

Administration des bases de données : mise en œuvre avec Oracle

Jean-Marc Petit

2012-2013

Département Informatique

Sur ce cours ... 1/2

- Objectifs du cours :
 - Maîtriser les concepts de l'administration des BD
 - Identifier les points clés de l'environnement
 - Mise en œuvre technique, dompter la doc
- Très orienté sur Oracle
 - Personne n'est parfait ...
 - **Permet de faire 2 séances de TPs**
 - Corollaire :
 - « Certains ordres présents dans ce document peuvent être erronés en fonction de la version d'Oracle que vous utilisez »

Sur ce cours ... 2/2

- Cours **très différent** de mon autre cours en 3IF
 - « Fondamentaux de la modélisation des données »
 - On passe de l'autre côté, de la théorie à la pratique
- Beaucoup de transparents utiles en TP uniquement
 - les transparents = support de cours + documentation technique en TP
- Pour une version indépendante des SGBDs
 - « Database system implementation », Garcia Molina, J. Ullman, J. Widom

3

Remerciements

- De nombreuses personnes ont contribué à ce support et aux TPs
 - Fabien De Marchi, Mdc, Univ Lyon1
 - Hélène Jaudoin, Mdc, ENSSAT, Lannion
 - **Farouk Toumani**, Pr, Univ Blaise Pascal

4

Thèmes du cours

- Concrétiser la **conception physique** des bases de données
- « Démystifier » les ordres SQL simples
 - create table EMP (...);
 - insert into EMP values (...);
 - select * from EMP where ... ;
- Comprendre ce qui se passe « derrière », qui n'est au fond que des lectures/écritures dans des fichiers ...

5

Objectifs sous Oracle

- Comprendre l'*architecture* d'un serveur de BD Oracle
- Démarrage et arrêt d'une *instance* et d'une *base de données* Oracle
- Création d'une base de données opérationnelle
- Gestion des *fichiers* d'une base de données Oracle
- Gestion de la *structure logique* (tablespaces, segments, extents et blocs)
- Gestion des *utilisateurs*, des *privileges* et des *ressources*

6

Pré-requis

- Autour de l'algorithme
 - structure de données : tableau, arbre, table de hachage
 - algorithmes classiques : tris, jointures, parcours de structures de données
- Autour des BD
 - Modèle relationnel
 - Langage de requêtes
 - Transactions
 - Langage de programmation et BD
- Autour des *systèmes d'exploitation*
 - Hiérarchie de la mémoire
 - Politique de gestion des défauts de pages

7

Oracle : une longue histoire

- **1977**
 - Relational Software Inc. (RSI - currently Oracle Corporation) established
- **1978**
 - Oracle V1 ran on PDP-11 under RSX, 128 KB max memory. Written in assembly language. Implementation separated Oracle code and user code. Oracle V1 was never officially released.
- **1980**
 - Oracle V2 released - the first commercially available relational database to use SQL. Oracle runs on on DEC PDP-11 machines. Code is still written in PDP-11 assembly language, but now ran under Vax/VMS.
- **1982**
 - Oracle V3 released, Oracle became the first DBMS to run on mainframes, minicomputers, and PCs (portable codebase). First release to employ transactional processing. Oracle V3's server code was written in C.
- **1983**
 - Relational Software Inc. changed its name to Oracle Corporation.
- **1984**
 - Oracle V4 released, introduced read consistency, was ported to multiple platforms, first interoperability between PC and server.
- **1986**
 - Oracle V5 released. Featured true client/server, VAX-cluster support, and distributed queries. (first DBMS with distributed capabilities).
- **1988**
 - Oracle V6 released - PL/SQL introduced.
- **1989**
 - Released Oracle 6.2 with Symmetric cluster access using the Oracle Parallel Server
- **1991**
 - Reached power of 1,000 TPS on a parallel computing machine. First database to run on a massively parallel computer (Oracle Parallel Server).
- **1992**
 - Released Oracle7 for Unix

8

suite

- **1994**
 - Oracle's headquarters moved to present location. Released Oracle 7.1 and Oracle7 for the PC.
- **1995**
 - Reported gross revenues of almost \$3 billion.
- **1995**
 - OraFAQ.com website launched.
- **1997**
 - Oracle8 released (supports more users, more data, higher availability, and object-relational features)
- **1998**
 - Oracle announces support for the Intel Linux operating system
- **1999**
 - Oracle8i (the "i" is for internet) or Oracle 8.1.5 with Java integration (JVM in the database)
- **2000**
 - Oracle8i Release 2 released
 - Oracle now not only the number one in Databases but also in ERP Applications
 - Oracle9i Application Server generally available: Oracle tools integrated in middle tier
- **2001**
 - Oracle9i Release 1 (with RAC and Advanced Analytic Service)
- **2002**
 - Oracle9i Release 2
- **2004**
 - Oracle10g Release 1 (10.1.0) available ("g" is for grid, the latest buzzword)
- **2005**
 - Oracle10g Release 2 (10.2.0) available
 - ... sans fin !

9

Plan du cours

- **Partie 1 -- Introduction** *diapo 13*
 - Le rôle de *l'administrateur base de données*
 - Composants de l'architecture d'Oracle
 - Utilisation des outils d'administration
- **Partie 2 -- Serveur Oracle** *diapo 49*
 - Gestion d'une instance Oracle
 - Création d'une base de données
 - Création de vues du dictionnaire de données
- **Partie 3 -- Structure physique de stockage** *diapo 105*
 - Fichier de contrôle, de reprise

10

Plan détaillé (2)

- Partie 4 -- Structure logique de stockage
 - Fichiers de données et tablespaces
 - Notions sur les segments
- Partie 5 -- Mise à jour de l'intégrité des données *diapo 203*
- Partie 6 – Sécurité *diapo 209*
 - Utilisateurs, profils, privilèges, rôles
- Partie 7 -- Sauvegarde et restauration *diapo 243*

Partie 1

Introduction à l'administration des bases de données

13

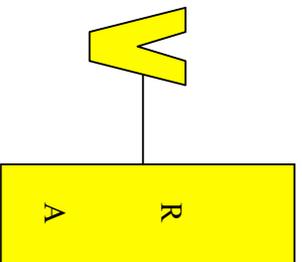
Rappel

- Verrou technologique
 - prédominance des coûts d'E/S entre mémoire vive et mémoire secondaire
- Explique, justifie l'architecture même des SGBD
- A garder en tête pour comprendre le fonctionnement des SGBD

14

Modèle de mémoire RAM

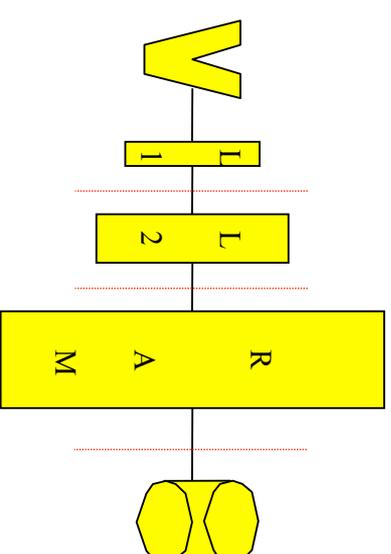
Slides adaptés de Lars Arge



- Modèle théorique ^Mstandard de calcul
 - Très simple
 - Mémoire virtuelle infini
 - Coût d'accès en temps constant
- Modèle crucial pour le succès de l'informatique

15

Mémoire hiérarchique



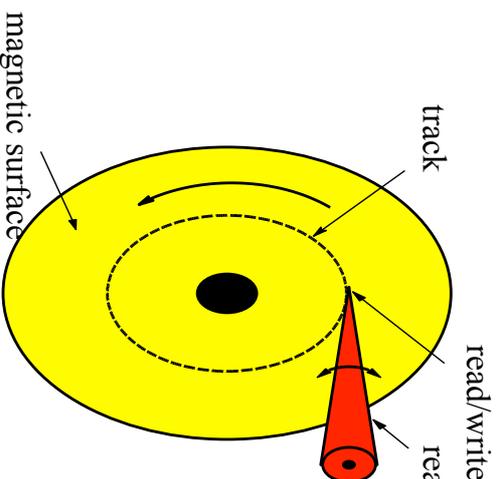
- Machines modernes
 - Plusieurs niveaux de mémoire
- Niveaux loin du processeur : plus grands et plus lents
- Mouvement des données entre les niveaux
 - Utilisation de "bloc" ou "page"
 - De plus en plus grand

16

E/S lente

Accès disque $\sim 10^6$

plus lent que accès
RAM



*“The difference in speed
between modern CPU
analogous to the
difference in speed in
sharpening a pencil
using a sharpener on
one’s desk or by taking
an airplane to the other
side of the world and
using a sharpener on
someone else’s
desk.” (D. Comer)*

Important de stocker et d’accéder
aux données de façon à tirer
profit des blocs (contiguïté)

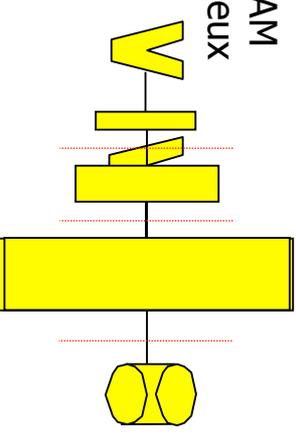
17

Verrou technologique

Programmes dans le modèle RAM

Fonctionne sur des grands jeux
de données

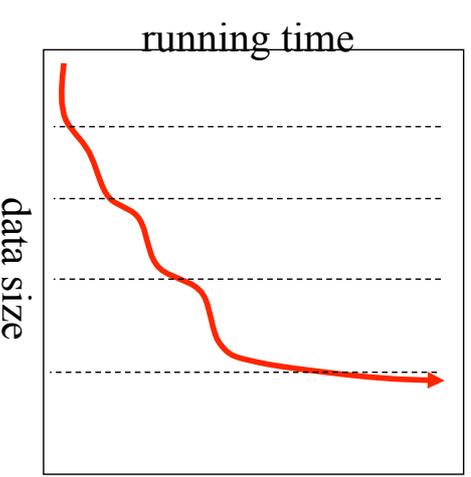
OS déplace les blocs



- OS récents
- stratégies de pagination et de “prefetching” sophistiquées
- Inefficace si le programme fait des accès aléatoires



problèmes de passage à
l’échelle (scalability)



18

Administration des bases de données

- **Conception physique** des bases de données
 - Dernière étape d'un processus de conception de BD
 - Nombreux choix possibles sur la façon d'organiser le serveur de bases de données
 - taille d'un bloc, emplacements des fichiers sur disque, nombre de fichiers de données, taille des fichiers, multiplexage, espace logique de stockage (tablespace), ...
- **La face cachée des bases de données**

19

Tâches de l'administrateur de BD

- Dans la phase de « conception »
 - définition du schéma conceptuel de la base
 - règles de gestion, cohérence des informations
 - cycle de vie des données, volumétrie
- Dans la phase de maintenance
 - Planification et création des BD
 - Gestion des structures physiques
 - Gestion des structures logiques suivant le « design »
 - Gestion de la sécurité, des utilisateurs
 - Sauvegarde et restauration
 - *Optimisation de la base de données*
 - *optimisation de requêtes*
 - *Administration du réseau*

20

Rôles d'un DBA

- Doit maîtriser la **technique** pour mieux s'en affranchir
- « **Penser globalement et fixer localement** »
 - Nécessite de maîtriser les concepts
- Grandes fonctions :
 - *installer* le SGBD et les applications clientes
 - *Créer* la base de données en faisant des choix au niveau physique et la *maintenir*
 - *Gérer les utilisateurs*
 - Assurer la *cohérence* et la *sécurité* des données
 - Echanger des données avec l'extérieur
 - *Améliorer* les performances
 - gestion des ressources mémoires
 - gestion des temps de réponses prohibitifs

21

Tendance actuelle

- Administrateur de bases de données
 - souvent formé "sur le tas"
 - à l'intersection de plusieurs domaines
 - beaucoup de demande, peu d'offre
 - mouton à 5 pattes, env 45 KE/an
- Tendances actuelles
 - progiciels intégrés
 - minimise les besoins en administration ... sans pour autant les supprimer
 - amélioration des outils d'administration par les fournisseurs de SGBD
 - Notion d'*Assistant*
 - pour la création des bases, la sauvegarde/restauration, ...
 - A terme, vers des BD qui s'auto-administrent

22

Exemple avec Oracle

- Deux produits principaux avec Oracle 10g
 - Oracle Entreprise Edition
 - dans la continuité des versions précédentes
 - Oracle Express Edition
 - En rupture avec la "tradition" Oracle
 - Le SGBD fait des choix à la place de l'administrateur
 - plus difficile à "tuner"
 - Objectif : concurrencer SQL server ?
- Exemple concret de self-tuning des BD

23

Composants de l'architecture d'Oracle

1. Les clients et les serveurs d'Oracle
2. Les processus, les structures de la mémoire et les fichiers d'Oracle
3. Étapes du traitement d'une requête d'interrogation
4. Étapes du traitement d'un ordre de modification/ajout/suppression
5. Étapes du traitement d'un ordre de fin de transaction (commit)

24

1. Les clients et les serveurs d'Oracle

1 *serveur* <--> plusieurs *clients*

Oracle Oracle

■ Utilisateurs

- Connexion directe à l'hôte
 - e.g. avec telnet + sql*plus
- Connexion client/serveur ou deux tiers
 - e.g. iSQL, SQL Worksheetou avec une application développée en Client/ Serveur

■ Administrateurs

- chargés de l'entretien du serveur Oracle

25

Connexion à une base de données

- via SQL*Plus ou application tierce => lancement d'un *processus utilisateur* sur le poste du client (qui peut être le serveur)

- lors de la connexion (login/passwd/ service BD)

=> lancement d'un *processus serveur* sur le serveur

- communication

- inter-processus si les 2 processus s'exécutent sur le même poste
- via un logiciel de réseau sinon

- *Session* : connexion spécifique entre un *utilisateur* et un serveur *prêt à l'emploi*, i.e. un serveur démarré.

26

Processus Utilisateur

- Fonctionne sur la machine du client
- Démarre lors de l'appel d'un outil ou d'une application
- Exécute l'outil ou l'application (SQL*Plus, Server Manager, Developer/2000)
- Inclut l'UPI (User Program Interface)
- Appelle le serveur Oracle

27

Processus Serveur

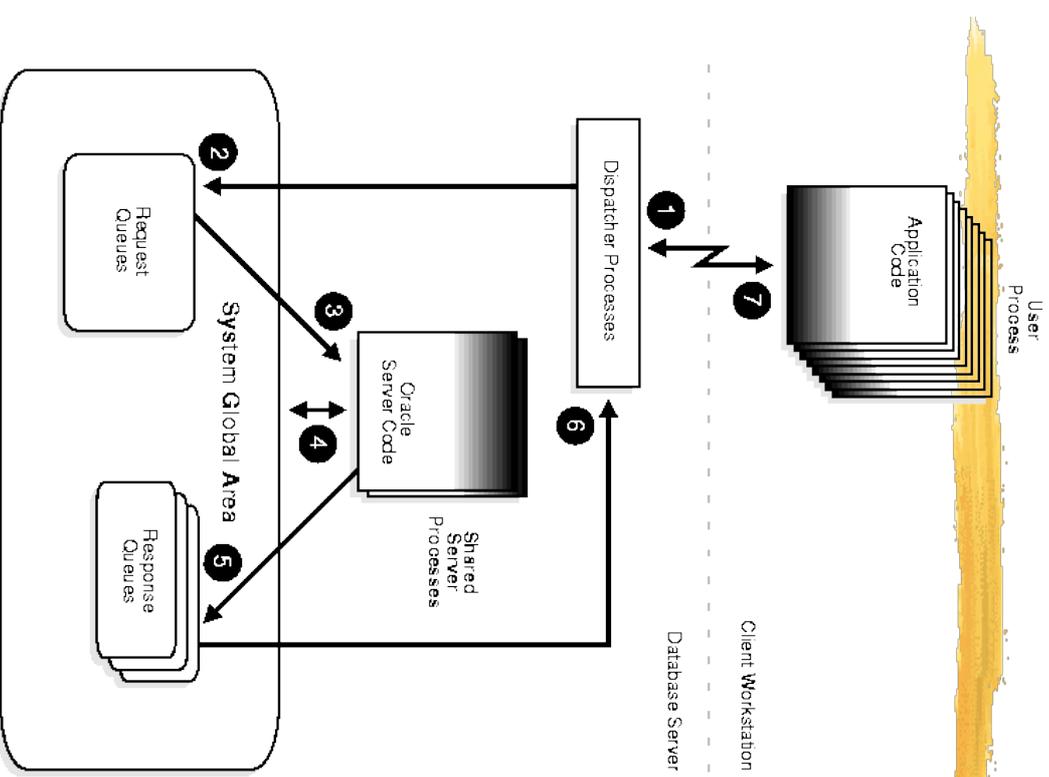
- Fonctionne sur la même machine que le serveur Oracle
- En cas de *serveur dédié*, prend en charge un *unique* processus utilisateur
- Utilise une PGA exclusive (*Program Global Area*)
- Inclut l'OPI (Oracle Program Interface)
- Traite les appels générés par le client
- Retourne les résultats au client
- en cas de *serveur partagé* (MTS), plusieurs processus utilisateurs partagent un processus serveur

28

Architecture

- Architecture à *serveur dédié*
 - 1 processus client <-> 1 processus serveur
- Architecture
 - la plus simple
 - la plus répandue
- Inadapté si beaucoup de clients connectés de façon simultanée
- Architecture à *serveur partagé*
 - n processus clients <-> 1 processus serveur
- Serveur partagé ou Multi-Thread Server (MTS)
 - Gestion d'une file de requêtes des processus clients et de réponses à retourner
 - Dispatcher + Listener

29



30

2. Serveur Oracle

- **Serveur Oracle** =
instance Oracle +
base de données Oracle

- **Instance Oracle**

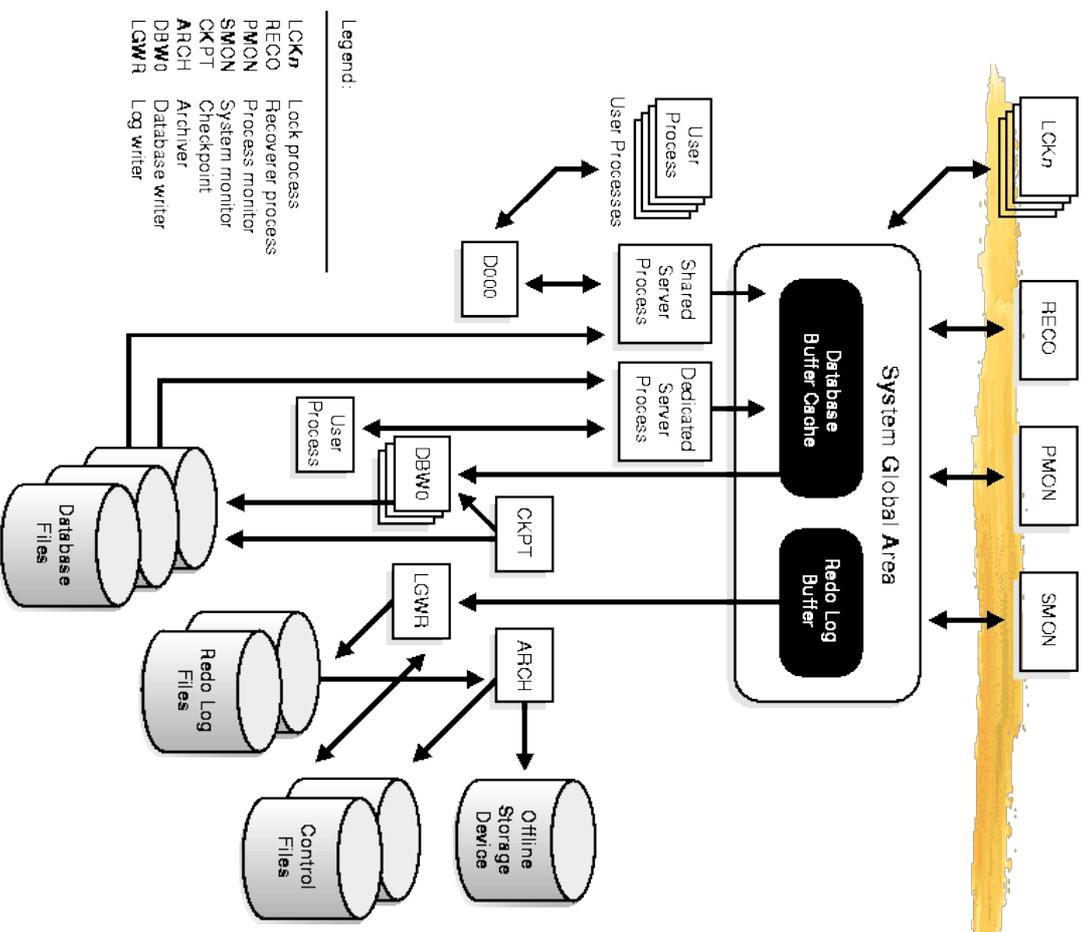
- C'est un moyen pour accéder à une base de données Oracle
- Ouvrir une *unique* base de données
- Instance Oracle = SGA + des processus en arrière plan pour gérer la base

Instance Oracle : la SGA

- SGA (Shared Global Area) ou Zone mémoire globale du système
- contient les données et les informations de contrôle pour le serveur Oracle
- Fait le lien avec le processus serveur
- alloué en mémoire virtuelle par le système d'exploitation du serveur
- Elle comprend
 - Une zone partagée (shared pool)
 - Le cache de données
 - Le cache de reprise (redo log)

Instance Oracle : Processus en arrière plan

- 5 principaux processus sont lancés pour une instance donnée
 - DBWR (processus d'écriture des blocs de données)
 - LGWR (processus d'écriture du fichier de reprise)
 - CKPT : point de synchronisation
 - SMON (processus System Monitor)
 - PMON (processus Process Monitor)
- D'autres processus sont lancés suivant le type de serveur
 - serveur dédié/partagé
 - suivant le mode d'archivage
 - archivelog ou noarchivelog
 - ...

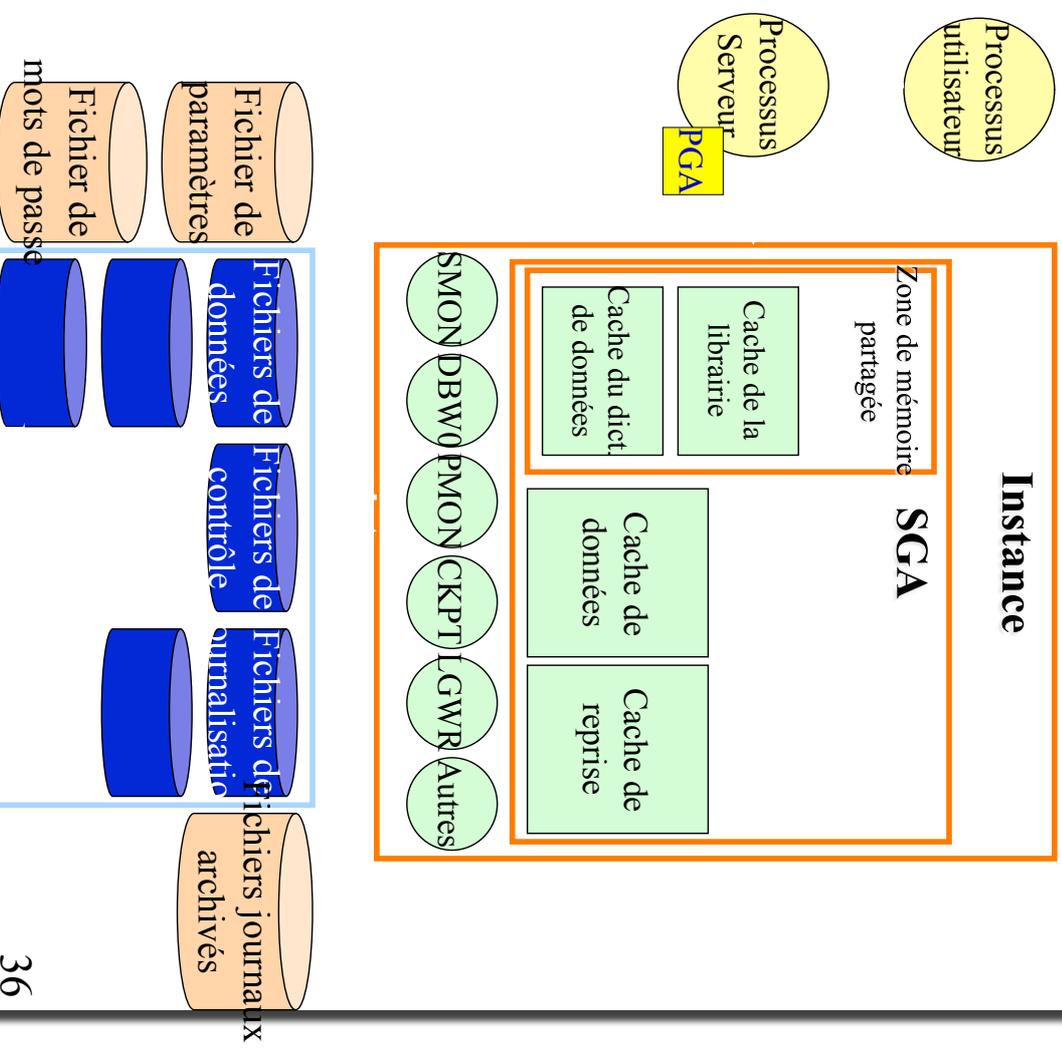


BD Oracle

- désignée par un nom (DB_NAME), souvent pris identique à celui de l'instance correspondante.
- représente les structures physiques des données
- se compose de *fichiers* du système d'exploitation
 - fichiers de données (>=1), (10g >=2)
 - fichiers de reprise (redo log) (>=2)
 - fichiers de contrôle (>=1)
 - fichier de paramètres
 - fichier mot de passe
 - fichiers de reprise archivés

35

Architecture Oracle



36

3. Traitement d'une requête d'interrogation

- Le processus client envoie la requête au processus serveur
- Le processus serveur effectue l'analyse syntaxique et sémantique de la requête
 - utilise la shared pool de la SGA pour compiler l'ordre
 - retourne l'état (analyse correcte ou incorrecte) au processus client.
- Exécution de la requête
- Récupération des résultats
 - le processus serveur envoie les lignes extraites par la requête (à partir du cache de données si les données y ont déjà été chargées)

37

La shared Pool de la SGA

- taille en octet définie par le paramètre SHARED_POOL_SIZE
 - 80 Mo sur 10g !
- Sert pour l'allocation de mémoire des requêtes SQL et code PL/SQL
- composée de :
 - « library cache »
 - contient le texte de la requête, le code analysé et un plan d'exécution déterminé par l'optimiseur
 - cache du dictionnaire de données
 - pour l'analyse sémantique de la requête

38

Le cache de données de la SGA

- Objectif :
 - minimiser les coûts d'entrées/sorties entre la mémoire centrale et la mémoire secondaire
- Stocke les blocs *les plus récemment utilisés* (LRU)
 - Une liste LRU ----- MRU
 - Une "dirty list"
- Paramètres
 - nombre de blocks défini par DB_BLOCK_BUFFERS (*plus maintenu*)
 - taille d'un bloc définie par DB_BLOCK_SIZE

39

La zone mémoire du programme (PGA)

- Zone mémoire utilisée par un seul processus serveur
 - à la différence de la SGA qui est partagée par tous les processus serveurs
- PGA contient :
 - une zone de tri
 - des informations sur la session
 - l'état du curseur
 - ...

40

4. Traitement d'un ordre de mise à jour (DML)

- Similaire à une requête d'interrogation pour la phase d'analyse
- Phase d'exécution différente :
 1. le processus serveur lit des blocs de données et de rollback à partir
 - des fichiers de données
 - du buffer cache de données
 2. Si des blocs ne sont pas déjà dans le cache, copie des blocs du disque dans le cache
 3. Mise à jour de *verrou* sur les données
 4. Enregistrement des modifications
 - à apporter au rollback (image avant)
 - et aux données (nouvelles valeurs)
 5. Mêmes opérations dans le *cache de données*
 - Ces blocs sont marqués comme modifiés (différents des blocs stockés sur disque)

41

Rollback Segment

- Permet la gestion des *transactions*
 - COMMIT : valide un ensemble de MAJ
 - ROLLBACK : annule un ensemble de MAJ
- Avant d'effectuer une modification des données, le processus serveur enregistre l'ancienne valeur dans un *rollback segment*
- Intérêts :
 - si la transaction échoue, les modifications "déjà faites" sont annulées
 - Assure une lecture cohérente des données par les autres transactions
 - en cas de panne, les données sont restaurées dans un état cohérent.

42

Cache de reprise

- Redo log = reprise
- Fait partie de la SGA
- Taille définie par LOG_BUFFER
- Stocke la trace des modifications
 - le bloc modifié
 - l'emplacement de la modification
 - la nouvelle valeur
 - les 3 constituent une "entrée redo"
- Pas de distinction sur le type du bloc
 - bloc d'index, de données, ...
- Cache circulaire
 - capacité infinie !

43

Log Writer (LGWR)

- Objectifs
 - Enregistrer *toutes* les modifications apportées aux données afin d'assurer la *reprise après panne*
- Enregistre les modifications du cache de reprise de la SGA *dans les fichiers de reprise*
- Effectue des écritures séquentielles
 - *quand le cache de reprise est **plein à 1/3***
 - *une transaction est validée*
 - *juste avant que DBWR ne fonctionne*
- Comme le cache de reprise, les fichiers de reprise sont organisés de façon circulaire
 - **capacité infinie**

44

Database Writer (DBWR)

- Objectifs :
 - Limiter les coûts d'entrées/sorties en "retardant" (de façon transparente à l'utilisateur) l'écriture des blocs modifiés
- DBWR
 - un des processus serveur en arrière plan
 - enregistre les modifications du cache de données *dans les fichiers de données*
 - Quand ?
 - Nombre de blocs modifiés atteint un seuil
 - pas assez de blocs libres
 - déclenchement d'un point de synchronisation

45

5. Traitement des opérations COMMIT

- SCN pour *System Change Number*
 - à chaque transaction validée -> 1 identifiant de modification SCN
 - sorte d'horodateur interne
 - permet de vérifier la cohérence indépendamment de la date/heure du SE
- Lorsqu'un ordre COMMIT est effectué:
 1. Le processus serveur enregistre dans le cache de reprise 1 SCN
 2. LGWR effectue une lecture contiguë de tout le cache de reprise (SCN inclus) dans les fichiers de reprise
 3. L'utilisateur reçoit "**commit complete**"
 4. Le processus serveur libère les verrous

46

Les outils d'administration

- Dépend du système d'exploitation sur lequel Oracle est installé
 - tendance actuelle : automatiser les tâches d'administration
 - avec des outils graphiques
 - des *assistants*
- Utilisation
 - des composants graphiques
 - du mode ligne de commande
- « Qui peut le plus, peut le moins »
- On favorisera le mode ligne de commande
 - procédure de **déploiement en production**

47

Exemples d'outils

- Mode ligne de commandes
 - Sqlplus sous NT
 - remplace SQL*DBA
 - SQL*Loader
 - import/export
 - oradim, orapwd, ...
- Mode graphique
 - Instance Manager
 - Storage Manager
 - SQL Worksheet
 - Schema Manager
 - Security Manager
 - Backup Manager
 - Data Manager
- Autres outils Oracle :
 - Oracle Net

48

Objectifs

- Décrire l'architecture d'un SGBD
 - Comprendre son interface avec le matériel, le système d'exploitation
- Etre capable de créer son propre serveur de bases de données
- Trois notions importantes
 - Instance
 - Base de données
 - Dictionnaire et packages

Partie 2

Serveur de bases de données

Prerequis

- Disposer d'une machine 😊
 - le serveur de bases de données
 - En être administrateur
 - En fonction du système
 - installer le logiciel Oracle
 - en mode serveur
- Si administration à distance sur un poste client
 - installer le logiciel Oracle Client
- Accessible sur le site :
 - www.oracle.com

Instance Oracle

Objectifs

- Initialisation du système d'exploitation
 - Différents types d'authentification
- Création du *fichier de paramètres*
- Démarrage d'une instance et ouverture de la base de données
- Fermeture de la base de données et arrêt de l'instance
- Obtention et initialisation des paramètres
- Gestion des *sessions*
- Contrôle des fichiers d'alertes et de traces

Instance Oracle

- Préalable obligatoire pour accéder une base de données Oracle
- Instance Oracle
 - associée à une *unique* base de données
 - SGA
 - processus en arrière plan
- Tâches courantes de l'administrateur
 - de démarrer l'instance
 - puis de monter la BD et de l'ouvrir
 - de l'arrêter
 - après avoir fermé puis démonter la BD

Utilisateurs administrateurs

- Utilisateur SYS et SYSTEM
 - créés automatiquement (rôle DBA)
- SYS
 - mot de passe : celui spécifié avec ORADIM ou « change_on_install »
 - propriétaire du dictionnaire
 - relations + vues dans le schéma SYS
 - possède tous les droits
- SYSTEM
 - mot de passe : celui spécifié avec ORADIM ou « manager »
 - moins de privilèges que SYS
- Utilisateur INTERNAL (plus supporté)
 - mot de passe *admin* sous Unix et *oracle* sur Windows

Authentification

- Deux possibilités :
 - laisser Oracle gérer l'authentification
 - télécharger Oracle en se référant au SE
- Authentification par Oracle
 - cf. gestion de la Sécurité
- Authentification par le système d'exploitation :
 - Définir l'utilisateur à être authentifié par le SE pour qu'il ait les privilèges administrateurs
 - *groupe* sous Oracle/Unix
 - *utilisateurs* du groupe
 - cf. doc pour la mise en œuvre
 - 4 cas de figure :
 - utilisateurs sans privilège
 - utilisateurs privilégiés (SYSOPER, SYSDBA)
 - utilisateur sys
 - affecter des *rôles* aux utilisateurs

57

Instance Oracle : Rappel

- SGA (Shared Global Area) ou Zone mémoire globale du système
- contient les données et les informations de contrôle pour le serveur Oracle
- alloué en mémoire virtuelle (gérée par le système d'exploitation du serveur)
- Elle comprend
 - la shared pool : stocke les informations les plus récemment utilisées (e.g. requêtes SQL)
 - le cache de données : stocke les données les plus récemment utilisées
 - le cache de reprise : utilisé pour enregistrer les modifications apportées à la base de données

58

Instance Oracle :

Rappel (2)

5 principaux processus lancés pour une instance Oracle :

- **DBWR** (processus d'écriture des blocs de données modifiés) : *écrit les données modifiées dans la base de données*
- **LGWR** (processus d'écriture des modifications) : *enregistre les modifications enregistrées dans le cache de reprise dans les fichiers de reprise*
- **SMON** (processus System Monitor) : *vérifie la cohérence et redémarre la base de données après panne*
- **PMON** (processus Process Monitor) : *nettoie les ressources lors de l'echec d'un processus*
- **CKPT** : *Synchronise et assure la cohérence des données*

59

Création d'une instance

- Pouvoir se connecter comme administrateur
- Editer le *fichier de paramètres* init<sid>.ora
- Sous sqlplus, commande STARTUP
 - Exemple :
 - connect sys/oracle as sysdba
 - startup nomount
 - file=d:\data\initU01.ora
- Aucune vérification de cohérence à ce niveau
- Allocation de la SGA

60

Instance et Service NT

- serveur Oracle = instance + BD
- Sous NT, à chaque serveur est associé un *service*
- Pour créer un service, il faut utiliser l'outil ORADIM
 - permet d'exécuter plusieurs serveurs Oracle sur la même machine
 - Attention au nombre de caractères
 - < à 4 pour les noms de service (V8)
 - Créer une instance, e.g. nommée IF3
 - ORADIM -NEW -SRVC OracleServiceIF3 [-SYSPWD sys_pwd] [-sid SRVCNAME] [-MAXUSERS NUMBER] [-STARTMODE AUTO, MANUAL] [-PFILE FILENAME]
- Supprimer une instance
 - ORADIM -DELETE -SID IF3

61

Estimer la taille de la SGA

- dépend de la quantité de mémoire vive disponible
- en général, on a :
 - taille SGA = 40% taille RAM
- Taille SGA =
 $db_block_size * db_block_buffer + shared_pool_size + log_buffer$
- Exemple : Soit 500 Mo de RAM disponible => SGA = 200 Mo pour fixer les paramètres, dépend de la version d'Oracle !
 - 10g :
 - $db_cache_size = db_block_size * db_block_buffer,$
 - db_block_buffer plus maintenu

62

Le fichier de paramètres

- **Paramètre obligatoire :**
 - **db_name** : idf de la bd, < à 8 caractères, pas de chiffre comme 1^{er} caractère
- **Paramètres communément modifiés:**
 - **control_files** : nom des fichiers de contrôle
 - **compatible** : version du serveur avec laquelle cette instance est compatible
 - **db_block_buffers** : nombre de blocs mis en cache dans la SGA (min. 50) **plus maintenu 10g !**
 - **shared_pool_size** : taille en octet de la zone de partage (minimum à 3.500.000 avec V8, 70 000 000 avec 10g !)
 - **log_buffer** : Nombre d'octets alloués au cache de reprise dans la SGA
 - **user_dump_dest** : emplacement où les fichiers trace sont créés
 - **processes** : nombre max de processus du SE pouvant se connecter simultanément à une instance
 - **sql_trace** : active ou pas un espion pour chaque session utilisateur
 - **max_dump_file_size** : taille max du fichier de trace
 - **timed_statistics** : active ou pas le minutage dans les fichiers trace
 - **undo_management=MANUAL** ou **AUTO** : délégation ou pas des rollback segments à Oracle (**depuis 9i**)

63

Fichier de paramètres

- Créer un fichier init<sid>.ora à partir d'un fichier existant
- Editer le pour modifier certains paramètres
 - choix difficiles, nombreux paramètres
- Possibilité de créer un fichier binaire avec ordre **CREATE SPFILE**
 - Exemple :

```
CREATE SPFILE='/u01/oracle/dbs/test_spfile.ora'  
FROM PFILE='/u01/oracle/dbs/test_init.ora';
```

(pas possible de spécifier la cible avec la 10.2)

64

Exemple de fichier de paramètres

```
control_files=(D:\oracle\product
\10.2.0\oradata\Mais\control.001.ctl,
D:\oracle\product\10.2.0\oradata\Mais
\control.002.ctl)

db_name          = mais
db_domain        = insa-Lyon.fr
db_block_size    = 4096
shared_pool_size = 100000000
undo_management  = AUTO

# define directories to store trace
# and alert files
#
background_dump_dest=D:\oracle\product
\10.2.0\admin\Mais\bdump
user_dump_dest=D:\oracle\product
\10.2.0\admin\Mais\udump
```

65

Démarrage par étapes

- 3 options lors du démarrage
 - **nomount** : seule l'instance est démarrée
 - **mount** : fichier de contrôle ouvert pour cette instance
 - **open** : tous les fichiers définis pour cette instance dans le fichier de contrôle sont ouverts
- Certaines commandes nécessitent que la base soit dans une de ces options
 - étape NOMOUNT pour la création de la base de données
 - étape MOUNT pour renommer des fichiers de reprise, ...

66

Paramètres d'initialisation dynamiques

- Certains paramètres du fichier init.ora peuvent
 - être modifiés base ouverte
 - la plupart !
 - être modifiés base fermée
 - en éditant le fichier init.ora
 - ex avec undo_management
 - ne jamais être modifiés
 - ex avec db_block_size
- Modification base ouverte
 - Ordre ALTER SYSTEM ou ALTER SESSION
 - Exemples :
 - ALTER SESSION SET SQL_TRACE=true;
 - modifie seulement la valeur pour la session courante
 - ALTER SYSTEM SET TIMED_STATISTICS=true;
 - modification permanente
 - ALTER SYSTEM SET SORT_AREA_SIZE=131072 DEFERRED;
 - modification pour les autres sessions
- Possibilité de modification permanente si fichier binaire de init.ora existe
 - clause SCOPE = SPFILE ou MEMORY

67

Restauration d'instance

- Fait suite à un problème
 - du SE, de l'instance
 - du matériel ...
- le processus **SMON**
 - restaure automatiquement l'instance
 - lorsque l'instance est redémarrée
 - le cache des données est restaurée à partir du fichier de reprise
- les étapes sont :
 - *rolling forward* : restaure les données enregistrées dans les fichiers de reprise mais non enregistrées dans le fichier de données
 - *ouverture de la base de données* (accès possible sur toutes les données non verrouillées)
 - *rolling back* : les transactions non validées sont annulées

68

La commande STARTUP

■ Syntaxe :

```
STARTUP [FORCE] [RESTRICT] [FILE=filename]
        [EXCLUSIVE | PARALLEL | SHARED]
        [OPEN [RECOVER][DATABASE]
         | MOUNT | NOMOUNT]
```

- | *open* : permet aux utilisateurs d'accéder à la base
- | *mount* : monte la base mais accès limités
- | *nomount* : sga + processus arrière plan. Pas d'accès à la base
- | *exclusive* : seule cette instance accède la base
- | *parallel* ou *shared* : avec server // Oracle
- | *force* : annule l'instance courante avant de démarrer normalement
- | *restrict* : accès réservés aux utilisateurs ayant le privilège RESTRICTED SESSION
- | *recover* : démarre la restauration quand la base démarre

La commande SHUTDOWN

■ Syntaxe :

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

| Mode d'arrêt | A | I | T | N |
|--|---|---|---|---|
| nouvelles connexions permises | 0 | 0 | 0 | 0 |
| attend la fin des sessions en cours | 0 | 0 | 0 | 1 |
| attend la fin des transactions en cours | 0 | 0 | 1 | 1 |
| force un point de synchronisation et ferme | 0 | 1 | 1 | 1 |

- Temps d'arrêt : N -> T -> I -> A

Vues dynamiques sur les performances

- Entretienues par le serveur Oracle
 - mises à jour automatiquement
 - image fidèle du serveur
- Décrivent les structures du disque et de la mémoire
- Utiles pour le tuning des performances
- Préfixées par **V\$**
 - définies à partir de tables normalisées préfixées par X\$
- Accessible par l'administrateur à l'étape NOMOUNT
- V\$FIXED_TABLE : les décrit toutes

71

Exemples

- SGA
 - V\$PARAMETER, V\$SGA, V\$OPTION, V\$PROCESS, V\$SESSION, V\$VERSION, V\$INSTANCE
- Fichiers :
 - V\$THREAD, V\$CONTROLFILE, V\$DATABASE, V\$DATAFILE, V\$DATAFILE_HEADER, V\$LOGFILE
- Afficher les valeurs courantes des paramètres
 - SHOW PARAMETER control
 - ou
 - select name from V\$PARAMETER where name like '%control%';

72

BACKUPS, ARCHIVE, RESTAURATION

V\$ARCHIVE
V\$ARCHIVED_LOG (V8)
V\$ARCHIVE_DEST (V8)
V\$BACKUP
V\$BACKUP_CORRUPTION (V8)
V\$BACKUP_DATAFILE (V8)
V\$BACKUP_DEVICE (V8)
V\$BACKUP_PIECE (V8)
V\$BACKUP_REDOLOG (V8)
V\$BACKUP_SET (V8)
V\$DELETED_OBJECT (V8)
V\$RECOVERY_FILE_STATUS
V\$RECOVERY_LOG
V\$RECOVERY_STATUS
V\$RECOVER_FILE

CACHE MANAGEMENT

V\$CACHE (V8)
V\$DB_OBJECT_CACHE
V\$LIBRARYCACHE
V\$ROWCACHE
V\$SUBCACHE (V8)

CONTROL FILES

V\$CONTROLFILE
V
\$CONTROLFILE_RECORD_SECTION (V8)

CURSORS et SQL

V\$OPEN_CURSOR

DATABASES et INSTANCES

V\$SQL
V\$SQLAREA
V\$SQLTEXT
V\$SQLTEXT_WITH_NEWLINES
V\$SQL_BIND_DATA (V8)
V\$SQL_BIND_METADATA (V8)
V\$SQL_CURSOR
V\$SQL_SHARED_MEMORY
V\$ACTIVE_INSTANCES
V\$BGPROCESS
V\$BH (V8)
V\$COMPATIBILITY
V\$COMPATSEG
V\$COPY_CORRUPTION (V8)
V\$DATABASE
V\$DATAFILE
V\$DATAFILE_COPY (V8)
V\$DATAFILE_HEADER (V8)
V\$DBFILE
V\$DBLINK
V\$DB_PIPES
V\$INSTANCE
V\$LICENSE
V\$OFFLINE_RANGE (V8)
V\$OPTION
V\$SGA
V\$SGASTAT
V\$TABLESPACE (V8)
V\$VERSION

SQL*LOADER (Chargement direct)

V\$LOADCSTAT
V\$LOADPSTAT (V8)
V\$LOADSTAT

FIXED Vues

V\$FIXED_TABLE
V\$FIXED_VIEW_DEFINITION
V\$INDEXED_FIXED_COLUMN

I/O

V\$FILESTAT
V\$WAITSTAT

Verrous

V\$BUFFER_POOL (V8)
V\$CACHE_LOCK
V\$CLASS_PING (V8)
V\$DLM_CONVERT_LOCAL (V8)
V\$DLM_CONVERT_REMOTE (V8)
V\$DLM_LATCH (V8)
V\$DLM_MISC (V8)
V\$ENQUEUE_LOCK (V8)
V\$EVENT_NAME
V\$FALSE_PING
V\$FILE_PING (V8)
V\$LATCH
V\$LATCHHOLDER
V\$LATCHNAME
V\$LATCH_CHILDREN

V\$LATCH_MISSES
V\$LATCH_PARENT
V\$LOCK
V\$LOCK_ACTIVITY

V\$LOCK_ELEMENT
V\$LOCKED_OBJECT

V\$LOCKS_WITH_COLLISIONS
V\$PING
V\$RESOURCE
V\$RESOURCE_LIMIT (V8)
V\$TRANSACTION_ENQUEUE (V8)

V\$LOCK
V\$LOCK1

Divers

V\$TIMER
V\$TYPE_SIZE
V\$SEQUENCES (V8)

Serveur MULTI-THREADED et PARALLEL

V\$CIRCUIT
V\$DISPATCHER
V\$DISPATCHER_RATE (V8)
V\$MTS
V\$QUEUE
V\$REQDIST
V\$SHARED_SERVER
V\$THREAD

PERFORMANCE

V\$GLOBAL_TRANSACTION (V8)
V\$OBJECT_DEPENDENCY
V\$SHARED_POOL_RESERVED
V\$SORT_SEGMENT
V\$SORT_USAGE (V8)
V\$STATNAME
V\$SYSSTAT
V\$SYSTEM_CURSOR_CACHE
V\$SYSTEM_EVENT
V\$TRANSACTION

OPTION REQUETE

PARALLELE

V\$EXECUTION
V\$EXECUTION_LOCATION (V8)
V\$PQ_SESSSTAT
V\$PQ_SLAVE
V\$PQ_SYSSSTAT
V\$PQ_TQSTAT

PARAMETRES ORACLE

V\$NLS_PARAMETERS
V\$NLS_VALID_VALUES
V\$PARAMETER
V\$SYSTEM_PARAMETER

REDO LOGS

V\$LOG
V\$LOGFILE
V\$LOGHIST

V\$LOG_HISTORY

ROLLBACK SEGMENTS

V\$ROLLNAME
V\$ROLLSTAT

SECURITE et PRIVILEGES

V\$ENABLEDPRIVS
V\$PWFILE_USERS

SESSIONS

V\$ACCESS
V\$MYSTAT
V\$PROCESS
V\$SESSION
V\$SESSION_CONNECT_INFO
V\$SESSION_CURSOR_CACHE
V\$SESSION_EVENT
V\$SESSION_LONGOPS (V8)
V\$SESSION_OBJECT_CACHE (V8)
V\$SESSION_WAIT
V\$SESSTAT
V\$SESS_IO

Mode

"restricted session"

- Utiliser l'option RESTRICT au démarrage de l'instance
 - l'accès est ainsi limité dès le début
- Utiliser ALTER SYSTEM
 - pour rendre l'accès à une instance limitée
 - exemples :
 - ALTER SYSTEM ENABLE RESTRICTED SESSION
 - ALTER SYSTEM DISABLE RESTRICTED SESSION

Fermeture de sessions

- Identifier la session à fermer
 - interroger la vue V\$SESSION
 - Exemple :
 - | select sid, serial#
 - | from V\$SESSION
 - | where username='SCOTT';
- Supprimer la avec un ordre ALTER SYSTEM
 - Exemple :
 - | ALTER SYSTEM KILL SESSION '7,117';

77

Fichiers de trace

- Remplis par le serveur et les processus en arrière plan
- Tout y est consigné
 - constitue un journal chronologique des messages et des erreurs
- Intérêts :
 - permet de détecter les erreurs internes et les corruptions de blocs
 - surveiller (espionner) les opérations sur la base de données
- Devrait se retrouver dans le dossier :
 - Oracle\10.2.0\admin\IF3\bdump
- Pour les traces de l'utilisateur
 - Oracle\10.2.0\admin\IF3\udump
- Possible d'activer d'autres traces
 - avec ALTER SESSION
 - le paramètre SQL_TRACE

78

Informations sur l'instance

- Interroger la vue V\$INSTANCE
 - utiliser la commande describe
 - faire des requêtes
- commande SHOW

Base de données Oracle

Prérequis

- Un compte autorisé DBA
 - tous les privilèges sur le système d'exploitation
- Une mémoire suffisante pour le démarrage de l'instance
 - SGA
 - exécutable Oracle
 - les processus
- Un espace disque suffisant pour stocker la base de données
 - au moins 1 fichier de données (2 sur 10g)
 - au moins 2 fichiers de reprise
 - au moins 1 fichier de contrôle
- Pour le disque, il faut *estimer* ...

BD Oracle : Rappel

- désignée par un nom DB_NAME
 - souvent le même que celui de l'instance
 - représente les structures physiques des données
- se compose de *fichiers* du SE
 - fichiers de données (>=1 ou 2 😊)
 - stocke le dictionnaire des données, les objets utilisateurs et les « images avant » des données
 - fichiers de reprise (redo log) (>=2)
 - stocke toutes les modifications apportées à la base de données (pour la reconstruire en cas de panne)
 - fichiers de contrôle (>=1)
 - contiennent les informations nécessaires à la mise à jour et à la vérification de l'intégrité des données.
- Autres fichiers :
 - fichier de paramètres, fichier mot de passe, fichiers de reprise archivés

83

Emplacements des fichiers

- Conserver au moins 2 copies du fichier de contrôle sur 2 machines et/ou disques différents
 - => multiplexer
- Multiplexer les fichiers de reprise
- Séparer les fichiers de données afin
 - éviter la fragmentation de la base
 - séparer les données applicatives ou temporaires dans différents tablespaces
 - minimiser les accès disques ou E/S
 - séparer les tables et les index dans différents tablespaces
- OFA (Optimal Flexible Architecture)
 - conçu pour optimiser l'organisation des fichiers Oracle sous Unix
 - Existe aussi sur NT

84

Création d'une base : Remarques

- Sous Unix :
 - Création automatique au cours d'une installation
 - Création manuelle après installation
- Sous Windows :
 - Création automatique au cours d'une installation
 - Création via Oracle Database Assistant ou création manuelle
- En cas de création manuelle, utiliser des *scripts de création (cf TP)*

85

Création manuelle d'une base de données

- Choisir un nom unique pour l'instance et la base de données
- Choisir un jeu de caractères
- Définir les variables du SE
- Préparer le fichier des paramètres
- Créer un fichier mot de passe
- Démarrer l'instance
- Créer la base de données
 - Deux groupes de fichiers logs
 - Deux tablespaces
 - SYSTEM avec éventuellement EXTENT MANAGEMENT LOCAL
 - SYSAUX
- Exécuter les scripts pour générer le dictionnaire de données

86

Démarrage de l'instance et création de la BD

- Se connecter en tant que SYSDBA
- Démarrer l'instance à l'étape NOMOUNT

```
STARTUP NOMOUNT PFILE=d:\data\initU01.ora
```

- Création de la base de données
- Ordre CREATE DATABASE

```
CREATE DATABASE MAIS
USER SYS IDENTIFIED BY adm
USER SYSTEM IDENTIFIED BY adm
LOGFILE
GROUP 1 ('D:\oracle\product\10.2.0\oradata\Mais\log1a.log') SIZE 10M,
GROUP 2 ('D:\oracle\product\10.2.0\oradata\Mais\log2a.log') SIZE 10M
MAXLOGFILES 5
MAXLOGMEMBERS 5
MAXLOGHISTORY 1
MAXDATAFILES 100
MAXINSTANCES 1
CHARACTER SET WE8ISO8859P1
DATAFILE 'D:\oracle\product\10.2.0\oradata\Mais\system01.dbf' SIZE 160M
SYSAUX DATAFILE 'D:\oracle\product\10.2.0\oradata\Mais\sysaux01.dbf'
SIZE 50M;
```

Après la création de la base de données

- La base contient :

- les fichiers de données constituant les tablespaces SYSTEM et SYSAUX
- les fichiers de contrôle et de reprise
- l'utilisateur SYS et SYSTEM
- le rollback segment SYSTEM
- Tables X\$ + vues V\$

- Pour *modifier* la base de données

- ordre ALTER DATABASE
- Exemple pour ouvrir une base de données qui est à l'état NOMOUNT :
 - ALTER DATABASE [MOUNT | OPEN]

- En particulier, elle ne contient pas encore le *dictionnaire de données*

- **NB** : pas de vue V\$ (ni sur le *dictionnaire*) pour la BD oracle

Dictionnaire de données et Packages standards

Objectifs

- Sur le dictionnaire :
 - Construction des vues du dictionnaire de données
 - script **catalog.sql**
 - Utilisation du dictionnaire
 - par exemple, pour l'évaluation sémantique des requêtes
- Sur les packages
 - Préparation de l'environnement PL/SQL à l'aide de scripts administratifs
 - script **catproc.sql**
 - Administration des procédures stockées et des packages

Dictionnaire de données

- Fournit des informations sur
 - la structure logique et physique d'une base de données
 - les noms, définitions, et allocation d'espace des objets des schémas
 - les contraintes d'intégrité
 - les utilisateurs, les privilèges de la base
 - l'audit
- Composé de tables et de vues :
 - Les tables :
 - tables *normalisées*
 - exemple : IND\$, OBJ\$...
 - créées à partir de script (e.g. [sql.bsq](#))
 - Les vues :
 - simplifient l'accès aux tables
 - créées à partir du script [catalog.sql](#)

91

Vues du dictionnaire

- Le préfixe USER
 - USER_XXX : objets qui *appartiennent* à l'utilisateur
 - Exemple :
 - USER_TABLES : contient toutes les tables d'un utilisateur
- Le préfixe ALL
 - ALL_XXX : objets qui sont *accessibles* aux utilisateurs (grâce colonne OWNER)
- Le préfixe DBA
 - DBA_XXX : objets de *toute la base*
 - Accédées par
 - l'administrateur ou
 - un utilisateur ayant le privilège système SELECT ANY TABLE
- Ordre d'inclusion :
 - USER <= ALL <= DBA

92

Exemples de vues

| Vues | Description |
|--|---|
| dictionary dict_columns | Généralités |
| dba_tables dba_objects dba_lobs dba_tab_constraints | Informations sur les objets utilisateurs, e.g. tables, colonnes ... |
| dba_users dba_sys_privs dba_roles | Informations sur les privilèges et les rôles utilisateurs |
| dba_extents dba_free_space dba_segments | Allocation d'espace pour les objets de la base |
| dba_rollback_segs dba_data_files dba_tablespaces | Structure générale de la base |
| dba_audit_trail dba_audit_objects dba_audit_obj_opts | Information sur l'audit |

VUES RELATIVES AUX OBJETS D'UN UTILISATEUR

| | |
|---|--|
| USER_CATALOG ou CAT | Liste des Tables, Vues, Synonymes et séquences que possède l'utilisateur. |
| TABLE_NAME TABLE_TYPE USER_OBJECTS ou OBJ | Liste des Objets appartenant à l'utilisateur. |
| OBJECT_NAME OBJECT_TYPE LAST_DDL_TIME STATUS USER_AUDIT_OBJECT | CREATED TIMESTAMP COMMENTS Trace des commandes exécutées sur des objets appartenant à l'utilisateur. |
| USER_AUDIT_SESSION | Trace des connexions et des déconnexions de l'utilisateur. |
| USER_AUDIT_STATEMENT | Trace des commandes suivantes: GRANT, REVOKE, AUDIT, NOAUDIT, et ALTER SYSTEM |
| USER_TAB_COLUMNS ou COLS | Liste des colonnes des tables, vues ou cluster créés par l'utilisateur |
| USER_COL_GRANTS | Liste des Autorisations sur les colonnes pour lesquelles l'utilisateur est le propriétaire ou a l'autorisation |
| USER_COL_GRANTS_MADE | Liste des Autorisations attribuées par cet utilisateur sur les colonnes ou tables ou vues lui appartenant. |
| USER_CONSTRAINTS | Liste des Contraintes définies sur les tables de l'utilisateur. |
| OWNER CONSTRAINT_TYPE SEARCH_CONDITION R_CONSTRAINT_NAME STATUS | CONSTRAINT_NAME TABLE_NAME R_OWNER DELETE_RULE |

USER_CONS_COLUMNS Listes des Colonnes intervenant dans la définition des contraintes sur les tables de l'utilisateur

USER_DEPENDENCIES Listes des dépendances entre tables ou vues et procédures, fonctions et packages

USER_ERRORS Listes des Erreurs courantes relatives aux procédures fonctions ou corps de package.

USER_IND_COLUMNS Liste des colonnes, des index définis sur les tables de l'utilisateur.

USER_INDEXES ou **IND** Liste des index créés par l'utilisateur

USER_OBJECTS ou **OBJ** Tous ceux appartenant à l'utilisateur

USER_RESOURCE_LIMITS Description des valeurs limites des ressources pour cet utilisateur.

USER_ROLE_GRANTS Liste des rôles attribués à cet utilisateur.

USER_SEQUENCES ou **SEQ** Liste des séquences créées par l'utilisateur

USER_SOURCE Liste des textes des procédures packages, fonctions et corps de package créés par l'utilisateur.

USER_SNAPSHOTS Liste des snapshots créés par l'utilisateur.

USER_SYNONYMS ou **SYN** Liste des synonymes privés créés par l'utilisateur.

USER_SYS_GRANTS Liste des privilèges systèmes attribués à cet utilisateur.

USER_TAB_COMMENTS Liste de commentaires relatifs aux tables et vues de l'utilisateur.

USER_TAB_GRANTS_MADE Liste des Autorisations sur les objets appartenant à l'utilisateur.

USER_TAB_GRANTS_RECDD Liste des objets sur lesquels l'utilisateur a reçu une autorisation.

USER_TABLES ou **TABS** Liste des tables créées par l'utilisateur.

USER_TABLESPACES Liste des tablespaces accessibles à cet utilisateur.

USER_TRIGGERS Liste des triggers créés par l'utilisateur.

USER_TS_QUOTAS Description des quotas de l'espace des tablespaces attribués à l'utilisateur.

USER_USERS Informations sur ce user.

USER_VIEWS Liste des vues créées par cet utilisateur.

CONSTRAINTS_DEFS Liste des définitions de contraintes relatives aux tables accessibles à l'utilisateur.

VUES RELATIVES AUX OBJETS ACCESSIBLES A UN UTILISATEUR

ALL_CATALOG Liste de toutes les Tables, vues, synonymes et séquences accessibles à l'utilisateur.

OWNER_TABLE_NAME
TABLE_TYPE

ALL_COL_COMMENTS Liste des commentaires relatifs aux colonnes des tables et vues accessibles à l'utilisateur.

ALL_COL_GRANTS Liste des colonnes pour lesquelles l'utilisateur est autorisé à accéder.

ALL_CONSTRAINTS Liste des contraintes définies sur les tables accessibles à l'utilisateur.

ALL_CONS_COLUMNS Liste des colonnes intervenant dans la définition des contraintes définies sur les tables accessibles à l'utilisateur.

ALL_DEPENDENCIES Liste des dépendances entre tables ou vues et procédures, fonctions et package, ces objets étant accessibles à l'utilisateur.

ALL_ERRORS Liste des erreurs courantes relatives aux procédures, fonctions et packages, ces objets étant accessibles à l'utilisateur.

ALL_IND_COLUMNS Liste des colonnes des index définis sur les tables accessibles à l'utilisateur.

ALL_INDEXES Liste des index définis sur les tables accessibles à l'utilisateur.

ALL_OBJECTS Liste des objets accessibles à l'utilisateur.
OWNER OBJECT_NAME
OBJECT_ID OBJECT_TYPE
CREATEDLAST_DDL_TIME
TIMESTAMP STATUS

ALL_SEQUENCES Liste des séquences accessibles à l'utilisateur.

ALL_SOURCE Liste des textes de procédures, fonctions et corps de package accessibles à l'utilisateur.

VUES RELATIVES AUX ADMINISTRATEURS

| | |
|----------------------|--|
| ALL_SNAPSHOTS | Liste des snapshots accessibles à l'utilisateur. |
| ALL_SYNONYMS | Liste des synonymes accessibles à l'utilisateur. |
| ALL_TAB_COLUMNS | Liste des colonnes, des tables, vues ou clusters accessibles à l'utilisateur. |
| ALL_TAB_COMMENTS | Liste des commentaires relatifs aux tables et vues accessibles à l'utilisateur. |
| ALL_TAB_GRANTS | Liste des autorisations sur les tables ou vues pour lesquelles l'utilisateur est autorisé explicitement. |
| ALL_TAB_GRANTS_MADE | Liste des autorisations attribuées ou reçues par l'utilisateur. |
| ALL_TAB_GRANTS_RECCD | Liste des objets sur lesquels l'utilisateur a reçu une autorisation. |
| ALL_TABLES | Liste des tables accessibles à l'utilisateur. |
| ALL_TRIGGERS | Liste des triggers accessibles à un utilisateur. |
| ALL_USERS | Liste des Informations sur les utilisateurs. |
| ALL_VIEWS | Liste des vues accessibles à l'utilisateur. |

97

| | |
|---------------------|--|
| DBA_2PC_PENDING | Liste des transactions réparées ayant échoué lors de la phase de préparation. |
| DBA_2PC_NEIGHBORS | Connexions vers ou à partir de bases distantes pour les traitements des transactions réparées. |
| DBA_AUDIT_EXISTS | Trace de la commande AUDIT EXISTS. |
| DBA_AUDIT_OBJECT | Trace les commandes exécutées sur tous les objets de la base. |
| DBA_AUDIT_SESSION | Trace les connexions et des déconnexions de tous les utilisateurs. |
| DBA_AUDIT_STATEMENT | Trace les commandes suivantes: GRANT, REVOKE, AUDIT, NOAUDIT, et ALTER SYSTEM par tous les utilisateurs. |
| DBA_CATALOG | Liste des Tables, Vues, Synonymes et séquence de la base. |
| DBA_CLUSTERS | Liste de tous les clusters de la base. |
| DBA_CLU_COLUMNS | Description de la correspondance entre colonnes d'une table et colonnes d'un cluster. |
| DBA_COL_COMMENTS | Liste des commentaires relatifs aux colonnes de toutes les tables et vues de la base. |
| DBA_COL_GRANTS | Liste des Autorisations sur toutes les colonnes de la base. |
| DBA_CONSTRAINTS | Liste de toutes les contraintes définies sur les tables. |
| DBA_CONS_COLUMNS | Liste des colonnes intervenant dans la définition de toutes les contraintes de la base. |
| DBA_DATA_FILES | Description de tous les fichiers de la base. |
| DBA_DB_LINKS | Liste de tous les liens de la base. |
| DBA_DEPENDENCIES | Liste des dépendances entre tables ou vues et procédures, fonctions et packages de la base. |
| DBA_ERRORS | Liste des erreurs courantes relatives aux procédures fonctions ou corps de package de la base. |
| DBA_EXP_FILES | Description des fichiers d'exportation. |
| DBA_EXP_OBJECTS | Description des objets ayant été importés d'une façon incrémentale. |

98

| | |
|-----------------------|--|
| DBA_EXP_VERSION | Numéro de version de la dernière session d'importation. |
| DBA_EXTENTS | Liste des extensions de tous les tablespaces de la base. |
| DBA_FREE_SPACE | Description des extensions libres dans tous les tablespaces de la base. |
| DBA_IND_COLUMNS | Liste des colonnes de tous les index définis sur toutes les tables de la base. |
| DBA_INDEXES | Liste des index de la base. |
| DBA_OBJECT_AUDIT_OPTS | Liste des options d'audit pour toutes les tables et vues de la base. |
| DBA_OBJECTS | Liste de tous les objets de la base. |
| DBA_PROFILES | Liste de s profils attribués aux utilisateurs. |
| DBA_ROLE_GRANTS | Liste des rôles attribués aux utilisateurs, et à d'autres rôles. |
| DBA_ROLES | Liste de tous les rôles dans la base. |
| DBA_ROLLBACK_SEGS | Liste des segments d'annulation de la base |
| DBA_SEGMENTS | Description de tous les paramètres de stockage de tous les segments de la base. |
| DBA_SEQUENCES | Liste de toutes les séquences de la base. |
| DBA_SOURCE | Liste des textes de toutes les procédures, packages, fonctions et corps de package |
| DBA_SNAPSHOTS | Liste de tous les snapshots de la base. |
| DBA_SNAPSHOTS_LOGS | Liste de tous les snapshots logs dans la base. |
| DBA_SYNONYMS | Liste de tous les synonymes de la base. |
| DBA_SYS_GRANTS | Liste des privilèges systèmes attribués aux utilisateurs et aux rôles. |
| DBA_TAB_COLUMNS | Liste des colonnes des tables, vues ou cluster |
| DBA_TAB_COMMENTS | Liste des commentaires relatif aux tables et vues |
| DBA_TAB_GRANTS | Liste d'autorisations sur tous les objets de la base. |
| DBA_TABLES | Liste des tables de la base. |
| DBA_TABLESPACES | Liste des tablespaces de a base. |
| DBA_TRIGGERERS | Liste des triggers de la base. |
| DBA_TS_QUOTAS | Description des quotas de l'espace des tablespaces |
| DBA_USERS | Informations sur tous les utilisateurs. |
| DBA_VIEWS | Liste des vues de la base. |

VUES DYNAMIQUES

| | |
|-------------------|--|
| V\$ACCESS | Liste des objets verrouillés actuellement. |
| V\$BGPROCESS | Description de l'activité des processus d'arrière plan. |
| V\$DATABASE | Description de la base de données à partir du fichier de contrôle. |
| V\$DATAFILE | Description de fichiers de données à partir du fichier de contrôle. |
| V\$DISPATCHERS | Description de l'activité des processus dispatchers. |
| V\$FILESTAT | Statistiques sur le opérations de lecture/écriture sur un fichier. |
| V\$LOGFILE | Description des fichiers de reprise à partir du fichier de contrôle. |
| V\$PARAMETER | Description des valeurs actuelles des paramètres d'initialisation. |
| V\$PROCESS | Description des processus actifs. |
| V\$SESSION | Description des sessions courantes. |
| V\$SGA | Description sommaire de la SGA. |
| V\$GASTAT | Description détaillée de la SGA. |
| V\$SHARED_SERVERS | Description des serveurs partagés. |
| V\$TRANSACTION | Description des transactions en cours. |

Scripts administratifs

- Information catalogue et dictionnaire de données
 - cat*.sql
 - exemple : catrman.sql pour l'utilitaire Recovery Manager
- Spécification des packages de la base
 - dbms*.sql
- Code compressé d'un package base de données
 - prvt*.plb
- Vues et tables pour les utilitaires de la base
 - utl*.sql
 - exemple : utlsampl.sql pour tables de démonstration

101

Exemples de packages fournis par Oracle

- DBMS_LOB
 - opérations sur type BLOB et CLOB
- DBMS_SESSION
 - génère commandes SQL tq
 - ALTER SESSION o u SET ROLE
- DBMS_UTILITY
 - routines utilitaires
- DBMS_SPACE
 - informations relatives à la disponibilité de l'espace des segments.
- DBMS_ROWID
 - fournit les informations sur le ROWID
- DBMS_SHARED_POOL
 - conserve et retire des objets dans la SHARED_POOL

102

Requêtes sur les objets stockés

- Interroger la vue du dictionnaire DBA_OBJECTS
 - colonnes OWNER, OBJECT_NAME, OBJECT_TYPE, STATUS (valid ou invalid)
 - exemple :
 - | select object_name, object_type, status
 - | from dba_objects
 - | where object_name like 'DBMS_%';

ou

- commande DESCRIBE
 - exemples :
 - | describe dbms_session
 - | describe dbms_session.set_role

Partie 3



Structure physique de stockage

105

Fichiers de contrôle



106

Objectifs

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

107

Utilisation du fichier de contrôle

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

108

Contenu du fichier de contrôle

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

109

Détails du fichier de contrôle

- Paramètres influençant sa taille :
 - MAXLOGFILES
 - MAXLOGMEMBERS
 - MAXLOGHISTORY
 - MAXDATAFILES
 - MAXINSTANCES
- Requêtes sur le fichier de contrôle
 - V\$CONTROLFILE
 - NAME
 - V\$PARAMETER
 - NAME
 - VALUE
 - V\$CONTROLFILE_RECORD_SECTION (V8)
 - TYPE
 - RECORDS_SIZE
 - RECORDS_TOTAL
 - RECORDS_USED

110

Multipléxage du fichier de contrôle

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

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

Fichiers de reprise

ou redo log files ou journaux

113

Objectifs

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

114

Utilisation des fichiers de reprise (rappel)

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

115

Groupes et membres de fichiers de reprise

- **Groupes** de fichiers de reprise :
 - ensemble de copies identiques de fichiers de reprise *online*
- Un **membre** est un élément d'un groupe, i.e. un fichier de reprise
- LGWR écrit *simultanément* sur chaque membre du groupe
 - tous les membres d'un groupe possèdent un *numéro de séquence log*
 - sert d'identifiant
 - le numéro courant est stocké dans le fichier de contrôle
- Les membres d'un groupe sur des disques différents
- au moins 2 groupes

116

Initialisation des fichiers de reprise

- Générés lors de la création de la base de données
- Paramètres
 - MAXLOGFILES : nombre max de groupe dans une base de données (<255)
 - MAXLOGMEMBERS : nombre max de membre par groupe
 - LOG_FILES (dans init<sid>.ora) : nombre max de fichiers de reprise à ouvrir :
 - doit être inférieur à MAXLOGFILES * MAXLOGMEMBERS

117

Rappel

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

118

Point de synchronisation

- Un point de synchronisation se produit:
 - quand LGWR change de groupe courant
 - quand une instance s'arrête (N,T ou I)
 - avec les paramètres d'initialisation
 - | LOG_CHECKPOINT_INTERVAL
 - | LOG_CHECKPOINT_TIMEOUT
 - commande de l'administrateur
 - | ALTER SYSTEM SWITCH LOGFILE
 - | ALTER SYSTEM CHECKPOINT GLOBAL
 - plus rapide, n'implique pas de switch logfile
 - Sauvegarde d'un tablespace activé ou quand un tablespace est désactivé
 - Actions faites :
 - tous les blocs modifiés du cache de données *qui sont aussi traités par le cache de reprise* sont écrits par DBWR
 - le processus en arrière plan CKPT met à jour les fichiers de contrôle pour signaler son aboutissement

119

Archivage

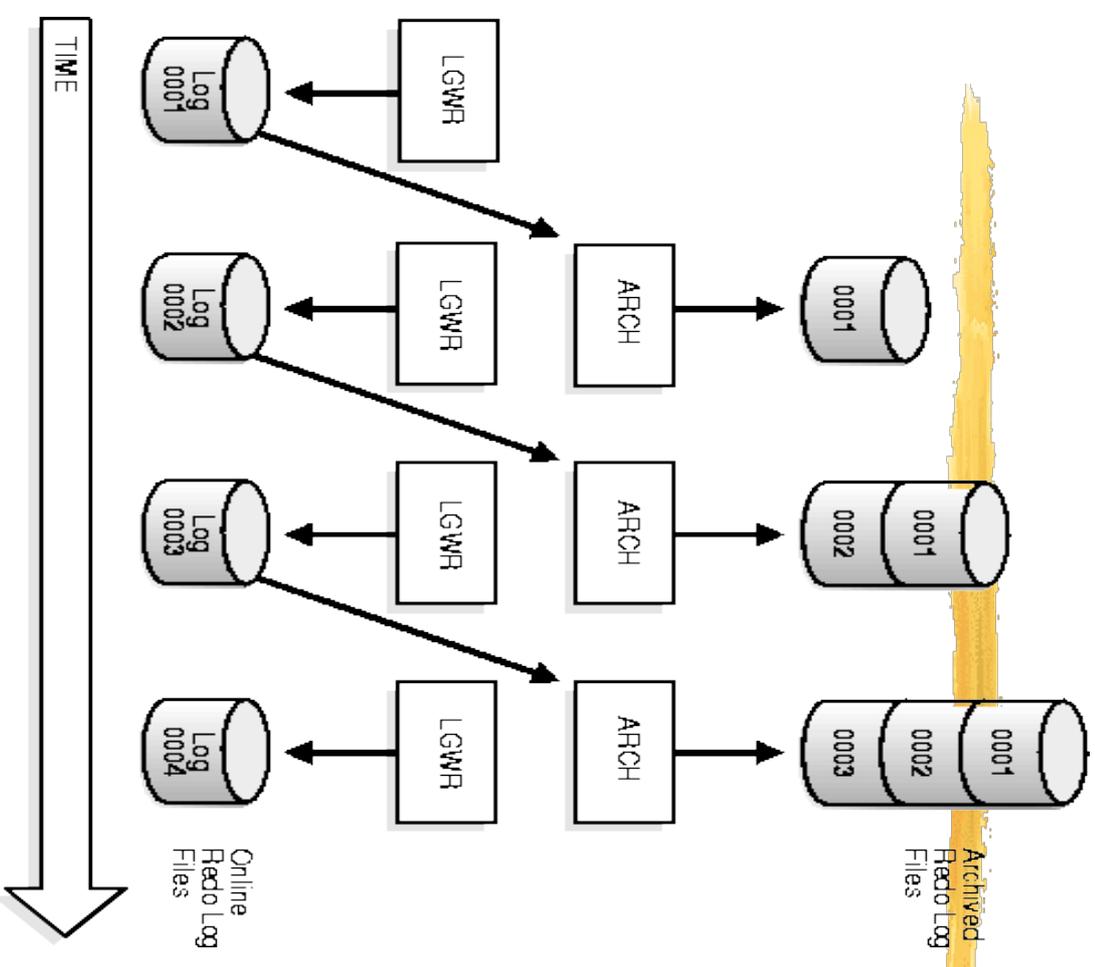
- Dépend de la politique locale vis à vis des données, i.e.
 - Une perte des données peut-elle être envisagée ?
 - Oui ? Non ?
 - Si oui, quelle est la durée maximale tolérable ?
 - 1 jour ? 1 semaine ?
- Décision importante de l'administrateur
 - | mode ARCHIVELOG
 - fichiers de reprise sont archivés
 - permet de garder **toutes** les modifications de la base
 - | mode NOARCHIVELOG
 - fichiers de reprise sont **écrasés** (quand une boucle est faite)
 - On accepte de perdre en cas de panne des modifications
- Distinction entre les fichiers de reprise *online* et *offline*

120

Archivage des fichiers de reprise

- Assure une cohérence parfaite des données en cas de panne
- Processus en arrière plan ARCH
 - déclenché à chaque basculement
- Archivage
 - soit manuel
 - c'est à l'administrateur de déclencher ou pas un archivage des fichiers de reprise
 - ALTER SYSTEM ARCHIVE LOG
 - pas conseillé par Oracle
 - soit automatique
 - paramètre de l'instance LOG_ARCHIVE_START=true
 - dans init.ora, voir aussi paramètres :
 - LOG_ARCHIVE_DEST
 - LOG_ARCHIVE_FORMAT
- possibilité de les mutiplexer

121



122

Requêtes sur les fichiers de reprise

- Sur l'archivage :
 - Sous SQLPlus :
 - | ARCHIVE LOG LIST
 - Interroger les vues
 - | V\$DATABASE
 - name, log_mode
 - | V\$INSTANCE
 - archiver
 - | V\$LOG_HISTORY, V\$ARCHIVED_LOG, V\$ARCHIVE_DEST
 - sur les groupes
 - | V\$THREAD
 - groups, current_group#, sequence#
 - Sur les membres
 - | V\$LOG
 - group#, members, status, sequence#, bytes

123

Modification de fichiers de reprise

- Ajout de groupes
 - ALTER DATABASE ADD LOGFILE ('D:\data\DISK3\log3a.rdo', 'D:\data\DISK4\log3b.rdo') size 150 K;
- Ajout de membres
 - ALTER DATABASE ADD LOGFILE MEMBER 'D:\data\DISK3\log1b.rdo' TO GROUP 1, 'D:\data\DISK3\log2b.rdo' TO GROUP 2;
- Déplacement des fichiers de reprise
 1. Arrêter la base de données
 2. Copier les fichiers de reprise au nouvel emplacement
 3. Monter la base de données
 4. Exécuter un ordre
ALTER DATABASE [db] RENAME FILE [file] [, 'file'] ... TO [file] [, 'file'] ...
 5. Ouvrir la base de données
 - (6. supprimer l'ancien fichier)

124

Suppression

- Suppression de *groupes* de fichiers de reprise

- ┆ ALTER DATABASE DROP LOGFILE
GROUP 3

- Suppression de *membres* fichiers de reprise

- ┆ ALTER DATABASE DROP LOGFILE
MEMBER 'D:\data\DISK3\log2b.rdo'

- ┆ Le groupe doit être *inactif*
- ┆ un *autre membre* du groupe doit exister
- ┆ Si mode ARCHIVELOG et que le groupe du membre n'a pas été archivé, opération non permise.

- Suppression des contenus des fichiers

- ┆ ALTER DATABASE CLEAR LOGFILE
'D:\data\DISK3\log2a.rdo'

Partie 4

Structure logique de stockage

127

Objectifs

- Faire le lien entre structure physique et logique
 - via les fichiers de données
- Comprendre les moyens logiques de structuration
 - Segment
- Etude de deux segments particuliers
 - Segment rollback
 - Segment temporaires

128

Fichiers de données et tablespaces

Objectifs

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

Généralités

Base de données Oracle

- structure physique
 - fichier de contrôle
 - fichiers de reprise
 - fichiers de données
 - structure logique
 - tablespace
 - segment
 - extent
 - bloc
- => Nécessaire de comprendre ces 2 niveaux de représentation

131

Utilisation du tablespace

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

132

Tablespaces

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

133

Fichiers de données

- Un ou plusieurs fichiers de données sont associés à un tablespace
 - un fichier de données ne peut appartenir qu'à un tablespace
 - un serveur Oracle crée toujours *un peu plus* d'espace disque que ce qui est spécifié
- Informations quantitatives :
 - nombre maximum de fichier par tablespace = 1023
 - nombre maximum de tablespace par base de données : 64 000

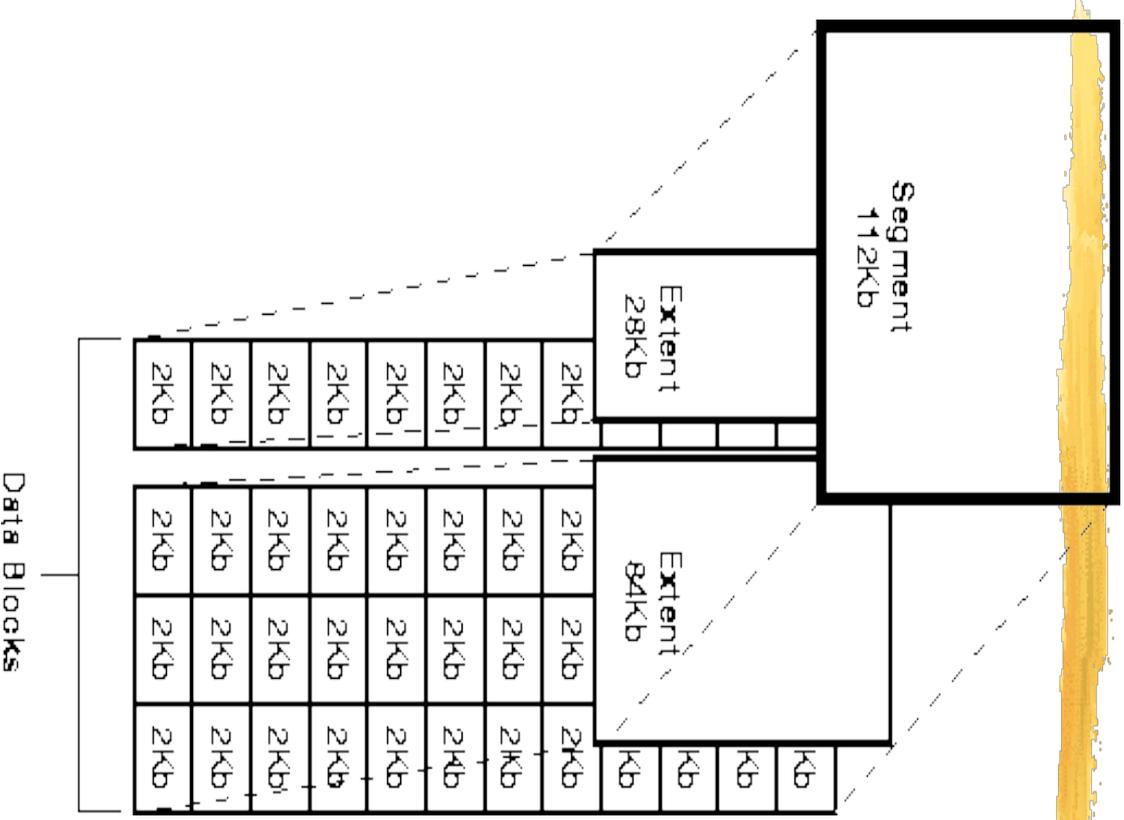
134

Lien segments/extents/ blocs

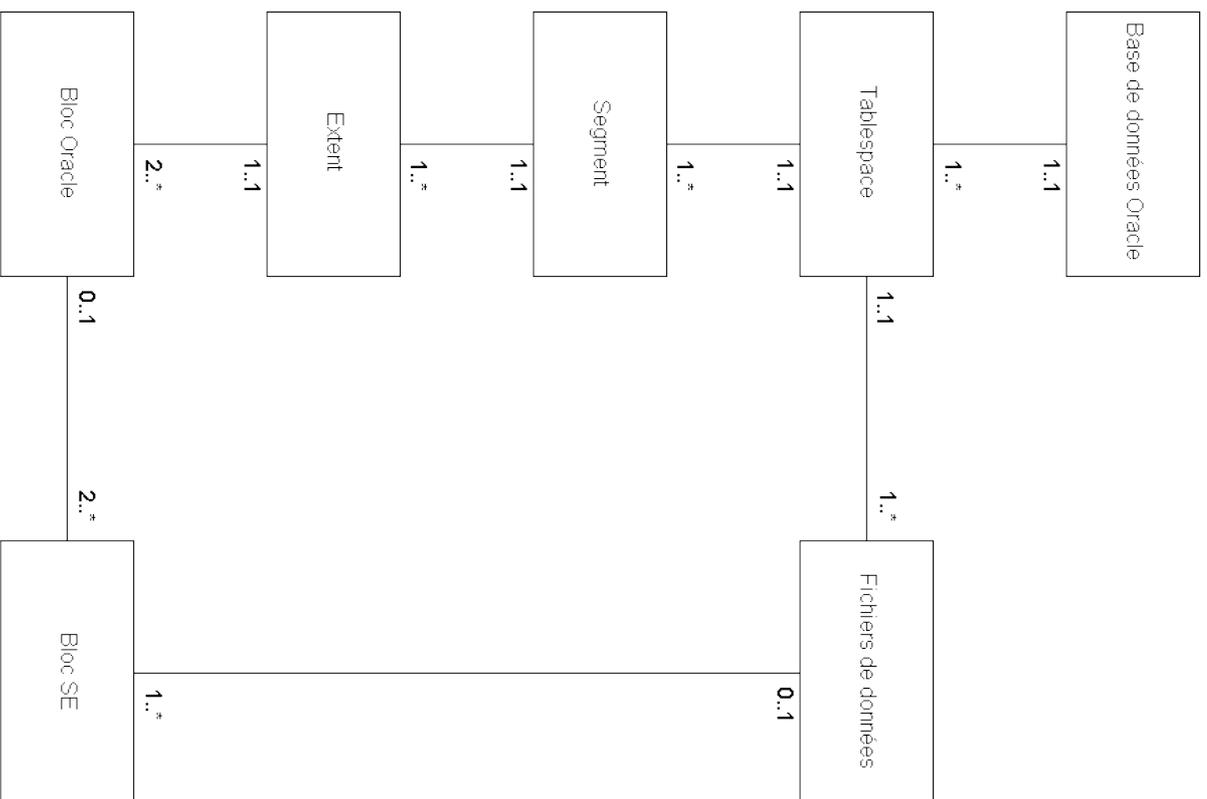
- Segments
 - Espace alloué à un type spécifique de structure logique de stockage dans un tablespace
 - segments de table, segments d'index, segment temporaire, rollback segment
- Extents
 - Ensemble de blocs contigus.
 - Chaque type de segment est composé d'un ou plusieurs extents
- Blocs de données
 - contient un ou plusieurs blocs de fichier physique alloué à partir d'un fichier de données existant
 - plus petite unité d'I/E/S
 - sa taille vaut DB_BLOCK_SIZE

135

Structure logique



136



Structure logique
de stockage

Structure physique
de stockage

137

Tablespace SYSTEM et non-SYSTEM

- Tablespace SYSTEM contient:
 - informations relatives au dictionnaire de données
 - Un rollback segment SYSTEM
- Tablespace non-SYSTEM contient:
 - les rollback segments
 - les segments temporaires
 - les données applicatives
 - les indexes applicatifs
- S'il en existe un seul, c'est lui !
- **Ne devrait pas contenir de données utilisateurs**
- Depuis 10 g, tablespace SYSAUX est nécessaire
- Introduisent plus de souplesse dans l'administration

138

Syntaxe

- CREATE TABLESPACE nom
DATAFILE file SIZE integer [K|M]
[autoextend_clause]
[, file [autoextend_clause]
[MINIMUM EXTENT integer [K|M]]]
[DEFAULT storage_clause]
[PERMANENT | TEMPORARY]
[ONLINE | OFFLINE]
- storage_clause ::= STORAGE (
[INITIAL integer [K|M]]
[NEXT integer [K|M]]
[MINEXTENTS integer]
[MAXEXTENTS [integer | UNLIMITED]
[PCTINCREASE integer])
- autoextend_clause ::= AUTOEXTEND
[OFF | ON
[NEXT integer [K|M]]
[MAXSIZE
[UNLIMITED | integer [K|M]]]]

139

Paramètres de stockage

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

140

Tablespaces temporaires

- Utilisé pour les opérations de tri
 - par les segments de tri (= segments temporaires)
- Ne peut pas contenir d'objets permanents

- Exemple :

```
CREATE TEMPORARY TABLESPACE tri
TEMPFILE '/DISK2/sort01' SIZE 50M
MINIMUM EXTENT 1M
DEFAULT STORAGE (
    INITIAL 2M
    NEXT 2M
    MAXEXTENTS 500
    PCTINCREASE 0)
;
```

141

Augmenter la capacité de stockage

- Deux façons de procéder :
 - Ajouter des fichiers de données au tablespace
 - ALTER TABLESPACE app_data
ADD DATAFILE
'd:\data\DISK5\app03.dbf' SIZE 200M;

- Modifier la taille des fichiers de données existants

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

- Manuellement avec la commande ALTER DATABASE

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

142

Statut OFFLINE

- Les données d'un tablespace offline ne sont plus accessibles
 - L'administrateur peut sauvegarder cette partie des données sans arrêter le fonctionnement du reste de la base
- Ne peuvent jamais être mis offline
 - le tablespace SYSTEM
 - tout tablespace avec des rollbacks segments actifs
- Exemple :
 - ALTER TABLESPACE app_data OFFLINE;

143

Déplacement de fichiers de données

- 1) BD ouverte
 - Le tablespace concerné doit être offline
 - Les fichiers de données cibles doivent exister (commandes SE)
 - Avec ALTER TABLESPACE :

```
ALTER TABLESPACE app_data
RENAME DATAFILE
'd:\data\ DISK4\app01.dbf' TO 'd:\data
\DISK5\app01.dbf'
```
- 2) BD fermée
 - Arrêter puis monter la base de données
 - Utiliser l'ordre ALTER DATABASE

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

144

Suppression d'un tablespace

■ Syntaxe :

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

- INCLUDING CONTENTS : supprime tous les segments du tablespace
- CASCADE CONSTRAINTS : supprime les contraintes d'intégrité référentielle des tables définies dans un autre tablespace qui référencent une table du tablespace à supprimer
- Un tablespace contenant encore des données ne peut être supprimé sans l'option INCLUDING CONTENTS
- Ne supprime que les pointeurs de fichiers du fichier de contrôle
=> les fichiers de données existent toujours

145

Informations sur les tablespaces

■ Requêtes sur la vue DBA_TABLESPACES

- TABLESPACE_NAME
- NEXT_EXTENT
- MAX_EXTENTS
- PCT_INCREASE
- MIN_EXTLEN
- STATUS
- CONTENTS

■ Informations sur les fichiers de données

Requêtes sur la vue DBA_DATA_FILES

- FILE_NAME
- TABLESPACE_NAME
- BYTES
- AUTOEXTENSIBLE
- MAXBYTES
- INCREMENT_BY

146

Informations (2)

- Informations sur les fichiers de données et les tablespaces à partir du *fichier de contrôle*
- Jointure entre V\$DATAFILE et V\$TABLESPACE sur l'attribut TS#
- A retenir :
 - utiliser de nombreux tablespaces
 - assigner des quotas aux utilisateurs

147

Quelques conseils

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

148

Notions sur les segments

Segment de table

- Rappels de terminologie
 - BD = ensemble de tables
 - table = ensemble de tuples
 - tuple = ensemble de valeurs
- Comment stocker les tuples sur disque?
 - Certes dans des fichiers, mais comment ?
 - Suivant quelle organisation ?

- Quelques rappels de fondamentaux ... vus en cours de BD

Codage des tuples

- Organisation d'un tuple dans un bloc

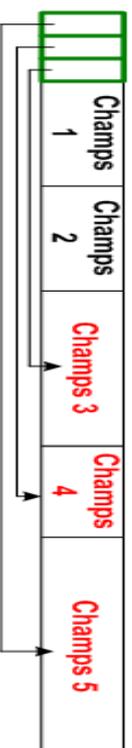


- 2 solutions pour ranger les tuples dans les blocs :

- Chaque champs variable est précédé de sa taille



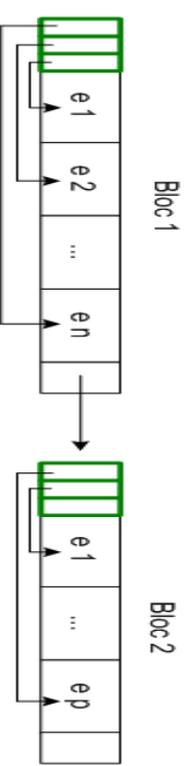
- En tête de tuple, *table d'accès* aux champs



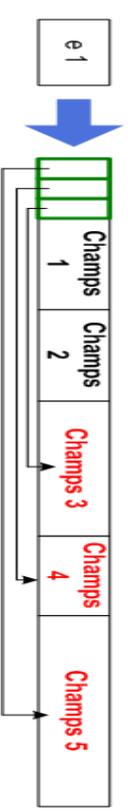
Codage d'un bloc

- Un bloc contient :

- des tuples (e.g. e_1, \dots, e_n)
- un catalogue d'accès aux tuples
- un pointeur vers le bloc suivant

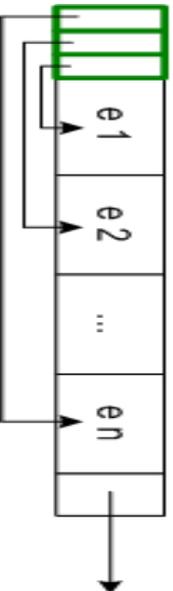


- Or, un tuple :



Parcours de table

- Notion de **blocs**
 - Plus petite unité de transfert entre MV et MS
 - Le temps de recherche d'un tuple de clef k dans un bloc B est négligeable par rapport au temps de chargement du bloc
 - Hypothèses courantes :
 - probabilité « fixe » qu'un bloc requis soit déjà en mémoire
 - 1 accès disque = Temps de recherche + Temps de transfert = Temps constant
- Si on sait où se trouve un tuple, alors il est désigné par : <clef, @bloc>



- Comment limiter le nombre de chargement de bloc du disque ? Comment limiter les accès disque ?

153

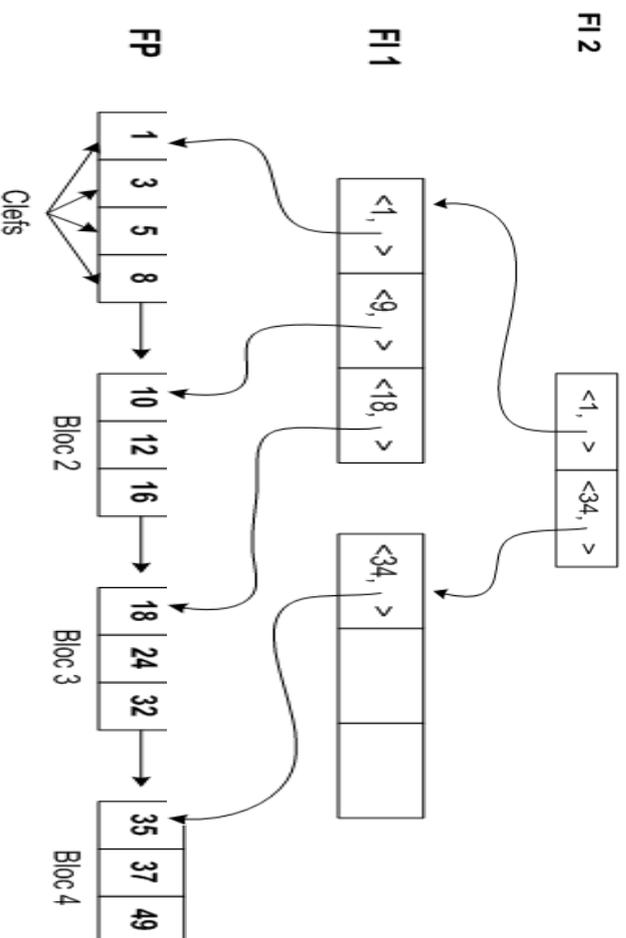
Exemple de recherche par la clé

- Tuples rangés dans un ordre quelconque (tas)
 - Soit $n = 10^6$ tuples
 - 1 tuple = 200 octets
 - Taille d'un bloc = 4096 – 50 réservés = 4046 octets
 - 1 accès bloc = 1/100 s
- Au max : $4046 \times 200 = E = 20$ tuples / bloc
- Nombre min de bloc : $10^6 / 20 = 50000$ blocs
- Chercher un tuple de clef K ?
 - => peut provoquer 50000 accès bloc
 - Au mieux : 1 accès bloc = 1/100 s**
 - Au pire : 50000/100 s ~ 8 min**

154

Codage implicite avec des B-Arbres

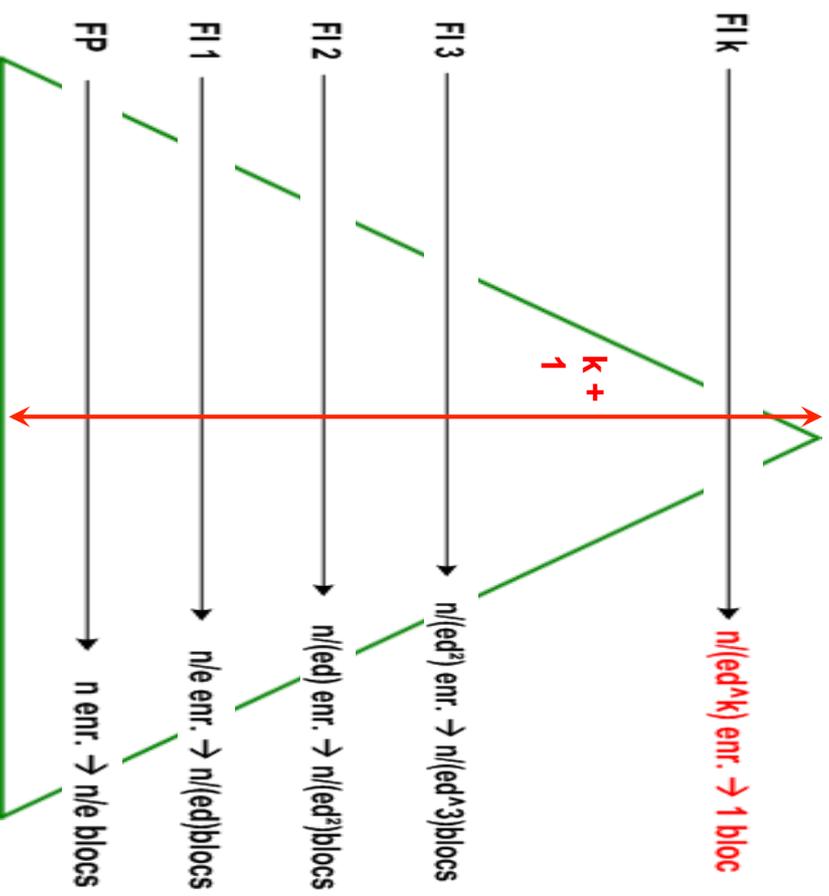
- Construire une table d'accès aux ≠ blocs



Les B-Arbres : formules de calcul

- Un tuple étant sous la forme : $\langle \text{clef}, @\text{bloc} \rangle$, on note :
 - e le plus petit nombre de tuples dans un bloc
 - d le petit nombre de tuples d'index dans un bloc
- Calcul de temps d'accès
 - Chaque bloc de FP contient entre : **e** et **2*e + 1** tuples
 - Chaque bloc de FI contient entre **d** et **2*d-1** enregistrements
- **Taille(Bloc) / Taille(enr.FI) = 2d+1**
- **# tuples par bloc de FP = 2*e-1**

Suite



157

Retour sur l'exemple

Soit $n = 10^6$ tuples

1 tuple = 200 octets

Taille d'un bloc = 4096 - 50 réservés = 4046 octets

On considère qu'1 accès bloc = 1/100 s

Taille(Clef + Pointeur) = taille tuple d'index FI = 24 octets

Temps d'accès à un tuple ?

→ $2 * e + 1 = \# \text{ tuples par bloc} = 4046 / 200 = 20$

→ $e = 9$

→ $2 * d + 1 = \text{Taille(Bloc)} / \text{Taille(Enr. FI)} = 4046 / 24$

→ $d = 83$

FP : 10^6 tuples de données → $10^6 / e$ blocs = 10^5 blocs

FI1 : 10^5 tuples d'index → $10^5 / d$ blocs = 13400 blocs

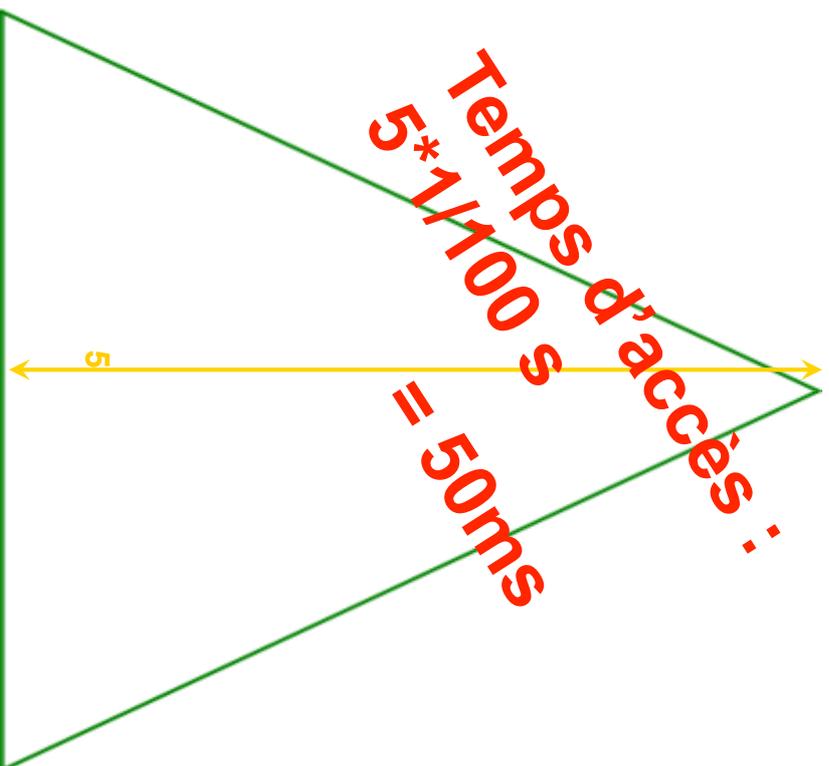
FI2 : 13000 tuples d'index → $13000 / d$ blocs = 160 blocs

FI3 : 160 tuples d'index → $160 / d$ blocs = 2 blocs

FI4 : 2 tuples d'index → 1 bloc !!!

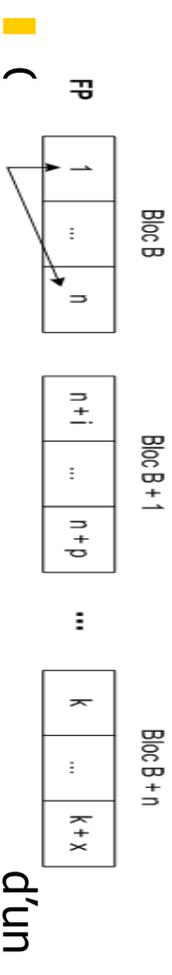
158

Suite et fin

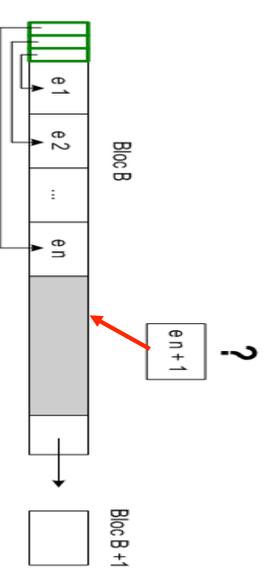


159

Mise à jour des blocs



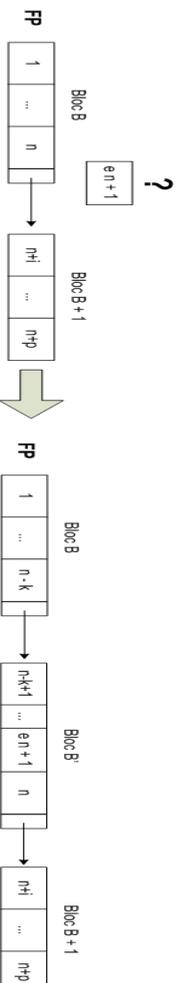
- On connaît la clef du tuple → on sait dans quel bloc il doit aller ! → on l'insère dedans !



160

suite

- Qui ! Mais si le bloc B est plein ?
 - On négocie avec les blocs voisins !
 - Déplacement d' tuples vers le bloc suivant (B + 1) ou le bloc précédent (B - 1) pour insertion !
 - Si bloc (B + 1) et bloc (B - 1) sont pleins ?
 - On crée un nouveau bloc B' et on rechaîne la liste des blocs
 - On répartit les tuples : $B + B' = B + \text{tuple } Y$
 - On insère un tuple dans FI (pour B)



Taille des données et des indexes

- **Autant de place prise par les données que par les indexes**
 - Hypothèse sous jacente : recherche par la clé, i.e. requêtes SQL avec des conditions « Clé = valeur » dans la clause WHERE
- Intimement lié au procédure de stockage des tuples
 - Notion de **blocs**
 - Minimiser le nombre d'E/S de blocs
 - Gérer les éventuels MAJ sur les tuples
 - Ne pas avoir trop de blocs pleins
 - Évite le rechaînage

Avec Oracle

- DB_BLOCK_SIZE : fixe la taille des blocs à la création de la base de données
 - **Ne peut pas être modifié !**
- 2 paramètres sous Oracle pour les éventuelles MAJ :
 - PCTFREE : spécifie à partir de quand un bloc libre est considéré comme plein
 - PCTUSED : spécifie à partir de quand un bloc plein est considéré comme libre
- Voir plus loin

Types de segments avec Oracle

- **Table** (ou *table non-clusterisée* ou *table non-partitionnée*)
 - ressource la plus courante pour stocker les données
 - données stockées sans ordre particulier
 - on a peu de contrôle sur l'emplacement des *lignes* dans les *blocs d'une table*.
 - Appartient à un seul tablespace
- **Partition de table** (V8 + option Partitioning)
 - Partition de la table suivant des clés
 - stockage des partitions dans plusieurs tablespaces
 - une partition est un segment
 - *Intérêts* : objets utilisés souvent et de façon simultanée

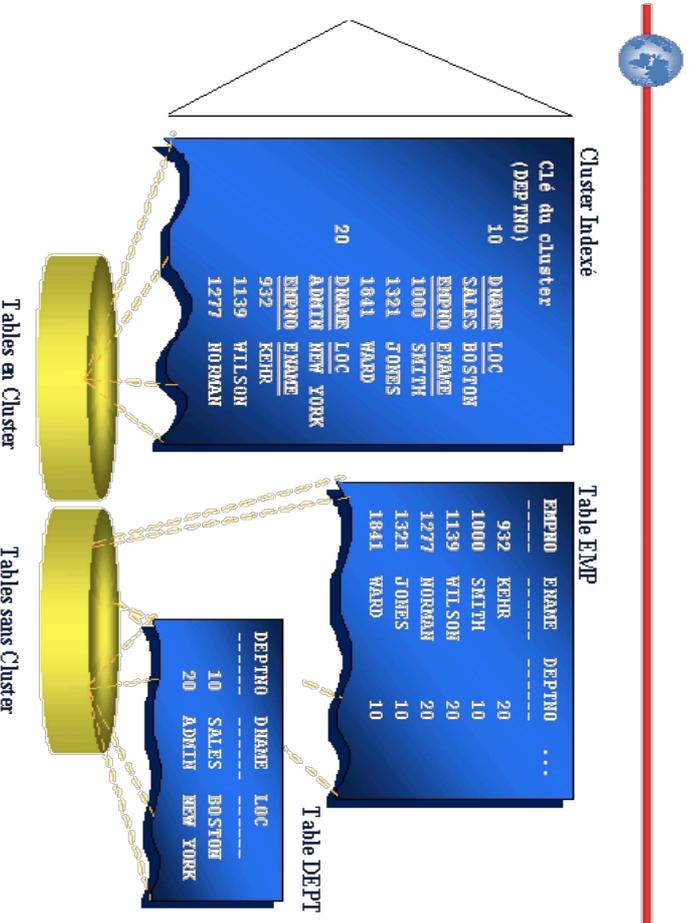
Types de segments (2)

Cluster

- les lignes d'un cluster sont stockées suivant une valeur clé de colonne
- un cluster contient une ou plusieurs tables
- le cluster et les tables qui y participent appartiennent au même segment
- Intérêts* : accélère les jointures

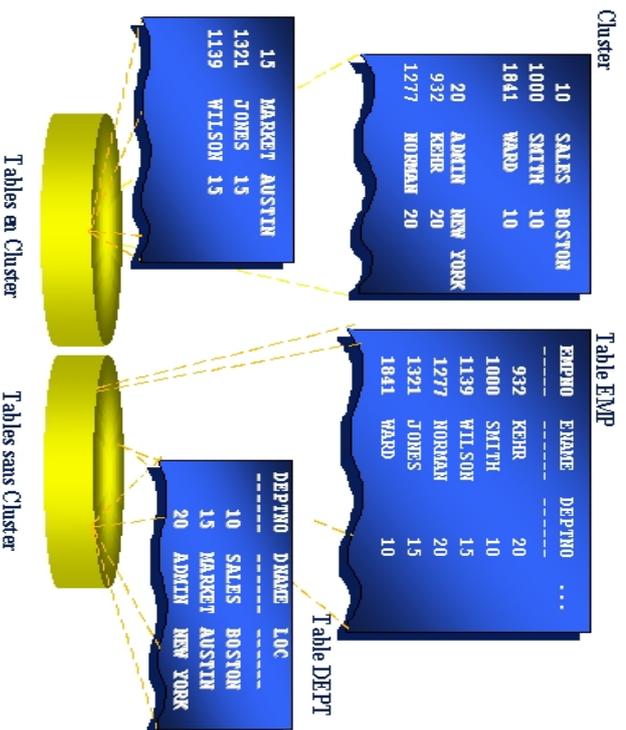
Cluster (suite)

Clusters Indexés



Cluster (fin)

Les Hash Clusters



Types de segments (3)

Index

- Toutes les entrées d'un index sont stockées dans un segment d'index
- Si une table à 3 index, 3 segments d'index sont utilisés
- *Intérêts* : accélère les recherches, les jointures ...

Index-Organized table

- les données sont stockées dans l'index basé sur une valeur clé

Partition d'index

- même idée que *Partition de table*

Types de segments (4)

■ Rollback segment

- utilisé par une transaction
 - avant de changer les blocs de données ou d'index, l'ancienne valeur est stockée dans le rollback segment
- 10g : possibilité de déléguer à Oracle leur gestion

■ Segment temporaire

- pour gérer l'opération de tri omniprésente en SQL
- permet d'implanter un algorithme de tri « out of core »

■ Segment LOB

- pour stocker des colonnes d'objets volumineux (LOB), e.g. documents textes, images, vidéos

169

Types de segments (5)

■ Index LOB

- implicitement créé lors de la création d'un segment LOB
- Pour accélérer les requêtes sur ces colonnes

■ Table imbriquée (V8 + Option Objects)

- table en non 1ere forme normal
- permet de faire de *l'orienté-objet*

■ Segment de démarrage (ou *segment en mémoire cache*)

- créé par le script sql.bsd
- ne nécessite pas d'attention de la part de l'administrateur !

170

Priorité des clauses de stockage

- Par ordre de priorité :
 - Segment
 - Tablespace
 - Valeur par défaut Oracle
- En cas de modifications des clauses de stockage
 - pas de changement sur ce qui existe
- Certaines clauses ne sont accessibles qu'au niveau des Segments

171

Allocation et restitution d'extents

- Alloué quand le segment est :
 - *créé* ou *étendu* ou *modifié*
- Restitué quand le segment est :
 - *supprimé* ou *modifié* ou *vidé* ou *automatiquement redimensionné* (rollback segment uniquement)

- Manuellement :
 - Avec l'ordre
 - ALTER TABLE table ALLOCATE EXTENT;
 - Avec les outils d'import/d'export

172

Fusion d'espace libre

- Libération d'espace contigu lorsque des extents sont libérés

- | ALTER TABLESPACE data01 COALESCE;

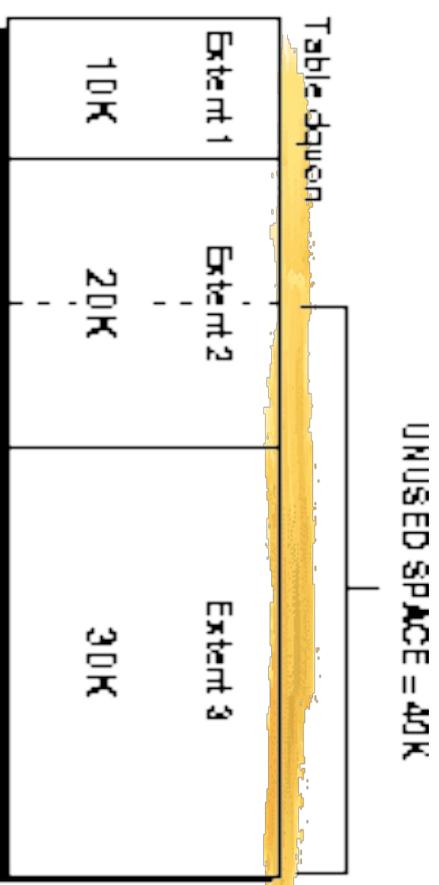
```
SELECT tablespace_name, total_extents,  
percent_extents_coalesced  
FROM dba_free_space_coalesced  
WHERE percent_extents_coalesced <> 100;
```

ou

- | Utiliser *Oracle Storage Manager*
 - tools->Coalesce Free Extents

- Sur une table
 - | ALTER TABLE table
DEALLOCATE UNUSED

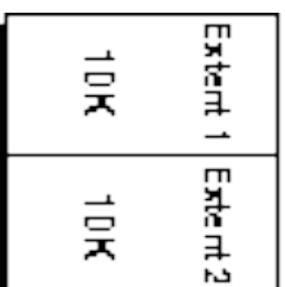
173



— ALTER TABLE space DEALLOCATE UNUSED; —

Before After

Table space

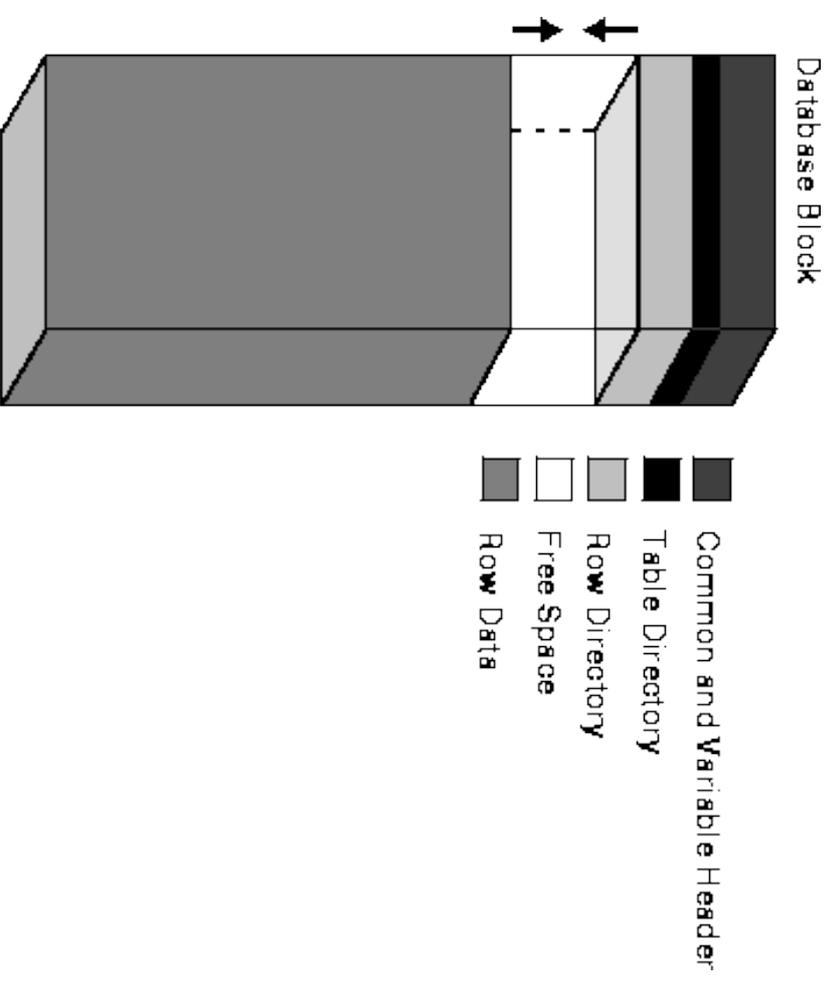


174

Blocs de base de données

- Rappels :
 - plus petite unité d'I/E/S
 - consiste en 1 ou plusieurs blocs SE
 - taille fixé par DB_BLOCK_SIZE : donné à la création de la base de données
- Un bloc contient :
 - *entête de bloc* : @ du bloc de données, répertoire de la table et de la ligne et entrées transactionnelles. Croit de façon *descendante*.
 - *espace des données* : les données d'une ligne sont insérées de manière *ascendante*.
 - *espace libre* : se trouve au milieu !

175



176

Paramètres d'un bloc

- Pour les accès concurrents :
 - INITRANS
 - | par défaut, vaut 1 si segment de données, 2 si segments d'index
 - MAXTRANS
 - | par défaut, vaut 255
- spécifie le nombre min/max d'entrées de transaction créées dans un bloc (de données ou d'index)
- Pour la gestion de l'espace libre :
 - PCTFREE : %tage d'espace réservé à la croissance due aux mises à jour
 - | par défaut, vaut 10%
 - PCTUSED : %tage d'espace au delà duquel le bloc n'est pas considéré comme libre
 - | par défaut, vaut 40%

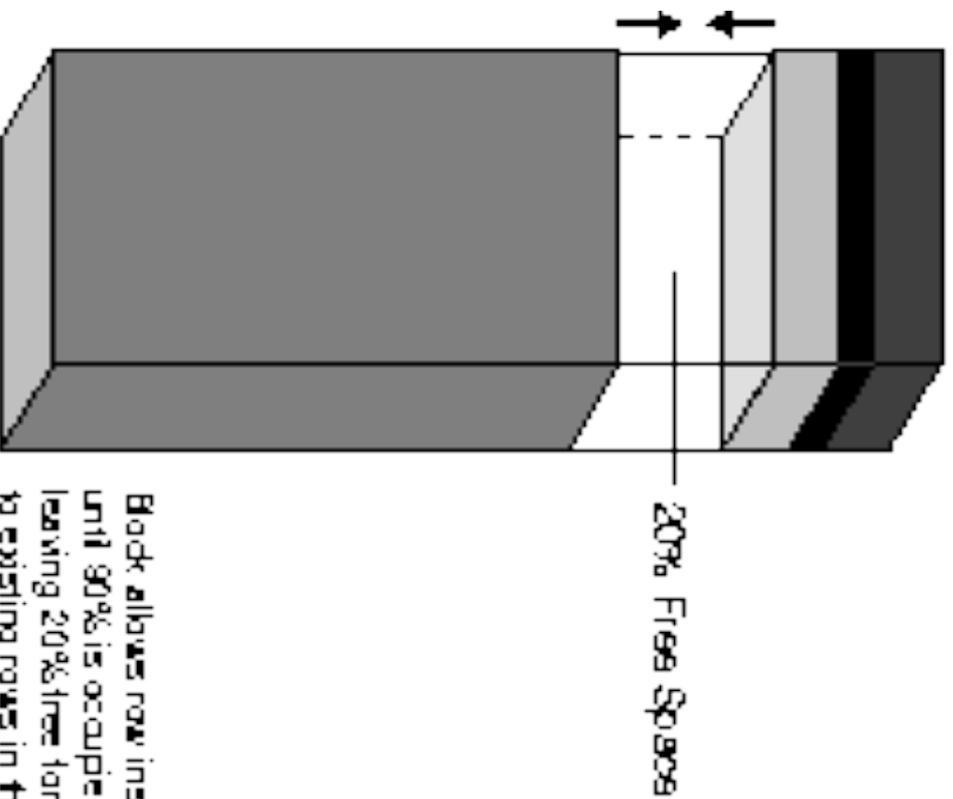
177

Gestion espace libre

- Pour un segment donné, Oracle gère une liste de *blocs libres*
- PCTFREE spécifique à partir de quand un *bloc libre* est considéré comme *plein*
- PCTUSED spécifique à partir de quand un *bloc plein* est considéré comme *libre*

178

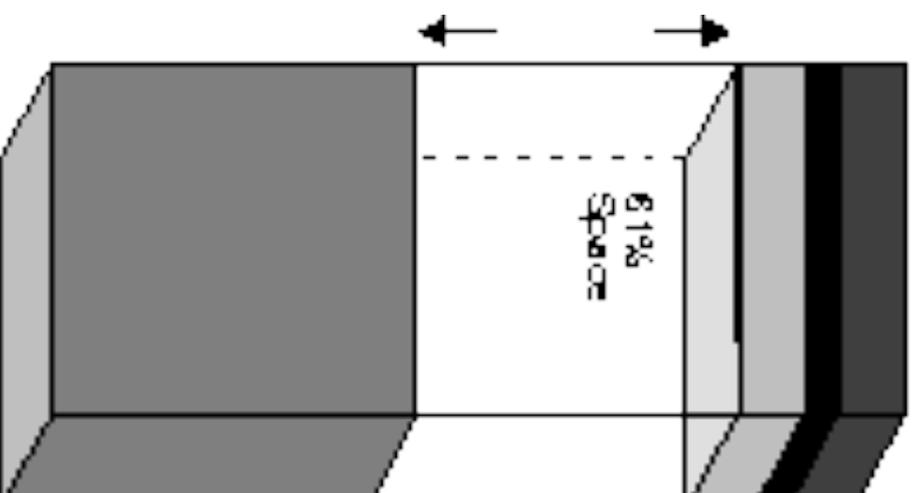
Database Block
PCTFREE = 20



Block allows row inserts until 80% is occupied, leaving 20% free for updates to existing rows in the block.

179

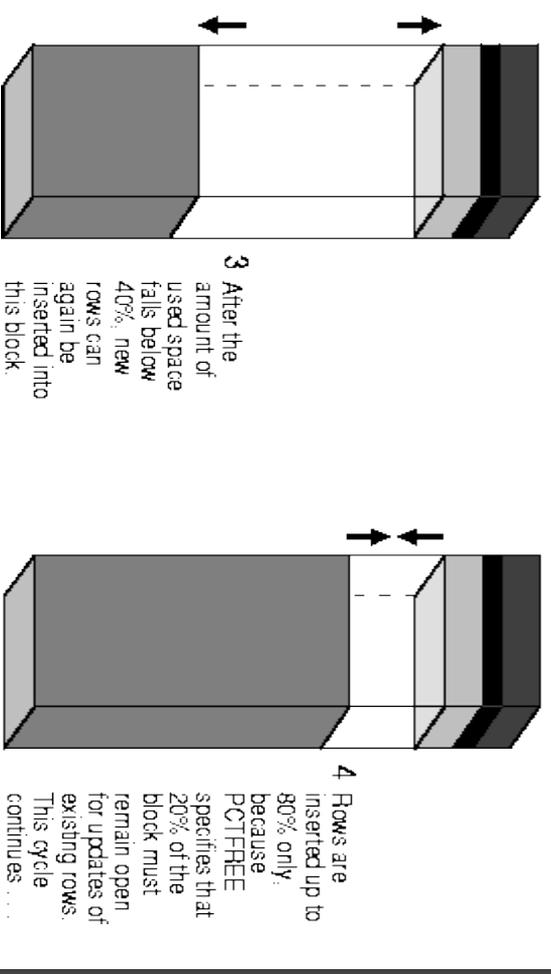
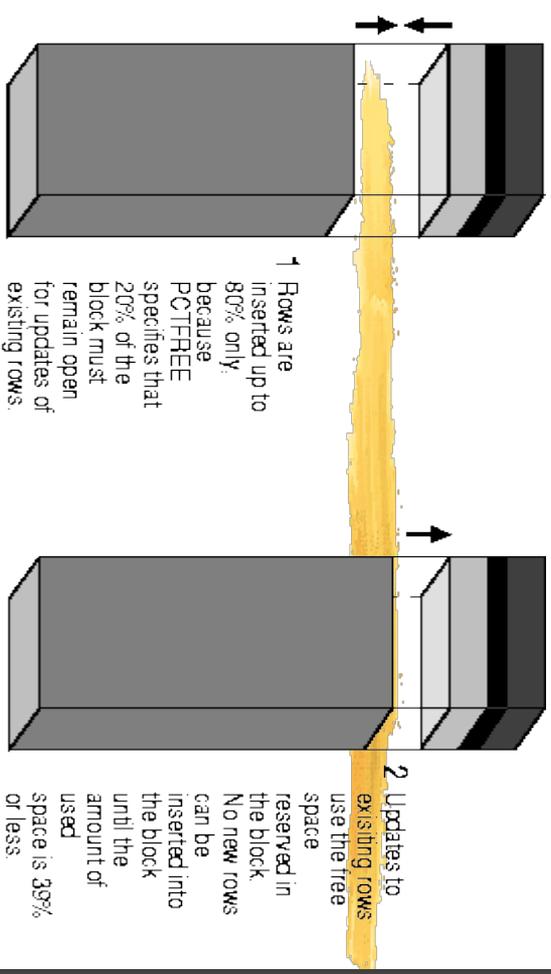
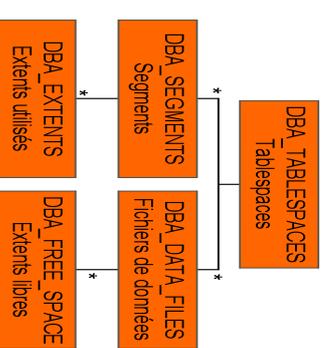
Database Block
PCTUSED = 40



No new rows are inserted until amount of used space falls below 40%.

180

Les vues du dictionnaire



Information sur le Segment

■ Interroger la vue

■ DBA_SEGMENTS

- | information générale :
 - owner, segment_name
 - segment_type, tablespace_name
- | taille :
 - extents, blocks
- | initialisation du stockage :
 - initial_extent, next_extent
 - min_extent, max_extent
 - pct_increase

■ Exemple :

```
SELECT segment_name, tablespace_name,
extents, blocks
FROM DBA_SEGMENTS
WHERE owner = 'SCOTT'
```

Information sur l'extent utilisé

■ Interroger la vue

■ DBA_EXTENTS

- | identification :
 - owner
 - segment_name
 - extent_id
- | Emplacement et taille
 - tablespace_name
 - relative_fno
 - file_id
 - block_id
 - blocks

■ Exemple :

```
SELECT extent_id, file_id, block_id, blocks
FROM DBA_EXTENTS
WHERE owner = 'SCOTT'
AND segment_name = 'EMP'
```

Information sur l'extent libre

■ Sur la vue

■ DBA_FREE_SPACE

| emplacements et taille

- tablespace_name
- relative_fno
- file_id
- block_id
- blocks

■ Exemple :

```
SELECT tablespace_name, count(*),  
       max (blocks), sum(blocks)  
FROM dba_free_space  
GROUP BY tablespace_name;
```

Tendance à la fragmentation

■ Tablespace

| | | |
|---------|---------------|------------|
| SYSTEM | Dico | Null |
| SYSAUX | | Null |
| TOOLS | Appli | Très basse |
| DATA_n | Seg. Données | Basse |
| INDEX_n | Seg. index | Basse |
| RBS_n | Rollback seg. | Élevé |
| TEMP_n | Seg. Tmp | Très élevé |

■ Influence sur la durée de vie ...

Gestion des rollbacks

Objectifs

- Maintenir des "informations" pour être capable d'annuler des mises à jour de données
- Deux types de gestion :
 - **au niveau d'un tablespace spécifique**
 - | "Automatic undo management" : UNDO TABLESPACE
 - | Introduit dans la 10g
 - | Fortement conseillé par Oracle
 - **au niveau des segments**
 - | "rollback segments"
 - | Oblige de gérer plusieurs rollbacks, leur taille, leur emplacement ...
 - | Obligatoire sur version antérieure à 10g

Intérêts

- Lecture cohérente
 - les autres transactions voient « l'image avant » des données grâce au rollback segment
- Rollback d'une transaction
 - la valeur avant modification est récupérée
- Restauration de transactions
 - En cas de panne, relecture du rollback segment dans un fichier de données
- NB :
 - Permet de gérer la cohérence des données lors des opérations de MAJ
 - **A ne pas confondre avec les fichiers de reprise**

189

Gestion au niveau d'un tablespace

- Paramètre d'initialisation :
UNDO_MANAGEMENT = AUTO
 - Les informations rollback sont désignées par **undo data**
 - Tout est géré par Oracle
 - Deux options
 1. Nommage explicite d'un tablespace spécifique
 - clause UNDO TABLESPACE lors de la création de base de données
 2. Création implicite d'un UNDO TABLESPACE nommé **SYS_UNDOTBS**
- => Option 1 est meilleure

190

Gestion au niveau segment 1/4

- 2 types de rollback segment :
 - SYSTEM
 - | dans le tablespace du même nom
 - | utilisé pour les modifications des objets de ce tablespace
 - Non-SYSTEM
 - | créé par l'administrateur
 - | deux cas :
 - Private
 - utilisation normale
 - Public
 - serveur // d'Oracle
- Allocation d'un rollback segment pour une transaction donnée :
 - | Oracle choisit le rbs "le plus" disponible et l'affecte à la transaction.
 - | Ou l'utilisateur le spécifie :
 - SET TRANSACTION USE ROLLBACK SEGMENT rbs012;

191

Création 2/4

- Ordre :
 - CREATE ROLLBACK SEGMENT
 - Exemple :

```
CREATE ROLLBACK SEGMENT rbs01
TABLESPACE rbs
STORAGE (
  INITIAL 100K
  NEXT 100K
  OPTIMAL 4M
  MINEXTENTS 20
  MAXEXTENTS 1000);
```
- créé OFFLINE par défaut
- PCTINCREASE ne peut pas être spécifié

192

Gestion

3/4

- Modification
 - ALTER ROLLBACK SEGMENT
 - Mise ONLINE
 - | ALTER ROLLBACK SEGMENT name ONLINE;
 - | ou ajouter dans init<sid>.ora :
 - ROLLBACK_SEGMENTS = (rbs01)
 - Libération d'espace
 - | ALTER ROLLBACK SEGMENT name SHRINK TO 2M;
 - Mise OFFLINE
 - | ALTER ROLLBACK SEGMENT name OFFLINE
- Suppression
 - DROP ROLLBACK SEGMENT name;

193

Informations

4/4

- Interroger la vue
 - DBA_ROLLBACK_SEGS
 - | identification
 - SEGMENT_ID
 - SEGMENT_NAME
 - | emplacement, type, status
 - TABLESPACE_NAME
 - OWNER
 - STATUS
- Statistiques
 - V\$ROLLNAME
 - V\$ROLLSTAT
- Activités courantes
 - V\$SESSION
 - V\$TRANSACTION

194

Gestion des segments temporaires

Intérêts

- Outil technique permettant d'implémenter une opération des SGBD : le **TRI**
- Nécessaire pour *tri* « out of core »
- Quantité de mémoire utilisée par un processus pour trier
 - paramètre SORT_AREA_SIZE
 - Allouée en PGA pour chaque session !!!
 - Allouée en SGA si architecture MTS
 - Si le volume de tri dépasse
 - plusieurs opérations de tri sont nécessaires
 - résultats intermédiaires stockés sur disque
- Segments temporaires créés dans le tablespace temporaire affecté à l'utilisateur pour trier

Types de segments temporaires

- Relatif au type de leur tablespace
 - Temporaire
 - Permanent
- Eviter les segments temporaires dans un tablespace permanent
 - fragmentation
- Segment temporaire dans un tablespace temporaire :
 - appelé aussi segment de tri
 - un seul segment de tri par tablespace et par instance
 - créé lorsque le premier tri se produit
 - libéré à l'arrêt de l'instance

197

Règles s'appliquant aux segments temporaires

- Initialiser plusieurs tablespaces TEMPORARY selon les besoins de tri
- Spécification du stockage par défaut:
 - PCTINCREASE = 0
 - assure que tous les extents ont la même taille
 - INITIAL = NEXT
 - = SORT_AREA_SIZE*entier + DB_BLOCK_SIZE
 - puisque un processus écrit sur disque les données de la zone de tri de la PGA (dont la taille vaut SORT_AREA_SIZE), on prévoit un multiple de cette taille.

198

Informations sur les segments temporaires

- Interroger les vues
 - V\$SORT_SEGMENT : Statistiques sur le segment de tri
 - V\$SESSION et V\$SORT_USAGE : activité du segment temporaire
 - V\$SORT_USAGE : tri actif en cours
 - DBA_SEGMENTS : donne le type du segment temporaire

- Exemple :

```
select s.username, u."USER",  
       u.tablespace, u.contents,  
       u.extents, u.blocks  
from   V$SESSION s, V$SORT_USAGE v  
where  s.saddr = u.session_addr;
```

Partie 5

Intégrité des données

201

Un peu de recul

- Quel sens donner à l'intégrité des données ?
- Plusieurs significations
 - Cohérence de la structure physique et adéquation avec la technologie matérielle
 - Cohérence de la structure logique
 - Cohérence des données

202

Intégrité des données

- Trois niveaux de codage :
 - le code applicatif
 - les triggers
 - les contraintes d'intégrité
 - PRIMARY KEY, UNIQUE, FOREIGN KEY
 - CHECK, NOT NULL
- Etat d'une contrainte
 - V7
 - DISABLE ou ENABLE
 - V8
 - DISABLE
 - ENABLE NOVALIDATE
 - les données existantes peuvent ne pas être cohérentes, les nouvelles le sont
 - ENABLE VALIDATE
 - toutes les données sont cohérentes
- Exemple :
 - ALTER TABLE emp
 - ENABLE NOVALIDATE pk1;

203

Contraintes différées

- Lors de mise à jour de la base de données (ordre du LMD), l'intégrité des données est vérifiée
 - soit à chaque ordre constituant la transaction
 - Option INITIALLY IMMEDIATE (par défaut)
 - soit en fin de transaction
 - option INITIALLY DEFERRED
 - Nécessite d'avoir spécifié que la vérification des contraintes serait déferée (alter table ...)
- Exemples :
 - au niveau de la session :
ALTER SESSION
SET CONSTRAINTS = IMMEDIATE;
 - au niveau de la transaction :
SET CONSTRAINTS pk1, fk2
DEFERRED;

- Cf syntaxe de la clause
CONSTRAINTS

204

Triggers

- Permet de définir des contraintes lors d'insertion/suppression/modification de lignes dans une table
- Activation
 - ALTER TABLE scott
ENABLE ALL TRIGGERS;
- Désactivation
 - ALTER TRIGGER scott.tg1 DISABLE;
- Suppression
 - DROP TRIGGER scott.tg1;

205

Informations relatives

- Pour les contraintes, interroger les vues
 - DBA_CONSTRAINTS
 - DBA_CONS_COLUMNS
- Pour les triggers, interroger les vues
 - DBA_TRIGGERS
 - DBA_TRIGGERS_COLS
 - DBA_OBJECTS

206

Qu'est-ce que la gestion de la sécurité ?

- Sécurité = protection des données
 - Piratage
 - Accès non autorisés
 - Manipulations non autorisées
 - ...
- Gestion de la sécurité sous Oracle
 - Plusieurs niveaux
 - Les utilisateurs
 - L'accès aux objets de la base de données
 - Les ressources du niveau système
 - Possibilité d'**audit** pour chaque niveau de sécurité

Partie 6

Gestion de la sécurité

Niveaux de sécurité

- Comptes utilisateurs
 - connexion (directe ou indirecte) à une base de données
 - protection par des mots de passe
- Rôles et privilèges
 - de niveau objets
 - gestion des accès aux objets (e.g., `select` sur la table EMP de SCOTT)
 - de niveau système
 - gestion des commandes de niveau système (e.g., `create table`, `alter index`, `create database`...)

209

Quelques notions de base

- Rôles
 - A mi-chemin entre utilisateurs et privilèges
 - Ordre de création similaire à celui des utilisateurs
 - Affectation de plusieurs privilèges
 - Gestion simplifiée des privilèges
 - Disponibilité sélective des privilèges
- Profils
 - Ensemble nommé de limites de ressources
 - e.g. nombre de connexions simultanées par utilisateur

210

Implémentation de la sécurité

- Niveau système d'exploitation
 - Autorisation d'accès au serveur
 - Protection des fichiers Oracle
 - fichiers de données, fichiers de reprise, fichiers de contrôle, dump, sauvegardes, ...
 - Protection des copies, réplicats, ...
- Niveau Oracle
 - Utilisateurs
 - Rôles
 - Profiles
 - Affectation directe des privilèges

Gestion des utilisateurs

Schéma de base de données

- Schéma de la BD
 - collection
 - de **tables**
 - de **vues**
 - d'index
 - de procédures stockées
 - ... référés par le terme **objets**
- Correspond à un *utilisateur*, i.e. celui qui crée les tables, les vues ...
- Utilisateur de la BD ?
 - correspond aussi à un utilisateur, mais qui exécute des requêtes SQL sur des tables, des procédures stockées ...
- Correspond aux 2 grands types d'utilisateur en BD

Liste de contrôles pour la création d'utilisateur

- Choisir un nom d'utilisateur et un mécanisme d'authentification
- Identifier les tablespaces dans lesquels l'utilisateur va stocker ses objets
 - typiquement 3 tablespaces pour les données, les tris et les index
 - Revient à affecter un tablespace par défaut et un tablespace temporaire
- Décider des quotas pour chaque tablespace
- Créer l'utilisateur
- Accorder des privilèges et des rôles à l'utilisateur

215

Exemples

- Sous sqlplus :
 - Authentification par Oracle
 - CREATE USER Paul IDENTIFIED BY edze!2z DEFAULT TABLESPACE data01 TEMPORARY TABLESPACE temp QUOTA 1M ON data01;
 - Authentification par le SE
 - CREATE USER Paul IDENTIFIED **EXTERNALLY** DEFAULT TABLESPACE data01 TEMPORARY TABLESPACE temp QUOTA 1M ON data01;
- *Conseil Oracle* : A utiliser avec modération !

216

Modification

- Avec ordre ALTER USER
- Exemples :
 - Changement du compte
 - | ALTER USER Paul IDENTIFIED BY edze!2z <- ligne optionnelle PASSWORD EXPIRE;
 - Suppression de quota
 - | ALTER USER Paul QUOTA 0 ON data01;
 - les données existantes restent
 - mais plus possible d'en insérer d'autres
 - quota illimité
 - | ALTER USER Paul QUOTA UNLIMITED ON data01;
 - Nom réel
 - | ALTER USER Paul GLOBALLY AS "Jean Paul Léponge";

217

Suppression

- Syntaxe :
 - | DROP USER utilisateur [CASCADE]
- Option CASCADE
 - supprime tous les objets du schéma
 - puis supprime l'utilisateur
- Exemple :
 - | DROP USER Paul CASCADE;
- Remarque :
 - impossible de supprimer un utilisateur connecté

218

Informations sur les utilisateurs

- Utiliser les vues
 - DBA_USERS
 - DBA_TS_QUOTAS

- Exemples :

```
select
  username, default_tablespace,
  temporary_tablespace
from dba_users
where username = 'PAUL';
```

- ```
select *
 from dba_ts_quotas
 where username= 'PAUL';
```

## Gestion des profils

### Profils

- Ensembles nommés de limites de ressource
  - e.g. nombre de connexions simultanées par utilisateur
- Affectés aux utilisateurs
  - lors de leur création
  - par modification
- Peuvent être *activés/désactivés*
- Peuvent dériver du profil DEFAULT

# Gestion des ressources à l'aide des profils

- Etapes à suivre :
  - créer les profils
    - | ordre CREATE PROFIL
  - les affecter à l'utilisateur
    - | ordre CREATE/ALTER USER
  - activer les limites de ressources
    - | soit ALTER SYSTEM
    - | soit fichier de paramètres d'initialisation
- Exemple

```
create profile new_prof limit
sessions_per_user 2
idle_time 1;
alter user Paul
profile new_prof;
alter system
set RESOURCE_LIMIT=true;
```

223

# Modification/suppression

- Modification
  - ordre ALTER PROFILE
  - Exemple :
    - | alter profile new\_prof limit
sessions\_per\_user 3
idle\_time 2;
- Suppression
  - ordre DROP PROFILE
  - option CASCADE : assure que tous les
utilisateurs ayant ce profil seront mis à jour!
  - le profile DEFAULT ne peut être supprimé
- Exemple :
  - | drop profile new\_prof cascade;
- Entre en vigueur pour les sessions
suivantes

224

# Informations sur les profils

- Utiliser les vues
  - DBA\_USERS
  - DBA\_PROFILES

- Exemples :

- ```
select distinct profile
from dba_profiles;
```

- ```
select *
from dba_profiles
where profile='DEFAULT';
```

## Gestion des privilèges

## Privilèges

- Deux types de Privilèges
  - **SYSTEME**
    - | permet aux utilisateurs d'effectuer des opérations dans la base de données
    - | Environ 80 privilèges systèmes
  - **OBJET**
    - | permet aux utilisateurs d'accéder à/manipuler un objet particulier
- Ordre **GRANT** permet d'ajouter un privilège à un utilisateur
  - Exemple :
    - | `GRANT create index, create table TO Paul;`
- Ordre **REVOKE** pour le supprimer
  - Exemple :
    - | `REVOKE create table FROM Paul;`

# Attribution de privilèges systèmes

- Avec l'option
  - | GRANT ... TO ...  
WITH ADMIN OPTION;
- Pour accorder un privilège système, il faut posséder le privilège WITH ADMIN OPTION
- Privilèges SYSDBA et SYSOPER
  - SYSDBA = privilèges de SYSOPER  
WITH ADMIN OPTION + CREATE  
DATABASE ...

229

# Privilèges Objets

- Les principaux privilèges Objet :
  - | ALTER, DELETE, INDEX, REFERENCES  
(pour les tables), SELECT, UPDATE,  
EXECUTE (pour les procédures)
- Exemples :
  - | GRANT execute ON dbms\_pipe  
TO public;
  - | GRANT select ON emp  
TO Bob  
WITH GRANT OPTION;
  - | REVOKE execute ON dbms\_pipe  
FROM scott;
- Attention à la formulation :
  - pour donner le privilège "with admin  
option", il faut faire :
    - | GRANT ... WITH GRANT OPTION

230

# Informations sur les privilèges

- Interroger les vues
  - pour les privilèges SYSTEM :
    - DBA\_SYS\_PRIVS
    - SESSION\_PRIVS
  - pour les privilèges OBJET :
    - DBA\_TAB\_PRIVS
    - DBA\_COL\_PRIVS

## ■ Exemple :

```
| select * from DBA_SYS_PRIVS;

| select * from SESSION_PRIVS;

| select *
 from DBA_TAB_PRIVS
 where grantee='BOB';
```

## Gestion des rôles

## Rôles

- A mi-chemin entre
  - les utilisateurs
    - Ordre de création de rôle similaire
  - les privilèges
    - Assure l'affectation de plusieurs privilèges
- Gestion simplifiée des privilèges
- Disponibilité sélective des privilèges

# Création/Affectation de Rôles

## ■ Création

### ■ Ordre CREATE ROLE

#### ■ Exemples :

- | CREATE ROLE Vente;
- | CREATE ROLE Vente IDENTIFIED BY bonus;
- | CREATE ROLE Vente IDENTIFIED EXTERNALLY;

## ■ Affectation

### ■ Ordre GRANT

#### ■ Exemples :

- | GRANT create any table TO Vente ;
- | GRANT Vente TO scott;
- | GRANT Vente TO scott WITH ADMIN OPTION;

# Rôles par défaut aux utilisateurs

## ■ Syntaxe :

- | ALTER USER ... DEFAULT ROLE [...|ALL|EXCEPT ...];

## ■ Exemples :

- | ALTER USER scott DEFAULT ROLE Vente, Achat;
- | ALTER USER scott DEFAULT ROLE ALL;
- | ALTER USER scott DEFAULT ROLE ALL EXCEPT Vente;
- | ALTER USER scott DEFAULT ROLE NONE;

# Activation on/off des rôles

- Relatif à la **session**
  - intéressant dans une application accédant la base pour s'assurer que la personne qui exécute l'application est bien autorisée.
- Commande SET ROLE
  - Exemples :
    - | SET ROLE Vente;
    - | SET ROLE Vente IDENTIFIED BY commission;
    - | SET ROLE ALL EXCEPT Achat;
    - | SET ROLE NONE;
- Suppression des rôles :
  - | DROP ROLE role
- Suppression des rôles d'un utilisateur
  - | REVOKE role FROM [util | PUBLIC];

237

# Conseils

- Créer un rôle pour *chaque tâche d'application*
  - e.g. traitement du salaire
- Attribuer aux *rôles d'application* les privilèges nécessaires pour exécuter la tâche
- Créer un rôle pour *chaque type d'utilisateur*
  - e.g. chef de service
- N'attribuer aux *rôles d'utilisateur* que des rôles d'application.
  - Ne pas affecter des privilèges individuels
- Accorder des rôles d'utilisateurs aux utilisateurs

238

# Informations sur les rôles

- Interroger les vues
  - DBA\_ROLES
    - | tous les rôles de la bd
  - DBA\_ROLE\_PRIVS
    - | rôles accordés aux utilisateurs et aux rôles
  - ROLE\_ROLE\_PRIVS
    - | rôles accordés aux rôles
  - DBA\_SYS\_PRIVS
    - | Privilèges accordés aux utilisateurs et aux rôles
  - ROLE\_SYS\_PRIVS
    - | Privilèges systèmes accordés aux rôles
  - ROLE\_TAB\_PRIVS
    - | Privilèges de table accordés aux rôles
  - SESSION\_ROLES
    - | rôles d'un utilisateur actuellement activés

## Sauvegarde logique vs physique

- au niveau logique
  - récupérer un fichier "dump" représentant les ordres de création "logique" de la BD
  - outils de migration entre différentes versions d'Oracle
    - **import/export**

- au niveau physique
  - on oublie SQL, on ne sauvegarde que les fichiers physiques

- **Aperçu du niveau physique ci après**

## Partie 7

### Sauvegarde / restauration

## Différents types de sauvegarde

- Sauvegarde base fermée
  - dite sauvegarde à *froid*
- Sauvegarde base ouverte
  - dite sauvegarde à *chaud*
- Différentes possibilités liées
  - au choix du mode d'archivage des fichiers de reprise
    - Mode NoArchivelog
      - la sauvegarde la plus récente est utilisée pour restaurer la base
        - perte d'information possible
    - Mode Archivelog
      - idem + fichiers de reprise archivés
        - pas de perte d'information
- de l'environnement technique
  - commandes du SE, outils NT, manuel, automatique
  - environnement disques (RAID)

243

## Influence de l'archivage

- mode NOARCHIVELOG
  - Restauration possible en cas d'incidents logiciel (processus SMON)
    - pas possible en cas d'incidents disques
  - Sauvegarde BD fermée
  - Pas de travail supplémentaire pour le DBA
- mode ARCHIVELOG
  - Restauration possible en cas d'incidents disque et logiciel
  - Sauvegarde possible tout en gardant la BD ouverte et prête à l'emploi
  - Beaucoup de travail supplémentaire pour le DBA pour maintenir les fichiers archivés

244

## Sauvegarde à froid

- La plus simple à mettre en œuvre
- Les étapes :
  - identification des fichiers
    - | `select name from v$datafile;`
    - | `select member from v$logfile;`
    - | `select name from v$controlfile;`
  - arrêter la base de données
  - sauvegarder tous ces fichiers
    - | commandes du SE
  - Redémarrer la base
- En cas d'erreur, copie de ces fichiers pour redémarrer
  - perte d'information possible
  - A privilégier si la BD n'est pas dans un environnement transactionnel élevé

245

## Sauvegarde à chaud

- Plus difficile à mettre en œuvre
- nécessite le mode ARCHIVELOG
  - possible aussi en mode Noarchiveolog mais les fichiers sauvés ne sont pas utilisables pour une restauration 😊
- Les grandes étapes :
  - base de données ouverte
  - soit mettre offline un tablespace
    - ALTER TABLESPACE tbs OFFLINE NORMAL;
    - soit utiliser l'ordre BEGIN BACKUP
  - ALTER TABLESPACE tbs BEGIN BACKUP;
  - sauvegarder les fichiers concernés
    - | commandes du SE
  - soit mettre online le tablespace
    - ALTER TABLESPACE tbs ONLINE
    - soit utiliser l'ordre END BACKUP
  - ALTER TABLESPACE tbs END BACKUP;
- Attention :
  - l'état d'un fichier de données sauvegardé online à partir du SE est *incertain*

246

## Exemple (mode ARCHIVELOG)

- Sauvegarde de fichiers de données
  - Identifier les fichiers de données d'un tablespace, eg 'tbs'
  - Tablespaces *activés*
    1. ALTER TABLESPACE tbs BEGIN BACKUP;
    2. Copier les fichiers à partir du SE
    3. ALTER TABLESPACE tbs END BACKUP;
  - Tablespaces *désactivés*
    1. ALTER TABLESPACE tbs OFFLINE NORMAL;
    2. Copier les fichiers à partir du SE
    3. ALTER TABLESPACE tbs ONLINE;
- Sauvegarde des fichiers de contrôle
  - ALTER DATABASE BACKUP CONTROLFILE TO 'e:\backup\control01.ctl';

247

## Requêtes

- Sur le mode d'archivage
  - select log\_mode from v\$database;
- En mode archivage
  - informations sur les tablespaces actuellement sauvegardés
    - v\$BACKUP, v\$DATAFILE\_HEADER,

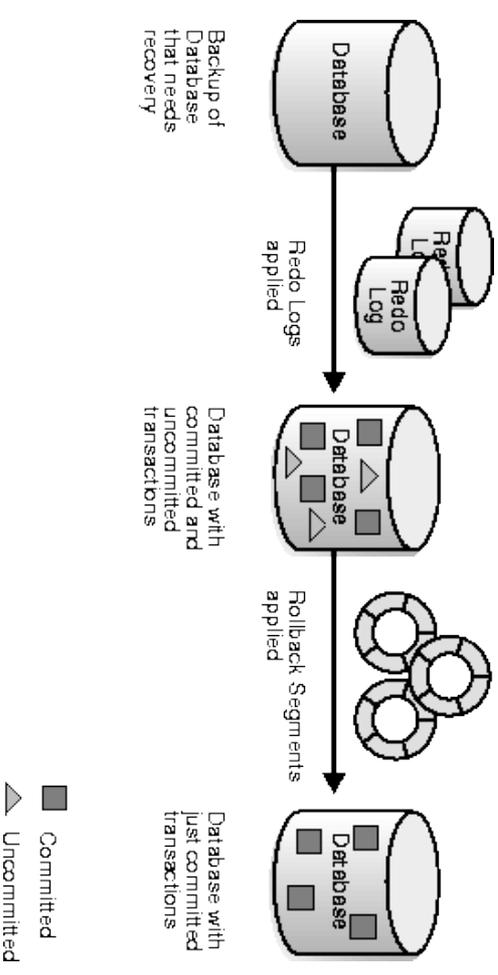
248

# Restaurations

- Mode Noarchivelog
  - perte des données saisies depuis la dernière sauvegarde
  - restauration de tous les fichiers
- Mode Archivelog
  - Différentes restaurations :
    - Restauration complète
      - Base de données fermée
      - Base de données ouverte, tablespaces désactivés
      - Base de données ouverte, tablespaces désactivés, fichiers de données endommagés
      - Fichier de contrôle corrompu
    - Restauration incomplète
      - basé sur l'annulation
      - basé sur les modifications et sur le temps
  - commande 'recover'
    - doc dans SQL\*Plus !

249

# Restauration: le principe



250

# Restauration complète

## Mode Noarchivelog

### ■ Les étapes

1. Fermer la base
  2. Réparer la panne disque
  3. Restaurer tous les fichiers de la sauvegarde la plus récente
  4. Ouvrir la base à l'étape OPEN
- Si les emplacements des fichiers ont changé, il faut renommer ces fichiers au niveau du fichier de contrôle
    - redémarrer la base à l'étape MOUNT
  - Des données peuvent avoir été perdues

251

# Restauration en mode Archivelog

### ■ Commande recover

```
RECOVER [AUTOMATIC] [FROM location]
{[STANDBY] DATABASE [UNTIL options]
 [USING BACKUP CONTROLFILE]}
|TABLESPACE {tablespace [, tablespace ...]}
|DATAFILE {datafilename [, datafilename ...]}
|STANDBY {TABLESPACE tablespace [,tablespace ...]
|DATAFILE datafilename [, datafilename ...]}
UNTIL CONTROLFILE
LOGFILE filename
|CONTINUE [DEFAULT]
|CANCEL}
[PARALLEL clause]
```

### ■ Possibilité de restaurer

- toute la base
  - des tablespaces
  - le fichier de contrôle
  - des fichiers de données
- Nécessite la présence des fichiers de reprise archivés

252

# Exemple de restauration partielle

Mode Archivlog

base ouverte, tablespace désactivé

1. Ouvrir la base à l'étape OPEN
2. Désactiver le(s) tablespace(s)  
`ALTER TABLESPACE nom OFFLINE`
3. Réparer la panne disque
4. Restaurer seulement les fichiers endommagés
5. Lancer la restauration  
`RECOVER TABLESPACE tablespace`  
ou pour chaque fichier de données  
`RECOVER DATAFILE nom_fich`
7. Activer le tablespace  
`ALTER TABLESPACE nom ONLINE`

## Partie 8

### Configuration du réseau

### Coté serveur

- Concept de base :  
*"mettre un processus sur le serveur NT à l'écoute de connexion sur un serveur Oracle par l'intermédiaire d'un ou plusieurs protocoles réseaux"*
- Protocoles réseaux
  - e.g., TCP/IP, NMP, ...
- Serveur BD Oracle
  - instance + base, e.g. ORCL, U01 ...
- Notion de *service*
  - utiliser pour la réplication, l'authentification par le SE, ...
- Processus « *listenerSID* » qui fait cette tâche

## Coté serveur (2)

- Les configurations sont enregistrées dans le fichier **listener.ora**
- Deux façons de procéder
  - à la main, en éditant le fichier
  - avec les outils graphiques
- Créer autant de *service* qu'il existe de façon d'accéder au serveur
- Redémarrer le PC pour prendre en compte les modifications
- Utiliser **lsnrctl**
  - commandes utiles :
    - **status**, **set password 'pwd'**, **stop 'listener'**, **start 'listener'**

257

## Exemple de fichier listener.ora

```
MESOREILLES =
 (DESCRIPTION_LIST =
 (DESCRIPTION =
 (ADDRESS = (PROTOCOL = TCP)(HOST = lmijmp)(PORT = 1521))
)
 (DESCRIPTION =
 (ADDRESS = (PROTOCOL = NMP)(SERVER = LMIJMP)(PIPE =
ORAPIPE))
)
)
)
SID_LIST_MESOREILLES =
(SID_LIST =
 (SID_DESC =
 (SID_NAME = ORCL)
)
 (SID_DESC =
 (SID_NAME = F3)
)
)
PASSWORDS_MESOREILLES = (test123)
```

258

## Coté Client

- Les configurations sont enregistrées dans le fichier **tnsnames.ora**
  - **c:\orant\net8\admin** ou
  - **c:\orant\sqlnet\admin**
- Deux façons de procéder
  - à la main, en éditant le fichier **tnsnames.ora**
  - avec *Net8 Easy Config*
- Créer un *nom de service*, i.e. un alias pour accéder la base de données
  - permet aux clients de se *connecter de façon transparente*
  - choisir un nom de service pertinent, e.g. *Paye*

259

## Exemple de fichier tnsnames.ora

```
BDTEST =
 (DESCRIPTION =
 (ADDRESS_LIST =
 (ADDRESS = (PROTOCOL = TCP)(HOST = lmjimp)(PORT = 1521))
)
 (CONNECT_DATA =
 (SERVICE_NAME = ORCL)
)
)
```

- Le SID de la BD est 'ORCL'
- La machine distante s'appelle 'lmjimp'
- Le protocole réseau est TCP/IP

260

## Coté Client (2)

### ■ Etapes à suivre :

1. Choisir un nom de service
2. Spécifier le protocole de communication avec le serveur
3. Donner le nom du serveur
4. Donner le SID de la base de données

## Audit

## Audit sous Oracle

- Possible sur toute **action** sur une base de données
  - tentatives de login
  - accès aux objets
  - actions bases de données
- Stockage des enregistrements de l'audit
  - SYS.AUD\$, ou
  - au niveau du système d'exploitation
- Pour activer l'audit
  - Paramètre AUDIT\_TRAIL (init.ora)
  - Valeurs possibles : NONE, DB, OS

263

## Audit des logins

- Commande *audit* et *noaudit*
- Exemples :  
*audit session whenever [not]*  
*successful*  
*noaudit connect*
- Résultats de l'audit
  - Vue DBA\_AUDIT\_SESSION
    - OS\_USERNAME
    - USERNAME
    - TERMINAL
    - RETURNCODE
    - ...

264

## Audit des actions

- Audit des commandes système (create, alter, drop, ...) sur un objet
  - Exemple  
*audit update table by scott;*
- Affectation d'un code pour chaque action pouvant faire l'objet d'un audit
  - Vue AUDIT\_ACTIONS
- Résultats de l'audit
  - Vue DBA\_AUDIT\_OBJECT  
pour déterminer comment un objet a été affecté par une action

265

## Groupes d'actions

- Possibilité de grouper les actions à auditer
  - exemple : audit de toutes les commandes qui affectent les rôles  
*audit role  
audit connet whenever not  
successful*
  - exemple de groupes fournis par Oracle
    - DBA : commande qui nécessitent une autorité DBA (grant, revoke, ...)
    - RESOURCE : create et drop pour les tables, clusters, vues, index, tablespaces, types et synonyms
    - ...

266

## Audit des objets

- Audit des commandes DML sur un objet
- Deux possibilités
  - par session  
*audit insert on scott.emp by session*
  - par accès  
*audit all on scott.emp by access*
- Toujours possible de spécifier en cas de réussite ou d'échec
- Résultats de l'audit
  - Vue DBA\_AUDIT\_OBJECT

267

## Remarques sur SYS.AUD\$

- Table SYS.AUD\$
  - contient des informations critiques
  - stockée dans le dictionnaire de données
  - peut croître rapidement
- ... donc il faut penser à :
  - ➡ l'archiver et la vider périodiquement
  - ➡ la sécuriser
  - ➡ l'auditer  
*audit all on SYS.AUD\$ by access;*

268

# Chargement des données

## Objectifs

- Aperçu de trois techniques :
  - Export et Import
    - | sauvegarde "logique" des données
    - | réorganisation des données
    - | exclusivement pour les environnements Oracle
    - | indépendant du SE et des versions d'Oracle
  - SQL\*Loader
    - | Pour charger des données stockées ailleurs que dans des bases Oracle
  - Direct-load insert
    - | chargement rapide de tables
- Communication des bases Oracles avec d'autres BD

# Utilitaire Export/ Import

## Réorganisation logique des données Migration

271

# Export/Import

- Trois modes
  - table
  - utilisateur
  - base de données
- Mode table
  - définition, données, index, triggers, contraintes, autorisations, ...
- Mode utilisateur
  - en fonction des droits de l'utilisateur
- Mode base de données
  - toute la base sauf les objets de l'utilisateur SYS
    - nécessite le privilège cf EXP\_FULL\_DATABASE

272

## Deux chemins d'export

- Chemin conventionnel
  - SQL SELECT
  - Chemin :
    - | données transférées du disque vers le buffer cache de données
    - | buffer d'évaluation
    - | transfert vers le fichier d'export
- Chemin direct (à partir de V7.3.3)
  - Chemin
    - | évite le buffer d'évaluation
  - plus rapide que le chemin conventionnel
  - n'influence pas de façon significative les performances de l'export

273

## Utilisation d'export

- Ligne de commande
  - utilitaire EXP
    - | exp help=y
  - exemple :
    - | exp scott/tiger tables=(dept,emp) \
    - | file=emp.dmp log=exp.log
- Une seule de ces options :
  - | tables=(*schéma.table*)
  - | owner=*utilisateur*
  - | full=y
    - option *incrémental, cumulative*
- Conventionnel :
  - | direct = n (par défaut)

274

# Utilisation d'Import

- Ligne de commande
  - utilitaire IMP
    - imp help=y
  - exemple :
    - imp scott/tiger file=emp.dmp log=imp.log ignore=y
- Privilège particulier
  - IMP\_FULL\_DATABASE
- Voir aussi *Oracle Data Manager*
  - Data -> Import

275

# Comportement d'Import

- Ordre d'import
  - lecture contiguë du fichier d'export
  - table -> données -> index -> contraintes
  - Si les tables existent déjà, il est recommandé de désactiver les contraintes d'intégrité référentielles avant de faire l'import
- Tablespace utilisé pour l'objet
  - si possible, même tablespace que dans la base de données source
  - sinon, tablespace par défaut de l'utilisateur
    - si quota suffisant

276

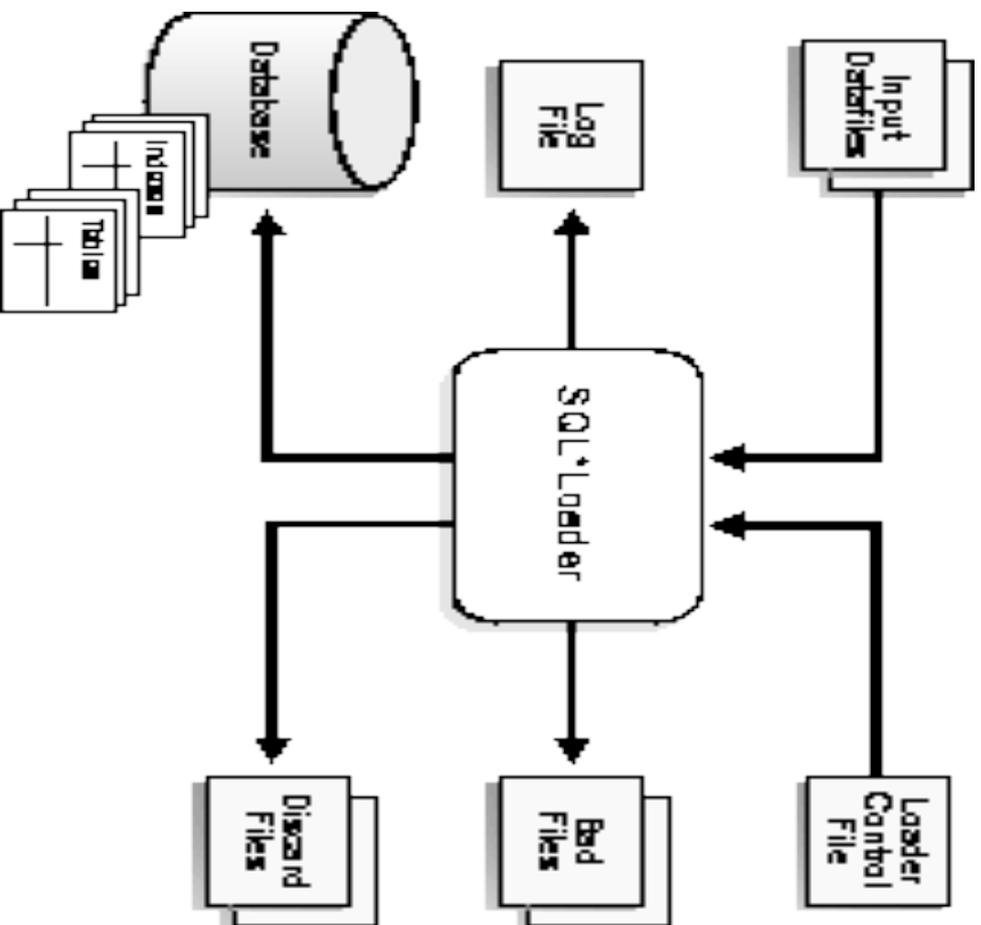
# Utilitaire SQL\*Loader

## Interface avec des données extérieures

## Les fichiers nécessaires

- **Fichiers de contrôle**
  - | spécifie le format d'entrée, les tables de sorties et des conditions sur le chargement
- **Fichiers de données**
  - | contiennent les données au format défini dans le fichier de contrôle
- **Fichiers de paramètres**
  - | facultatif, alternative au mode ligne de commande
- **Fichier "bad"**
  - | enregistrements refusés lors de l'insertion
- **Fichier "log"**
  - | trace du chargement
- **Fichier "discard"**
  - | facultatif, contient les enregistrements qui ne correspondraient pas aux critères de selection

# SQL\*Loader



279

## Deux chemins de chargement

- Chemin conventionnel
  - utilise l'ordre SQL INSERT
  - fichiers de reprise sont mis à jour
  - les contraintes sont respectées
  - comportement *classique*
- Chemin direct
  - écriture directe dans des blocs de données
  - pas d'informations enregistrées dans les fichiers de reprise (sauf si mode ARCHIVELOG)
  - applique seulement les contraintes de clé primaire, unique et not null.

280

# Utilisation

- Sous invite SE, par ex msdos :
  - SQLDR scott/tiger \ control=ulcase6.ctl \ log=ulcase6.log direct=true
  - Ne fonctionne pas en mode interactif
- Exemple de fichier de contrôle :

```
load data
infile 'ulcase6.dat'
replace
into table emp
sorted indexes (empix)
(empno position(1:4),
ename position(6:15),
job position(17:25),
mgr position(27:30) nullif mgr=blanks,
sal position(32:39) nullif sal=blanks,
comm position(41:48) nullif comm=blanks,
deptno position(50:51) nullif empno=blanks)
```

# Direct-Load insert

## Chargement direct

- V8 uniquement
- Utile pour charger des volumes de données important
  - plus rapide que les chemins conventionnels
- Ordre INSERT avec le *hint* APPEND
- Exemple :
  - ```
INSERT /*+APPEND*/ INTO bob.emp  
NOLOGGING  
SELECT * FROM scott.emp;
```
- L'option NOLOGGING spécifie que les modifications ne sont pas enregistrées dans les fichiers de reprise

SGA

- Vues dynamiques :
 - V\$SGA : généralité
 - V\$SGAINFO : info sur taille composants
 - V\$SGASTAT : info détaillée
- Exemple :
 - ```
select sum(value) from V$SGA
```
- Paramètres d'initialisation (**10g**)
  - db\_block\_buffer, sga\_target
  - shared\_pool\_size, java\_pool\_size, Large\_pool\_size, db\_cache\_size
  - log\_buffer, db\_keep\_cache\_size