLE DATABASE pdb1, pdb2;
```

5. Sauvegarde partielle - a. Présentation

Sauvegarde **partielle**

- Sauvegarde d'une **partie** (Ex. tablespaces) de la BD
 - Exemple : `BACKUP AS COMPRESSED BACKUP SET TABLESPACE system, tbsusers;`
 - Les fichiers sauvegardés ne sont pas synchrones et peuvent subir des modifications post-sauvegarde
 - Forcément incohérente
- ⇒ Exploitable uniquement si mode ARCHIVELOG

N.B. La sauvegarde du tablespace SYSTEM, entraîne également la sauvegarde du fichier de contrôle et de celui des paramètres

5. Sauvegarde partielle - b. Scénario

■ Sauvegarde partielle BD ouverte :

- 1 Jour 1 : `BACKUP TABLESPACE system, tbsusers`
- 2 Jour 2 : `BACKUP TABLESPACE index`
- 3 Jour 3 : `BACKUP DATABASE NOT BACKED UP SINCE TIME='SYSDATE-3' PLUS ARCHIVELOG INCLUDING CONTROLFILE`

■ Sauvegarde partielle BD fermée : possible, mais aucun intérêt

⇒ sauvegarde partielle exploitable uniquement si ARCHIVELOG, or si ARCHIVELOG autant la faire BD ouverte

En résumé...

- Sauvegarde BD ouverte (*hot backup*), complète ou partielle, si ARCHIVELOG
- Sauvegarde complète BD fermée (*Cold Backup*), si NOARCHIVELOG

6. Éléments de configuration de RMAN

- Affichage des paramètres de configuration : `rman> show all`
- Modification des paramètres : commande `CONFIGURE`
- Exemples

Sauvegarde sur disque dur (sbt pour mémoire de type bande)

```
CONFIGURE CHANNEL DEVICE TYPE DISK MAXPIECESIZE 4G
```

Durée de conservation des données de sauvegarde (fenêtre de restauration, les sauvegardes obsolètes sont supprimées)

```
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF n DAYS
```

Nombre de copies de sauvegarde

```
CONFIGURE RETENTION POLICY TO REDUNDANCY n
```

Sauvegarde automatique du fichier de contrôle

```
CONFIGURE CONTROLFILE AUTOBACKUP ON
```

7. Zone de récupération rapide (Falsh Recovery Area)

- Sauvegarde sur disque : moins sûre mais plus rapide (minutes au lieu d'heures) qu'une sauvegarde sur bande
- *Falsh Recovery Area* : zone sur disque réservée au stockage de toutes données et les activités de sauvegardes (depuis la version 10g)
- Permet de bénéficier de certaines opérations automatiques de sauvegarde/récupération
- Contient
 - Une copie du fichier de contrôle (multiplexage possible)
 - Fichiers des archives log : LOG_ARCHIVE_DEST_10 automatiquement mis sur l'emplacement de la zone de récupération rapide
 - Copie du fichier de paramètres
 - Backup sets (jeux de sauvegarde) générés par RMAN

7. Zone de récupération rapide (Fast Recovery Area)

Configuration

- 1 ALTER SYSTEM SET DB_RECOVERY_FILE_DEST_SIZE = *int* [M|G]
SCOPE=BOTH
- 2 ALTER SYSTEM SET DB_RECOVERY_FILE_DEST = '*Chemin*' SCOPE=BOTH

Chapitre 7 - Sauvegarde et récupération

- Introduction
- I. Concepts de base
- II. Sauvegarde avec RMAN
- III. Restauration et récupération avec RMAN
 - 1. Vue d'ensemble
 - 2. Principe généraux de la récupération
 - 3. Commandes de restauration/récupération
 - 4. Scénarios de récupération
- Quiz
- Exercice

1. Vue d'ensemble

■ En cas de problème(s)

- 1 Identifier la nature du problème : perte du fichier de paramètres, de contrôle, de journalisation, etc.
- 2 Planifier la récupération en fonction du problème et du mode de fonctionnement de la BD (ARCHIVELOG ou NOARCHIVELOG)

■ "Récupération" : deux étapes, **restauration**, puis **récupération**

1 Restauration

- Extraire d'une sauvegarde les fichiers nécessaires : fichier(s) endommagé(s) et si disponibles les archives permettant la récupération
- Commande RMAN : `rman> restore cible`

2 Récupération

- Application des fichiers de journalisation au(x) fichier(s) restauré(s)
- Commande RMAN : `rman> recover cible`

2.a. Récupération en mode ARCHIVELOG

■ Procédé :

- 1 Restaurer le(s) fichier(s) endommagé(s) (dans l'ordre le fichier de paramètres, de contrôle, puis les autres)
- 2 Récupération : appliquer les fichiers de journalisation (les archives, puis ceux en ligne)
- 3 Redémarrer la BD

- Toutes les modifications apportées depuis la sauvegarde sont récupérées

2.b. Récupération en mode NOARCHIVELOG

■ Procédé :

- 1 Restaurer la dernière sauvegarde complète de la BD
- 2 Redémarrer la BD

■ Récupération (RECOVER)

- Possible uniquement si depuis la dernière sauvegarde un cycle complet de basculement entre groupes de journalisation n'a pas eu lieu
- Sinon, les modifications faites depuis la dernière sauvegarde sont définitivement perdues

3. Commandes de restauration/récupération

- Syntaxe (non exhaustive) : {RESTORE | RECOVER} *cible* [*options*]
- *cible*
 - DATABASE
 - TABLESPACE *ListeNomTablespaces*
 - CONTROLFILE [FROM AUTOBACKUP]
 - SPFILE [FROM AUTOBACKUP]

4.a. Récupération du fichier de paramètres

- Deux possibilité : à partir d'un PFILE, si disponible ou avec RMAN

SET DBID <i>dbid</i> ;	# placer RMAN sur la BD à restaurer
STARTUP NOMOUNT;	# RMAN utilise un SPFILE "temporaire"
RESTORE SPFILE FROM AUTOBACKUP	# restauration du SPFILE
SQL "ALTER DATABASE OPEN"	# ouvrir la base

Remarques

- 1 La récupération du fichier de paramètres et de contrôle, si endommagés, doit précéder la récupération des autres fichiers de la BD
- 2 Il est possible de connaître le dbid soit à partir du message de démarrage de rman soit en interrogeant la vue v\$database.

4.b. Récupération d'une partie de la BD en mode ARCHIVELOG

- Peut se faire BD ouverte, si la BD est restée ouverte et que le tablespace endommagé n'est pas le tablespace SYSTEM

```
STARTUP MOUNT;           # démarrer la BD en mode MOUNT
RESTORE TABLESPACE nomTbs;  # Restauration de nomTbs
RECOVER TABLESPACE nomTbs  # suppression des archives au fur et
DELETE ARCHIVELOG          # à mesure de leur application
SQL "ALTER DATABASE OPEN"   # ouvrir la base
```


4.c. Récupération de tous les fichiers de contrôle

SET DBID <i>dbld</i> ;	
STARTUP NOMOUNT;	# démarrer la BD en mode NOMOUNT
RESTORE CONTROLFILE FROM	# Restauration du fichier de
AUTOBACKUP;	contrôle
RECOVER DATABASE	# Obligatoire car on repart d'une
	copie du fichier de contrôle
SQL "ALTER DATABASE RESETLOGS"	# Remise à zéro du numéro de
	séquence des logs
SQL "ALTER DATABASE OPEN"	# ouvrir la base

N.B. La BD ainsi récupérée est considérée comme une nouvelle incarnation de la BD endommagée. Elle garde le même nom, le même DBID, mais le SCN et le numéro de séquence log sont remis à zéro

Chapitre 7 - Sauvegarde et récupération

- Introduction
- I. Concepts de base
- II. Sauvegarde avec RMAN
- III. Restauration et récupération avec RMAN
- **Quiz**
- Exercice

Quiz

- 1 Expliquez la raison pour laquelle une sauvegarde partielle se fait généralement à chaud.
- 2 Quelle est la différence entre une sauvegarde logique et une sauvegarde physique ?
- 3 Qu'est-ce que le mode ARCHIVELOG et pourquoi est-il important ?
- 4 Expliquez la différence entre une sauvegarde cohérente (*cold backup*) et incohérente (*hot backup*).
- 5 Quelle est la différence entre un format d'image et un jeu de sauvegarde (*backup set*) dans RMAN ?
- 6 Quels sont les deux étapes principales de la récupération d'une base de données ?
- 7 Donnez un exemple de commande RMAN pour sauvegarder une base ouverte avec les journaux archivés.
- 8 Expliquez ce qu'est une "zone de récupération rapide" (*Flash Recovery Area*) et son utilité.



Chapitre 7 - Sauvegarde et récupération

- Introduction
- I. Concepts de base
- II. Sauvegarde avec RMAN
- III. Restauration et récupération avec RMAN
- Quiz
- Exercice

Exercice

Un administrateur d'un serveur de BD Oracle souhaite établir un plan de sauvegarde et de restauration. Les contraintes qu'il doit prendre en compte sont les suivantes :

- Il **n'est pas** nécessaire que le serveur fonctionne en permanence (7 jours/7, 24 heures/24);
 - La perte de données est tolérable;
 - Les tablespaces de la base n'ont pas la même fréquence de sauvegarde (e.g. certains tablespaces doivent être sauvegardés toutes les semaines et d'autres tous les mois).
 - La base de données est volumineuse et il est impératif de réduire autant que possible l'espace de stockage occupé par les sauvegardes;
 - Si une restauration de la base est nécessaire, la restauration doit prendre le minimum de temps possible.
- 1 Proposez à l'administrateur le meilleur plan de sauvegarde possible. Indiquez notamment : (a) si la base doit fonctionner en mode archivage ou sans archivage; et (b) le type et le format de sauvegarde qui vous semblent les plus adéquats.
 - 2 En supposant que le serveur fonctionne en mode sans archivage (NOARCHIVELOG) et que l'un de ses tablespaces ainsi que tous les fichiers de contrôle ont été endommagés, dites quelles sont les étapes nécessaires à la restauration.