

# Administration des bases de données sous Oracle

Fabien De Marchi

Université de Lyon

# Prérequis et objectifs

- Prérequis
  - Modèle relationnel (structure, contraintes, SQL)
- Objectifs
  - Connaître les tâches d'un DBA
  - Connaître les concepts et points clés de l'architecture Oracle
  - Savoir effectuer les principales tâches sous Oracle

## Pour en savoir plus...

- Documentation Oracle : [www.oracle.com](http://www.oracle.com)
  - En anglais, 2000 pages sur l'administration : très complet...



# Les métiers autour des bases de données

- Administrateur
- Responsable de la sécurité
- Administrateur réseaux
- Développeurs d'application
- Administrateurs d'application
- Utilisateurs : modifier les données, créer des rapports

## Note

Dans des environnements de petite taille, l'administrateur peut jouer quasiment tous les rôles

# Rôles du DBA

- Installer les logiciels Oracle
  - un serveur, des applications clientes,
  - En fonction de l'OS et des paramètres systèmes
  - Si fonctionnement en réseau : composants réseaux d'Oracle
- Planifier et créer des bases de données
- Gérer l'espace et implanter les schémas des données
- Assurer la sécurité, l'intégrité et la pérennité des données
- Effectuer des réglages pour optimiser les performances



# Vue d'ensemble

- Deux architectures possibles:
  - client/serveur : des **applications clientes** envoient les requêtes SQL et PL/SQL à un **serveur**.
  - Multitier : des **serveurs d'application** allègent la charge du serveur en réalisant certains accès pour les clients.
- Un serveur de bases de données est composé :
  - d'une **instance** = plusieurs processus et une zone de mémoire
  - d'une **base de données**
  - de plusieurs **schémas**, assimilés à des **utilisateurs**
- Dans le cas de **clusters de machines**, Oracle peut associer plusieurs instance à une même base de données.



# La connexion client/serveur

- Un processus utilisateur est créé quand un utilisateur lance une application cliente
  - Une **connexion** va être créée avec l'instance Oracle, l'utilisateur va ouvrir une **session**
- Un processus serveur va analyser et exécuter les requêtes, retourner les données
  - Mode dédié : une processus serveur pour un processus client
  - Mode partagé : les clients partagent un groupe de processus serveurs
    - Evite les processus serveurs inactifs

# La mémoire

- La SGA (Système Global Area)
  - Zone partagée par tous les utilisateurs de la base de données
  - Allouée au démarrage de l'instance en mémoire principale : doit-être la plus grosse possible.
  - Son but est d'économiser les E/S. Elle contient :
    - le cache de données (database buffer cache) :
    - le cache de reprise (redo log buffer) : log des changements récents
    - le cache d'exécution partagé (shared pool) pour les requêtes SQL et PL/SQL. Contient le dictionnaire de données en cache.
- La PGA (Program Global Area)
  - Zone d'exécution des processus du serveur
  - Allouée au lancement de chaque processus

# Les processus de fond serveur

- Exécutent des actions asynchrones d'écriture et de contrôle
- DBWn (Database Writer) - Ecrit le contenu du cache de données dans les fichiers de données
  - Lorsque la place manque en SGA
  - De façon périodique, en mettant à jour le point de restauration dans les fichiers de log
  - on peut en créer plusieurs : DBW0, DBW1, ... , DBWj
- LGWR (Log Writer) - Ecrit le contenu du cache de reprise dans les fichiers de log
  - en cas de "commit" d'un utilisateur
  - Toutes les trois secondes
  - Quand le cache de reprise est plein au tiers
  - Quand un processus DBWn décharge des données modifiées

# Les processus de fond serveur (cont.)

- CKPT (Checkpoint)
  - Pour assurer la synchronisation et la cohérence des données
- SMON (System Monitor)
  - Effectue la restauration lors de reprise après panne
  - Nettoie les segments temporaires
  - Fusionne certains extents libres contigües
- PMON (Process Monitor)
  - Pour gérer les pannes des processus clients
- RECO (Recover)
  - Pour les reprises après panne de transactions distribuées
- ARCn (Archiver)
  - Pour l'archivage, lorsqu'il est activé
- ...

# Structure physique

- Des fichiers sur le disque
  - un **fichier de contrôle** qui spécifie le nom et l'emplacement des fichiers, le nom de la base,...
  - plusieurs **fichiers de données** pour stocker les données
  - au moins deux **fichiers de reprise** après panne qui contiennent les modifications récentes
  - des **fichiers d'archivage** (optionnel) pour archiver les fichiers de contrôle
  - un **fichier de paramètres** (optionnel) qui stocke tous les paramètres de la base
  - des **fichiers de trace** pour répertorier toutes les tâches et erreurs effectuées

# Structure logique

- Des concepts pour gérer finement la mémoire
  - le **block de données** est l'unité logique la plus fine, 8ko par défaut.
  - l'**extent** est un ensemble contiguë de blocks, alloués en une seule fois.
  - le **segment** est un ensemble d'extents dédiés à un même objet (table, index, ...)
  - le **tablespace** : regroupe un ensemble d'objets (=segments) pour faciliter leur administration. Peut atteindre 8 exabytes !
- Un tablespace = un ou plusieurs fichiers de données

# Contenu d'une schéma

- Correspond à un utilisateur : les deux portent le même nom
- Ensemble d'objets de l'utilisateur manipulables en SQL
- Exemple : tables, index, clusters, triggers, vues, dimensions, fonctions, ...
- Un objet correspond à :
  - plusieurs extents
  - un segment
  - un tablespace
  - un ou plusieurs fichiers du tablespace
- Pas de correspondance schema/tablespace

# Stockage des tables

- Pour des tables de moins de 256 attributs
  - si possible : un tuple est entièrement dans un block
  - sinon : chaînage **inter-blocs** (augmente les E/S)
- pour tes tuples de plus de 256 attributs (compris)
  - chaque partie de 255 tuples est chaînée **intra-bloc** (si possible !)
- Les tuples sont chaînés en utilisant le **ROWID**
- A la création d'une table, on peut :
  - spécifier le tablespace
  - spécifier la taille du segment
  - spécifier la quantité d'espace libre laissée dans chaque bloc



# Les privilèges SYSDBA et SYSOPER

- Permettent d'effectuer les opérations **au-dessus de la base de données**, en se connectant à l'instance
  - Démarrer et arrêter le serveur
  - Créer ou supprimer des bases de données
  - Changer les modes d'archivage de la base
  - ...
- Leur identification est gérée soit :
  - Par le SE,
    - l'utilisateur doit appartenir au groupe dba ou oper (ora\_dba et ora\_oper sous WINDOWS)
    - connect / as sysdba ; connect / as sysoper.
  - Par le fichier password d'Oracle.
    - Créer un fichier password avec l'outil ORAPWD
    - accorder le privilège à partir de l'utilisateur SYS
    - connect user/user as sysdba;

# Le rôle DBA

- Accès au catalogue, gestion des utilisateurs, de la mémoire...
- Dès la création de la base, deux utilisateurs sont créés avec le rôle "DBA"
  - SYS/CHANGE\_ON\_INSTALL
    - Possède toutes les tables systèmes, dont le dictionnaire
    - Ne jamais rien écrire sur ce compte !
  - SYSTEM/MANAGER
    - Pour créer des tables d'administration uniquement
- Il est fortement conseillé de spécifier d'autres mots de passe dans la phase de création.
- Ne pas utiliser dans l'administration quotidienne

# Informations du dictionnaire

- Ensemble de tables appartenant à l'utilisateur SYS
- principales informations
  - La définition de tous les objets et leur espace alloué
  - Les valeurs par défaut des attributs
  - Les contraintes d'intégrité
  - La liste des utilisateurs, leurs privilèges
  - Des informations d'audit sur les accès et modifications
  - ...

# Structure du dictionnaire

- Des tables de base, uniquement accédées par Oracle et SYS
- Des vues mises à la disposition des utilisateurs (selon leurs droits)
  - Vues utilisateurs : USER\_View. Pour les objets de l'utilisateur qui demande
  - Vues d'ensemble : ALL\_VIEWS. Pour une vue d'ensemble sur les objets **auxquels on a accès**
  - Vues d'administration : DBA\_VIEWS. Sur l'ensemble de la base, uniquement SYS.
  - Vues dynamiques sur l'activité de la base : V\_\$VIEWS.

# Enchaînement type (1)

- Une instance Oracle est démarrée sur le serveur
- Une application cliente établit une connexion et ouvre une session
- Le serveur détecte la requête de connexion et crée un processus serveur dédié
- L'utilisateur lance une requête SQL et un commit
- Le processus serveur recherche dans la "shared pool" si la requête existe
  - OUI : elle sera utilisée pour répondre
  - NON : la nouvelle requête est insérée dans la shared pool, analysée et exécutée

## Enchaînement type (2)

- Le processus serveur récupère les données
  - Dans le cache de données
  - dans les fichiers de données (et les charge dans le cache)
- Le processus serveur modifie éventuellement les données dans le cache
- Puisqu'il y a validation, LGWR écrit la transaction dans le fichier de reprise
  - Les changements seront répertoriés dans les fichiers de données par DBWn
  - Le résultat, ou une confirmation, est envoyé au processus utilisateur



# Préliminaires

- Deux possibilités
  - Utiliser l'assistant Oracle : graphique
  - Créer manuellement à l'aide de scripts
- En cas de mise à jour d'Oracle, une procédure particulière existe
  - Compatibilité ascendante, on ne recrée pas les bases de données.
- Prérequis :
  - Oracle doit-être installé
  - Vous devez être administrateur du SE et de l'instance Oracle
  - La mémoire principale et mémoire disque doit être suffisante



# Planifier la base

Phase de réflexion avant création, pour faire les bons choix.

- Réfléchir aux tables et indexes à venir, estimer leur taille
- Planifier la distribution de ses fichiers, l'espace libre dans les blocks
- Décider entre une gestion classique ou automatique des fichiers
- Choisir l'encodage des caractères (peut-être surchargé par les clients)
- Déterminer la taille des blocs de données
- Choisir le mode de gestion de l'annulation
  - Tablespace dédié
  - Segments d'annulation
- Déterminer la stratégie de sauvegarder et reprise après panne

## Etapes de création (1/2)

- Spécifier le nom de l'instance SID
- Créer le fichier de paramètres PFILE (partir de l'exemple Oracle)
  - Nom de la base, emplacement des fichiers de contrôle
- Se connecter à l'instance inactive
  - SQLPLUS /nolog puis CONNECT / AS SYSDBA
- Créer un fichier de paramètre serveur SPFILE (à partir du fichier manuel)
- Lancer l'instance : STARTUP NOMOUNT (Aucune vérification de cohérence à ce niveau)
- Créer la base de données : CREATE DATABASE...

```
CREATE DATABASE mynewdb
USER SYS IDENTIFIED BY pz6r58
USER SYSTEM IDENTIFIED BY y1tz5p
LOGFILE GROUP 1 ('/u01/oracle/oradata/mynewdb/redo01.log') SIZE 100M,
GROUP 2 ('/u01/oracle/oradata/mynewdb/redo02.log') SIZE 100M,
GROUP 3 ('/u01/oracle/oradata/mynewdb/redo03.log') SIZE 100M
MAXLOGFILES 5
MAXLOGMEMBERS 5
MAXLOGHISTORY 1
MAXDATAFILES 100
MAXINSTANCES 1
CHARACTER SET US7ASCII
NATIONAL CHARACTER SET AL16UTF16
DATAFILE '/u01/oracle/oradata/mynewdb/system01.dbf' SIZE 325M REUSE
EXTENT MANAGEMENT LOCAL
SYSAUX DATAFILE '/u01/oracle/oradata/mynewdb/sysaux01.dbf' SIZE 325M REUSE
DEFAULT TABLESPACE tbs_1
DEFAULT TEMPORARY TABLESPACE tempts1
TEMPFILE '/u01/oracle/oradata/mynewdb/temp01.dbf'
SIZE 20M REUSE
UNDO TABLESPACE undotbs
DATAFILE '/u01/oracle/oradata/mynewdb/undotbs01.dbf'
SIZE 200M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;
```

## Etapes de création (2/2)

- Créer de nouveaux tablespace, par exemple :
  - Un USERS pour les utilisateurs
  - un INDX pour les index
- Lancer les scripts de création du dictionnaire (catalog.sql, catproc.sql) et éventuellement d'autres scripts optionnels
- Créer une sauvegarde complète de l'installation dans cet état

## Alternative : fichiers gérés par Oracle

- Oracle peut prendre en charge les fichiers
- Spécifié dans les paramètres d'initialisation
- Simplifie les commandes
- L'administrateur ne gère que la partie logique : tablespace, extents...

```
CREATE DATABASE rbdb1
  USER SYS IDENTIFIED BY pz6r58
  USER SYSTEM IDENTIFIED BY y1tz5p
  UNDO TABLESPACE undotbs
  DEFAULT TEMPORARY TABLESPACE tempts1;
```

## Précisions sur les paramètres de l'instance (pfile)

- le nom de la base ne comporte que 8 caractères
- créer au moins deux fichiers de contrôles
  - Sur des disques différents si possible
- Laisser Oracle choisir la taille des blocs
- On peut contrôler la SGA
  - En lui fixant une taille maximale
  - en fixant la taille de différents composants
  - On peut changer plus tard, de façon dynamique
- On peut fixer le nombre de processus autorisés
  - Nombres de processus utilisateurs + 10
- On doit choisir un mode de gestion des annulations
  - Comment on gère les données d'annulation ? (Avant Commit)
  - Automatique : gérées dans un tablespace
  - Manuel : gérées dans les segments d'annulation

## Remarques complémentaires

- Si la création échoue, visualier le fichier d'alertes
  - Arrêter l'instance
  - Supprimer les fichiers créés pendant la tentative
- Pour supprimer une base :
  - Drop database
  - La base doit être montée et fermée
  - Conserve les fichiers d'archivage et les copies
- Pour modifier des paramètres
  - Les commandes ALTER SYSTEM pour les paramètres d'instance (modifie le SPFILE, pas le PFILE !)
  - Les commande ALTER DATABASE pour revenir sur les choix du CREATE DATABASE
- Pour voir les informations sur la base
  - Show parameters, vues 'database\_properties' et V\$DATABASE



# Configuration du réseau

- Côté serveur
  - Un processus d'écoute utilisant un protocole réseau : listenerSID
  - Un serveur BD Oracle = instance + base
  - Les configurations (=services d'écoute) sont enregistrées dans listener.ora
  - Editable à la main, ou via outils graphiques
- Côté client
  - Configurations (services d'accès) enregistrées dans tnsname.ora
  - Permet au client de se connecter de façon transparente

# Démarrer une base de données

- Oracle Enterprise Manager (OEM)
- SQL\*Plus
  - SQLplus /nolog : lancer SQL\*plus
  - connect / as sysdba
- Commande STARTUP pour démarrer l'instance
  - Utilise le SPFILE ou le PFILE à préciser
  - NOMOUNT : base fermée et non montée
  - MOUNT : base fermée et montée
  - FORCE : ouvre de force, en tuant une éventuelle instance démarrée
- Commande ALTER DATABASE si l'instance est déjà ouverte
  - MOUNT pour monter la base
  - OPEN pour ouvrir la base
  - OPEN READ ONLY ouverte en lecture seule
- ALTER SYSTEM ENABLE RESTRICTED SESSION : seuls les DBA peuvent ouvrir des sessions

# Fermer une base de données

- Commande SHUTDOWN
- Garde la main jusqu'à l'arrêt complet
- Plus personne ne peut se connecter
- Plusieurs modes d'arrêt :
  - NORMAL : attend la déconnexion de tous les utilisateurs
  - IMMEDIATE : annule toutes les transactions non validées et tue les sessions en cours
  - TRANSACTIONAL : attend la fin des transactions puis tue les sessions
  - ABORT : tue les sessions, mais n'annule pas les transactions non validées
- ABORT est à utiliser en dernier recours
- ABORT nécessite une restauration de la base pour retrouver sa cohérence
- On peut suspendre une base de données : ALTER SYSTEM SUSPEND/RESUME

## Mode dédié et mode partagé

- Le mode dédié est par défaut
- Le mode partagé est préférable, sauf :
  - Lorsque l'utilisateur envoie des tâches en batch
  - pour effectuer certaines tâches de restauration
- Pour démarrer le mode partagé
  - Spécifier le nombre de serveurs dans le paramètre d'instance `SHARED_SERVERS`
  - Dans le fichier de paramètre, ou avec `ALTER SYSTEM`
  - Ce nombre peut-être augmenté par Oracle en fonction de la charge

# Terminer des sessions

- On peut tuer une session avec la commande ALTER SYSTEM KILL SESSION 'SI,SN'
  - SI est l'identifiant système
  - SN est le numéro de série
- récupérer ces identifiants dans la vu V\$SESSION
- Les transactions en cours sont annulées

# Surveiller l'activité de la base (1)

- Doit être constamment fait en tâche de fond du DBA
  - Pour repérer des erreurs éventuelles
  - Pour devenir familier du comportement normal...
- Alertes du serveur
  - à partir de seuils : trop de validations, trop d'accès disque, temps de réponse trop long...
  - à partir d'évènements : une vue est trop ancienne, ...
  - Paramétrables via OEM ou package PL/SQL :  
DBMS\_SERVER\_ALERTS
  - Accessibles dans des vues du dictionnaire

## Surveiller l'activité de la base (2)

- Chaque processus serveur est associé à un fichier trace.
- le **fichier d'alerte** recueil en particulier
  - erreurs internes, corruption de blocs, blocages
  - toutes les commandes d'administration
  - les erreurs relatives aux processus serveurs partagés
  - les erreurs durant le rafraîchissement des vues
  - les valeurs des paramètres d'initialisation spécifiés par l'utilisateur
- Il est possible de maîtriser :
  - l'emplacement des fichiers trace
  - leur taille
  - le traçage des processus serveurs, pour collecter des statistiques





## Qu'est ce que c'est ?

- Fichier binaire qui décrit la structure physique de la base
  - Le nom de la base
  - Les fichiers de données et de reprise
  - La date de création de la base
  - La position courante dans les logs
  - Des informations sur les point de synchronisation (checkpoint)
- Nécessaire pour monter la base
- Fortement recommandé en cas de restauration
- Généré lors de la création de la base
  - Avec au moins une réplication
  - On peut faire d'autres réplifications plus tard

# Grandes lignes

- On peut en spécifier lors de la création (nom par défaut sinon)
- Ils doivent être répliqués (multiplexage) sur des disques séparés
  - Lorsque l'instance est arrêtée, créer la copie du fichier
  - Déclarer le nouveau fichier dans le fichier de paramètres
- Même procédure pour le renommage
- Ils doivent être sauvegardés. Deux options :
  - ALTER DATABASE BACKUP CONTROLFILE TO filename
  - ALTER DATABASE BACKUP CONTROLFILE TO TRACE
    - génère un script CREATE CONTROL FILE... dans le fichier trace

## Création de nouveaux fichiers de contrôle

- Toutes les réplications sont endommagées, pas de sauvegarde
- On modifie un paramètre de création qui affecte la taille
  - MAXLOGFILES, MAXLOGMEMBERS, MAXLOGHISTORY, MAXDATAFILES, MAXINSTANCES
  - le nom de la base
- Procédure :
  - répertorier tous les fichiers données et reprise
  - arrêter l'instance, sauvegarder les fichiers, redémarrer en NOMOUNT
  - CREATE CONTROL FILE (cf doc SQL), puis sauvegarde du nouveau fichier
  - Spécifier les nouveaux fichiers de contrôle dans le fichier de paramètres
  - Restaurer éventuellement la base, si des fichiers ont été perdus
  - Ouvrir la base
- Visualiser le fichier d'alertes pour vérifier les erreurs

# Suppression et visualisation

- Pour supprimer, il suffit de supprimer dans le fichier de paramètre
  - Attention : toujours garder 2 fichiers de contrôle !
- Informations sur les fichiers de contrôle :
  - V\$DATABASE : ce que contient le fichier de contrôle
  - V\$CONTROLFILE : nom des fichiers de contrôle
  - V\$PARAMETER : voir les paramètres d'initialization

## Qu'est-ce que c'est ?

- Fichiers "journaux" qui stockent tous les changements, pour la reprise après panne
- Organisation physique et logique :
  - Un **groupe** est un ensemble de **fichiers multiplexés** (membres)
  - Il faut au moins deux groupes avec un fichier chacun
  - Un fichier doit être vu comme un ensemble d'**enregistrements**
  - Un enregistrement est un ensemble de **vecteurs de changement**
- Un vecteur est créé pour chaque bloc de données modifié
- Stocke aussi les informations d'annulation

# Fonctionnement

- Accédés uniquement par le processus LGWR (Log Writer)
- De façon circulaire : groupe après groupe
- Déclenchement : validation, ou cache de reprise rempli
  - Assigne un identifiant unique à l'enregistrement créé
  - Si pas de validation : possibilité d'annulation
- En fin de boucle, les premiers fichiers sont écrasés (attente synchronisation)
- Possibilité d'**archiver** les fichiers de reprise pleins
- Si problème d'écriture sur tout un groupe : l'instance s'arrête.
  - Erreur reportée dans le fichier d'alerte
  - Il faut corriger et redémarrer

# Généralités et conseils

- Si possible, un disque par fichier de reprise
  - Archivage sur un autre disque
  - Fichiers de données sur un autre disque
- Pour choisir la taille des fichiers, penser à l'archivage
- Commencer par deux groupes, puis vérifier le fichier de trace
  - Si trop de blocages, augmenter le nombre de groupes
- On peut créer ou supprimer des groupes ou des membres, renommer des membres
  - sans dépasser MAXLOGFILES et MAXLOGMEMBERS
  - privilège ALTER DATABASE, en général base démmontée.
- Vues V\$LOG, V\$LOGFILE

## Qu'est-ce que c'est ?

- Fichiers physiques où sont stockées toutes les structures logiques
- Un fichier de données est associé à un tablespace
  - Possède un numéro unique dans la base (**absolu**)
  - possède un numéro unique dans le tablespace (**relatif**)
- Au moins deux : tablespaces SYSTEM et SYSAUX
- Le nombre max est fixé par le paramètre DB\_FILES
- Certaines limites sont imposées par le SE
- Les fichiers temporaires sont des fichiers de données un peu spéciaux



# Manipulations des fichiers de données

- Création des tablespaces ou avec la base
  - CREATE/ALTER TABLESPACE
  - CREATE/ALTER DATABASE
- On peut modifier leur taille à la volée
  - Automatiquement (ici par pas de 512k) : AUTOEXTEND ON NEXT 512k
  - Manuellement : ALTER DATAFILE ... RESIZE ...
- Peuvent être mis ONLINE ou OFFLINE
- peuvent être renommés ou déplacés
  - Renommer physiquement, puis logiquement avec ALTER TABLESPACE ... RENAME DATAFILE ... TO
  - Manuellement : ALTER DATAFILE ... RESIZE ...
- Informations : DBA\_DATA\_FILES, V\$DATAFILES

# Grandes lignes

- Espace de stockage logique des objets
- Un tablespace = plusieurs fichiers de données
  - Un seul dans le cas des tablespace à gros fichier (bigfile)
- Il est nécessaire d'utiliser plusieurs tablespace
  - Séparer les données des utilisateurs du dictionnaire
  - Séparer les données d'applications différentes, pour alléger les accès et réduire les risques de panne des applis
  - Tablespaces optimisés pour certaines tâches : lecture seule ou mises à jour fréquentes, espace temporaire
  - Sauvegarde individuelle des tablespace
- Stocker un tablespace par disque, pour réduire les concurrences E/S
- Pour chaque utilisateur il faut assigner un quota

# Différents types de tablespaces

- Tablespaces à gestion locale des extents (par défaut !)
  - Gère la mémoire localement avec des bitmaps
  - Améliore les performances
  - Allège les accès au dictionnaire
  - Permet le choix automatique de l'unité d'extention
  - Autorise la gestion automatique de l'espace dans les segments
  - peut s'appliquer à tous les tablespaces
- Tablespaces gérer par le dictionnaires
  - Gestion plus précise
  - Nécessite une fusion périodique des extents libres
- Bigfile tablespaces (jusqu'à 128 tera)
  - Lorsqu'on a besoin de plus de capacité (64 000 fichiers de données au maximum)
  - Simplifie la gestion du tablespace (un seul fichier)
  - Uniquement pour les tablespace à gestion locale et gestion automatique de l'espace

## Différents types de tablespaces (cont.)

- Tablespaces temporaires : pour la gestion des tris
  - Une instance + un tablespace → un segment de tri
  - Affectés à chaque utilisateurs
  - Forcément à gestion locale depuis 10g
  - ne génère pas d'écriture dans les fichiers de reprise
  - Possibilité de créer des groupes, pour des tris parallèles
- Tablespace d'annulation, pour les opérations non validées

# Le tablespace d'annulation

- Stocke les opérations non validées. Utilisées dans les annulations, et les restaurations
- Jusqu'à la version 8 : Rollback Segments
- Maintenant : tablespace dédié à l'annulation et géré automatiquement
  - Fortement recommandé, mais pas par défaut (paramètres du PFILE)
- Les opérations validées sont écrasées, mais possibilité de les conserver

# Maintenance des tablespaces

Les opérations suivantes sont réalisables sur un tablespace :

- demander la non-génération de log: NO LOGGING
  - Exemple : CREATE TABLE NOLOGGING
- marquer indisponible : OFFLINE
  - Impossible pour SYSTEM, d'annulation ou temporaire
- Placer en mode lecture seule
- Renommer
- Supprimer (pas de restauration possible !)
- Les transporter sur une autre base de données ou plateforme

## Gestion automatique des fichiers

- Décharge le DBA de la gestion des fichiers physiques
- S'applique aux tablespaces, fichiers de reprise, de contrôle, d'archivage, ...
- Autorisé par des paramètres du PFILE : spécifient les répertoires
- Des noms standards sont donnés aux fichiers

# Gestion automatique du stockage

- Niveau de décharge supplémentaire pour le DBA
- Optimise automatiquement les E/S.
- Protège les données : gère la réplication des fichiers, leur répartition sur les disques
- Principe :
  - Le DBA définit des disques, ou groupes de disques : partagés par plusieurs bases
  - Ces groupes peuvent être créés ou modifiés après la création de la base
  - Les fichiers ne sont pas visibles à partir du SE !
  - Trois niveaux de redondance : normal, élevé ou externe
  - Possibilité de définir des  **patrons de fichiers**  dans les groupes
  - Notion de "failure groups" : des disques qui courent un même risque, et ne peuvent donc se servir mutuellement de miroirs
  - Géré par une instance particulière d'Oracle !



## Instance de stockage automatique

- Instance particulière dédiée au stockage automatique
- Utilisée par une ou plusieurs instances de base de données
- Ne peut pas monter elle-même de base de données
- Gère simplement la répartition des fichiers, pas leur exploitation
- Deux nouveaux processus de fond : RBAL et ARB, pour répartir l'activité sur les disques
- Implique un nouveau processus sur les instances de BD : ASMB pour communiquer avec l'ASM
- Doit posséder les mêmes SYSDBA que les instances de BD liées
- Paramètres d'initialisation (PFILE) spécifiques



# Introduction

- Il faut toujours avoir à l'esprit une politique de sécurité
- Assurée par le DBA, ou un administrateur dédié
- Principales tâches :
  - Gérer les utilisateurs
  - Affecter les ressources : tablespaces, quotas,...
  - Gérer les privilèges et les rôles
  - Surveiller l'usage de la base de données (Audit)
- Deux niveaux de sécurité
  - Les comptes utilisateurs : login et mot de passe
  - Rôles, privilèges et profils : contrôle l'accès aux objets et aux commandes systèmes
- Sans oublier la sécurité du SE pour les fichiers, et la sécurité "physique" des serveurs...

# Introduction

- 1 N'installer que ce qui est nécessaire

# Introduction

- 1 N'installer que ce qui est nécessaire
- 2 Sécuriser les compte par défaut après installation

# Introduction

- 1 N'installer que ce qui est nécessaire
- 2 Sécuriser les compte par défaut après installation
- 3 Utiliser des mots de passe sécurisés, renouveler régulièrement

# Introduction

- 1 N'installer que ce qui est nécessaire
- 2 Sécuriser les compte par défaut après installation
- 3 Utiliser des mots de passe sécurisés, renouveler régulièrement
- 4 S'assurer que `O7_DICTIONARY_ACCESSIBILITY=FALSE`

# Introduction

- 1 N'installer que ce qui est nécessaire
- 2 Sécuriser les compte par défaut après installation
- 3 Utiliser des mots de passe sécurisés, renouveler régulièrement
- 4 S'assurer que `O7_DICTIONARY_ACCESSIBILITY=FALSE`
- 5 Toujours accorder le minimum de privilèges aux utilisateurs



# Introduction

- 1 N'installer que ce qui est nécessaire
- 2 Sécuriser les compte par défaut après installation
- 3 Utiliser des mots de passe sécurisés, renouveler régulièrement
- 4 S'assurer que `O7_DICTIONARY_ACCESSIBILITY=FALSE`
- 5 Toujours accorder le minimum de privilèges aux utilisateurs
- 6 S'assurer que `remote_os_authentication=FALSE`

# Introduction

- 1 N'installer que ce qui est nécessaire
- 2 Sécuriser les compte par défaut après installation
- 3 Utiliser des mots de passe sécurisés, renouveler régulièrement
- 4 S'assurer que `O7_DICTIONARY_ACCESSIBILITY=FALSE`
- 5 Toujours accorder le minimum de privilèges aux utilisateurs
- 6 S'assurer que `remote_os_authentication=FALSE`
- 7 S'assurer de la sécurité du SE

# Introduction

- 1 N'installer que ce qui est nécessaire
- 2 Sécuriser les compte par défaut après installation
- 3 Utiliser des mots de passe sécurisés, renouveler régulièrement
- 4 S'assurer que `O7_DICTIONARY_ACCESSIBILITY=FALSE`
- 5 Toujours accorder le minimum de privilèges aux utilisateurs
- 6 S'assurer que `remote_os_authentication=FALSE`
- 7 S'assurer de la sécurité du SE
- 8 S'assurer de la sécurité du réseau (Ex. : SSL)

# Introduction

- 1 N'installer que ce qui est nécessaire
- 2 Sécuriser les compte par défaut après installation
- 3 Utiliser des mots de passe sécurisés, renouveler régulièrement
- 4 S'assurer que `O7_DICTIONARY_ACCESSIBILITY=FALSE`
- 5 Toujours accorder le minimum de privilèges aux utilisateurs
- 6 S'assurer que `remote_os_authentication=FALSE`
- 7 S'assurer de la sécurité du SE
- 8 S'assurer de la sécurité du réseau (Ex. : SSL)
- 9 Appliquer les correctifs de sécurité d'Oracle
  - <http://otn.oracle.com/deploy/security/alerts.htm>

# Introduction

- 1 N'installer que ce qui est nécessaire
- 2 Sécuriser les compte par défaut après installation
- 3 Utiliser des mots de passe sécurisés, renouveler régulièrement
- 4 S'assurer que `O7_DICTIONARY_ACCESSIBILITY=FALSE`
- 5 Toujours accorder le minimum de privilèges aux utilisateurs
- 6 S'assurer que `remote_os_authentication=FALSE`
- 7 S'assurer de la sécurité du SE
- 8 S'assurer de la sécurité du réseau (Ex. : SSL)
- 9 Appliquer les correctifs de sécurité d'Oracle
  - <http://otn.oracle.com/deploy/security/alerts.htm>
- 10 Signaler les failles à [secalert\\_us@oracle.com](mailto:secalert_us@oracle.com)

# Types d'utilisateurs

- Un compte utilisateur = Un schéma de BD
  - C'est un ensemble de d'objets : tables, vues, index,...
  - L'utilisateur crée, modifie,... ses objets
- On peut créer des utilisateurs "simples"
  - Droits de requêtes sur un schéma précis
  - Pas de droits de créations

# Etapes de création

- Choisir un nom et un mode d'identification
  - identification Oracle ou SE (EXTERNALLY)
- Identifier les tablespaces : trois en général
  - Données = tablespace par défaut de l'utilisateur
  - Tris = tablespace temporaire
  - Index = tablespace dédié aux index en général
- Décider les quotas pour chaque tablespace
- Créer l'utilisateur
- Accorder les rôles et privilèges

# Modification et suppression

- Modifications avec l'ordre ALTER USER
  - Changement de mot de passe
  - Suppression ou modification de quotas
  - Changement de tablespace
  - Gestion des droits
- Suppression avec DROP USER
  - Option CASCADE : supprime aussi les objets
  - Impossible de supprimer un utilisateur connecté
- Vues DBA\_USERS, DBA\_TS\_QUOTAS



# Les profils

- Ensemble nommé de limites de ressources
  - Nombre de connexions simultanées,...
- Ils sont affectables aux utilisateurs
- Opérations possibles :
  - Création, modification, suppression
  - Activation/désactivation
- Vues DBA\_USERS, DBA\_PROFILES

# Les privilèges

- Granularité fine des droits
  - Systèmes : opérations sur la base ou un type d'objet (plus de 100)
  - Objets : opération sur un objet précis
- Accordés avec l'ordre GRANT, retirés avec l'ordre REVOKE
- vues DBA\_SYS\_PRIVS, SESSION\_PRIVS, DBA\_TAB\_PRIVS, DBA\_COL\_PRIVS

# Les rôles

- Ensembles de privilèges, pour simplifier leur gestion
- créés et supprimés comme des utilisateurs
- Accordés avec l'ordre GRANT, retirés avec l'ordre REVOKE
- Peuvent être accordés/retirés d'une session
- Quelques conseils
  - Un rôle pour chaque tâche d'application
  - Un rôle pour chaque type d'utilisateur
  - Attribuer aux rôles utilisateurs des rôles d'application : Pas de privilèges individuel
- vues DBA\_ROLES, DBA\_ROLE\_PRIVS, ROLE\_ROLE\_PRIVS,...

## Attention

Pour manipuler des objets via une procédure, il faut posséder explicitement les privilèges requis. Il ne doivent pas être accordés par un rôle.

# L'audit sous Oracle

- Pour surveiller des activités ciblées sur la base
- Possible sur toute action sur la base
  - Tentatives de login
  - Accès aux objets
  - Actions bases de données
- Enregistré dans la table SYS.AUD\$, ou dans le SE
- Activation dans les paramètre d'instance (AUDIT\_TRAIL)

# Exemples d'Audit

- Audit de logins
  - AUDIT SESSION WHENEVER NOT SUCCESSFULL
- Audit d'actions
  - AUDIT update table BY SCOTT
- Commandes DML sur un objet
  - AUDIT insert ON scott.emp
- résultats dans DBA\_AUDIT\_SESSION,  
DBA\_AUDIT\_OBJECT,



# Introduction

La pérennité des données est assurée par :

- Les fichiers de reprise
  - En cas de panne d'instance
- Les sauvegardes de fichiers
  - A froid : instance éteinte, à partir du SE
  - Pour les pannes de disque, pertes de fichiers
- L'archivage
  - Réalisé à chaud par l'instance
  - Archive tous les fichiers de reprise
  - Seule solution pour sécurité totale
  - Augmente la charge du DBA

# Sauvegarde à froid

- La plus simple à mettre en oeuvre
- 1 : identifier les fichiers de données/reprise/contrôle
- 2 : Arrêter la base de données
- 3 : Sauvegarder tous les fichiers avec le SE
- 4 : Redémarrer la base
- En cas de panne : il suffit de reprendre ces fichiers
  - Intéressant dans des bases avec peu de modifications



# Sauvegarde à chaud

- Fichiers de données
  - Sauvegarde avec le SE sur un fichier OFFLINE
  - Si le fichier est ONLINE : on le “protège” avec les commandes BEGIN /END BACKUP
- Fichiers de contrôle
  - Commande ALTER DATABASE
- ATTENTION : a faire en mode archivage automatique
  - Sinon pas de restauration possible

- Dépend de la politique locale vis-à-vis des données
  - Peut-on envisager une perte de données
  - Si oui, quelle est la durée tolérable ?
- Assure une cohérence parfaite des données en cas de panne
- Effectué à chaque basculement de groupe dans les logs
- Possibilité de le lancer ponctuellement "à la main"
  - ALTER SYSTEM ARCHIVE LOG (déconseillé)

- Sans le mode archivage
  - Perte des données depuis la dernière sauvegarde
  - Restauration complète
- En mode archivage
  - Restauration complète ou non
  - Perte de fichiers de données ou du fichiers de contrôle

## Restauration complète sans archivage

- 1 : Fermer la base
- 2 : Restaurer tous les fichiers de la sauvegarde la plus récente
- 3 : Ouvrir la base
- Si les emplacements on changé, modifier les déclarations avant de monter la base

# Restauration avec archivage

- Commande RECOVER
- Possibilité de restaurer :
  - Toute la base
  - des tablespaces
  - le fichier de contrôle
  - des fichiers de données
- A partir :
  - des fichiers de reprise archivés et
  - de la sauvegarde la plus récente

## Exemple : restauration partielle

- Ouvrir la base
- Désactiver les tablespace (offline)
- Réparer la panne disque
- Restaurer seulement les fichiers endommagés
- lancer la restauration
  - RECOVER TABLESPACE/DATAFILE
- Activer le tablespace



# Le réglage des bases de données ?

- Réglage de la configuration initiale en fonction de la charge
- Réglage de l'instance et du SE
  - Identifier les points difficiles et les surmonter
  - Récolter des statistiques sur le fonctionnement
    - des applications
    - de la base et du SE
    - des E/S sur les disques
    - du réseau
  - Découvrir les causes à partir des symptômes et corriger
    - En général : Applications, BD ou matériel
- Réglage des requêtes
  - Souvent écrites à l'intérieur d'outils clients, non optimisées
  - Réglage différent pour OLTP/OLAP
  - Visualiser et contrôler les choix de l'optimiseur



# Les outils Oracle pour le réglage

- Gestionnaire automatique de charge
  - collecte, traite et maintient des stats sur les performances
- Moniteur de diagnostic automatique sur la BD
  - Analyse la charge pour suggérer des problèmes sur la base
- Assistant de réglage SQL
  - Conseils pour l'optimisation des expressions
- Assistant d'accès SQL
  - Conseils sur les index et les vues
- Traceur d'application
  - Pour identifier des surcharges par des applications ou utilisateurs précis
- Sans oublier les alertes serveur (cf la surveillance de la base)

# Concevoir et développer pour la performance

- Eviter absolument les conflits et limites de ressource
- Ne pas penser que l'investissement en matériel va assurer les performances
- Penser en terme de " passage à l'échelle "
  - Comportement linéaire dans la charge de travail
  - Spécificités internet
    - disponibilité 24/24
    - nombre d'accès imprévisible
    - souplesse des requêtes
    - Volatilité et exigence des utilisateurs (7s. d'attente au maximum)
    - Concevoir/développer vite et bien !
  - Causes de mauvaises performances
    - Mauvaise conception, ou mauvaise implémentation
    - Mauvais dimensionnement matériel
    - Limitations logicielles : application, DBMS ou SE

# Concevoir et développer pour la performance (cont.)

- Savoir répondre aux questions suivantes
  - Combien d'utilisateurs à supporter ? très peu, peu à beaucoup, une infinité
  - Quelle mode d'interaction ? Navigateur web ou application cliente personnalisée
  - Où sont les utilisateurs ? (Temps de transfert réseaux)
  - Quelle charge d'accès, combien de données en lecture seule ?
  - Quel est le temps de réponse requis par les utilisateurs ?
  - Quelle disponibilité requise par les utilisateurs ?
  - Mises à jour en temps réel ?
  - Quel taille à prévoir pour les données ?
  - Quelles sont les contraintes budgétaires ?

# Principes pour la conception

- Ne pas faire compliqué quand on peut faire simple
  - Eviter les schémas ou requêtes incompréhensibles (utiliser des vues si besoin)
  - Eviter les superpositions de couches logicielles
- Soigner la modélisation des données pour les parties principales
- Implémenter un schéma en 3NF au moins pour assurer la flexibilité
  - Optimiser avec vues matérialisées, clusters, colonnes calculées
  - Bien organiser les index
- Organiser des campagnes de tests crédibles facilitera le déploiement

# Étapes pour la résolution des problèmes

- 1 Vérifications préliminaires (avant les problèmes)
  - 1 Récolter les impressions de base, les projets des utilisateurs
  - 2 Récolter le maximum de statistiques (SE, DB, applications) lorsque les performances sont bonnes et lorsqu'elles sont mauvaises
  - 3 Vérifier régulièrement les SE des utilisateurs (matériel, ressources...)
- 2 Comparer les symptômes avec les "10 erreurs fréquemment commises"
- 3 Réaliser une modélisation conceptuelle du système lors de l'apparition des symptômes
- 4 Lister toutes les solutions et les appliquer une à une jusqu'à l'obtention du résultat, ou l'identification des contraintes extérieures conduisant à l'échec.

## Traitement des urgences...

- Bien souvent, un problème doit-être traité dans l'urgence avant une résolution rigoureuse
- Les étapes sont alors "raccourcies" :
  - 1 Faire l'inventaire des problèmes, des symptômes, des changements récents
- Vérifier l'état du matériel : CPU, disques, mémoire, réseau de chaque tier
- Déterminer si le problème est au niveau du CPU ou de l'attente d'évènement. Utiliser les vues dynamiques sur les performances du catalogue.
- Appliquer des mesures d'urgence pour stabiliser le systèmes : suspendre une application, réduire la charge, tuer un processus...
- vérifier la stabilité du système, récolter des statistiques, et suivre la procédure complète de résolution.

# Les 10 erreurs fréquentes selon Oracle

- 1 Multiplication des connexions à l'instance (une par interaction)
- 2 Mauvaise utilisation des curseurs et variables liées
  - Les curseurs évitent de recalculer une requête pour usages multiples
  - Les variables liées permettent d'identifier des requêtes similaires
  - Attention au SQL généré dynamiquement par les applications
- 3 Requêtes SQL inapproprié aux exigences
- 4 Utilisation de paramètres d'instance non standards
- 5 Mauvaise répartition des E/S sur les disques
- 6 Blocages dans les fichiers de reprise
- 7 Mauvaise gestion des blocks et des segments d'annulation
- 8 Parcours entier de grandes tables
- 9 Trop de SQL récursif de la part de SYS (ex. allocation des extents)

# Choix dans la configuration de l'instance

- Considérations initiales
  - Influence de certains paramètres du PFILE
    - Compatible, db\_block\_size, SGA\_TARGET, PROCESSES, SESSIONS, UNDO\_MANAGEMENT
  - Gestion automatique des annulations conseillée. Voir V\$UNDOSTATS et V\$ROLLSTATS
  - Dimension des fichier de reprise : un basculement toutes les 20 minutes...
  - Créer suffisamment de tablespaces
    - Gestion locale recommandée
    - Ne pas oublier les tablespaces temporaires
- Gestion des tables
  - Compresser les tables en lecture seule
  - Récupérer l'espace libre dans les segments
  - Créer les index après le remplissage des tables
  - Laisser Oracle gérer la mémoire pour les tris dans la PGA



## Choix dans la configuration de l'instance (cont.)

- Performances en mode processus serveurs partagés
  - Rappel : un groupe de serveurs pour tous les clients
  - Surveiller la charge des dispatcher avec V\$DISPATCHER et V\$DISPATCHER\_RATE
    - Le taux d'utilisation courant doit être loin du taux maximal
  - Identifier les blocages des processus serveurs
    - Avec V\$QUEUE, pour le temps moyen d'attente des requêtes ou le nombre de serveurs qui tournent actuellement
    - Ajouter si nécessaire des serveurs avec les paramètres d'instance
    - Attention, vérifier si les blocages ne sont pas en mémoire SGA

# Statistiques et diagnostics automatiques

- Récolter des statistiques est crucial pour résoudre ou devancer les problèmes
  - Concernant la base de données (donnés par Oracle)
    - Les attentes de processus
    - Le temps cumulé de travail de la base (DBTIME) ou de certaines actions
    - Sur le système et les sessions...
  - Concernant le SE (à récupérer avec des outils externes)
    - Sur l'activité des CPU
    - Sur le swapping
    - Les accès disques
    - Le réseau
- Des rapports automatiques de statistiques peuvent être générés en html par ORACLE
- Des diagnostics et conseils automatiques sont générés à partir des stats
  - Son but est de minimiser DBTIME...

# Gestion de la mémoire

- En général, il est recommandé de laisser Oracle gérer la SGA
  - paramètre `SGA_TARGET` différent de 0
  - ne gère pas, notamment, le cache de reprise (paramètre `LOG_BUFFER`)
- Pour une gestion plus fine, on peut récolter des statistiques sur chaque partie de la SGA et gérer sa taille manuellement.
- Surveiller que le cache de reprise n'entraîne pas de ralentissement
  - Stocker les fichiers de logs sur des disques rapides
  - Dans les scripts, ne pas valider à chaque opération
  - Lors de chargement de grands volumes de données, utiliser `NOLOGGING`
- Utiliser la gestion automatique de la PGA
  - Surveiller et redéfinir si nécessaire

# Gestion des entrées/sorties

- Rechercher la simplicité, tout en assurant les besoins requis. On n'isole des fichiers que pour des raisons de performances ou de pérennité des données, ou s'il s'agit d'une contrainte de gestion
  - Séparer les fichiers qui requièrent beaucoup d'E/S
  - Une table et son index n'ont pas de raison d'être séparés
  - Essayer tout d'abord de réduire les E/S en optimisant les requêtes...
  - Si le problème vient des fichiers de reprise, les isoler
  - Séparer les fichiers de reprise et les archives
- Penser au mode de gestion automatique des fichiers
- Bien choisir la taille de blocks
  - 8 kilo en général
  - éventuellement plus petit pour un système fortement transactionnel
  - Plus grand en entreposage de données

# Introduction : Résumé des tâches et outils

- L'un des aspects théorique et pratique les plus importantes en BD
- Trois tâches principales:
  - identifier les charges SQL cruciales pour les performances
  - Vérifier leur plan d'exécution choisi par l'optimiseur
  - Implémenter des améliorations pour améliorer les performances
- Trois directions pour le tuning
  - Réduire la charge : plans d'exécution et index
  - Répartir la charge dans le temps
  - Paralléliser la charge : typiquement en OLAP
- Outils pour le réglage automatique de requêtes
  - Moniteur de diagnostic automatique
  - Moniteur de réglage SQL
  - Moniteur d'accès SQL (index, vues)

# Introduction : Développer des requêtes efficaces

- Tenir à jour les statistiques de l'optimiseur
- Surveiller les plans d'exécution
- Ecrire efficacement les requêtes
  - Utiliser au maximum des AND et = dans les prédicats
  - Ne pas utiliser de fonctions dans les clauses WHERE, en particulier sur des colonnes indexées
  - Décomposer le plus possible les tâches faites par les requêtes
- Choisir le bon connecteur
  - Préférer IN lorsque la sous-requête est la plus sélective
  - Préférer EXISTS dans le cas contraire
- Si nécessaire, maîtriser les choses en utilisant les "HINT"
- Structurer soigneusement les index, supprimer ceux qui sont inutiles
- Limiter les passes sur les données (utiliser le case par exemple)

# Les capacités de réglage automatique de requêtes

- Processus entièrement automatique de réglage des ordres SQL
  - Mode normal : optimiseur classique choisi un “bon” plan
  - Mode tuning : produit des actions possibles pour améliorer
- Processus coûteux : a utiliser uniquement pour les requêtes problématiques
- Base son analyse sur quatre points :
  - les statistiques
  - les profiles SQL = informations avancées sur une requête, dans le dictionnaire
  - les chemins d'accès : index et vues
  - la syntaxe SQL (ex. : UNION ALL plutôt que UNION)
- Peut-être utilisé avec SQLAdvisor (graphique ou package PLSQL DBMS\_SQLTUNE)
- Possibilité de régler des ensembles de requêtes (SQL Tuning Sets)
  - Un ensemble de requêtes

# Optimiseur de requêtes : ses tâches

- Rappel : SQL est un langage déclaratif
- Optimisation de requête = choix d'un plan d'exécution
  - 1 Ordre d'évaluation des expressions
  - 2 transformation des expressions (ex. requêtes imbriquées en jointures)
  - 3 choix du but, le paramètre à optimiser
  - 4 Choix des chemains d'accès
  - 5 ordre de réalisation des jointures
- Les moyens de maîtrise de l'utilisateur :
  - Déterminer le but
  - Récolter des stats
  - Forcer des choix de l'optimiseur avec "HINT"



# Optimiseur de requête : quel objectif ?

- Deux possibilités sous Oracle
  - Minimiser le temps de réponse global (ALL\_ROWS, par défaut)
  - Minimiser le temps des n premières réponses (FIRST\_ROW)
- n peut être égal à 1, 100 ou 1000
- FIRST\_ROWS est à utiliser en cas d'interaction directe avec l'utilisateur
- C'est un paramètre de session
- On peut forcer l'optimiseur avec un HINT
- On peut se baser sur le temps CPU (par défaut) ou les E/S

# Optimiseur de requêtes : paramètres

- Deux possibilités sous Oracle
  - Minimiser le temps de réponse global (ALL\_ROWS, par défaut)
  - Minimiser le temps des n premières réponses (FIRST\_ROW)
- n peut être égal à 1, 100 ou 1000
- FIRST\_ROWS est à utiliser en cas d'interaction directe avec l'utilisateur
- C'est un paramètre de session
- On peut forcer l'optimiseur avec un HINT
- On peut se baser sur le temps CPU (par défaut) ou les E/S

# Les chemins d'accès du plan d'exécution

- FULL TABLE SCAN : parcours séquentiel complet d'une table
- ROWID SCAN : Accès direct à un ensemble de tuples (après un INDEX SCAN)
- INDEX SCAN : Accès à un index
  - 1 Unique scan, range scan, full scan, index joins, bitmap joins
- CLUSTER ACCESS : parcours d'un cluster
- SAMPLE TABLE SCAN : accès à un extrait de la base, spécifié dans la requête
- On peut toujours les forcer avec un "HINT"

# Les différents types de jointure

- Jointure imbriquée : double boucle
  - Lorsque peu de lignes doivent être jointes
  - La seconde table est accédée rapidement à partir de l'attribut de jointure
  - $O(N \times M)$
- jointure par hachage : la table la plus petite est "hachée" en mémoire
  - Lorsque la table hachée tient en mémoire
  - Un seul parcours de chaque table
- Jointure par tri fusion : tri puis fusion de chaque opérande
  - Lorsque les sources sont déjà triées
  - La condition de jointure est une inégalité
- Jointure cartésienne : produit cartésien, pas de condition de jointure
- Toutes ces variantes existent en jointure externe.

# Gestion des statistiques

- Ensemble d'information quantitatives sur les données, utilisées par l'optimiseur
  - Sur les tables : Nombre de tuples et de blocks, taille des tuples
  - Sur les colonnes : nb valeurs distinctes et de valeurs NULL, histogramme
  - Sur les index : nb blocks feuilles et niveaux, facteur de regroupement
  - Sur le système : performances E/S et CPU
- Stockées dans le dictionnaire
- Récoltées automatiquement par un processus de fond
  - Pendant une fenêtre temporelle définie (par défaut : 22h-18h + week-end)
  - Cette fenêtre peut-être paramétrée
- On peut lancer manuellement après d'importantes modifications
- Les opérations suivantes peuvent être réalisées sur les stats :
  - restaurer des versions anciennes

# Index : quels colonnes ?

- Quelles colonnes faut-il indexer ?
  - Attributs utilisés fréquemment dans les clause WHERE (jointures ou sélections)
  - L'efficacité augmente avec la sélectivité de l'attribut
  - Automatique pour les clés primaires, unique
  - Eviter d'indexer des colonnes fréquemment modifiées
- Inutile si la clé d'indexation est passée en paramètre d'une fonction
  - Utiliser des index de fonctions
- On peut faire des index composés de plusieurs colonnes
  - Si utilisées ensemble avec une clause AND
  - Placer en premier les attributs les plus fréquemment utilisés
  - Sinon, placer en premier celui sur lequel est ordonnée la table

# Les types d'index

- Par défaut : Oracle utilise des arbres équilibrés (B-arbres)
- Clusters : regroupement de tables dans des blocs communs
  - Autour d'un ensemble de colonnes communes
  - La jointure sur ces colonnes est immédiate
  - A décider à la création des tables
- Index Bitmap
  - Pour des valeurs à faible sélectivité sur des grandes tables
  - Bien adapté à des grandes conjonctions de prédicats
- Tables organisées sur l'index : la table est stockée avec l'index
- On peut indexer les résultats d'une fonction sur une colonne (UPPER, LOWER, ...)
- On peut partitionner un index aussi bien qu'une table

# visualiser les plans d'exécution

## ■ EXPLAIN PLAN

- Permet de visualiser le plan choisi par l'optimiseur
- On peut détecter rapidement les points coûteux du plan choisi
- La requête n'est pas exécutée !

## ■ V\$SQL\_PLAN

- Pour voir tous les plans d'exécution des requêtes récemment exécutées
- On a le plan réel, avec les coûts réels, pas des estimations

## ■ Le plan est stocké dans la table PLAN\_TABLE

- Récupérer les requête d'interrogation de la doc pour une sortie formatée



# Visualiser la charge de travail

- 1 Fixer les paramètres d'instance pour la collecte de trace
- 2 Activer la collecte de la trace
  - Produis un fichier avec les statistiques sur toutes les requêtes SQL
- 3 lancer l'utilitaire TKPROF
  - Pour formater le fichier trace et rendre une sortie lisible
- 4 Interpréter le fichier généré (voir la doc de TKPROF)
- 5 Eventuellement, on peut stocker ce résultat dans la base
  - Un script pour cela est fourni par TKPROF