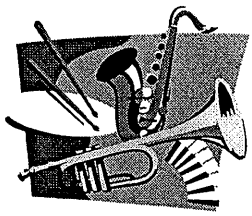
SUPPORT DE COURS ORACLE

CHAPITRE 6

STRUCTURES D'UNE BASE DE DONNEES ORACLE



1 ☐ Structures d'une base de données Oracle
2 ☐ Structure physique d'une base de données Oracle
3 ☐ Fichiers de données
4 ☐ Fichiers de reprise
5 ☐ Fichiers de reprise (suite)
6 ☐ Fichiers de contrôle
7 ☐ Structure logique d'une base de données Oracle
8 ☐ Bloc de données ou page
9 ☐ Structure d'un bloc
10 ☐ Paramètres de gestion des blocs
11 ☐ Blocs et statistiques
12 ☐ Extension (ou extent) et segment
13 ☐ Gestion des extensions et segments
14 ☐ Paramètres de gestion des extensions
15 ☐ Récupération de l'espace inutilisé
16 ☐ Types de segments
17 ☐ Types de segments (suite)
18 ☐ Les segments d'annulation
(ou rollback segments)
19 ☐ Les segments d'annulation (suite)
20 ☐ Tablespace: définition
21 ☐ Le tablespace System
22 ☐ Tablespace: création (syntaxe)
23 ☐ Tablespace: création (exemple)
24 ☐ Tablespace: modification
25 ☐ Tablespace: suppression
26 ☐ Redimensionnement manuel de la base de données
27 ☐ Redimensionnement automatique des fichiers de données
28 ☐ Création d'une table
29 ☐ Création d'une contrainte
UNIQUE ou PRIMARY KEY
30 ☐ Création d'une contrainte (suite)

Structures d'une base de données Oracle

- Architecture ANSI/SPARC (1975)
  - niveau physique
  - niveau conceptuel (ou logique)
  - niveau externe

La description de ces 3 niveaux et la correspondance entre eux est faite à travers un dictionnaire de données

Structure physique d'une base de données Oracle

- Fichiers de données (Data files)
- Fichiers de reprise (Redo Log Files)
- Fichiers de contrôle (Control files)

Spécification des fichiers de données ou de reprise faite lors de la création ou modification de la base

Fichiers de données

- Un ou plusieurs fichiers associés à une seule base
- Assurent le stockage des objets des utilisateurs
- Assurent le stockage du dictionnaire de données
- Tailles déterminées lors de leur création (2Mo minimum pour le premier fichier)

Fichiers de reprise

- Au nombre minimum de 2
- Contiennent les modifications des données les plus récentes
- Toute opération de m.a.j. est enregistrée dans ces fichiers
- Fichiers circulaires

Fichiers de reprise (suite)

- Utilisés pour la *reprise à chaud*
- Peuvent être dupliqués (Fichiers de reprise *multiples*; l'ensemble des fichiers de reprise actifs constitue un *groupe*; chacun de ces fichiers est dit *membre*)

Fichiers de contrôle

- Au nombre minimum de 1
- Contient les informations relatives à la structure de la base (nom de la base, noms et localisation des fichiers de données et de reprise)
- Mis automatiquement à jour

Structure logique d'une base de données Oracle

Éléments de la structure logique d'une base de données:

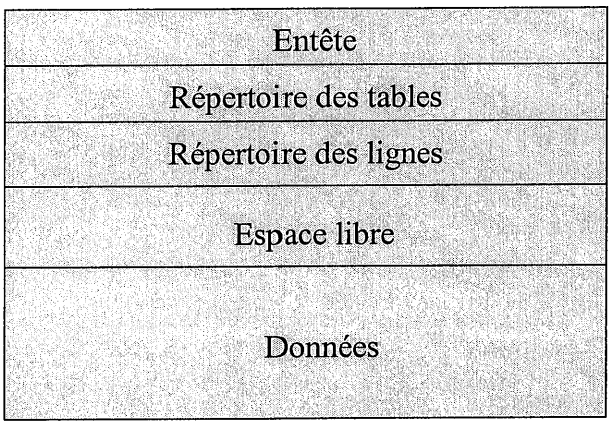
- tablespaces
- segments
- extensions (extents):
- blocs de données (ou pages)

Ces éléments permettent de stocker les objets de schéma (tables, index, clusters, vues, séquences, procédures stockées, fonctions, packages, triggers, etc.)

Bloc de données ou page

- C'est la plus petite unité logique d'E/S
- DB_BLOCK_SIZE dans le fichier d'initialisation (INIT.ORA) détermine sa taille
- DB_BLOCK_SIZE est fixé à la création de la base

Structure d'un bloc



Paramètres de gestion des blocs

- PCTFREE: pourcentage d'espace réservé pour de futures modifications; valeur par défaut: 10%
- PCTUSED: pourcentage d'espace à partir duquel le bloc est de nouveau candidat pour l'insertion;

  valeur par défaut: 40%

  ↳ Remarque:PCTFREE +PCTUSED<=100%
- INITRANS:nombre d'entrée de transactions initialement alloués dans l'entête du bloc.
- MAXTRANS:nombre maximum de transactions pouvant accéder simultanément au bloc.

Blocs et statistiques

ANALYZE type_objet [schéma].objet
COMPUTE STATISTICS
/ESTIMATE [SAMPLE entier ROWS/PERCENT] STATISTICS
/DELETE STATISTICS
/LIST CHAINED ROWS [INTO [schéma].table]

Avec type_objet ou objet :table, index ou cluster

Extension (ou extent) et segment

- **Extension**: unité logique d'allocation d'espace composée d'un ensemble contigu de blocs de données alloué à un segment.
- **Segment**: ensemble d'une ou de plusieurs extensions contenant les données d'un objet; ne peut pas s'étendre, de façon générale, sur plus d'un tablespace sauf dans le cas d'un objet partitionné(Oracle8i).

Gestion des extensions et segments

- Une extension est allouée quand:
  - le segment est créé (INITIAL EXTENT)
  - le segment croît (NEXT EXTENT)
- Une extension est libérée quand:
  - le segment est supprimé
  - la commande TRUNCATE est utilisée

Paramètres de gestion des extensions

- Ces paramètres peuvent être positionnés sur les tables, clusters, indexes, rollback segments et tablespaces; quand ils ne sont pas spécifiés les valeurs par défaut du serveur sont utilisées.
  - INITIAL: taille en octets de la 1ère extension allouée à un bloc; la valeur par défaut équivaut à 5 blocs.
  - NEXT: taille en octets de l'extension suivante allouée à un bloc; la équivaut à 5 blocs.
  - MAXEXTENTS; valeur par défaut: 121
  - MINEXTENTS ; valeur par défaut: 1
  - PCTINCREASE: la valeur par défaut est de 50%

Récupération de l'espace inutilisé

ALTER TABLE [schéma.]table
DEALLOCATE UNUSED [KEEP entier [K/M]] (version 7.3)
KEEP : ne libère pas tout l'espace

- Vérifier l'espace récupéré en interrogeant, avant et après la manipulation, la vue DBA_FREE_SPACE

Types de segments

- **Segments de données:** contiennent les données des tables et des clusters (des utilisateurs ou du dictionnaire)
- **Segments d'index**: contiennent les index
- **Segments temporaires**:sont utilisés pour le traitement des commandes nécessitant un espace disque temporaire (jointures, distinct, order by, group by, création d'index, opérations ensemblistes...); si aucun tablespace temporaire n'a été défini pour un utilisateur, c'est le tablespace system qui est utilisé

Types de segments (suite)

- **Segment d'amorçage (bootstrap)**: est créé dans le tablespace *System*, contient les définitions du dictionnaire de données.
- **Segment d'annulation (rollback segment)**: contiennent les données avant qu'elles soient modifiées par la transaction qui les manipule (*images avant*); à une transaction correspond un seul rollback segment; chacune de ses composantes (rollback entries) contient:
  - l'identification du bloc modifié (nom de fichier et du bloc)
  - les valeurs des données avant modification par la transaction

Les segments d'annulation (ou rollback segments)

Création:

- CREATE [PUBLIC] ROLLBACK SEGMENT rollback_segment
  [TABLESPACE nom_du_tablespace]
  [STORAGE (spécifications de stockage)]

Initialement le rollback segment est hors ligne. Pour le rendre accessible aux transactions il faut utiliser la commande permettant sa

Modification:

- ALTER ROLLBACK SEGMENT rollback_segment ONLINE /OFFLINE
  /STORAGE (spécifications de stockage)

Les segments d'annulation (suite)

- La taille des segments d'annulation peuvent s'avérer insuffisante pour une transaction; en effet une transaction par lots peut nécessiter un espace auquel la configuration des segments d'annulation de production ne peut répondre.
- Des exigences transactionnelles particulières (chargement de données, suppression massive etc.) devraient être gérées à l'aide d'un autre segment d'annulation créé dans un tablespace lui même créé pour l'occasion et pouvant être supprimé après utilisation afin de procéder à une récupération de l'espace disque.
- Une fois le segment d'annulation créé, il faut que la transaction y fasse référence par la commande:
- **SET TRANSACTION USE ROLLBACK SEGMENT rollback_segment**

Tablespace: définition

- Identifié par un nom
- Au nombre minimum de 1
- Ensemble d'unités logiques composant la base
- Regroupe un ensemble d'objets logiques
- Chaque objet logique est associé à un et un seul tablespace
- Permet de regrouper les objets logiques d'une même application
- Est constitué d'un ou de plusieurs fichiers
- Peut être activé (*on line*) ou désactivé (*off line*)

Le tablespace System

- Le tablespace *system* (obligatoire) contient le dictionnaire de données
- Indispensable au fonctionnement de la base
- Contient le *system rollback segment*
- Pourrait contenir des données des utilisateurs
- Ne peut être désactivé

Tablespace: création (syntaxe)

```
CREATE TABLESPACE nom_tablespace
    DATAFILE spécif_fichier {, spécif_fichier}
    [DEFAULT STORAGE (spécif_stockage)]
    [ONLINE / OFFLINE] [PERMANENT / TEMPORARY]
spécif_fichier: 'nom_de_fichier' [SIZE entier] [REUSE]
spécif_stockage:  INITIAL entier [K/M]  /*taille du 1er extent devant
                                        être alloué pour l'objet */
                  NEXT    entier [K/M]  /*taille du prochain extent */
                  MINEXTENTS  /* nombre total d'extents alloués
                        quand le segment est créé; défaut:1 */
                  MAXEXTENTS /* nombre total d'extents que le
                        serveur peut allouer pour l'objet; défaut:121 */
                  PCTINCREASE /* pourcentage d'accroissement d'un
                        nouvel extent; défaut:50% */
```

Tablespace: création (exemple)

```
CREATE TABLESPACE stock
DATAFILE '/home/oracle/dbs/users2.dbs' SIZE 5M
   DEFAULT STORAGE
        ( INITIAL  100K
         NEXT 50K
         MINEXTENTS 10
         MAXEXTENTS 121
         PCTINCREASE 1 ) ;
```


Tablespace: modification

```
ALTER TABLESPACE nom_tablespace
   ADD DATAFILE spécif_fichier {, spécif_fichier}
   /RENAME nom_de_fichier[s] TO nom_de_fichier[s]
   [DEFAULT STORAGE (spécif_stockage)] /* nouveau
                        paramêtre pour les nouveaux objets */
   [ONLINE / OFFLINE] [PERMANENT / TEMPORARY]
```

Tablespace: suppression

DROP TABLESPACE nom_tablespace

[INCLUDING CONTENTS] /* supprime tout le contenu du tablespace; clause obligatoire si le tablespace contient encore des données */

[CASCADE CONSTRAINTS] /* supprime toutes les contraintes d'intégrité référentielle */

- Les fichiers correspondants de l'OS doivent être physiquement supprimés

Redimensionnement manuel de la base de données

- ALTER DATABASE
  DATAFILE 'nom_de_fichier'
  RESIZE entier [K/M]

Redimensionnement automatique des fichiers de données

- La clause AUTOEXTEND (à partir de la 7.2) active ou désactive l'extension automatique des fichiers de données.
- Les commandes suivantes utilisent cette clause:
  - CREATE DATABASE
  - CREATE TABLESPACE
  - ALTER TABLESPACE
- ...AUTOEXTEND OFF/ON [NEXT entier [K/M]] [MAXSIZE][UNLIMITED]/entier [K/M]

Création d'une table

- Syntaxe simplifiée:

CREATE TABLE [shéma.]table
(... ...)
[PCTFREE entier]
[PCTUSED entier]
[INITRANS entier]
[MAXTRANS entier]
[TABLESPACE tablespace]
[STORAGE (spécif_stockage)] /* voir paramêtres définis en page *Tablespace:création* */
/ [CLUSTER cluster (liste de colonnes)]

Création d'une contrainte UNIQUE ou PRIMARY KEY

- La définition d'une contrainte d'intégrité UNIQUE ou PRIMARY KEY génère un index qui par défaut est stocké dans le même tablespace que celui de la table. Les caractéristiques de stockage de cet index sont par défaut celles du tablespace.
- Cette définition d'une contrainte d'intégrité peut être utilisée dans la définition ou la modification d'une table (CREATE / ALTER TABLE)

Création d'une contrainte (suite)

Syntaxe:

- CONSTRAINT nom_de_la_contrainte
  UNIQUE/ PRIMARY KEY (liste_de_colonnes)
  [USING INDEX [TABLESPACE nom_du_tablespace]
  [PCTFREE entier][STORAGE (spécifications de stockage)]]
- La clause USING INDEX permet de caractériser l'index créé

Exemple:

- ALTER TABLE prix
  ADD CONSTRAINT prix_pk PRIMARY KEY (nfou,refpiece)
  USING INDEX TABLESPACE commandes_I
- Déplacement d'un index:

Exemple:

- ALTER INDEX fournisseurs_pk **REBUILD** TABLESPACE commandes_I
- La clause REBUILD permet de reconstruire l'index

Exercices: structures d'une base Oracle

1. Donner les noms des tablespaces composant la base.
2. Donner les noms des fichiers des différents tablespaces.
3. Ajouter un tablespace ***tbs_test*(n)** s'appuyant sur un fichier '*tbs*(n)' de 2M dans le répertoire '*data2*'.
4. Modifier le tablespace ***tbs_test* (n)**, en y ajoutant un fichier '*test*(n)' de 1M, toujours dans le répertoire '*data2*'.
5. Créer une table *matable* dans le tablespace ***tbs_test*(n)** avec un espace libre de 80% et un taux de croissance de 0. Est-ce possible et pourquoi ? Rectifiez les valeurs.
6. Donner les caractéristiques de stockage des tables *departements* et *matable* (vue *user_tables* ou *tabs*).
7. Déplacer l'index de la clé primaire de la table *departements* vers le tablespace ***tbs_test*(n)**. Vérifier en consultant la liste de vos index.
8. Déplacer les fichiers du tablespace ***tbs_test*(n)** du répertoire '*data2*' vers le répertoire '*data3*'.
9. Supprimer le tablespace ***tbs_test*(n)**. Que se passe-t-il ? Remédier au problème .Une fois le tablespace ***tbs_test*(n)** supprimé, la table *matable* est-elle toujours accessible et les fichiers correspondant existent ils toujours?

Exercice sur les Rollback Segments :

<u>Objectifs</u> : Création et paramétrage d'un Rollback Segment ; affection d'une transaction à Rollback Segment.

1. Créer une table d'importance (par exemple à partir de colonnes de la vue dba_objects). Lui attribuer des paramètres conséquents. Insérer les données provenant de dba_objects. Observer dans dba_segments les conséquences.
2. Créer un tablespace RBS2 de 2M.
3. Créer un rollback segment avec un maxextents de 4 ou 8.
4. Le mettre en ligne.
5. Affecter la transaction à ce rollback segment.
6. Supprimer les données de la table -> échec car ce rollback segment est trop petit.
7. Le mettre hors ligne.
8. Modifier le rollback segment avec un maxextents unlimited.
9. Le remettre en ligne.
10. Affecter de nouveau la transaction à ce rollback segment.
11. Supprimer les données de la table -> succés.
12. Mettre le rollback segment hors ligne.
13. Supprimer le rollback segment ou/et le tablespace.

<u>Remarque</u> : Ne pas oublier les fins de transaction aux bons endroits.

SUPPORT DE COURS ORACLE

CHAPITRE 7

ARCHITECTURE ET MISE EN ŒUVRE D'ORACLE7

Sommaire du chapitre 7

Architecture et mise en œuvre d'Oracle7

- Une instance Oracle est la combinaison :
  - de structures mémoires
  - de certains process Oracle (les process détachés)
- Une instance démarrée peut monter la base de données constituée:
  - de fichiers

Structure de la mémoire

- Trois types de zones:
  - zones réservées au code l'applicatif
  - zone globale système (la SGA)
  - zone globale programme (la PGA)

Zones réservées au code de l'applicatif

- Composées de parties de la mémoire permettant de stocker le code des programmes en cours d'exécution:
  - noyau (zone séparée des autres)
  - outils Oracle (SQL*Plus, SQL*Forms etc...)
  - applicatif faisant appel à Oracle
- Accessibles uniquement en lecture, elles peuvent être partagées ou exclusives (sauf certains OS)

La SGA: System Global Area

- Groupe de srtuctures de mémoire **partagée** contenant des données et des informations de contrôle (aussi appellée Shared Global Area)
- Allouée au démarrage du serveur
- En mémoire non-paginée et non-swappée
- Taille déterminée par 4 paramêtres:
  - DB_BLOCK_SIZE (taille d'un bloc)
  - DB_BLOCK_BUFFERS (nombre de tampons)
  - LOG_BUFFER (taille de la zone tampon de reprise: redo log buffer)
  - SHARED_POOL_SIZE (taille de la shared pool)

Zones de la SGA: la Shared Pool

- la zone Pool Partagé (la Shared Pool) contient:
  - des zones de requêtes SQL partagées (textes des ordres SQL, formes analysées, plan d'exécution) ou privées (serveur multi-thread) gérées par un algorithme LRU (Least Recent Used)
  - un cache mémoire du dictionnaire de données
- sa taille est déterminée par le paramêtre: SHARED_POOL_SIZE
- pour libérer manuellement la Shared Pool utiliser:
  - ALTER SYSTEM FLUSH SHARED POOL.

Zones de la SGA: le Database Buffer Cache

- le cache des buffers de la base de données est un ensemble de copies des blocs de données lus à partir des fichiers
- son espace est géré à l'aide de deux listes:
  - une liste de tampons modifiés mais non encore écrits sur disque (*dirty list*)
  - une liste de tampons les moins utilisés récemment (*least recently used list* ou LRU) qui contient les tampons libres (pouvant être utilisés), les tampons en cours d'utilisation par certains processus, et les tampons modifiés mais non encore transférés dans la 1ére liste (cf. paramêtre DB_BLOCK_MAX_SCAN et processus DBWR)

Zones de la SGA: le Database Buffer Cache

- Définitions:
  - le cache *miss*: le bloc de données n'est pas présent en mémoire; il est lu à partir du disque
  - le cache *hit* : le bloc de données est présent en mémoire; l'accès est beaucoup plus rapide
- Contenu de la liste LRU:

Types de buffers	Description
Free buffers	Buffers qui n'ont pas été modifiés et sont disponbles
Pinned buffers	Buffers qui sont en cours d'accès
Dirty Buffers	Buffers modifiés qui doivent écrits sur disque

Zones de la SGA: le Redo log buffer

- la zone de tampon de reprise est un tampon circulaire qui contient des informations relatives aux modifications apportées à la base de données; ces informations peuvent être utilisés en cas de reprise.
- le contenu de ce tampon est écrit dans le(s) fichier(s) de reprise (Redo log file) par la processus LGWR.

Zones de la SGA: autres informations

- les autres informations que peut contenir la SGA sont:
  - informations communiquées entre processus telles que des informations relatives au verrouillage
  - informations relatives au dictionnaire

La PGA: Global Program Area

- la zone programme globale contient des données et informations relatives à un processus (serveur ou d'arrière plan).
- une zone PGA est allouée par Oracle lorsqu'une session est ouverte.
- la PGA est modifiable et non partagée
- le contenu de la PGA varie selon que l'on utilise un serveur multi-threaded ou non

Les processus Oracle

- Deux types de processus SGBD:
  - les processus d'arrière-plan ou processus détachés (Background process) effectuent d'une façon asynchrone l'ensemble des tâches du SGBD pour tous les utilisateurs.
  - les processus serveurs permettent d'associer à un ensemble de processus utilisateurs un seul processus serveur et de minimiser ainsi la charge de l'OS.
- Les processus utilisateurs:
  - un processus utilisateur est créé lorsqu'un outil Oracle est exécuté par un utilisateur ou à travers une application

Les Background process

- DBWR (Database Writer) obligatoire
- LGWR (Log Writer) obligatoire
- CKPT (Checkpoint)
- SMON (System monitor) obligatoire
- PMON (Process monitor) obligatoire
- ARCH (Archiver)
- RECO (Recoverer) option supplémentaire
- LCKn (Lock) option supplémentaire
- Dnnn (Dispatcher)
- Listener
- Snnn (Shared Server)
- Parallel Query (Pnnn) option supplémentaire
- SNPn (Snapshot Refresh) option supplémentaire

PMON

- PMON
  - nettoie les connexions terminées de facon anormale
  - défait les transactions non validées
  - libère les verrous qui avaient été posés par un process qui s'est terminée en erreur
  - redémarre les serveurs partagés et les process *dispatcher* en erreur

SMON

- SMON
  - réalise la restauration automatique d'instance
  - récupère l'espace occupé par des segments temporaires qui ne sont plus utilisés
  - fusionne les zones contiguës d'espace libre dans les fichiers de données

DBWR

- DBWR
  - Gére le Database Buffer Cache
  - Ecrit périodiquement les modifications dans les fichiers de données mais diffère les écritures en vue d'optimiser les E/S
- DBWR écrit les buffers dirty sur le disque quand:
  - la dirty list atteint une valeur limite
  - un process a parcouru un nombre spécifié de buffers dans la LRU list sans trouver un buffer libre
  - un time-out se produit ou un checkpoint de DBWR se produit

LGWR

- LGWR
  - Assure l'écriture des Redo log buffers dans le fichier de reprise (Redo log)
  - Si il existe des groupes de fichiers Redo log, LGWR effectue une écriture synchrone sur tous les fichiers
  - le processus est activé:
    - lorsqu'un commit se produit
    - lorsque le tiers de la zone tampon de reprise est plein
    - lorsque le DBWR effectue une écriture des tampons modifiés sur disque

CKPT

- Pendant un point de synchronisation (*checkpoint*), le DBWR écrit les données modifiées du cache base de données sur disque. Un numéro de séquence de checkpoint est mis à jour dans chaque fichier de la base et dans le(s) fichier(s) de contrôle.
- Pour améliorer les performances il est possible d'activer CKPT qui prend en charge l'opération mené par LGWR

ARCH

- Copie les fichiers redo log actifs sur un support de stockage déterminé (disque ou bande), chaque fois que le LGWR passe sur un nouveau groupe.
- N'est actif que si les fichiers de reprise sont utilisés en mode ARCHIVELOG

Dnnn

- Permet le partage d'un nombre limité de processus serveurs par les processus utilisateurs
- Le nombre de processus Dnnn (Dispatchers) est fixé par l'administrateur.
- Ces processus Dispatchers sont utilisés en configuration '*multi-threaded server*'

Listener

- Se met en attente des connexions des utilisateurs et les aiguille vers un processus Dispatcher.
- Ce processus Listener est utilisé en configuration '*multi-threaded server*'

RECO, LCKn, Parallel Query, SNPn

- RECO (Recoverer) résout les erreurs concernant une transaction distribuée
- LCKn (Lock) réalise le verrouillage inter-instance dans un système *parrallel server*
- Parallel Query (Pnnn) permet le parallélisme des requêtes
- SNPn (Snapshot Refresh) réalise les rafraîchissements automatiques des *snapshots* (tables répliquées en lecture seule)

Les processus serveurs

- Une requête SQL est traitée par un processus client et un processus serveur.
- Le rôle du processus serveur est:
  - d'analyser et exécuter les commandes SQL
  - transférer les blocs de données nécessaires du disque vers la SGA
  - communiquer les résultats de requêtes aux applications

Les architectures multiprocessus

- Architecture combinée:
  - pour chaque utilisateur, le processus utilisateur et le processus serveur sont combinés en un seul processus utilisateur; il n'y donc pas de processus serveur;
  - il faut que les processus de l'OS permette la séparation entre le code de l'applicatif et celui d'Oracle; c'est le cas de VMS et non d'Unix.

Les architectures multiprocessus

- Architecture à serveur dédié:
  - à chaque processus utilisateur correspond un processus serveur, dit processus serveur dédié. Les 2 processus sont séparés. Le processus utilisateur est généralement sur une station de travail, le processus serveur dédié sur la machine servant de serveur de données

Les architectures multiprocessus

- Architecture à serveur partagé (*multi-threaded server)*:
  - ici plusieurs processus utilisateurs peuvent partager un même processus serveur.

Gestion et surveillance de la base

- Outils de connexion:
  - sql dba (avant Oracle7.1)

  $ sqldba lmode=yes
  - server manager (à partir d'Oracle7.2)

  $ svrmgrl (en mode ligne)

Les états d'une instance

- SHUTDOWN
- NOMOUNT: instance démarrée, permet de créer la base; la zone mémoire et les processus ne sont encore associés à aucune base de données.
- MOUNT: fichier de contrôle ouvert pour cette instance, place dans un état de maintenance; l'administrateur peut administrer la base.
- OPEN: tous les fichiers définis dans le fichier de contrôle sont ouverts.