Administration Oracle 10G
Partie I

G. Mopolo-Moké
prof. MBDS / UNSA NICE
2005/ 2006

Plan Général

Plan Général

Plan Général

Plan Général

Plan Général

Plan Général

1. Introduction

Plan

1.1 Objectifs du cours

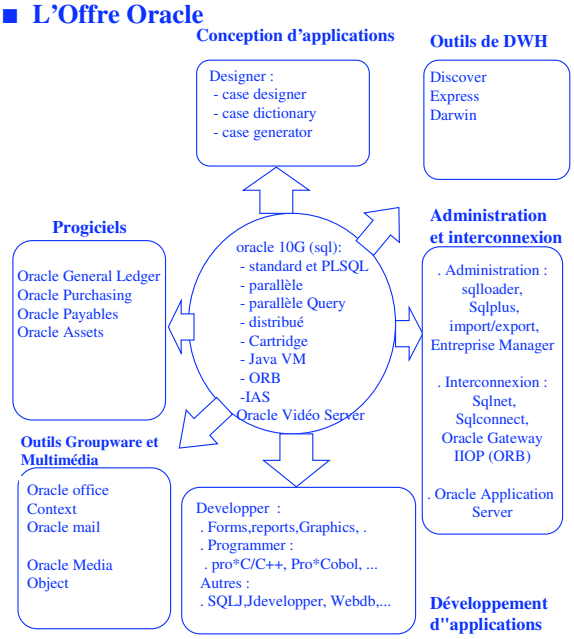
- Comprendre le fonctionnement du moteur Oracle 10G
- Comprendre l'architecture logique et physique d'une BD Oracle
- Effectuer les tâches d'administration de base
- Administrer la sécurité (utilisateurs, rôles, profiles,...)
- Gérer l'intégrité de données
- Gérer les procédures stockées et les alertes
- Manipuler et administrer des données réparties
- Faire une introduction à l'optimisation d'Oracle

1.2 Les tâches de l'administrateur

- installer et upgrader Oracle
- allouer des espaces systèmes et planifier des besoins futurs
- créer des objets des schémas
- gérer la sécurité et les ressources
- planifier l'archivage de données
- sauvegarder et restaurer les données
- arrêter et démarrer la base
- contrôler et optimiser les performances

1.3 L'offre Oracle

- L'Offre Oracle



1.4 Oracle 10G Présentation Générale

- Le noyau de base d'Oracle 10G
  - Moteur SQL
  - Moteur PL/SQL (option procédurale intégrée depuis la version 7.1), Moteur Java
  - Verrouillage en ligne
- Les Versions d'Oracle
  - Version Oracle 7.X.X.X.X, 8.X.X.X.X
  - Version Oracle 10.0.1.1.1
  - Exemple
    10.0.1.1.1.1 => Version 10, New Features Realese number 0, Maintenance release Number 1, Patch release number 1, Generic patch set number 1, plateform specific patch set number 1
    - SELECT * FROM PRODUCT_COMPONENT_VERSION ou V$version

1.4 Oracle 10G Présentation Générale

Les Options d'Oracle

- Oracle cluster
  - accès à une même base par des machines en cluster
  - accès par plusieurs instances aux données d'une même base
  - pas de partage de la mémoire central
- Option parallèle (Oracle Parallèle Query : Oracle PQ)
  - exécution d'une même requête par plusieurs processeurs
- Oracle Replication
- Option distribuée
  - Interrogation distribuée (y compris via des Gateways)
  - Mise à jour distribuée
  - réplication de données
  - COMMIT à deux phases
- Option objet relationnel

1.4 Oracle 10G Présentation Générale

Les Architectures d'Oracle

- Architecture standard
  - une SGA
  - des process tâche de fond
  - un process serveur par process utilisateur
  - Machine virtuelle java
  - ORB(Object Request Brocker) intégré
  - Sqlnet
- Architecture multithread
  - une SGA
  - des process tâche de fond
  - un process serveur pour plusieurs process utilisateurs
  - Machine virtuelle java
  - ORB(Object Request Brocker) intégré
  - à partir de SQLNET

1.4 Oracle 10G Présentation Générale

Intégration de la technologie Java

- une machine virtuelle Java dans le serveur
- fourniture de deux types de drivers JDBC (OCI, un Driver pour les applets Java)
- Support d'une extension Java à SQL (JSQL)
  - JSQL code->Préprocesseur JSQL -> Code Java avec des appels JDBC -> Compilateur Java -> Code Java
- Support des Java Beans
- Utilitaire d'import / export de programmes Java

Intégration dans l'environnement CORBA

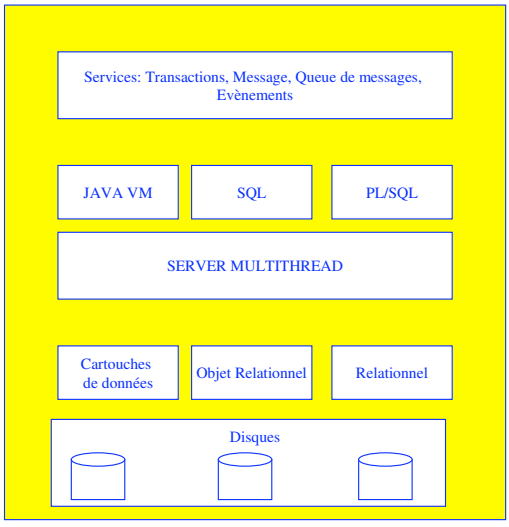
- Support du protocole IIOP
- accès transparent aux objets stockés dans une base Oracle

Extensibilité du moteur Oracle via des cartouches

- cartouches de base : spatial, Image, Time, Text, Audio, Video
- Possibilité d'introduire de nouveaux mécanismes d'indexation reconnus par l'optimiseur statistique d'Oracle

1.4 Oracle 10G Présentation Générale

Plate forme ouverte, en résumé

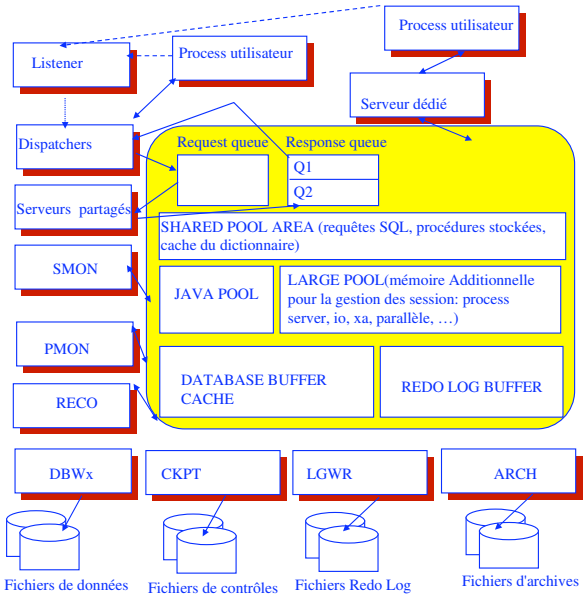


2. Architecture d'Oracle

Plan

2.1 Architecture Générale

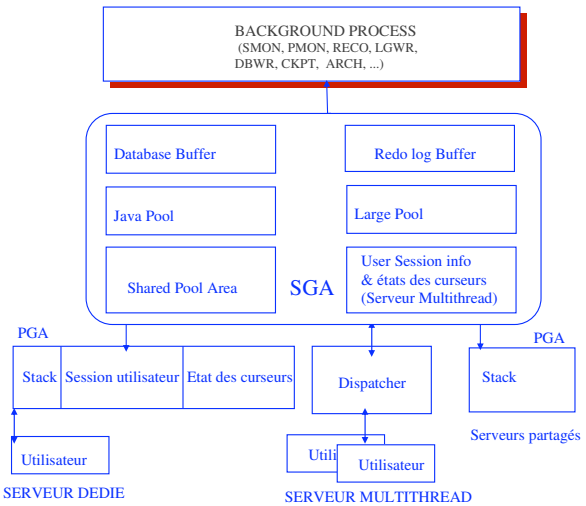
Représentation schématique



2.2 La notion d'instance

Une instance Oracle correspond à :

- une SGA (System Global Aréa)
- des Background Process
- des Process Serveurs



2.3 Structure de la mémoire d'Oracle

Principales composantes :

- la SGA (System Global Area)
- la PGA (Program Global Area)
- la zone de Tri (Sort Area Size)
- Java Pool
- Large Pool

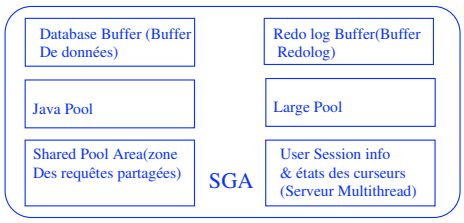
Gérer essentiellement selon le principe dit *LRU* (Last Recent Used)

La taille de ces zones est déterminée grâce à des *paramètres d'initialisation*

Un *sous-dimensionnement* peut entraîner des pertes importantes de performances

2.4 La SGA

- **La figure ci-dessous représente les principales composantes de la SGA (System Global Area)**



2.4 La SGA

- **Buffers de données**
  - Zone de chargement et de mise à jour en mémoire des blocs de données (blocs les plus récemment utilisés)
  - Ces blocs proviennent des fichiers de données
  - Les blocs concernés peuvent être :
    - des blocs de tables et clusters
    - des blocs d'index (B-tree, Bitmap, Reverse Key, …)
    - des blocs des rollback segments
  - Le buffer de données est géré selon le mécanisme LRU (Last Recent Used). Seul les blocs les plus récemment utilisés sont maintenus en mémoire. Les blocs les moins récemment utilisés sont éjectés du Buffer de données, ceux modifiés écrits dans les fichiers de données

2.4 La SGA

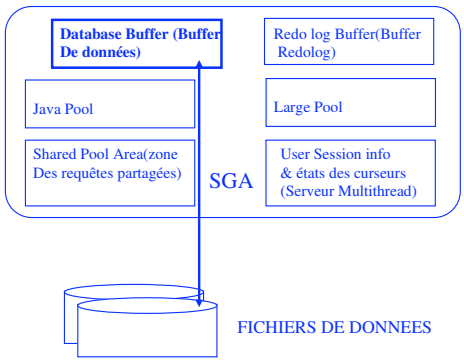
- **Buffers de données (suite)**
  - Paramètres d'initialisation influençant sa taille
    - DB_BLOCK_BUFFERS : nombre de blocs jusqu'à Oracle 8
    - DB_CACHE_SIZE : nombre de blocs du buffer de données par défaut.
    - DB_BLOCK_SIZE taille du bloc par défaut
    - DB_16K_CACHE_SIZE : nombre de blocs du buffer de données de blocs de 16K
    - DB_2K_CACHE_SIZE : nombre de blocs du buffer de données de blocs de 2K
    - DB_32K_CACHE_SIZE : nombre de blocs du buffer de données de blocs de 32K
    - DB_4K_CACHE_SIZE: nombre de blocs du buffer de données de blocs de 4K
    - DB_8K_CACHE_SIZE : nombre de blocs du buffer de données de blocs de 8K
    - **Oracle peut gérer plusieurs buffers** de données avec des tailles de blocs différents si DB_CACHE_SIZE et au moins un DB_xK_CACHE_SIZE sont posés
    - **DB_CACHE_SIZE et DB_xK_CACHE_SIZE** sont modifiables dynamiquement via ALTER SYSTEM

2.4 La SGA

- **Buffers de données (suite)**
  - Les Buffers Pools multiples (3 pools)
    - Les paramètres d'initialisations BUFFER_POOL_KEEP et BUFFER_POOL_RECYCLE permettent de définir deux buffers pools supplémentaires. Le premier étant celui par défaut
    - La zone définie par **BUFFER_POOL_KEEP** permet de définir un espace ou fixer les objets en mémoire
    - La zone définie par **BUFFER_POOL_RECYCLE** permet de définir un espace ou fixer les objets qui s'y trouvent libérés aussitôt qu'on ne les utilisent plus
    - **NOTE:** Seule le buffer de données par défaut définit par le paramètre d'initialisation DB_CACHE_SIZE peut être géré de la sorte.

Buffers de données (suite)

- Les buffers de données sont organisés en deux listes : la **dirty list** et la **LRU list.** La Dirty List contient la liste des blocs en modification. La connaissance de cette Liste permet d'accélérer l'écriture des blocs modifiés dans les fichiers de données



Buffers de données (suite)

- les performances sont bonnes si le ratio R est >= 60 ou 70%

$$R = 1 - \frac{\text{Physical read}}{\text{db block gets + consistent gets}}$$

  - **Physical read** : nombre de lecture sur disque
  - **db block gets + consistent gets** : nombre total de lecture sur disque ou en mémoire.

- La table v$sysstat contient les statistiques utiles :

```
SELECT name, value
       FROM v$sysstat
       WHERE name IN ('db block gets',
              'consistent gets', 'physical reads');
```

Name	Value
db block gets	85792
consistent gets	278888
physical reads	13182

Zone de partage des ordres SQL

- **se compose des données suivantes** :
  - les plans d'exécution et les résultats d'analyse des ordres venant des processus utilisateurs
  - les procédures stockées (PL/SQL)
  - les requêtes récursives (requêtes sur le dictionnaire)

- **Condition de partage**
  - le plan d'exécution et les résultats d'analyse sont encore dans le buffer
  - les objets composants la requête n'ont pas évolués
  - le texte de la requête est identique au caractère prêt y compris le code PL/SQL

Exemple:

```
SELECT * FROM DEPT
est différent de
Select * FROM DEPT
ou
SELECT * FROM  . DEPT
est différent de
SELECT * FROM  .. DEPT
```

  - NOTE : le dba peut nettoyer le buffer via la commande

```
Alter System Flush Shared Pool
```


Zone de partage des ordres SQL (suite)

- **Informations sur les requêtes**
  - v$sqlarea (texte des requêtes)
  - v$librarycache (tuning de requêtes partagées)
  - v$rowcache (tuning du dictionnaire d'Oracle)

- **Optimisation du cache de la librairie**

```
SELECT sum(pins) "Executions",
  sum(reloads) "Défaut de cache",
  sum(reloads) / (sum(pins) + sum(reloads))*100 "R"
  FROM v$librarycache ;
```

reloads : défaut de lecture dans le cache de librairie d'exécutions
pins : nombre d'exécutions sans défaut de cache

si R >= 1% alors augmenter **SHARED_POOL_SIZE**

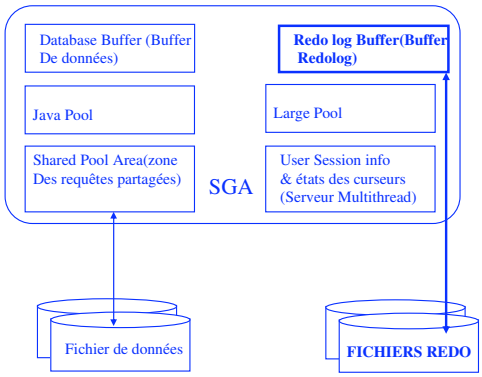
- **Optimisation du cache du dictionnaire**

```
SELECT sum(gets) "DC Gets",
  sum(getmisses) "DC cache get Misses",
  sum(getmisses) / (sum(gets)+sum(getmisses))*100 "R"
FROM v$rowcache ;
```

R doit être <= 10% ou 15% sinon accroitre **SHARED_POOL_SIZE**

Le Buffer REDO LOG

- Tampon contenant les informations en cours de modification dans les blocs de données :
  - données avant et données après
- sa taille est déterminée par le paramètre
  - LOG_BUFFER



Buffer Redo log (suite)

- un **buffer Redo log *trop petit*** entraîne des attentes
- **Optimisation du buffer Redo log**
  - la table des performances v$sysstat contient les information utiles
  - SELECT name, value
    FROM v$sysstat
    WHERE name = 'redo log space requests' ;

    name : nom de la statistique
    value : valeur de la statistique
  - **interprétation**

    Si **value** est très proche de 0 alors OK
    Si **value** croit souvent alors il y a attente : augmenter LOG_BUFFER par palier de 5%

Large Pool

- Le DBA peut configurer une zone de la SGA appelé Large Pool pour soulager le Buffer de données ou la Zone des requêtes partagés pour certaines opérations gourmandes en mémoire
- Que peut fournir la Large pool ? :
  - L'espace mémoire nécessaire pour les sessions gérés par les serveurs partagés
  - L'espace mémoire pour les transactions XA (moniteur transactionnel)
  - L'espace mémoire pour effectuer les Backup et Restauration
  - L'espace mémoire pour le traitement des requêtes parallèles si paramètre AUTOMATIC_TUNING=TRUE
- Dimensionnement de la Large POOL
  - Large_pool_size=valeur

Java POOL

- Zone de mémoire nécessaire pour la machine virtuelle Java intégré dans Oracle
- Cette zone permet d'exécuter le code Java stocké dans le noyau Oracle
- Dimensionnement de la Java POOL
  - java_pool_size=33554432

Maintenir la SGA en mémoire centrale

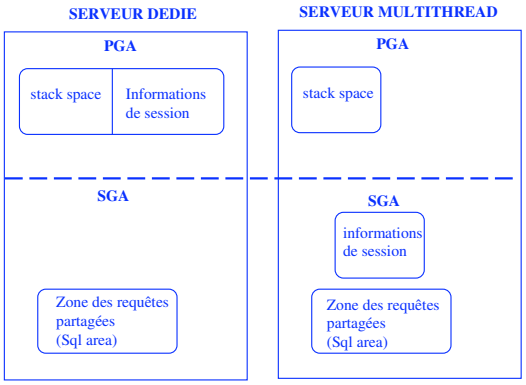
- LOCK_SGA : ce paramètre vaut par défaut FALSE, le poser à TRUE si l'on souhaite que la SGA soit défini en mémoire centrale uniquement (pas en mémoire virtuelle)

2.5 La PGA

- **Buffer contenant des données et des informations de contrôle pour un process serveur**
- **La PGA est une zone de mémoire privée**
- **Les tables v$sesstat, v$statname, permettent de déterminer la taille de la PGA pour une session**
  - Select ss.sid, ss.value, sn.name
    FROM v$sesstat ss, v$statname sn, v$session se
    WHERE ss.statistic#=sn.statistic#
    and sn.name in ('session pga memory')
    and se.sid=ss.sid and type != 'BACKGROUND';

2.5 La PGA

- **Contenu de la PGA selon le type de Serveur**



STACK SPACE (zone mémoire contenant) :
. des variables de sessions
. des tableaux
. ...

Informations de session :
. en SGA avec le serveur Multithread
. contient la sql private area

2.5 La PGA

- **La taille maximum de la PGA est influencée en plus par les paramètres d'initialisations suivants :**
  - sort_area_size
  - hash_area_size
  - bitmap_merge_area_size and create_bitmap_area_size
- **D'autres paramètres influence aussi la taille de la PGA d'une session**
  - OPEN_LINKS : nombre de databases link ouverts
  - DB_FILES : nombre de fichiers de données pouvant être ouverts
- **En mode serveur dédié il est difficile de gérer l'allocation des paramètres **_area_size*. Depuis la 9i le DBA peut fixer sa PGA maximale grâce au paramètre :**
  - PGA_AGGREGATE_TARGET

2.6 La Zone de tri

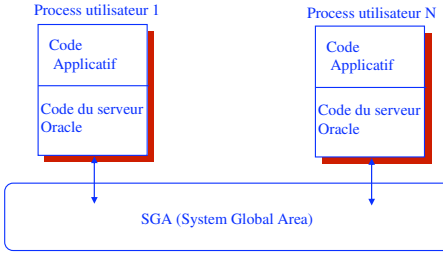
- **Une zone de tri** est associée à un Serveur (dédié ou non) pour traiter des ordres nécessitant des tris (Group by, Order by, Join, ...)
- **la taille de la zone de tri** est déterminée par le paramètre **SORT_AREA_SIZE** (en bytes)
  - Par défaut cette taille est de **65000 bytes**
  - Si cette zone est pleine un **Segment temporaire** est généré
- **SORT_AREA_RETAINED_SIZE** (exprimée en byte, 0 min, Sort_area_size par défaut et max) : espace à ne pas libérer en cas d'écriture dans le segment temporaire
- **tuning de la zone de tri** ; table v$sysstat
  - SELECT name, value FROM v$sysstat
    WHERE name in ('sorts (memory)', 'sorts (disk)');
    NOTE : Si le nombre de tris sur disque croit, augmenter Sort_area_size. Maisattention au Swapping de l'OS.

2.7 Différentes configurations d'Oracle

- **Trois approches**
  - Single task Oracle
  - Serveur dédié
  - Serveur Multithread

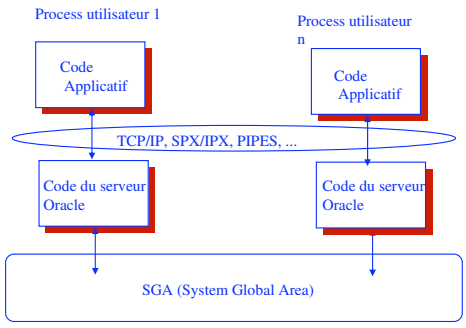
2.7 Différentes configurations d'Oracle

- **Single task Oracle**
  - **Combinaison du code du process serveur et du process utilisateur**
  - **plus rapide** (par exemple lors d'un import massif) par contre le système d'exploitation doit permettre une séparation étanche entre le code de l'application et le code Oracle (exemple VAX VMS)
  - **Architecture**



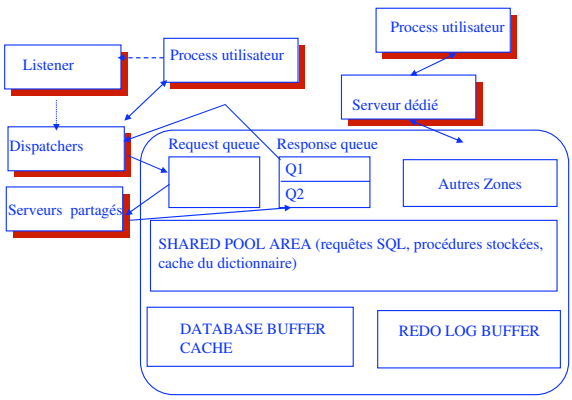
2.7 Différentes configurations d'Oracle

- **Serveur dédié**
  - code du process utilisateur séparé de celui du process serveur
  - à un process utilisateur est associé à un process serveur
  - configuration aussi appelée TWO - TASK



2.7 Différentes configurations d'Oracle

- **Serveur Multithread**
  - Permet une meilleure exploitation des ressources
  - SQL*NET V2 au moins est un pré requis



2.8 Les process autour d'Oracle

- **Deux classes de process autour d'Oracle**
  - Les process utilisateurs (liés à l'exécution d'un outil, d'un programme d'application, ...)
  - Les process Oracle
- **Process Oracle**
  - Les process tâches de fond (SMON, PMON, LGWR, DBWR, CKPT, ARCH, RECO, ...)
  - Les process serveurs
  - autres process

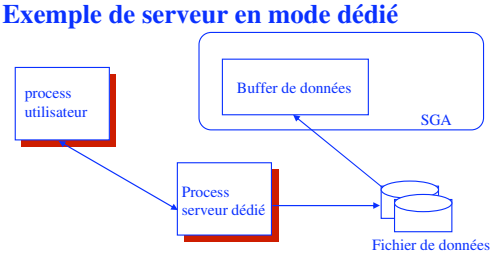
NOTE : Certains de ces process sont facultatifs (ARCH, CKPT)

2.9 Les process utilisateurs

- **Process client exécutant le code d'une application** (PRO*C, FORMS, ...) ou d'un Outil Oracle (SQL*PLUS, ENTREPRISE MANAGER, ...)
- **Process souvent exécuté sur une machine différente de celle ou réside le serveur Oracle**
- **process qui établit une communication avec Oracle via un protocole de communication et SQLNET**
- **La communication est gérée via le User Programme Interface (UPI)**

2.10 Les Processus Serveurs

- **Un process serveur peut être dédié ou non**
- **Il est aussi appelé *shadow process***
- **Son rôle consiste :**
  - à assurer la **communication directe ou indirecte** avec les process utilisateurs
  - à **analyser et exécuter** les requêtes
  - à **lire** les blocs de données dans les fichiers de données
  - à **restituer directement ou indirectement** le résultat au process utilisateur
  - à **déplacer les blocs modifiés** dans la DIRTY LIST
- **Exemple de serveur en mode dédié**



2.11 Les Process tâches de fond

- **Le process Database Writer (DBWn)**
  - **Ecrit les blocs de données modifiés** de la SGA vers les fichiers de données
  - est optimisé pour minimiser les accès disques
  - il peut avoir un ou plusieurs DBWn (**DB_WRITER_PROCESSES=n** allant de 0 à 9)
  - **Quand s'active DBWn** ?
    - Lors d'un CHECKPOINT (LGWR ou CKPT l'avertit)
    - pour libérer de la place dans le Buffer de données à la demande du process serveur
    - sur un TIMEOUT (toute les 3s)

2.11 Les Process tâches de fond

Le Process LOG WRITER (LGWR)

- **trace le contenu du buffer REDO LOG** dans les fichiers REDO LOG
- en cas de checkpoint (**CKPT absent**) LGWR réveille DBWR et modifie l'entête des fichiers de données et de contrôles
- **quand s'active LGWR** ?
  - Si un COMMIT à été passé
  - Sur un time out toute les 3 secondes
  - Si le buffer REDO LOG est plein au 1/3
  - quand DBWR libère des blocs de données du buffer de données(en cas de TIMEOUT ou de checkpoint)
- **Optimisation**
  - activer le process CHECKPOINT (CKPT)

2.11 Les Process tâches de fond

Le Process CHECKPOINT (CKPT)

- s'obtient en fixant le paramètre CHECKPOINT_PROCESS=TRUE
- Si présent informe DBWR qu'un CHECKPOINT est intervenu
- note le Checkpoint dans l'entête des fichiers de données et de contrôles
- Un **checkpoint intervient** :
  - si le TIMEOUT a été atteint (LOG_CHECKPOINT_TIMEOUT : 0 par défaut)
  - si la fin d'un groupe de fichiers Redo log est atteint
  - si une taille correspondant à LOG_CHECKPOINT_INTERVAL (en blocs OS) a été écrite dans le fichier REDO LOG
  - si les commandes suivantes sont passées :
    - ALTER SYSTEM CHECKPOINT (pas de changement de REDO LOG)
    - ALTER SYSTEM SWITCH LOGFILE
- **Optimisation**
  - Favoriser le recouvrement ou les performances en dimensionnant mieux LOG_CHECKPOINT_INTERVAL et LOG_CHECKPOINT_TIMEOUT

2.11 Les Process tâches de fond

Le Process CHECKPOINT (CKPT) et Intérêt d'un checkpoint

- permet de forcer l'écriture dans les fichiers de données des blocs de données restant en mémoire car fréquemment modifiés (Mécanisme LRU)
- permet d'accélérer le recouvrement de données : les données avant le checkpoint dans le fichier Redo log ne seront plus appliquées au fichiers de données car elles y sont déjà présentes.

2.11 Les Process tâches de fond

Le Process ARCH

- sauvegarde le fichier REDO LOG lorsqu'il est plein
- c'est un process facultatif. Jusqu'à 10 process : ARC0…10 peuvent être activés.
  - Le paramètre ***log_archive_max_processes*** permet fixer le nombre maximum. Il est conseillé de garder la valeur par défaut car Oracle alloue lui-même de nouveaux process en cas de besoin
- Les archives peuvent être redirigées sur une bande
- La base doit être démarrée en mode avec Archive

2.11 Les Process tâches de fond

Le Process ARCH

- Activation et configuration de ARCH sous UNIX :
  - **log_archive_start** = TRUE # activation des process ARCn
  - **log_archive_dest** = /dev/rmt0:100M
    ou /backup/DB1/COURSarch
    - Destination des archives jusqu'à la version 8 d'Oracle
  - **log_archive_format** = %t_%s.arc
    ou t : THREAD, s : Séquence number
  - **log_archive_duplex_dest** : destination alternative jusqu'à la version 8 d'Oracle
  - **log_archive_min_succeed_dest**=1 à 5 depuis la version (1 à 2 avant).Ce paramètre indique le nombre de copie à faire lors de l'archivage d'un fichier Redo log
  - **log_archive_dest_state_n** (n de 1 à 10) = enable ou Defer. Permet de gérer l'état d'une destination de sauvegarde log_archive_dest_n
  - **log_archive_dest_n** (n de 1 à 10)=/backup/DB1/COURSarch
  - **standby_archive_dest** : destination des archives pour une base en Standy

2.11 Les Process tâches de fond

System Monitor(SMON)

- **Répare** l'instance au démarrage en cas d'arrêt brutal
- **Libère** les segments temporaires
- compacte les extensions libres pour les rendre contiguës (Alter Tablespace nomtablespace Coalesce)
- recouvre **les process suspendus** suite à un crash

Process Monitor (PMON)

Fait le ménage en cas de disparition brutale d'un process utilisateur.

- Supprime au **niveau Oracle les process en erreur**
- Annule les transactions en cours
- Libère les verrous
- contrôle les process **dispatchers et serveurs**. S'ils ne sont plus présents, il les redémarre.

2.11 Les Process tâches de fond

Recover (RECO)

- Sert uniquement si option distribuée
- termine les transactions distribuées en suspens dû à une erreur réseau ou système
- se réveille par intervalles réguliers pour finir ou annuler des transactions suspendues

Dispatcher (Dnnn)

- réceptionne les requêtes des process utilisateurs et les met à disposition d'un process serveur
- lit les résultats d'une requête et les redirige vers le process utilisateur concerné
- au moins un dispatcher par type **de protocole réseau**
- permet de partager les process serveurs

LOCK (LCKn)

- Utile avec l'option parallèle
- jusqu'à 10 Lock process peuvent être démarrés

2.12 Autres process

tnslnr : permet la connexion de client SQL*NET protocole TCP/IP

dnslnr : permet la connexion de client SQL*NET protocole DECNET

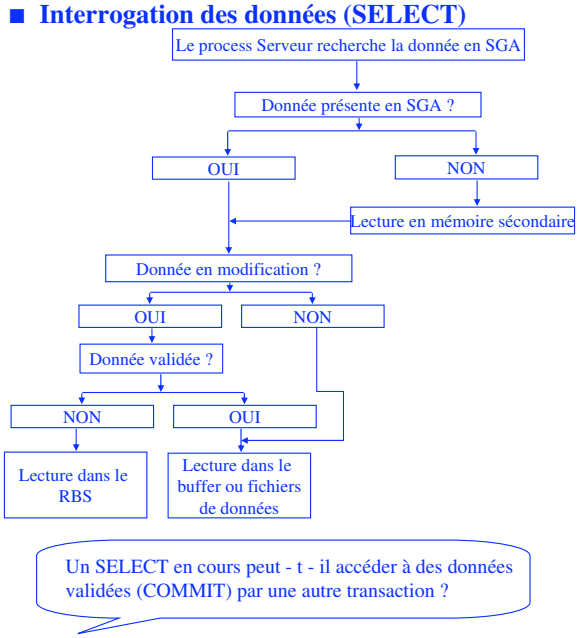
Jn (J000 à J999): Job queue Processes

- Permet de traiter des tâches Batch programmées dans Oracle
- Un process coordinateur CJQ0 est le seul au départ démarré. C'est lui qui démarre automatiquement les process Jnnn et leur affecte des tâches
- Les tâches peuvent être aussi des tâches de réplication entre bases Oracle
- Avant Oracle 10G on parlait de SNP0 à SNP36

Queue Monitor (QMNn avec n de 0 à 9) : ces process servent au monitoring des queues de messages en cas de gestion de queues de messages avancée

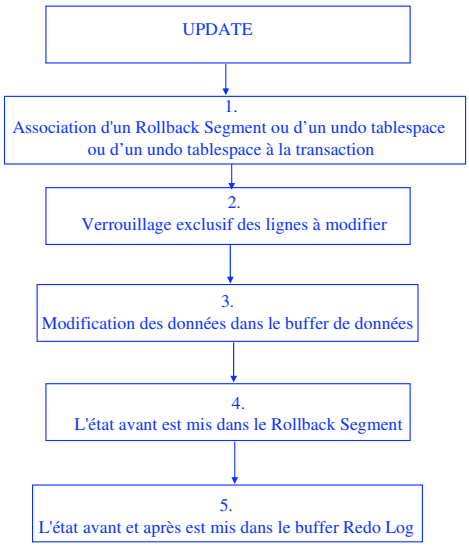
2.13 Mécanismes de lecture et écriture

- **Interrogation des données (SELECT)**



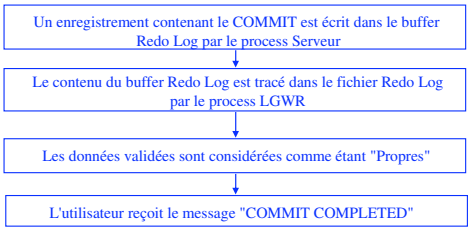
2.13 Mécanismes de lecture et écriture

- **Mise à jour (UPDATE)**

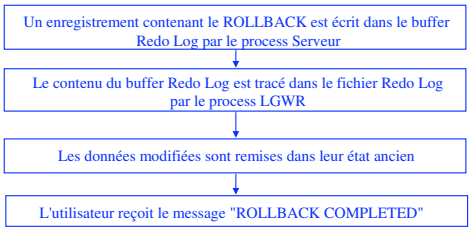


2.14 Mécanisme de validation et invalidation

- **COMMIT**



- **ROLLBACK**



L'espace alloué dans le ROLLBACK SEGMENT est - t - il désalloué immédiatement ?

3 Tâches d'administration de base

- **Plan**

3.1 Installation du noyau et des outils Oracle

- **Tâches avant installation**
- **Installation**
- **Tâches après installation**

3.1 Installation du noyau et des outils Oracle

- **Tâches avant installation**
  - **Etudier le manuel d'installation** d'Oracle ou du produit
  - **préparer l'OS et la machine**
    - **Cas Unix** : création d'un compte Oracle, création d'un groupe dba, Modification du noyau si utile UNIX : ajout de sémaphore, …
    - **Cas NT** : se connecter sur la machine comme administrateur de la celle-ci
  - **estimer l'espace disque** nécessaire pour les produits à installer et la base de départ
  - **choisir le type de fichiers** (fichiers OS ou Raw Device)
  - **Positionner si utile (UNIX) les variables d'environnement** ORACLE_HOME et ORACLE_SID, ...

3.1 Installation du noyau et des outils Oracle

- **Installation**

1. lancer l'exécutable fournit avec la distribution :

- **sous UNIX :** dans l'environnement motif lancer oracleInstaller
- **sous NT :** lancer SETUP depuis le premier CD

2. choix de la HOMEDIR d'Oracle

3. choix d'un type d'installation (complete, custom, minimal, ...)

4. Choix du ou des produits à installer ou à upgrader

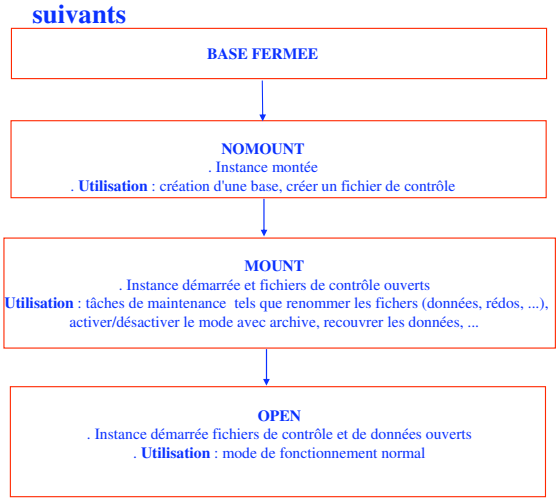
5. Saisie des informations liées à la base de départ (si première installation)

3.1 Installation du noyau et des outils Oracle

- **Tâches après installation**
  - Vérifier le fichier de trace (oraInstall.err et oraInstall.out). Windows platform: c:\temp\oraInstall\ , UNIX: /temp/oraInstall
  - Exécuter si nécessaire le fichier "*root.sh*" en tant que root pour modifier les permissions(sous Unix)
  - Vérifier et modifier les paramètres d'initialisation de la base : fichier "*init.ora*"
  - éditer le fichier "*oratab*" pour y déclarer une instance : utile(sous UNIX)
  - Permettre aux utilisateurs de lancer oraenv (sous UNIX)
  - Mettre dans rc.local **dbstart** et **dbshutdown** (sous UNIX)
  - Créer de nouvelles bases

3.2 Etats d'une base

- **Une Base de données comporte les états suivants**



3.3 Privilèges utiles

- **Privilèges nécessaires pour créer, démarrer ou arrêter une base (UNIX ou NT)**
  - SYSDBA (arrêt, démarrage, modification, création d'une base, …)
  - SYSOPER(arrêt démarrage d'une base, modification)
  -
- **Rôles attribuables à partir de l'OS (si supporté)**
  - **Rôles à attribuer**
    - OSOPER (shutdown, alter database)
    - OSDBA (create database, osoper, + tous les privilèges système)
    - autres rôles Oracle
  - **Activation**
    - OS_ROLE = TRUE # dans init.ora
    - ORA_SID_ROLE_D(default)/A(with admin option) nom d'un groupe dans le fichier /etc/group
      - Exemple
        ORA_COURS_OSOPER_D :none:3:clement
        ORA_COURS_OSDBA_A:mopolo, miranda

3.4 Création d'une base

- **Variables d'environnement à positionner**
  - ORACLE_SID=nom_de_instance
  - ORACLE_HOME=home_oracle
  - ORACLE_BASE=home_des_bases_Oracle
- **Outils de lancement des commandes**
  - Sqlplus (***Server Manager*** jusqu'à la version 8)
  - Le ***Database Configuration Assistant***. Outil graphique permettant de créer et modifier les bases
  - implicite lors de l'installation du serveur Oracle

3.4 Création d'une base

- **Fichier d'initialisation init.ora**
  - Permet de fixer un certain nombre de paramètres

```
# Cache et E/S
    db_block_size=4096
    db_cache_size=33554432
# Configuration du fichier
    control_files=("C:\oracle\oradata\DBTEST\CONTROL01.CTL",
      "C:\oracle\oradata\DBTEST\CONTROL02.CTL",
      "C:\oracle\oradata\DBTEST\CONTROL03.CTL")
# Curseurs et cache de bibliothèque
    open_cursors=300
# Diagnostics et statistiques
    background_dump_dest=C:\oracle\admin\DBTEST\bdump
    core_dump_dest=C:\oracle\admin\DBTEST\cdump
    timed_statistics=TRUE
    user_dump_dest=C:\oracle\admin\DBTEST\udump

# Distribué, réplication et cliché
    db_domain=CERAM.FR
    remote_login_passwordfile=EXCLUSIVE

# Divers
    compatible=9.0.0
    db_name=DBTEST
# Inscription réseau
    instance_name=DBTEST
```

3.4 Création d'une base

- **Fichier d'initialisation init.ora**

  - Permet de fixer un certain nombre de paramètres

```
# MTS
    dispatchers="(PROTOCOL=TCP)(SER=MODOSE)",
      "(PROTOCOL=TCP)(PRE=oracle.aurora.server.GiopServer)",
      "(PROTOCOL=TCP)(PRE=oracle.aurora.server.SGiopServer)"

# Pools
    java_pool_size=33554432
    large_pool_size=1048576
    shared_pool_size=33554432

# Processus et sessions
    processes=150
# Segments d'annulation (Undo et Rollback) gérés par le système
    undo_management=AUTO
    undo_tablespace=UNDOTBS
# Tri, jointures par hachage, index bitmap
    sort_area_size=524288
```

NOTE :se référer à la description de chaque paramètre pour plus de détails (vue : V$PARAMETER ou Utiliser Entreprise Manager)

3.4 Création d'une base

- **Etapes à suivre pour créer une base avec le *Database Configuration Assistant***

1. ***Démarrer l'assistant*** *start->programme->OracleHome->Configuration and Migration Tools->Database Configuration Assistant*
2. ***Choisir créer une base***
3. ***Choisir un template (modèle) de base***
4. ***Donner le nom global de la base*** *(exemple DBTEST.CERAM.FR) ainsi que le SID (DBTEST)*
5. ***Sélectionner le mode de déploiement de la base*** *(mode dédié) ou (mode serveurs multithreads)*
6. ***Fixer la taille de la SGA****(buffer cache, shared pool, large pool, java pool)*
7. ***sélectionner le mode d'archivage de la base*** *(Archivelog ou noarchivelog).Localiser les fichiers d'archive*
8. ***Fixer la taille maximale de la zone de tri ainsi que les caractères sets***
9. ***Localiser les fichiers de contrôles, de données et redo log***
10. ***Modifier à tout moment les paramètres d'itialisation supplémentaires et démarrer la création de la base***

3.4 Création d'une base

- **Etapes à suivre pour créer une base manuellement**

1. ***Définir l'arborescence de la base***
   ora9data->dbtest->admin, tssys, tsusers, tstemp, tsrbs, …
2. ***Définir les scripts de création de la base****(crDBTEST.sql)*
3. ***Définir le fichier d'initialisation*** *(initDBTEST.ora)*
3. ***Si NT : Créer le service NT pour la base****.*

C:\>oradim –new –sid dbtest –intpwd manager –startmode auto –pfile c:\ora9data\dbtest\admin\initDBTEST.ora

4. ***Lancer Sqlplus***
   c:\> sqlplus
   username:sys as sysdba password:manager
5. ***Exécuter les commandes contenues dans crDBTEST.sql***

- *démarrer une instance*
- *exécuter CREATE DATABASE …*
- *exécuter catalog.sql (pour les vues du dictionnaires)*
- *ajouter des tablespaces supplémentaires pour une meilleure orgainisation tstemp, tsrbs, tsutil*
- *ajouter des rbs supplémentaires si mode de gestion d'annulations manuelles*
- *exécuter catproc.sql (pour l'option procédurale)*
- *exécuter les scripts supplémentaires suivants : catdbsyn.sql*
- *fixer le tablespace temporaire des users SYS et SYSTEM vers le tablespace temporaire*

Voir Annexes *pour plus de détails sur initSid.ora et crsid.sql*

3.4 Création d'une base

- **Syntaxe générale**

```
CREATE DATABASE [nombase]
    [CONTROLFILE REUSE]
    [LOGFILE {[GROUP entier] logFileSpec, …}]
    [MAXLOGFILES entier]
    [MAXLOGMEMBERS entier]
    [MAXLOGHISTORY entier]
    [MAXDATAFILES entier]
    [MAXINSTANCES entier]
    [{NOARCHIVELOG | ARCHIVELOG}]
    [CHARACTER SET nomCaracterSet]
    [NATIONAL CHARACTER SET  nomCaracterSet]
    [DATAFILE {dataFileSpec [ClauseAutoExtend],  …}]
    [default_temp_tablespace]
    [undo_tablespace_clause]
    [SET STANDBY DATABASE {PROTECTED
          UNPROTECTED}]
    [set_time_zone_clause]
```

3.4 Création d'une base

■ Syntaxe générale

***logFileSpec* ::=**
{ 'Nomfichier' | ('Nomfichier' [, 'Nomfichier'] ...)}
[SIZE *entier* [K|M]] [REUSE]

***dataFileSpec* ::=**
'Nomfichier' [SIZE *entier* [K|M] [REUSE]

ClauseAutoExtend ::=
{OFF | ON [NEXT *entier* [K | M]]
[MAXISZE {UNLIMITED | *entier*
[K|M]}] }

default_temp_tablespace::=
[DEFAULT TEMPORARY TABLESPACE tablespace
[TEMPFILE filespec] temp_tablespace_extent_clause]

temp_tablespace_extent::=
EXTENT MANAGEMENT LOCAL UNIFORM SIZE
integer [K | M]

[undo_tablespace_clause::= UNDO TABLESPACE
tablespace
[DATAFILE filespec1 [autoextend_clause1], …]]

3.4 Création d'une base

Description des mots clés et des paramètres

Mot clé ou paramètre	Description	Default (Max)
nombase	nom de la base (8 caractères max)	
CONTROLFILE REUSE	réutiliser les fichiers de contrôle existant en cas de recréation de la base	
LOGFILE	permet de spécifier les fichiers Redo log	
GROUP *entier*	Group de fichiers redolog en miroir	
MAXLOGFILES *entier*	nombre maximum de fichiers Redo log	16 (255)*
MAXLOGMEMBERS *entier*	nbre max de membres dans un groupe Redo log	2 (5)*
MAXLOGHISTORY	Nbre maximum de fichier d'archive à noter dans le fichier de contrôle pour recouvrement automatique si BD parallèle	
MAXDATAFILES *entier*	Nbre maximum de fichiers de la bas	30(65533)*
MAXINSTANCES *entier*	Nbre max d'instances actives sur cette base	63 (63)*
NOARCHIVELOG | ARCHIVELOG	Base créée en mode sans archive(avec archive)	noarchivelog
CHARACTER SET	Langage de stockage des données	us7ascii
NATIONAL CHARACTER SET	Langage nationale de stockage des données (type NCHAR, NVARCHAR2, …)	
DATAFILE *dataFileSpec*	Fichiers de données du tablespace System	

**** Dépendent de l'OS***

3.4 Création d'une base

Description des mots clés et des paramètres

Mot clé ou paramètre	Description	Default (Max)
default_temp_tablespace	**permet de définir le TS temporaire par défaut.**	
undo_tablespace_clause	**permet de définir le TS des données Rollback par défaut**	
SET STANDBY DATABASE	**lié à une base en standby**	
set_time_zone_clause	**fixer l'horloge**	

3.4 Création d'une base

■ Spécification des fichiers

Exemple

```
Create database DBCOURS
        CHARACTER SET us7ascii
        MAXDATAFILES 100
        MAXINSTANCES 1
        MAXLOGFILES 24
        MAXLOGMEMBERS 3
        NOARCHIVELOG
datafile
        'oracle/oradata/DBCOURS/tssys/sys1dbcours.dbf' size 110 M,
        'oracle/oradata/DBCOURS/tssys/sys2dbcours.dbf' size 110 M
                AUTOEXTEND ON NEXT 10M MAXSIZE
                UNLIMITED
logfile 'oracle/oradata/DBCOURS/disk1/log1adbcours.dbf' size 500K,
        'oracle/oradata/DBCOURS /disk1/log2adbcours.dbf' size 500K,
        'oracle/oradata/DBCOURS/disk1/log3adbcours.dbf' size 500K
DEFAULT TEMPORARY TABLESPACE temp_ts TEMPFILE
        'oracle/oradata/DBCOURS/tstemp/temp1dbcours.dbf' size 110 M
                AUTOEXTEND ON NEXT 10M MAXSIZE
                UNLIMITED
UNDO TABLESPACE undo_ts
        datafile 'oracle/oradata/DBCOURS/tsrbs/rbs1dbcours.dbf'
                size110 M AUTOEXTEND ON NEXT 50M
                MAXSIZE UNLIMITED
SET TIME_ZONE = '+02:00';
```

3.4 Création d'une base

- **Résultat de la création d'une base**
  - Création ou réutilisation de Fichiers de **contrôles**
  - Création ou réutilisation de Fichiers **Redo log**
  - Création du tablespace SYSTEM
  - Création du rollback segment SYSTEM
  - Création du dictionnaire de données de base (sql.bsq)
  - Création d'au moins deux utilisateurs SYS et SYSTEM
  - Création d'un tablespace pour les segments temporaires temp_ts
  - Création d'un tablespace (undo_ts) pour les données Rollback. Création de 10 RBS implicite pour ce RBS
- **Travail complémentaire (scripts à exécuter)**
  - création des vues et synonymes publics (CATALOG.SQL)
  - Installation de l'option procédurale(CATPROC.SQL)
  - création des synonymes publics pour les tables virtuelles (UTLMONTR.SQL)
  - Création des Synonymes sur les vues dba_* du dictionnaire (CATDBSYN.SQL) pour le DBA

3.5 Le fichier d'initialisation init.ora

- **Les paramètres**
  - environ **250 paramètres** dont la plupart sont renseignés par défaut
  - trois **classes de valeurs** de paramètres
    - les booléens (TRUE, FALSE)
    - des chaînes de caractères
    - et des entiers.
  - ils sont **regroupés par catégorie**
    - db_* : paramètres liés à la base
    - log_* : paramètres liés au REDO LOG
    - distributed_* : paramètres liés à l'option distribuée
    - mts_* : paramètres liés à l'architecture Multithread
    - gc_* : paramètres liés à l'option parallèle
    - nls_ *: paramètres liés au National Language Support
    - sort_*: paramètres liés au tri
    - ...
    - non classés (processes, sessions, ...)

Voir Annexes A3 et A7

3.6 Démarrage d'une base

- **Procédure Générale**
  - Positionner les variables d'environnement ORACLE_SID et ORACLE_HOME
  - démarrage de la base possible par paliers (***startup nomount, startup mount, et startup open***)
  - posséder les privilèges appropriés (SYSDBA, SYSOPER, OSDBA ou OSOPER, ...)
  - indiquer si nécessaire le fichier des paramètres
  - démarrer la base sous *sqlplus* (Unix et NT) ou via *les services*(NT) ou *Oradim* (NT), à travers Entreprise Manager: *database administration assistant* (NT ou Unix)
- **Syntaxe de la commande**
  - STARTUP [FORCE] [RESTRICT] [PFILE=fich_param] [OPEN | MOUNT | NOMOUNT]
  - ALTER DATABASE [nom_base] MOUNT | OPEN

3.6 Démarrage d'une base

- **Description des mots et paramètres**

FORCE	Si instance ouverte, Fermeture puis démarrage
RESTRICT	sert à des tâches de maintenance. Il faut avoir le privilège *Restricted Session*
PFILE	= Indique le fichier des paramètres
OPEN	démarrer et ouvrir les fichiers de la base
MOUNT	Instance démarrée fichier(s) de contrôle ouvert(s)
NOMOUNT	Seule l'instance est démarrée
ALTER DATABASE	permet le démarre par paliers après un startup NOMOUNT ou MOUNT

3.6 Démarrage d'une base

- **Exemple**

  - **Variables d'environnement** (sous UNIX)
    $ export ORACLE_SID = COURS
    $ export ORACLE_HOME=/user/oracle/v9

  - **Lancer SQLPLUS**
    $sqlplus
    username : sys as sysdba password : manager

    #démarrer et ouvrir les fichiers de la base
    SQL >startup ;
    idem
    SQL>startup OPEN
    pfile= /user/oracle/v9/dbs/initCOURS.ora

    #démarrer la base par paliers
    $ SQLPLUS
    username : sys as sysdba password : manager
    # Instance démarrée, fichiers de contrôles ouverts
    SQL >startup mount ;
    #En plus les fichiers de données sont ouverts
    SQL > alter database open ;

3.7 Arrêt d'une base

- **Procédure Générale**
  - Posséder les privilèges OS (osdba, osoper)
  - positionner les variables d'environnement (ORACLE_SID et ORACLE_HOME)
  - Lancer l'outil SQLPLUS
- **Syntaxe de la commande**
  - SHUTDOWN [NORMAL | IMMEDIATE | ABORT | TRANSACTIONAL]
    - **NORMAL**
      attend la déconnexion de l'ensemble des utilisateurs
    - **IMMEDIATE**
      Invalide les transactions en cours et déconnecte les users
    - **ABORT**
      Arrêt violent de l'instance (recouvrement utile : SMON)
    - **TRANSACTIONAL**
      Contrairement à IMMEDIATE, laisse finir les transactions en cours

3.7 Arrêt d'une base

- **Exemple**

  - Positionner les variables d'environnement (UNIX)
    $export ORACLE_SID=COURS
    $export ORACLE_HOME=/users/oracle/v9

  - **Lancer SQL**
    $SQL
    username : sys as sysdba password : manager

    # arrêt normal avec attente de déconnexion
    SQL>shutdown
    ou
    # **Arrêt immédiat :** Annulation des transactions en
    # cours et déconnexion .
    # Les utilisateurs ont le message suivant :
    # ORA-03113 : end-of-file on communication channel
    SQL>shutdown immediate ;
    ou
    # **Arrêt brutal** (pas d'attente de déconnexion, pas
    # d'invalidation de transactions en cours)
    SQL >shutdown abort;

3.8 Passage de la base en mode restreint pour le DBA

- **Afin de permettre au DBA d'opérer des tâches incompatibles avec la concurrence d'accès tels que : la modification du schéma, la suppression / recréation d'index, … il est possible de passer la base en mode QUIESCED alors qu'elle est ouverte normalement.**
- **Le passage en mode QUIESCED est obtenu base en ligne**
  - SQL> ALTER SYSTEM QUIESCE RESTRICTED
- **Le retour en mode normal est obtenu comme suit :**
  - SQL>ALTER SYSTEM UNQUIESCE
- **La vue V$INSTANCE permet d'avoir les information le mode d'activation d'une instance**
- **NOTE :**
  - Le mode QUIESCE peut être aussi obtenu en démarrant la base en mode RESTRICTED SESSION
  - Seul les utilisateur SYS et SYSTEM peuvent activer ce mode

3.9 Suspension et réactivation de la base

- **Afin de permet une sauvegarde base en ligne sans activité de mise à jour dans les fichiers de données il est possible maintenant de suspendre l'activité de mise à jour**
- **Commande à exécuter pour suspendre la base**
  - SQL>ALTER SYSTEM SUSPEND
- **Commande à exécuter pour revenir en mode normal**
  - SQL>ALTER SYSTEM RESUME
- **NOTE : les règles de sauvegarde *base ouverte* doivent être respectées (voir le chapitre Sauvergarde et Restauration)**

3.10 Le dictionnaire de données d'Oracle

- **un dictionnaire par base de données**
- **Ensemble de tables, vues et synonymes permettant la gestion des objets d'une base** (propriété de l'utilisateur SYS)
- **Il est accessible via des ordres SQL**
- **Il est modifié indirectement via des ordres dits DDL**
- **Il ne doit être modifié directement**
- **Contient des informations persistantes** (objets créés tels que les tables, les index, les clusters, ...) et des informations dynamiques (tels les sessions ouvertes, les E/S effectuées, ...)

3.8 Le dictionnaire de données d'Oracle

- **Contenu du dictionnaire**
  - des **tables de base** du dictionnaire stockées en cluster et finissant par $. Exemple tab$, ind$, obj$, seg$, ... (script les contenant : sql.bsq)
  - les **vues de performances** exemple :
    - v_$process, v_$session, v_$syststat, v_$sesstat, ...
  - des **vues sur** le dictionnaire de base commençant par :
    - all_* : informations sur les tous les objets accessibles par l'utilisateur connecté
    - user_* : information sur tous les objets dont l'utilisateur connecté est propriétaire
    - dba_* : information sur tous les objets de la base. Il faut avoir le privilège SELECT ANY TABLE pour y accéder
  - des **synonymes** sur les vues pour simplifier. Exemple:
    - v$process, v$session, v$systat, v$sesstat, ...

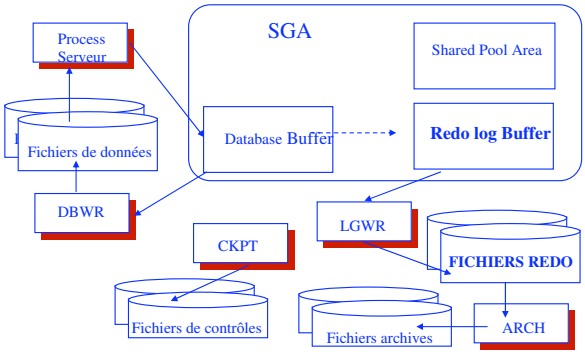
Voir aussi les Annexes A4 et A5

4. Structure d'une Base de Données Oracle

- **PLAN**
  - 4.1 Structure Physique d'une Base de Données Oracle
    - 4.1.1 Les fichiers de données
    - 4.1.2 Les fichiers Redo log
    - 4.1.3 Les fichiers d'Archives
    - 4.1.4 Les fichiers de Contrôle
  - 4.2 Structure logique d'une Base de données Oracle
    - 4.2.1 Les tablespaces
    - 4.2.2 Les segments et leurs composants
    - 4.2.3 Les segments de données de type table
    - 4.2.4 Les segments de données de type cluster
    - 4.2.5 Les segments d'index
    - 4.2.6 Les segments temporaires
    - 4.2.7 Les segments rollback (manuels ou automatiques)

4.1 Structure Physique d'une Base de Données Oracle

- **Physiquement, une base de données Oracle est composée d'un certain nombres de fichiers :**
  - des Fichiers de données
  - des fichiers REDO LOG
  - des fichiers de contrôle
  - des fichiers d'Archivage (si fonctionnement avec Archive)



4.1.1 Les fichiers de données

- **Ils contiennent toutes les données relatives à une base de données** (dictionnaire Oracle, Tables, index, clusters, rollback segments, segment temporaires)
- **Ces fichiers sont lus par les process Serveurs et modifiés par DBWn**
- **L'unité de découpage est le bloc (2K, 4K, 8K, 16K, 32K) selon l'OS**
- **Ces fichiers sont de taille fixe et optionnellement variables depuis la 7.2**
- **Ils appartiennent à une et une seule base**

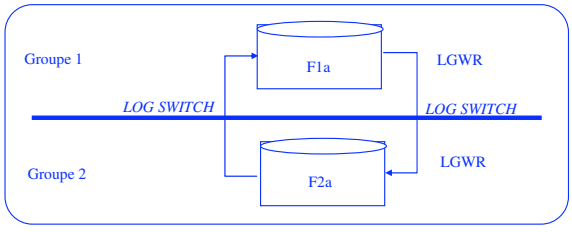
Que se passe - t - il si MAXDATAFILES est atteint ?
Que se passe - t - il si DB_FILES est atteint ?
Quelles sont les mesures à prendre ?

4.1.2 Les Fichiers Redo-Log

- **Ces fichiers contiennent la trace de l'activité en terme de mise à jour sur la base**
- **deux groupes au moins** avec chacun au moins un fichier **sont obligatoires**
- **ils doivent être multiplexés** pour plus de sécurité
- **ne sont utiles qu'en cas de perte des fichiers de données ou d'arrêt anormal de la base**
- **Fonctionnent de façon cyclique**

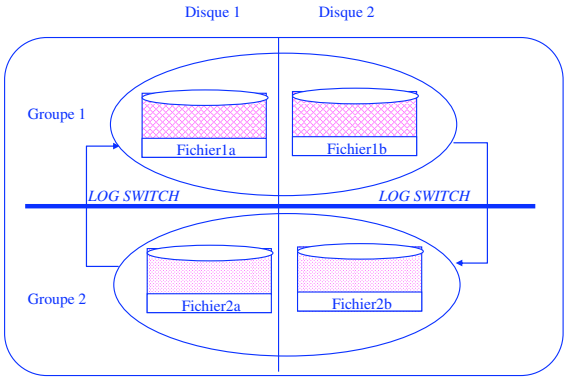
4.1.2 Les Fichiers Redo-Log

- **Configuration minimale (au moins 2 groupes avec au moins un fichier chacun)**
- **En cas de Log Switch => changement de groupe**
- **Un checkpoint intervient si :** fin de fichier, TIMEOUT, intervalle en taille atteinte, Begin Backup, shutdown normal ou immediate, ...
- **Configuration minimale** (sans multiplexage)



4.1.2 Les Fichiers Redo-Log

- **Utilisation en mode multiplexés (au moins deux groupes)**
  - Un **groupe à au moins 2 fichiers** qui sont identiques et mis à jour simultanément
  - il est **conseillé de stocker chaque fichier** d'un même groupe **sur des disques différents**
  - il est conseillé d'avoir un **même nombre de membres** par groupe



4.1.2 Les fichiers REDO LOG

- **Ajout et suppression de fichiers REDO**
  - Ajout d'un groupe de fichiers

```
ALTER DATABASE [database ]
  ADD  LOGFILE [THREAD integer ]
                [ GROUP integer ] filespec
                [, [ GROUP integer ] filespec...
```

  - Ajout d'un membre dans un groupe connaissant son Numéro ou le nom d'un fichier

```
ALTER DATABASE [database ]
  | ADD LOGFILE MEMBER
                'filename' [ REUSE ]
                [,'filename' [REUSE]]...
          TO  {  GROUP integer
               |  ( 'filename'[REUSE] [,'filename']... )
               | 'filename' [REUSE]}
          [TO  { GROUP  integer
               | ( 'filename'['filename']...)
               | 'filename'} ]...
```


4.1.2 Les Fichiers REDO LOG

- **Ajout et suppression de fichiers REDO (suite)**
  - ALTER DATABASE [database]
    DROP LOGFILE
    { GROUP integer
    | ('filename'[,'filename']...)
    | 'filename'}
    [, { GROUP integer
    | ('filename'[,'filename']...)
    | 'filename'}]...
    | DROP LOGFILE MEMBER
    'filename'[,'filename']...
- **Exemple**
  - ALTER DATABASE DBCOURS
    ADD LOGFILE GROUP 4
    (' **'/oracle/oradata/DBCOURS/disk1/log*4a*dbcours.dbf'**)
    size 500K;
  - ALTER DATABASE DBCOURS
    ADD LOGFILE MEMBER
    '/oracle/oradata/DBCOURS/disk2/log*4b*dbcours.dbf' TO
    GROUP 4;
  - ALTER DATABASE DBCOURS
    ADD LOGFILE MEMBER
    '/oracle/oradata/DBCOURS/disk1/log*4a*dbcours.dbf' TO
    '/oracle/oradata/DBCOURS/disk3/log*4c*dbcours.dbf';

4.1.2 Les Fichiers REDO LOG

- **Forcer un checkpoint**
  - pour rendre le groupe courant inactif en cas de problème
  - pour des besoins de maintenance
  - pour provoquer un archivage immédiat si le REDO LOG est trop grand
  - Commande à exécuter (privilège requis : ALTER SYSTEM)
    - ALTER SYSTEM SWITCH LOGFILE ;
- **Forcer un Checkpoint sans Switch**
  - LGWR continue d'écrire dans le REDO LOG courant
  - La commande à exécuter est :
    - ALTER SYSTEM CHECKPOINT ;
- **Informations sur les REDO LOG**
  - vlog , vlogfile, v$log_history, v$thread

Si un fichier Redo d'un groupe ou un groupe entier est perdu, que se passe - t - il pour l'instance qui tourne ?

4.1.2 Les Fichiers REDO LOG

■ Informations sur les REDO LOG

v$logfile : table contenant les noms des fichiers Redo Log et leur état.

Name	Type	Description
GROUP#	NUMBER	Numéro du groupe redo Log
STATUS	VARCHAR2(7)	INVALID: fichier inaccessible STALE : contenu incomplet DELETED : fichier supprimé blanc : En utilisation
TYPE	VACHAR2(7)	ONLINE : actif, …
MEMBER	VARCHAR2(513)	Nom du fichier membre

Select group, status, member From v$logfile;

Group#	Status	MEMBER
1		C:\ORADATA\DBCOUR\DISK1\LOG1ACOUR.DBF
1		C:\ORADATA\DBCOUR\DISK2\LOG2ACOUR.DBF
2	STALE	C:\ORADATA\DBCOUR\DISK1\LOG1BCOUR.DBF
2	STALE	C:\ORADATA\DBCOUR\DISK2\LOG2BCOUR.DBF
3		C:\ORADATA\DBCOUR\DISK1\LOG1CCOUR.DBF
3		C:\ORADATA\DBCOUR\DISK2\LOG2CCOUR.DBF

6 rows selected.

4.1.2 Les Fichiers REDO LOG

■ Informations sur les REDO LOG

v$log : Vue contenant les informations issues du fichier de contrôle sur les fichiers Redo Log.

Name	Type	Description
GROUP#	NUMBER	Numéro de groupe de redo log
THREAD#	NUMBER	Numéro du Thread d'un fichier log
Sequence#	NUMBER	Numéro de séquence d'un fichier log
BYTES	NUMBER	Taille du fichier log en bytes
MEMBERS	NUMBER	Nombre de membre dans un group de Log
ARCHIVED	Varchar2	Fichier en Archivage ? :YES, NO
STATUS	Varchar2	UNUSED : Fichier non utilisé (il vient d'être ajouté, il ya eu un resetlogs) CURRENT :Fichier courant et actif ACTIVE : log actif mais pas courant CLEARING : Recréation à la suite d'un Alter Database Clear Logfile CLEARING_CURRENT : effacement du log courant INACTIVE : log non utile pour le recouvrement
first_change#	NUMBER	Plus petit SCN dans le fichier Log
First_time	DATE	Date du 1er SCN dans le fichier Log

4.1.2 Les Fichiers REDO LOG

■ Informations sur les REDO LOG

v$log_history : Vue contenant les informations sur l'historique des fichiers Log

Name	Type	Description
THREAD#	NUMBER	Numéro de thread du log archivé
Sequence#	NUMBER	Numéro de séquence du log archivé
First_time	DATE	Date du plus petit SCN dans le fichier Log
first_change#	NUMBER	Plus petit SCN dans le fichier log
next_change#	NUMBER	Plus grand SCN dans le fichier log.
RECID	NUMBER	Nr. d'enregistrement dans le fichier de controle
STAMP	NUMBER	Controlfile record stamp

4.1.2 Les Fichiers REDO LOG

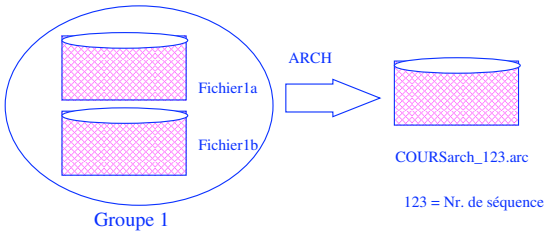
■ Informations sur les REDO LOG

V$THREAD: vue contenant des informations issues du fichier de contrôle sur les Log Thread.

Name	Type	Description
THREAD#	NUMBER	numéro du Thread
STATUS	Varchar2	status du Thread: OPEN, CLOSED
ENABLED	Varchar2	Enabled status: DISABLED, (enabled) PRIVATE, or (enabled) PUBLIC
ENABLE_		
CHANGE#	NUMBER	SCN at which thread was enabled
ENABLE_TIME	DATE	Time of enable SCN
disable_change#	Number	SCN at which thread was disabled
DISABLE_TIME	DATE	Time of disable SCN
GROUPS	NUMBER	Nombre de groupe redo affectées à ce thread
INSTANCE	Varchar2	Nom de l'instance
OPEN_TIME	DATE	Dernière date de lancement du thread
Current_group#	Number	Groupe Redo Log courant
SEQUENCE#	NUMBER	Nr. De séquence du fichier Log courant
CHECKPOINT _CHANGE#	Number	SCN du dernier checkpoint
Cheickpoint_time	DATE	Date du dernier checkpoint

4.1.3 Les Fichiers d'Archives

- **présents si mode avec *Archives***
- **contiennent l'archivage des fichiers Redo log plein**
- **à la même taille que le fichier REDO LOG correspondant**



Si un groupe REDO LOG est plein ou si un CHECKPOINT est intervenu alors ARCH se met action.

4.1.3 Les Fichiers d'Archives

- **Localisation des fichiers d'Archives**
  - **log_archive_dest** = /dev/rmt0:100M
    ou /backup/DB1/COURSarch
    (jusqu'à Oracle8)
  - **log_archive_format** = %t_%s.arc
    ou t : THREAD, s : Séquence number
  - **log_archive_duplex_dest** : destination alternative jusqu'à la version 8 d'Oracle
  - **log_archive_min_succeed_dest**=1 à 5 depuis la version (1 à 2 avant).Ce paramètre indique le nombre de copie à faire lors de l'archivage d'un fichier Redo log
  - **log_archive_dest_state_n** (n de 1 à 10) = enabled ou Deffered ou Alternate. Permet de gérer l'état d'une destination de sauvegarde log_archive_dest_n
  - **log_archive_dest_n** (n de 1 à 10)=/backup/DB1/COURSarch

Que fait Oracle lorsque LGWR tente d'écraser un fichier en cours d'Archivage ?

4.1.4 Les fichiers de contrôle

- **Un fichier de contrôle contient des informations de contrôle d'une base de données Oracle**
- **Le contenu de ce fichier est :**
  - nom de la base
  - date ('timestamp') de création de la base
  - noms et localisation des fichiers de données et redo log
  - numéro de séquence du fichier redo log courant
  - informations sur les CHECKPOINT
- **Oracle recommande de mettre les fichiers de contrôle en miroir et de les localiser sur des disques différents. Attention aux performances**

Comment est nommé un fichier de contrôle ?
Si les fichiers de contrôle sont en miroir lequel Oracle lit - t - il ?
Que se passe - t - il si Oracle perd un des fichiers en miroir ?

4.1.4 Les fichiers de contrôle

- **Ajout de copies supplémentaires de fichiers de contrôles**
  - 1. Arrêter la base et sortir du logiciel d'Administration (Sqlplus)
    2. dupliquer sous l'OS le fichier de contrôle
    3.relancer le logiciel d'administration et redémarrer la base.
- **Backup d'un fichier de contrôle**
  - Sauvegarde en dupliquant un fichier de contrôle

    ALTER DATABASE
    BACKUP CONTROLFILE TO arch_ctl.dbf;
  - Sauvegarde sous forme de requête SQL dans le fichier de trace (ora_xxx.trc)

    ALTER DATABASE
    BACKUP CONTROLFILE TO TRACE;

4.1.4 Les fichiers de contrôle

- **Création d'un nouveau Fichier de contrôle**
  - pour augmenter MAXLOGFILES et MAXDATAFILES
  - en cas de perte de tous les fichiers REDO LOG
- **Les étapes**
  - 1. Lister les fichiers à sauvegarder
    - Les fichiers de données (vue dba_data_files)
    - les fichiers redo log (vues v$logfile)
  - 2. Arrêter la base et faire une sauvegarde complète
  - démarrer une nouvelle instance en NOMOUNT
  - créer un nouveau fichier fichier de contrôle
  - Modifier le fichier des paramètres de la base
  - Recouvrer la base si utile et redémarrer
- **La commande**
  - CREATE CONTROLFILE [REUSE][RESET]
    DATABASE nombase
    LOGFILE ...
    DATAFILE ...

4.1.4 Les fichiers de contrôle

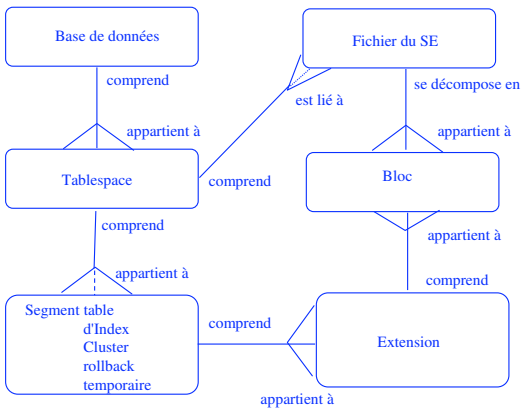
- **Informations sur les fichiers de contrôle**
  - Vues : v$controlfile, v$parameter, v$controlfile_record_section
  - Sql>select name, status from v$controlfile;

NAME	STATUS
C:\ORACLE\ORA9DATA\DBGM\DISK1\CONTROL01.CTL	
C:\ORACLE\ORA9DATA\DBGM\DISK2\CONTROL02.CTL	
C:\ORACLE\ORA9DATA\DBGM\DISK3\CONTROL03.CTL	

  - Sql>select name, value from v$parameter where name ='control_files'
  - Sql>Select * from v$controlfile_record_section;

4.2 Structure logique d'une Base de données Oracle

- **Les structures logiques servent à organiser le stockage et la gestion des données d'une base Oracle**
- **Corrélation entre les concepts logiques**



4.2.1 Les tablespaces

- **Généralités**
  - Un Tablespace est l'espace de stockage logique des données
  - Une Base de données Oracle comprend 1 à N Tablespaces. Un tablespace est composé de 0 ou N Fichiers OS
  - Un Tablespace **peut être mis OFFLINE**. Toutefois le Tablespace SYSTEM **ne peut être mis OFFLINE**
  - Un tablespace peut être **TEMPORARY ou PERMANENT** (mode par défaut). Un tablespace temporaire contient des données temporaires liés au TRI par exemple
  - Il maintenant possible de créer des tablespaces pour contenir les tables temporaires. CREATE TEMPORARY TABLESPACE nom_tablespace TEMPFILE ….Il peut aussi contenir les données de tri s'il est affecté à un utilisateur
  - Un tablespace **peut être réservé à contenir des données Rollback : UNDO TABLESPACE**

4.2.1 Les tablespaces

Généralités

- Un tablespace peut maintenant **être géré localement**. Pas de données dans le dictionnaire lors de la création de segment (mot clé LOCAL) : ATTENTION ! OPTION PAR DEFAUT
- Un tablespace peut aussi être depuis la Oracle 8 **transportable** en bases de données
- **Deux TABLESPACES** peuvent depuis Oracle 10G avoir **des blocs de taille différente**. Gérés dans des segments de SGA différents (Voir le chapitre concernant la SGA)

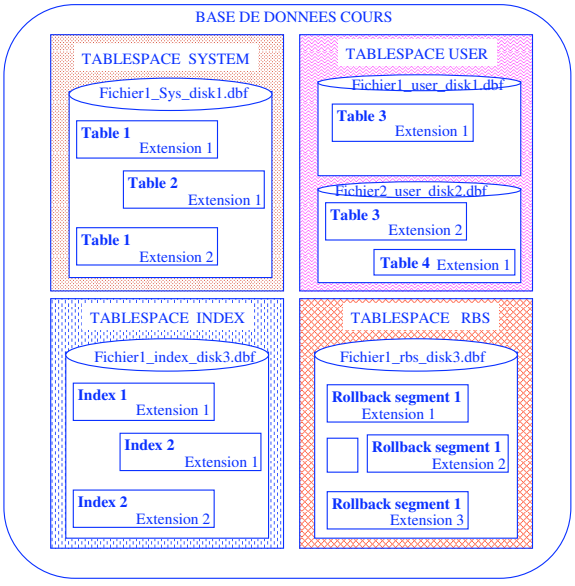
4.2.1 Les tablespaces

Rôle d'un Tablespace

- **Organiser logiquement le stockage des données** (localisation d'une application dans un Tablespace, cantonnement des utilisateurs dans un tablespace)
- **Contrôler l'allocation des espaces disques** au utilisateurs via des quotas
- **Augmenter la disponibilité des données** (arrêt partiel d'une base via l'arrêt d'un Tablespace)
- **Sauvegarder et restaurer** partiellement une base
- **Améliorer les performances** grâce à la distribution des informations sur des disques différents (index dans tablespace et les tables dans un autre)
- **Fournir les paramètres de stockages** des données par défaut

4.2.1 Les Tablespaces

Exemple d'Organisation d'une base



4.2.1 Les Tablespaces

Le tablespace SYSTEM

- Tablespace créé par défaut lors de la création de la base
- contient le dictionnaire de données
- contient un Rollback segment créé par défaut SYSTEM
- ne peut être mis OFFLINE
- il est déconseillé d' y stocker des données autres que celles du dictionnaire

Que doit-t-on faire pour renommer un fichier dans le tablespace System ?

4.2.1 Les Tablespaces

Les autres tablespaces

- **permettent de regrouper les données par type** d'applications ou d'activités (exemple le tablespace TOOLS, le tablespace COMPTA, ...)
- permettent de **séparer les données** pour augmenter les performances
- sont **par défaut ONLINE** à la création
- peuvent être mis OFFLINE
- peuvent être **sauvegardés et restaurés sans arrêter** la base entière
- sont à créer explicitement par l'administrateur
- peuvent être mis **READ ONLY** (à partir Oracle 7.2)

4.2.1 Les Tablespaces

La commande CREATE TABLESPACE

- CREATE [UNDO] [TEMPORARY] TABLESPACE tablespace DATAFILE filespec
  [autoextent_clause]
  [, filespec [autoextent_clause]] ...
  [DEFAULT STORAGE storage_clause]
  [BLOCKSIZE integer [K]]
  [ONLINE | OFFLINE]
  [PERMANENT | TEMPORARY]
  [extents_management_clause]
  [segments_management_clause]

4.2.1 Les Tablespaces

La commande CREATE TABLESPACE

- Les clause FileSpec et auto Extents
  - Valide pour tout type de tablespace
  - *FileSpec* ::=
    'Nomfichier' [SIZE *entier* [K|M] [REUSE]
  - autoextent_clause ::= AUTOEXTEND [OFF | ON] [NEXT integer [K | M] [MAXSIZE UNLIMITED | integer [K | M]]]
    - La clause auto extent permet d'étendre dynamiquement la taille d un fichier

4.2.1 Les Tablespaces

La commande CREATE TABLESPACE

- La clause storage
  - Valide uniquement pour les Tablespaces gérés dans le dictionnaire. Les informations sur le tablespace sont stockées dans le dictionnaire d'Oracle
  - storage_close ::=
  - STORAGE ([INITIAL integer [K | M]]
    [NEXT integer [K | M]]
    [PCTINCREASE integer]
    [MINEXTENTS integer]
    [MAXEXTENTS integer]

	VALEUR/ DEFAUT	MIN	MAX
INITIAL	5 blocs	2 blocs(3 si LM)	OS
NEXT	5 blocs	1 bloc	OS
MINEXTENT	1	1 (2 pour RBS)	OS
MAXEXTENT	dépend taille bloc	1	
PCTINCREASE	50%	0	OS

LM : Local Management

4.2.1 Les Tablespaces

La commande CREATE TABLESPACE

- Les clauses Extents management et segment management.
  - Clauses valides uniquement pour les tablespaces gérés localement
  - extents_management_clause ::=
    EXTENT MANAGEMENT {DICTIONARY |
    LOCAL [{AUTOALLOCATE |
    UNIFORM [SIZE integer [K | M]}]}
  - segments_management_clause ::=
    SEGMENT SPACE MANAGEMENT {MANUAL | AUTO}
  - Si AUTO spécifié alors les blocs libre pour l'insertion sont gérés via des Bitmap autrement ils sont gérés via les FREE LISTS

4.2.1 Les Tablespaces

Création des tablespaces gérés dans le dictionnaire

- **Généralités**
  - Les informations sur les extensions des segments sont gérées dans le dictionnaire
  - C'est l'approche historique de création de tablespaces sous Oracle jusqu'à la version 8i
- Les extensions des segments peuvent avoir des tailles différentes
- Les extensions sont définies par le DBA en s'appuyant sur les clauses DEFAULT STORAGE du tablespace ou STORAGE d'un segment
- Ce type de tablespace à partir de la version 9i ne peut être crée que si le tablespace System est lui même géré dans le dictionnaire

4.2.1 Les Tablespaces

Création des tablespaces gérés dans le dictionnaire

- **Exemple 1**
  - CREATE TABLESPACE APPLI_COMPTA
    DATAFILE '/home/Fichier1_app_cpt.dbf' SIZE 10M
    AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED
    EXTENT MANAGEMENT DICTIONARY;
  - CREATE TABLESPACE APPLI_PERSONNEL
    DATAFILE
    '/home/Fichier1_app_pers.dbf' SIZE 100M,
    '/home/Fichier2_app_pers.dbf' SIZE 10M
    DEFAULT STORAGE (INITIAL 100K
    NEXT 50K MINEXTENTS 4
    MAXEXTENTS 100 PCTINCREASE 25)
    EXTENT MANAGEMENT DICTIONARY;

Ou sont localisés les fichiers du tablespace si le chemin n'est pas donné?

Combien de fichiers peut avoir un tablespace au maximum?

Quels sont les paramètres de stockage du tablespace APPLI_COMPTA ?

Quels sont les critères de choix deparamètres de stockage d'un tablespace?

4.2.1 Les Tablespaces

Création de tablespaces gérés localement

- **Généralités**
  - Un tablespace géré localement prend en charge la gestion de ses extensions
  - une Bitmap est créé dans l'entête de chaque fichier pour identifier les Blocks ou les Extensions
  - Option par défaut d'Oracle
- La **taille d'une extension ici peut être calculée automatiquement** (AUTOALLOCATE) on peut être fixe (UNIFORM SIZE…)
- Si l'on utilise la clause **AUTOALLOCATE** Oracle détermine la taille optimale des extensions de chaque segment avec un minimum de 64K. **Vous pouvez préciser la taille de l'extension INITIALE lors de la création du segment**. Oracle s'en sert pour calculer la taille des extensions du segment
- Si l'on utilise la clause **UNIFORM SIZE** toutes les extensions de tous les segments auront la même taille : celle spécifiée lors de la création du tablespace ou celle part défaut qui est de 1 Mo

4.2.1 Les Tablespaces

■ Création de tablespaces gérés localement

- **Avantages des Tablespaces gérés localement** contrairement aux tablespaces gérés via le Dictionnaire :
  - pas de requêtes récursives
  - défragmentation(Coalesce) automatique du Tablespace
- NOTES :
  - ***Les paramètres de stockage NEXT, PCTINCREASE, MINEXTENTS, MAXEXTENTS, DEFAULT STORAGE sont ignorés pour les Tablespaces gérés localement***
  - ***Avec la clause UNIFORM SIZE, INITIAL n'est pas en plus pris en compte***

4.2.1 Les Tablespaces

■ Création de tablespaces gérés localement

- **Exemple 2 :** Création d'un tablespace Local avec la clause AUTOALLOCATE (option par défaut)

```
Sql>CREATE TABLESPACE tslocal
DATAFILE
'F:\oracle\oradata\dbtests\TSLOCAL\ts_local1.dbf'
SIZE 10M  EXTENT MANAGEMENT LOCAL
AUTOALLOCATE;

CREATE TABLE TestLocal(c1 char(4) )
tablespace tslocal
storage(initial 500k next 500K minextents 4
        maxextents 10);
```

SEGMENT_NAME	EXTENT_ID	BYTES	BLOCKS
TESTLOCAL	0	1048576	128
TESTLOCAL	1	1048576	128

4.2.1 Les Tablespaces

■ Création de tablespaces gérés localement

- **Exemple 3 :** Création d'un tablespace Local avec la clause UNIFORM SIZE

```
Sql>CREATE TABLESPACE tslocaluniform
DATAFILE
'F:\oracle\oradata\dbtests\TSLOCAL\ts_local_unif1.dbf'
SIZE 10M  EXTENT MANAGEMENT LOCAL
UNIFORM SIZE 200K;
CREATE TABLE Testuniform1(c1 char(4))
tablespace tslocaluniform
storage(initial 100k next 50K minextents 1 maxextents 2);

select SEGMENT_NAME , EXTENT_ID, BYTES, BLOCKS
from dba_extents
where      SEGMENT_NAME ='TESTUNIFORM1';
```

SEGMENT_NAME	EXTENT_ID	BYTES	BLOCKS
TESTUNIFORM1	0	204800	25

4.2.1 Les Tablespaces

■ Création de Tablespaces ayant sa propre taille de bloc

- Ne peut fonctionner qu'avec des SGA multiples
- **Exemple 4 :** Création d'un tablespace avec sa propre taille de bloc.

```
SQL>CREATE  TABLESPACE ts_block_2K
   DATAFILE
   'D:\oracle\ora9data\DBGM\TSBLK_2K\ts_temp_tab01.
   dbf' SIZE 10M
   BLOCKSIZE 2K;
```

4.2.1 Les Tablespaces

Modification d'un Tablespace

```
– ALTER TABLESPACE    tablespace
  { ADD DATAFILE
                    filespec [autoextent_clause]
                    [ , filespec [autoextent_clause] ] ...
        | RENAME  DATAFILE
                    'filename'[,'filename'] ...
                    TO 'filename' [,'filename'] ...
        | DEFAULT STORAGE  storage_clause
        | ONLINE
        | OFFLINE  {          NORMAL
                              | TEMPORARY
                              |  IMMEDIATE }
         |{ BEGIN  | END } BACKUP }
  | [READ {ONLY | WRITE}]
  | [PERMANENT]
  | [TEMPORARY]
  | [COALESCE]
```

- **Ajout d'un fichier**
  – ALTER TABLESPACE APPLI_COMPTA ADD DATAFILE '/home/Fichier1_app_cpt.dbf' SIZE 10M;
- **Modification des paramètres de stockage**
  – ALTER TABLESPACE APPLI_COMPTA default storage (initial 15K next 10K);

4.2.1 Les Tablespaces

Modification d'un tablespace

- **Rélocalisation des fichiers**
  ALTER TABLESPACE appli_compta
  RENAME DATAFILE '/home/Fichier1_app_cpt.dbf'
  TO '/new_home/Fichier1_app_cpt.dbf'
- **Arrêt d'un tablespace**
  ALTER TABLESPACE appli_compta OFFLINE;
- **Arrêt d'un fichier dans un tablespace**

  - avec archive
  ALTER TABLESPACE appli_compta
  '/new_home/Fichier1_app_cpt.dbf'
  OFFLINE;
- **Un Tablespace devient temporaire**
  ALTER TABLESPACE appli_compta TEMPORARY;
- **Compacter les extensions d'un Tablespace**
  ALTER TABLESPACE appli_compta COALESCE;

4.2.1 Les Tablespaces

Modification d'un tablespace (suite)

- Un tablespace arrêté peut être mis :
  – **offline normal**
  – **offline temporary** (il y a des erreurs et un Checkpoint est passé sur les fichiers sans erreurs)
  – **offline immediate** (il y a des erreurs et aucun Checkpoint n'est passé mode avec ARCHIVE)
- Un tablespace peut être mis *read-only* (oracle 7.2)
  ALTER TABLESPACE appli_compta READ ONLY

Quand doit - t - on mettre un fichier OFFLINE ou ONLINE ?

Pourquoi l'arrêt d'un fichier nécessite l'option DROP en mode sans archive ?

Quelles sont les conditions pour rendre un tablespace read-only?

Que doit-t-on faire pour renommer un fichier dans le un tablespace tablespace autre que System ?

4.2.1 Les Tablespaces

Suppression d'un Tablespace

- Tout tablespace peut être supprimé sauf le tablespace SYSTEM

  DROP TABLESPACE APPLI_COMPTA
  INCLUDING CONTENTS ;

  Si les données du tablespace sont référencées alors

  – DROP TABLESPACE APPLI_COMPTA
  INCLUDING CONTENTS
  CASCADE CONSTRAINTS ;

4.2.1 Les Tablespaces

- **Vues contenant des informations sur les Tablespaces et les fichiers**
  - Vues liées aux extensions : *User_extents, dba_extents*
  - Vues liées aux Segments : *User_segments, dba_segments, v$sort_segment*
  - Vues liées aux espaces libres dans les fichiers : *user_free_space, dba_free_space*
  - vues liées aux utilisateurs : *dba_users, v$sort_user*
  - vues liées aux quotas : *dba_ts_quotas*
  - vues liées au tablespaces : *user_tablespaces, dba_tablespaces, v$tablespace*
  - vues liées aux fichiers permanents ou temporaires : *dba_data_files, v$datafiles, v$tempfile, dba_temp_files, v$temp_extent_map, v$temp_extent_pool*

4.2.2 Les Segments et leurs composants

- **Objets du schéma**
  - Le nom d'un schéma équivaut à celui d'un utilisateur
  - **les objets suivants appartiennent à un schéma** : les tables, les vues, les séquences, les synonymes, les indexes, les procédures, les fonctions, les packages, les snapshots, les index, les contraintes, les clusters, les database links, les triggers, les types, …
  - **des segments sont associés aux objets du schéma**. On distingue :
    - des segments de données (TABLES, CLUSTERS)
    - des segments d'index (INDEX)
    - des segments temporaires
    - des segments ROLLBACK
    - le segment de démarrage (contient le dictionnaire)

4.2.2 Les Segments et leurs composants

- **Objets du schéma (suite)**
  - un **segment est un ensemble d'extensions**
  - une **extension est un ensemble de blocs** contiguës
  - les **paramètres de stockage données** et les **paramètres de dimensionnement des blocs** de données permettent de contrôler l'allocation d'espaces à un segment.

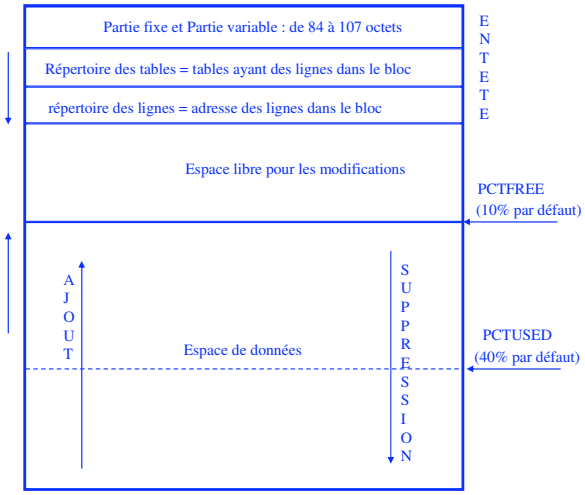
4.2.2 Les Segments et leurs composants

- **Le bloc**
  - un bloc est la plus petite unité de stockage et d'échange
  - la *taille d'un bloc* Oracle dépend de l'OS (2K, 4K, 8K, 16K, 32K)
  - un bloc comprend une *entête et une zone pour stocker les enregistrements*
  - les ***paramètres*** suivants permettent de contrôler la taille d'un bloc en insertion ou en mise à jour :
    - **PCTFREE** : 10 % par défaut, réserve l'espace pour les mises à jour, permet un meilleur remplissage en insertion, ...
    - **PCTUSED** : 40% par défaut, lorsqu'un bloc est plein, les insertions ne peuvent y reprendre que si PCTUSED <= 40%
    - **INITRANS** : Nombre de transactions pré-allouées pouvant être actives dans un bloc, par défaut 1 pour un segment de table et 2 pour un d'index ou de cluster.

4.2.2 Les Segments et leurs composants

Le bloc

- Format d'un bloc (table, index, cluster)



D'ou vient la différence de 23 octets dans l'entête ?

4.2.2 Les Segments et leurs composants

Le bloc

- Pour les tablespaces locaux dont l'espace dans les segments est géré automatiquement, les blocs libres sont identifiés via les bitmaps. PCTUSED, FREELIST, GROUP LIST ne sont pas utilisés

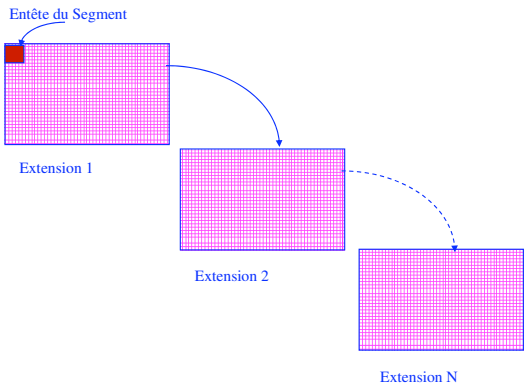
CREATE TABLESPACE tslocaluniformauto
DATAFILE
'D:\oracle\oradata\dbtp92\TSLOCAL\ts_local_unif3.dbf'
SIZE 50M
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 200K
SEGMENT SPACE MANAGEMENT AUTO;

NOTE : l'option par défaut est
SEGMENT SPACE MANAGEMENT *MANUAL*;

4.2.2 Les Segments et leurs composants

Une extension

- Ensemble de blocs contiguës appartenant à un segment
- est alloué soit statiquement (à la création de la table) soit dynamiquement



4.2.2 Les Segments et leurs composants

Une extension

- Les *paramètres de stockage* permettent de contrôler le nombre d'extensions pour un segment
- le *nombre maximum d'extensions* allouables à un segment dépend de la taille du bloc (jusqu'à Oracle 7.2). Il peut maintenant être illimité.
- *mode d'allocation des extensions DICTIONARY*
  - 1.1 Oracle recherche les blocs contiguës équivalents à la demande
  - 1. 2 Si pas trouvé, Oracle recherche les blocs contiguës > ou = à la demande
  - 1.3 si pas trouvé, Oracle regroupe les blocs libres proches dans le tablespace (SMON)
  - 1.4 si pas trouvé, erreur.
  - 2. si trouvé, Oracle alloue l'extension
  - 3. Oracle met à jour l'entête du segment et le dictionnaire de données si **nécessaire**

4.2.2 Les Segments et leurs composants

Les types de données Oracle

- La création d'une table ou d'un cluster, ... nécessite la connaissance du **volume de la table** ou du cluster

Les types entre autre permettent d'évaluer la taille d'une table ou d'un cluster

Type de données	Description	Longueur
char(taille)	taille fixe	1 à 255 bytes
varchar2(size)	taille variable	1 à 2000 bytes
number(p,s)	taille variable	21bytes max
date	taille fixe	7 bytes
LONG	taille Variable	2Go
BLOB	**taille Variable**	**4Go**
CLOB	**taille Variable**	**4Go**
BFILE		
RAW(size)	taille variable	jusqu'à 255 bytes
LONG RAW	taille variable	2Go
ROWID	Binaire	6 bytes
MLSLABEL	trusted Oracle	2 à 5 bytes
REF	**taille fixe**	**42 octets**

4.2.2 Les Segments et leurs composants

Les types de segments : récapitulatif

- Les segments de données :
  - Segments de données de type table
  - segments de données de type cluster
- Les segments d'index (index Btree, bitmap, Table Oraganisé dans un Index)
- les segments temporaires
- les segments rollback ou d'annulation

4.2.2 Les Segments et leurs composants

Les segments de données

- Ce sont des segments de type table et cluster(indexé ou haché)
- Permettent de répartir les données d'une table ou d'un cluster dans les fichiers du tablespace
- Seront étudiés selon les points de vues suivants :
  - création du segment
  - modification du segment
  - suppression du segment
  - dimensionnement du segment

4.2.3 Les Segments de données de type table

création du segment

```
{Relational_table_clause | Object_table_clause}

Relational_table_clause :: =

CREATE [GLOBAL TEMPORARY] TABLE [ schema. ] table
   ( { column datatype [ DEFAULT expr ] [ column_constraint ] ...
     | table_constraint } ...)
   [ [ PCTFREE integer ] [ PCTUSED integer ]
     [ INITRANS integer ] [ MAXTRANS integer ]
     [ TABLESPACE tablespace ]
     [ STORAGE (
     [ INITIAL                  integer [ K | M ] ]
               [ NEXT           integer [ K | M ] ]
               [ PCTINCREASE    integer ]
               [ MINEXTENTS             integer ]
               [ MAXEXTENTS integer ]
               [ FREELIST GROUPS        integer ]
               [ FREELIST     integer ] ) ]
   |  [ CLUSTER cluster (column [, column ] ... ) ] ]
     [ ENABLE enable_clause
     | DISABLE disable_clause ] ...
     [ AS subquery ]
     [Large_Object_clause]
     [Partition_clause]
     [ORGANIZATION {HEAP segment_attr_clause
                  | INDEX segment_attr_clause
                  | EXTERNAL ext_clause }
     ]
```

NOTES :

1. Un segment de type table est créé de façon implicite lors de la création de la table.
2. Les tables objets, les partitions, les LOB sont traités dans un autre cours

4.2.3 Les Segments de données de type table

Création du segment (suite)

Elément de la clause	description
déclaration_colonne	nom/type/contrainte niveau colonne
column_constraint	contrainte au niveau colonne
storage_clause	clause de définition des paramètres de stockage
enable_clause	clause d'activation des contraintes
disable_clause	clause de désactivation des contraintes
PCTFREE	Espace libre pour les modifications
PCTUSED	Seuil au dessous duquel les insertions sont à nouveau autorisées
INTITRANS	Nbre d'entrée de transactions réservée à la création d'un bloc(1 par défaut)
MAXTRANS	Nbre de transaction max pouvant être actives dans un bloc
FREELISTS	Nbre de Listes de blocs utilisable en insertion
FREELISTS GROUP	Nbre de groupes de listes de blocs utilisable en insertion (option parallèle)
Large_object_clause	Clause pour localisation des objets volumineux
Partition_Clause	Clause pour le partitonnement des tables
ORGANIZATION index	Table orgnisée dans un index
ORGANIZATION heap	Table orgnisée sans ordre particulier. Valeur par défaut
ORGANIZATION external	Table orgnisée dans un fichier en dehors d'Oracle.

4.2.3 Les Segments de données de type table

Création d'un segment (suite)

Exemple d'un segment géré dans le dictionnaire:

```
CREATE TABLE scott.emp
  ( empno     NUMBER    CONSTRAINT pk_emp PRIMARY KEY,
    ename     VARCHAR2(10)
                        CONSTRAINT nn_ename NOT NULL
                        CONSTRAINT upper_ename
                        CHECK (ename = UPPER(ename)),
    job       VARCHAR2(9),
    mgr       NUMBER            CONSTRAINT fk_mgr
                                REFERENCES scott.emp(empno),
    hiredate  DATE              DEFAULT SYSDATE,
    sal       NUMBER(10,2)      CONSTRAINT ck_sal  CHECK (sal > 500),
    comm      NUMBER(9,0) DEFAULT NULL,
    deptno    NUMBER(2)         CONSTRAINT nn_deptno NOT NULL
                                CONSTRAINT fk_deptno REFERENCES
                                scott.dept(deptno)
  )
  PCTFREE 5 PCTUSED 75
  STORAGE(initial 100K  next 50K minextents 3 maxextents 5
    pctincrease 50)
  TABLESPACE TS_USERS
  EXTENT MANAGEMENT DICTIONARY;
```


4.2.3 Les Segments de données de type table

Création d'un segment (suite)

Exemple (suite) :

Extensions allouables au segment de la table ***scott.emp***

Nr. Extension	Taille extension	Valeur de l'extension suivante
1 (initial/min 1)	100K	50K
2 (min 2)	50K	75K=1,5*50K
3 (min 3)	75K	112,5K=1,5*75K
4	112,5K	168,75=1,5*112,5K
5	168,75	253,125=1,5*168,75K

4.2.3 Les Segments de données de type table

Modification du segment

```
ALTER TABLE [ schema.] table
  [ ADD {column datatype [ DEFAULT  EXPR ] [ column_constraint ] ...
    | table_constraint ]}
  [ MODIFY { column  [ datatype ] [ DEFAULT  EXPR ] [ column_constraint ] ...}]
  [ PCTFREE  integer ] [ PCTUSED integer ]
  [ INITRANS  integer ] [ MAXTRANS integer ]
  [ STORAGE  storage_clause ]
  [ DROP drop_clause ] ...
  [ ALLOCATE  EXTENT [ (  [ SIZE  integer [ K | M ]  ]
        [ DATAFILE 'filename' ]
        [ INSTANCE integer ]  ) ]
  {[ ENABLE   enable_clause
    | DISABLE   disable_clause ] ...}
```

4.2.3 Les Segments de données de type table

■ Modification d'un segment (suite)

clause	Description
add_clause	Ajout d'une colonne ou d'une contrainte niveau table
modify_clause	modification d'une colonne
storage_clause	clause de définition des paramètres de stockage
enable_clause	clause d'activation des contraintes
disable_clause	clause de désactivation des contraintes
PCTFREE	Espace libre pour les modifications
PCTUSED	Seuil au dessous duquel les insertions sont à nouveau autorisées
INTITRANS	Nbre d'entrée de transactions réservée à la création d'un bloc(1 par défaut)
MAXTRANS	Nbre de transaction max pouvant être actives dans un bloc
FREELISTS	Nbre de Listes de blocs utilisable en insertion
FREELISTS GROUP	Nbre de groupes de listes de blocs utilisable en insertion (option parallèle)
ALLOCATE EXTENT	Allocation explicite d'une extension dans un fichier donné (sert à distribuer une grande table sur plusieurs fichiers)
drop_clause	Clause de suppression d'une contrainte d'intégrité

4.2.3 Les Segments de données de type table

■ Modification d'un segment (suite)

Exemple

```
ALTER TABLE scott.emp
   STORAGE( NEXT 300K
            MAXEXTENT 10
            PCTINCREASE 10)
   ALLOCATE EXTENT (SIZE 400K
                    DATAFILE 'user_data1.dbf') ;
```

Pourquoi n'est -t -il pas possible de modifier INITIAL et MINIEXTENTS ?

Quels sont les différentes exploitations possibles de la clause ALLOCATE EXTENT ?

4.2.3 Les Segments de données de type table

■ suppression d'un segment

- trois approches pour supprimer les données d'un segment :
  - DROP TABLE : supprime les données et le segment

    DROP TABLE [schema.] table
    [CASCADE CONSTRAINTS]
  - TRUNCATE TABLE : supprime les données et libère à la demande l'espace alloué au segment

    TRUNCATE { TABLE [schema.] table
    | CLUSTER [schema.] CLUSTER }
    [{ DROP | REUSE } STORAGE]
  - DELETE FROM : supprime toutes les données d'une table mais l'espace alloué au segment est conservé.

4.2.3 Les Segments de données de type table

■ suppression du segment (suite)

	Avantages	Inconvénients
Drop table	Lorsque l'on ne veut plus de la table	Supprime les index les contraintes
truncate table	pas de RBS conservation des Contraintes, triggers, grant pas d'exécution de trigger, gel des Contraintes	Pas de retour arrière
delete from	un rollback reste possible	l'utilisation du RBS peut être très long

4.2.3 Les Segments de données de type table

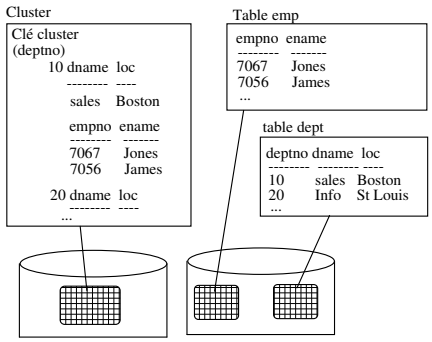
dimensionnement du segment

Voir Annexe A10

4.2.4 Les Segments de données de type cluster

Généralités

- un cluster permet de stocker dans un même bloc les données de 2 ou plusieurs tables
- Favorise certains chemins d'accès (jointure physique)
- un cluster peut être index ou hashé



4.2.4 Les Segments de données de type cluster

création d'un segment de cluster

```
CREATE CLUSTER [schéma.]cluster_name
                (column datatype [,column datatype] ...)
                [PCTUSED integer] [ PCTFREE integer]
                [SIZE integer][K | M]
                [INITRANS integer] [MAXTRANS integer]
                [TABLESPACE tablespace_name]
                [STORAGE storage_clause]
                [INDEX]
                [[HASH IS colonne] HASHKEYS integer]
```

clause	Description
SIZE	taille d'un bloc logique pour le cluster
HASH IS	Colonne qui joue le rôle de fonction de hashage
HASHKEYS	Nombre de valeurs de clés

4.2.4 Les Segments de données de type cluster

création d'un segment de cluster (suite)

Exemple de création d'un cluster indexé

a)

```
create cluster cluster_emp_dept (deptno number(4))
  pctused 80
  pctfree 20
  size 600
  tablespace student95
  storage (   initial 200K
              next 300K minextents 2
              maxextents 20
              pctincrease 33));
```

b)

```
create table emp(...) cluster cluster_emp_dept (deptno);
create table dept(...) cluster cluster_emp_dept (deptno);
```

c)

```
create index idx_emp_dept  on cluster cluster_emp_dept ;
```

Exemple de création d'un cluster haché

```
a) create cluster cluster_dept_emp (deptno number(3))
   SIZE 2K          HASHKEYS 1000;
b) create table dept (deptno number(3), ...)
   CLUSTER cluster_dept_emp;
```

4.2.4 Les Segments de données de type cluster

- **Modification d'un segment de cluster**

```
ALTER CLUSTER [schema.]cluster
  [PCTUSED integer] [PCTFREE integer]
  [SIZE integer [K | M] ]
  [INITRANS integer] [MAXTRANS integer]
  [STORAGE storage_clause ]
  [ALLOCATE  EXTENT [ ( [size integer [K|M]  ]
    [DATAFI LE'filename']
    [INSTANCE integer ]
```


4.2.4 Les Segments de données de type cluster

- **suppression du segment de type cluster**

a) suppression du segment de type cluster

```
DROP CLUSTER [ schema. ] cluster
        [ INCLUDING  TABLES [ CASCADE CONSTRAINTS ] ]
```

Exemple:

```
drop cluster cluster_dept_emp including tables ;
```

b) suppression du contenu d'un cluster

```
TRUNCATE { TABLE [ schema. ] table
          | CLUSTER [ schema. ] CLUSTER }
                [ { DROP | REUSE } STORAGE ]
```

Exemple :

```
TRUNCATE CLUSTER cluster_dept_emp DROP STORAGE ;
```


4.2.4 Les Segments de données de type cluster

- **dimensionnement du segment**

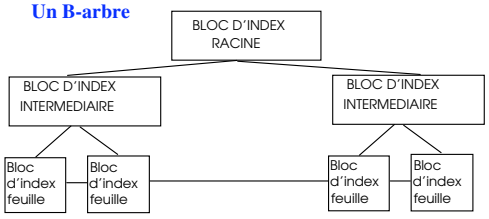
Voir Annexe A10

4.2.5 Les Segments d'index

- **Généralités**

  - Accélérateur de requête
  - peut être unique ou non unique
  - peut être créé explicitement ou implicitement (depuis Oracle7)
  - sa structure interne est un arbre balancé (B-arbre)

Un B-arbre



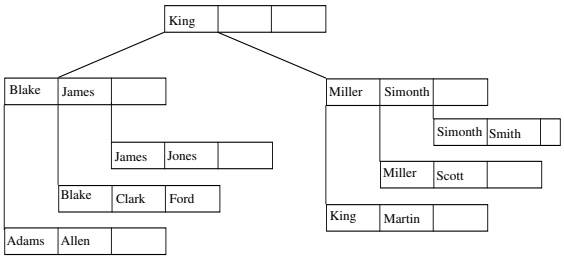
Est - t - il possible de contrôler la localisation d'un index implicite ?

Quels sont les conditions de choix d'un bon Index ?

4.2.5 Les Segments d'index

Le segment d'index

- Généralités (suite)



Notes :

A chaque feuille on trouve : la valeur clé et le ROWID (index unique) ou la liste des ROWID (Index non unique). Par exemple ADAMS '00000012.0002.0002 '.

4.2.5 Les Segments d'index

création d'un segment d'index

```
CREATE {BITMAP | INDEX } [ schema. ] index
  ON { [ schema. ] table ( column [ ASC | DESC ]
                  [, column [ ASC | DESC ] ] ...]
        | CLUSTER [ schema. ] cluster }
        [ INITRANS integer ] [ MAXTRANS integer ]
        [ TABLESPACE tablespace ]
        [ STORAGE storage_clause ]
        [ PCTFREE integer ]
        [ NOSORT ]
        [REVERSE]
        [PARTITION clause_de_partionnement]
```

- **Modification du segment**

```
ALTER INDEX [ schema. jindex
                  [ INITRANS integer ] [MAXTRANS integer ]
                  [ STORAGE storage_ clause ]
```

- **Suppression du segment**

```
DROP INDEX [ schema.] index
```


4.2.6 Les Segments temporaires

Généralités

- **Oracle créé automatiquement** ces segments si la zone de tri (paramètre SORT_AREA_SIZE) s'avère petite
- **plusieurs Ordres SQL provoquent des tris** : Group by, Ordrer by, Opérateurs ensemblistes, Distinct, Merge join, ...
- **pour augmenter les performances, il est utile de les séparer des segments de données**
- C'est à la création d'un Utilisateur Oracle qu'on lui indique son segment temporaire.
- Il est maintenant possible depuis la 7.3 de créer un tablespace temporaire pour requêtes nécessitant les tris

4.2.6 Les Segments temporaires

Exemple 1 : Création d'un tablespace pour stocker les données temporaires (tri)

- CREATE TABLESPACE temp_ts
  DATAFILE 'ts_temp_01.dbf' SIZE 20M
  TEMPORARY
  AUTOEXTEND ON;

Exemple 2 : Création d'un tablespace temporaire (pour stocker les tables temporaires voir les données de tri)

- SQL>CREATE **TEMPORARY** TABLESPACE temp_tab_ts
  TEMPFILE 'ts_temp_tab01.dbf' SIZE 20M
  AUTOEXTEND ON;

4.2.6 Les Segments temporaires

- **Exemple 3: Création d'une table temporaire de session (ligne supprimée à la fin de la session)**

```
CREATE GLOBAL TEMPORARY
        TABLE itineraire_vol (
        startdate DATE,
        enddate DATE,
        cost NUMBER)
ON COMMIT PRESERVE ROWS
TABLESPACE temp_tab_ts ;
```


4.2.7 Les Segments rollback manuels ou automatique

- **Généralités**
  - un **RBS** contient **l'état avant** des transactions en cours
  - il permet d'effectuer des *Lectures* consistantes, d'*annuler* les transactions et de *recouvrer* la base de données.
  - une **entrée Rollback** contient les informations sur le fichier Rollback, le numéro du bloc et l'état avant
  - les **entrées rollback** sont stockées aussi dans le buffer redo log puis dans le fichier Redo Log
  - En **cas de crash**, Oracle restaure d'abord les RBS
  - En **cas de problème**, Oracle annule les transactions non commitées
  - une **extension d'un RBS** peut contenir les blocs de N transactions : les blocs rollback d'une transaction sont chaînés.
  - *Depuis la version 9i **deux approches de création de RBS existent** : création manuelle ou automatique (Tablespace d'Annulation : TSA)*

4.2.7 Les Segments rollback manuels ou automatique

- **Création d'un RBS manuel**

Syntaxe

```
CREATE  [ PUBLIC ] ROLLBACK  SEGMENT  rollback_segment
        [ TABLESPACE  tablespace ]
        [ STORAGE  (              storage_clause
         [ OPTIMAL                integer  [ K | M ]  |  NULL ] )
```

Exemple

```
CREATE PUBLIC ROLLBACK SEGMENT r5
  TABLESPACE  ts_rbs
  STORAGE (INITIAL 100K MINEXTENTS 3 MAXEXTENTS 10
  NEXT 100K
  OPTIMAL 400K ) ;
```

PUBLIC : segment accessible par toutes les instances d'une même base

OPTIMAL : Taille optimale du RBS obtenue par Oracle par désallocation dynamique

NOTE :
par défaut un RBS créé est inactif

4.2.7 Les Segments rollback manuels ou automatique

- **Modification d'un RBS manuel**
  - Il est possible d'activer ou désactiver un RBS et de modifier ses paramètres de stockage
  - un Rollback *segment actif* ne peut être mis OFFLINE

Syntaxe

```
ALTER  ROLLBACK  SEGMENT   rollback_ segment
                { ONLINE
                | OFFLINE
                |  STORAGE   storage_clause  }
```

Exemple

```
ALTER ROLLBACK SEGMENT r5 ONLINE ;
```

4.2.7 Les Segments rollback manuels ou automatique

Suppression d'un RBS manuel

- Un segment avec des *transactions actives* ne peut être supprimé
- il est utile de *séparer* les segments RBS des segments de données
- Le RBS SYSTEM ne peut être ni arrêté ni supprimé.

Syntaxe

DROP ROLLBACK SEGMENT robllack_segment

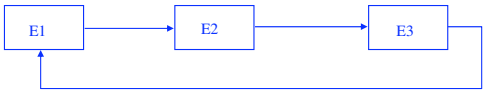
Exemple

DROP ROLLBACK SEGMENT r5;

4.2.7 Les Segments rollback manuels ou automatique

Allocation et libération d'extensions dans un RBS

- Un RBS doit avoir au **moins 2 extensions**
- un bloc d'une extension est affecté à une et une seule transaction
- **Règles d'allocation d'espace à une transaction**
  - dans l'extension courante tant qu'il y a de la place
  - dans l'extension suivante ayant de la place si l'extension courante est pleine (allocation cyclique)
  - dans une nouvelle extension dynamiquement allouée si les extensions déjà allouées sont pleines
- **Représentation schématique**



NOTE : Le paramètre OPTIMAL permet de désallouer dynamiquement une extension d'un RBS manuel

4.2.7 Les Segments rollback manuels ou automatique

Assignation implicite d'une transaction à RBS

- Le commencement d'une transaction entraîne son assignation à un RBS actif
- la règle d'assignation implicite d'une transaction est celle de la **répartition équitable des charges** entre RBS

Assignation explicite d'une transaction à RBS *manuel*

- une transaction peut être orientée vers un RBS créé manuellement particulier

SET TRANSACTION USE ROLLBACK SEGMENT nom_rollback_segment ;

4.2.7 Les Segments rollback manuels ou automatique

Activation et Estimation du nombre de RBS manuels

- **Les paramètres de init.ora**
  - TRANSACTIONS : Nombres maximum de transactions concurrentes(par défaut 1.1 * SESSIONS)
  - TRANSACTION_PER_ROLLBACK_SEGMENT : Nombre de transactions par Rollback segment (30 par défaut)
- **Activation automatique des RBS manuels**
  - Nombre de Transactions activées = CEIL(Transaction / Transaction_per_rollback_segment)
- **Activation explicite** des RBS (paramètre ROLLBACK_SEGMENTS dans le fichier d'initialisation init.ora ou par une commande SQL)
  - ROLLBACK_SEGMENTS=(r1, r2, r3, r4)
  - sql>ALTER ROLLBACK SEGMENT r1 ONLINE ;

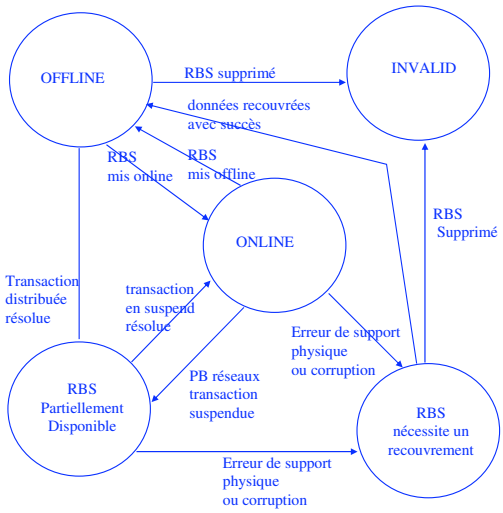
4.2.7 Les Segments rollback manuels ou automatique

■ Estimation du nombre de RBS manuels

N transactions simultanées	Nombre de RBS utiles
n <= 16	4
16<= n <= 32	8
32 <= n	nb= n / 4 avec nb < 50

4.2.7 Les Segments rollback manuels ou automatique

■ Etats d'un RBS



4.2.7 Les Segments rollback manuels ou automatique

■ Visualisation des RBS

- Les vues suivantes contiennent les informations sur les ROLLBACK SEGMENT
  - dba_rollback_segs, dba_segments, user_segments
  - v$waitstat, v$sysstat
  - v$rollname, v$transaction, v$session, v$rollstat (servent aussi au dimensionnement)

Exemples :

a) Informations sur les RBS

```
SELECT segment_name, tablespace_name, status FROM sys.dba_rollback_segs ;
SELECT segment_name, status FROM dba_rollback_segs;
```

SEGMENT_NAME	TABLESPACE_NAME	STATUS
SYSTEM	SYSTEM	ONLINE
RB_TEMP	SYSTEM	OFFLINE
RB1	ROLLBACK_DATA	ONLINE
RB2	ROLLBACK_DATA	ONLINE
RB3	ROLLBACK_DATA	ONLINE
RB4	ROLLBACK_DATA	ONLINE

4.2.7 Les Segments rollback manuels ou automatique

■ Visualisation des RBS (suite)

b) Affichage des informations sur les RBS à partir de la vue dba_segments

```
SELECT segment_name, tablespace_name, bytes, blocks, extents
        FROM dba_segments
        WHERE segment_type = 'ROLLBACK';
```

SEGMENT_NAME	TABLESPACE_NAME	BYTES	BLOCKS	EXTENTS
SYSTEM	SYSTEM	204800	100	4
RB_TEMP	SYSTEM	1484800	725	15
RB1	ROLLBACK_DATA	102400	50	2
RB2	ROLLBACK_DATA	102400	50	2
RB3	ROLLBACK_DATA	102400	50	2
RB4	ROLLBACK_DATA	102400	50	2

c) Transaction actives dans les RBS

```
SELECT distinct name, xacts "Transactions actives"
        FROM v$rollname vr, v$rollstat vs
        WHERE vs.usn = vr.usn  and xacts != 0;
```

NAME	Transactions actives
RB1	1

4.2.7 Les Segments rollback manuels ou automatique

■ Contention sur les RBS

La requête suivante permet d'accéder aux statistiques :

```
sql>  SELECT class, count
      FROM v$waitstat
      WHERE class IN ('system undo header', 'system undo block',
        'undo header', 'undo blocs');
```

class	count
System undo header	2089
system undo block	633
undo header	1235
undo block	942

Calculer le nombre de demandes de buffers

```
sql>SELECT sum(value) "Buffers demandés"
  FROM v$sysstat
  WHERE name in ('db block gets', 'Consistent Gets');
```

Buffers demandés

929530

Ratio par class ci = classi / sum(value)

Si le ratio est > 1% alors ajouter des RBS

4.2.7 Les Segments rollback manuels ou automatique

■ Les Tablespaces d'Annulations (TSA)

- Généralités
  - Tablespace **destinées à ne contenir que des données d'annulations.**
  - **On peut créer créer plusieurs TSA**, mais un et un seul sera associé à une instance
  - Des **RBS sont créés implicitement** lors de la création d'un TSA
  - La **gestion des segments d'annulation** peut maintenant être manuelle ou automatique (voir init.ora).

    UNDO_MANAGEMENT = AUTO | MANUAL

    MANUAL indique que vous devez vous même créés vos ROLLBACK SEGMENTS

4.2.8 Les Segments rollback automatiques

■ Les tablespaces d'Annulations (TSA)

- Création d'un TSA
  - A la création de la base

    CREATE DATABASE DBGM
    CONTROLFILE REUSE ...
    UNDO TABLESPACE undotbs_01 DATAFILE
    'c:\oracle\ora9data\dbgm\tsundo1\undo0101.dbf'
    size 10M AUTOEXTEND ON
  - Explicitement après la création de la base

    CREATE UNDO TABLESPACE undotbs_02
    DATAFILE
    'c:\oracle\ora9data\dbgm\tsundo2\undo0201.dbf'
    size 10M AUTOEXTEND ON
- Suppression d'un TSA
  - DROP TABLESPACE undotbs_02 ;

4.2.8 Les Segments rollback automatiques

■ Les tablespaces d'Annulations (TSA)

- Modification d'un TSA

  ALTER TABLESPACE undotbs_02 ADD DATAFILE
  'c:\oracle\ora9data\dbgm\tsundo2\undo0202.dbf'
  size 100M AUTOEXTEND ON MAXSIZE
  UNLIMITED
- Assignation d'un nouveau TSA

  a) Dans init.ora

  UNDO_TABLESPACE = undotbs_02
  UNDO_MANAGEMENT = AUTO

  b) Dynamiquement

  ALTER SYSTEM SET UNDO_TABLESPACE =
  undotbs_02;
- Fixer la période de rétention des données dans un TSA

  a) Dans init.ora

  UNDO_RETENTION = 100 (en seconde)

  b) Dynamiquement

  ALTER SYSTEM SET UNDO_RETENTION = 100;

4.2.8 Les Segments rollback automatiques

- **Les tablespaces d'Annulations (TSA)**
  - Vues contenant des informations sur les TSA
    - V$UNDOSTAT : Contains statistics for monitoring and tuning undo space.
    - V$ROLLSTAT For automatic undo management mode, information reflects behavior of the undo segments in the undo tablespace
    - V$TRANSACTION Contains undo segment information
    - DBA_UNDO_EXTENTS : Shows the commit time for each extent in the undo tablespace.

5. Gestion de la sécurité et des ressources

- **Plan**
  - 5.1 Généralités
  - 5.2 Les Privilèges
    - Introduction
    - Les privilèges Systèmes
    - Les privilèges Objets
  - 5.3 Les rôles
    - Intérêts des rôles
    - Création et suppression des rôles
    - Affectation des privilèges ou des rôles à un rôle
    - Sécurité des rôles
    - Les rôles du Système d'exploitation
    - Les rôles prédéfinis
  - 5.4 Les profiles
    - Intérêt des profiles
    - Création , Suppression et Modification de profiles
    - Utilisation des limites composites
    - Affectation d'un profile à un utilisateur

5. Gestion de la sécurité et des ressources

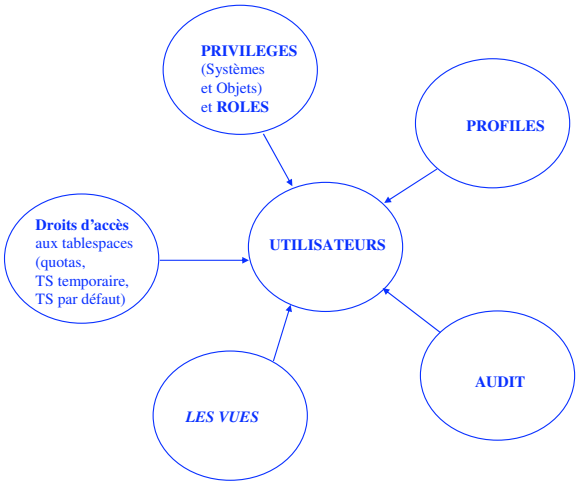
- **PLAN**
  - 5.5 Les utilisateurs
    - La politique de licences d'Oracle
    - Les Utilisateurs prédéfinis
    - Les différents mode d'authentification
    - Création d'un Utilisateur
    - Modification d'un Utilisateur
    - Suppression d'un Utilisateur
    - Affectation des droits à un Utilisateur
  - 5.6 L'audit
    - Intérêt de l'audit
    - Types d'audit et modes d'activation
    - Audit Système
    - Audit objet
    - La table aud$

5.1 Généralités

- **Rôle de l'Administrateur de sécurité et des ressources**
  - Définir une politique de sécurité
  - Faire les **choix du type de sécurité** : au niveau système, au niveau Oracle, au niveau Global (LDAP)
  - Gérer les **utilisateurs**
  - Gérer les ressources (**profiles**)
  - Assurer **l'affectation et le retrait** des droits
  - **Affiner la politique** de sécurité par l'utilisation des rôles
  - Effectuer les **audits**

5.1 Généralités

Moyens pour la Gestion de la sécurité



5.2 Les Privilèges

Intrtoduction

- Un privilège donne le droit d'exécuter certaines commandes SQL ou le droit d'accéder à certaines ressources
- Oracle possède deux types de privilèges : les *privilèges systèmes* et les *privilèges objets*.
- Un privilège peut être affecté (retiré) à un Utilisateur, un Rôle ou tous les utilisateurs (PUBLIC)

5.2 Les privilèges

Les privilèges Systèmes

- Oracle possède près de ***127 privilèges Systèmes*** (la V6 en avait 3 : *Connect, resource, dba*)
- Les privilèges donnent le droit de réaliser des opérations systèmes
- Ces privilèges sont classés par catégories d'objets

ANALYZE	AUDIT
CLUSTER	CONTEXT
DATABASE	DATABASE LINK
DIMENSION	INDEX
INDEXTYPE	LIBRARY
MATERIALIZED VIEW	MISCELLANEOUS
OPERATOR	OUTLINE
PRIVILEGE	PROCEDURE
PROFILE	Public Database Link
PUBLIC SYNONYM	ROLE
ROLLBACK SEGMENT	SESSION
SEQUENCE	SNAPSHOT
SYNONYM	SYSTEM
TABLE	TABLESPACE
TRANSACTION	TRIGGER
TYPE	USER
VIEW	

NOTE : Voir l'Annexe A6 pour obtenir la liste complète des privilèges

5.2 Les privilèges

Les privilèges systèmes (suite)

- **Exemple de privilèges systèmes de la catégorie TABLE:**

CREATE TABLE	CREATE ANY TABLE
ALTER ANY TABLE	BACKUP ANY TABLE
DROP ANY TABLE	LOCK ANY TABLE
LOCK ANY TABLE	*SELECT ANY TABLE*
INSERT ANY TABLE	UPDATE ANY TABLE
DELETE ANY TABLE	COMMENT ANY TABLE
UNDER ANY TABLE	

- **Affectation d'un privilège Système**

GRANT { system_priv | role }
TO { user | role | PUBLIC }
[WITH ADMIN OPTION]

System_priv	: nom d'un privilège système
role	: Nom d'un rôle
user, role ou PUBLIC	: droit affecté à un utilisateur, un rôle ou public
With Admin Option	: le rôle pourra être redistribué par celui qui le reçoit

5.2 Les privilèges

■ Les privilèges systèmes (suite)

- **Affectation des privilèges systèmes (suite)**

  – L'affectation d'un privilège avec l'option "WITH ADMIN OPTION" suit les règles suivantes :

  - Celui qui reçoit le droit peut le redistribuer
  - **Son retrait** à un utilisateur qui lui-même l'a affecté à un autre **ne peut se faire en cascade**
  - ne peut être affecté à un ROLE

  – **Exemple**

```
GRANT ALTER TABLESPACE TO scott ;

GRANT CREATE USER,
      CREATE SESSION TO scott
      WITH ADMIN OPTION ;

GRANT ALTER ANY TABLE TO PUBLIC ;
```


5.2 Les privilèges

■ Les Privilèges Systèmes (suite)

- **Révocation d'un privilège Système**

Syntaxe

```
REVOKE { system_priv | role }
       FROM { user | role | PUBLIC }
```

Exemple :

```
REVOKE ALTER ANY TABLE FROM  PUBLIC ;
REVOKE CREATE SESSION FROM SCOTT ;
```

- **Les vues du dictionnaire**

```
SELECT * FROM DBA_SYS_PRIVS
  ORDER BY grantee, privilege ;
```

GRANTEE	PRIVILEGE	ADM
CONNECT	ALTER SESSION	NO
CONNECT	CREATE CLUSTER	NO

NOTE : *S*upposant qu'un utilisateur U1 attribut un privilège P1 "WITH ADMIN OPTION" à un utilisateur U2 et que U2 l'attribut à son tour à U3. La révocation de P1 à U2 n'entraîne pas la révocation de P1 à U3.

5.2 Les privilèges

■ Les privilèges Objets

- Ces privilèges contrôlent l'accès aux objets des tables, vues, séquences, procédures, fonctions et packages, vue matérialisée (VM) ....
- Classification selon les types d'objets

Privilèges objets	Libellé	Objets concernés
ALTER	droit de modifier	*table, séquence*
DELETE	droit de supprimer	*table , vue, VM*
EXECUTE	droit d'exécuter	*procédure, fonction, package,type user, opérateur, indextype, library*
INDEX	droit de créer un index (*ne peut être affecté à un rôle*)	*table*
INSERT	droit d'insérer	*table, vue,VM*
ON COMMIT REFRESH	droit de créer une vue matérialisé ON COMMIT REFRESH sur une table	*table*
QUERY REWRITE	droit de créer une vue matérialisé QUERY REWRITE sur une table	*table*

5.2 Les privilèges

■ Les privilèges Objets

- Classification selon les types d'objets

Privilèges objets	Libellé	Objets concernés
REFERENCES	droit de référencer une table lors d'un alter ou create table (*ne peut être affecté à un rôle*)	*table*
READ	droit de lire dans une directory	*directory*
SELECT	droit de consulter	*table, vue, snapshot, sequence*
UPDATE	droit de mise à jour	*table ou vue*
UNDER	droit de créer des sous vue	*vue, type user*
WRITE	droit d'écrire dans une directory	*directory*

5.2 Les privilèges

■ Les privilèges Objets (suite)

- **Affectation de privilèges objets**

Syntaxe

GRANT { object_priv | ALL [PRIVILEGES] } [(column [,column] ...)]
[, { object_priv | ALL [PRIVILEGES] } [(column [,column] ...)]]
ON [schema.] object
TO { user | role | PUBLIC } [WITH GRANT OPTION]

Notes :

ALL : n'est pas un privilège mais signifie "tous les privilèges sur un objet"

object_priv: Nom du privilège

column : Nom d'une colonne si object_priv= *insert, update ou references*

schema.objet : Nom de l'objet concerné

With Grant Option: L'utilisateur qui reçoit le privilège peut le réaffecter.

Exemple

sql> GRANT INSERT (ename, job) ON emp TO scott **with grant option** ;

sql> GRANT UPDATE (SAL), DELETE ON emp TO scott ;

sql> GRANT REFERENCES, UPDATE ON bonus TO dupont ;

sq>GRANT SELECT ON emp to dupond;

5.2 Les privilèges

■ Les privilèges Objets (suite)

- **Révocation de privilèges objets**

Syntaxe

REVOKE { object_priv | ALL [PRIVILEGES] }
ON [schema.] object
FROM { user | role | PUBLIC } [CASCADE CONSTRAINTS]

Notes

CASCADE CONSTRAINTS : s'emploie avec le privilège REFERENCES, supprime les contraintes d'intégrité mises.

Retrait d'un privilège et WITH GRANT OPTION:

Si un utilisateur U1 a affecté un privilège P1 à U2 et U2 l'a affecté à U3, le retrait à U2 entraîne le retrait à U3 : le retrait se fait en cascade.

Exemples

sql>REVOKE DELETE ON Bonus FROM scott ;
sql>REVOKE UPDATE ON emp FROM public;
sql>REVOKE REFERENCES ON scott.emp FROM dupont ;
sql>REVOKE ALL ON bonus FROM PUBLIC ;

5.2 Les privilèges

■ Les privilèges Objets (suite)

- **Visualisation des privilèges objets**

DBA_TAB_PRIVS	DBA_COL_PRIVS
ALL_TAB_PRIVS	ALL_COL_PRIVS
USER_TAB_PRIVS	USER_COL_PRIVS
All_TAB_PRIVS_MADE	DBA_COL_PRIVS
USER_TAB_PRIVS_MADE	ALL_COL_PRIVS_MADE
USER_TAB_PRIVS_MADE	USER_COL_PRIVS_MADE
ALL_TAB_PRIVS_RECD	ALL_COL_PRIVS_RECD
USER_TAB_PRIVS_RECD	ALL_COL_PRIVS_RECD
TABLE_PRIVILEGES	COLUMN_PRIVILEGES

Principales Colonnes de vues ci-dessus

GRANTEE	: utilisateur ayant reçu le privilège
OWNER	: propriétaire de la table
TABLE_NAME	: nom de la table
COLUMN_NAME	: Nom de la colonne concerné
GRANTOR	: Utilisateur ayant affecté le privilège
PRIVILEGE	: privilège affecté
GRANT	: privilège reçu.

5.2 Les privilèges

■ Les privilèges Objets (suite)

- **Visualisation des privilèges objets**

Visualisation de tous les droits sur les objets de la base

SELECT * FROM sys.dba_tab_privs
WHERE table_name = 'BONUS' OR
table_name = 'EMP';

GRANTEE	OWNER	TABLE_NAME	GRANTOR	PRIVILEGE
DUPONT	SCOTT	BONUS	SCOTT	ALTER
DUPONT	SCOTT	BONUS	SCOTT	DELETE
DUPONT	SCOTT	BONUS	SCOTT	INDEX
DUPONT	SCOTT	BONUS	SCOTT	INSERT
DUPONT	SCOTT	BONUS	SCOTT	SELECT
DUPONT	SCOTT	BONUS	SCOTT	UPDATE
DUPONT	SCOTT	BONUS	SCOTT	REFERENCES

Tous les droits sur toutes les colonnes des tables dans la base

SELECT * FROM sys.dba_col_privs ;

5.3 Les rôles

Plan

- Généralités
- Création d'un rôle
- Modification d'un rôle
- Suppression d'un rôle
- Affectation de privilèges à un rôle
- Affectation d'un rôle à un utilisateur
- Rôles prédéfinis
- Informations sur les rôles

5.3 Les rôles

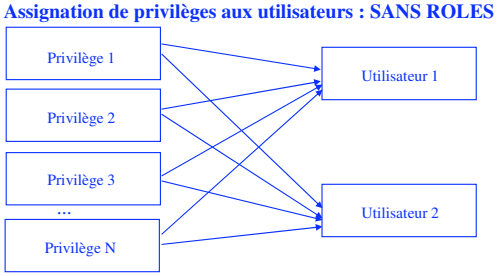
Généralités

- **Définition**
  Un rôle est un concept Oracle qui permet de regrouper plusieurs privilèges et / ou rôles afin de les affecter ou retirer en bloc à un utilisateur et / ou un rôle.
- un rôle facilite la gestion des privilèges
- l'**affectation** d'un rôle à un utilisateur peut se faire sous Oracle ou **à travers l'OS**
- pour des raisons de sécurité, un **mot de passe** peut être assigné à un rôle
- Oracle fournit un certain nombre de **rôles par défaut** (connect, resource, dba, exp_full_database, imp_full_data_base, select_catalog_role, delete_catalog_role / execute_catalog_role, …)
- **pour créer un rôle**, il faut **avoir le privilège** "CREATE ROLE"

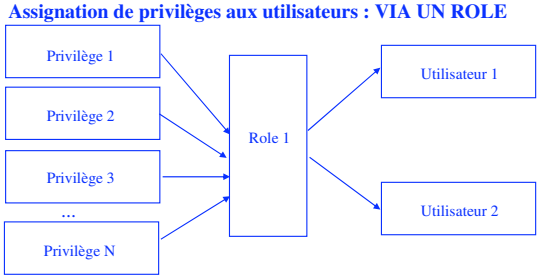
5.3 Les rôles

Généralités (suite)

Assignation de privilèges aux utilisateurs : SANS ROLES



Assignation de privilèges aux utilisateurs : VIA UN ROLE



5.3 Les rôles

Création d'un rôle

- A sa création, un rôle ne contient aucun privilège

Syntaxe

```
CREATE   ROLE   role
      [ { NOT  IDENTIFIED
      | IDENTIFIED { BY  password  | EXTERNALLY | GLOBALLY |
         USING package} ]
```

Mots clés et paramètres

role	: nom du rôle à créer
NOT IDENTIFIED	: permet de créer un rôle sans mot de passe
Password	: mot de passe assigné au rôle
EXTERNALLY	: mot de passe est contrôlé au niveau de l'OS
GLOBALLY	: Rôle autorisé au niveau de l'annuaire
USING package	: rôle applicatif

Exemple

```
sql> CREATE ROLE  rl_etudiant ;
sql> CREATE ROLE  rl_admin_backup;
sql> CREATE ROLE  rl_admin_secu IDENTIFIED BY secu_pass ;
```

Modification d'un rôle

- On peut modifier le *niveau de sécurité* d'un rôle
- privilège requis pour modifier un rôle ALTER ANY ROLE.

Syntaxe

```
ALTER  ROLE   role
                    { NOT  IDENTIFIED
                    | IDENTIFIED  { BY password |   EXTERNALLY |
                                    Globally | USING package} }
```

Mots clés et paramètres

role	: nom du rôle à créer
NOT IDENTIFIED	: permet d'inhiber le mot de passe d'un rôle
Password	: nouveau mot de passe assigné au rôle
EXTERNALLY	: mot de passe contrôlé au niveau de l'OS
GLOBALLY	: Rôle autorisé au niveau de l'annuaire
USING package	: rôle applicatif

Exemple

```
sql> ALTER ROLE ROLE  rl_etudiant  IDENTIFIED  EXTERNALLY ;
sql> ALTER ROLE  rl_admin_backup IDENTIFIED BY backup_pass;
sql> ALTER ROLE  rl_admin_secu NOT IDENTIFIED;
```


Suppression d'un rôle

- Un rôle supprimé est retiré IMMEDIATEMENT du domaine de sécurité de l'utilisateur (connecté ou non) ou du rôle l'ayant reçu
- Le privilège DROP ANY ROLE et le fait d'avoir acquis un rôle avec WITH ADMIN OPTION permettent de le supprimer
- **Syntaxe**

  DROP ROLE role ;

- **Exemple**

  DROP ROLE rl_admin_secu ;

Affectation de privilèges ou de rôles à un rôle

- **Exemple**

Création de deux rôles

```
# rôle rassemblant les privilèges pour se connecter
CREATE ROLE rl_connect ;

# rôle rassemblant les privilèges pour administrer la sécurité
CREATE ROLE rl_admin_secu ;
```

Affectation des privilèges aux rôles

```
GRANT create session, alter session,
                Restricted session TO rl_connect ;
GRANT create role, create user, create profile
        TO rl_admin_secu;
```

Affectation d'un Rôle à un autre Rôle

```
GRANT rl_connect TO rl_admin_secu ;
```


Affectation de privilèges à un rôle

- **Privilèges ne pouvant être affectés à un ROLE**
  - **Privilège Système**

    UNLIMITED TABLESPACE

    Ce privilège inhibe tous les quotas et autorise l'utilisateur à créer des objets dans n'importe quel tablespace.

  - **Privilèges Objets**

    INDEX # droit de créer un index sur les tables d'autres utilisateurs

    REFERENCES # droit de référencer une table dans le schéma d'autres utilisateurs

5.3 Les rôles

Affectation d'un rôle à un Utilisateur

- **Elle peut se faire au niveau** :
  - Oracle
  - du Système d'Exploitation (OS)
  - De l'annuaire de l'entreprise

- **Affectation d'un Rôle au niveau Oracle**

  GRANT role to user [WITH ADMIN OPTION]

  L'utilisateur ayant reçu le rôle avec WITH ADMIN OPTION peut le réaffecter, supprimer ou modifier.

5.3 Les rôles

Affectation d'un rôle à un utilisateur (suite)

- **Affectation d'un rôle au niveau de l'OS**

  - Positionner le paramètre OS_ROLE dans init.ora afin que l'affectation et la révocation des rôles se fassent au niveau de l'OS

    OS_ROLE = TRUE

  - Déclarer (sous UNIX) dans le fichier de groupe chaque rôle comme étant un groupe

    Syntaxe
    ora_<SID>_<role>[_[D][A] : [user1, [user2], [...]]

    Avec

SID	: nom de l'instance
role	: nom du rôle
D	: rôle par défaut
A	: WITH ADMIN OPTION

    Exemple :
    ora_COURS_rl_connect_D:scott, mopolo, tintin
  - ora_COURS_rl_admin_secu_DA:mopolo,osmani

5.3 Les rôles

Affectation d'un rôle à un utilisateur (suite)

- **NOTES sur l'affectation d'un rôle à partir de l'OS**

  - L'affectation et la **révocation de rôles ne se fait plus qu'au niveau de l'OS**. Impossible d'utiliser l'ordre GRANT role TO user

  - un rôle affecté via l'OS **peut être activé ou désactivé par l'utilisateur** avec la commande ALTER USER ... SET ROLE ...;

  - les rôles **non gérés au niveau de l'OS ne peuvent être activés ou désactivés** même s'ils avaient étés affectés lorsque OS_ROLES était égal à FALSE

  - le paramètre MAX_ENABLED_ROLES limite le nombre de rôles pouvant être activés

  - Si les rôles sont gérés par l'OS et on est en architecture Multithread, les connexions distantes exploitant ses rôles ne seront possibles que si le paramètre de init.ora REMOTE_OS_ROLE=TRUE

5.3 Les rôles

Rôles prédéfinis

NOM DU ROLE	PRIVILEGES AFFECTES AU ROLE
CONNECT	ALTER SESSION, CREATE CLUSTER, CREATE DATABASE LINK, CREATE SEQUENCE, CREATE SESSION, CREATE SYNONYM, CREATE TABLE, CREATE VIEW
RESOURCE	CREATE CLUSTER, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER
DBA	tous les privilèges (annexe A6) WITH ADMIN OPTION
exp_full_database	SELECT ANY TABLE, BACKUP ANY TABLE, INSERT, DELETE AND UPDATE ON tables SYS.INCVID, SYS.INCFIL, SYS.INCEXP
imp_full_database	BECOME USER, WRITEDOWN (trusted Oracle)
Execute_catalog_role	Privilège d'exécuter les procédures du dictionnaire
Select_catalog_role	Privilège de consulter tout le dictionnaire Oracle
Delete_catalog_role	Privilège de supprimer la table d'audit aud$
Recovery_catalog_owner	Fournit les privilèges pour le propriétaire du catalogue de recouvrement
Aq_administrator_role	Fournit les privilèges pour l'administration des Queue
Snmpagent	Founit les privilèges utiles à l'agent SNMP d'entreprise manager

5.3 Les rôles

Rôles prédéfinis

NOTES :

. Les rôles CONNECT, RESOURCE, DBA correspondent aux anciens privilèges systèmes sous Oracle V6

. **l'affectation des rôles RESOURCE et DBA** à un utilisateur entraîne un GRANT ***unlimited tablespace***

5.3 Les rôles

Informations sur les rôles

- Les vues suivantes contiennent des informations sur les rôles :
  dba_roles,
  user_role_privs, dba_role_privs
  role_role_privs, role_sys_privs, role_tab_privs
  session_roles
- Exemple 1 : listing de tous les rôles de la base

  sql> SELECT * FROM sys.dba_roles ;

ROLE	PASSWORD
CONNECT	NO
RESOURCE	NO
DBA	NO
EXP_FULL_DATABASE	NO
IMP_FULL_DATABASE	NO
MONITORER	NO
RL_ADMIN_SECU	NO
RL_CONNECT	NO

5.3 Les rôles

Informations sur les rôles (suite)

- Exemple 2 : liste des rôles affectés à un role ou un utilisateur.

  sql>SELECT * FROM sys.dba_role_privs
  WHERE grantee = 'RL_ADMIN_SECU' ;

GRANTEE	GRANTED_ROLE	ADM DEF	Default
RL_ADMIN_SECU	RL_CONNECT	NO	YES

- Exemple 3 : liste des rôles actifs pour la session

  sql>SELECT * FROM session_roles ;

ROLE
MONITORER
DBA
EXP_FULL_DATABASE
IMP_FULL_DATABASE
RL_ADMIN_SECU

5.4 Les profiles

Généralités

- Un **profile** est un concept Oracle qui **permet** à l'administrateur d'une base **de contrôler la consommation des ressources systèmes et des mots de passes**
- Il existe un profile par défaut appelé **DEFAULT**. Il est par défaut affecté à un utilisateur lors de sa création
- Les limites du profile DEFAULT sont positionnées à UNLIMITED
- Le profile DEFAULT ne peut être supprimé. *Les limites de ce profile peuvent par contre être modifiées*
- **activation et contrôle des limites** :
  - dans le fichier init*SID*.ora positionner :
    RESOURCE_LIMIT = TRUE
  - ou dynamiquement faire sous sqlplus par exemple :
    SQL> ALTER SYSTEM SET resource_limit = true;

5.4 Les profiles

Création d'un profile

- Privilège requis CREATE PROFILE

Syntaxe partie limite des ressources

```
CREATE PROFILE profile LIMIT
  [ SESSIONS_PER_USER { integer | UNLIMITED | DEFAULT} ]
  [ CPU_PER_SESSION { integer | UNLIMITED | DEFAULT } ]
  [ CPU_PER_CALL { integer | UNLIMITED | DEFAULT } ]
  [ CONNECT_TIME { integer | UNLIMITED | DEFAULT } ]
  [ IDLE_TIME { integer | UNLIMITED | DEFAULT } ]
  [LOGICAL_READS_PER_SESSION {integer | UNLIMITED|DEFAULT}]
  [LOGICAL_READS_PER_CALL {integer | UNLIMITED|DEFAULT}]
  [ COMPOSITE_LIMIT { integer | UNLIMITED | DEFAULT } ]
  [PRIVATE_SGA {integer [K | M] | UNLIMITED | DEFAULT}]
```

Mots clés et paramètres

Session_per_user : Nombre maximum de sessions par utilisateur
Logical_read_per_session : Nbre de blocs de données à lire pour une session
cpu_per_session : temps CPU max par session en % de sécondes
cpu_per_call : temps CPU pour un appel (en cas de parse, execute ou fetch) en % de secondes
connect_time : temps écoulé maximum (en minutes)
idle_time : temps maximum d'inactivité.
private_sga : taille privée de la SGA allouée à un utilisateur
unlimited : limite de la ressource illimitée
default : prend la limite par défaut de la ressource

5.4 Les profiles

Création d'un profile

- Privilège requis CREATE PROFILE

Syntaxe partie password

```
CREATE PROFILE profile LIMIT
  [FAILED_LOGIN_ATTEMPTS {expr | UNLIMITED | DEFAULT}]
  [PASSWORD_LIFE_TIME {expr | UNLIMITED | DEFAULT}]
  [PASSWORD_REUSE_TIME {expr | UNLIMITED | DEFAULT}]
  [PASSWORD_REUSE_MAX {expr | UNLIMITED | DEFAULT}]
  [PASSWORD_LOCK_TIME {expr | UNLIMITED | DEFAULT}]
  [PASSWORD_GRACE_TIME {expr | UNLIMITED | DEFAULT}]
  [PASSWORD_VERIFY_FUNCTION {function, NULL, DEFAULT}]
```

Mots clés et paramètres

Failed_login_attempts : nombre d'échecs avant le blocage du compte
password_life_time : durée en jours avant l'expiration du mot de passe
password_reuse_time : durée en jours avant la réutilisation d'un password
password_reuse_max : nombre de modif du password avant réutilisation
password_lock_time : durée en jours du verrouillage d'un compte
password_grace_time : délai de tolérance du password avant son expiration
password_verify_function : fonction de contrôle des mots de passes

5.4 Les profiles

Création d'un profile (suite)

- Exemple (suite)

Exemple 1

```
CREATE PROFILE pf_secretaire LIMIT
  sessions_per_user          2
  cpu_per_session            unlimited
  cpu_per_call               1000
  logical_reads_per_session  unlimited
  logical_reads_per_call     100
  idle_time                  30
  connect_time               480 ;
```

Exemple 2

```
CREATE profile pf_agent LIMIT
  sessions_per_user          2
  cpu_per_session            unlimited
  cpu_per_call               1000
  composite_limit            20000
  private_sga                32K ;
```

Exemple 3

```
CREATE PROFILE pf_admin
PASSWORD_LIFE_TIME 200
LIMIT PASSWORD_REUSE_MAX DEFAULT
PASSWORD_REUSE_TIME UNLIMITED
CPU_PER_SESSION UNLIMITED
```


5.4 Les profiles

Assignation d'un profile à un utilisateur

- A la création d'un nouvel utilisateur

```
CREATE USER rackham IDENTIFIED BY lerouge
PROFILE pf_secretaire ;
```

- A la modification d'un utilisateur

```
ALTER USER rackham
  PROFILE pf_agent ;
```

5.4 Les profiles

Modification d'un profile

- Privilège requis : ALTER PROFILE

Syntaxe partie limite des ressources

```
ALTER PROFILE profile LIMIT
  [ SESSIONS_PER_USER { integer | UNLIMITED | DEFAULT} ]
  [ CPU_PER_SESSION { integer | UNLIMITED | DEFAULT } ]
  [ CPU_PER_CALL { integer | UNLIMITED | DEFAULT } ]
  [ CONNECT_TIME { integer | UNLIMITED | DEFAULT } ]
  [ IDLE_TIME { integer | UNLIMITED | DEFAULT } ]
  [LOGICAL_READS_PER_SESSION {integer | UNLIMITED|DEFAULT}]
  [LOGICAL_READS_PER_CALL {integer | UNLIMITED|DEFAULT}]
  [ COMPOSITE_LIMIT { integer | UNLIMITED | DEFAULT } ]
  [PRIVATE_SGA {integer [K | M] | UNLIMITED | DEFAULT}]
```

Mots clés et paramètres

Session_per_user : Nombre maximum de sessions par utilisateurs
Logical_read_per_session: blocs de données en lecture par session
cpu_per_session : temps CPU max par session en % de sécondes
cpu_per_call : temps CPU pour un appel (en acs de parse, execute ou fetch) en % de secondes
connect_time : temps écoulé maximum (en minutes)
idle_time : temps maximum d'inactivité.
private_sga : taille privée de SGA allouée à un utilisateur
unlimited : limite de la ressource illimitée
default : positionne la limite par défaut de la ressource

5.4 Les profiles

Modification d'un profile

- Privilège requis : ALTER PROFILE

Syntaxe partie password

```
ALTER PROFILE profile LIMIT
  [FAILED_LOGIN_ATTEMPTS {expr | UNLIMITED | DEFAULT}]
  [PASSWORD_LIFE_TIME {expr | UNLIMITED | DEFAULT}]
  [PASSWORD_REUSE_TIME {expr | UNLIMITED | DEFAULT}]
  [PASSWORD_REUSE_MAX {expr | UNLIMITED | DEFAULT}]
  [PASSWORD_LOCK_TIME {expr | UNLIMITED | DEFAULT}]
  [PASSWORD_GRACE_TIME {expr | UNLIMITED | DEFAULT}]
  [PASSWORD_VERIFY_FUNCTION {function, NULL,
DEFAULT}]
```

Mots clés et paramètres

Failed_login_attempts
: nombre d'échecs avant le blocage du compte
password_life_time
: durée en jours avant l'expiration du mot de passe
password_reuse_time
: durée en jours avant la réutilisation d'un password
password_reuse_max
: nombre de modif du password avant réutilisation
password_lock_time
: durée en jours du verrouillage d'un compte
password_grace_time
: délai de tolérance du password avant son expiration
password_verify_function
: fonction de contrôle des mots de passes

5.4 Les Profiles

Modification d'un profile

Exemple 1 :

Modification des limites du profile par défaut DEFAULT

```
ALTER PROFILE default LIMIT
        CPU_PER_SESSION 600
```

Exemple 2 :

Modification des limites du profile *pf_agent*

```
ALTER PROFILE pf_agent LIMIT
        CPU_PER_SESSION default
```

Que vaut CPU_PER_SESSION pour le profile pf_agent ?

5.4 Les profiles

Utilisations des limites composites

- Généralités
  - Fixe le **coût total des limites** pour une session
  - **à chaque limite est associé un poids** :
    - par défaut les poids sont à 0
    - un fort poids implique un coût élevé de la limite
    - un poids ne peut être qu'associer aux limites suivantes (cpu_per_session, connect_time, logical_reads_per_session, private_sga)
  - **privilège requis** : ALTER RESOURCE COST

5.4 Les profiles

Création de limites composites

Syntaxe

```
ALTER  RESOURCE  COST
            [ CPU_PER_SESSION    integer ]
            [  CONNECT_TIME        integer  ]
            [  LOGICAL_READS_PER_SESSION integer]
            [ PRIVATE_SGA integer ]
```

Mots clés et paramètres

integer : poids pour chaque ressource
Logical_read_per_session : blocs de données en lecture par session
cpu_per_session : temps CPU max par session en % de sécondes
connect_time : temps écoulé maximum (en minutes)
private_sga : taille de la SGA privé à ne pas dépasser

La limite PRIVATE_SGA est - t - elle toujours valide quel que soit l'architecture d'Oracle ?

5.4 Les profiles

Création d'une limite composite (suite)

Formule d'évaluation du coût total des ressources pour une session.

T = ∑ poids * consommation_per_resource_limit

Exemple 1 :

```
SQL> ALTER RESOURCE COST
            cpu_per_session        100
            connect_time           1;
```

T = 100*cpu_per_session_consommé + 1*connect_time_consommé +
0*logical_reads_per_session + 0*private_sga

Exemple 2 :

```
SQL> ALTER RESOURCE COST
            logical_reads_per_session        2
            connect_time                     0;
```

T = 100*cpu_per_session_consommé + 0*connect_time_consommé +
2*logical_reads_per_session + 0*private_sga

Note :
Le résultat **T** est à comparer avec la valeur de **COMPOSITE_LIMIT**

5.4 Les profiles

Suppression d'un profile

- En cas de suppression d'un profile existant affecté à un utilisateur, ce dernier se verra automatiquement attribué le profile DEFAULT
- Le profile DEFAULT ne peut être supprimé
- Privilège requis : DROP PROFILE

Syntaxe

```
DROP PROFILE nom_profile  [CASCADE]
```

Mots clés et paramètres

nom_profile : nom du profile à supprimer
CASCADE : retire le profile aux utilisateurs l'ayant puis, suppression du profile

Exemple

```
sql>DROP PROFILE pf_secretaire CASCADE ;
```


5.4 Les profiles

Visualisation des informations des profiles

- Vues contenant les informations sur les profiles :
  - dba_profiles, resource_cost, user_resource_limit

Exemple 1 : Liste de tous les profiles

```
sql> SELECT profile, resource_name, limit FROM dba_profiles
ORDER BY profile, resource_name;
```

PROFILE	RESOURCE_NAME	LIMIT
DEFAULT	COMPOSITE_LIMIT	UNLIMITED
...		
DEFAULT	CPU_PER_SESSION	600
PF_AGENT	COMPOSITE_LIMIT	20000
PF_AGENT	CONNECT_TIME	DEFAULT
PF_AGENT	CPU_PER_CALL	1000
PF_AGENT	PRIVATE_SGA	32768
PF_AGENT	SESSIONS_PER_USER	2
...		
PF_SECRETAIRE	COMPOSITE_LIMIT	DEFAULT
PF_SECRETAIRE	CONNECT_TIME	480
PF_SECRETAIRE	CPU_PER_CALL	1000
PF_SECRETAIRE	CPU_PER_SESSION	UNLIMITED
PF_SECRETAIRE	IDLE_TIME	30
PF_SECRETAIRE	LOGICAL_READS_PER_CALL	100
PF_SECRETAIRE	SESSIONS_PER_USER	2
...		

5.4 Les profiles

Visualisation des informations des profiles (suite)

Exemple 2 : Liste des coûts (poids) des ressources pour la session courante

```
sql> SELECT resource_name, limit FROM resource_cost
```

RESOURCE_NAME	UNIT_COST
CPU_PER_SESSION	100
LOGICAL_READS_PER_SESSION	2
CONNECT_TIME	0
PRIVATE_SGA	0

Exemple 3 : Liste des limites des ressources de l'utilisateur courant

```
sql> SELECT resource_name, limit FROM user_resource_limits
```

RESOURCE_NAME	LIMIT
COMPOSITE_LIMIT	UNLIMITED
SESSIONS_PER_USER	UNLIMITED
CPU_PER_SESSION	600
CPU_PER_CALL	UNLIMITED
LOGICAL_READS_PER_SESSION	UNLIMITED
LOGICAL_READS_PER_CALL	UNLIMITED
IDLE_TIME	UNLIMITED
CONNECT_TIME	UNLIMITED
PRIVATE_SGA	UNLIMITED

5.5 Les utilisateurs

Généralités

- La notion d'utilisateur est fondamentale pour accéder aux données d'une base Oracle
- Le site d'un client Oracle doit être tenu à jour au niveau des licences :
  - les paramètres de *init.ora* pour le contrôle de la licence (valeurs par défaut 0) : *license_max_session, license_sessions_warning, license_max_users*
    - Si la limite en nombre de sessions est illimitée conserver les valeurs par défaut
  - Seuls les utilisateurs avec le privilège RESTRICTED SESSION peuvent se connecter en cas de dépassement de la limite
  - en cas de mise en oeuvre de l'architecture parallèle, chaque instance à ses limites mais la somme doit équivaloir à la somme des limites du site
  - visualisation des limites des licences ***v$license***

5.5 Les utilisateurs

Généralités (suite)

- **Contrôle de la limitation du nombre d'utilisateurs**
  - Au moment du lancement d'une instance
    LICENSE_MAX_USERS = 80
  - Au moment ou l'instance tourne
    sqlplus >ALTER SYSTEM
    SET LICENSE_MAX_USERS=100;
- **Authentification des utilisateurs**
  - A partir d'Oracle
    CREATE USER scott IDENTIFIED BY tiger ;
  - A partir de l'OS
    CREATE USER OPS$mopolo IDENTIFIED EXTERNALLY ;
    - Les utilisateurs authentifiés par l'OS sont précédés d'une chaîne définie par le paramètre de init*sid*.ora OS_AUTHENT_PREFIX qui vaut par défaut **OPS$**

5.5 Les utilisateurs

Généralités (suite)

- **Authentification des utilisateurs**
  - Globalement à partir de l'annuaire LDAP
    CREATE USER scott
    IDENTIFIED GLOBALLY AS
    'CN=scott,OU=division1,O=oracle,C=US'

Généralités (suite)

- **Utilisateur et schéma**
  - A **chaque utilisateur est associé un schéma**
  - Les objets appartenant à un schéma sont : tables, index, vues, séquences, synonymes, clusters, database links, fonction, procédures et package, ...
  - La **commande** *CREATE SCHEMA AUTHORIZATION* permet créer en un trait des *tables*, des *vues* et d'attribuer des *droits*. En cas d'erreur, un ROLLBACK peut être effectué. **Exemple :**

```
CREATE SCHEMA AUTHORIZATION ottocar

CREATE TABLE VOL (
        vol#        number(4)   primary key,
        plnum       number(4)   references pilote,
        vd          char(12),
        va          char(12))

CREATE TABLE PILOTE (
        pl#         number(4)   primary key,
        plnom       varchar2(20),
        sal         number (5, 2)          not null)

GRANT select, update(plnom, sal) ON pilote TO tintin;
```


Création d'un utilisateur

- Lors de la création d'un utilisateur, il est possible de lui affecter : un mot de passe, un tablespace par défaut, un tablespace temporaire, un profile (explicite ou implicite), des quotas sur les tablespaces.

Syntaxe

```
CREATE  USER user
   IDENTIFIED  { BY  password | EXTERNALLY| GLOBALLY
AS 'nom_externe' }
   [ DEFAULT TABLESPACE  tablespace ]
   [ TEMPORARY  TABLESPACE  tablespace ]
   [ QU0TA { integer  [ K | M ] | UNLIMITED } ON tablespace ] ...
   [ PROFILE   profile ]
   [PASSWORD EXPIRE]
   [ACCOUNT {LOCK | UNLOCK}]
```

Mots clés et paramètres

User	: nom de l'utilisateur à créer
password	: mot de passe
Externally	: utilisateur authentifié par l'OS
tablespace	: nom du tablespace
profile	: nom du profile
globally as	: accès autorisé l'annuaire LDAP

Création d'un utilisateur (suite)

Exemple 1 : création d'un utilisateur nommé TINTIN identifié au niveau de l'OS dont le tablespace par défaut est USERS. Cet utilisateur à un quota de 2Mo sur les tablespaces SYSTEM et USERS.

```
sql> CREATE USER OPS$tintin IDENTIFIED EXTERNALLY
       DEFAULT TABLESPACE users
       QUOTA 2 M ON system
       QUOTA 2 M ON users;
```

Exemple 2 : Création d'un utilisateur nommé DUPOND ayant DUPONT comme mot de passe.

```
sql> CREATE USER dupond IDENTIFIED BY dupont ;
```

Notes

. le tablespace temporaire par défaut est SYSTEM
. le tablespace par défaut est SYSTEM
. il est *obligatoire d'affecter des quotas sur les tablespaces* ou d'affecter le privilège UNLIMITED TABLESPACE
. les rôles affectés lors de la création d'un utilisateur sont par défaut
. le privilège CREATE USER est requis.

Modification d'un utilisateur

Syntaxe

```
ALTER  USER  user
      [ IDENTIFIED  { BY password |  EXTERNALLY | Globally} ]
      [ DEFAULT  TABLESPACE   tablespace ]
      [ TEMPORARY  TABLESPACE  tablespace ]
      [ QUOTA { integer [ K | M ] UNLIMITED } ON tablespace ] ...
      [ PROFILE  profile ]
      [DEFAULT  ROLE { role [, role ] ...
        | ALL [ EXCEPT  role [, role ] ... ] | NONE } ]
      [PASSWORD EXPIRE]
      [ACCOUNT {LOCK | UNLOCK}]
      [Proxy_clause]
```

Mots clés et paramètres

password	:	Nouveau mot de passe
tablespace	:	Nom du tablespace par défaut et/ou du tablespace temporaire
profile	:	Nom du nouveau profile de l'utilisateur
role	:	Nom du ou des nouveaux rôles par défaut ou à exclure
ALL	:	Tous les rôles deviennent par défaut
NONE	:	Aucun rôle par défaut
EXECPT	:	Les rôles à exclure apparaissent après ce mot clé
Proxy_clause	:	Authentification des utilisateurs via un proxy

NOTES :

Rôle par défaut = Rôle affecté directement à un utilisateur

Modification d'un utilisateur

- **Exemple 1**

  Modification de l'utilisateur DUPONT: nouveau mot de passe BOULES, **tablespace** par défaut USER quota sur ce tablespace illimité et 0 sur le tablespace SYSTEM

```
sql> ALTER USER dupont IDENTIFIED BY boules
        DEFAULT TABLESPACE user
        QUOTA UNLIMITED ON user
        QUOTA 0 ON system ;
```

- **Exemple 2**
  Modification de l'utilisateur TINTIN : assignation d'un nouveau tablespace par temporaire TEMP et assignation de tous rôles par défaut sauf rl_admin_secu.

```
sql> ALTER USER tintin
  TEMPORARY TABLESPACE temp
  DEFAULT ROLE ALL EXECPT rl_admin_secu;
```


Suppression d'un utilisateur

- La suppression d'un utilisateur entraîne la suppression des objets de son schéma (tables, vues, séquences, synonymes, indexes, clusters indexés, clusters hashés, ...)
- Privilège requis : DROP USER

Syntaxe

```
DROP USER user [ CASCADE ]
```

Mots clés et paramètres

user	: Nom de l'utilisateur à supprimer
CASCADE	: supprime aussi les objets du schéma de l'utilisateur et les contraintes d'intégrité de référence.

Exemple

```
DROP USER tsang ;
DROP USER dupont CASCADE ;
```


Affectation de droits à un Utilisateur

- **Exemple 1** : Affectation de droits systèmes

```
sql> GRANT create tablespace, create user TO tintin ;
```

- **Exemple 2** : Affectation d'un rôle à un utilisateur

```
sql> GRANT rl_admin_secu TO tintin ;
```

- **Exemple 3** : Affectation d'un privilège objet à un utilisateur

```
sql> GRANT SELECT, UPDATE (ename, sal)
        ON EMP TO tintin ;
```

- **Exemple 4** : Affectation de privilèges à tous les utilisateurs

```
sql> GRANT drop any table TO PUBLIC ;
```

NOTE !!! Attention danger

Informations sur les utilisateurs

- **Quelques vues sur les utilisateurs**
  user_users, all_users, dba_users
  user_ts_quotas dba_ts_quotas

Exemple 1: informations concernant l'utilisateur actuel
SELECT username, user_id, default_tablespace, temporray_tablespace, created FROM user_users;

username	user_id	default_tablespace	temporary_tablespace	created
OTTOCAR	23	USER_DATA	TEMPORARY_DATA	07/05/96

Exemple 2 : informations sur tous les utilisateurs
SELECT * FROM all_users;

USERNAME	USER_ID	CREATED
SYSTEM	5	26/09/95
SCOTT	8	26/09/95
DUPONT	13	29/04/96
OPS$MOPOLO	22	05/05/96
OTTOCAR	23	07/05/96

5.5 Les utilisateurs

■ Informations sur les utilisateurs (suite)

Exemple 3 : toutes les informations sur tous les utilisateurs
SELECT username, user_id, password, default_tablespace, temporray_tablespace, created FROM user_users;

USERNAME	USER_ID	PASSWORD	DEFAULT_ TABLESPACE	TEMPORARY_ TABLESPACE	CREATED	PROFILE
SYSTEM	5	D4DF7931AB130E37	USER_DATA	TEMPORARY_DATA	26/09/95	DEFAULT
SCOTT	8	F8948444 402B67	USER_DATA	TEMPORARY_DATA	26/09/95	DEFAULT
DUPONT	13	38BD25849F 94015A	SYSTEM	SYSTEM	29/04/96	DEFAULT
OPS$MOPOLO	22	EXTERNAL	SYSTEM	SYSTEM	05/05/96	DEFAULT
OTTOCA	23	5006630829F6EA0E	USER_DATA	TEMPORARY_DATA	07/05/96	DEFAULT

Exemple 4 : Informations sur les quotas de l'utilisateur actuel
SELECT * FROM user_ts_quotas;

TABLESPACE_NAME	BYTES	MAX_BYTES	BLOCKS	MAX_BLOCKS
USER_DATA	0	552960	0	270
TTEST	0	20480	0	10

5.5 Les utilisateurs

■ Informations sur les utilisateurs (suite)

Exemple 5 :

Informations sur les quotas de tous les utilisateurs

slq> SELECT * FROM dba_ts_quotas;

TABLESPACE_ NAME	USERNAME	BYTES	MAX_ BYTES	BLOCKS	MAX_ BLOCKS
TEMPORARY_DATA	OTTOCAR	0	-1	0	-1
USER_DATA	OTTOCAR	40960	-1	20	-1

Max_bytes : quota disque en octets autorisés pour un utilisateur
bytes : nombre d'octets déjà consommés
blocks : nombre de blocs déjà consommés
Max_blocks : quota disque en blocs pour un utilisateur

5.5 Les utilisateurs

■ Suppression d'une session

– En cas de problèmes avec une session (interblocage, consommation excessive de resources, ...), sa suppression peut être décidée

– cette suppression entraîne :
  • l'annulation de la transaction concernée,
  • la libération des verrrous et des resources consommées

– la commande *Alter System Kill Session ...* permet de supprimer une session.

select sid, serial#, username FRom v$session ;

sid	Serial#	Username
13	8	tintin
14	11	sadm

ALTER SYSTEM KILL SESSION '13, 8';

Note : Si l'on tente de supprimer une session active, Oracle note la demande. La suppression aura lieu dès que la session cesse d'être active.

5.6 L'audit

■ L'objectif de l'audit est de contrôler les accès mal intentionnés ou non autorisés à la base

■ Règles pour un bon audit

- Bien choisir le lieu de mise en oeuvre de l'audit
  – Audit au niveau Oracle(**audit_trail**=TRUE). Avantage :
    • audit ciblé grâce à SQL
    • possibilité d'exploiter les outils Oracle pour générer des rapports
  – Audit au niveau de l'OS(**audit_trail** = OS). Avantage :
    • possibilité de consolider des audits de sources diverses
  – Inhiber l'audit (**audit_trail** = NONE)
- Bien cibler l'audit pour en limiter le volume
- Avoir une démarche d'audit par raffinement successif
- protéger la table d'audit (sys.aud$) en limitant l'affectation du privilège DROP ANY TABLE
- créer les vues d'audit en exécutant le script cataudit.sql (catnoaudit.sql permet de les supprimer)

5.6 L'audit

Types d'audits

- **par utilisateur** : auditer l'activité d'un ou plusieurs utilisateurs
- **par session** (en cas de succès ou d'insuccès) : Pour N activations d'une action dans une session, conserver une seule trace
- **par accès** (en cas de succès ou d'insuccès) : Pour N activations d'une action dans une session, conserver N trace

Types de surveillance

- Audit sur les ordres SQL et les privilèges systèmes
- Audit sur les objets.

5.6 L'audit

Audit des ordres sql et des privilèges systèmes

- Privilège requis : AUDIT SYSTEM

Syntaxe

```
AUDIT  { {statement_opt | ALL}  |
                {system_priv | ALL PRIVILEGES} }
          [{ {statement_opt | ALL}  |
                {system_priv | ALL PRIVILEGES} }]
            [ BY  user [, user ] ...
            [BY PROXY ON BEHALF OFF {ANY | user…}]
            [ BY { SESSION  | ACCESS } ]
            [ WHENEVER  [ NOT ]  SUCCESSFUL ]
```

Mots clés et paramètres

statement_opt	: option des ordres sql à auditer
system_priv	: privilège système à auditer
user	: audit des ordres sql et/ou privilèges pour un user
BY SESSION	: audit des ordres sql et/ou privilèges par session
BY ACCESS	: audit des ordres sql et/ou privilèges par accès
SUCCESSFUL	: audit des ordres sql et/ou privilèges si succès
ALL	: audit de tous les ordres SQL
ALL PRIVILEGES	: audit de tous les privilèges

5.6 L'audit

Audit des ordres SQL et des privilèges systèmes (suite)

Exemples 1

auditer les connexions et déconnexions des utilisateurs *rackham* et *cyclone* (en cas de succès ou d'insuccès).

```
sql> AUDIT SESSION BY rackham, cyclone ;
```

Exemples 2

auditer les ordres *select*, *insert*, *delete* pour toutes les tables et le privilège *execute any procedure* par accès et ceci en cas d'échec.

```
sql> AUDIT select table, insert table, delete table,
          execute any procedure
       BY ACCESS WHENEVER NOT SUCCESSFUL ;
```

ou

```
sql> AUDIT table, execute any procedure
       BY ACCESS WHENEVER NOT SUCCESSFUL ;
```


5.6 L'audit

Options d'audit des ordres SQL

Options	Ordres SQL audités
alter system	alter system
cluster	create cluster, alter cluster, drop cluster, truncate cluster
database link	create database link, drop database link
index	create index, alter index, drop index
not exists	tout ordre sql qui retourne l'erreur : objet inexistant (create table, ...)
procedure	create[or replace] function, create[or replace] package, create[or replace] package body, create[or replace] procedure, drop package, drop procedure
public database link	create public database link, drop public database link
public synonym	create public synonym drop public synonym
role	create role, alter role, set role, drop role

Options d'audit des ordres SQL(suite)

Options	Ordres SQL audités
rollback segment	create rollback segment, alter rollback segment, drop rollback segment
sequence	create sequence, drop sequence
synonym	create synonym, drop synonym
system audit	audit, no audit
system grant	GRANT system_privileges/roles TO user/role REVOKE system_privileges/roles FROM user/role
table	create table, alter table, drop table
tablespace	create tablespace, alter tablespace, drop tablespace
trigger	create trigger, alter trigger enable or disable alter table with enable, disable and drop clauses
user	create user, alter user, drop user
view	create[or replace] view, drop view

Options d'audit des ordres SQL(suite)

Options	Ordres SQL audités
alter sequence	alter sequence nom_sequence
alter table	alter table nom_table
comment table	comment on table, vue, snapshot, colonne
delete table	delete from nom_table, bnom_vue
execute procedure	appel de procedure et de fonction
grant procedure	**grant** privilège **on** procedure, **revoke** privilège **on** procedure
grant table	**grant** privilège **on** table, vue, snapshot **revoke** privilège **on** table, vue, snapshot
insert table	insert into table, vue
lock table	lock table table, vue
select sequence	référence à une séquence
select table	select * from table, vue, snapshot
update table	update, vue
Context	create context, drop context
DIMENSION	CREATE DIMENSION, ALTER DIMENSION DROP DIMENSION
DIRECTORY	CREATE DIRECTORY, DROP DIRECTORY

Audit des objets du schéma

- Privilège Requis : AUDIT ANY

Syntaxe

```
AUDIT  object_opt  [, object_opt] ...
          ON {[ schema.]object  | DIRECTORY
                directory_name | DEFAULT}
          [ BY { SESSION  | ACCESS } ]
          [ WHENEVER  [ NOT ]  SUCCESSFUL ]
```

Mots clés et paramètres

object_opt	: option objet à auditer
object	: nom de l'objet à auditer
DEFAULT	: option d'audit par défaut pour les nouveaux objets à créer ultérieurement
BY SESSION	: audit des ordres sql et/ou privilèges par session
BY ACCESS	: audit des ordres sql et/ou privilèges par accès
SUCCESSFUL	: audit des ordres sql et/ou privilèges si succès
DIRECTORY	: audit des Directories

Audit des objets du schéma (suite)

Exemple 1

auditer l'ordre SQL *select* sur la table *tintin.emp* en cas d'échec d'exécution de cet ordre.

```
sql> AUDIT select ON tintin.emp
     WHENEVER NOT SUCCESSFUL ;
```

Exemple 2

auditer l'*insertion*, la *modification* et la *sélection* sur la table *haddock.dept* par accès.

```
sql> AUDIT insert, update, select
       ON haddock.dept BY ACCESS ;
```

Audit des objets du schéma (suite)

- **Les options d'audit des objets**

Options	Ordres SQL audités	Objet du schéma concerné
alter	alter table | sequence	table, séquence, MV, Obj. type
audit	audit (sur les objets)	table, vue, séquence, procédure
comment	comment table | view	table, vue
delete	delete from table | view	table | vue
execute	execute procedure	procédure
grant	grant (objets)	table, vue, séquence, procédure
index	create index on	table
insert	insert into table| view	table, vue
lock	lock table | view	table, vue
rename	rename ...	table, vue, procédure[1]
select	select * from ...	table, vue, snapshot
update	update table | vue	table, vue

Note : [1] procédure est mis ici pour: procédure, fonction ou package

all : remplace toutes les options d'audit des objets

Exploitation de l'audit

- la **table de base AUD$** contient l'ensemble des informations sur les audits
- les **vues d'audit** :Elles sont créées en lançant le script *cataudit.sql* et supprimées en lançant *catnoaudit.sql*

USER_AUDIT_CONNECT
USER_AUDIT_RESOURCE
DBA_AUDIT_DBA
DBA_PRIV_AUDIT_OPTS
DBA_TAB_AUDIT_OPTS
ALL_DEF_AUDIT_OPTS
DBA_STMT_AUDIT_OPTS
USER_OBJ_AUDIT_OPTS, DBA_OBJ_AUDIT_OPTS
USER_AUDIT_TRAIL, DBA_AUDIT_TRAIL
USER_AUDIT_STATEMENT, DBA_AUDIT_STATEMENT
USER_AUDIT_OBJECT, DBA_AUDIT_OBJECT
DBA_AUDIT_EXISTS,
USER_AUDIT_SESSION, DBA_AUDIT_SESSION
USER_TAB_AUDIT_OPTS

Exploitation de l'audit (suite)

Les colonnes des vues d'audit

Colonne	Description
OS_USERNAME	Nom au niveau OS de l'utilisateur à auditer
USERNAME	Nom Oracle de l'utilisateur à auditer
USERHOST	Numéro de l'instance de connexion du user
TERMINAL	identification du terminal de l'utilisateur
TIMESTAMP	date et heure d'enregistrement de l'action
OWNER	propriétaire de l'objet concerné par l'action
OBJ_NAME	nom de l'objet concerné par l'action
ACTION	code de l'action
ACTION_NAME	nom de l'action
NEW_OWNER	propriétaire de l'objet désigné dans la colonne new_name
NEW_NAME	nouveau nom de l'objet après RENAME
OBJ_PRIVILEGE	privilègesobjets affectés ou retirés à l'objet
SYS_PRIVILEGE	privilèges systèmes affectés ou retirés
ADMIN_OPTION	privilèges systèmes ou rôles donnés avec with ADMIN OPTION

Exploitation de l'audit (suite)

Les colonnes des vues d'audit (suite)

Colonne	Description
GRANTEE	nom de celui qui reçoit le privilège
AUDIT_OPTION	options d'audit positionnées avec AUDIT
SES_ACTIONS	chaîne de 11 caract. chacun vaut "-" pour none ou "S" pour success ou "F" pour failure ou "B" pour les deux. Chaque position correspond dans l'ordre aux actions : alter, audit, comment, delete, grant, index, insert, lock, rename, select, update.
LOGOFF_TIME	date et heure de déconnexion
LOGOFF_LREAD	nbre de lectures logiques dans la session
LOGOFF_PREAD	nbre de lectures physiques dans la session
LOGOFF_LWRITE	nbre de blocs Oracle modifiés dans session
LOGOFF_DLOCK	nbre de deadlocks détectés durant la session
COMMENT_TEXT	Commentaire sur la trace de l'action
SESSIONID	Numéro de la session Oracle
ENTRYID	Numéro de l'ordre exécuté
STATEMENTID	numéro de la requête exécutée
RETURNCODE	Erreur oracle provoquée par l'action
PRIV_USED	Privilèges systèmes utilisés par l'action
OBJECT_LABEL	Label de l'objet (Oracle sécurisé)
SESSION_LABEL	Label associé à la session (Oracle sécurisé)

Exploitation de l'audit (suite)

Exemple

Problème

Nous souhaitons effectuer l'audit sur les activités suspectes suivantes :

- l'affectation des mots de passes, *tablespace* et quotas pour certains utilisateurs ont étés modifiés sans autorisation
- un trop grand nombre de verrous exclusifs est posé
- des lignes sont supprimées arbitrairement de la table *emp* de l'utilisateur *tintin*

Nous suspectons pour cela les utilisateurs *rackham* et *lama*.

Actions effectuées par l'administrateur

```
sql>AUDIT ALTER, INDEX, RENAME ON DEFAULT
      BY SESSION ;
sql> CREATE VIEW tintin.employes
      AS SELECT * FROM tintin.emp ;
sql>AUDIT session BY rackham, lama;
sql>AUDIT alter user ;
sql>AUDIT lock table BY ACCESS WHENEVER SUCCESSFUL ;
sql> AUDIT delete ON tintin.emp BY ACCESS
      WHENEVER SUCCESSFUL ;
```


Exploitation d'audit (suite)

Exemple (suite)

Actions effectuées par *rackham*

```
sql>ALTER USER tintin QUOTA 0 ON USER_DATA;
sql>DROP USER dupont ;
```

Actions effectuées par *lama*

```
sql>LOCK TABLE tintin.emp IN EXCLUSIVE MODE ;
sql>DELETE FROM tintin.emp WHERE mgr=7698 ;
sql>ALTER TABLE tintin.emp
      ALLOCATE EXTENT (SIZE 100K) ;
sql>CREATE INDEX tintin.idx_ename ON
      tintin.emp(ename);
sql>CREATE PROCEDURE
       tintin.fire_employe(idemp NUMBER) AS
       BEGIN
         DELETE FROM tintin.emp WHERE empno=idemp;
       END;
       /
sql>EXECUTE tintin.fire_employe(7902);
```


Exploitation de l'audit (suite)

Exemple (suite)

Visualisation des options d'audit des ordres SQL actives

```
sql>SELECT * FROM dba_stmt_audit_opts ;
```

USER_NAME	AUDIT_OPTION	SUCCESS	FAILURE
LAMA	CREATE SESSION	BY ACCESS	BY ACCESS
RACKHAM	CREATE SESSION	BY ACCESS	BY ACCESS
	ALTER USER	BY ACCESS	BY ACCESS
	LOCK TABLE	BY ACCESS	NOT SET

Visualisation des options actives d'audit des privilèges

```
sql>SELECT * FROM dba_priv_audit_opts ;
```

USER_NAME	PRIVILEGE	SUCCESS	FAILURE
LAMA	CREATE SESSION	BY ACCESS	BY ACCESS
RACKHAM	CREATE SESSION	BY ACCESS	BY ACCESS
	ALTER USER	BY ACCESS	BY ACCESS

Exploitation de l'audit (suite)

Exemple (suite)

Visualisation des options d'audit objet sur la table *tintin.emp*

```
sql>SELECT * FROM dba_obj_audit_opts
     where owner = 'TINTIN' AND object_name like 'EMP%';
```

OWNER	OBJECT_NAME	OBJECT_TY	ALT AUD COM DEL GRA IND INS LOC ...
TINTIN	EMP	TABLE	-/- -/- -/- A/- -/- -/- -/--/- -/- -/- -/- -/- -/-...
TINTIN	EMPLOYES	VIEW	-/- -/- -/- -/- -/- -/- -/- -/--/- S/S -/- -/- -/- -/-...

Visualisation des options d'audit objet par défaut

```
sql>SELECT * FROM all_def_audit_opts;
```

```
ALT AUD COM DEL GRA IND INS LOC REN SEL UPD REF EXE
S/S  -/-  -/-  -/-  -/-  S/S  -/-  -/-  S/S  -/-  -/-  -/-  -/-
```

Notes :

-/- : Audit non positionné (ni en cas de succès ou d'insuccès)

S/- : Option d'audit positionnée par Session

A/- : Option d'audit positionnée par accès (en cas de succès)

S/S : Par session (en cas de succès ou d'insuccès)

■ Exploitation de l'audit (suite)

Exemple (suite)

Listing des lignes d'audit générées par l'audit des ordres et audit des objets

```
sql>SELECT username, obj_name, action_name, ses_actions
FROM dba_audit_object;
```

USERNAME	OBJ_NAME	ACTION_NAME	SES_ACTIONS
LAMA	EMP	LOCK	
LAMA	EMP	DELETE	
LAMA	IDX_ENAME	CREATE INDEX	

Listing des lignes d'audit pour l'audit des sessions

```
sql>SELECT username, logoff_time, logoff_lread, logoff_pread,
  logoff_lwrite, logoff_dlock
  FROM dba_audit_session ;
```

USERNAME	LOGOFF_TIME	LOGOFF_LREAD	LOGOFF_PREAD	LOGOFF_LWRITE	LOGOFF_DLOCK
LAMA					
RACKHAM	11/05/96	57	7	7	0

■ Désactivation de l'audit sur les Ordres SQL et les privilèges systèmes

Syntaxe

```
NOAUDIT  { statement_opt  | system_priv }
            [, { statement_opt | stystem_priv } ] ...
            [ BY  user [, user ] ...
            [ WHENEVER  [ NOT ]  SUCCESSFUL ]
```

Mots clés et paramètres

statement_opt	: option des ordres sql à ne plus auditer
system_priv	: privilège système à ne plus auditer
user	: désactiver l'audit des ordres SQL sur un user
SUCCESSFUL	: désactiver seulement en cas de succès

Exemples

```
sql>NOAUDIT session;
sql>NOAUDIT role;
sql>NOAUDIT session BY tintin, cyclone ;
sql>NOAUDIT delete any table;
sql>NOAUDIT select any table, insert any table,
      delete any table, execute any procedure;
sql>NOAUDIT select table, insert table, delete table;
sql>NOAUDIT ALL ;
sql>NOAUDIT ALL PRIVILEGES ;
```

Note : le privilège requis est AUDIT SYSTEM

■ Désactivation de l'audit des objets du schéma

Syntaxe

```
NOAUDIT  object_opt  [, object_opt] ...
            ON {[ schema.]object  | DEFAULT}
            [ WHENEVER  [ NOT ]  SUCCESSFUL ]
```

Mots clés et paramètres

object_opt	: option d'objet d'audit à désactiver
object	: nom de l'objet sur lequel l'audit va être désactivé
DEFAULT	: désactiver les options d'audit par défaut
SUCCESSFUL	: désactiver l'audit uniquement en cas de succès

Exemples

```
sql>NOAUDIT select ON tintin.emp ;
sql>NOAUDIT select
      ON tintin.emp WHENEVER SUCCESSFUL;
sql>NOAUDIT ALL ON tintin.emp ;
sql>NOAUDIT ALL ON DEFAULT ;
```

Note : le privilège requis est AUDIT SYSTEM

■ Administration et utilisation de l'audit

•1. **Activation de l'audit**. Positionner au niveau du fichier init.ora le paramètre init.ora :
. AUDIT_TRAIL=TRUE ou AUDIT_TRAIL=OS

•2. **installer les vues d'audit** (cataudit.sql)

•3. **Utiliser la commande AUDIT** pour positionner l'audit

•4. **Sécuriser la table d'audit** : attribuer le privilège *delete any table* à l'administrateur de sécurité uniquement. Auditer la table d'audit.
AUDIT insert, update, delete ON sys.aud$ BY ACCESS

•5. **Utiliser si nécessaire la commande NOAUDIT** pour annuler les positionnements faits avec AUDIT

•6. **Exploiter les résultats d'audit**

•7. **Purger ou réduire la taille de l'audit**
DELETE FROM sys.aud$
DELETE FROM sys.aud$ where obj$name='EMP'; ...

•8. **Désactiver l'audit** (dans init.ora AUDIT_TRAIL = NONE)

6 Sauvegarde et restauration

- **Plan**

  - 6.1 Généralités
  - 6.2 Sauvegarde en mode NOARCHIVELOG
  - 6.3 Sauvegarde en mode ARCHIVELOG
  - 6.4 Restauration d'une Base de données
  - 6.5 Recovery Manager (voir Annexe)

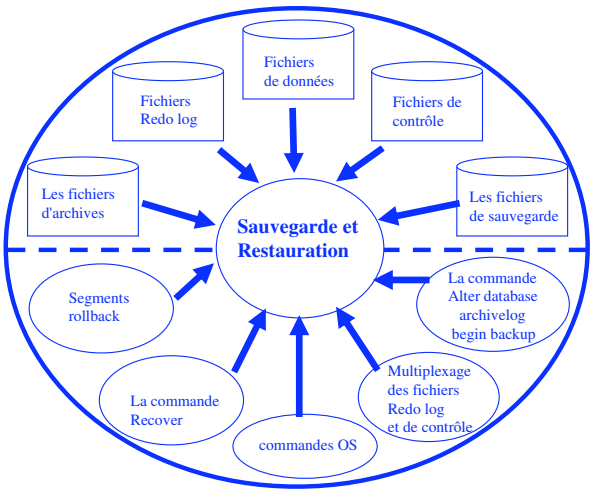
6.1 Généralités

- **Bien définir une stratégie de sauvegarde et restauration**
- **Les questions à se poser**
  - est - t - il acceptable de perdre des données en cas de panne des fichiers de données ?
  - est - t - il utile de recouvrer des données perdues ?
  - la base de données ne doit - t - elle jamais s'arrêter ?
- **Bien tirer partie des mécanismes d'Oracle pour se prémunir des pertes de données**
  - technique **de mise en miroir** de fichier redo log et de contrôle
- **Tester votre stratégie de sauvegarde et restauration**
- **adapter le rythme des sauvegardes par rapport au besoins**
- **Conserver les sauvegardes tant que utile**

Bien choisir son mode d'archivage (ARCHIVELOG, NOARCHIVELOG) car de ce choix dépendra les possibilités de restauration.

6.1 Généralités

- **Composants Oracle pour la sauvegarde et restauration**



6.2 Sauvegarde en Noarchivelog

- **Hypothèses**

  - nous acceptons de perdre les données
  - l'arrêt de la base n'est pas génant

- **Mode de sauvegarde**

  - **Backup complet** à la création de la base et à chaque modification importante de sa structure (ajout d'un fichier, d'un tablespace, ...). Ce **backup doit se faire en mode normal** après un SHUTDOWN NORMAL ou IMMEDIATE

  - les fichiers à sauvegarder sont :
    - les fichiers de données de tous les tablespaces
    - les fichiers REDO LOG
    - les fichiers de contrôle

  - **NOTE** :
    Il faut aussi définir une périodicité des sauvegardes adaptées aux besoins.

6.2 Sauvegarde en Noarchivelog

Les étapes de sauvegarde en mode sans archive

1. **liste les noms des fichiers à sauvegarder**
   . Fichiers de données
   SELECT * FROM dba_data_files

   . Fichiers de contrôles
   SELECT * FROM v$parameter
   WHERE name LIKE 'control_files';

   . Les fichiers Redolog
   SELECT * FROM v$logfile

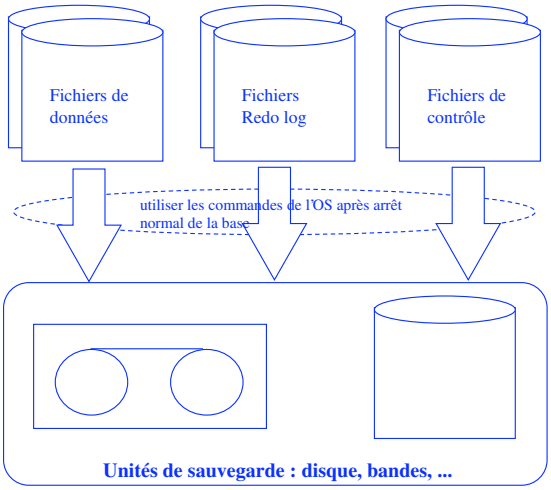
2. **Arrêter la base normalement** (shutdown normal ou immediate) et utiliser les commandes de l'OS (tar, cp, rcp, dump, copy, arj, ...) pour effectuer les sauvegardes

3. **redémarrer la base** après un CONNECT sys as sysdba
   SQL >startup

4. **Effectuer en fonction de la quantité des données qu'on accepte de perdre des *exports* réguliers** et dimensionner si utile les fichiers Redo log en fonction

6.2 Sauvegarde en Noarchivelog

Sauvegarde complète de tous les fichiers de la base



6.3 Sauvegarde en mode Archivelog

Hypothèses

- Nous n'acceptons pas de perdre des données
- Nous n'acceptons pas l'arrêt total de la base

Modes de sauvegarde

- Backup complet à la création ou à intervalles réguliers (**suivre les mêmes étapes qu'en mode *noarchivelog***)
- Backup partiel de la base
- Archivage **automatique ou manuelle** de fichiers REDO LOG
- Backup du fichier de *contrôle* en cas de modification de la structure de la base (ajout d'un tablespace ...)

6.3 Sauvegarde en mode Archivelog

Backup partiel

1. Backup d'un Tablespace OFFLINE en 4 étapes

Etape 1 :

. Identifier les fichiers appartenant au tablespace X

```
sql>SELECT file_name FROM sys.dba_data_files
    WHERE tablespace_name = 'X';
```

Etape 2 :

. Mettre le Tablespace X OFFLINE NORMAL

```
sql>ALTER TABLESPACE X OFFLINE NORMAL ;
```

Etape 3 :

. Utiliser les commandes de l'OS (cp, tar, dump, ...) pour sauvegarder effectivement les fichiers sur bande ou disque

Etape 4 :

. Réactiver le Tablespace X

```
sql>ALTER TABLESPACE X ONLINE ;
```

6.3 Sauvegarde en mode Archivelog

■ Backup partiel d'une base

2. Backup d'un Tablespace (TS) ONLINE en 4 étapes

Etape 1 :

. Identifier les fichiers appartenant au tablespace X

```
sql>SELECT file_name FROM sys.dba_data_files
      WHERE tablespace_name = 'X';
```

Etape 2 :

. Indiquer à Oracle le début de la sauvegarde de X

```
sql>ALTER TABLESPACE X BEGIN BACKUP ;
```

Etape 3 :

. Utiliser les commandes de l'OS (cp, tar, dump, ...) pour sauvegarder effectivement les fichiers sur bande ou disque

Etape 4 :

. Informer Oracle de la fin de la sauvegarde du tablespace X

```
sql>ALTER TABLESPACE X END BACKUP ;
```

Note : La sauvegarde de plusieurs TS peut se faire séquentiellement ou en parallèle BEGIN BACKUP X BEGIN BACKUP Y - sauvegardes - END BACKUP X END BACKUP Y.

6.3 Sauvegarde en mode Archivelog

■ Sauvegarde du fichier de contrôle

1. Sauvegarde du fichier de contrôle Base fermée

Utiliser les commandes de l'OS pour sauvegarder

2. Sauvegarde du fichier de contrôle Base Ouverte

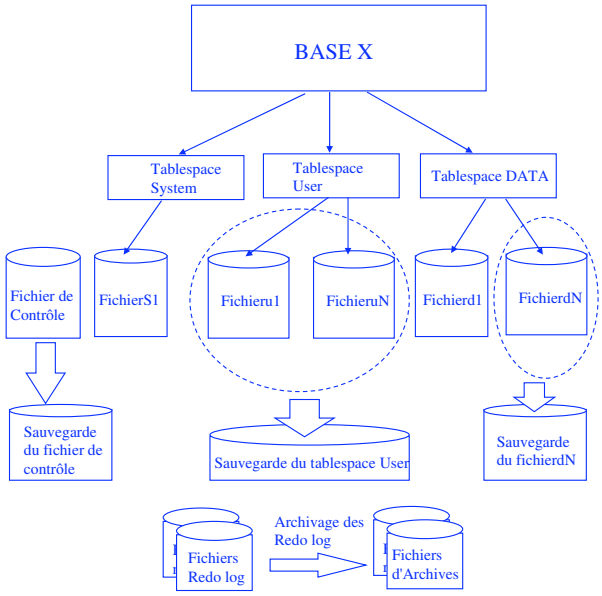
```
sql>ALTER DATABASE BACKUP CONTROLFILE
      TO TRACE ;
    # Génère le texte de la commande de création du fichier de
    # contrôle
ou
sql>ALTER DATABASE
      BACKUP CONTROLFILE nomfic [REUSE] ;
    # Génère une copy du fichier de contrôle
```

1. Les EXPORTS sont - t - ils nécessaires en mode ARCHIVELOG

2. Pourquoi est - t - il inutile de sauvegarder les REDO LOG en mode NOARCHIVELOG

6.3 Sauvegarde en mode Archivelog

■ Synthèse



6.4 Restauration d'une Base

■ Pourquoi restaurer ?

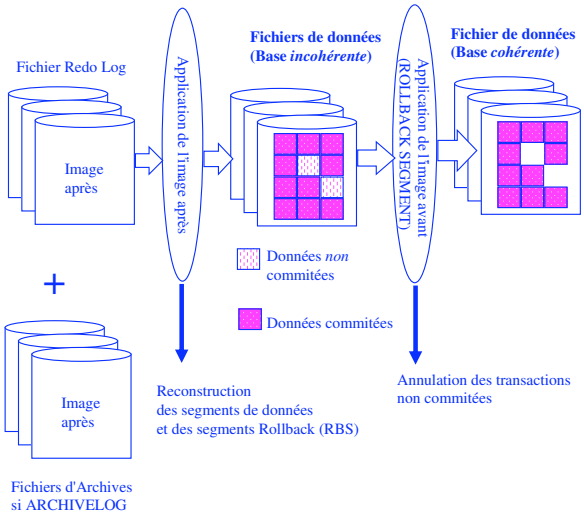
- Au moment d'une panne, les blocs de données modifiés par une transaction peuvent ne pas avoir été écrits dans les fichiers REDO LOG
- Le fichier REDO LOG peut contenir des données non validées
- Les fichiers Redo log, les fichiers d'archives, les fichiers de contrôles ou les fichiers de données peuvent être corrompus ou perdus

■ Deux types de recouvrement

- Recouvrement d'instance (arrêt brutal d'un ou plusieurs process Oracle tel DBWR, LGWR, ...)
- Recouvrement après une panne disque

6.4 Restauration d'une base

Principe de la restauration des données



6.4 Restauration d'une base

Restauration d'instance

- Survient à la suite de l'arrêt brutal des process tâches de fond (DBWR, LGWR, SMON, PMON, ...)
- est valable aussi bien en mode avec ou sans archive.
- Oracle réalise ***automatiquement*** les étapes suivantes :
  - application de l'image après
  - application de l'image avant.

6.4 Restauration d'une base

Restauration après une panne disque : En mode NOARCHIVELOG

- Si **perte d'un fichier de données** : repartir d'un Backup à froid
- Si **perte d'un fichier Redo Log** : récupérer le fichier en miroir sinon restaurer la base entière
- Si **perte d'un fichier de contrôle** : récupérer le fichier en miroir ou recréer le fichier de contrôle sinon restaurer la base entière!!!!!!!!!!!!!!!

6.4 Restauration d'une base

Restauration après une panne disque : En mode ARCHIVELOG

La commande RECOVER

* Cette commande est l'outil de recouvrement en mode ARCHIVELOG
* c'est aussi une clause de la commande ALTER DATABASE (forme d'utilisation déconseillée par Oracle)
* privilège requis : ALTER DATABASE

Syntaxe

```
[ALTER DATABASE]
  RECOVER [ AUTOMATIC ] [ FROM  'location' ]
    { [ DATABASE ] [ UNTIL  CANCEL
                   | UNTIL TIME date
                   | UNTIL CHANGE integer
                   | UNTIL BACKUP  CONTROLFILE ]
    | TABLESPACE tablespace [, tablespace ] ...
    | DATAFILE  'filename' [, 'filename' ] ...
    | LOGFILE 'filename'
    | CONTINUE [ DEFAULT ]
    | CANCEL }
```

6.4 Restauration d'une base

Restauration après une panne disque : En mode ARCHIVELOG (suite)

La commande RECOVER (suite)

Mots clés ou paramètres	Description
AUTOMATIC	générer automatiquement les noms des fichiers Redo à appliquer
FROM location	indiquer la localisation des archives
DATABASE	recouvrer la base entière (par défaut)
UNTIL CANCEL	recouvrer jusqu'au fichier Redo log le plus récent possible
UNTIL TIMES date	recouvrer jusqu'à la date ~ de la panne
UNTIL CHANGE integer	recouvrer jusqu'aux transactions consistantes avant le SCN (integer)
UNTIL BACKUP CONTROLFILE	Utiliser la sauvegarde du fichier de contrôle à la place du fichier courant
TABLESPACE tablespace	recouvrer uniquement un tablespace
DATAFILE filename	recouvrer un fichier de données
LOGFILE filename fichier	continuer le recouvrement avec ce
CONTINUE [DEFAULT]	continuer en appliquant les redo automatiquement
CANCEL	finir la restauration basée sur l'annulation

6.4 Restauration d'une base

Restauration après une panne disque : En mode ARCHIVELOG (suite)

- **Recouvrement complet base fermée**

CAS OU CELA EST UTILE:

- perte d'un ou plusieurs **fichiers de données** du **Tablespace System**
- perte d'un **fichier de contrôle** (pas de copie)
- perte **d'un fichier de données** contenant un segment RBS
- un fichier pour les RBS a été endommagé
- la base ne peut être ouverte en mode NORMAL
- **NOTES** :
  - pas de perte des Redo Log courants
  - pas de perte des fichiers d'archives

6.4 Restauration d'une base

Restauration après une panne disque : En mode ARCHIVELOG (suite)

- **Recouvrement complet base fermée (suite)**

Etapes à suivre

1. arrêter la base
2. remédier à la panne
3. reprendre la dernière sauvegarde et les fichiers d'archives
4. redémarrer la base en mode MOUNT
   sql>connect sys as sysdba;
   sql >startup MOUNT ;
5. Renommer ou rélocaliser les fichiers si nécessaire
   sql>ALTER DATABASE RENAME FILE ... TO ...
6. Commencer le RECOUVREMENT
   sql>RECOVER AUTOMATIC
   DATABASE ...;
7. Ouvrir normalement la base
   sql>ALTER DATABASE OPEN ;

6.4 Restauration d'une base

Restauration après une panne disque : En mode ARCHIVELOG (suite)

- **Recouvrement complet base ouverte, tablespace OFFLINE**

CAS OU CELA EST UTILE :

- les fichiers d'un ou plusieurs tablespaces sont endommagés
- les fichiers du tablespace SYSTEM ne sont pas endommagés
- les fichiers contenant les RBS ne sont pas endommagés

Dans quels états doivent être les fichiers REDO LOG et d'ARCHIVES ?

6.4 Restauration d'une base

■ Restauration après une panne disque : En mode ARCHIVELOG (suite)

Recouvrement complet base ouverte, tablespace OFFLINE (suite)

Etapes à suivre

1. Si la Base n'est pas déjà ouverte, démarrer une nouvelle instance (base montée puis ouverte). Mettre le tablespace ayant des fichiers endommagés OFFLINE.

   sql>connect sys as sysdba
   (sql>STARTUP MOUNT puis OPEN ;)
   sql>ALTER TABLESPACE nomtablespace OFFLINE ;

2. Remédier à la panne
3. Remplacer les fichiers endommagés par leur sauvegarde
4. Recouvrer les fichiers endommagés

   sql >RECOVER TABLESPACE
   nomtablespace1, ..., nomtablespaceN;
   sql>RECOVER DATAFILE nomfichier1, ..., nomfichierN;

5. Remettre le ou les tablespaces ONLINE
   sql>ALTER DATABASE nomtablespace1, ... ONLINE ;

6.4 Restauration d'une base

■ Restauration après une panne disque : En mode ARCHIVELOG (suite)

- **Recouvrement incomplet**

- Le recouvrement ***incomplet*** est nécessaire :
  . en cas de perte des fichiers Redolog
  . en cas de suppression accidentelle d'une table
  . ...

- Le recouvrement ***incomplet*** peut être basé sur :
  . l'***annulation*** (UNTIL CANCEL). En cas de perte d'un ou plusieurs groupes de Redo Log, le recouvrement s'arrête sur le fichier Redo Log le *plus récent*
  . le ***temps*** (UNTIL TIME date) : connaissant la date approximative à laquelle est survenue une panne, on souhaite récupérer l'activité sur la base d'avant cette date
  . le ***SCN*** (UNTIL CHANGE scn) : connaissant un Système Change Number donné, on souhaite récupérer toutes les transactions consistantes d'avant ce SCN

6.4 Restauration d'une base

■ Restauration après une panne disque : En mode ARCHIVELOG (suite)

- **Recouvrement incomplet (suite)**

Etapes du recouvrement basé sur l'annulation, le temps et le SCN

1. Fermer si utile la base avec l'option ABORT
   sql >connect sys as sysdba ;
   sql >shutdown abort ;
2. Sauvegarder tous les fichiers de la base (données, contrôles, paramètres, ...)
3. Remédier à la panne disque si nécessaire
4. Si le fichier **de contrôle** ne correspond pas, chercher une bonne version sauvée ou recréer le fichier de contrôle
5. Récupérer la sauvegarde des fichiers de la base mis en cause. **Si aucune sauvegarde n'existe, recréer des fichiers de même taille vide**
6. sous sql, démarrer une instance (base montée)

(à suivre)

6.4 Restauration d'une base

■ Restauration après une panne disque : En mode ARCHIVELOG (suite)

- **Recouvrement incomplet (suite)**

Etapes du recouvrement basé sur l'annulation, le temps et le SCN (suite)

7. Rélocaliser ou renommer si utile les fichiers endommagés
8. Mettre les fichiers à réparer ONLINE si recouvrement basé sur le temps ou le SCN
9. Utiliser les commandes suivantes pour effectuer le recouvrement :
   sql> RECOVER database UNTIL CANCEL ;
   ou
   sql >RECOVER database
   UNTIL TIME 'YYYY-MM-HH24:MI:SS' ;
   ou
   sql>RECOVER database
   UNTIL CHANGE scn;

   Note : si l'on utilise la sauvegarde du fichier de contrôle, préciser l'option USING BACKUP ...

6.4 Restauration d'une base

■ Récapitulatif des pertes et actions (1/2)

Type de fichiers endommagés				Mode d'archivage	
Fichiers de données	Fichiers redo log	Fichiers redo log archives	Fichiers de contrôles	Archivelog	Noarchivelog
X				recouvrement complet	partir d' une sauve-garde complète
	X			Recouvrer le fichier manquant	Recouvrer le fichier manquant
		X		Faire un backup de tous les fichiers de données	sans effet
			X	recouvrer le fichier de contrôle	recouvrer le fihchier de contrôle
X	X	X	X	recouvrer le fichier de contrôle puis faire un recouvre-ment incomplet	sans effet
X	X	X		Faire un recouvre-ment incomplet	sans effet
X		X	X	Recouvrer le fichier de contrôle puis recouvrement incomplet	sans effet
X	X		X	Recouvrer le fichier de contrôle puis recouvrement incomplet	Recouvrer le fichier manquant

6.4 Restauration d'une base

■ Récapitulatif des pertes et actions (2/2)

Type de fichiers endommagés				Mode d'archivage	
Fichiers de données	Fichiers redo log	Fichiers redo log archivés	Fichiers de contrôles	Archivelog	Noarchivelog
X	X			recouvrement incomplet	partir d' une sauve-garde complète
X		X		recouvrement incomplet	sans effet
X			X	Recouvrer le fichier de contrôle puis les les fichiers de données	partir d' une sauve-garde complète
	X	X	X	Recouvrer le fichier de contrôle. Faire Reset redo log ...	sans effet
	X	X		Faire un recouvre-ment incomplet	sans effet
	X		X	Recouvrer le fichier de contrôle puis les les fichiers de données *puis sauver*	Recréer un fichier de contrôle si arrêt nor-mal sinon partir d'un backup complet
		X	X	Recouvrer le fichier de contrôle puis fai-re un recouvrement incomplet	sans effet

6.4 Restauration d'une base

■ Exemples de problèmes et restauration

Exemple 1: perte d'un fichier de données en NOARCHIVELOG

Solution : En cas de perte d'un fichier de données en mode NOARCHIVELOG, on doit répartir d'un backup à froid

Etapes

1. Si la base est encore ouverte, faire un shutdown abort
   sql > shutdown abort ;
2. Si la panne disque est corrigée (les fichiers de données peuvent reprendre leur ancienne localisation) faire 2.1 puis 3. Sinon faire 2.1, 2.2, 2.3 puis 3
   2.1 Récupérer le plus récent backup (tous les fichiers pas seulement ceux endommagés).
   $ cp, tar, ...
   2.2 modifier si utile la localisation des fichiers de contrôle CONTROL_FILES=(new_path/nom1, new_path/nom2,...)
   2.3 démarrer l'instance
   sql >startup mount nombase pfile=path/init*sid*.ora
   2.4 Procéder si nécessaire au renommage et au déplacement des fichiers. sql>ALTER DATABASE RENAME ...
3. Ouvrir la base en réinitialisant si nécessaire les Redo Log
   sql >ALTER DATABASE OPEN RESETLOGS ;

6.4 Restauration d'une base

■ Exemples de problèmes et restauration(suite)

Exemple 2 : perte d'un fichier de données en ARCHIVELOG occasionnant l'arrêt de la base

1. Si la base est encore ouverte, faire un shutdown abort
   sql>connect sys as sysdba;
   sql> shutdown abort ;
2. A partir de la sauvegarde la plus récente, reconstruire uniquement les fichiers perdus (vers l'ancienne localisation ou vers une nouvelle localisation)
   $ cp ...
   sql> startup mount nombase pfile=path/init*sid*.ora
   sql>ALTER DATABASE RENAME ...
3. Effectuer un recouvrement de données avec la commande RECOVER en donnant les noms des archives à appliquer

   sql>RECOVER DATABASE ...

4. Ouvrir normalement la base

   sql>ALTER DATABASE OPEN ;

6.4 Restauration d'une base

■ Exemples de problèmes et restauration(suite)

Exemple 3 : perte d'un fichier redo log : plusieurs solutions sont possibles

1. Les Fichiers Redo Log sont multiplexés

   1.1 si le Fichier Redo Log est le fichier courant, provoquer un SWITCH LOG FILE pour passer au groupe suivant.

   sql>ALTER SYSTEM SWITCH LOGFILE ;

   1.2 Supprimer les fichiers à problème et les remplacer par la copie (miroir)

2. Un groupe de fichiers Redo Log entier est Inaccessible

   2.0 Recréer le groupe endommagé

   2.1 Le groupe n'était pas actifs alors une réparation Base ouverte est possible

   2.2 Si le groupe était courant, arrêter la base

   2.3 Redémarrer la base et faire si nécessaire un

   RECOVER DATABASE UNTIL CANCEL

6.4 Restauration d'une base

■ Exemples de problèmes et restauration(suite)

Exemple 4 : un utilisateur a supprimé accidentellement ses tables (mode avec Archive)

1. Sauvegarder la base existante (BE)

2. Reconstruire une copie temporaire de la base (BT)

   - Faire une Recouvrement basé sur le temps (date approximative de la suppression de données)

   sql>RECOVER DATABASE UNTIL TIME date;

3. Export les données supprimées accidentellement

   $ **exp** system/manager **owner**=nomuser

4. Supprimer la base temporaire

   $ rm ... /* des fichiers de données, contrôles et Redo log*/

5. Revenir sur la base courante (BE) et importer les données de l'utilisateurs

   $ **imp** system/manager fromuser=nomuser touser=nomuser

6.4 Restauration d'une base

■ Exemples de problèmes et restauration(suite)

Exemple 5 : un tablespace a été supprimé accidentellement

1. Sauvegarder la base existante

2. Reconstruire une copie temporaire de la base

   - Faire une Recouvrement basé sur le temps (date approximative de la suppression de données) ou le SCN

   sql>RECOVER DATABASE UNTIL TIME date;

   ou

   sql>RECOVER DATABASE UNTIL CHANGE scn ;

3. Exporter les données du tablespace supprimées accidentellement
   $ **exp** system/manager ...

4. Supprimer la base temporaire
   $ **rm** ... /* des fichiers de données, contrôles et redos*/

5. Revenir sur la base courante et importer les données exportées
   $ **imp** system/manager...

7. Les outils d'administration et les NLS

■ Plan

7.1 Export/Import

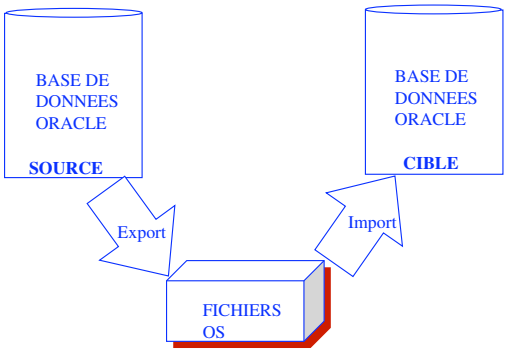
- **Plan**
  - 7.1.1 Généralités
  - 7.1.2 Export
  - 7.1.3 Import

7.1.1 Généralités

- **Pourquoi Exporter et importer ?**
  - Archivage de données
  - Upgrade vers de nouvelles versions
  - Sauvegarde et restaurations de données
  - Déplacement de données entre bases Oracle
- **Tâches réalisables avec Export/Import**
  - Stocker des données *indépendamment* du SGBD
  - *extraire* uniquement *le schéma* de données
  - *stocker* des données utiles *temporairement* ou *inactives*
  - opérer des sauvegardes intelligentes (un ou plusieurs objets, un ou plusieurs utilisateurs, toute la base, ...)
  - faire des *exports / imports incrémentaux* ou cumulatifs
  - *déplacer des données* (entre versions Oracle, entre bases, entre environnements, entre tablespaces, ...)
  - *réduire la fragmentation* des données
  - *corriger les paramètres de stockage*

7.1.1 Généralités

- **Représentation schématique de l'import/export**



Les bases de données Oracle **source** et **cible** peuvent être :
- la même
- différentes sur une même machine
- différentes sur deux machines différentes
- ...

7.1.2 Export

- **Que peut - t - on exporter ?**

1. tables

   Export d'une ou N tables dans un schéma. Le DBA peut exporter les tables de N schéma

2. Utilisateur ou schéma

   Export de tous les objets appartenant au schéma c'est à dire à un utilisateur (tables, vues, séquences, données, privilèges, index, ...)

3. La base entière

   Export de toute la base sauf les objets de l'utilisateur SYS d'Oracle. Il est utile d'avoir le privilège EXP_FULL_DATABASE.

Pourquoi est - t - il interdit de travailler SYS ?
Pourquoi est - t - il si fondamental d'avoir le privilège EXPORT_FULL_DATABASE ?

7.1.2 Export

Que peut - t - on exporter ?

Mode Table	Mode Utilisateur	Mode FULL
schéma de la table données de la table grant sur les tables Index sur la table contraintes sur la table trigger de la table Types utilisateurs Tables imbriquées Indexes d'autres users Triggers d'autres users	schéma de la table données de la table grant fait sur les tables Index sur la table contraintes sur la table trigger de la table Types utilisateurs Tables imbriquées Indexes d'autres users Triggers d'autres users cluster databases link job queue refresh group séquences snapshot snapshot log procédure stockée synonymes privés Les vues	schéma de la table données de la table grant sur les tables Index sur la table contraintes sur la table trigger de la table Types utilisateurs Tables imbriquées Indexes d'autres users Triggers d'autres users cluster databases link job queue refresh group séquences snapshot snapshot log procédure stockée synonymes privés Les vues profiles catalogue de réplication coût des ressources rôles définition des RBS Options d'audit système privilèges systèmes définition des tablespaces quotas sur les tablespaces définition des utilisateurs

7.1.2 Export

Utilisation de l'utilitaire EXPORT

- s'assurer que le script catview.sql a été exécuté
- s'assurer qu'on a les privilèges qu'il faut :
  - être propriétaire d'un schéma permet d'exporter les objets du schéma
  - Avoir le privilège BACKUP ANY TABLE pour exporter les objets d'un autre utilisateur
  - Avoir le privilège EXP_FULL_DATABASE pour pouvoir exporter les objets de toute la base
- Activer l'export dans l'un des modes suivants :
  - mode interactif (ne supporte pas toutes les commandes)
  - mode commande
  - mode batch
  - Via Entreprise Manager

7.1.2 Export

Les paramètres d'exports

- Ces paramètres permettent de fixer les options d'export
- Ces paramètres peuvent être définis de façon :
  - implicite (si l'on est en mode interactif)
  - explicite (si l'on est en mode commande et batch)
- Aide en ligne sur les paramètres
  $ exp help=y

7.1.2 Export

Les paramètres d'exports (suite)

Nom paramètre	Description	Valeur par défaut	Utilisé en mode interactif ?
userid	username/password	none	
buffer	rows_in_array*max_row_size	4096	
file	nom du fichier en sortie	expdat.dmp	
compress	compresser les extents en une	Y	
grants	exporter aussi les privilèges	Y	
indexes	exporter les index	Y	
rows	exporter les lignes	Y	
constraints	exporter les contraintes	Y	
log	fichier log	none	non
full	exporter toute la base	N	
owner	utilisateur(s) à exporter	undefined	
tables	tables à exporter	undefined	
record lenght	taille(bytes) de l'enreg. du fichier	OS dépendant	non
inctype	incrémental, cumulatif, complet ?	none	non
record	utile si inctype est mis	Y	non
parfile	fichier des paramètres	none	non
consistent	lecture consistante (read only mis)	N	non
statistics	type de stat à générer en import	estimate	non
help	aide	N	

Notes: certains paramètres sont mutuellement exclusifs(table, user et full)

7.1.2 Export

Les paramètres d'exports (suite)

Nom paramètre	Description	Valeur par défaut	Utilisé en mode interactif ?
Direct	Chemin direct	N	
feedkack	affiche la progression toute les N lignes	0	
Consistent	cohérence entre les tables		
filesize	taille max de chaque fichier		
resumables	suspension en cas d'erreurs d'espace	N	
Transport_tablespace	impport des métadonnées des TS	N	
Tablespace	TS à transporter		
Datafile	fichier à transporter		
TTS_OWNER	Utilisateurs propriétaires des données		

Notes: certains paramètres sont mutuellement exclusifs(table, user et full)

7.1.2 Export

Export en mode interactif et commande

. Export en mode interactif (options complètes)

```
$ exp
User name :
password :
Enter array fetch buffer size : 4096 >
Export file : expdat.dmp >
E (ntire database) U(ser), T(ables) : U>
Export grants (Y/N) : Y>
Export table data (Y/N) : Y>
Compress extents (Y/N) :Y>
Schema to export : (return to quit)>
table to be exported : (return to quit)>
```

. Export en mode commande

$ exp [username/passwd] [param1 = value 1] ... [paramN = value N]

Notes : 1) un des paramètres au moins doit être mis explicitement
2) si table, owner ou full n'est pas mis, il s'agit d'exporter l'utilisateur qui fait l'export

7.1.2 Export

Export en mode interactif et commande (suite)

Exemple 1 : export de la table emp de l'utilisateur tintin

a) en mode commande

$ exp tintin/milou file=tintin_emp.dmp tables=emp

b) en mode interactif

```
$ exp tintin/milou
  Enter array fetch buffer size : 4096 >
  Export file : expdat.dmp >tintin_emp.dmp
  U(ser), T(ables) : U>T
  Export grants (Y/N) : Y>
  Export table data (Y/N) : Y>
  Compress extents (Y/N) :Y>
  table to be exported : (return to quit)>emp
```


7.1.2 Export

Export en mode interactif et commande (suite)

Exemple 2 : tous les objets de l'utilisateur tintin

a) en mode commande

$ exp tintin/milou file=tintin.dmp

b) en mode interactif

```
$ exp tintin/milou
  Enter array fetch buffer size : 4096 >
  Export file : expdat.dmp >tintin.dmp
  U(ser), T(ables) : U>U
  Export grants (Y/N) : Y>
  Export table data (Y/N) : Y>
  Compress extents (Y/N) :Y>
```

7.1.2 Export

Export en mode interactif et commande (suite)

Exemple 3 : export des objets d'un utilisateur par un autre

a) en mode commande

```
$ exp system/manager owner=rackham file=rackham.dmp
```

b) en mode interactif

```
$ exp system/manager
  Enter array fetch buffer size : 4096 >
  Export file : expdat.dmp >rackham.dmp
  E(ntire database) U(ser), T(ables) : U>U
  Export grants (Y/N) : Y>
  Export table data (Y/N) : Y>
  Compress extents (Y/N) :Y>
  Schema to export: (return to quit)>rackham
```


7.1.2 Export

Export en mode interactif et commande (suite)

Exemple 4 : export de la la base entière

a) en mode commande

```
$ exp system/manager full=y
```

b) en mode interactif

```
$ exp system/manager
  Enter array fetch buffer size : 4096 >10000
  Export file : expdat.dmp >cours.dmp
  E(ntire database) U(ser), T(ables) : U>E
  Export grants (Y/N) : Y>
  Export table data (Y/N) : Y>
  Compress extents (Y/N) :Y>
```


7.1.2 Export

Export en mode batch

- les paramètres sont passés dans un fichier de paramètres
- permet de s'affranchir de la limite de la taille de la ligne de commande
- permet de faciliter l'automatisation de l'export
- Syntaxe admise dans le fichier des paramètres
  MOT CLE= valeur
  MOT CLE= (valeur)
  MOT CLE=(Valeur 1, Valeur 2, ..., Valeur N)
- Exemple de **fichier de paramètres** PARFIC.PAR
  FULL=Y
  FILE=cours.dmp
  GRANT=Y
  INDEX=y
- Activation $ exp parfile=parfic.par

7.1.2 Export

Export incrémental, cumulatif et complet

- **Généralités**
  - stratégie de sauvegarde étalée dans le temps
  - valide uniquement si FULL = Y et si l'utilisateur qui le fait a le privilège EXP_FULL_DATABASE
  - permet de racourcir les temps d'exports
- **trois stratégie de backup incrémental**

Type de stratégie	Description
Backup incrémental	Backup uniquement des objets modifiés depuis le dernier backup Incrémental ou Cumulatif ou Complet.
Backup cumulatif	Backup uniquement des objets modifiés depuis le dernier backup Cumulatif ou Complet
Backup complet	Backup de tous les objets.

7.1.2 Export

■ Export incrémental, cumulatif et complet (suite)

Exemple d'export incrémental

Type d'export	Objet présent	objet exporté
1er export Incrémental	**table1**, index1, **table2** , table3, table4	table1, table2
2ème export Incrémental	**table1**, index1 table2 , table3, table4 **table5, table6**	table1, table5, table6
1er Export Cumulatif	**table1**, index1 **table2** , table3, table4 **table5, table6**	table1, table2,table5,table6 (fusion des fichiers d'export incrémental)
1er Export Complet	**table1**, index1 **table2** , table3, table4 **table5, table6**	table1, index1 table2 , table3, table4 table5, table6

Exemple de session d'export incrémental

$ exp system/manager inctype=incremental
$ exp system/manager inctype=cumulative
$ exp system/manager inctype=complete

Quel est la différence entre "export full database" et "export complete" ?

7.1.2 Import

■ Que peut - t - on importer ?

1. tables

import d'une ou N tables vers un schéma. Le DBA peut importer les tables de N schéma

2. Utilisateur ou schéma

Import de tous les objets appartenant au schéma c'est à dire à un utilisateur (tables, vues, séquences, données, privilèges, index)

3. La base entière

Import de toute la base. Il est utile d'avoir le privilège EXP_FULL_DATABASE.

7.1.3 Import

■ Que peut - t - on importer ?

Mode Table	Mode Utilisateur	Mode FULL
schéma de la table données de la table grant sur les tables Index sur la table contraintes sur la table trigger de la table Types utilisateurs Tables imbriquées Indexes d'autres users Triggers d'autres users	schéma de la table données de la table grant fait sur les tables Index sur la table contraintes sur la table trigger de la table Types utilisateurs Tables imbriquées Indexes d'autres users Triggers d'autres users cluster databases link job queue refresh group séquences snapshot snapshot log procédure stockée synonymes privés Les vues	schéma de la table données de la table grant sur les tables Index sur la table contraintes sur la table trigger de la table Types utilisateurs Tables imbriquées Indexes d'autres users Triggers d'autres users cluster databases link job queue refresh group séquences snapshot snapshot log procédure stockée synonymes privés Les vues profiles catalogue de réplication coût des ressources rôles définition des RBS Options d'audit système privilèges systèmes définition des tablespaces quotas sur les tablespaces définition des utilisateurs

7.1.3 Import

■ Utilisation de l'Import

- **s'assurer que le script catview.sql a été exécuté**
- **s'assurer qu'on a les privilèges qu'il faut :**
  - import d'objet dans son propre schéma :
    - avoir les privilèges utiles pour créer les objets concernés. ***Exemple*** : import d'une table, avoir le privilège *Create table*
    - avoir les quotas nécessaires sur les tablespaces
  - importer les grants (2 cas)
    - être propriétaire de l'objet
    - ou avoir reçu le privilège avec With Grant Option ou With Admin Option
  - importer les objets vers un autre schéma : Avoir le privilège EXP_FULL_DATABASE
  - importer les objets systèmes (profiles, DB link, la définition des tablespaces, ...). Avoir le privilège EXP_FULL_DATABASE
- **activer l'import :** en interactif, en mode commande ou en batch

7.1.3 Import

Les paramètres d'Import

- Ces paramètres permettent de fixer les options d'import
- Ces paramètres peuvent être définis de façon :
  - implicite (si l'on est en mode interactif)
  - explicite (si l'on est en mode commande et batch)
- Aide en ligne sur les paramètres
  $ imp help=y

7.1.3 Import

Import en mode interactif et commande

Nom paramètre	Description	Valeur par défaut	Utilisé en mode interactif ?
buffer	rows_in_array*max_row_size	4096	oui
charset	jeux de caractère utilisé à l'export	none	
commit	commit après chaque tableau	N	
destroy	inclus l'option reuse de fichier	N	
feedback	Nbre de lignes à afficher / objet	0 (toutes)	
file	nom du fichier en entrée	expdat.dmp	oui
fromuser	liste de schémas à importer	NONE	
full	importer toute la base	N	oui
grants	importer aussi les prvilèges	Y	oui
help	aide	N	
ignore	ignorer les erreur de création	N	oui
inctype	incrémental, cumulatif, complet ?	none	
indexes	exporter les index	Y	
indexfile	y insérer les create index	NONE	
log	fichier log	none	
mls*	utile si trusted Oracle		
record lenght	taille(bytes) de l'enreg. du fichier	OS dépendant	
rows	importer les lignes ?	Y	oui
show	visualiser au lieu d'importer	N	oui
tables	tables à importer	non	
touser	schéma cible	none	
userid	username/password	none	oui

7.1.3 Import

Les paramètres d'imports (suite)

Nom paramètre	Description	Valeur par défaut	Utilisé en mode interactif ?
Destroy	écraser le fichier de données	N	
Indexfile	Ecrire les infos de tables/index dans le fichier indiqué		
Skip_unusable_index	Ignorer la maintenance d'index unitilisables	N	
feedkack	affiche la progression toute les N lignes	0	
toid_novalidate	sauter la validation des id		
filesize	taille max de chaque fichier vidage		
statistics	importer les statistiques	Y	
resumables	suspension en cas d'erreurs d'espace	N	
Compile	Compiler les procédures		
Transport_tablespace	impport des métadonnées des TS	N	
Tablespace	TS à transporter		
Datafile	fichier à transporter		
TTS_OWNER	Utilisateurs propriétaires des données		

Notes: certains paramètres sont mutuellement exclusifs(table, user et full)

7.1.3 Import

Import en mode interactif et commande (suite)

. Import en mode interactif (options complètes)

```
$ imp
  User name :
  password :
  Import file : expdat.dmp >
  Enter array fetch buffer size : 10240 >
  List contents of import file only (Y/N) : N>
  Ignore create errors due to object existence (Y/N) : N>
  Import grants (Y/N) : Y>
  Import table data (Y/N) : Y>
  Import entire export file (Y/N) : Y>
  Si N (non) alors renseigner :
    Username :
    tablename:
```

. Import en mode commande

$ imp [username/passwd] [param1 = value 1] ... [paramN = value N]

Notes : 1) un des paramètres au moins doit être mis explicitement
2) si table, owner ou full ne sont pas mis, il s'agit d'importer vers l'utilisateur qui fait l'export

7.1.3 Import

Import en mode interactif et commande (suite)

Exemple 1 : Import de la table emp de l'utilisateur tintin par l'administrateur

a) en mode commande

```
$ imp system/manager file=tintin.dmp
      fromuser=tintin tables="(emp)"
```

b) en mode interactif

```
$ imp system/manager
Import file : tintin.dmp >
Enter array fetch buffer size : 10240 >
List contents of import file only (Y/N) : N>
Ignore create errors due to object existence (Y/N) : N>
Import grants (Y/N) : Y>
Import table data (Y/N) : Y>
Import entire export file (Y/N) : Y>N
Username :titin
Enter table names. Null list means all tables for user
Enter table name or . if done :emp
Enter table name or . if done : .
```


7.1.3 Import

Import en mode batch

- les paramètres sont passés dans un fichier de paramètres
- permet de s'affranchir de la limite de la taille de la ligne de commande
- permet de faciliter l'automatisation de l'import
- Syntaxe admise dans le fichier des paramètres
  MOT CLE= valeur
  MOT CLE= (valeur)
  MOT CLE=(Valeur 1, Valeur 2, ..., Valeur N)
- Exemple de fichier de paramètres PARIMP.PAR
  FILE=tintin.dmp
  GRANT=Y
  INDEX=y
  FROMUSER=tintin
  tables="(emp)"
- Activation
  $ imp system/manager parfile=parimp.par

7.1.3 Import

Import incrémental, cumulatif ou complet

- **Généralités**
  - stratégie d'import étalée dans le temps
  - valide uniquement si FULL = Y et si l'utilisateur qui le fait a le privilège EXP_FULL_DATABASE
  - permet de raccourcir les temps d'imports
- **La commande d'IMPORT**
  IMP username/passwd
  INCTYPE={SYSTEM|RESTORE}

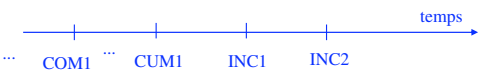
SYSTEM :
importe les objets systèmes les plus récents grâce à l'export incrémental le plus récent. Pas d'import des objets et données des utilisateurs.

RESTORE :
importe tous les objets et données des utilisateurs en appliquant les export incrémentaux dans un ordre chronologie ascendant (du plus vieux export au plus récent)

7.1.3 Import

Import incrémental, cumulatif ou complet (suite)

- **Etapes à suivre**
  - 1. importer les objets system avec le plus récent import INCREMENTAL export ou le récent CUMULATIF export si aucun INCREMENTAL n'existe
  - 2. importer le plus récent export COMPLET
  - 3. importer tous les exports CUMULATIFS après le dernier export COMPLET
  - 4. importer tous les exports INCREMENTAUX après le dernier export CUMULATIF
- **Exemple**




```
$ imp system/manager INCTYPE=SYSTEM FULL=Y FILE=INC2
$ imp system/manager INCTYPE=RESTORE FULL=Y FILE=COM1
$ imp system/manager INCTYPE=RESTORE FULL=Y FILE=CUM1
$ imp system/manager INCTYPE=RESTORE FULL=Y FILE=INC1
$ imp system/manager INCTYPE=RESTORE FULL=Y FILE=INC2
```

7.1.4 Les tablespaces transportables

Généralités

- Depuis la version 8i Oracle **permet de déplacer un Tablespace d'une base vers une autre**
- Cette **méthode de transport** de données **est plus rapide** que le traditionnel Import/Export
- Le **système d'exploitation et le matériel** de la machine source doivent être le même que celui de la machine cible
- les **bases source et cible doivent avoir la même taille de bloc et même CHARACTER SET**
- Avant de déplacer un tablespace, **il doit être mis READONLY** puis **le schéma des données contenus dans le tablespace doit être exporté**.

7.1.4 Les tablespaces transportables

Avantages des tablespaces transportables

- **Copie rapide des informations** entre les bases de production, la base du DatawareHouse et les bases DataMart
- **Archivage rapide** d'information surtout dans un contexte DWH
- **Publication rapide** de données d'un service vers un autre. D'une base vers une autre.

Limitations ; le déplacement d'un tablespace n'est pas possible s'il contient :

- Snapshot/replication
- Function-based indexes
- Scoped REFs
- Domain indexes (un nouveau type d'index créé par l'utilisateur)
- 8.0-compatible advanced queues with multiple recipients

7.1.4 Les tablespaces transportables

Etapes de déplacement d'un Tablespace

- **1. Vérification que le tablespace est transportable.**
  - SQL> EXECUTE sys.dbms_tts.**transport_set_check**('USERS', TRUE);
  - SQL> SELECT * FROM sys.transport_set_violations;
- **2. Création d'un pool de fichiers transportables**.
  - Un pool contient les fichiers du ou des tablespaces et le ou les fichiers contenant les métadonnées
  - Sql>ALTER TABLESPACE tstest READ ONLY;
  - Exporter le schéma des données du ou des tablespaces
  - EXP TRANSPORT_TABLESPACE=y TABLESPACES=(tstest) TRIGGERS= y/n CONSTRAINTS= y/n GRANTS= y/n FILE=tstest.dmp
- **3. Transport du pool de fichiers**
  - Copie des fichiers de données et d'exports du schéma vers la base cible (via les commandes de l'OS)
- **4. Attacher le tablespace à la base cible**
  - faire un import pour attacher les fichiers de données à la base cible. Emplacement identique à la base source.
  - c:>IMP TRANSPORT_TABLESPACE=y DATAFILES=('f:\oracle\oradata\dbtests\tstest2\tstest_1.dbf') TABLESPACES=(tstest) TTS_OWNERS=(sadm) FROMUSER=(sadm) TOUSER=(scott) FILE=tstest.dmp

7.2 Sqlloader

Plan

- Généralités
- Fonctionnement de SQLLOADER
- Le langage de définition de données de SQLLOADER
- Le mode chargement traditionnel et direct
- Les exemples

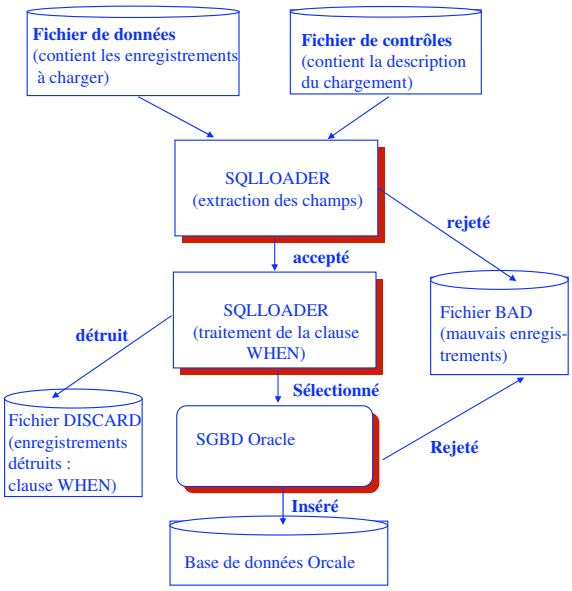
7.2 Sqlloader

Généralités

- permet de charger des données de **format ASCII, EBCDIC, DBASE, EXCEL,** .. dans une base Oracle
- peut traiter **différents formats d'enregistrements** (enregistrements à champs fixes et/ou variables encapsulés ou non avec des délimiteurs spéciaux)
- permet à partir de plusieurs enregistrements physiques de créer une ligne logique dans la base
- permet à partir d'un enregistrement physique de créer plusieurs lignes logiques dans la base
- permet de charger des **données depuis un disque ou une bande**
- permet d'accélérer le chargement de données grâce au **chemin DIRECT**
- Fournit des **rapports précis** de chargement

7.2 Sqlloader

Fonctionnement de SQLLOADER



7.2 Sqlloader

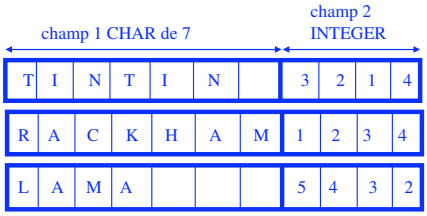
Le fichier de données

- Le fichier de données contient les enregistrements à charger
- Un enregistrement dans le fichier est composé de N champs séparés par des séparateurs (virgule, espace, ...)
- Un champ peut être :
  - numérique ou de type chaîne de caractères
  - fixe ou variable
  - encapsulé entre des côtes ou libre. Exemple "Rackham le Rouge"

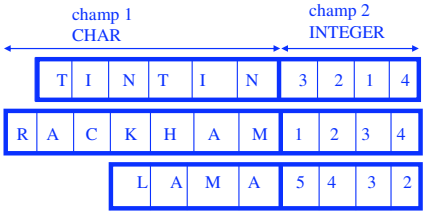
7.2 Sqlloader

Le fichier de données (suite)

Exemple 1: Fichier avec des enregistrements fixes



Exemple 2: Fichier avec des enregistrements variables



7.2 Sqlloader

Le fichier de contrôle

- Ce fichier contient les commandes de description des enregistrements : C'EST LE LDD DE SQLLOADER
- Ce fichier peut contenir aussi les données
- Il est possible de récupérer conditionnellement des enregistrements CLAUSE WHEN
- ***Notes*** : Nous reviendrons plus loins sur le LDD de SQLLOADER

7.2 Sqlloader

Le fichier LOG

- Ce fichier contient les informations suivantes :
  - 1) Informations d'en-tête (version Sqlloader, la date, ...)
  - 2) Les informations générales
    - nom des fichiers utilisés (fichiers de contrôle, de données, bad et discard)
    - la commande lancée avec ces arguments
    - bilan du chargement
  - 3) les informations sur les tables chargées
  - 4) les informations sur le fichier de données (lignes traitées, rejetées, détruites, ...)
  - 5) Résumé des statistiques (espace utilisé, début / fin de chargement, temps écoulé, cpu consommée, ...).

7.2 Sqlloader

Le fichier Discard

- Ce fichier reçoit les enregistrements valides mais ne vérifiant pas la condition derrière la clause WHEN
- son format est celui du fichier de données
- son extension est nomfic.dis

Le fichier bad

- ce fichier reçoit deux types d'enregistrements :
  - ceux non décrits dans le fichier de Contrôle
  - et ceux violant les contraintes de la Base Oracle

7.2 Sqlloader

La commande SQLLOADER

Syntaxe

sqlldr username/passwd [param1=value1] ... [paramN=vauleN]

Tableau des paramètres

Mot clé	description	valeur / défaut
BAD	spéficie le fichier de rejet	nomcontrolfile.BAD
BINDSIZE	taille max de la bind array	os dépendant
CONTROL	nom fichier de contrôle	ctl.if
DATA	nom du fichier ayant les enreg.	nomcontrolefile.dat
DIRECT	mode direct	
DISCARD	Ficher des enreg. détruits	nomcontrolfile.date
DISCARDMAX	Max de lignes rejetable	
ERRORS	nbre d'erreurs d'insertion	
FILE	utile en option parallèle	
LOAD	Nbre max d'enreg. logique charger	tous
LOG	Fichier d'info. sur le chargement	nomcontrolfile.log
PARFILE	fichier de paramètre	aucun
PARALLEL	chargement concurrent	
ROWS	NB Lignes dans le bind array	
SILENT	suppression de messages	
SKIP	Enreg. à sauter depuis le début	
USERID	nom utilisateur + passwd	

La commande SQLLOADER (suite)

Exemple d'activation de sqlloader en mode commande

```
$ sqlldr system/manager CONTROL=utlexple1.ctl
        LOG=utlexple1.log BAD=utlexple1.bad
        SILENT=(HEADER, ERRORS, DISCARDS)
        ERRORS=200
```

Exemple d'activation de sqlloader en mode batch

Les paramètres peuvent être regroupés dans un fichier de paramètres

```
$ sqlldr PARFILE=utlexple1.par
```

Contenu de utlexple1.par

```
userid= system/manager
control=utlexple1.ctl
log=utlexple1.log
bad=utlexple1.bad
silent=(HEADER, ERRORS, DISCARDS)
errors=200
```


Le LDD de SQLLOADER

Syntaxe

```
OPTIONS (options)
LOAD DATA
[{INFILE | INDDN} {nom_fichier | * }
[STREAM | RECORD | FIXED n [BLCKSIZE taille]
   | VARIABLE [m]]
[{BADFILE | BADDN} nom_fichier]
[{DISCARDFILE | DISCARDDN} nom_fichier]
[{DISCARDS | DISCARDMAX} n]
[{INFILE | INDDN} ...]
[APPEND | REPLACE | INSERT]
[RECLEN n]
[{CONCATENATE n | CONTINUEIF
      {[THIS | NEXT] (début [:fin]) | LAST}
      opérateur {'chaine_de_caractère' | X'chaîne_hexa'} }]

INTO TABLE nom_table
   [APPEND | REPLACE | INSERT]
   [WHEN conditions]
   [FIELDS [spécification_délimiteur]]
   (  nom_colonne )
      RECNUM | CONSTANT valeur | SEQUENCE ({n | MAX | COUNT}[,
         incrémentation]) |
      POSITION ({début [: fin] | * [+n]}] spécification_de_type_de_données
         [NULLIF condition] [EFAULT condition] } [, ...] )

[INTO TABLE ...]
[BEGINDATA]
données
```


Le LDD de SQLLOADER (suite)

Mot clé	Description
OPTIONS (options)	positionne des options de sqlloader (load, skip, errors, rows, binsize, silent)
INFILE | INDDN	indique le nom du fichier de données. Si * alors les données sont à la fin du fichier de contrôle (clause BEGIN DATA)
STREAM	Fichier de données = suites de caractères. Newline indique la fin de chaque enregistrement
RECORD	sqlloader utilise le système de gestion de fichiers de l'OS hôte
FIXED	sqlloader utilise un enreg. de taille fixe
APPEND | REPLACE | INSERT	mode d'insertion d'enregistrements dans la base
RECLEN n	longueur max d'un enregistrement logique
CONCATENATE n | CONTINUEIF	construction d'un enreg. logique à partir de plusieurs enreg. physiques
INTO TABLE	description de la table cible

Le LDD de SQLLOADER (suite)

Exemple

Exemple 1 : Fichier de contrôle et de données

```
LOAD DATA
INFILE *
INTO TABLE DEPT
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY "'"
(DEPTNO, DNAME, LOC)
BEGIN DATA
12,RESEARCH,"SARATOGA"
10,"ACCOUNTING",CLEVELAND
11,"ART",SALEM
13,FINANCE,"BOSTON"
21,"SALES",PHILA.
22,"SALES",ROCHESTER
42,"INT'L","SAN FRAN"
```

Le LDD de SQLLOADER (suite)

Exemple (suite)

Exemple 2 :

Fichier de contrôle : ulcase2.ctl

```
LOAD DATA
INFILE 'ulcase2.dat'
INTO TABLE EMP
( EMPNO      POSITION(01:04) INTEGER EXTERNAL,
ENAME        POSITION(06:15) CHAR,
JOB          POSITION(17:25) CHAR,
MGR          POSITION(27:30) INTEGER EXTERNAL,
SAL          POSITION(32:39) DECIMAL EXTERNAL,
COMM         POSITION(41:48) DECIMAL EXTERNAL,
DEPTNO   POSITION(50:51) INTEGER EXTERNAL)
```

Fichier de données : ulcase2.dat

```
7782 CLARK      MANAGER    7839   2572.50              10
7839 KING       PRESIDENT          5500.00              10
7934 MILLER     CLERK      7782    920.00              10
7566 JONES      MANAGER    7839   3123.75              20
7499 ALLEN      SALESMAN   7698   1600.00    300.00    30
7654 MARTIN     SALESMAN   7698   1312.50   1400.00    30
7658 CHAN       ANALYST    7566   3450.00              20
```


Le LDD de SQLLOADER (suite)

Exemple (suite)

Exemple 3 :

Fichier de contrôle : ulcase3.ctl

```
-- Variable length, delimited and enclosed data format
LOAD DATA
INFILE *
APPEND
INTO TABLE EMP
FIELDS TERMINATED BY "," OPTIONALLY ENCLOSED BY '"'
(empno, ename, job, mgr,
hiredate DATE(20) "DD-Month-YYYY",
sal, comm, deptno   CHAR TERMINATED BY ':',
projno, loadseq  SEQUENCE(MAX,1) )

BEGINDATA
7782, "Clark", "Manager", 7839, 09-June-1981, 2572.50,, 10:101
7839, "King", "President", , 17-November-1981, 5500.00,, 10:102
7934, "Miller", "Clerk", 7782, 23-January-1982, 920.00,, 10:102
7566, "Jones", "Manager", 7839, 02-April-1981, 3123.75,, 20:101
7499, "Allen", "Salesman", 7698, 20-February-1981, 1600.00, 300.00,
        30:103
7654, "Martin", "Salesman", 7698, 28-September-1981, 1312.50, 1400.00,
        30:103
7658, "Chan", "Analyst", 7566, 03-May-1982, 3450,, 20:101
```


Le LDD de SQLLOADER (suite)

Exemple (suite)

Exemple 4 :

Fichier de contrôle : ulcase4.ctl

```
LOAD DATA
INFILE "ulcase4.dat"
DISCARDFILE "ulcase4.dis"
DISCARDMAX 999
REPLACE
CONTINUEIF (1) = '*'
INTO TABLE EMP
(  EMPNO    POSITION(01:04) INTEGER EXTERNAL,
   ENAME    POSITION(06:15) CHAR,
   JOB      POSITION(17:25) CHAR,
   MGR      POSITION(27:30) INTEGER EXTERNAL,
   SAL      POSITION(32:39) DECIMAL EXTERNAL
   COMM     POSITION(41:48) DECIMAL EXTERNAL,
   DEPTNO   POSITION(50:51) INTEGER EXTERNAL,
   HIREDATE POSITION(52:60) INTEGER EXTERNAL)
```

Fichier de données: utlcase4.dat

```
*7782 CLARK      MANAGER   7839  2572.50   -10    2512-NOV-85
*7839 KING       PRESIDENT       5500.00          2505-APR-83
*7934 MILLER     CLERK     7782   920.00          2508-MAY-80
```


Le LDD de SQLLOADER (suite)

Exemple (suite)

Exemple 5:

Fichier de contrôle : ulcase5.ctl

```
LOAD DATA
INFILE 'ulcase5.dat'
BADFILE 'ulcase5.bad'
DISCARDFILE 'ulcase5.dis'
REPLACE
INTO TABLE EMP
   (EMPNO    POSITION(1:4)   INTEGER EXTERNAL,
   ENAME    POSITION(6:15)  CHAR,
   DEPTNO   POSITION(17:18) CHAR,
   MGR      POSITION(20:23) INTEGER EXTERNAL)

INTO TABLE PROJ
-- PROJ has two columns, both not null: EMPNO and PROJNO
WHEN PROJNO != '  '
(EMPNO    POSITION(1:4)   INTEGER EXTERNAL,
PROJNO   POSITION(25:27) INTEGER EXTERNAL)   -- 1er projet

INTO TABLE PROJ
WHEN PROJNO != '  '
(EMPNO    POSITION(1:4)   INTEGER EXTERNAL,
PROJNO   POSITION(29:31) INTEGER EXTERNAL)   -- 2 ème projet

INTO TABLE PROJ
WHEN PROJNO != '  '
(EMPNO    POSITION(1:4)   INTEGER EXTERNAL,
PROJNO   POSITION(33:35) INTEGER EXTERNAL)   -- 3 ème projet
```

Le LDD de SQLLOADER (suite)

Exemple (suite)

Exemple 5 (suite) :

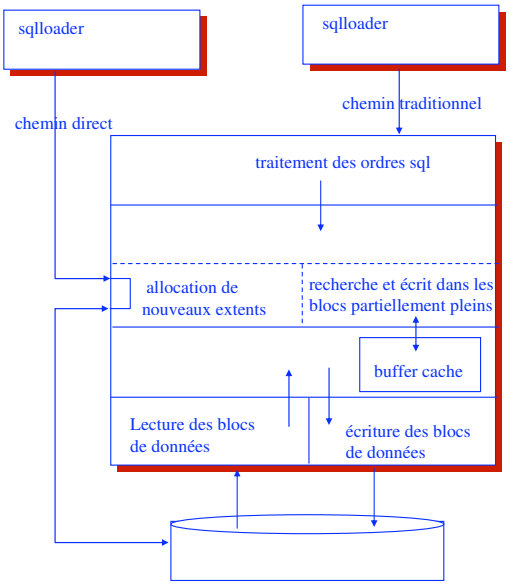
Fichier de données: utlcase5.dat

1234	BAKER	10	9999	101	102	103
1234	JOKER	10	9999	777	888	999
2664	YOUNG	20	2893	425	abc	102
5321	OTOOLE	10	9999	321	55	40
2134	FARMER	20	4555	236	456	
2414	LITTLE	20	5634	236	456	40
6542	LEE	10	4532	102	321	14
2849	EDDS	xx	4555	294	40	
4532	PERKINS	10	9999	40		
1244	HUNT	11	3452	665	133	456
123	DOOLITTLE	12	9940	132		
1453	MACDONALD	25	5532	200		

Sqlloader chemin traditionnel et direct

- deux modes de chargement de données : le chemin tradionnel et le chemin direct
- le chemin traditionnel :
  - utilise la commande INSERT
  - vérifient toutes les contraintes d'intégrité
- le chemin direct
  - courcircuite le moteur SQL
  - accélère les chargements dans certains cas.

Sqlloader chemin traditionnel et direct (suite)



Sqlloader chemin traditionnel et direct (suite)

- **Quand utiliser le chemin traditionnel ?**
  - quand il y a **accès concurrent** à une table indexée ou modifiée
  - quand il y a **chargement de données via sqlnet** (problème de jeu de caractères)
  - en cas de **chargement de données dans une table en cluster**
  - si le **nombre de lignes à charger dans une table indexée** (index grand) **est faible**
  - si l'on veut **appliquer des fonctions SQL**
  - si l'on veut **vérifier les contraintes d'intégrité** au chargement

7.2 Sqlloader

Sqlloader chemin traditionnel et direct (suite)

- **Quand utiliser le chemin Direct ?**
  - en cas de besoin de chargement rapide d'une grande quantité de données
  - en cas de besoin de parallélisation du chargement de données
  - en cas de besoin de chargement des données dans une police de caractères non supportées pendant la session
- **Activation du chemin direct**

  $ sqlldr system/manager DIRECT=Y ...

7.2 Sqlloader

Sqlloader chemin traditionnel et direct (suite)

- **Avantages du chemin direct**
  - pas d'utilisation de blocs partiellement pleins
  - pas de génération d'Ordre SQL
  - pas d'accès au cache d'Oracle
  - parallélisation des I/O possible
  - tri rapide
  - pas d'écriture dans les Redo log en mode sans archive
- **contraintes**
  - impossible de charger dans des tables en cluster
  - pas de condition SQL dans le fichier de contrôle
  - pas d'Ordre Select sur une table indexée

7.3 Sqlplus

Généralités

- Outil d'administration Graphique
- supporte les principaux GUI du marché
- Serveur manager permet :
  - d'effectuer les tâches traditionnelles d'administration
  - de travailler en mode concurrent (N fenêtres peuvent être ouvertes)
  - d'administrer une ou plusieurs bases
  - de centraliser l'administration de BD distribuées
  - d'exécuter dynamiquement des Ordre SQL et PL/SQL
- Serveur manager fonctionne en mode *graphique* (srvmgr) ou ligne (srvmgrl)

7.5 Sqlplus

Commandes de Sqlplus (sql)

Commande	Description
ARCHIVELOG	Démarrage, arrêt ou affichage des informations sur l'archivage des Redo Log
CONNECT	Connexion à une base
DESCRIBE	Description d'une table, d'une vue, d'une procédure
DISCONNECT	déconnexion à une base
EXECUTE	Exécution d'une commande PL/SQL limitée à une ligne
EXIT	Quitter Sqlplus
HELP	Afficher les informations d'aide
HOST	Exécuter une commande de l'OS sans quitter Sqlplus
MONITOR	Observer les statistiques sur : FILE, PROCESS, I/O, LATCHES, LOCKS, ROLLBACK, SESSIONS, STATISTICS, TABLE (non valable en mode ligne)
PRINT	affichage du contenu d'une variable
RECOVER	restauration d'un fichier, d'un Tablespace ou d'un Base de données
REMARK	Indique une ligne de commentaire
SET	Modification des paramètres de la session courante de sql
SHOW	Affichage des paramètres de la session courante de sql
SHUTDOWN	Arrêt de l'instance courante
SPOOL	activation/désactivation de la rédirection des résultats vers un fichier
STARTUP	démarrage d'une base base (MOUNT , OPEN)
VARIABLE	définition d'une variable utilisable dans un bloc PL/SQL ou lors d'un EXECUTE ou PRINT

7.4 Les NLS

Plan

- Généralités
- Activation des NLS au niveau de l'OS
- Activation des NLS au niveau d'une instance
- Modification des NLS au niveau d'une session
- Modification des NLS au niveau d'une fonction SQL

7.4 Les NLS

Généralités

- Les NLS permettent de mettre en oeuvre une ou plusieurs applications dans des langues différentes
- L'entreprise et le marché se mondialisent
- certains pays pratiquent plusieurs langues
- les NLS permettent de prendre en compte les particularités d'un pays (monnaie, date, ...)
- les NLS permettent d'avoir les messages Oracle dans la langue de son choix
- sqlnet assure la conversion des jeux de caractères entre le poste client et la base de données si les jeux sont différents
- le posistionnement d'un jeu de caractères peut se faire : au niveau OS, instance, session ou fonction SQL

7.4 Les NLS

Activation des NLS au niveau de l'OS

- le paramètre d'environnement NLS_LANG permet de positionner (pour une session) :
  - la langue, le territoire et le jeu de caractères

Positionnement du paramètre NLS_LANG

NLS_LANG=langue_pays.jeux de caractères

Exemple

NLS_LANG=AMERICAN_AMERICA.US7ASSCII
NLS_LANG=FRENCH_FRANCE.ISO8859P1
NLS_LANG=FRENCH_CANADA.WE8DEC
NLS_LANG=JAPANESE_JAPAN.JA16EUC

Notes

- positionnement dépendant de l'environnement (NLS_LANG est une variable d'environnement)
- NLS_LANG efface et remplace les positionnements faits avec NLS_LANGUAGE et NLS_TERRITORY
- un ALTER SESSION est exécuté à la connexion

7.4 Les NLS

Activation des NLS au niveau d'une instance

- Les paramètres d'instance NLS_LANGUAGE et NLS_TERRITORY permettent de fixer les propriétés d'une langue et d'un pays
- Ce sont les valeurs par défaut pour toutes les sessions
- Le paramètre **NLS_LANGUAGE** fixe :
  - la langue utilisée pour les messages du serveur
  - la langue utilisée pour les noms des jours, mois et leurs abréviations
  - la séquence de tri par défaut utile dans les commandes SQL tels que *Order by, Group by*, ...

Exemple

NLS_LANGUAGE=FRENCH

Notes: a) la valeur de ce paramètre est fixée dans init.ora
b) la valeur par défaut est AMERICAN

Activation des NLS au niveau d'une instance (suite)

- Le paramètre d'initialisation **NLS_TERRITORY** (à définir dans init.ora) permet de définir :
  - le format de la date
  - le caractère décimal et le séparateur de milliers
  - le symbole monétaire local
  - le symbole monétaire ISO du pays
  - le 1er jour de la semaine.

Exemple

NLS_LANGUAGE=FRANCE

format date	: JJ/MM/AA
caractère décimal	: ,
séparateur de millier	: .

Notes: a) la valeur de ce paramètre est fixée dans init.ora
b) la valeur par défaut est AMERICA

Activation des NLS au niveau d'une instance (suite)

- **Les paramètres NLS_LANGUAGE et NLS_TERRITORY *fixe implicitement*** le format de la date, la langue, les noms de mois, jours, ..., l'ordre de tri, ...
- **Les paramètres suivants** (à fixer dans *init.ora*) :
  - NLS_DATE_FORMAT
  - NLS_DATE_LANGUAGE
  - NLS_NUMERIC_CHARACTERS
  - NLS_CURRENCY
  - NLS_ISO_CURRENCY
  - NLS_SORT

  permettent d'agir explicitement.

Exemple

```
NLS_DATE_FORMAT='DD RM YY'
NLS_DATE_LANGUAGE=FRENCH
NLS_ISO_CUYRRENCY=FRANCE
NLS_NUMERIC_CHARACTERS =',.'
...
```


Format des dates et des Nombres

En général

Les formats des dates et des nombres sont fournis avec les fonctions : **to_date, to_char, to_number** de SQL.

Format additionnel de nombres

D (Décimal)	rend le caractère décimal
G (Groupe)	rend le séparateur de milliers
L(monnaie locale)	rend le symbole de monnaie local
C(monnaie ISO)	rend le symbole de monnaie ISO

Exemple

```
NLS_ISO_CUYRRENCY=FRANCE
NLS_NUMERIC_CHARACTERS =',.'   -- 'DG'
NLS_ISO_CUYRRENCY='FF '

'1234' '9G999'   => 1.234

'3,234.44' 'L9G999D99'      => FF3.234,44
```


Activation des NLS au niveau d'une session

- Les **paramètres d'initialisation** :
  - NLS_LANGUAGE
  - NLS_TERRITORY
  - NLS_DATE_FORMAT
  - NLS_DATE_LANGUAGE
  - NLS_NUMERIC_CHARACTERS
  - NLS_CURRENCY
  - NLS_ISO_CURRENCY
  - NLS_SORT

  peuvent être **modifiés à la volée** pour une session en cours grâce à la commande **ALTER SESSION**

Exemple

```
ALTER SESSION SET
    NLS_LANGUAGE=FRENCH
    NLS_TERRITORY=FRANCE
    NLS_DATE_FORMAT='DD RM YY'
    NLS_DATE_LANGUAGE=FRENCH
    NLS_NUMERIC_CHARACTERS=',.'
    NLS_CURRENCY='FF '
    NLS_ISO_CURRENCY=FRANCE ;
```

7.4 Les NLS

Activation des NLS au niveau d'une session (suite)

Exemple d'ordre SQL exploitant les NLS

```
sql> SELECT to_char(sysdate) date_du_jour,
            to_char(sysdate, 'Day : Dd Month YYYY')  encore_date_jour,
            to_number('1.234', '9G999')) "un nombre",
            to_number('12,66673.49', 'L099G999D99') "Total local"),
             to_number('12,66673.49', 'C099G999D99') "Total ISO")
     FROM DUAL ;
```


7.4 Les NLS

Modification des NLS au niveau d'une fonction SQL

- **Les paramètres pouvant être modifiés au niveau d'une fonction SQL** pour une requête sont :
  - NLS_DATE_FORMAT
  - NLS_DATE_LANGUAGE
  - NLS_NUMERIC_CHARACTERS
  - NLS_CURRENCY
  - NLS_ISO_CURRENCY
  - NLS_SORT
- Les paramètres NLS_LANGUAGE et NLS_TERRITORY ne peuvent être modifiés au niveau des fonctions
- **Les fonctions SQL pouvant modifier** les paramètres sont : ***to_char, to_date, to_number, nls_upper, nls_lower, nls_initcap, nlssort***.

7.4 Les NLS

Modification des NLS au niveau d'une fonction SQL(suite)

Tableau des paramètres et les fonctions pouvant agir dessus

TO_DATE :	NLS_DATE_LANGUAGE
TO_NUMBER :	NLS_NUMERIC_CHARACTERS, NLS_CURRENCY, NLS_ISO_CURRENCY
TO_CHAR :	NLS_DATE_LANGUAGE NLS_NUMERIC_CHARACTERS, NLS_CURRENCY, NLS_ISO_CURRENCY
NLS_UPPER :	NLS_SORT
NLS_LOWER :	NLS_SORT
NLS_INITCAP :	NLS_SORT
NLSSORT :	NLS_SORT

7.4 Les NLS

Modification des NLS au niveau d'une fonction SQL(suite)

Exemple

```
SELECT to_number('13.000,00', '99G999D99',
                    'nls_numeric_characters=",."')
FROM dual;

SELECT to_date('1-MAY-89', 'DD-MON-YY',
          'nls_date_language=FRENCH')
          FROM dual;

SELECT to_char(hiredate, 'DD/MON/YYYY',
          'nls_date_language=FRENCH')
          FROM EMP ;

SELECT ename FROM emp
          ORDER BY NLSSORT(ename, 'nls_sort=GERMAN');

SELECT ename  FROM emp
          ORDER BY nls_lower(ename, 'nls_sort=GERMAN');
```

7.5. Oracle Entreprise Manager(OEM)

- **OEM**
  - Outil d'administration graphique d'oracle
  - Voir Annexe

8. L'option procédurale

- **PLAN**
  - 8.1 Généralités
  - 8.2 Les procédures et fonctions
  - 8.3 Les packages
  - 8.4 Les Triggers

8.1 Généralités

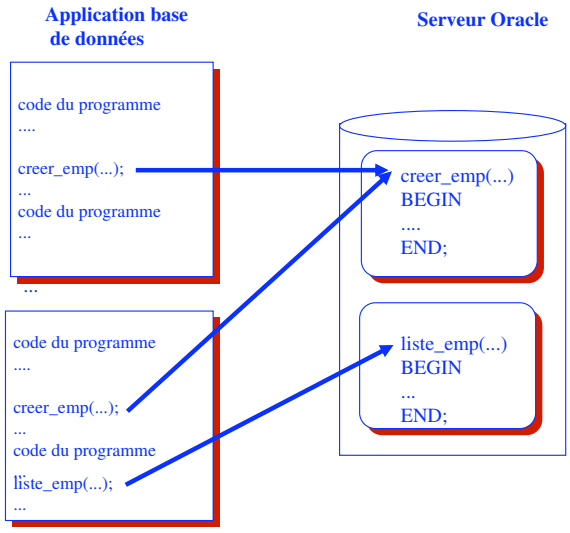
- **L'option procédurale se compose de :** procédures, fonctions, packages et triggers
- **permet à Oracle de stocker des traitements** (*code PL*SQL uniquement*) au niveau du moteur base de données
- **Elle est maintenant *intégré* dans le noyau Oracle**
- **C/S 2ème Génération :** Jusqu'à 40 % des traitements transposables du ***coté serveur***
- **Activation de l'option procédurale**
  - Exécuter le script *catproc.sql* étant SYS
    sql>@@$ORACLE_HOME/rdbms/admin/catproc

8.2 Procédures et fonctions

- **Définition**
  - Unité de traitement PL/SQL pouvant contenir des Ordres SQL, des variables, des constantes, des curseurs et des exceptions
- **Une fonction à la différence d'une procédure rend une valeur**
- **une procédure ou une fonction peut être :**
  - exécutée en interactif sous SQL*PLUS ou SQL*DBA
  - appelée depuis une application
  - appelée depuis une autre procédure ou fonction
- **une procédure ou une fonction comporte deux parties créées en même temps :**
  - la partie spécification (l'entête)
  - la partie implémentation (le corps)

8.2 Procédures et fonctions

Appel de procédures et fonctions depuis un applicatif



8.2 Procédures et fonctions

Avantages des procédures et fonctions stockées dans la base

- *plus de sécurité* : il n'est plus utile de donner des droits aux utilisateurs sur les objets de base manipulés par les procédures (tables, vues, ...)
- *plus de performance* :
  - échanges réduits entre l'application et la base
  - une compilation pour plusieurs exécutions
  - moins d'E/S, procédure déjà dans la zone des requêtes
- *Economie d'espace mémoire* : une seule copie en SGA mais des exécutions par plusieurs applications possibles
- *une plus grande productivité* :
  - évolutivité : la modification du corps d'une procédure n'implique pas celle des applications qui l'appelle
  - centralisation des modifications

8.2 Procédures et fonctions

Création de procédures et de fonctions

- privilège requis : CREATE PROCEDURE ou CREATE ANY PROCEDURE

Syntaxe de création d'une procédure

```
CREATE OR REPLACE PROCEDURE [schema.]procedure
  [ (<liste d'arguments>)]
  {is | as}
  BEGIN    -- bloc PLSQL
    -- corps de la procédure
  END;
  external_body; -- procedure externe
```

Syntaxe de création d'une fonction

```
CREATE OR REPLACE FUNCTION [schema.]fonction
  [ (<liste d'arguments>)] RETURN type
  {is | as}
  BEGIN    -- bloc PLSQL
    -- corps de la fonction
  END;
  external_body; -- procedure externe
```

Liste d'arguments ::= nom argument {IN | OUT | IN OUT} type;

8.2 Procédures et fonctions

Création de procédures et de fonctions (suite)

Exemple de création de procédure

```
CREATE OR REPLACE PROCEDURE nouveau_sal (empid IN
    NUMBER, taux IN NUMBER) IS
  BEGIN
    UPDATE emp  SET sal = sal* (1 + taux)  WHERE empno = empid ;
    EXCEPTION
      WHEN NO_DATA_FOUND THEN
          DBMS_OUTPUT.PUT_LINE('Employe inexistant');
          RAISE_APPLICATION_ERROR(-20012, 'Employe Nr. ' ||
              to_char(empid) || 'Inexistant');
  END;
```

Exemple de création d'une fonction

```
CREATE OR REPLACE FUNCTION emp_info(empid IN number) return
emp%ROWTYPE AS
   emprow emp%rowtype ;
 BEGIN
  SELECT empno, ename, job, mgr, hiredate, sal, comm, deptno
     INTO emprow FROM emp
       WHERE empno = empid;
  RETURN (emprow);
END;
```

Note : sous sqlplus ou sql la dernière ligne de la procédure doit être /

8.2 Procédures et fonctions

Modification et récompilation de procédures et fonctions

- **récompilation** :En cas d'évolution des objets du schéma (tables, ...) manipulés dans une procédure ou une fonction
- **privilège requis** : ALTER PROCEDURE ou ALTER ANY PROCEDURE
- **Syntaxe**

  ALTER { FUNCTION | PROCEDURE }
  [schema.]nom COMPILE ;

Mots clés ou paramètres	Description
nom	nom de la procédure ou fonction
schema	propriétaire de l'objet
COMPILE	récompiler l'objet en cas d'évolution des objets référencés

- **Exemple**
  ALTER FUNCTION emp_info COMPILE;

8.2 Procédures et fonctions

Suppression d'une procédure ou d'une fonction

- privilège requis DROP PROCEDURE ou DROP ANY PROCEDURE
- **Syntaxe**

  DROP FUNCTION | PROCEDURE
  [schema.]nomproc;

- **Exemple**

  DROP PROCEDURE tintin.nouveau_sal ;

8.3 Packages

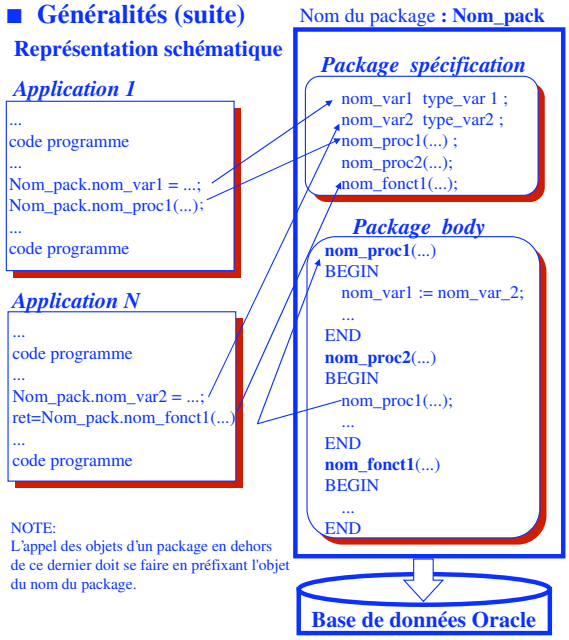
Généralités

- **Définition**
  Un package Oracle est à l'image du package ADA une unité de traitement PL/SQL nommée, regroupant des procédures, des fonctions avec des curseurs et des variables qu'elles utilisent ensemble
- **Un package comporte deux parties** : la partie spécification (PACKAGE SPECIFICATION) et la partie implémentation (PACKAGE BODY)
- Les composants d'un package peuvent être **publics** ou **privés** : Les composants publics sont déclarés au niveau de la partie spécification et les composants privés au niveau la partie implémentation

8.3 Packages

Généralités (suite)

Représentation schématique



NOTE:
L'appel des objets d'un package en dehors de ce dernier doit se faire en préfixant l'objet du nom du package.

Création d'un Package

- La création d'un package consiste à créer la spécification puis le corps du package

Syntaxe de création de la partie spécification

CREATE [OR REPLACE] PACKAGE [schéma.]nom_package
{IS | AS} spéficification PL/SQL

spécification PL/SQL ::=
déclaration de variable |
déclaration d'enregistrement |
déclaration d'execption |
déclaration de table PL/SQL |
déclaration de fonction |
déclaration de procédure ...

Mots clés ou paramètres	Description
schéma	Nom du schéma auquel le package appartient
nom_package	Nom du package dans le schéma
spécification PL/SQL	déclaration de variables, fonctions, procédures, ... globales

Création d'un package (suite)

Syntaxe de création de la partie implémentation

CREATE [OR REPLACE]
PACKAGE BODY [schéma.]nom_package
{IS | AS} corps PL/SQL

corps PL/SQL ::=
déclaration de variable |
déclaration d'enregistrement |
déclaration d'execption |
déclaration de table PL/SQL |
corps de fonction |
corps de curseur |
corps de procédure ...

Mot clé ou paramètre	Description
schéma	Nom du schéma auquel le package appartient
nom_package	Nom du package dans le schéma
corps PL/SQL	déclaration de variables, ..., corps fonctions, des procédures, ...

Note : Les noms utilisés dans la partie spécification doivent être les mêmes que dans la partie implémentation.

Création d'un package (suite)

Exemples de création de la partie spécification

```
CREATE OR REPLACE      PACKAGE tintin.gestion_employes
IS
  PROCEDURE pnouveau_sal (empid IN NUMBER, taux IN NUMBER);
  FUNCTION pemp_info(empid IN number) RETURN emp%ROWTYPE;

END gestion_employes ;
```


Création d'un package (suite)

Exemple de création de procédure

```
CREATE OR REPLACE
      PACKAGE BODY tintin.gestion_employes
        IS

 PROCEDURE pnouveau_sal (empid IN
      NUMBER, taux IN NUMBER) IS
  BEGIN
      UPDATE emp  SET sal = sal* (1 + taux)  WHERE empno = empid ;
        EXCEPTION
          WHEN NO_DATA_FOUND THEN
            DBMS_OUTPUT.PUT_LINE('Employe inexistant');
              RAISE_APPLICATION_ERROR(-20012, 'Employe Nr. ' ||
                  to_char(empid) || 'Inexistant');

  END pnouveau_sal;

  FUNCTION pemp_info(empid IN number)
      RETURN emp%ROWTYPE AS
       emprow emp%rowtype ;
      BEGIN
        SELECT empno, ename, job, mgr, hiredate, sal, comm, deptno
          INTO emprow FROM emp
            WHERE empno = empid;
        RETURN (emprow);
      END pemp_info             ;

END gestion_employes;
```

8.3 Packages

Avantages des packages

- Encapsulation et modularité : le package se comporte comme une boîte noire
- Possibilité de cacher des informations
- Séparation de la spécification des implémentations permet d'augmenter la productivité des développeurs
- + tous les avantages énumérés au niveau des fonctions et procédures (performance, partage, récompilation intelligente, ...)

8.3 Packages

Exécution d'une procédure

- **Etapes**
  - 1. Vérifier les droits de l'utilisateur (privilège objet EXECUTE)
  - 2. Vérifier la validité de la procédure
    - un objet référencé a - t - il changé (table, vue, procédure , ...) ?
    - un privilège sur le package a - t - il été retiré ?
    - un privilège sur un objet référencé a - t - il été révoqué ?
  - 3. La procédure à exécuter réside - t -elle déjà en SGA? sinon la charger et récompiler
  - 4. Exécuter la procédure.
- **Exemple en interactif**

```
sql > EXECUTE
          gestion_employes.pemp_info(7856);
sqlplus>EXECUTE nouveau_sal(7856, 0.5);
```


8.3 Packages

Informations sur les packages et les fonctions

Listes des vues utiles

. ALL_ERRORS, USER_ERRORS, DBA_ERRORS
. ALL_SOURCE, USER_SOURCE , DBA_SOURCE
. USER_OBJECT_SIZE, DBA_OBJECT_SIZE

Exemple 1 : Vue USER_SOURCE

Visualisation du code source et du nombre de lignes de la procédure de l'utilisateur courant

```
sql> SELECT line, text FROM user_source
     WHERE name = 'EMP_INFO';
```


8.3 Packages

Informations sur les packages et les fonctions (suite)

Exemple 2 : Vue USER_ERRORS

Visualisation des erreurs de compilation de l'utilisateur courant

```
sql>  SELECT name, type, line, position, text
      FROM user_errors
      WHERE name = 'EMP_INFO';
```

Exemple 3 : vue USER_OBJECT_SIZE

Espace consommé par la procédure EMP_INFO

```
sql>  SELECT name, sorce_size+parsed_size+ code_size +
          error_size "Taille totale" FROM user_object_size
      WHERE name = 'EMP_INFO';
```

8.3 Les packages

Packages fournis par Oracle et utilisables à la place de certains ordres SQL

Package dbms_session : gestion des infos. de session

Procédures : close_database_link, reset_package, set_label, set_nls, set_mls_label_format, set_role, set_sql_trace, unique_session_id, is_role_enabled, set_close_cached_open_cursors, free_unused_user_memory

Package dbms_ddl : Analyse et compilation des objets du schéma

Procédures : alter_compile, analyze_object

Package dbms_transaction : Gestion des transactions

Procédures : advise_commit, advise_rollback, advise_nothing, commit, commit_comment, commit_force, read_only, read_write, rollback, rollback_force, rollback_savepoint, savepoint, use_rollback_segment, purge_mixed, begin_discrete_transaction, local_transaction_id, step_id

Package dbms_utility : Ensemble de fonctions utilitaires

Procédures : compile_schema, analyze_schema, format_error_stack, format_call_stack, is_parallel_server, get_time, name_resolve

8.3 Les packages

Packages additionnels fournis par Oracle

Nom du package	Description	Procédures
dbms_alert	Gestion asynchrone des alertes des events de la BD	register, remove, signal, waitany, waitone, set_defaults
dbms_describe	permet de décrire les arguments des procédures stockées	describe_procedure
dbms_job	permet de gérer la soumission des job	submit, remove, change, what, next_date, sys as sysdba, broken, run
dbms_lock	Permet d'utiliser les mécanismes de verrouillage	allocate_unique, request, convert, release, sleep
dbms_output	Permet d'utiliser les mécanismes de verrouillage	allocate_unique, request, convert, release, sleep
dbms_pipe	permet la communication inter-session dans une même instance	create_pipe, pack_message, send_message, receive_message, next_item_type, unpack_message remove_pipe, purge, reset_buffer, unique_session_name
dbms_shared_pool	permet de conserver des objets dans shared pool area	sizes, keep, unkeep, aborted_request_threshold
dbms_application_info	permet de gérer les infos d'une application à des fins de performance ou d'audit	set_module, set_action, set_client_info, read_module read_client_info
dbms_system	permet d'activer des utilitaires systèmes tel tracer des requêtes	set_sql_trace_in_session

8.3 Les packages

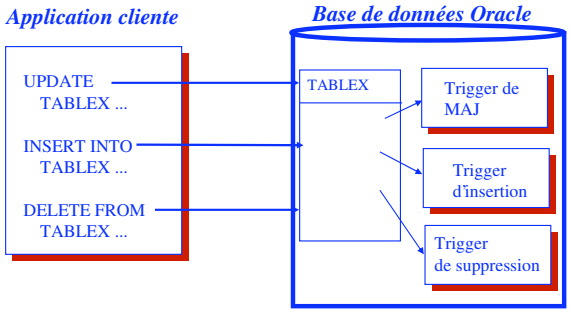
Packages additionnels fournis par Oracle (suite)

Nom du package	Description	Procédures
dbms_space	fourni les infos sur l'espace des segments non accessibles via les vues	
dbms_sql	permet d'écrire des procédures stockées et des blocs PL/SQL anonymes en utilisant le SQL dynamique	open_cursor, parse, execute, bind_variable, define_column, define_column_long, is_open, execute_and_fetch, fetch_rows column_value, variable_value, column_value_long, last_error_position, close_cursor, last_row_count, last_row_id, last_sql_function_code
dbms_refresh	permet de gérer des groupes de snapshot	(voir dbmssnap.sql)
dbms_snapshot	permet de rafraichir des snapshot non membre d'un groupe	(voir dbmssnap.sql)
dbms_defer, dbms_defer_sys, dbms_defer_query	permet construire et administrer des appels de procédures distantes	(voir dbmdefr.sql)
dbms_repcat	permet d'utilier les mécanismes de réplication statique	(voir dbmsrepc.sql)
dbms_repcat_auth, dbms_repcat_admin	permet de créer des utilisateurs ayant le privilège de répliquer	(voir dbmsrepc.sql)

8.4 Les triggers Base de données

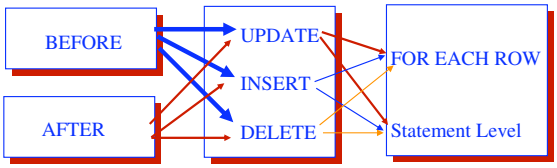
Généralités

- Un trigger BD est une action qui se déclenche avant ou après un événement.
- Les triggers BD Oracle sont des blocs PL/SQL stockés dans la base pouvant se déclencher automatiquement avant ou après l'exécution d'un Ordre SQL d'insertion, modification ou suppression sur une table.
- **Représentation schématique**



Types de triggers BD

- Le type d'un trigger dépend de trois paramètres :
  - l'ordre sur lequel le trigger est défini (UPDATE, INSERT, DELETE)
  - la fréquence de déclenchement du trigger :
    - pour chaque ligne traitée (FOR EACH ROW)
    - pour toutes les lignes
  - le moment de son déclenchement (avant le début des traitements PRE-CONDITION ou à la fin des traitements POST-CONDITION)
    - BEFORE
    - AFTER



Note : Un déclenchement conditionnel(clause WHEN) est possible.
La clause WHEN ne peut être posée qu'en évaluation ligne par ligne

Création d'un trigger

. Privilège requis : CREATE TRIGGER

Syntaxe de création d'un trigger

```
CREATE [ OR REPLACE ] TRIGGER [ schema.] trigger
  { BEFORE | AFTER }
  { DELETE | INSERT | UPDATE [ OF column [, column ] ...] }
  [ OR { DELETE | INSERT | UPDATE [ OF column [, column ] ...] } ] ...
  ON [ schema.] table
  [ [ REFERENCING { OLD [ AS ] old [ NEW [ AS ] new ]
    | NEW [ AS ] new [ OLD [ AS ] old ] } ]
  FOR EACH ROW [ WHEN ( condition ) ] ]
  pl / sql_block
```

Mots clés ou paramètres	Description
OR REPLACE	supprime le contenu du trigger et le recréé
schéma	Nom du propriétaire
trigger	Nom du trigger
BEFORE	déclencher d'abord le trigger puis exécuter l'ordre
AFTER	exécuter d'abord l'ordre puis déclencher le trigger
Delete, update, insert	déclencher le trigger respectivement à la suppression, modification ou insertion.
on schema.table	table concernée
FOR EACH ROW	déclenche le trigger pour chaque ligne traitée
WHEN	déclenche pour les lignes ayant satisfait la condition
PL/SQL BLOCK	bloc de code PL/SQL pour le trigger

Création d'un trigger (suite)

Limite des triggers

- **1 trigger par Type par Table** (12 possibilités).
  Exemple (idem pour INSERT et DELETE):
  BEFORE UPDATE FOR EACH ROW
  BEFORE UPDATE --statement_level
  AFTER UPDATE FOR EACH ROW
  AFTER UPDATE -- statement_level
- **pas de commande DDL** dans le corps du trigger (pas de CREATE TABLE ...)
- **pas de gestion de transaction** (COMMIT, ROLLBACK, SAVEPOINT) dans le corps du trigger même pas à travers une procédure appelée
- **pas de déclaration de variables de type LONG ou LONG ROW**
- **Ordre d'évaluation des lignes pas garantie** (ne pas faire dépendre le déclenchement d'un trigger de l'ordre d'évaluation)

Création d'un trigger BD(suite)

Exemples 1 :

Création d'un trigger qui permet d'assurer la contrainte d'intégrité de mise à jour et de suppression d'une clé dans la table maître et d'assigner la clé étrangère à NULL dans la table ayant la clé étrangère.

<UPDATE - DELETE - SET NULL>

```
CREATE TRIGGER updateset
  AFTER DELETE OR UPDATE OF deptno ON dept
  FOR EACH ROW
  -- Avant de supprimer une ligne dans la table dept ou modifier la clé dans
  -- cette table. Mettre les clés étrangères à NULL
  BEGIN
    IF UPDATING AND :OLD.deptno != :NEW.deptno
      OR DELETING THEN
        UPDATE emp SET emp.deptno = NULL
          WHERE emp.deptno = :old.deptno ;
    END IF ;
  END;
  /
```

8.4 Les triggers Base de données

■ Création d'un trigger BD(suite)

Exemples 2

Création d'un trigger qui permet (lors d'une suppression, modification ou insertion) de mettre à jour automatiquement le montant de la commande pour chaque ligne de commande enregistrée.

```
CREATE TRIGGER modif_commande
   AFTER DELETE OR UPDATE OR INSERT ON item FOR EACH ROW
   DECLARE

BEGIN
   IF DELETING OR UPDATING OR INSERTING THEN
      UPDATE ord SET ord.total = ord.total + :NEW.itemtot - :OLD.itemtot
      WHERE ord.ordid =
         DECODE(:NEW.ordid, NULL, :OLD.ordid, :NEW.ordid);

   END IF;

END;

/
```


8.4 Les triggers Base de données

■ Exécution d'un trigger BD

Etapes

1. Le trigger doit être armé
2. le code du trigger sera **récompilé à sa première exécution** (code absent de la SGA) ou si modification des objets référencés dans le code du trigger (*note*: récompilation inutile depuis la 7.3)
3. Exécuter le trigger

Notes

a) un trigger doit avoir **moins de 60 lignes**

b) **utiliser les procédures pour étendre la taille** d'un trigger

c) pour un trigger ligne avec la clause WHEN, ce dernier ne sera exécuté que si la clause **WHEN** est vérifée

d) en cas d'utilisation de trigger pour alimenter une table distante, le **commit à deux phases** est assuré

e) si N trigger du même type (INSERT, UPDATE, DELETE) existe, Oracle les **exécute** tous **sans un ordre** particulier

f) un trigger ne peut lire les données impropres

g) un trigger **ne peut violer les contraintes** d'intégrité d'une table

8.4 Les triggers Base de données

■ Modification d'un trigger BD

- Modifier un trigger consiste à l'**activer**, **désactiver** ou **récompiler** (privilège requis : ALTER TRIGGER)
- **Syntaxe**
  ALTER TRIGGER [schema.]trigger
  {ENABLE | DISABLE | COMPILE}
- **Exemple**

  sql> ALTER TRIGGER updateset COMPILE ;
  sql> ALTER TRIGGER updateset DISABLE ;

■ Suppression d'un trigger

- **Privilège requis** : CREATE TRIGGER ou DROP ANY TRIGGER
- **Syntaxe**: DROP TRIGGER [schema.]trigger ;
- **Exemple** : DROP TRIGGER updateset;

8.4 Les triggers Base de données

■ Domaines d'applications

- fournir des **mécanismes sophistiqués d'audit**
- renforcer les **contraintes d'intégrité** non supportées par Oracle en natif
- mettre en oeuvre **des règles complexes de gestion** par exemple : changer la catégorie d'un employé si son salaire change
- **renforcer la sécurité** : interdire par exemple la modification des salaires les jours fériés et les week-end
- **assurer la réplication synchrone** à distance de tables
- **propager des actions sur d'autres tables** en fonction des événements survenus

Domaines d'applications (suite)

Exemples 1 : Utilisation de trigger pour l'audit

Création d'un trigger qui permet (lors d'une suppression, modification ou insertion) d'insérer dans une table journal, l'état avant et après d'une ligne de la table EMP.

```
CREATE TRIGGER audit_employe
   AFTER INSERT OR DELETE OR UPDATE ON emp FOR EACH ROW
   BEGIN
      -- La raison du déclenchement du trigger doit être passée en paramètre
      -- à la fonction ci-dessous.
      -- AUDITPACKAGE.SET_REASON(texte_de_la_raison_d'audit)


      IF auditpackage.reason IS NULL THEN
         raise_application_error(-20201, 'Une raison doit être donnée'
         || 'grâce à AUDITPACKAGE.SET_REASON(
               texte_de_la_raison_d'audit)');
      END IF ;

      --  Si la condition précédente est remplie alors ...
      INSERT INTO audit_employe VALUES
         (:old.EMPNO, :old.ENAME, :old.JOB, :old.MGR, :old.HIREDATE,
            :old.SAL, :old.COMM, :old.DEPTNO, :new.EMPNO,
            :new.ENAME, :new.JOB, :new.MGR, :new.HIREDATE,
            :new.SAL, :new.COMM, :new.DEPTNO,
            auditpackage.reasion, user, sysdate);
/
```


Visualisation des informations sur les triggers

- **Les vues contenant les informations sur les triggers**

  . USER_TRIGGERS, ALL_TRIGGERS, DBA_TRIGGERS

Exemple

```
sql>SELECT trigger_type, triggering_event, table_name
 FROM user_triggers WHERE trigger_name = 'UPDATESET';
```

TRIGGER_TYPE	TRIGGERING_EVENT	TABLE_NAME
AFTER EACH ROW	UPDATE OR DELETE	DEPT

```
sql>SELECT trigger_body
   FROM user_triggers  WHERE trigger_name = 'UPDATESET';
```

TRIGGER_BODY

```
BEGIN
   IF UPDATING AND :OLD.deptno != :NEW.deptno
      OR DELETING THEN
         UPDATE ...
      ...
END;
```