

# Administration des Bases de Données Oracle



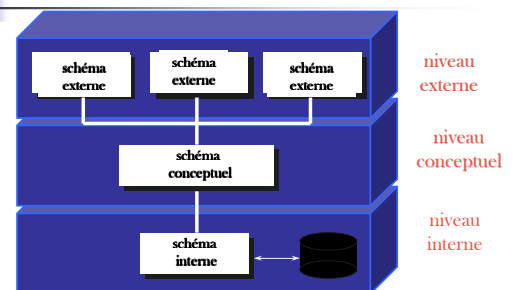
## Plan

- Introduction à l'administration de données
- Structure d'une base de données
- Architecture interne d'Oracle
- Sécurité des données
- Utilitaires

## Introduction

- L'administration de données consiste à :
  - créer la base de données
  - définir les objets de la base
  - veiller à la bonne utilisation des données.
- Utilisation de l'architecture ANSI/SPARC
  - niveau conceptuel
  - niveau externe
  - niveau interne

## Architecture ANSI/SPARC



## Rôle de l'administrateur

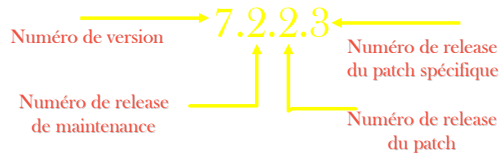
- L'administrateur peut avoir un double rôle:
  - rôle organisationnel
  - rôle technique
- ces deux rôles peuvent être assurés par une ou plusieurs personnes.

## Rôle technique

- Installation du SGBD et des outils associés
- Création de la base de données et assurer son évolution
- Gestion des privilèges d'accès
- Amélioration des performances
- Sécurité et cohérence des données
- Echange de données entre la base et le monde extérieur
- Assistance aux utilisateurs

## Identification du produit Oracle

- Pour identifier un des produits logiciels, Oracle utilise jusqu'à 5 chiffres :



## Identification du produit Oracle

```

Oracle SQL*Plus
Enter SQL statement or \Q to quit

SQL*Plus: Release 3.2.2.0.1 - Production on Fri Jan 17 00:08:54 1997
Copyright (c) Oracle Corporation 1979, 1994. All rights reserved.

Connected to:
Personal Oracle7 Release 7.2.2.3.1 - Production
With the distributed and replication options
PL/SQL Release 2.2.2.3.1 - Production

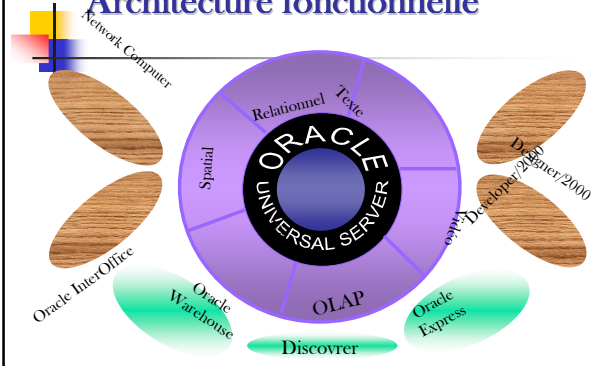
SQL> column product format a20
SQL> column version format a10
SQL> column status format a10
SQL> select * from product_component_version;

PRODUCT                VERSION        STATUS
-----
CORE                    3.1.4.2.2.0   Production
MSRTL                   3.1.4.0.0     Production
PL/SQL                  2.2.2.3.1     Production
Personal Oracle7        7.2.2.3.1     Production
SQL>
    
```

## Architecture d'Oracle

- Oracle Universal Server inaugure une nouvelle génération de systèmes d'information pour gérer tout type de données.
- Architecture fonctionnelle
- Architecture Interne du noyau

## Architecture fonctionnelle



## Architecture fonctionnelle (suite)

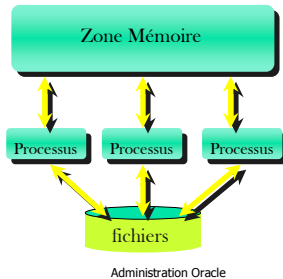
- Caractéristiques de l'offre Oracle
  - Portabilité** : critère décisif pour les décideurs
  - Interopérabilité** : client/serveur, protocoles réseaux, passerelles
  - Conformité aux normes et standards** : ANSI, ISO, OSF, X/OPEN, OMG etc.

## Oracle Universal Server

- Les options d'Oracle Universal Server sont regroupées en deux types :
  - Options techniques** : Oracle distributed, Oracle Symmetric Replication, Oracle Parallel Server, Advanced Networking Option, Enterprise Manager Performance Pack, Oracle VLM.
  - Options fonctionnelles** : Context Option, TextServer Option, Video Option, Spatial Data Option

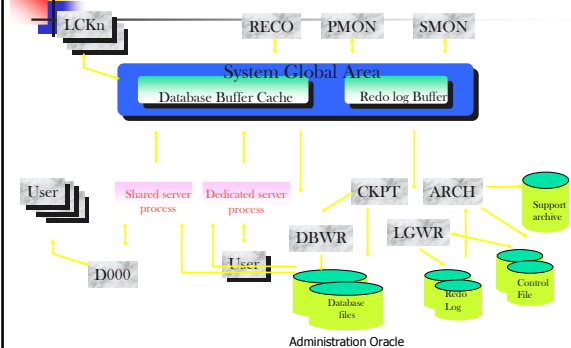
## Architecture Interne d'Oracle

- L'architecture interne d'Oracle est composée:



13

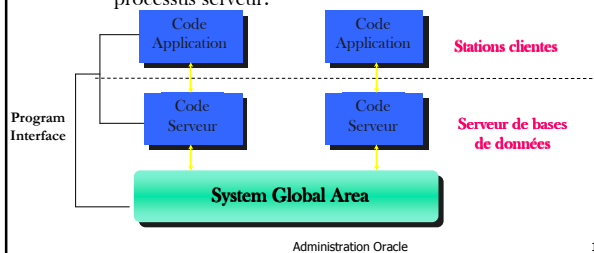
## Architecture globale



14

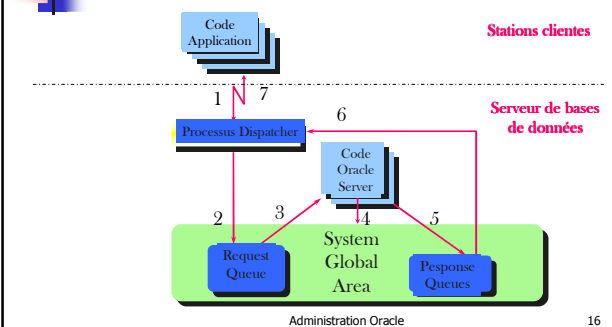
## Configuration en serveur dédié

- Pour chaque processus client exécuté est associé un processus serveur.



15

## Configuration en Multi-Threaded



16

## Les processus

- Deux types de processus :
  - Processus utilisateurs : pour exécuter une application SQL\*Plus, Pro\*C, etc.
  - Processus Oracle
    - Processus SERVER : interaction entre processus utilisateur et l'instance (mémoire de la SGA).
    - Processus en arrière plan (BACKGROUND) : effectue des tâches bien définies pour l'ensemble des utilisateurs.

Administration Oracle

17

## Les processus (suite)

- **PMON** : Process Monitor est chargé du nettoyage en cas de crash d'un processus utilisateur.
- **SMON** : System Monitor permet de restaurer la base au démarrage de l'instance.
- **DBWR** : Database Writer est chargé de l'écriture des buffers de données sur disque.
- **LGWR** : Log Writer est chargé de la gestion du buffer redo log de la SGA.

Administration Oracle

18

## Les processus (suite)

- **CKPT** : Checkpoint est optionnel pour écrire les données les plus fréquemment modifiées.
- **ARCH** : Archiver est chargé de copier le fichier Redo Log courant dès qu'il est plein sur la destination des fichiers d'archives.
- **RECO** : Recover est chargé de résoudre les anomalies de transactions distribuées.
- **LCKn** : Lock prend en charge la gestion de verrous dans une architecture Parallel Server.

## PMON (Process Monitor)

- Assure le recouvrement des processus utilisateurs en cas de problème en libérant les ressources qui étaient utilisées par ces processus.
- Assure la libération des segments temporaires qui ne sont pas utilisés.
- réveille régulièrement pour effectuer ce recouvrement.

## SMON (System Monitor)

- Assure le recouvrement d'instance lors du démarrage d'instance.
- Assure la libération des segments temporaires qui ne sont pas utilisés.
- réveille régulièrement pour effectuer cette opération
- peut être déclenché par un autre processus.

## DBWR (Database Writer)

- gère le buffer de base de données
- écrit les modifications apportées au buffer dans les fichiers de données
- maintient ce buffer «propre» avec un minimum de tampons modifiés
- Utilisation de l'algorithme LRU (Least Recently Used) qui permet de minimiser les opérations d'entrée/sortie.

## LGWR (Log Writer)

- Écrit les buffers Redo Log dans les fichiers de reprise.
- En cas d'existence de groupe de fichiers de reprise, LGWR effectue une écriture synchrone sur tous les fichiers activés.
- Ce processus est activé de la façon suivante :
  - toutes les 3 secondes
  - lorsque le tiers du buffer Redo Log est plein
  - lorsque le DBWR effectue une écriture des buffers modifiés sur disque.

## CKPT (Checkpoint)

- le processus DBWR effectue régulièrement une écriture sur les fichiers de données de toutes les modifications dans le buffer de données.
- Cette opération est dite (checkpoint).
- Utilité :
  - permet l'écriture des données qui sont dans le buffer de base de données et qui sont modifiées fréquemment (l'algorithme LRU n'entraîne pas leur écriture).
- Cette opération de checkpoint est normalement effectuée par le processus LGWR.

## RECO (Recover)

- Utilisé uniquement en cas de base de données répartie.
- permet d'effectuer le recouvrement en cas d'échec d'une transaction répartie.
  - il effectue une connexion avec toutes les bases distantes concernées par la transaction et active la transaction.
- activé au moment du démarrage d'une instance si l'option est installée.

Administration Oracle

25

## ARCH (Archiver)

- Effectue la copie des fichiers de reprise ayant atteint leur taille maximale sur un support d'archive.
- Actif uniquement si la base est en mode ARCHIVELOG et l'archivage automatique est activé.

Administration Oracle

26

## LCKn (LOCKn)

- N'existe qu'avec l'option Parallel Server
- Permet de gérer le verrouillage inter-instance.
- On peut avoir jusqu'à 10 processus de type (LCK0,LCK1,.....,LCK9).

Administration Oracle

27

## Dnnn (Dispatcher)

- Permet le partage d'un nombre limité de processus serveurs par les processus utilisateurs.
- Utilisé avec la configuration Multi-Threaded Server.
- Permet de minimiser le nombre des processus serveurs lorsque le nombre d'utilisateurs est important.
- Le nombre de processus dispatchers est fixé par l'administrateur dans init.ora.

Administration Oracle

28

## Processus serveur

- Utilise l'architecture client/serveur
- Fonctions des processus serveurs :
  - analyser et exécuter les commandes SQL soumises par les applications (utilisateurs).
  - transférer les blocs de données nécessaires du disque vers la zone SGA.
  - communiquer les résultats de requêtes aux applications (utilisateurs).

Administration Oracle

29

## La mémoire

- Oracle utilise les structures suivantes au niveau de la mémoire centrale :
  - le code de l'applicatif
  - les informations relatives à une session ouverte
  - les données nécessaires pendant l'exécution d'un programme.
  - les informations partagées par différents processus Oracle (relatives au verrouillage par exemple).
  - les données échangées avec les mémoires auxiliaires (Bloc de données par exemple).

Administration Oracle

30

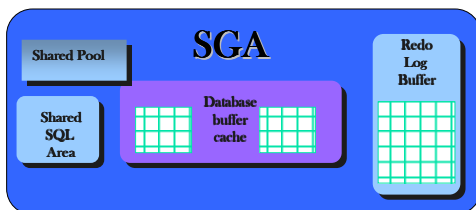
## La mémoire (suite)

- La structure de la mémoire est composée de trois types de zones :
  - zones réservées au code de l'appliquatif.
  - zone globale système (System Global Area ou **SGA**).
  - zone globale programme (Program Global Area ou **PGA**).

## Zone du code de l'appliquatif

- Permet de stocker le code des programmes en cours d'exécution
  - code du noyau
  - outils Oracle (SQL\*Forms, SQL\*Plus...)
  - programmes d'application en Pro\*langage
- le code du noyau est stocké dans une zone séparée des autres zones.
- les zones réservées au code de l'appliquatif sont accessibles uniquement en lecture.

## System Global Area (SGA)



## Estimation de la taille de la SGA

- Déterminée en fonction des paramètres :
  - DB\_BLOCK\_SIZE
  - DB\_BLOCK\_BUFFERS
  - LOG\_BUFFER
  - SHARED\_POOL\_SIZE
- Formule :

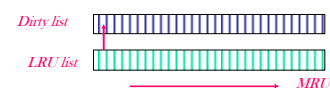
$$\text{Taille de la SGA} = (\text{DB\_BLOCK\_SIZE} * \text{DB\_BLOCK\_BUFFERS}) + \text{LOG\_BUFFER} + \text{SHARED\_POOL\_SIZE}$$

## Database buffer cache

- ensemble de zones buffer contenant des copies de données lues à partir des fichiers de données.
- taille déterminée par : DB\_BLOCK\_BUFFERS
- Oracle gère l'espace de ce cache à l'aide de deux listes :
  - une liste des buffers modifiés mais non encore écrits sur disque (**DIRTY LIST**)
  - une liste des buffers les moins utilisés récemment (**LEAST RECENTLY USED LIST**) LRU list.

## Database buffer cache (suite)

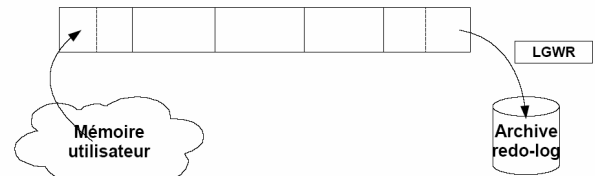
- La liste LRU contient :
  - les buffers libres pouvant être utilisés
  - les buffers utilisés actuellement par certains processus
  - les buffers modifiés mais non encore transférés dans la première liste (dirty list).



## Algorithme LRU

- utilisé pour transférer un bloc de données du disque vers le cache de la base de données
  - recherche d'un buffer libre dans la liste LRU en commençant par le début
  - arrêt dès qu'on trouve un buffer libre ou qu'on ait parcouru un certain nombre de buffers (DB\_BLOCK\_SCAN\_CNT)
  - pendant cette recherche, transfert des buffers modifiés vers la dirty list
  - dès qu'on trouve un buffer libre, on écrit le bloc dans le buffer et on le transfère vers la fin de la zone MRU
  - si DB\_BLOCK\_SCAN\_CNT buffers sont parcourus et on ne trouve aucun buffer libre, DBWR est activé pour écrire les buffers modifiés sur disque pour libérer leur espace.

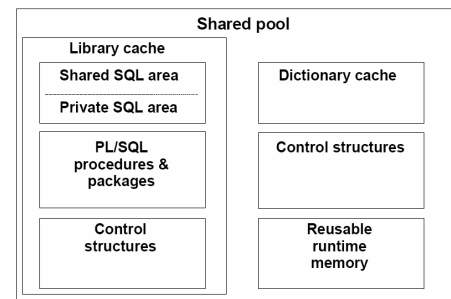
## Redo Log buffer



## Redo Log buffer

- buffer circulaire qui contient des informations relatives aux modifications apportées à la base de données. Ces informations (entrées redo log) sont utilisées en cas de restauration.
- le contenu de ce buffer est écrit dans le fichier Redo Log par le processus LGWR.
- sa taille est déterminée par LOG\_BUFFER

## Shared Pool



## Shared Pool (Suite)

- la shared pool contient :
  - des zones SQL partagées ou privées : une zone SQL partagée contient le plan d'exécution d'un ordre SQL. Cette zone est partagée par toutes les commandes SQL identiques.
  - un cache du dictionnaire contenant des informations relatives au dictionnaire.
- sa taille est déterminé par SHARED\_POOL\_SIZE
- gérée à l'aide de l'algorithme LRU

## Program Global Area (PGA)

- zone mémoire contenant des données et des informations de contrôle relatives à un processus (processus serveur ou processus d'arrière plan).
- une zone PGA est allouée lorsqu'un processus utilisateur se connecte à Oracle (ouverture d'une session).
- la taille de la PGA est déterminée par :
  - OPEN\_LINKS
  - DB\_FILES
  - LOG\_FILES

## Sort Area

- zone de mémoire allouée par le processus utilisateur en cas de besoin de tri
- taille déterminée par SORT\_AREA\_SIZE
- pour un nombre important d'occurrences et en cas où la zone de tri est petite, Oracle effectue ses opérations de tri sur disque (tablespace spécifique).

## Les fichiers

- Fichiers de données
- Fichiers Redo Logs
- Fichier de contrôle
- Fichier de configuration

## Structure d'une base de données Oracle



## Structure d'une base de données

- Toute base de données Oracle possède :
  - une structure logique
  - une structure physique
  - un ensemble de vues donnant des visions partielles aux utilisateurs
- La description de ces trois niveaux et la correspondance entre eux est faite à travers le dictionnaire de données.

## Structure physique

- La structure physique est composée d'un ensemble de fichiers qui constituent le support physique de stockage de données.
- Trois types de fichiers :
  - fichiers de données (**Data Files**)
  - fichiers de reprise (**Redo Log File**)
  - fichiers de contrôle (**Control Files**)
- La spécification des Data files et Redo Log se fait lors de la création ou la modification de la base de données.



## Fichiers de données (Data Files)

- Les fichiers de données assurent le stockage
  - des objets créés par les utilisateurs : tables, index, clusters etc.
  - des structures nécessaires au fonctionnement d'Oracle : Dictionnaire de données.
- Lors de la création d'une base de données, il doit y avoir au moins un fichier de données pour stocker le dictionnaire de données.
- Un fichier de données est associé à une et une seule base de données



## Taille d'un Data File

- Exprimée en Kilo ou en Méga octets
- Attribuée au moment de la création du fichier, peut être modifiée.
- Son choix dépend de la taille de la base de données et de son évolution.
- Le premier fichier de données créé doit avoir une taille supérieure ou égale à 2Mo.

## Nom d'un Data File

- Dépend du système d'exploitation
- Peut être modifié, sauf pour le premier fichier.
- La modification se fait en deux étapes :
  - modification du nom du fichier en utilisant les commandes OS
  - répercussion de cette modification dans les fichiers de contrôle.

## Création d'un Data File

- Se fait dans quatre configurations :

```
CREATE DATABASE base
DATAFILE spécification_fichier
```

```
CREATE TABLESPACE tbs
DATAFILE spécification_fichier
```

```
ALTER DATABASE base
CREATE DATAFILE fichier AS spécification
```

```
ALTER TABLESPACE tbs
ADD DATAFILE spécification_fichier
```

## Modification d'un Data File

- Deux cas de modifications :
  - modification du nom et de localisation
  - modification de la taille
- Modification du nom : TBS en Offline

```
ALTER TABLESPACE tbs
RENAME DATAFILE 'fich1', 'fich2' TO 'fich11', 'fich22'
```

```
ALTER DATABASE base
RENAME FILE 'fich1', 'fich2' TO 'fich11', 'fich22'
```

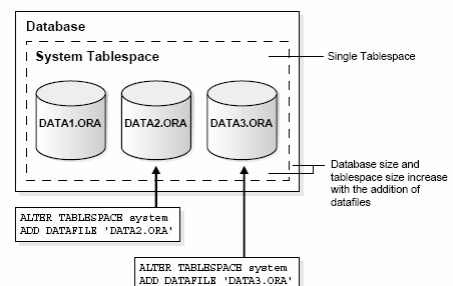
## Modification d'un Data File

- Modification de la taille :

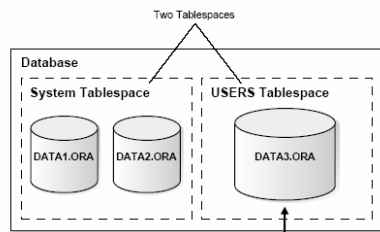
```
ALTER TABLESPACE tbs
ADD DATAFILE spécification
AUTOEXTEND [OFF | ON [NEXT entier [K|M]]
[MAXSIZE UNLIMITED | entier [K|M]]
```

```
ALTER DATABASE base
DATAFILE 'fich1' RESIZE entier [K|M]
AUTOEXTEND [OFF | ON [NEXT entier [K|M]]
[MAXSIZE UNLIMITED | entier [K|M]]
```

## Élargir une BD en ajoutant un Datafile au TableSpace



## Élargir une BD en ajoutant une nouvelle Tablespace

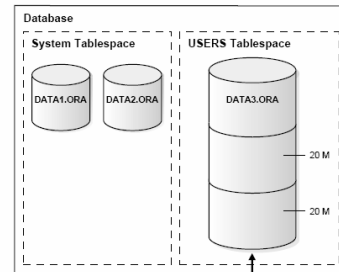


```
CREATE TABLESPACE users
DATAFILE 'DATA3.ORA'
```

Administration Oracle

55

## Élargir une BD en modifiant d'une manière dynamique la taille des Datafiles



```
ALTER DATABASE
DATAFILE 'DATA3.ORA'
AUTOEXTEND ON NEXT 20M
MAXSIZE 1000M;
```

56

## Suppression d'un Data File

- Deux étapes :
  - Suppression du tablespace associé au Data file
 

```
DROP TABLESPACE tbs
INCLUDING CONTENTS
```
  - Suppression des fichiers physiques manuellement par commande OS.

Administration Oracle

57

## Fichiers Redo Log

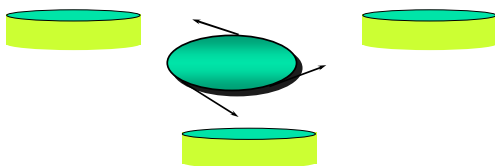
- contiennent les modifications les plus récentes des données de la base.
- utilisés par Oracle pour remettre la base dans un état cohérent après une panne sans perdre les maj qui n'ont pas été enregistrées dans les data files.
- lors du redémarrage de l'instance, Oracle applique les maj des Red Log sur les Data file (reprise à chaud).

Administration Oracle

58

## Fichier Redo Log (suite)

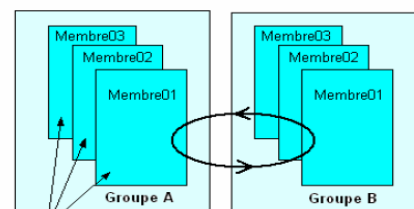
- Au moins deux fichiers par instance. Utilisés d'une façon circulaire.



Administration Oracle

59

## Fichiers Redo Log (Suite)



Oracle écrit en miroir dans chaque membre de Redo-Log

Groupes et membres de fichiers Redo Log

Administration Oracle

60

## Redo Log

- Redo Log multiple : duplication des fichiers Redo Log sur différents disques. Lorsqu'un des fichiers est perdu ou endommagé, son image sur l'autre disque peut être utilisé.
- L'ensemble des Redo Logs actifs simultanément constitue un *groupe*. Chaque fichier de ce groupe est dit *membre*.
- Avant de réutiliser un Redo Log plein, Oracle peut l'archiver si la base est en mode *ARCHIVELOG*.

## Création d'un Redo Log

- Il est possible de créer ou d'ajouter des Redo Logs dans une base existante :

```
CREATE DATABASE base  
LOGFILE [GROUP entier] ('fich', 'fich',...) SIZE entier [K|M]...
```

```
ALTER DATABASE base  
ADD LOGFILE [THREAD entier] [GROUP entier] ('fich', 'fich',...) SIZE  
entier [K|M]....  
ADD LOGFILE MEMBER 'fich' [REUSE] TO [GROUP entier | 'fich']
```

## Suppression d'un Redo Log

- La suppression d'un Redo Log, d'un groupe ou d'un membre se fait par :

```
ALTER DATABASE base  
DROP LOGFILE [GROUP entier | ('fich', 'fich',...) ] ....  
DROP LOGFILE MEMBER ('fich' | ('fich', 'fich',...))
```

- Un Redo Log peut être initialisé par :

```
ALTER DATABASE base  
CLEAR [UNARCHIVED] LOGFILE [GROUP entier |  
( 'fich', 'fich',...) ) ] .... [UNRECOVERABLE DATAFILE]
```

## Control File

- Contient les informations relatives à la structure physique de la base de données:
  - nom de la base de données
  - noms et localisations des Datafile et Redo Logs
  - timestamp de création de la base
  - le nombre de séquence log courant
  - informations sur le checkpoint
- Créé au moment de la création de la base

## Control File (Suite)

- Utilisé lors de l'ouverture d'une base pour identifier les Datafiles et les Redo Logs.
- Au moins un fichier de contrôle (recommandation : dupliquer ce fichier plusieurs fois)
- Mise à jour automatique à chaque modification de structure de la base.
- Noms spécifiés dans init.ora

## Structure logique

- La structure logique d'une base est composée
  - des *tablespaces*
  - des *segments*
  - des *extensions* (extents)
  - des *blocs*
- Un ensemble d'objets logiques constitué d'objets de schéma (*schema objects*).
- Les objets de schéma constituent la structure relationnelle de la base.

## Tablespace

- Une base de données est composée d'un ensemble d'unités logiques dites tablespaces
- Un tablespace permet de regrouper un ensemble d'objets logiques (tables, index...)
- il peut être utilisé pour regrouper des objets logiques d'une application pour que les opérations de sauvegarde et de restauration soient efficaces.

## Tablespace (Suite)

- Chaque objet logique doit être associé à un et un seul tablespace.
- Chaque tablespace est identifié par un nom.
- Un tablespace peut être supprimé même s'il contient des données.
- Une base de données doit avoir au moins un tablespace appelé **SYSTEM** qui contient le dictionnaire de données.

## Tablespace (Suite)

- Un tablespace peut être activé ou désactivé
  - OnLine
  - OffLine
- Le tablespace **SYSTEM** ne peut jamais être désactivé.
- Il est recommandé de créer d'autres tablespaces pour les objets de la base.
- Un tablespace peut être créé, modifié et supprimé

## Création de tablespace

```
CREATE TABLESPACE tbs
DATAFILE spec [AUTOEXTEND {OFF|ON}
[NEXT entier {M|K}]
[MAXSIZE {UNLIMITED | entier {M|K}}]...
[DEFAULT STORAGE clause_stockage]
[ONLINE | OFFLINE]
[PERMANENT | TEMPORARY]
```

## Modification d'un tablespace

```
ALTER TABLESPACE tbs
ADD DATAFILE spec [AUTOEXTEND
{OFF|ON}
[NEXT entier {M|K}]
[MAXSIZE {UNLIMITED | entier {M|K}}]...
[RENAME 'fich', ... TO 'fich', ...]
[ONLINE | OFFLINE]
[READ {ONLY | WRITE}]
[PERMANENT | TEMPORARY]
```

## Suppression d'un tablespace

```
DROP TABLESPACE tbs
[INCLUDING CONTENTS]
[CASCADE CONSTRAINTS]
```

➤ **INCLUDING CONTENTS** supprime le contenu de tablespace

➤ **CASCADE CONSTRAINTS** supprime les contraintes d'intégrité des tables d'autres tablespaces qui se réfèrent aux clés primaires des tables du tablespace spécifié.

## Utilisation des tablespaces

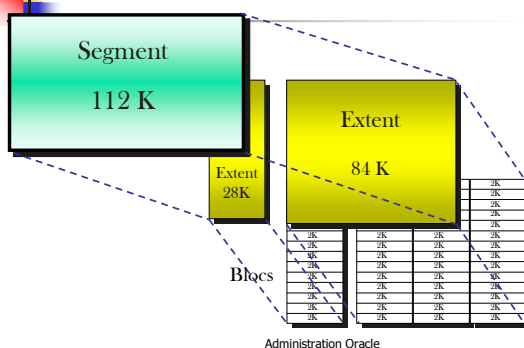
- Un tablespace peut être attribué par défaut à un utilisateur.
- Les objets de cet utilisateur seront stockés dans ce tablespace sauf contre indication.

```
CREATE | ALTER user IDENTIFIED BY mdp
[DEFAULT TABLESPACE tbs]
[TEMPORARY TABLESPACE tbs]
[QUOTA [entier [M | K] | UNLIMITED] ON tbs]
```

## SEGMENT, EXTENSION, BLOC

- Lors de la création d'un fichier, Oracle réserve tout l'espace qui lui est associé. A l'intérieur de ce fichier, l'espace disque est géré dynamiquement au fur et à mesure de l'utilisation de la base de données.
- Cette gestion se fait selon trois niveaux de granularité :
  - segment
  - extension
  - bloc

## SEGMENT, EXTENSION, BLOC

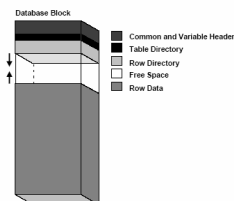


## BLOC de données

- Un bloc de données est la plus petite unité logique d'entrée/sortie utilisée par Oracle.
- Appelé également *Bloc logique* ou *Page*
- Taille d'un bloc de données
  - attribuée par défaut selon OS (2 ou 4Ko)
  - modifiable par DB\_BLOCK\_SIZE = n

## Format d'un bloc de données

- Les blocs de données sont organisés de la même façon quelque soit leur contenu.



## Format d'un bloc de données

- **L'entête** : contient des informations générales (adresse du bloc, type du segment, etc)
- **Liste des tables** : Contient des informations relatives aux tables groupées ayant des données dans ce bloc.
- **Liste des lignes** : contient des informations relatives aux lignes situées dans ce bloc (adresse).
- **Espace libre** : utilisé pour l'insertion de nouvelles lignes ou la maj des lignes nécessitant de l'espace.
- le reste du bloc est réservé aux données.

## Extention (Extent)

- Unité logique d'allocation d'espace composée d'un ensemble contiguë de blocs de données alloués simultanément à un segment.
- Tout segment est initialement créé avec au moins une extension appelée extension initial (*INITIAL EXTENT*).
- lorsque l'espace d'un segment est complètement utilisé, Oracle lui attribue une nouvelle extension dite extension supplémentaire (*INCREMENTAL EXTENT*).

Administration Oracle

79

## Attribution d'une nouvelle extension

- recherche dans le tablespace contenant le segment le *premier ensemble contiguë* de blocs libres de taille au moins égale à celle de l'extension supplémentaire.
- si la recherche n'aboutit pas, Oracle *fusionne* les blocs libres pour constituer des ensembles de blocs libres de taille supérieure, puis reprend la recherche. Si de nouveau la procédure de recherche n'aboutit pas, message d'erreur signalé.
- modification de l'entête du segment et du dictionnaire de données pour tenir compte de la nouvelle extension.

Administration Oracle

80

## Libération des extensions

- les extensions d'un segment ne sont libérées que lorsque le segment est supprimé suite à la suppression d'une table, d'un cluster, etc
- lorsque des extensions sont libérées, Oracle met à jour le dictionnaire de données.

Administration Oracle

81

## Taille des extensions

- La taille des extents est définie par les paramètres de la clause **STORAGE** :

```
STORAGE ( INITIAL n1 NEXT n2  
          MINEXTENTS m1 MAXEXTENTS m2  
          PCTINCREASE p)
```

n1 : taille du premier extent alloué lorsque un segment est créé

n2 : taille du second extent alloué au segment

m1 : nombre d'extents alloués à la création du segment

m2 : nombre total d'extents pouvant être alloués à segment

p : pourcentage d'accroissement de la taille du segment I+1 par rapport à celle du segment I (I > 2).

La taille du segment I+1 = (taille du segment I) \* (1+P/100).

Administration Oracle

82

## Exemple

```
STORAGE ( INITIAL 50K NEXT 50K  
          MINEXTENTS 1 MAXEXTENTS 5  
          PCTINCREASE 50)
```

- Si la taille d'un bloc = 2K, la taille des différents extents sera :
  - 1er EXTENT : 50K
  - 2me EXTENT : 50K
  - 3me EXTENT :  $\text{ARRONDI}(50 * 1,5) = 76\text{K}$
  - 4me EXTENT :  $\text{ARRONDI}(76 * 1,5) = 114\text{K}$
  - 5me EXTENT :  $\text{ARRONDI}(114 * 1,5) = 172\text{K}$

Administration Oracle

83

## SEGMENT

- Un segment est composé d'un ou de plusieurs extents contenant les données d'une structure logique dans un tablespace.
- Quatre types de segments :
  - segments de données
  - segments d'index
  - segments de rollback
  - segments temporaires

Administration Oracle

84

## Segments de données

- servent à stocker les données des tables utilisateurs et système.

## Segments d'index

- servent à stocker les données d'index séparément des données.

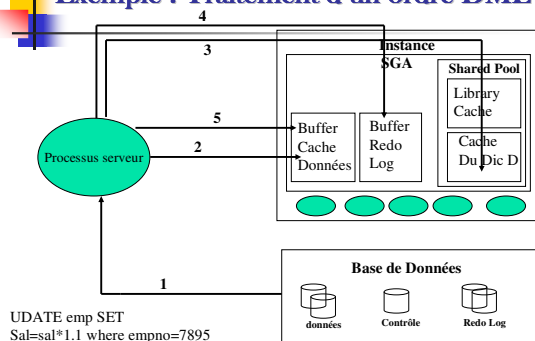
## Segments temporaires

- utilisés pour le traitement des commandes SQL nécessitant un espace disque temporaire (order by, group by, distinct, union, intersect ou minus).

## Segments rollback

- enregistrent les actions effectuées par les transactions. Ils contiennent les données avant modification par les transactions, et permettent d'annuler leur effet en cas de besoin.

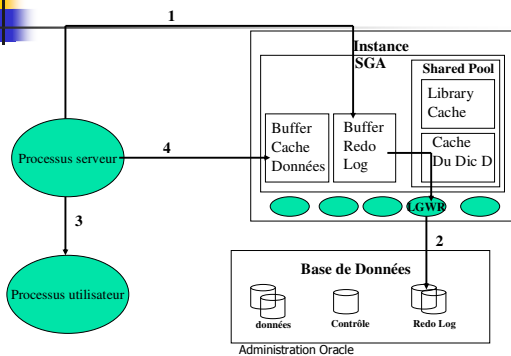
## Exemple : Traitement d'un ordre DML



## Exécution

- 1- Le Processus serveur lit les blocs de données et de rollback à partir des fichiers de données si ces derniers ne sont pas déjà dans le buffer cache.
- 2- Les copies des blocs lus sont placées dans le buffer cache
- 3- Le processus serveur met des verrous sur les données
- 4- Le processus serveur enregistre **les modifications à apporter** au rollback (« image avant ») et aux données (« nouvelle valeur ») **dans le buffer redo log**
- 5- Dans le buffer cache de données, le processus serveur enregistre l'image avant dans le rollback et met à jour le bloc de données. Ces 2 blocs modifiés dans le buffer cache sont identifiés comme buffers modifiés, car il diffèrent des blocs correspondants sur le disque.

## Exemple : Traitements des opérations Commi



91

## Étapes de commit

- 1- Le processus serveur place un enregistrement de validation avec le SCN dans le buffer redo log
- 2- Le LGWR effectue une écriture contigue de toutes les entrées du buffer redolog dans les fichiers redo log allant jusqu'à l'enregistrement de validation. Le serveur oracle peut garantir que les modifications ne seront pas perdues même en cas de panne.
- 3- L'utilisateur est informé de l'achèvement de l'opération commit
- 4- Le processus serveur enregistre les informations indiquant que la transaction est complète et que les verrous mis sur les ressources peuvent être supprimés

Administration Oracle

92

## System Change Number (SCN)

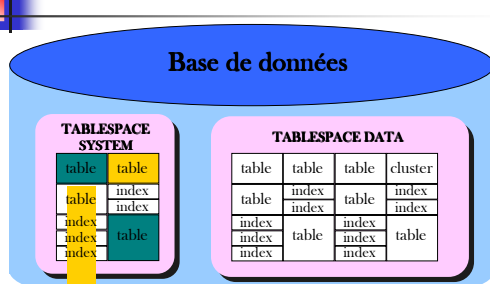
Que doit retourner une requête de consultation longue (dans la durée) durant laquelle d'autres utilisateurs ont effectué des modifications de courte durée ?

Toujours dans le but de garantir la consistance des informations fournies, une requête de consultation reçoit toujours les informations telles qu'elles étaient lors du début d'exécution de la requête. Pour ce faire plusieurs valeurs, pour une même information, sont stockées dans la base. Ce qui les différencie (outre la valeur), c'est la connaissance du **SCN** (moment auquel l'ajout dans la base a lieu, même si c'était une modification). A partir de là une requête select ne pourra jamais renvoyer une valeur dont le SCN est supérieur à celui de la transaction en cours (un SCN est associé à une transaction).

93

## Création, démarrage et arrêt d'une base de données

## Structure d'une base de données



Administration Oracle

95

## Création d'une base de données

- **Précautions à prendre avant de créer une base de données :**
  - être administrateur de base de données et disposer du plus haut niveau de privilège.
  - évaluer l'espace disque nécessaire et s'assurer que cet espace est disponible.
  - prévoir les moyens assurant la sécurité de la base de données (fichiers de reprise, archivage, sauvegarde et restauration de données, ...).

Administration Oracle

96



## Procédure de création de BD

- Sauvegarde des bases existantes
- Création d'un nouveau fichier d'initialisation
- Edition du nouveau fichier d'initialisation
- Edition de tous les fichiers d'initialisation existants
- Arrêt et redémarrage d'Oracle
- Création de la nouvelle base de données

## Sauvegarde des bases existantes

- Il est conseillé d'effectuer une sauvegarde des bases de données existantes avant de commencer la création d'une nouvelle base. Pour chaque base, il faut sauvegarder :
  - les fichiers d'initialisation (ou de paramètres),
  - les fichiers de données,
  - les fichiers de reprise (redo log),
  - les fichiers de contrôle

## Création du fichier init

### ■ Création d'un nouveau fichier init

- Chaque base de données doit avoir son fichier d'initialisation utilisé lors du démarrage d'une instance.
- Cette étape consiste à créer un nouveau fichier d'initialisation contenant les paramètres de la base à créer.
- Se fait en copiant l'un des fichiers d'initialisation existants (INIT.ORA par exemple) puis on l'édite pour l'adapter à la nouvelle base de données.

## Edition du nouveau fichier init

- Il s'agit d'adapter le nouveau fichier d'initialisation à la base de données à créer.
- On doit utiliser un nom de base de données et un nom de fichier de contrôle différents de ceux de la base existante.
- On doit donc modifier les paramètres suivants dans le fichier de paramètres :
  - DB\_NAME = *nouveau\_nom\_de\_bd*
  - DB\_DOMAIN = *nom\_domaine*
  - CONTROL\_FILES = (*fichier\_control\_1*, *fichier\_control\_2*, ...)
  - INT\_SQL\_FILES = (SQL.BSQ, CATALOG.ORA, *fichier\_1*, *fichier\_2*, ...)

## Edition des fichiers init existants

- S'il existe d'autres bases de données, mettre à jour leurs fichiers d'initialisation en spécifiant les paramètres DB\_NAME, DB\_DOMAIN et CONTROL\_FILES.
- Vérifier aussi que l'identificateur de l'instance Oracle est conforme au nom de la base de données (paramètre DB\_NAME).
- Vérifier que les valeurs de ces paramètres sont différentes de celles de la base à créer

## Arrêt et démarrage d'Oracle

- Si une base de données est ouverte, l'arrêter (SHUTDOWN).
- Se connecter ensuite avec le privilège INTERNAL et démarrer une nouvelle instance sans ouverture de base de données.

**STARTUP NOMOUNT PFILE=fichier\_init;**

## Création de la base

```
CREATE DATABASE [nom_base]
[CONTROLFILE REUSE]
[LOGFILE [GROUP entier] spécif_fichier] [, ...]
[MAXLOGFILES entier]
[MAXLOGMEMBERS entier]
[MAXLOGHISTORY entier]
[DATAFILE spécif_fichier [, spécif_fichier]...]
[MAXDATAFILES entier]
[MAXINSTANCES entier]
[ARCHIVELOG | NOARCHIVELOG]
[EXCLUSIVE]
[CHARACTER SET code];
```

## Création de la base

- La commande CREATE DATABASE entraîne la création des éléments suivants :
  - des fichiers de données,
  - des fichiers de reprise (redo log),
  - des fichiers de contrôle,
  - un tablespace SYSTEM et un rollback segment SYSTEM,
  - un dictionnaire de données (tables, vues et synonymes),
  - deux utilisateurs identifiés par SYS et SYSTEM.

## Modification d'une base

```
ALTER DATABASE [nom_base]
[ADD LOGFILE [THREAD entier] [GROUP entier] spécif_fichier...
[ADD LOGFILE MEMBER 'fichier'[REUSE] [, 'fichier'[REUSE]]...
TO [GROUP entier] ( 'fichier', 'fichier'...) | 'fichier'...]
[DROP LOGFILE [GROUP entier] ( 'fichier', 'fichier'...) | 'fichier']]...
[DROP LOGFILE MEMBER 'fichier' [, 'fichier'...]...]
[RENAME FILE 'fichier', 'fichier'... TO 'fichier', 'fichier'...]
[ARCHIVELOG | NOARCHIVELOG] [MOUNT [EXCLUSIVE |
PARALLEL]]
[OPEN [RESETLOGS | NORESETLOGS]]
[ENABLE [PUBLIC] THREAD entier] [DISABLE THREAD entier]
[BACKUP CONTROLFILE TO 'fichier'[REUSE]]
[DATAFILE 'fichier'[ONLINE | OFFLINE]]
[CREATE DATAFILE 'fichier', 'fichier'... AS spécif_fichier, spécif_fichier]...
[RENAME GLOBAL_NAME TO nom_base.domaine]...
[RECOVER clause_recover];
```

## Démarrage d'une base

- Le démarrage d'une base de données se fait en trois étapes :
  - démarrage d'une instance,
  - chargement de la base,
  - ouverture de la base.
- Il se fait suite à une connexion avec le privilège INTERNAL (CONNECT INTERNAL).

## Démarrage d'une instance

- Le démarrage d'une instance consiste à préparer le contexte nécessaire pour l'utilisation d'une base de données.
  - Allocation de l'espace mémoire constituant la SGA et
  - Lancement des différents processus d'Oracle.
- La zone mémoire allouée et les processus créés ne sont encore associés à aucune base de données.
- Avant le démarrage d'une instance, Oracle consulte le fichier init afin de déterminer les caractéristiques de l'instance à créer.

## Chargement d'une base

- Le chargement d'une base de données consiste à l'associer à une instance déjà démarrée.
- La base de données devient accessible uniquement aux utilisateurs ayant le privilège DBA.
- L'administrateur peut ainsi effectuer les opérations suivantes :
  - Renommer les fichiers de données,
  - Ajouter, supprimer ou renommer des fichiers de reprise (Redo Log),
  - Activer ou désactiver l'option d'archivage des fichiers de reprise,
  - Effectuer un recouvrement total de la base de données.

## Chargement d'une base (suite)

- Ces opérations peuvent être effectuées à l'aide de la commande ALTER DATABASE.
- Lorsque la base est chargée, Oracle effectue les opérations suivantes :
  - ouverture du fichier de contrôle
  - vérification que les fichiers de données et de reprise qui sont indiqués dans le fichier de contrôle existent, sans les ouvrir.
- S'il ne trouve pas l'un de ces fichiers, il signale un message d'erreur.

## Chargement d'une base (suite)

- Une base peut être chargée en :
  - **Mode exclusif** : aucune autre instance ne peut charger cette base.
  - **Mode partagé** (dit aussi parallèle) : d'autres instances peuvent charger la même base en mode partagé.

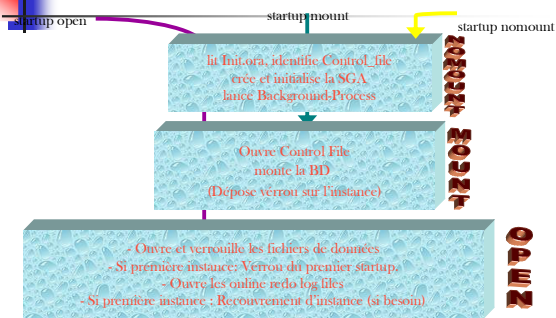
## Ouverture d'une base

- L'ouverture d'une base de données consiste à la rendre accessible à tous les utilisateurs.
- Les fichiers de données et de reprise sont ouverts.
- Si l'un de ces fichiers ne peut pas être ouvert, une erreur est signalée et la base n'est pas ouverte.
- Oracle vérifie qu'elle a été normalement fermée. Si elle a été fermée en avortant une instance, ou suite à une panne, Il effectue automatiquement un recouvrement (reprise).

## Commande de démarrage

```
STARTUP [FORCE] [RESTRICT] [PFILE=nom_fichier_param]
        {[OPEN] [RECOVER] [nom_base] | [MOUNT]} [options]}
[NOMOUNT];
```

## Startup

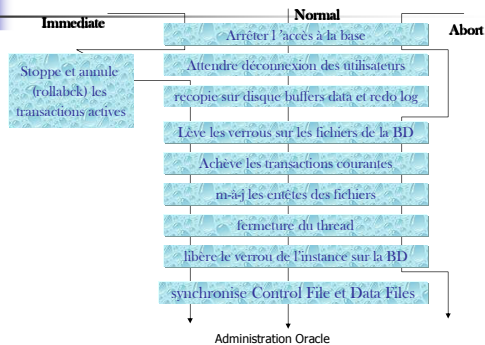


## Fermeture d'une base

- La fermeture d'une base de données se fait, comme l'ouverture, en trois étapes :
  - fermeture de la base,
  - déchargement de la base,
  - arrêt de l'instance.
- Elle se fait suite à une connexion avec le privilège INTERNAL (CONNECT INTERNAL).

```
SHUTDOWN [ABORT | IMMEDIATE | NORMAL];
```

## Shutdown



115

## Le dictionnaire de données

## Présentation générale

- Le dictionnaire de données d'Oracle est un ensemble de tables et de vues.
- Table du dictionnaire :
  - enregistrement des informations relatives aux objets de la base
  - accessibles et maintenues dynamiquement par le noyau
  - accès direct à ces tables fortement déconseillé aux utilisateurs
  - contenu souvent crypté pour des raisons de sécurité
- Vues du dictionnaire :
  - protègent le contenu des tables du dictionnaire tout en rendant accessibles les informations qu'elles contiennent.

Administration Oracle

117

## Vues du dictionnaire

- Répartition en quatre classes :
  - Vues relatives aux objets d'un utilisateur (USER\_vue)
  - Vues relatives aux objets accessibles à un utilisateur (ALL\_vue)
  - Vues relatives aux administrateurs (DBA\_vue)
  - Vues relatives aux suivi des performances (V\$vue)
- La vue **DICTIONARY** permet d'avoir une liste complète des tables, vues et synonymes du dictionnaire.

Administration Oracle

118

## Création du dictionnaire

- Lors de la création d'une base de données, le paramètre `INIT_SQL_FILES` permet de spécifier les fichiers SQL qui seront exécutés après pour construire le dictionnaire de données et d'autres fichiers initiaux :
 

```
INIT_SQL_FILES = (fichier1, fichier2,...)
```
- Par défaut, Oracle exécute les deux fichiers :
  - sql.bsq
  - catalog.sql

Administration Oracle

119

## Scripts sql.bsq et catalog.sql

- Le script **sql.bsq** doit être spécifié en premier :
  - crée un tablespace **SYSTEM** et un rollback segment **SYSTEM**
  - crée les tables du dictionnaire dans **SYSTEM**
  - charge quelques tables du dictionnaire.
- Le script **catalog.sql** :
  - crée les vues du dictionnaire
  - crée les synonymes publics associés à ces vues
  - attribue un accès **PUBLIC** à ces synonymes

Administration Oracle

120

## Scripts supplémentaires

- **audit.sql** : crée les vues donnant la trace de l'activité du système.
- **blocking.sql** : crée la vue **BLOCKING\_LOCKS** qui donne la liste des verrouillages bloquants.
- **catalog6.sql** : crée des anciennes vues de la version 6.
- **dba\_synonyms.sql** : crée des synonymes des vues de l'administrateur.
- **dropcat6.sql** : supprime les vues de la version 6.
- **expview.sql** : crée des vues utilisées lors de l'import et l'export des données.
- **monitor.sql** : attribue à **PUBLIC** les droits d'accès aux vues relatives aux performances.

## Utilisation du dictionnaire

- Le dictionnaire de données est utilisé par :
  - Oracle pour assurer les différentes opérations de la base.
  - L'administrateur comme outil de base pour gérer les objets de la base.
  - Les utilisateurs à travers des vues pour une vision globale des objets leur appartenant ou pour lesquels ils ont acquis des droits.

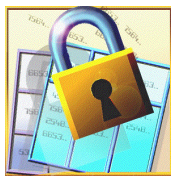
## Accès au dictionnaire par Oracle

- Oracle accède au dictionnaire en lecture et en écriture.
- Il est le seul à pouvoir modifier son contenu
- Accès en lecture pour l'analyse des commandes LMD soumises par les utilisateurs.
- Accès en écriture lors de l'exécution des commandes LDD. Les tables d'audit sont mises à jour en permanence.
- Une bonne partie du dictionnaire est chargé dans la **SGA**.

## Accès au dictionnaire par les utilisateurs

- Les utilisateurs accèdent au dictionnaire uniquement à travers les vues.
- Le dictionnaire peut être utilisé pour deux aspects
  - aspect de documentation
  - aspect d'administration
- Les vues du dictionnaire sont accessibles à travers le langage **SQL**.
- Toutes les variantes de la commande **SELECT** peuvent être utilisées.

## Sécurité BD



## Introduction

Assurer la sécurité d'une base de données consiste à:

\* Empêcher l'accès, la modification et la destruction des données par des accès non autorisés tel que la malveillance ou les inconsistances accidentelles.

\* Spécifier les autorisations, c'est à dire les règles qui permettent de définir qui a le droit d'effectuer un tel type d'opération sur telles données. Elle sont généralement réservées à l'administrateur de la BD.

## Sécurité et Intégrité

Sécuriser une BD ➡ Pourquoi sécurise-t-on la base ? (Connaître les causes de violation de la sécurité)

### 1-Causes de la violation de la sécurité:

- Crashes pendant le traitement des transactions (les interruptions système ou bien matériels).
- Anomalies dues à la répartition des données sur plusieurs sites (ce problème apparaît dans l'approche BD répartie).

## Causes de la violation (Suite)

- Erreurs logiques contradictoires avec l'hypothèse de conservation de la consistance de la base par des transactions qui s'y déroulent lors de l'échange des données entre plusieurs utilisateurs. C'est à dire la perte de consistance lors de l'échange des données entre différents utilisateurs ou transactions

- La malveillance des utilisateurs c'est à dire la mauvaise utilisation des données d'une façon intentionnelle (modification, destruction,...).

## Sécurité et intégrité (Suite)

Pour protéger une BD, des mesures de sécurité doivent être prises sur plusieurs niveaux:

### 2- Niveaux de sécurité:

**\*Physique:** Les sites qui hébergent les données et les SGBDs doivent être physiquement armés contre les intrusions en force.

**\*Humain:** Les autorisations doivent être accordées d'une façon sélective afin qu'un utilisateur ne cède ses autorisations à une personne malveillante.

## Niveaux de sécurité (Suite)

**\*Système:** Quelle que soit la sûreté de la BD des faiblesses éventuelles dans la sécurité de systèmes de gestion de base de données peuvent être mises à profit pour pénétrer la base. Donc, la sécurité du logiciel système est aussi importante que la sécurité physique de la base.

**\*Base de données:** Le système doit s'assurer que les restrictions des accès ne sont pas violées.

## Sécurité et intégrité (Suite)

### 3- Contraintes d'intégrité:

Les contraintes d'intégrité sont des règles par rapport à elles se fait le contrôle de validité sur les opérations effectuées.

Ils existent plusieurs types de contraintes d'intégrité (temporelles, référentielles ,etc.)

Les contraintes d'intégrité protègent la base contre une perte de consistance lors des modifications effectuées par les utilisateurs autorisés.

## Niveau SGBD(système)

### 1- Niveaux de sécurité:

Les systèmes de gestion de bases de données assurent leur sécurité en deux niveaux:

#### \*Niveau des relations:

Un utilisateur peut être autorisé ou non à une relation donnée.

#### \*Niveau des vues:

Un utilisateur peut être autorisé ou non à accéder aux données qui font partie d'une vue donnée.

## Niveau SGBD(système)

Une vue cache les données qu'un utilisateur n'a pas besoin de les voir, donc, elle facilite l'accès à la base et renforce sa sécurité.

Les raisons d'utiliser la vue comme une approche de protection sont les suivantes :

- Elle est mieux adaptée pour restreindre l'accès à la base de données.
- Sa définition est statique mais les données auxquelles on accède à travers la vue sont dynamiques.
- Elle est adaptée pour implémenter les niveaux dans une multibase de données.

Administration Oracle

133

## Niveau SGBD(système)

### 2-Autorisations pour les vues et les relations:

Un utilisateur peut bénéficier de plusieurs habilitations lors de son accès à la base. Parmi celles-ci notons:

- L'autorisation de lecture ,sans avoir le droit de modifier des données (**READ**).
- L'autorisation d'insertion (**INSERT**).
- L'autorisation de mise à jour, sans avoir le droit de modifier des données (**UPDATE**).
- L'autorisation d'effacement des données (**DELETE**).

Ces autorisations font partie de langage de manipulation des données.

Administration Oracle

134

## Niveau SGBD(système)

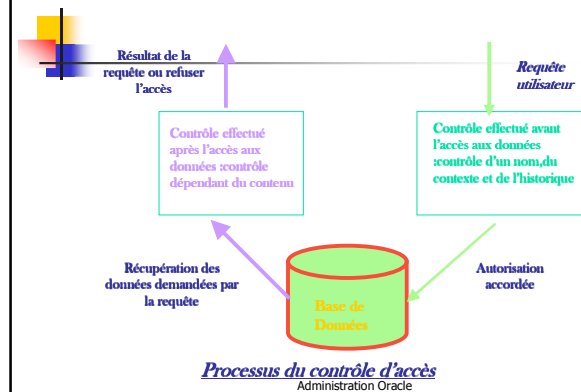
Il y a aussi celles qui font partie de langage de définition des données.

- L'autorisation d'accès aux index (pour création et effacement des index)
- L'autorisation d'accès aux relations ,pour créer des nouvelles relations (**RESOURCE**).
- L'autorisation d'altération des relations ,par ajout ou élimination des attributs (**ALTERATION**).
- L'autorisation d'abandon, pour effacer des relations (**DROP**).

Administration Oracle

135

## Niveau SGBD(système)



Administration Oracle

136

## Sécurité et Confidentialité

La sécurité et la confidentialité consiste, surtout, à définir les privilèges d'accès.

### \*Affectation de privilège:

Quand un utilisateur crée une relation de base, il dispose de tous les droits sur cette relation et de même pour les vues.

L'utilisateur qui donne les privilèges est appelé **donneur** et celui qui les reçoit est appelé **receveur**.

Administration Oracle

137

## Sécurité et Confidentialité (Suite)

La liste des privilèges qu'un utilisateur peut accorder :

- Sur les tables, vues et snapshot qu'il possède:
  - INSERT;
  - UPDATE (tout ou des colonnes spécifiques);
  - DELETE;
  - SELECT;

Les privilèges INSERT, UPDATE et DELETE peuvent être accorder sur des snapshot uniquement s'ils sont actualisables.

Administration Oracle

138

## Sécurité et confidentialité (Suite)

- Sur les tables qu'il possède:
  - ALTER (une table entière ou des colonnes spécifiques ou une séquence);
  - REFERENCES;
  - INDEX (colonnes d'une table);
  - ALL (tous les privilèges précités);

## Sécurité de données

- La sécurité de données est une tâche de l'administrateur de la base qui est chargé de la mettre en place au niveau :
  - système d'exploitation
  - objets de la base de données
  - utilisateurs de la base

## Niveau Système d'exploitation

- Les programmes d'Oracle sont divisés en deux ensembles de sécurité :
  - les programmes exécutés par tous (*others*)
  - les programmes exécutés par DBA
- L'approche typique de sécurité est la suivante :
  - créer un groupe dba avant l'installation
  - lui affecter *root* et le propriétaire *oracle* (programmes exécutés par dba auront la permission 710).

## Niveau Système d'exploitation (Suite)

- ajouter un autre groupe oracle pour autoriser l'utilisation de certains outils (SQL\*Plus etc.)
- mettre ces outils avec l'ID du groupe créé et changer la permission en 710.
- les autres programmes restent avec la permission 711 et peuvent être exécutés par others.
  - mettre les programmes \$ORACLE\_HOME/bin avec le propriétaire oracle.
  - mettre les outils (sqlplus, imp, exp) à la protection 711
  - mettre les outils DBA (Server Manager) à la protection 700

## Rôles Oracle et Groupes Unix

- Une autre méthode de sécurité consiste à faire une correspondance rôles Oracle avec les groupes Unix pour les bases locales en respectant le format

*ora\_sid\_rôle[\_d|a]*

*sid* :           identifiant de la base  
*rôle* :          nom du rôle Oracle  
*d* :            rôle par défaut (optionnel)  
*a* :            rôle avec option WITH ADMIN

### Exemple de fichier /etc/group :

```
ora_test_osoper_d:NONE:1:jean, mary,scott
ora_test_osdba_a:NONE:3:malek
ora_test_role1:NONE:bob,marie
```

## Utilisateurs Oracle

- Deux super-utilisateurs sont créés dans une base de données :
  - sys/change\_on\_install
  - system/manager
- Penser à les changer sur une base de production :

```
ALTER USER SYS
IDENTIFIED BY nouveau_pwd;

ALTER USER SYSTEM
IDENTIFIED BY nouveau_pwd;
```



## Gestion des utilisateurs

- Les utilisateurs sont gérés dans Oracle comme tous les objets de la base :
  - action de création
  - action de modification
  - action de suppression
- Chaque utilisateur peut avoir des rôles, profils, des tablespaces par défaut etc.

## Création d'un utilisateur

- La commande de création d'utilisateur est :

```
CREATE USER utilisateur
  IDENTIFIED ( BY nouveau_pwd | EXTERNALLY)
  [DEFAULT TABLESPACE tbs]
  [TEMPORARY TABLESPACE tbs]
  [QUOTA {entier [K|M] | UNLIMITED} ON tbs]
  PROFILE profil;
```

- Cette commande possède un domaine de privilèges utilisateurs vide.

## Modification d'un utilisateur

- La modification d'un utilisateur se fait par :

```
ALTER USER utilisateur
  IDENTIFIED ( BY nouveau_pwd | EXTERNALLY)
  [DEFAULT TABLESPACE tbs]
  [TEMPORARY TABLESPACE tbs]
  [QUOTA {entier [K|M] | UNLIMITED} ON tbs]
  PROFILE profil
  [DEFAULT RÔLE {rôle,... | [ALL][EXCEPT rôle,...] | NONE];
```

## Suppression d'un utilisateur

- La commande de suppression d'un utilisateur est :

```
DROP USER utilisateur [CASCADE];
```

- L'option CASCADE permet de supprimer l'utilisateur et tous ses objets (tables, vues, contraintes etc.)
- En l'absence de CASCADE, l'utilisateur ne peut pas être supprimé s'il possède des objets.
- Cette option doit être manipulée avec prudence.

## Notion de schéma

- Un schéma est une structure logique de stockage de données qui est associé à un utilisateur de la base.
- Un schéma contient des ordres de création de :
  - tables
  - index
  - vues
  - clusters
  - octroi de privilèges

## Notion de schéma

- **Avantages :**

- Oracle gère la création du schéma comme une transaction.
- Si un élément du schéma échoue, tous les éléments sont annulés.

- **Inconvénients**

- un seul schéma par utilisateur
- le nom du schéma est forcément celui de l'utilisateur
- pas de commande de suppression ou de modification de schéma

## Création d'un schéma

- La commande de création d'un schéma est la suivante :

```
CREATE SCHEMA AUTHORIZATION schéma
{
    commande CREATE TABLE...
    commande CREATE VIEW...
    commande GRANT
}
```

## Les profils

- Un profil est un ensemble de limites ressources identifié par un nom.
- Utilisé pour contrôler la consommation des ressources.
- Permet une gestion plus facile des ressources.
- Existence d'un profil **DEFAULT** assigné automatiquement aux utilisateurs qui n'ont pas de profil.
- Deux niveaux de profil : SESSION, APPEL

## Profil au niveau session

- Permet de contrôler :
  - le nombre de sessions autorisées pour chaque utilisateur.
  - le temps CPU pour chaque session
  - le temps écoulé par session
  - le temps écoulé sans occupation
  - le nombre maximum de blocs lus par session
- La session s'arrête avec annulation des transactions non validées si dépassement de limites.

## Profil au niveau appel

- Permet de contrôler :
  - le nombre maximum de blocs lus par appel
  - le temps CPU maximum par appel
- Un ordre SQL peut faire plusieurs appels (FETCH).
- L'ordre SQL s'arrête avec annulation si dépassement des limites.

## Création de profil

```
CREATE PROFILE nom_profil LIMIT
[SESSIONS_PER_USER           {entier/UNLIMITED/DEFAULT}]
[CPU_PER_SESSION             {entier/UNLIMITED/DEFAULT}]
[CPU_PER_CALL                 {entier/UNLIMITED/DEFAULT}]
[CONNECT_TIME                 {entier/UNLIMITED/DEFAULT}]
[IDLE_TIME                     {entier/UNLIMITED/DEFAULT}]
[LOGICAL_READS_PER_CALL       {entier/UNLIMITED/DEFAULT}]
[LOGICAL_READS_PER_SESSIONS   {entier/UNLIMITED/DEFAULT}]
[PRIVATE_SGA                  {entier[K|M]/UNLIMITED/DEFAULT}]
[COMPOSITE_LIMIT               {entier/UNLIMITED/DEFAULT}]
```

## Création d'un profil

- Les limites doivent être différentes de 0.

**SESSIONS\_PER\_USER** : nombre de sessions concurrentes par utilisateur  
**CPU\_PER\_SESSION** : temps max CPU autorisé par session (en 1/100 s)  
**CPU\_PER\_CALL** : temps max CPU autorisé par appel (en 1/100 s)  
**CONNECT\_TIME** : temps max écoulé par session (en minutes)  
**IDLE\_TIME** : temps max autorisé sans activité (en minutes)  
**LOGICAL\_READS\_PER\_SESSION** : nombre de blocs max autorisé en lecture par session.  
**LOGICAL\_READS\_PER\_CALL** : nombre de blocs max autorisé en lecture par appel.  
**PRIVATE\_SGA** : taille maximale privée dans la SGA.

## COMPOSITE\_LIMIT

- Définit un coût global qui est la somme des poids de chacune des ressources suivantes :
  - CPU\_PER\_SESSION
  - CONNECT\_TIME
  - LOGICAL\_READS\_PER\_SESSION
  - PRIVATE\_SGA
- Il se calcule par la formule suivante :
$$T = \text{somme (poids * ressource)}$$

## Calcul du composite\_limit

- La commande SQL suivante :

```
ALTER RESOURCE COST
[CPU_PER_SESSION entier ]
[CONNECT_TIME entier ]
[LOGICAL_READS_PER_SESSION entier ]
[PRIVATE_SGA. entier]
```
- Le poids déterminé est utilisé par COMPOSITE\_LIMIT du profil.
- Le poids total est exprimé en unités de services.

## Exemple

```
ALTER RESOURCE COST
CPU_PER_SESSION 150
CONNECT_TIME 2;
```

La formule de calcul de ce coût est la suivante :

- $T = 150 * CPU + 2 * CON$
- CPU est le temps CPU utilisé par la session (en 1/100 s)
- CON est le temps écoulé pour la session en minute.
- T est le total des ressources exprimé en unités de services.

## Modification d'un profil

```
ALTER PROFILE nom_profil LIMIT
[SESSIONS_PER_USER {entier|UNLIMITED/DEFAULT}]
[CPU_PER_SESSION {entier|UNLIMITED/DEFAULT}]
[CPU_PER_CALL {entier|UNLIMITED/DEFAULT}]
[CONNECT_TIME {entier|UNLIMITED/DEFAULT}]
[IDLE_TIME {entier|UNLIMITED/DEFAULT}]
[LOGICAL_READS_PER_CALL {entier|UNLIMITED/DEFAULT}]
[LOGICAL_READS_PER_SESSIONS {entier|UNLIMITED/DEFAULT}]
[PRIVATE_SGA {entier[K|M]|UNLIMITED/DEFAULT}]
[COMPOSITE_LIMIT {entier|UNLIMITED/DEFAULT}]
```

## Suppression d'un profil

- La suppression d'un profil se fait par :

```
DROP PROFILE profil [CASCADE];
```
- L'option CASCADE permet de :
  - supprimer ce profil pour tous les utilisateurs auxquels il a été attaché.
  - leur affecter le profil DEFAULT

## Activation des ressources limites

- Au démarrage d'une instance, Oracle active ou désactive les limites ressources utilisées par les profils.
- Deux méthodes :
  - dans le fichier init.ora, mettre le paramètre RESSOURCE\_LIMIT = TRUE ou FALSE
  - commande SQL :

```
ALTER SYSTEM
SET RESSOURCE_LIMIT = {TRUE|FALSE}
```

## Privèges d'accès

- Oracle offre une multitude de privilèges d'accès aux données pour :
  - augmenter la sécurité des données
  - mieux gérer les ressources
  - mieux gérer les utilisateurs
  - regrouper les privilèges (rôles)
- On distingue deux familles de privilèges :
  - privilèges **systèmes**
  - privilèges **objets**

## Privèges Systèmes

- Un privilège système permet d'exécuter certaines actions touchant à la structure de la base de données (table, trigger, profil,...).
 

```
GRANT [privilège_système | rôle],...
TO {utilisateur | rôle | PUBLIC},...
[WITH ADMIN OPTION]
```
- Un rôle regroupe plusieurs privilèges pour :
  - faciliter la formulation de requête
  - structurer les privilèges des bases et des applications

## Privèges systèmes (Suite)

	A	C	D		A	C	D
ANY CLUSTER	X	X	X	CLUSTER			X
ANY INDEX	X	X	X	PROCEDURE			X
ANY PROCEDURE	X	X	X	ROLE			X
ANY ROLE	X		X	SEQUENCE			X
ANY SEQUENCE	X	X	X	SNAPSHOT			X
ANY SNAPSHOT	X	X	X	TABLE			X
ANY TABLE	X	X	X	TRIGGER			X
ANY TRIGGER	X	X	X	DATABASE		X	
PROFILE	X	X	X	RESOURCE COST		X	
ROLLBACK SEGMENT	X	X	X	SESSION		X	X
SYSTEM	X		X	TABSPACE		X	X
USER	X	X	X	VIEW			X
ANY VIEW		X	X	ANY SYNONYM			X
SYNONYM		X		DATABASE LINK			X
PUBLIC DATABASE LINK		X		PUBLIC SYNONYM			X

## Privèges systèmes (suite)

ANALYSE ANY	AUDIT ANY
AUDIT SYSTEM	BECOME USER
BACKUP ANY TABLE	COMMENT ANY TABLE
EXECUTE ANY PROCEDURE	FORCE ANY TRANSACTION
FORCE TRANSACTION	GRANT ANY PRIVILEGE
GRANT ANY ROLE	INSERT ANY TABLE
LOCK ANY TABLE	MANAGE TABLESPACE
READUP	RESTRICTED SESSION
SELECT ANY SEQUENCE	SELECT ANY TABLE
UNLIMITED TABLESPACE	UPDATE ANY TABLE
WRITE DOWN	WRITEUP

## Privèges Objets

- Un privilège objet donne le droit d'exécuter une action particulière sur :
  - TABLE,
  - VUE,
  - FONCTION,
  - PROCÉDURE,
  - SÉQUENCE
  - PACKAGE.
- ces droits peuvent être attribués à un ou plusieurs utilisateurs ou rôles dans la même commande.

## Commande GRANT

- La commande d'octroi de privilèges objets est la suivante :
 

```
GRANT privilège_objet [ (colonne ,...) ] ,...
ON [schéma.]objet
TO {utilisateur | rôle | PUBLIC},...
[WITH GRANT OPTION]
```

## Privilèges Objets

- Les privilèges objets sont :

	Table	Vue	Sequence Fonction Package	Procédure	Snapshot
ALTER	X		X		
DELETE	X	X			
EXECUTE				X	
INDEX	X				
INSERT	X	X			
REFERENCES	X				
SELECT	X	X	X		X
UPDATE	X	X			

## Suppression de privilèges

- Privilèges systèmes

```
REVOKE (privilege_système | rôle) ,...
FROM (utilisateur| rôle| PUBLIC),...
```

- Privilèges objets

```
REVOKE privilege_objet ,...
ON [schéma] objet
FROM (utilisateur| rôle| PUBLIC) ,...
[CASCADE CONSTRAINTS]
```

- L'option **CASCADE CONSTRAINTS** supprime toutes les contraintes d'intégrité référentielles définies sur les objets pour lesquels on retire les privilèges.

## Les Rôles

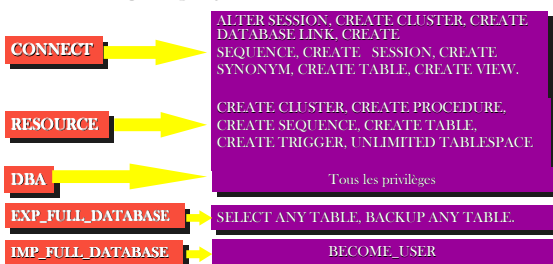
- Un rôle est une agrégation de droits d'accès aux données et de privilèges système.
- Un rôle renforce la sécurité de données et réduit significativement la difficulté et le coût de son administration.
- Un rôle peut être :
  - créé
  - modifié
  - supprimé
  - activé ou désactivé

## Les rôles (Suite)

- Oracle crée automatiquement des rôles qu'il attribue à SYS et SYSTEM.
- Ces rôles sont des privilèges système des anciennes versions :
  - CONNECT.
  - RESOURCE
  - DBA.
- L'importation et l'exportation de données ont été mises dans les rôles respectifs :
  - IMP\_FULL\_DATABASE
  - EXP\_FULL\_DATABASE

## Les privilèges Système

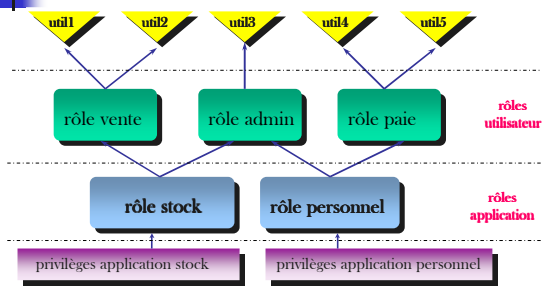
Les privilèges système attribués à ces rôles sont



## Les rôles (suite)

- Un rôle peut servir deux aspects :
  - APPLICATION
  - UTILISATEUR.
- Un rôle d'application regroupe tous les privilèges nécessaires à l'exécution d'une application.
- Il peut être composé d'autres rôles, chacun d'eux offre un ensemble de privilèges d'accès à des données différentes de l'application.
- Un rôle utilisateur gère des privilèges communs nécessaires aux utilisateurs de la base, comme la création de session.

## Exemple de rôles



Administration Oracle

175

## Création de rôle

```
CREATE ROLE rôle
[NOT IDENTIFIED
| IDENTIFIED {BY mot_de_passe | EXTERNALLY}]
```

- Un rôle est créé initialement sans aucune caractéristique.
- Il est alimenté par des ordres SQL GRANT.
- Il peut être protégé par un mot de passe.
- Il peut être activé et désactivé.

Administration Oracle

176

## Activation d'un rôle

- Pour qu'un rôle devienne disponible à l'utilisateur de la session, il faut qu'il soit activé.
- ```
SET ROLE {rôle [IDENTIFIED BY mot_de_passe]
[, rôle [IDENTIFIED BY mot_de_passe] ]...
| ALL [EXCEPT rôle [, rôle] ...]
| NONE }
```
- Oracle établit au moment de la connexion tous les privilèges d'un utilisateur.
  - Ces privilèges ont été attribués soit :
    - directement par des GRANT OBJETS,
    - indirectement par des GRANT RÔLES.

Administration Oracle

177

## Activation d'un rôle

- L'utilisation des options **ALL** et **ALL EXCEPT** nécessitent que tous les rôles ne possèdent pas de mot de passe ou utilisent une authentification par OS.
- Le rôle qui ne figure pas dans la liste est considéré comme désactivé.
- Il n'existe pas de commande spécifique pour désactiver explicitement un rôle.

Administration Oracle

178

## Modification d'un rôle

- La commande de modification de rôle est :  

```
ALTER ROLE rôle
{ NOT IDENTIFIED
| IDENTIFIED {BY mot_de_passe | EXTERNALLY}}
```
- Exemple :**  
ALTER ROLE rôle1 NOT IDENTIFIED;

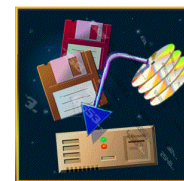
- La commande de suppression de rôle est :  

```
DROP ROLE rôle
```

Administration Oracle

179

## Outils d'administration



## Outils d'administration

- Server Manager : administrer la base
- Import / Export : échanger les données avec l'extérieur dans un format Oracle
- SQL\*Loader : échanger les données avec l'extérieur dans un format à définir
- Outil de vérification de la base : db\_verify
- Outil NLS
- Oracle Enterprise Manager

Administration Oracle

181

## Import/Export

- Deux outils distincts pour échange de données entre des base Oracle.
- Export crée une copie logique d'une partie ou de la totalité de la base pour être utilisée comme :
  - sauvegarde logique
  - copie vers d'autres bases Oracle
- Import intègre dans la base les données des fichiers générés par Export.



Administration Oracle

182

## Export

- Export permet de créer une copie logique d'une partie ou de la totalité des objets d'une base de données. Ces objets peuvent être :
  - structure des tables,
  - données des tables,
  - privilèges attribués à des utilisateurs,
  - vues,
  - clusters,
  - synonymes,
  - séquences,
  - contraintes d'intégrités.

Administration Oracle

183

## Modes d'Export

- **Mode table** : exportation explicite de tables avec droits, index et contraintes d'intégrité.
- **Mode Utilisateur** : exportation de tous les objets d'un utilisateur explicite (tables, vues, synonymes, etc.).
- **Mode base de données** : exportation de toute la base par un utilisateur ayant les droits DBA. Subdivisé en cinq modes :

Administration Oracle

184

## Modes d'Export (suite)

- **entiere** : tous les objets de la base y compris ceux de SYS
- **full** : tous les objets de la base sauf ceux de SYS
- **cumulative** : tous les objets qui ont été modifiés depuis le dernier Export cumulative ou complète
- **incremental** : tous les objets qui ont été modifiés depuis le dernier Export incrémental, cumulative ou complète.
- **complete** : tous les objets de la base sans ceux de SYS avec remise à zéro des tables systèmes qui gèrent le mode incrémental et cumulative (SYS.INCVID et SYS.INCEXP).

Administration Oracle

185

## Exemple

- Export en mode complet (T) une fois toutes les quatre semaines.
- Export en mode cumulative (C) à la fin de chaque semaine.
- Export en mode incrémental (I) au fin de chaque jour.

| 1re semaine   | 2ème semaine  | 3ème semaine  | 4ème semaine  |
|---------------|---------------|---------------|---------------|
| 1 2 3 4 5 6 7 | 1 2 3 4 5 6 7 | 1 2 3 4 5 6 7 | 1 2 3 4 5 6 7 |
| T I I I I I C | I I I I I I C | I I I I I I C | I I I I I I C |

Administration Oracle

186

## Lancement de Export

exp help-y  
Vous pouvez laisser l'Export vous demander les param en tapant EXP Commande suivie de votre nom d'utilisateur/mot de passe:

Exemple: EXP SCOTT/TIGER

Ou pour contr\_ler l'ex\_cution de l'export, entrer la commande EXP suivie de divers arguments. Pour sp\_cifier les param, utiliser des mots-cl\_s:

Format: EXP KEYWORD=valeur ou KEYWORD=(valeur1,valeur2,...,valeurN)  
Exemple: EXP SCOTT/TIGER GRANTS=Y TABLES=(EMP,DEPT,MGR)

Mot-cl\_ Description (D\_faut) Mot-cl\_ Description (D\_faut)

USERID nom util/mot de passe FULL exporter l'int\_gralit\_ du fich (N)  
BUFFER taille m\_moire tampon donn\_es OWNER liste noms util. propri\_t.  
FILE fich sortie (EXPDAT.DMP) TABLES liste de noms de tables  
COMPRESS importer dans une extent.(Y) RECORDLENGTH longueur enreg. ES  
GRANTS privil\_ges d'export (N) INCTYPE type d'export incr\_mentiel  
INDEXES index d'export (Y) RECORD recherche export incr. (Y)  
ROWS lignes donn\_es export. (Y) PARFILE nom de fich param\_trs  
CONSTRAINTS contraintes export (Y) ~~ADMINISTRATOR~~ Oracle table r\_f\_rence  
LOG fich\_journal sortie\_crau STATISTICS analyse des objets (ESTIMATE)

187

## Import

- Charge dans une base la totalité ou une partie d'un fichier export.
- Trois modes :
  - mode table
  - mode utilisateur
  - mode base de données : deux variantes
    - mode «system» : import seulement des modifications du dictionnaire de données.
    - mode «restore» : import des objets ayant été modifiés depuis la dernière exportation incrémentale, cumulative ou complète.

Administration Oracle

188

## Lancement de Import

Ou pour contrôler l'exécution de l'import, taper la commande IMP suivie de divers arguments. Pour spécifier param, utiliser des mots-cl\_s:

Format: IMP KEYWORD=valeur ou KEYWORD=(valeur1,valeur2,...,valeurN)  
Exemple: IMP SCOTT/TIGER IGNORE=Y TABLES=(EMP,DEPT) FULL=N

Mot-cl\_ Description (D\_faut) Mot-cl\_ Description (D\_faut)

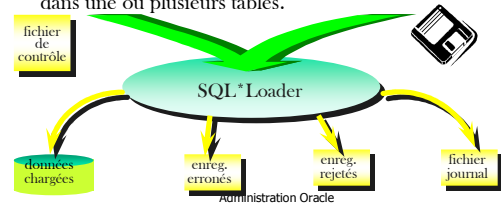
USERID nom util/mot de passe FULL importer tout le fichier(N)  
BUFFER taille m\_moire tampon donn\_es FROMUSER liste noms util. propri\_t.  
FILE fich sortie (EXPDAT.DMP) TOUSER liste de noms util.  
SHOW lister uniq contenu fich(N) TABLES liste de noms de tables  
IGNORE ignore erreurs de cr\_at (Y) RECORDLENGTH long d'enregis ES  
GRANTS import des privil\_ges (Y) INCTYPE type import incr\_mentiel  
INDEXES import d'index (Y) COMMIT valid. insert tableaux(N)  
ROWS import des lignes de donn\_es (Y) PARFILE nom fich. de param\_trs  
LOG fich\_journal de sortie\_crau  
DESTROY espace de stockage de remplacement du fich de donn\_es (N)  
INDEXFILE\_cra les info. table/index dans fichier sp\_cif\_  
CHARSET jeu de caract\_res de fichier d'export (NLS\_LANG)

Administration Oracle

189

## SQL\*Loader

- Permet de charger des données provenant de fichiers non Oracle.
- Les données peuvent provenir d'une ou de plusieurs fichiers de formats identiques ou différents et chargées dans une ou plusieurs tables.



Administration Oracle

190

## Fichier de contrôle

- décrit les données à charger ainsi que leur destination dans la base de données :
  - nom et structure des fichiers de données
  - liste de tables cibles et éventuellement les colonnes utilisées
  - correspondance entre les enregistrements des fichiers de données et les colonnes.

Administration Oracle

191

## Fichier de données

- contient les données à charger dans la base de données.
- peut être de format binaire ou de format caractère
- les enregistrements peuvent être de format fixe ou variable.

Administration Oracle

192



## Fichier journal

- compte rendu de l'exécution de chargement :
  - section en-tête
  - section informations globales
  - section informations sur les tables
  - section informations sur les fichiers de données
  - section informations sur les tables chargées
  - section de statistiques

## Enregistrements rejetés et erronés

- Fichier enregistrements erronés :
  - contient tous les enregistrements qui n'ont pas pu être interprétés et chargés dans la base de données.
- Fichier enregistrements rejetés :
  - contient tous les enregistrements qui n'ont pas pu être chargés (ne vérifient pas la condition de la clause WHEN).
  - créé uniquement à la demande de l'utilisateur

## Lancement de SQL\*Loader

**SQLLOAD option=valeur [option=valeur ...]**

- Liste des options
  - USERID nom\_utilisateur/mot\_de\_passe
  - CONTROL fichier de contrôle
  - LOG fichier journal
  - BAD fichier de données erronées
  - DATA fichier de données
  - DISCARD fichier de rejet
  - DISCARDMAX nombre max d'enregistrements rejetés
  - SKIP nombre d'enregistrements à ignorer
  - LOAD nombre d'enregistrements à charger
  - ERRORS nombre d'erreurs autorisées
  - SILENT pas de messages pendant l'exécution
  - PARFILE fichier de paramètres