

# ARCHITECTURES DES SYSTÈME DE BASE DE DONNÉES

# Architectures des Système de base de données

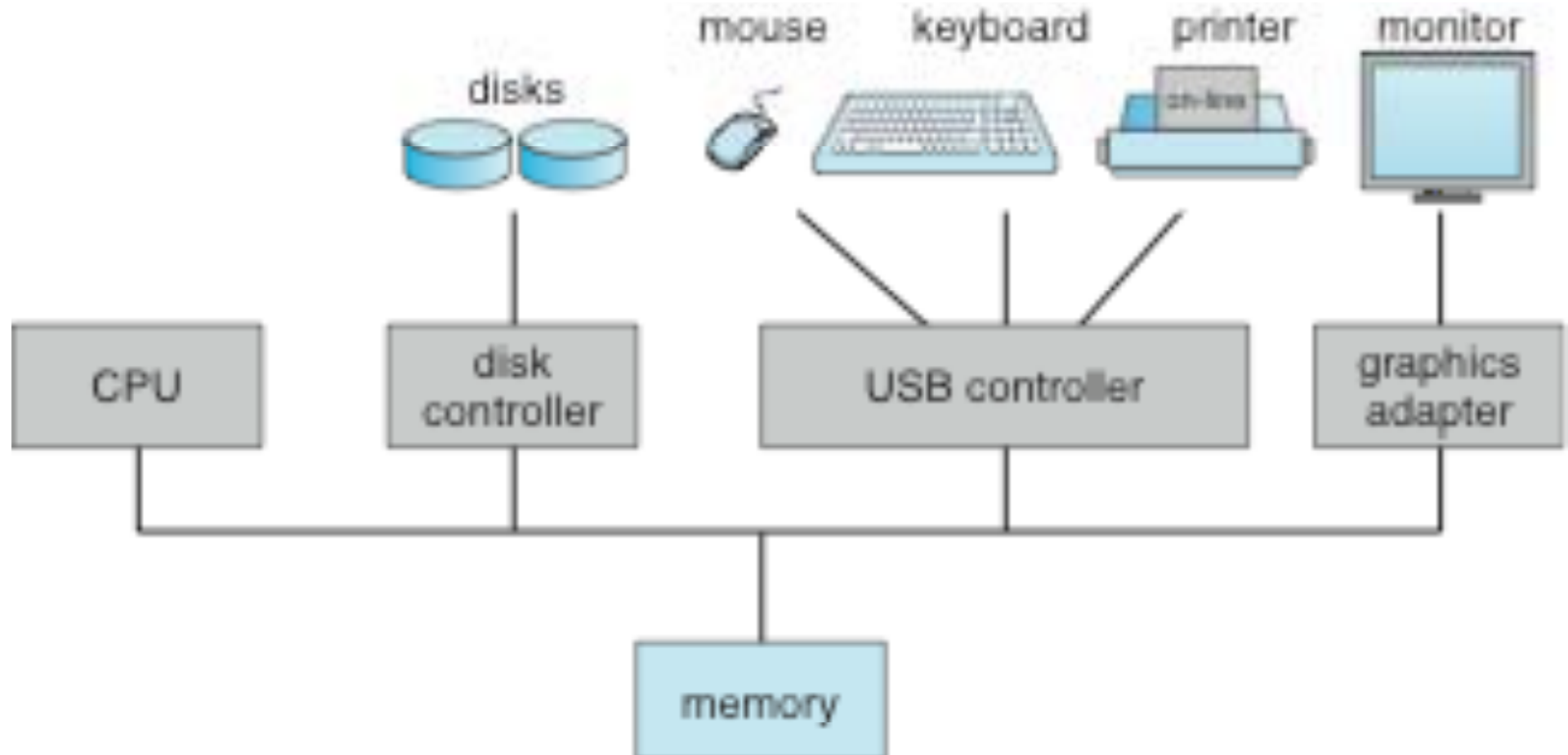


- Systèmes centralisés et client-serveur
- Server System Architectures
- Systèmes parallèles
- Systèmes distribués
- Types de réseau

# Les systèmes centralisés

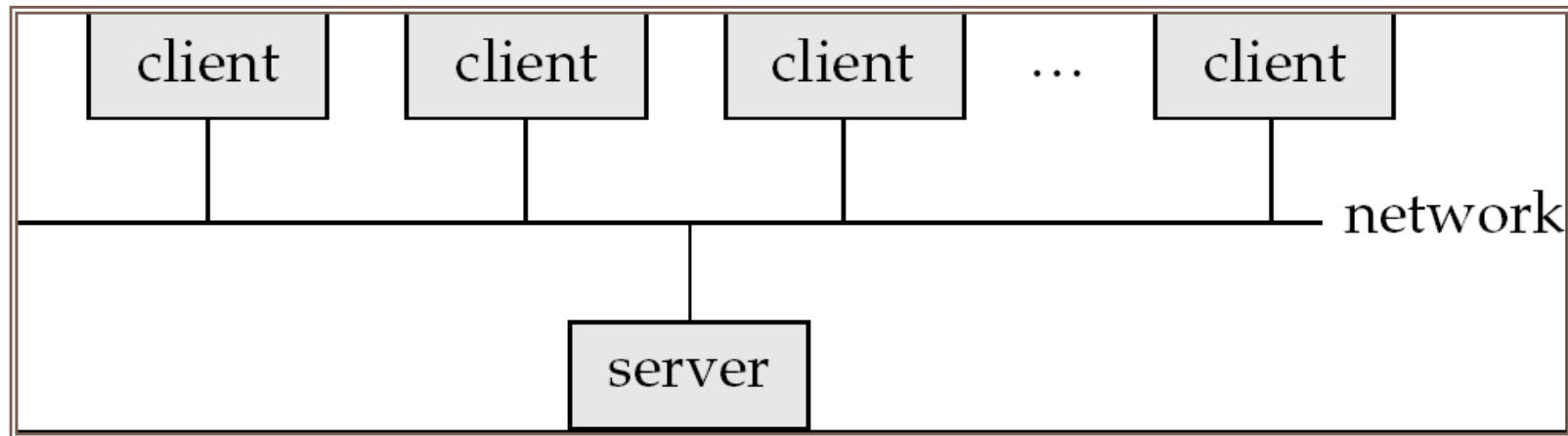
- Tournent sur un système informatique unique et n'interagissent pas avec d'autres systèmes informatiques.
- **système informatique à l'usage général:** un à deux CPU et un nombre de contrôleurs de périphériques qui sont connectés via un bus commun qui donne accès à la mémoire partagée.  
**système mono-utilisateur** (par exemple, un ordinateur personnel ou poste de travail): Une seule unité, un seul utilisateur, a généralement un seul processeur et un ou deux disques durs, l'OS peut prendre en charge un seul utilisateur.  
**système multi-utilisateurs:** plus de disques, plus de mémoire, plusieurs processeurs, et un système d'exploitation multi-utilisateurs. Servir un grand nombre d'utilisateurs qui sont connectés aux bornes de vie du système. Souvent appelé systèmes de serveurs.

# Les systèmes centralisés



# Systèmes client-serveur

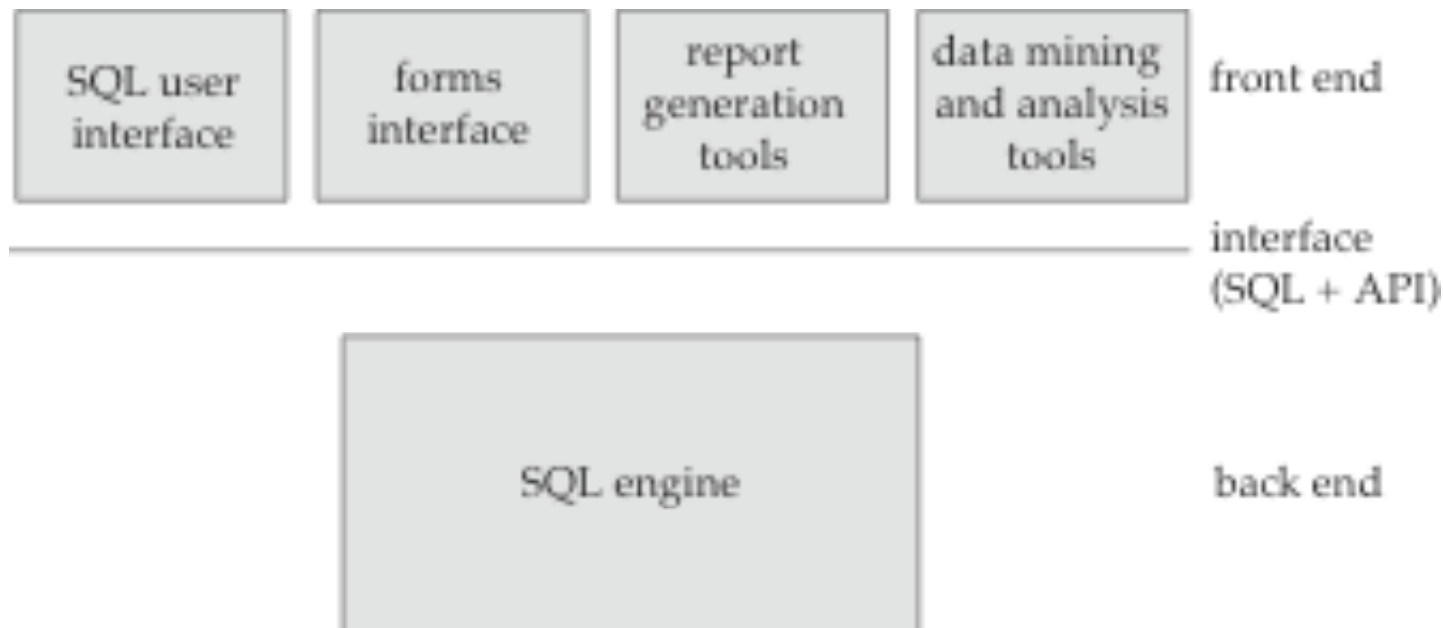
- systèmes Serveurs répondent aux demandes générées à des systèmes (m) clients, dont la structure générale est indiqué ci-dessous:



# Systemes client-serveur

- Fonctionnalité de base de données peut être divisé en:
- **Back-end:** gère des structures d'accès, évaluation et optimisation des requêtes, contrôle de concurrence et de récupération.
- **Front-end:** il comprend des outils tels que des formulaires, rapport-écrivains, des installations et l'interface utilisateur graphique.

L'interface entre le front-end et le back-end est par SQL ou par l'intermédiaire d'une interface de programme d'application.



# Systemes client-serveur



- ❑ une meilleure fonctionnalité pour le coût  
flexibilité dans la localisation des ressources et  
l'agrandissement des installations
- ❑ interfaces utilisateur plus meilleurs
- ❑ un entretien plus facile

# Architecture du système de serveur



systèmes de serveur peuvent être classés en deux types:

- **serveurs de transaction** qui sont largement utilisés dans les systèmes de base de données relationnelle, et
- **serveurs de données**, utilisés dans les systèmes de base de données orientée objet



# Serveurs de Transaction

- Également appelé serveur de requête ou de systèmes de systèmes de serveurs SQL

Les clients envoient des requêtes au serveur

Les transactions sont exécutées sur le serveur

Les résultats sont envoyés au client.

Les demandes sont spécifiées dans SQL, et communiquées au serveur via un appel de procédure distante (RPC) mécanisme.

Transactionnelle RPC permet à de nombreux appels RPC pour former une transaction.

Open Database Connectivity (ODBC) est un langage C application interface de programmation standard de Microsoft pour se connecter à un serveur, l'envoi de requêtes SQL, et les résultats de résidence.

JDBC standard est similaire à ODBC, pour Java

# Processus de Système de transaction

Un serveur de transaction typique se compose de plusieurs processus d'accéder aux données dans la mémoire partagée.

- Processus Serveur :

- Ceux-ci reçoivent les requêtes des utilisateurs (transactions), les exécuter et renvoient les résultats

Les processus peuvent être **multithreads**, ce qui permet d'un seul processus pour exécuter simultanément plusieurs requêtes de l'utilisateur

- En règle générale plusieurs serveurs multithread processus

- processus gestionnaire de verrous

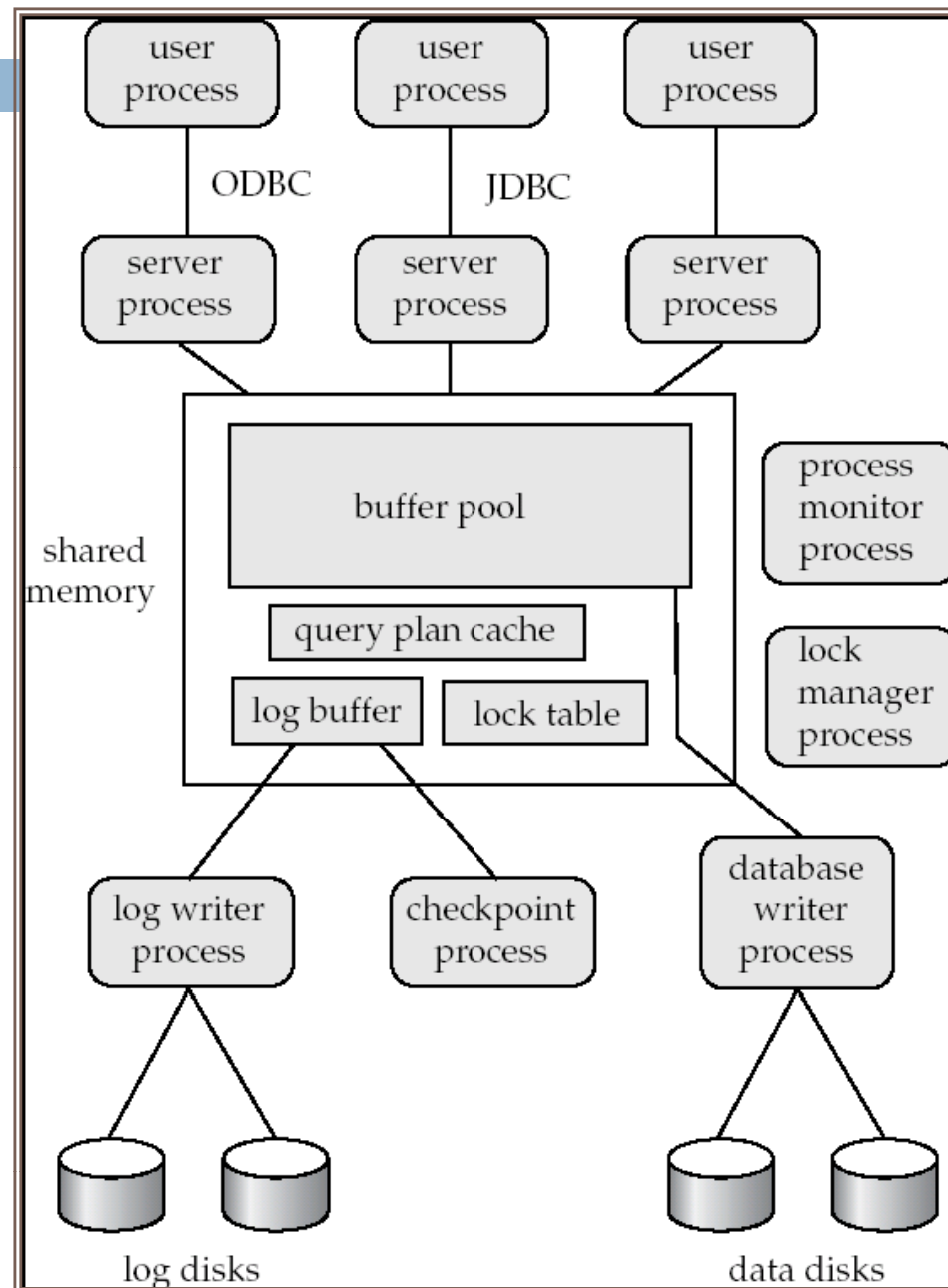
- processus d'écriture dans la base de données

- les blocs de mémoire tampon de sortie sont sans cesse modifiés dans le disques

# Processus de Système de transaction

- processus log writer
  - ▣ Sauvegarde dans les fichiers de journalisations (secours)
- processus de Checkpoint
  - Effectuer des points de contrôle périodique
- processus de surveiller de processus
  - ▣ Moniteurs d'autres processus, et prend des mesures de rétablissement si l'un des processus échec des autres
  - ▣ Par exemple, interrompre toutes les transactions en cours d'exécution par un processus du serveur et de le redémarrer

# Processus de Système de transaction



# Processus de Système de transaction

- Mémoire partagée, contient des données partagées
  - pool de mémoire tampon
  - Verrou de table  
mémoire tampon du journal
  - Les plans de requête mis en cache (réutilisé si même requête soumise à nouveau)
- Tous les processus de base de données peut accéder à la mémoire partagée
- Pour s'assurer que les deux processus ne sont pas accéder à la même structure de données au même temps, les systèmes de bases de données de mise en œuvre s'exclus mutuelle en utilisant soit
  - sémaphores Système d'exploitation
  - instructions atomique, comme test-and-set
- Pour éviter une surcharge de communication interprocessus pour la demande de verrouillage / subvention, chaque processus de base de données fonctionne directement sur la table de verrouillage
- Au lieu d'envoyer des demandes de verrouillage de Process Manager
  - Process Manager de verrouillage sera encore utilisée pour la détection de blocage

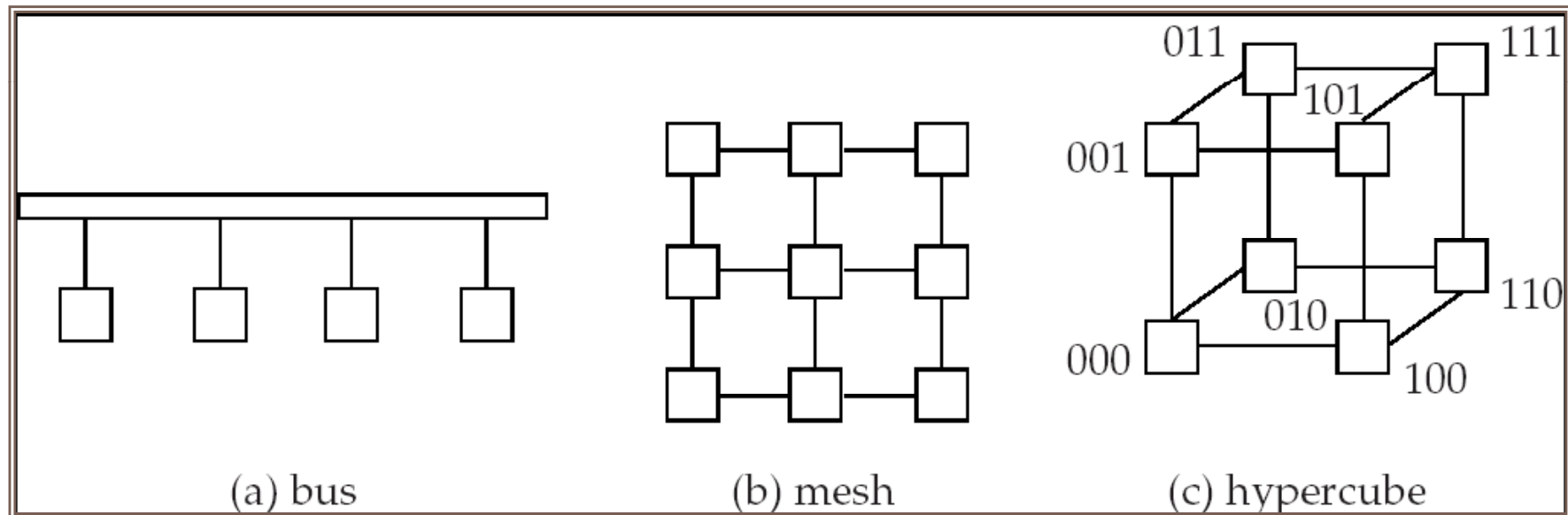
# Systèmes parallèles

- systèmes de base de données parallèle composé de plusieurs processeurs et plusieurs disques reliés par un réseau d'interconnexion rapide.
- Une machine parallèle à gros grain se compose d'un petit nombre de processeurs puissants  
Une machine massivement parallèle du grain fine utilise des milliers de processus
- Deux mesures principales de performances:
  - ▣ **débit** --- le nombre de tâches qui peuvent être exécutées dans un intervalle de temps donné
  - ▣ **temps de réponse** --- le temps qu'il faut pour accomplir une tâche unique à partir de moment où il est soumis

# Architectures des Réseau d'interconnexion

- **Bus.** Les composants du système permet d'envoyer des données et de recevoir des données à partir d'un bus de communication unique;
- Ne s'adapte pas bien avec un parallélisme plus en plus.
- **Mesh.** Les composants sont disposés comme des nœuds dans un réseau, et chaque composant est connecté à tous les composants adjacents
- **Hypercube.** Les composants sont numérotées en binaire, et les composants sont connectés les uns aux autres.

# Architectures des Réseau d'interconnexion

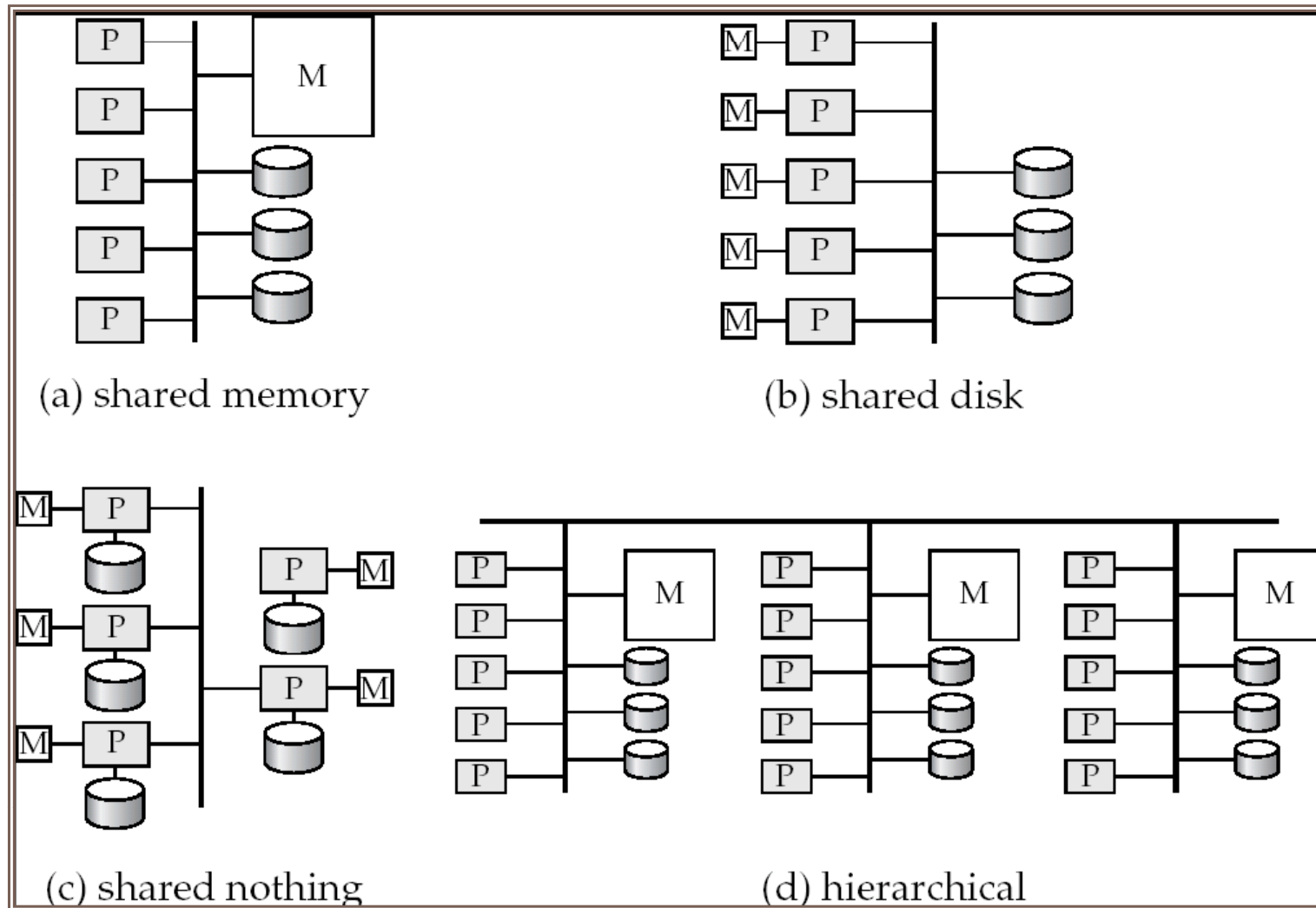




# Architectures des Base de Données Parallèles

- **Mémoire partagée** - processeurs partagent une mémoire commune
- **disque partagé** - processeurs partager un disque commun
- **partagée rien** - processeurs ne partagent ni une mémoire commune, ni de disque commun
- **Hiérarchique** - hybride de l'architectures ci-dessus

# Architectures des Base de Données Parallèles



# Mémoire partagée

- ❑ Les Processus et les disques ont accès à une mémoire commune, généralement via un bus ou par un réseau d'interconnexion.
- ❑ **Extrêmement efficace pour la communication entre les Processus** - les données dans la mémoire partagée peuvent être consultées par n'importe quel processeur sans avoir à le déplacer en utilisant un logiciel.
- ❑ **Inconvénients** - l'architecture n'est pas extensible au-delà de 32 ou 64 processeurs, depuis le bus ou le réseau d'interconnexion est un goulot d'étranglement
- ❑ Largement utilisé pour des degrés moindres de parallélisme (4 à 8).

# Disque partagé

- Tous les processeurs peuvent accéder directement à tous les disques via un réseau d'interconnexion, mais les processeurs ont la mémoire privée.
- Le bus mémoire n'est pas un goulot d'étranglement
- L'Architecture offre un degré de tolérance aux pannes - si un processeur échoue, les autres processeurs peuvent prendre plus de ses tâches depuis la base de données résidente sur les disques qui sont accessibles à partir de tous les processeurs.
- **Exemples:** IBM Sysplex et DEC clusters (qui fait maintenant partie de Compaq) en cours d'exécution Rdb (aujourd'hui Oracle RDB) ont été au début des utilisateurs commerciaux
- **Inconvénients:** goulot d'étranglement se produit maintenant à l'interconnexion avec le sous-système disque.
- systèmes de disque partagé peut s'adapter à un nombre un peu plus de processeurs, mais la communication entre les transformateurs est plus lente.

# Shared Nothing

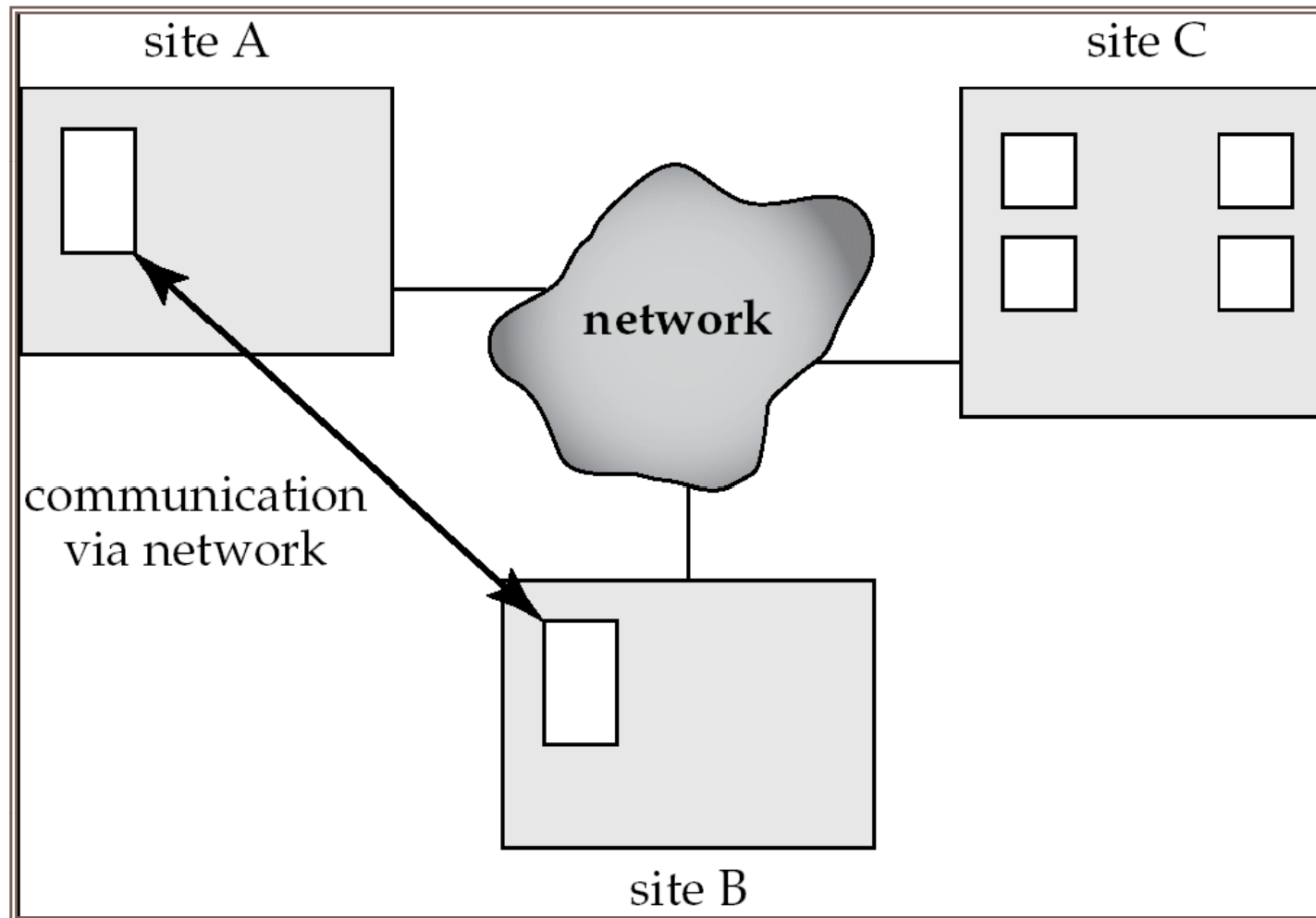
- Noeud se compose d'un processeur, la mémoire, et un ou plusieurs disques. Processeurs à un nœud de communiquer avec un autre processeur à un autre nœud utilisant un réseau d'interconnexion. Un nœud de fonctions en tant que serveur pour les données sur le disque ou le noeud possède des disques.
- **Exemples:** Teradata, Tandem, CUBE Oracle-n
- Les données accessibles à partir des disques locaux (et accède à la mémoire locale) ne passent pas par le réseau d'interconnexion, ce qui réduit l'interférence de partage des ressources.
- multiprocesseurs shared-nothing peuvent être étendus à des milliers de processeurs sans interférence.
- **inconvénient:** le coût de la communication et l'accès au disque non-locales; l'envoi de données implique une interaction de logiciels à ses deux extrémités.


# Hiérarchique

- Combine les caractéristiques de mémoire partagée, disques partagés, et des architectures shared-nothing.
- une architecture shared-nothing est de haut niveau - noeuds reliés par un réseau d'interconnexion, et les disques ne partagent pas la mémoire ou les uns avec les autres.
- Chaque noeud du système pourrait être un système à mémoire partagée avec quelques processus.
- Alternativement, chaque nœud peut être un système de disque partagé, et chacun des systèmes de partage d'un ensemble de disques pourrait être un système à mémoire partagée.
- Réduire la complexité de la programmation de tels systèmes par les architectures de mémoire virtuelle distribuée
- Aussi appelée architecture de mémoire non uniforme (NUMA)

# Systemes distribués

- diffusion de données sur plusieurs machines (aussi appelés sites ou des noeuds).
- Un Réseau relie les machines
- Les données partagées sont par les utilisateurs sur plusieurs machines



- 
- bases de données distribuées Homogène :
    - ▣ même logiciel / schéma sur tous les sites, les données peuvent être partagés entre les sites
    - ▣ Objectif: fournir une vue d'une base de données unique, cachant les détails de la distribu
  - Bases de données distribuées hétérogènes
    - ▣ Différents logiciels / schéma sur les différents sites
    - ▣ Objectif: intégrer les bases de données existantes pour fournir des fonctionnalités utiles
  - Distinguer entre les opérations locales et globales
    - ▣ Une transaction de données locales accède au site cours de laquelle l'opération a été amorcée.
    - ▣ Une transaction globale accède à des données dans un site différent de celui auquel la transaction a été amorcée ou accède à des données dans plusieurs sites différents.



# Compromis dans les systèmes distribués

- **Le partage des données** - les utilisateurs sur un site capable d'accéder aux données résidant à d'autres sites.
- **Autonomie** - chaque site est en mesure de conserver un certain degré de contrôle sur les données stockées localement.
- **la disponibilité du système supérieur grâce à la redondance** - les données peuvent être répliquées sur des sites distants, et le système peut fonctionner même si un site tombe en panne.
- **Inconvénient:** la complexité ajoutée requis pour assurer une bonne coordination entre les sites.
  - ▣ coût de développement de logiciels.
  - ▣ Grand potentiel pour les bugs.
  - ▣ Augmentation de charge de traitement.

# Types de réseau



- réseaux locaux (LAN) - composé de processeurs qui sont réparties sur de petites zones géographiques, comme un seul bâtiment ou à quelques bâtiments adjacents.
- réseaux étendus (WAN) - composé de processeurs répartis sur une vaste zone géographique.

# Types de réseau

- WAN avec une connexion en continu (par exemple l'Internet) sont nécessaires pour la mise en œuvre des systèmes distribués de base de données
- applications Groupware tels que Lotus Notes peut fonctionner sur les réseaux étendus avec une connexion discontinue:
  - Les données sont répliquées.
  - Mises à jour sont propagées à des répliques périodiquement.
  - Des copies des données peuvent être mis à jour indépendamment.
  - exécutions non sérialisable peut donc en résulter. La résolution est fonction de l'application.

FIN DU CHAPITRE



# COMPOSANTS DE L'ARCHITECTURE D'UN SGBD

# Généralité

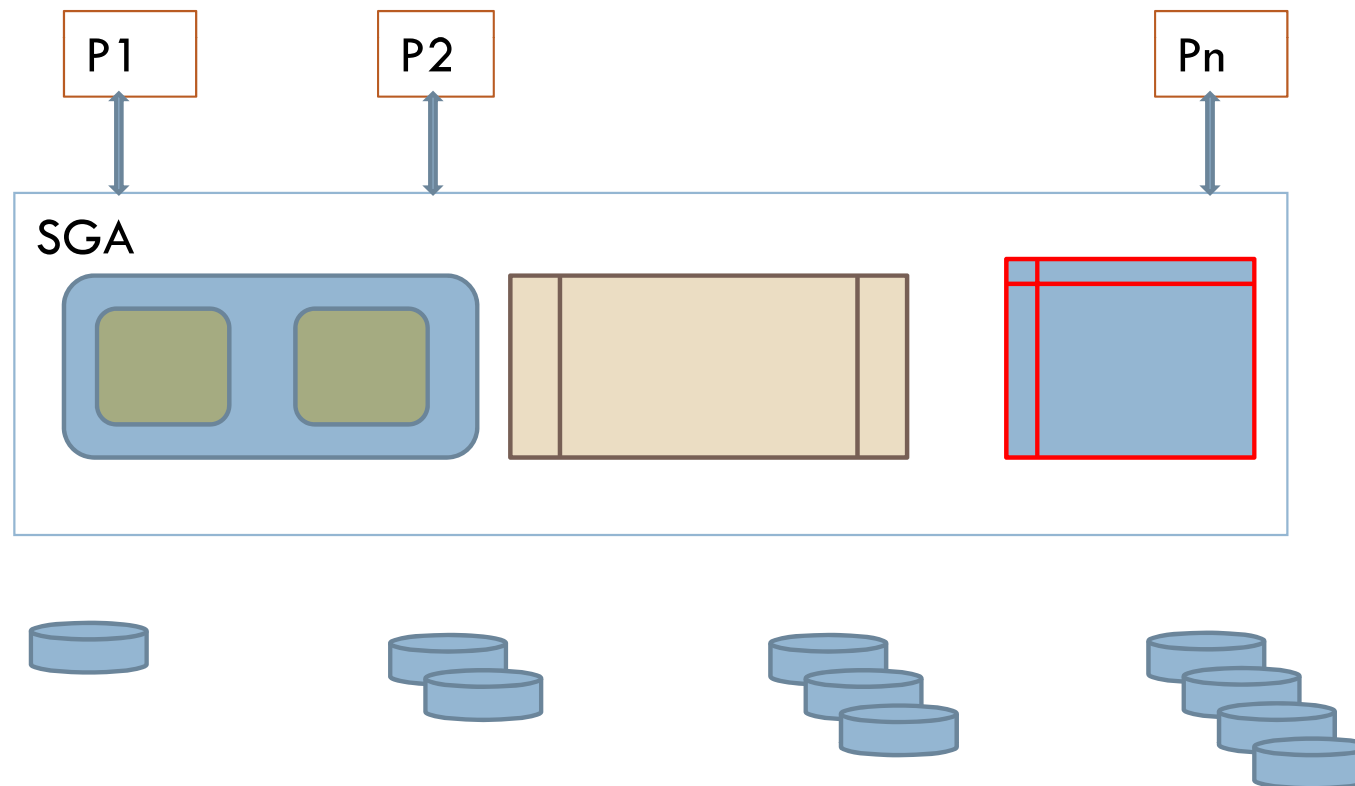
- 2
- Les composants principaux de l'architecture d'un SGBD

Sont:

Les processus

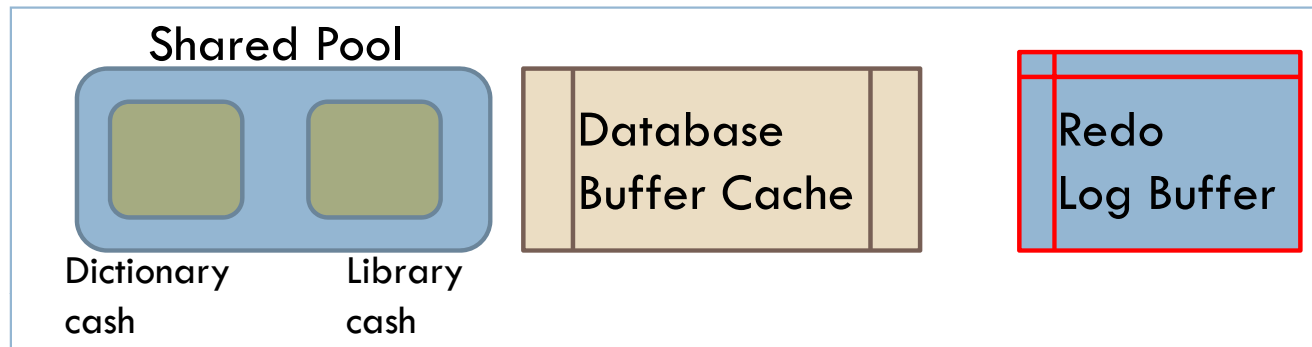
Les structures mémoires

Les fichiers



# Oracle SGA (System Global Area)

3



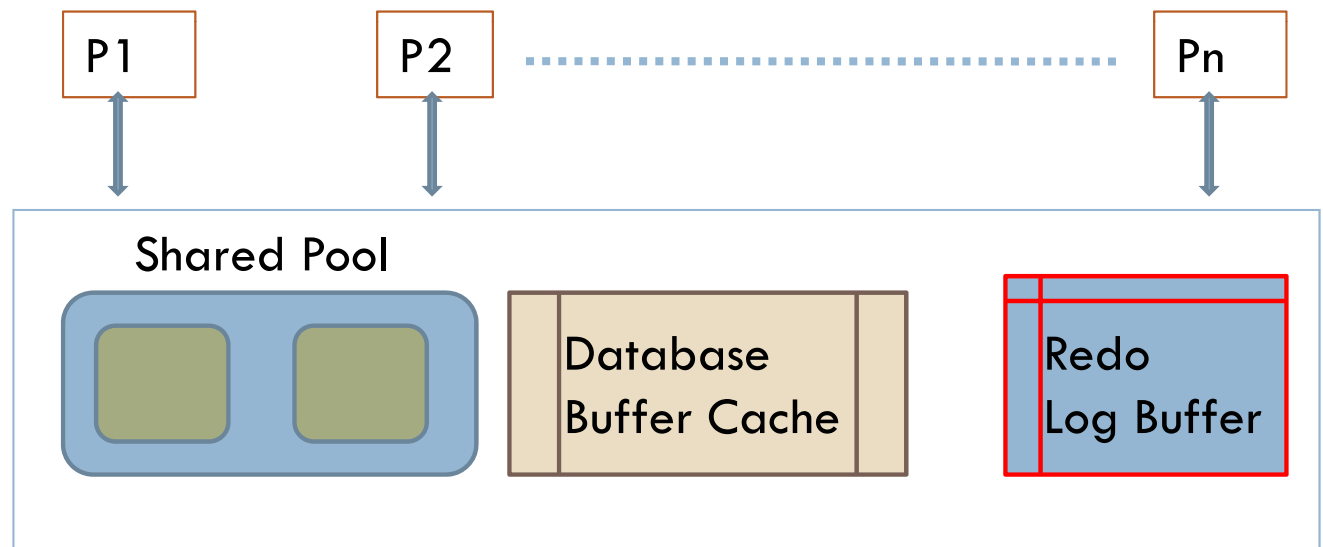
SGA est chargé au mémoire à chaque démarrage, c'est un groupe de mémoire partagé qui contient des données et informations de contrôle pour un SGBD.

Elle est appelé aussi **Shared Global Area**

# Les processus Background

4

- Database Writer (DBWR)
- Log Writer (LGWR)
- System Monitor (SMON)
- Process Monitor (PMON)
- Checkpoint (CKPT)
- Archive (ARCH)
- Recover (RECO)
- Lockn(LCKn)
- Snapshot Refresh(SNPn)
- Shared Server(Snnn)
- Dispatcher (Dnnn)
- Parallel Query

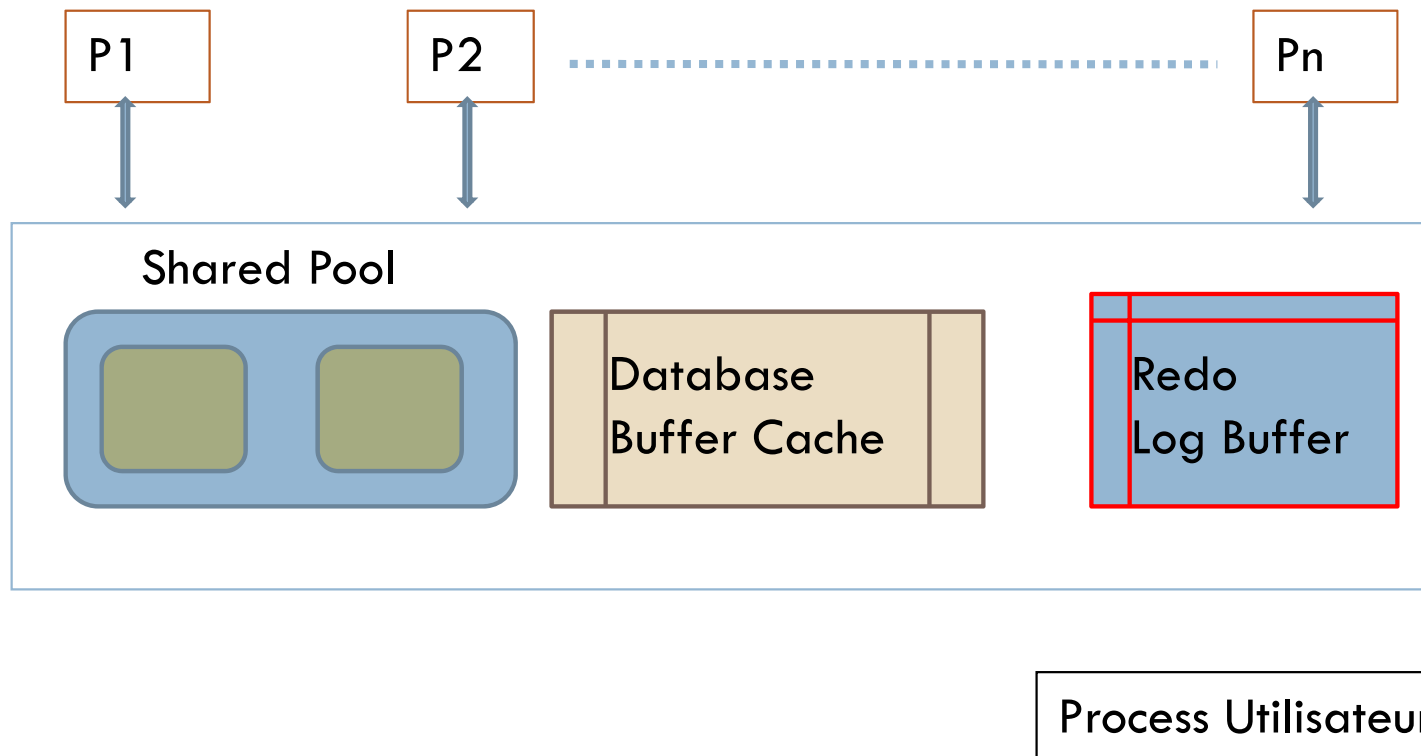


Selon la configuration et afin d'améliorer les performances et accueillir de nombreux utilisateurs, le système utilise un certain nombre de processus



# Les processus utilisateurs

5



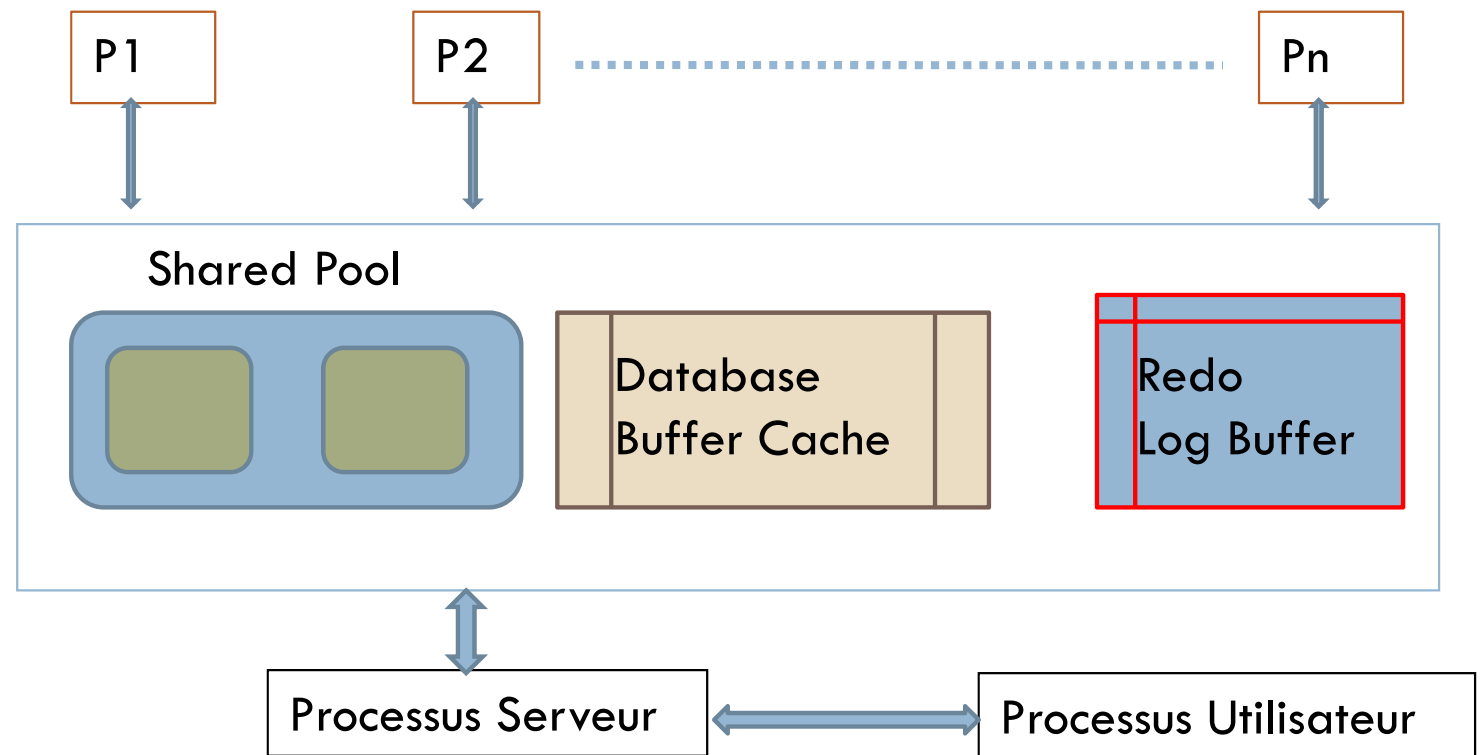
Un processus utilisateur est démarré lorsqu'un utilisateur exécute un programme applicatif

Il transmet le SQL au processus serveur et en reçoit les résultats

# Le processus serveurs

6

- Le processus serveur utilise de la mémoire partagée dans la SGA.
- Les tâches du processus serveur :
  1. Analyser et exécuter les commandes SQL.
  2. Lire les blocs de données à partir du disque dans les zones mémoires partagées de SGA.
  3. Renvoyer les résultats des commandes SQL au processus utilisateur

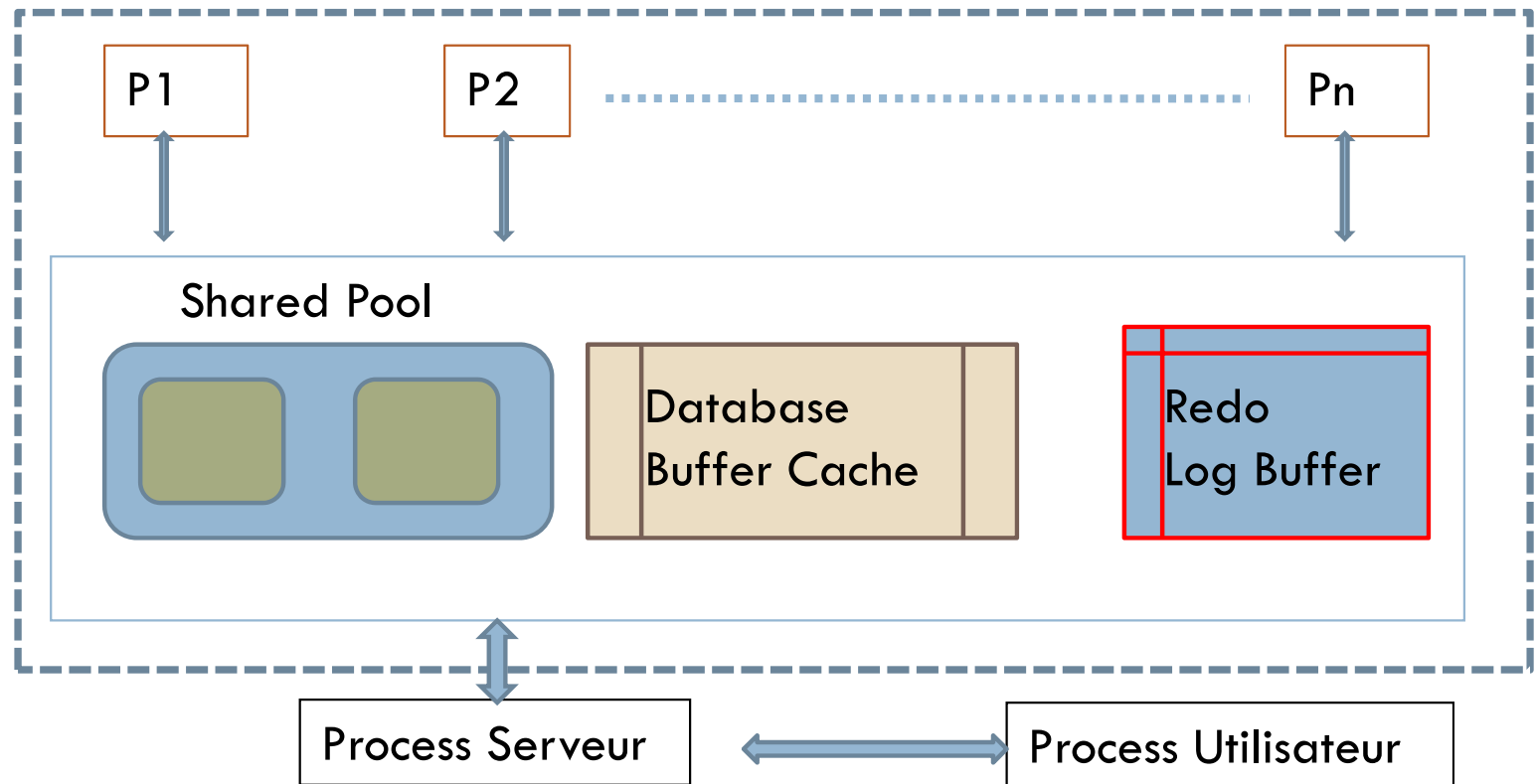


# L'instance de base de données

7

C'est la combinaison de la mémoire SGA et des processus détachés.

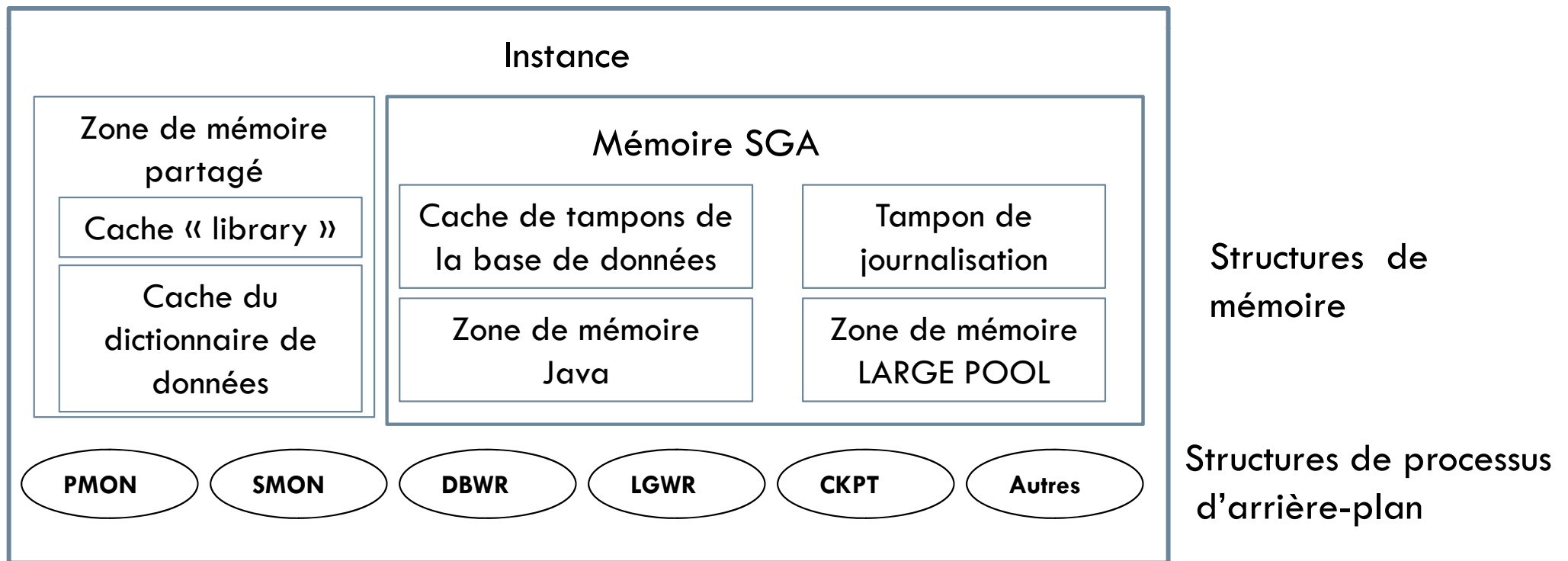
- Quand une instance est démarrée, les zones mémoires de la SGA sont allouées et les processus détachés sont lancés.
- L'instance est démarrée puis elle monte la base (ouverture des fichiers)



# Instance Oracle

8

- permet d'accéder à une base de données Oracle,
- n'ouvre qu'une seule base de données,
- est constituée de structures de processus d'arrière plan et de structures mémoire.

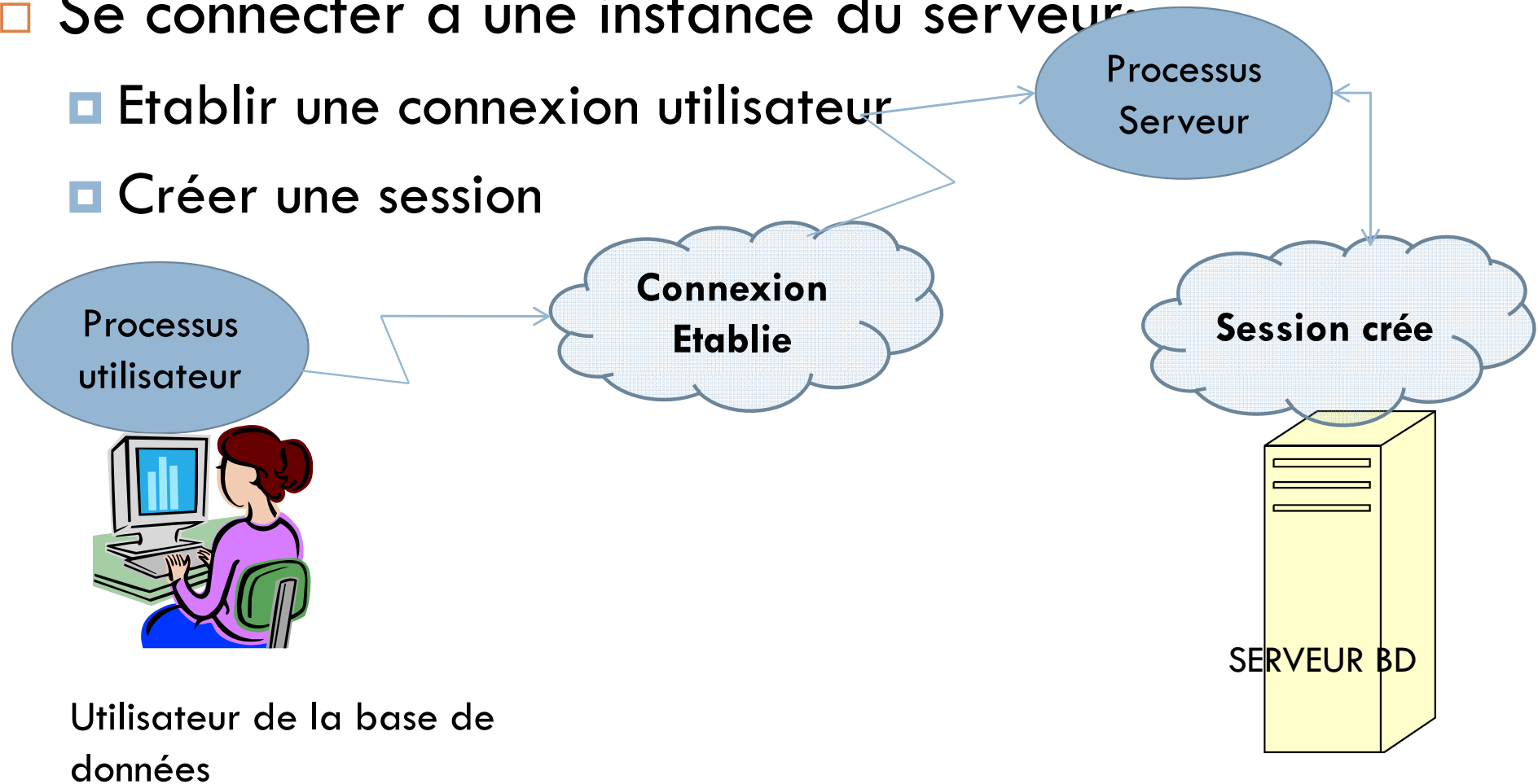


# Etablir une connexion et créer une session

9

## □ Se connecter à une instance du serveur

- ▣ Etablir une connexion utilisateur
- ▣ Créer une session



# Etablir une connexion et créer une session

10

- SQL\*PLUS , SQL DEVELOPPER sont des exemples de processus utilisateurs
  - ▣ Lorsqu'un utilisateur se connecte au serveur, processus serveur sera crée sur l'ordinateur qui exécute le serveur.
- **Connexion:** c'est un chemin de communication entre un processus utilisateur et un serveur.
  - ▣ IPC(interprocessus communication) du système d'exploitation hôte.
  - ▣ Client-serveur,
  - ▣ Une communication à 3 niveaux (three-tier)
- Une **session** est une connexion spécifique d'un utilisateur à un serveur. Elle démarre lorsque l'utilisateur est authentifié par le serveur. Un utilisateur peut ouvrir plusieurs sessions en parallèle s'il se connecte simultanément à partir de plusieurs outils, application ou terminaux.

# Base de données

11

Structure physique	Définition
Fichiers de données (1)	Contiennent toutes les données de la base; les structures logiques telles que les tables et index
Fichiers Redo Log(2) (journalisation)	Gardent des enregistrement de toutes les modifications effectuées sur BD dans un but de restauration
Fichiers de contrôle	Conservent la définition des structures physiques et les états de BD.

- Une base de données oracle utilise d'autres fichiers qui ne font pas partie de la base de données:
- Le fichier de paramètres définit les caractéristiques d'une instance de BD.
- Le fichier de mots de passe.
- Les fichiers de journalisation archivés

# Disponibilité de la base

12

Pour assurer une haute disponibilité de la base il est possible de:

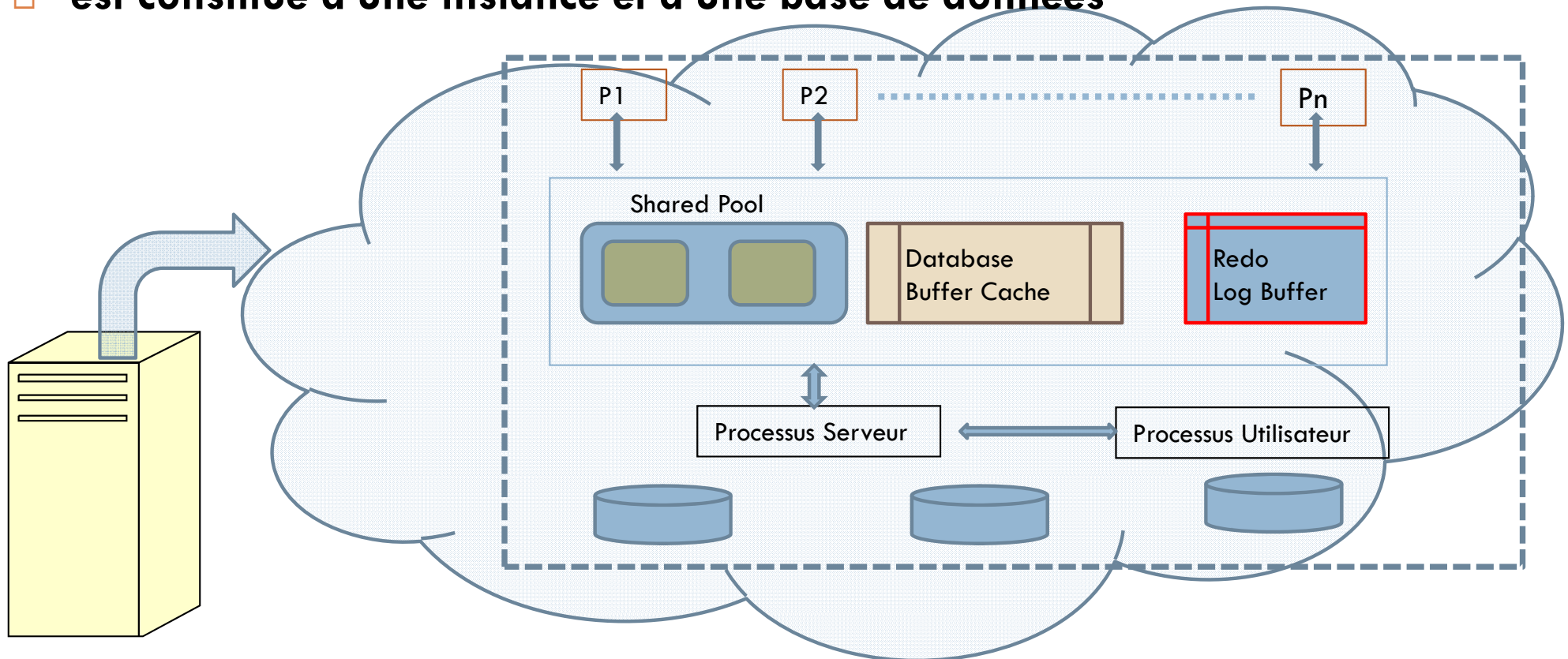
- **Multiplexer les fichiers de contrôle.** En cas de problème avec un fichiers, la restauration de ce fichier et de l'instance ne dure que quelques minutes.
- **Multiplexer les fichiers journaux :** En cas de problème avec un fichiers, la restauration de ce fichier peut se faire sans arrêt de l'instance.
- **Archiver les fichiers journaux.** Les archives de journaux servent en cas de perte d'un ou plusieurs fichiers de la base et permettent de restaurer la base jusqu'au moment de la perte.
- **Effectuer les sauvegardes en ligne,** base ouverte. Ces sauvegardes consistent en copies de fichiers séquentiels qui ne nécessitent pas de coordination avec les transactions en cours.



# Serveur de base de données

13

- C'est un système de gestion de base de données qui offre une méthode de gestion des informations ouverte, complète et intégrée,
- est constitué d'une instance et d'une base de données



**Il doit empêcher tout accès non autorisé et proposer des solutions efficaces de récupération des données après incident.**

# La structure physique

14

- ❑ **La structure physique d'une base de données comprend trois types de fichier :**
- ❑ **Fichiers de contrôle**
- ❑ **Fichiers de données**
- ❑ **Fichiers de journalisation**

## Structure mémoire

**La structure mémoire d'oracle est constituée des deux zones de mémoire suivantes:**

- **La mémoire SGA qui est allouée au démarrage de l'instance et qui est une composante fondamentale d'une instance.**
- **La mémoire PGA, qui est allouée au démarrage du processus serveur.**

# Mémoire SGA

15

- La mémoire **SGA** est constituée de plusieurs structures mémoire :
  - ▣ la zone de mémoire partagée,
  - ▣ le cache de tampons de la base de données,
  - ▣ le tampon de journalisation,
  - ▣ d'autres structures (gestion des verrous externes (lock) et des verrous internes (latch), données statistiques, par exemple).
- Deux structures mémoire supplémentaires peuvent également être configurées dans la mémoire **SGA** :
  - ▣ la zone de mémoire LARGE POOL,
  - ▣ la zone de mémoire Java.

Taille définie à l'aide du paramètre **SGA – MAX\_SIZE**

# Zone de mémoire partagée

16

- Elle permet de stocker :
  - ▣ Les dernières instructions SQL exécutées,
  - ▣ Les dernières définitions de données utilisées.
- Elle est constituée de deux structures mémoire liées aux performances :
  - ▣ Cache "library"
  - ▣ Cache du dictionnaire de données
- Sa taille est définie par le paramètre
- **SHERD\_POOL\_SIZE.**

```
SQL> ALTER SYSTEM SET  
SHERD_POOL_SIZE= 64M
```

# Cache "library »

17

- **Le cache "library" conserve des informations sur les dernières instructions SQL et PL/SQL utilisées**
- **Il permet le partage des instructions fréquemment utilisées.**
- **Il est géré par un algorithme LRU (Least Recently Used).**
- **Il est composé de deux structures :**
  - la zone SQL partagée,
  - la zone PL/SQL partagée.
- **Sa taille dépend du dimensionnement de la zone de mémoire partagée.**

# Cache du dictionnaire de données

18

- Le cache du dictionnaire de données contient les dernières définitions utilisés dans la base
- Il contient des informations sur les fichiers, les tables, les index, les colonnes, les utilisateurs, les privilèges et d'autres objets de la base de données.
- Au cours de l'analyse, le processus serveur recherche les informations dans le cache du dictionnaire pour résoudre les noms d'objet et valider l'accès.
- La mise en mémoire cache des informations du dictionnaire de données réduit le temps de réponse aux interrogations et aux instructions LMD.
- La taille du cache dépend du dimensionnement de la
- zone de mémoire partagée.

# Cache de tampons de la base de données

19

- ❑ **Ce cache conserve des copies des blocs de données extraits des fichiers de données.**
- ❑ **Il permet des gains de performances considérables lors de l'obtention et de la mise à jour de données**
- ❑ **Il est géré par un algorithme LRU.**
- ❑ **Le paramètre DB - BLOCK - SIZE détermine la taille du bloc principal.**

# Tampon de journalisation

20

- ❑ Il enregistre toutes les modifications apportées aux blocs de données de la base.
- ❑ Sa principale fonction est la récupération de données.
- ❑ Les modifications enregistrées constituent des entrées de journalisation.
- ❑ Les entrées de journalisation contiennent des informations permettant de reconstruire des modifications.
- ❑ La taille du tampon est définie par le paramètre LOG-BUFFER



# Zone de mémoire LARGE POOL

21

- ❑ **Zone facultative de la mémoire SGA**
- ❑ **Elle réduit la charge de la zone de mémoire partagé**
- ❑ **La mémoire allouée par session (UGA) au serveur partagé**
- ❑ **les processus serveur d'E/S**
- ❑ **les opérations de sauvegarde et de restauration RMAN (Recovery Manager)**
- ❑ **Elle n'utilise pas de liste LRU.**
- ❑ **Sa taille est définie par le paramètre** `LARGE-POOL-SIZE`

# Zone de mémoire Java

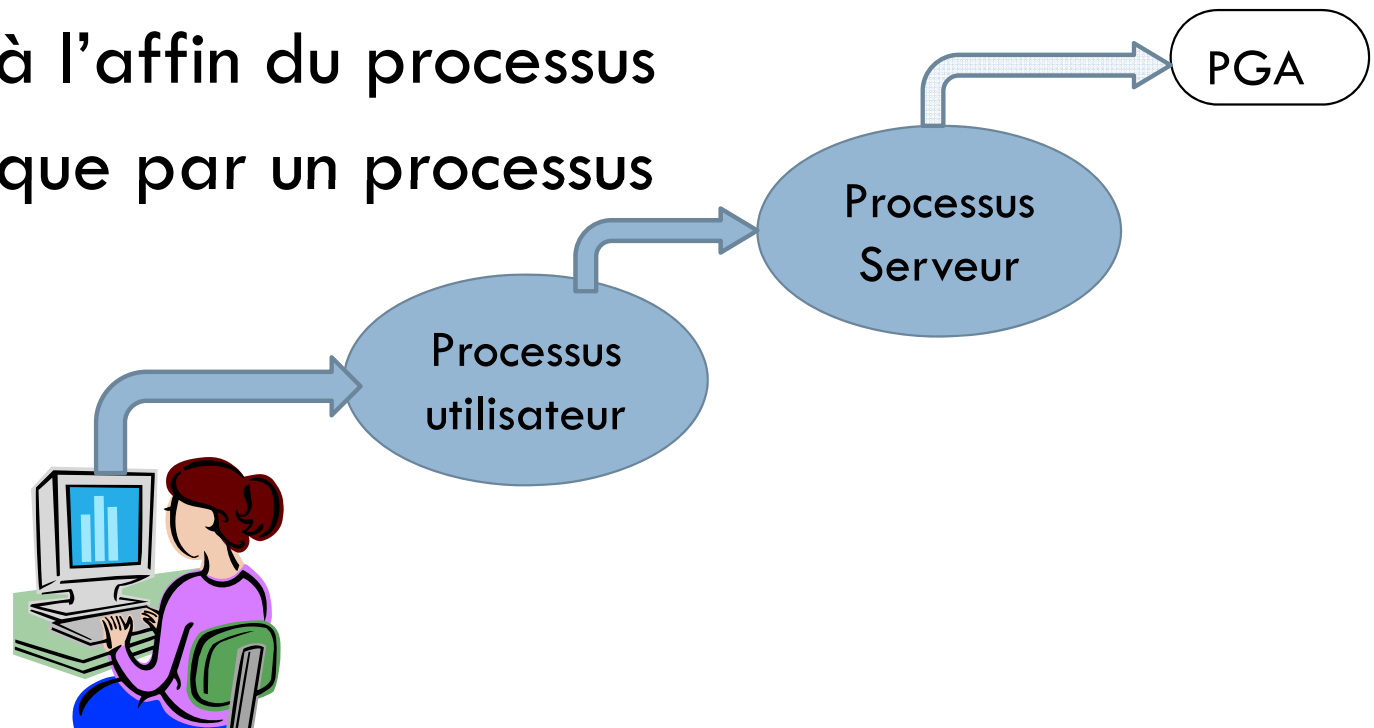
22

- **La zone de mémoire Java répond aux besoins d'analyse des commandes Java.**
- **Elle est nécessaire si Java est installé et utilisé.**
- **Sa taille est définie par le paramètre JAVA-POOL-SIZE.**

# Mémoire Program Global Area (PGA)

23

- Mémoire réservée à chaque processus utilisateur qui se connecte à une base de données Oracle.
- Elle est allouée lorsque un processus est créé
- Elle est libérée à l'arrêt du processus
- Elle n'est utilisée que par un processus



# Structure de processus

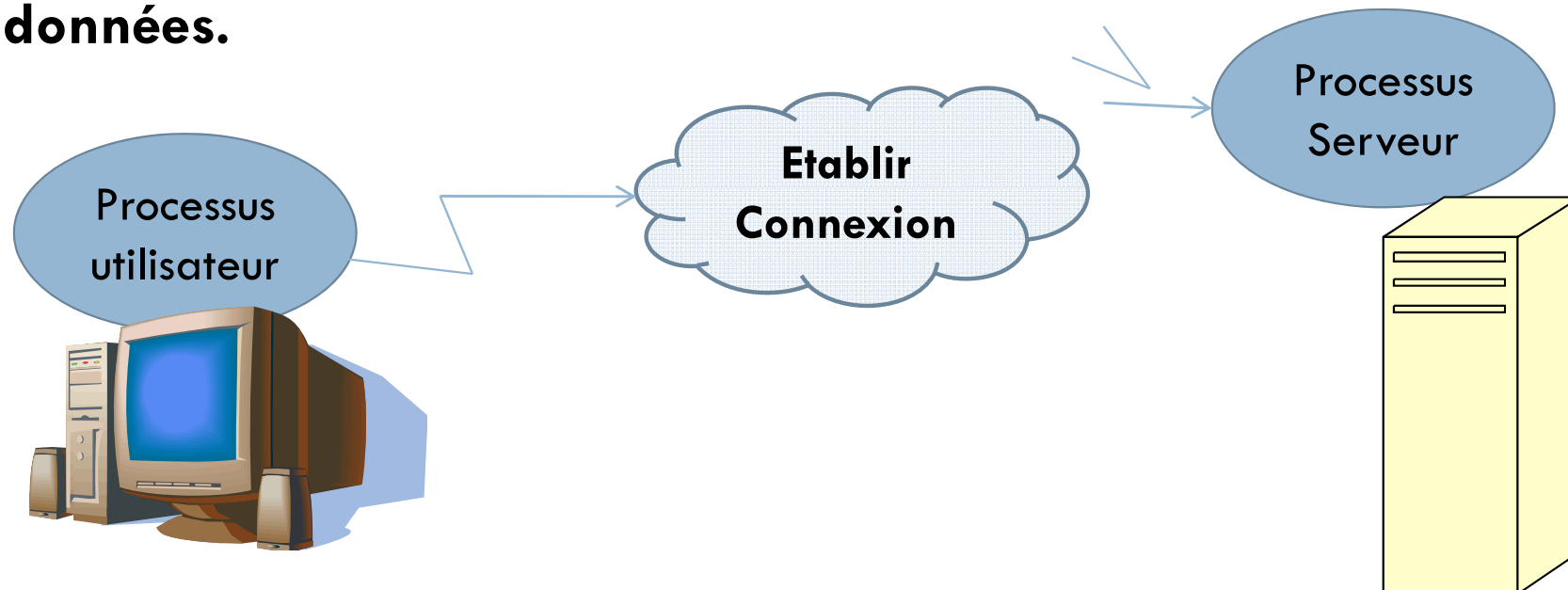
24

- Les différents types de processus sont:
- le processus utilisateur, qui est démarré au moment où un utilisateur de la base de données tente de se connecter au serveur ,
- le processus serveur, qui établit la connexion à l'instance et démarre lorsqu'un utilisateur ouvre une session,
- les processus d'arrière-plan, lancés au démarrage d'une instance.

# Processus utilisateur

25

- Programme qui demande une interaction avec le serveur de base de données.
- Ce processus doit d'abord établir une connexion. .
- Il n'entre pas directement en interaction avec le serveur base de données.

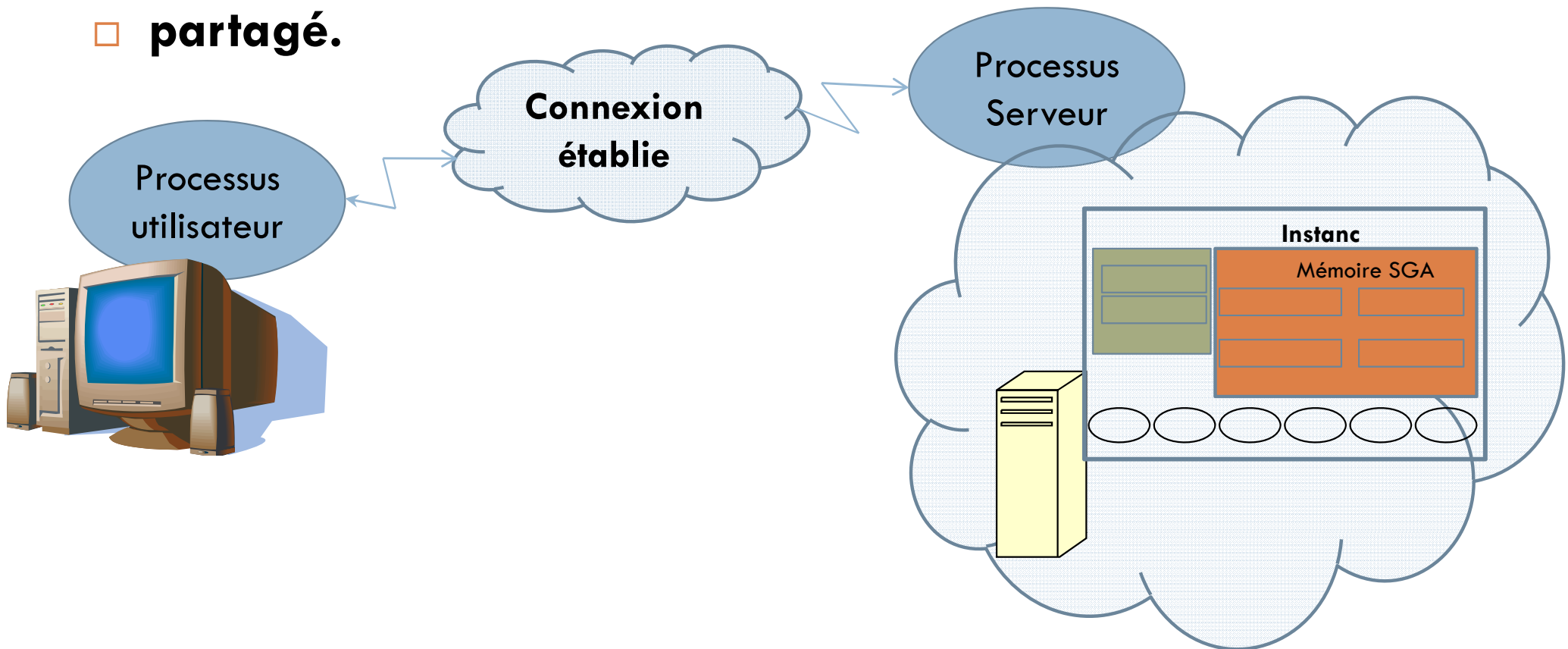


processus utilisateur génère des appels via l'interface UPI (User Program Interface) qui crée une session et démarre **un processus serveur**

# Processus serveur

26

- Programme qui entre directement en interaction avec le serveur de base de données.
- Il répond aux appels générés et renvoie les résultats.
- Il peut s'agir d'un serveur dédié ou d'un serveur
- partagé.



# Processus d'arrière-plan

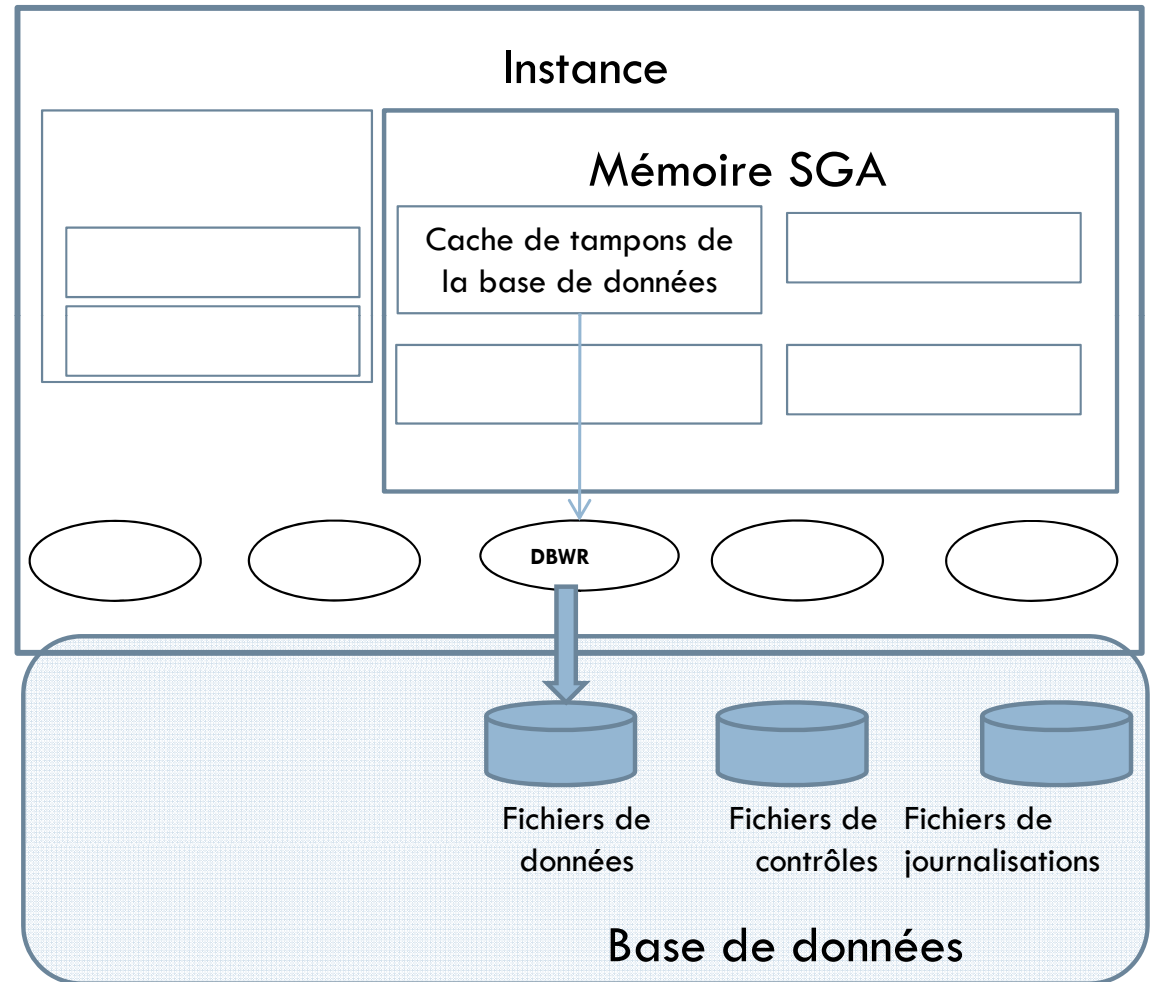
27

- **Gèrent et appliquent les relations entre les structures physiques et les structures mémoire.**
- **Processus d'arrière-plan obligatoires**
  - ▣ **DBW<sub>n</sub> PMON CKPT**
  - ▣ **LGWR SMON**
- **Processus d'arrière-plan facultatifs**
  - ▣ **ARC<sub>n</sub> LMD<sub>n</sub> RECO**
  - ▣ **CJQO LMON Snnn**
  - ▣ **Dnnii Pnnn**
  - ▣ **LCK<sub>n</sub> QMN<sub>n</sub>**

# Processus database writer (DBWn)

28

- Le processus enregistre les modifications des blocs d'annulation et des blocs de données dans le cache de tampons de la base de données.
- Le processus DBWn écrit les tampons "dirty" du cache de tampons de la base de données dans les fichiers de données.

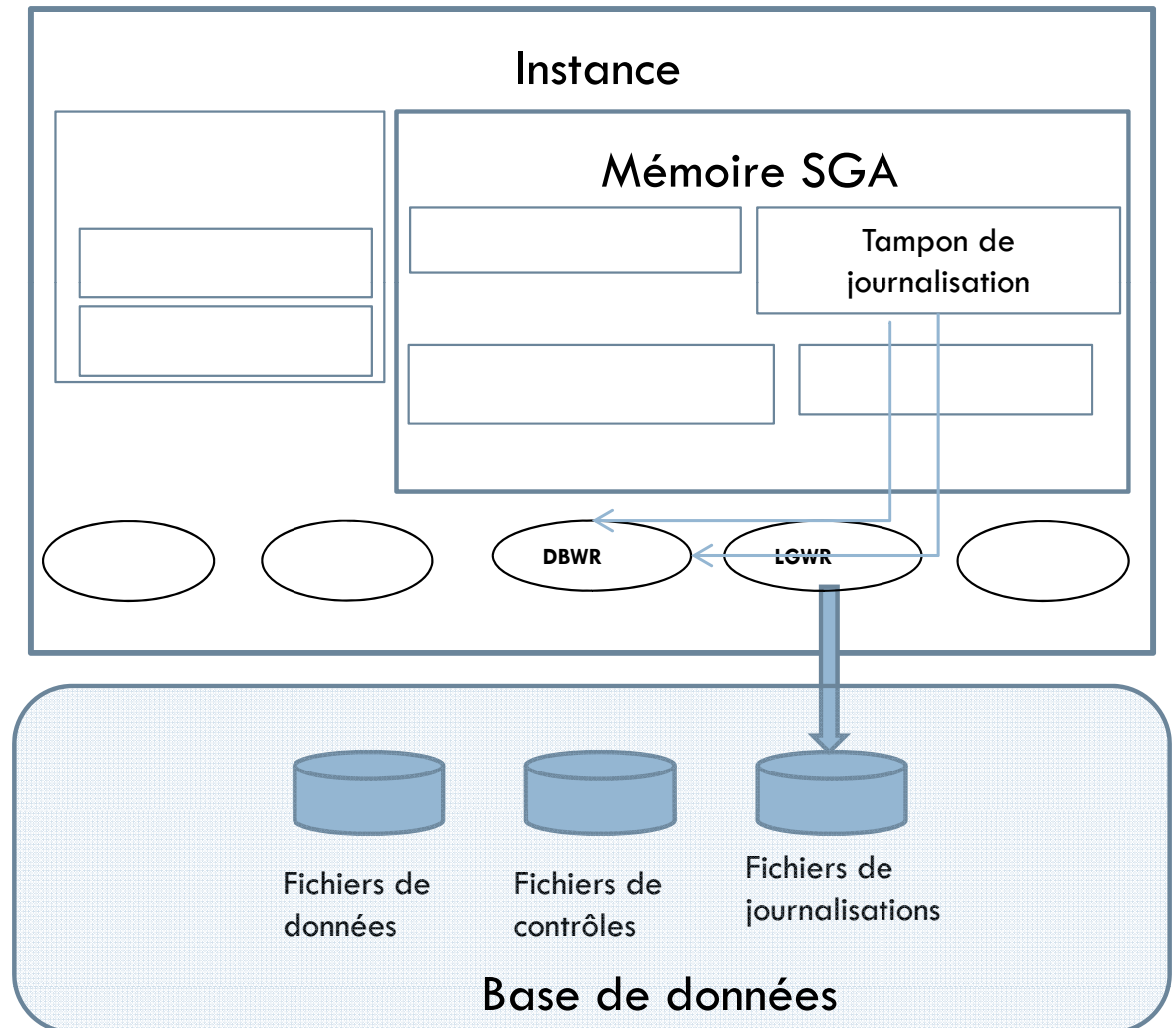




# Processus LGWR (Log Writer)

29

- ❑ **LGWR écrit dans les cas suivants:**
- ❑ **Validation (commit)**
- ❑ **Un tiers du cache est occupé**
- ❑ **La journalisation atteint 1M**
- ❑ **toutes les trois secondes**
- ❑ **avant que le processus DBWn ne procède à une opération d'écriture**

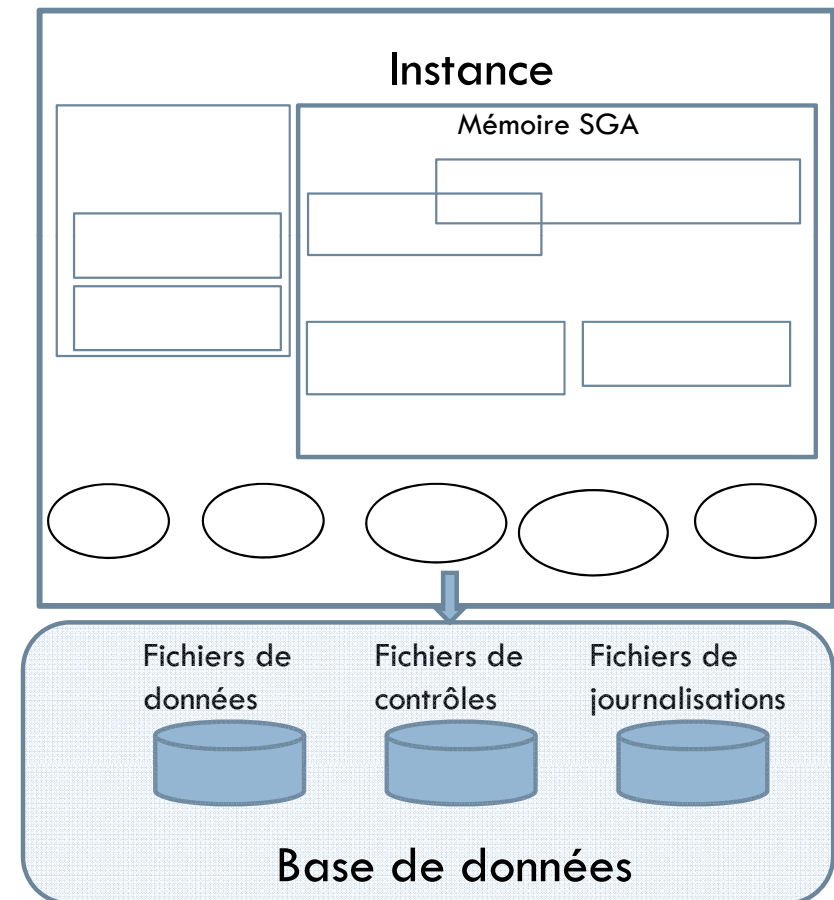


# Processus SMON (System Monitor)

30

Lorsqu'une instance a échoué, le processus d'arrière-plan SMON la récupère automatiquement lors de la réouverture de la base de données.

- **Responsabilités :**
- **Récupération de l'instance :**
  - ▣ réimplémente des modifications dans les fichiers de *journalisation*,
  - ▣ ouvre la base de données pour permettre l'accès aux utilisateurs,
  - ▣ annule les transactions non validées.
- **Fusion de l'espace libre**
- **Libération des segments temporaires**



# Processus PMON (Processus Monitor)

31

**Suite à l'échec de l'exécution du processus, PMON exécute les opérations de nettoyage :**

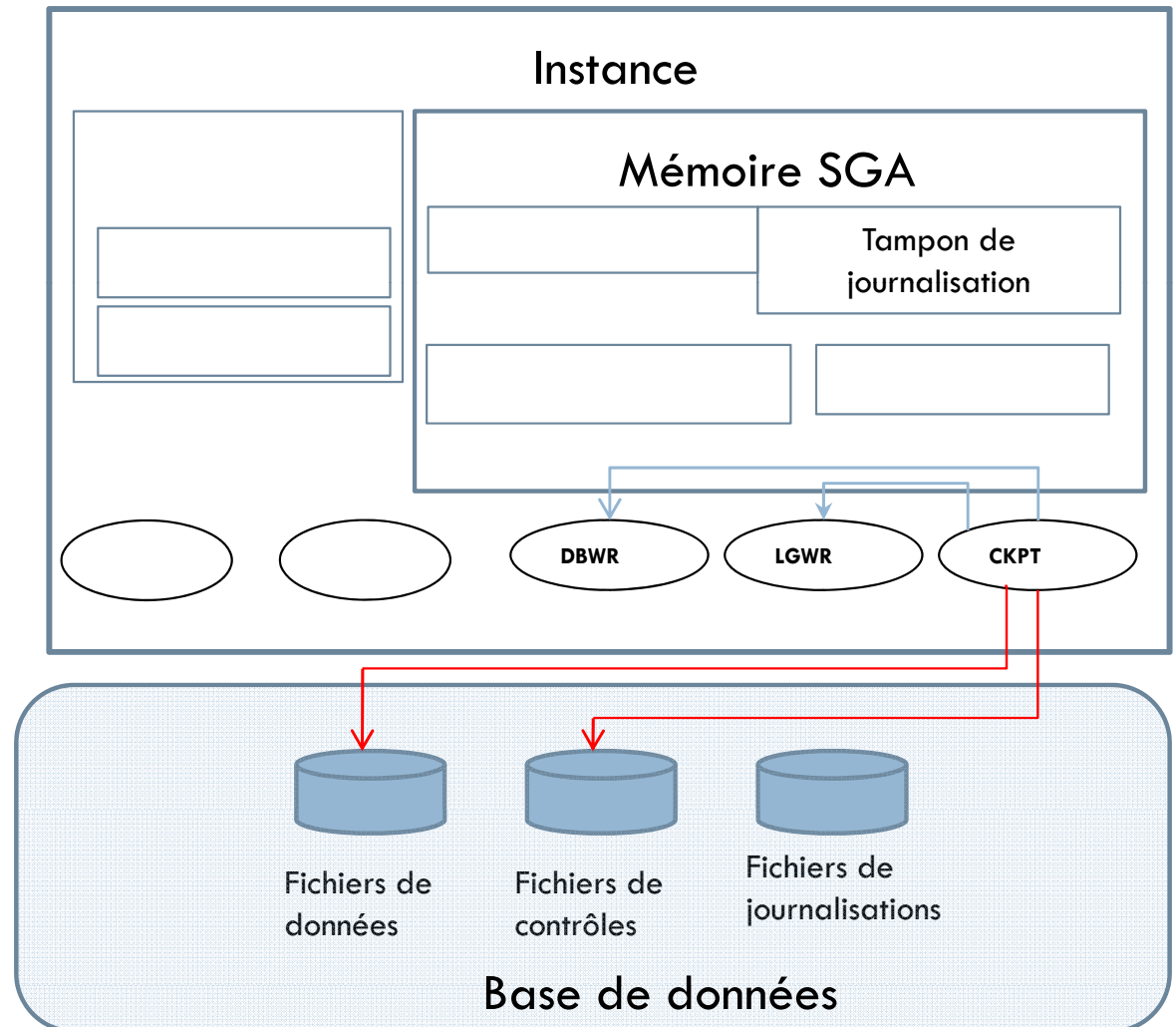
- ▣ annule la transaction en cours de l'utilisateur**
- ▣ libère des verrous sur des tables ou de lignes**
- ▣ libère d'autres ressources réservés par l'utilisateur**
- ▣ redémarre les répartiteurs interrompus**

# Processus CKPT (Checkpoint)

32

**Ce processus est chargé :**

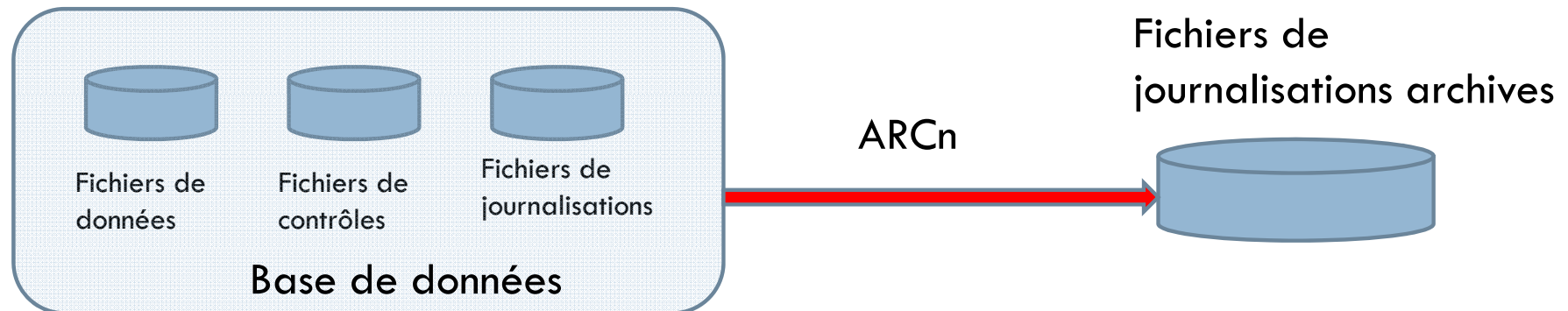
- de signaler DBWn aux points de reprise,
- de mettre à jour les entêtes de fichiers de données avec les informations sur le point de reprise,
- de mettre à jour les fichiers de contrôle avec les informations sur le point de reprise.



# Processus ARCn (processus d'archivage)

33

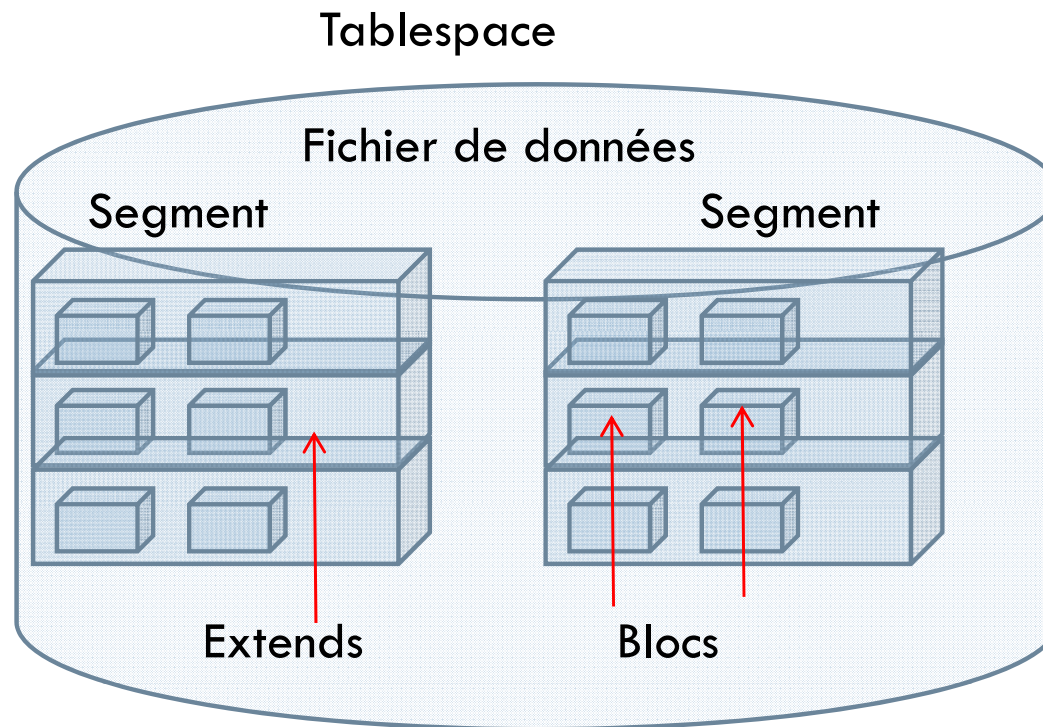
- Processus d'arrière plan facultatif
- Il joue un rôle essentiel dans la récupération d'une base suite à une défaillance du disque
- En mode ARCHIV\_LOG Il archive automatiquement les fichiers de journalisation en ligne
- Il enregistre toute les modifications apportées à la base de données.



# Structure logique

34

- La structure logique définit le mode d'utilisation de l'espace physique d'une base de données
- Cette structure possède une hiérarchie composé de tablespaces, de segments, d'extends et de blocs.



# Traiter les instructions SQL

35

Connexion à une instance via :

- ▣ le processus utilisateur,
- ▣ le processus serveur.

Les composants du serveur Oracle utilisés dépendent du type d'instruction SQL :

- ▣ Les interrogations renvoient des lignes.
- ▣ Les instructions LMD consignent les modifications.
- ▣ La validation garantit la récupération de la transaction.
- ▣ Certains composants du serveur Oracle n'interviennent pas dans le traitement des instructions SQL.

# GÉRER UNE INSTANCE ORACLE



# Fichiers de paramètres d'initialisation

2

- ▶ Les entrées sont propres à l'instance en cours de démarrage
- ▶ Il existe deux types de paramètres:
  - ❑ Explicite - le fichier contient une entrée.
  - ❑ Implicite – le fichier ne contient aucune entrées, mais prend en compte les valeurs par défaut.
- ▶ Une instance peut présenter plusieurs fichiers de paramètres d'initialisation
- ▶ Les modifications apportées aux entrées du fichier s'appliquent en fonction du type de fichier de paramètres d'initialisation utilisé:
  - ❑ fichiers de paramètres statique, PFILE (initSID.ora)
  - ❑ fichier de paramètres persistant, SPFILE (spfileSID.ora)

# fichiers de paramètres statique, PFILE (initSID.ora)

3

- Il s'agit d'un fichier texte
- Il peut être modifié à l'aide d'un éditeur du système d'exploitation.
- Toute modification est apporté manuellement
- Les modification sont effectives au démarrage suivant
- Il ne peut être ouvert que lors du démarrage de l'instance
- Sont répertoire par défaut est  
\$ORACLE\_HOME/dbs sous le nom initSID.ora

# Créer un fichier PFILE

4

- Créer ce fichier à partir d'un exemple de fichier init.ora
- Ce fichier est créer au cours de l'installation
  - ❑ Copier le fichier vers  
\$ORACLE\_HOME/dbs/initdba01.ora
- Modifier le fichier initSID.ora
  - ❑ Editer les paramètres
  - ❑ Affectez les valeurs qui répondent aux besoins de la base de données.

# Exemple de fichier PFILE

5

► # Initialisation de fichier de paramètres initdba01.ora

dbname =dba01

Instance\_name = dba01

control\_files = (home/dba01/ORADATA/u01/control01 dba01.ctl,  
home/dba01/ORADATA/u01/control01 dba01.ctl).

db\_block\_size = 4096

db\_cache\_size =4M

shared\_pool\_size =50000000

java\_pool\_size = 50000000

max\_dump\_file\_size = 10240

background\_dump\_dest = /home/dba01/ADMIN/DBUMP

user\_dump\_dest = /home/dba01/ADMIN/UDUMP

core\_dump\_dest = /home/dba01/ADMIN/CDUMP

undo\_management = AUTO

undo\_tablespace = UNDOTBS

# Fichier SPFILE spfileSID.ora

6

- Il s'agit d'un fichier binaire (n'est modifié manuellement)
- Sa mise à jour est effectuée par le serveur de base de données.
- Il réside toujours côté serveur
- Il permet de rendre les modifications persistantes après l'arrêt et le redémarrage.
- Il peut régler les valeurs des paramètres.

# Créer un fichier SPFILE

7

- Créer ce type fichier à partir d'un fichier PFILE
- `CREATE SPFILE='$ORACLE_HOME/dbs/spfileDB01.ora'`
- `FROM PFILE= '$ORACLE_HOME/dbs/initDBA01.ora';`

Où

- SPFILE-NAME correspond au fichier SPFILE à créer,
- PFILE -NAME correspond au fichier PFILE qui sert de base à la création du fichier SPFILE.
- Il peut être exécuté avant ou après le démarrage de l'instance
- `SQL> CREATE SPFILE FROM PFILE;`
- `SQL> CREATE PFILE FROM SPFILE`

# Fonctionnement de la commande STARTUP

8

- Ordre de priorités:
  - spfileSID.ora
  - SPFILE par défaut
  - initSID.ora
  - PFILE par défaut
- Vous pouvez modifier ces priorités si vous indiquez un PFILE:
- `STARTUP PFILE=$ORACLE_HOME/dbs/initDBA01.ora`
- Un PFILE peut indiquer qu'un SPFILE doit être utilisé
- `SPFILE=/database/startup/spfileDBA1.ora`
- [suite](#)

# Modifier des paramètres du fichier SPFILE

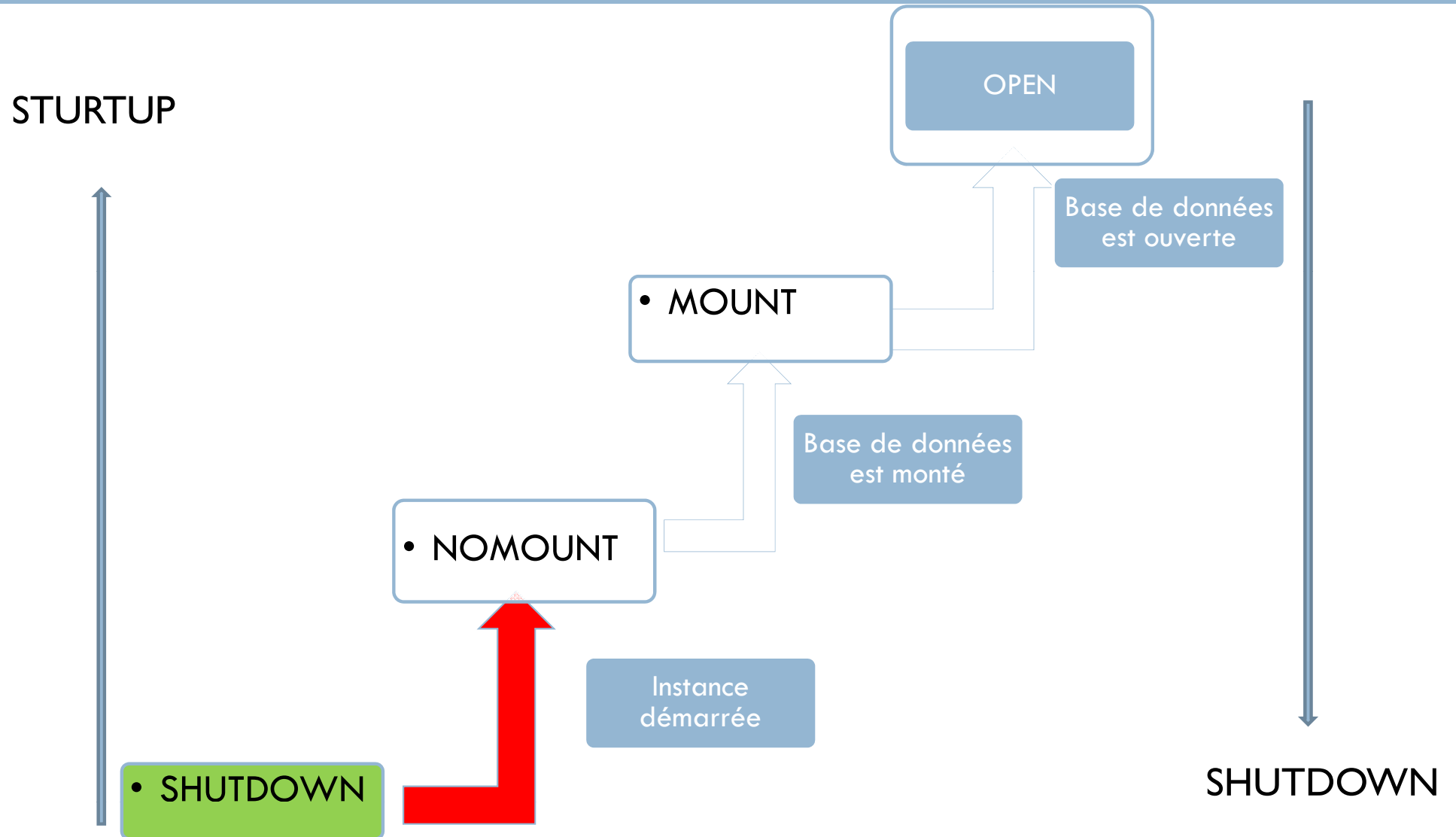
9

- ▶ Utiliser la commande `ALTER SYSTEM` pour apporter des modifications aux valeurs de paramètres:
  - `ALTER SYSTEM SET UNDO_TABLESPACE='UNDO2'`
- ▶ Indiquez si ces modifications sont temporaires ou persistantes:
  - `ALTER SYSTEM SET UNDO_TABLESPACE='UNDO2' SCOPE=BOTH;`
- ▶ Supprimez ou réinitialisez des valeurs:
  - `ALTER SYSTEM RESET UNDO_SUPPRESS_ERRORS SCOPE=BOTH SID='*';`



# Démarrer une base de données en mode NOMOUNT

10



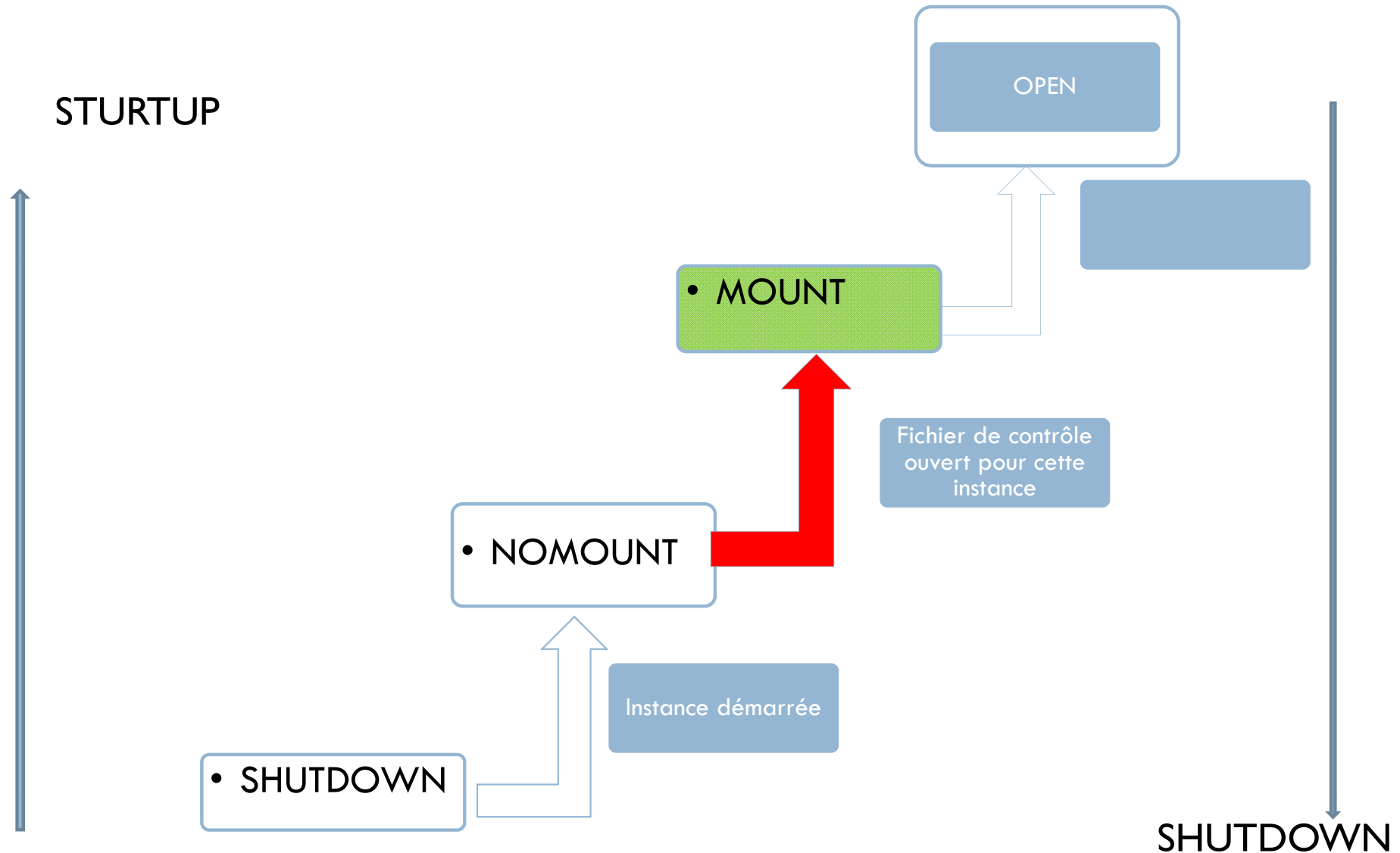
# Démarrez l'instance (NOMOUNT)

11

- Ce démarrage d'une instance en ce mode ne s'effectue qu'à la création d'une base de données ou à la création de fichiers de contrôle.
  - Le démarrage d'une instance comprend les tâches suivantes (voir page 8) :
  - la déclaration du paramètre PFILE avec STARTUP remplace les valeurs par défaut.
- L'affectation de la mémoire SGA
- Le démarrage des processus d'arrière-plan
- L'ouverture du fichier alertSID.log et des fichiers trace.

# Démarrer une base de données en mode MOUNT

12



# Exemple de tâches de maintenances en mode MOUNT

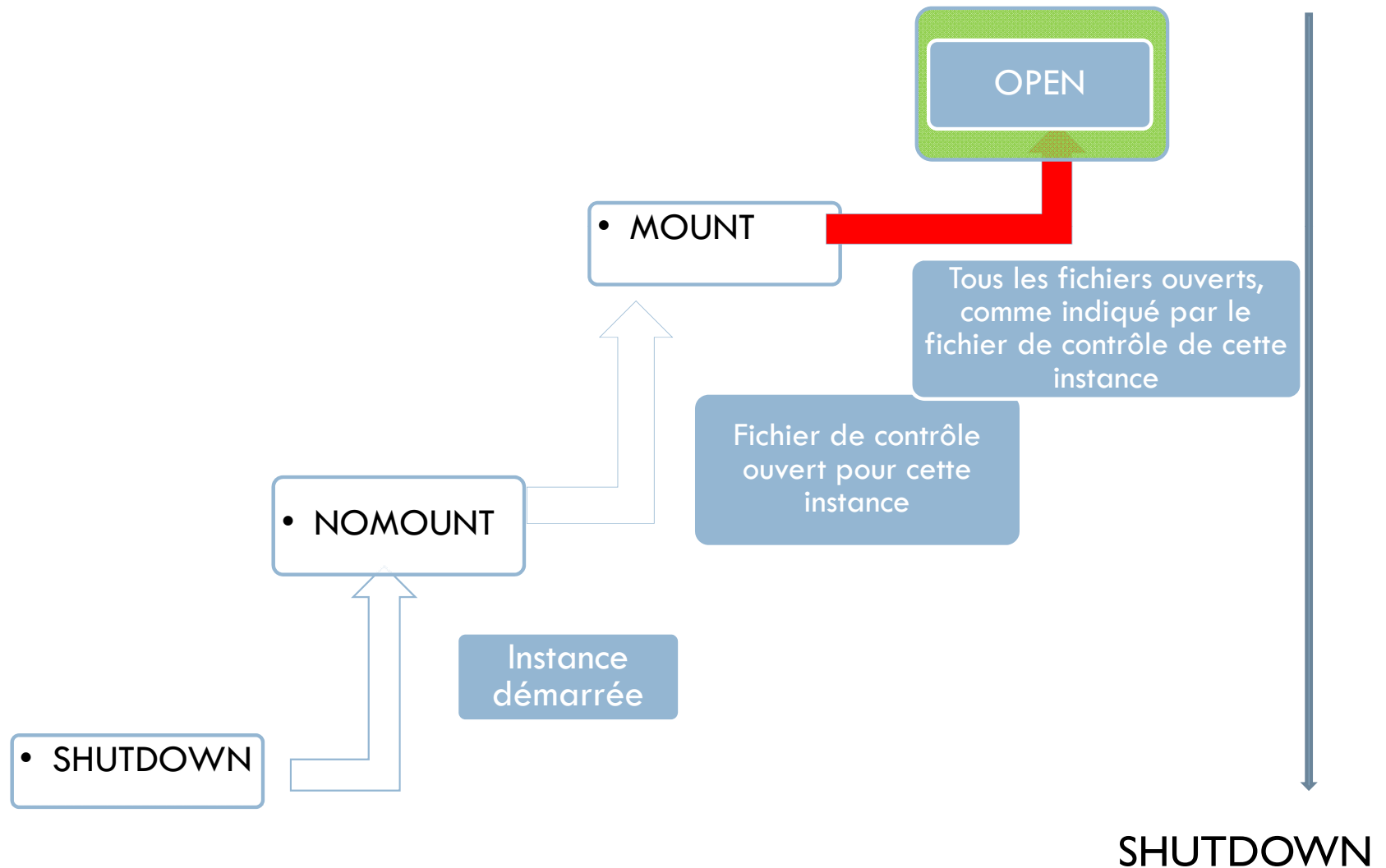
13

- ▶ Pour effectuer des opérations de maintenance, vous démarrez une instance et montez une base de données sans l'ouvrir. Exemple de tâches
  - ▶ Renommer des fichiers de données
  - ▶ Activer ou désactiver des options d'archivage de fichiers de journalisation
  - ▶ Effectuer une récupération complète de la base de données
- ▶ Le montage d'une base de données comprend les tâches suivantes:
  - ▶ Association d'une base de données à une instance démarrée
  - ▶ Localisation et ouverture des fichiers de contrôle indiqués dans les fichiers de paramètres
  - ▶ Lecture des fichiers de contrôle pour extraire le nom et le statut des fichiers de données et des fichiers journalisation. Toutefois, l'existence des fichiers de données et des fichiers de journalisation en ligne n'est pas vérifiée à ce stade.

# Démarrer une base de données en mode OPEN

14

STURTUP



# Démarrer une base de données en mode OPEN

15

- ▶ En mode de fonctionnement normal, le démarrage d'une instance se fait avant l'ouverture de la base. Ainsi, les utilisateurs peuvent se connecter à la base et effectuer des opérations standard sur les données.
- ▶ L'ouverture de la base de données comprend les tâches suivantes:
  - ▶ Ouverture des fichiers de données en ligne
  - ▶ Ouverture des fichiers de journalisation en ligne
- ▶ Au cours de cette dernière étape, le serveur vérifie que tous les fichiers de données et de journalisation en ligne peuvent être ouverts et contrôle la cohérence de la base de données. Si nécessaire, le processus d'arrière plan SMON( System Monitor) déclenche la récupération de l'instance.

# Commande STARTUP

16

- ▶ Pour démarrer une instance, exécuter la commande suivante:

- ▶ `STARTUP [FORCE] [RESTRICT] [PFILE=filename]  
[OPEN [RECOVER] [database] | MOUNT | NOMOUNT ]`

**OPEN:** Permet aux utilisateurs d'accéder à la base de données

**MOUNT:** Monte la base de données pour certaines tâches d'administration, ne permet pas aux utilisateurs d'y accéder

**NOMOUNT:** crée la mémoire SGA et lance les processus d'arrière-plan, mais ne permet pas d'accéder à la base de données.

**PFILE=paramfile:** permet de configurer l'instance à partir d'un fichier contenant des paramètres qui ne sont pas des paramètres par défaut.

**FORCE:** interrompt l'instance en cours, puis exécute un démarrage normal

**RESTRICT:** n'autorise l'accès à la base de données qu'aux utilisateurs disposant du privilège RESTRICTED SESSION

**RECOVER:** Lance la procédure de restauration physique au démarrage de la base de données

# Commande ALTER DATABASE

17

- ▶ Remplacez le statut NOMOUNT de la base de données par le statut MOUNT:
- ▶ ALTER DATABASE db01 MOUNT;
- ▶ Ouvrir la base de données en lecture seule:
- ▶ ALTER DATABASE db01 OPEN READ ONLY;
- ▶ Pour faire passer la base de données du statut NOMOUNT à MOUNT ou du statut MOUNT à OPEN.
- ▶ Vous pouvez ouvrir la base de données en mode lecture seule pour éviter que les transactions utilisateurs ne modifient pas des données.
- ▶ [READ WRITE | READ ONLY]
  - ▶ READ WRITE ouvre la base de données en lecture-écriture pour permettre aux utilisateurs de générer des fichiers de journalisation



# Ouvrir une base de données en mode d'accès restreint

18

- ▶ Utiliser la commande **STARTUP** pour restreindre l'accès à une base de données
- ▶ **STARTUP RESTRICT**
- ▶ Utiliser la commande **ALTER SYSTEM** pour placer une commande en mode d'accès restreint:
  - ▶ **ALTER SYSTEM ENABLE RESTRICTED SESSION;**
  - ▶ Ce type de session est utile par exemple lors de la mise à jour des structures ou l'import/export d'une base de données.
  - ▶ **ALTER SYSTEM DISABLE RESTRICTED SESSION;**
  - ▶ Pour mettre fin à une session:  
**ALTER SYSTEM KILL SESSION** 'int1,int2' (Colonne SID de la vue V\$SESSION);
- ▶ A l'exécution de cette commande, le processus d'arrière plan PMON effectue les tâches suivantes:
  - ▶ Annulation de la transaction en cours de l'utilisateur
  - ▶ Libération de tous les verrous de table ou de ligne
  - ▶ Libération de toutes les ressources réservées par l'utilisateur

# Ouvrir une base de données en mode lecture seule

19

- ▶ Ouvrir une base de données en mode lecture seule
- ▶ STARTUP MOUNT
- ▶ ALTER DATABASE OPEN READ ONLY;
- ▶ Une base de données en lecture seule permet:
  - ❑ D'exécuter des interrogations,
  - ❑ D'exécuter des tris sur disque à l'aide de tablespaces gérés localement,
  - ❑ De mettre des fichiers de données hors ligne et en ligne, mais pas des tablespaces,
  - ❑ De récupérer des fichiers de données et des tablespaces hors ligne.

# Arrêter la base de données

20

Mode d'arrêt	A	I	T	N
Permet de nouvelles connexions	Non	Non	Non	Non
Attend la fin des sessions en cours	Non	Non	Non	Oui
Attend la fin des transactions en cours	Non	Non	Oui	Oui
Applique un point de reprise et ferme les fichiers	Non	Oui	Oui	Oui

Mode d'arrêt

A = ABORT

I = IMMEDIATE

T = TRANSACTIONAL

N = NORMAL

Arrêter la BD pour effectuer la sauvegarde hors ligne de toutes les structures physiques via l'OS et pour que les modifications appliquées aux paramètres d'initialisation statiques soient effectives au redémarrage

Connecter vous en tant que SYSOPER ou SYSDBA:

SHUTDOWN [NORMAL | TRANSACTIONAL | IMMEDIATE | ABORT]

# Option d'arrêt

21

## Phase d'arrêt:

- ▣ Le cache tampons de la base de données est écrit dans les fichiers de données.
- ▣ Les modifications non valides sont annulées
- ▣ Les ressources sont libérées

Arrêt en mode  
Normal,  
Transactional  
ou Immediate

## Phase de redémarrage:

- Aucune récupération d'instance

Base de données cohérente (base « propre »)

# Effectuer un arrêt

22

## □ Mode Normal

C'est le mode par défaut. Il s'effectue dans les conditions suivantes:

- Aucune nouvelle connexion ne peut être établie.
- Le serveur attend la déconnexion préalable de tous les utilisateurs
- Les tampons de journalisation et de la base de données sont écrits sur disque
- Les processus d'arrière plan prennent fin et la zone SGA est supprimée de la mémoire.
- Le serveur ferme et démonte la base de données avant d'arrêter l'instance.
- La récupération de l'instance n'est pas nécessaire lors du redémarrage.

## □ Mode Transactional

- Ce mode évite aux utilisateurs de perdre leurs travaux en cours. Il s'effectue dans les conditions suivantes:
- Aucun client ne peut lancer de nouvelle transaction pour l'instance indiquée.
- Le client est déconnecté lorsqu'il termine la transaction en cours.
- La fin de toutes les transactions entraîne l'arrêt immédiat de la base de données.
- La récupération de l'instance n'est pas nécessaire lors du redémarrage.

# Option d'arrêt

23

## Phase d'arrêt:

- ▣ Les mémoires tampon modifiées ne sont pas écrites dans les fichiers de données.
- ▣ Les modifications non valides ne sont pas annulées

Arrêt en mode  
Abort, échec de  
l'instance ou  
démarrage forcé

## Phase de redémarrage:

- ▶ Les fichiers de journalisation permettent de réappliquer les modifications .
- ▶ Des segments d'annulation sont utilisés pour annuler les modifications non validées.
- ▶ Les ressources sont libérées.

Base de données incohérente (base non « propre »)

# Effectuer un arrêt

24

## □ Mode Immediate

Il s'effectue dans les conditions suivantes:

- ▣ Les instructions SQL en cours de traitement par le serveur ne sont pas terminées.
- ▣ Le serveur n'attend pas la déconnexion des utilisateurs de la base de données
- ▣ Le serveur annule les transactions actives et déconnecte tous les utilisateurs
- ▣ Le serveur ferme et démonte la base de données avant d'arrêter l'instance
- ▣ La récupération de l'instance n'est pas nécessaire lors du redémarrage.

## □ Mode Abort

Il s'effectue dans les conditions suivantes:

- ▣ Les instructions SQL en cours de traitement par le serveur ne sont pas terminées.
- ▣ Le serveur n'attend pas la déconnexion des utilisateurs de la base de données
- Les tampons de journalisation et de la base de données ne sont pas écrits sur disque
- Les transactions non valides ne sont pas annulées
- L'instance est interrompue sans fermeture des fichiers
- La base de données n'est pas fermée, ni démontée
- La récupération de l'instance est nécessaire lors du redémarrage; elle s'effectue automatiquement.

# Surveiller une instance à l'aide de fichiers de diagnostic

25

- Les fichiers de diagnostic:
  - Contiennent des informations relatives aux événements significatifs qui se sont produits.
  - Permettre de résoudre des incidents
  - Permettre d'améliorer la gestion quotidienne de la BD
- Ils peuvent être de plusieurs types:
  - Fichiers alertSDI.log
  - Fichiers trace de processus en arrière-plan
  - Fichier trace utilisateur



# Fichier d'alertes

26

- Le fichier alertSDI.log
- Enregistre les commandes
- Enregistre les résultats des principaux évènements
- Conserve quotidiennement des informations opérationnelles,
- Etablit le diagnostic des erreurs de la base de données
- Chaque entrée est associée à un horodatage
- Il doit être géré par l'administrateur de base de données
- Son emplacement est défini par le paramètre BACKGROUND\_DUMP\_DEST

# Fichiers trace de processus en arrière-plan

27

- Les fichiers trace de processus en arrière-plan:
- Consignent les erreurs détectées par les processus d'arrière-plan permettent de diagnostiquer et de corriger les erreurs
- Ils sont créés lorsqu'un processus d'arrière-plan détecte une erreur.
- Leur emplacement est défini par le paramètre `BACKGROUND_DUMP_DEST`
  - Exemple (bd01\_lgwr\_23845.trc)

# Fichier trace utilisateur

28

- Les fichiers trace utilisateur:
- Sont créés par le processus utilisateur
- Peuvent être générés par un processus serveur
- Contiennent des statistiques destinées aux instructions SQL tracées,
- Contiennent des messages d'erreur utilisateur
- Ce type de fichier est créé lorsqu'un utilisateur rencontre des erreurs au cours d'une session.
- Son emplacement est défini par le paramètre `USER_DUMP_FILE_SIZE`
- Sa taille est définie par le paramètre `MAX_DUMP_FILE_SIZE`
- Convention d'appellation `sid_oracle_PID.trc`( `db01_oracle_23845.trc`)

# Activer ou désactiver la fonction de trace utilisateur

29

## □ Niveau session :

### ▣ A l'aide de la commande

```
ALTER SESSION SET SQL_TRACE=TRUE
```

### ▣ A l'aide de la procédure SGBD:

```
Dbms_system.SET_SQL_TRACE_IN_SESSION
```

## □ Niveau instance

### ▣ A l'aide du paramètre d'initialisation

```
SQL_TRACE=TRUE;
```

# **Créer le fichier de contrôle et gérer les fichiers de journalisation**

# objectifs

2

A la fin de ce chapitre, vous pourrez:

- expliquer les fonctions du fichier de contrôle
- énumérer le contenu du fichier de contrôle
- multiplexer et gérer le fichier de contrôle
- Gérer le fichier de contrôle à l'aide d'Oracle-Managed Files (OMF)
- obtenir des informations sur les fichiers de contrôle

# Fichier de contrôle

3

- Il s'agit d'un petit fichier binaire nécessaire au démarrage.
- Il définit l'état actuel de la base de données physique.
- Il assure l'intégrité de la base de données et constamment mis à jour par le serveur.
- Il est requis:
  - lors de l'étape **MOUNT** lors du démarrage de la base de données
  - pour le fonctionnement de la base de données
- Il est lié à une seule base de données
- La perte de ce fichier peut nécessiter la récupération de la base de données

# Contenu de fichier de contrôle

4

Le fichier de contrôle contient les entrées suivantes:

- Nom et identificateur de la base de données
- Horodatage de création de la base de données
- Nom des tablespaces
- Nom et emplacement des fichiers de données et des fichiers de journalisation (redo log)
- Numéro de séquence du fichier de journalisation en cours
- informations sur les points de reprise (checkpoints)
- Début et fin des segments d'annulation (undo segments)
- Informations (emplacement et statut) sur l'archivage des fichiers de journalisation
- informations sur les sauvegardes

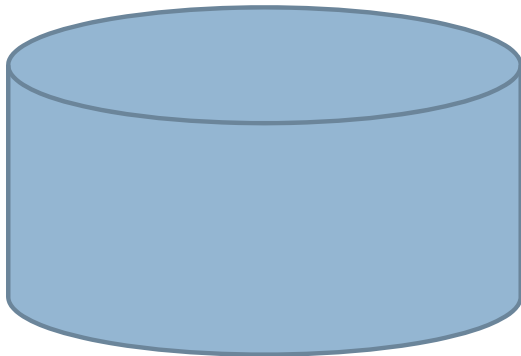


# Multiplexer le fichier de contrôle

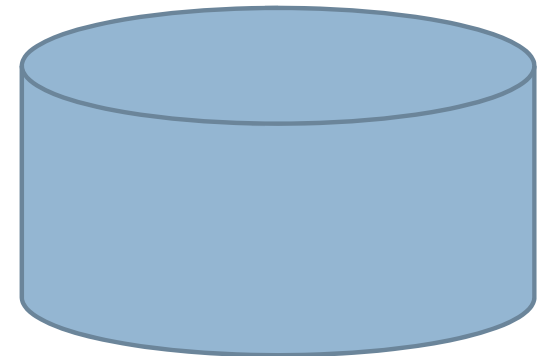
5

- Le fichier de contrôle enregistre la structure physique de la base de données.

Disque 1



Disque 2



# Multiplexer le fichier de contrôle lorsqu'un fichier SPFILE est utilisé

6

## □ Modifier le SPFILE

```
ALTER SYSTEM SET control_files=  
'$HOME/ORADATA/u01/ctrl01.ctl',  
'$HOME/ORADATA/u02/ctrl02.ctl' SCOPE=SPFILE;
```

### • Arrêter la base de données

```
SHUTDOWN IMMEDIATE;
```

### • Créez des fichiers de contrôle supplémentaires:

```
cp  
$HOME/ORADATA/u01/ctrl01.ctl  
$HOME/ORADATA/u02/ctrl02.ctl
```

### • Démarrez la base de données

```
STARTUP;
```

# Multiplexer le fichier de contrôle lorsqu'un fichier PFILE est utilisé

7

- Arrêter la base de données

```
SHUTDOWN IMMEDIATE;
```

- Créez des fichiers de contrôle supplémentaires:

```
cp  
$HOME/ORADATA/u01/ctrl01.ctl  
$HOME/ORADATA/u02/ctrl02.ctl
```

- Ajouter les noms de fichiers de contrôle au fichier PFILE:

```
control_files=  
(DISK1/ctrl01.ctl, DISK1/ctrl02.ctl )
```

- Démarrez la base de données

```
STARTUP;
```

# Obtenir des informations sur les fichiers de contrôle

8

- Vous pouvez interroger les vues suivantes pour obtenir des informations sur le statut et l'emplacement des fichiers de contrôle:
- V\$CONTROLFILE répertorie le nom et le statut de tous les fichiers de contrôle associés à l'instance  
(select name from V\$CONTROLFILE).
- V\$PARAMETER répertorie le statut et le l'emplacement de tous les paramètres.  
(select name,value from V\$PARAMETER where name='control\_files').
- V\$CONTROLFILE\_RECORD\_SECTION fournit des informations sur les enregistrements des différentes sections des fichiers de contrôle.  
(select type,record\_size, record\_total, record\_used from V\$CONTROLFILE\_RECORD\_SECTION where type='datafile')
- SHOW PARAMETERS CONTROL\_FILE répertorie le nom, le statut et l'emplacement des fichiers de contrôle

# Gérer les fichiers de journalisation

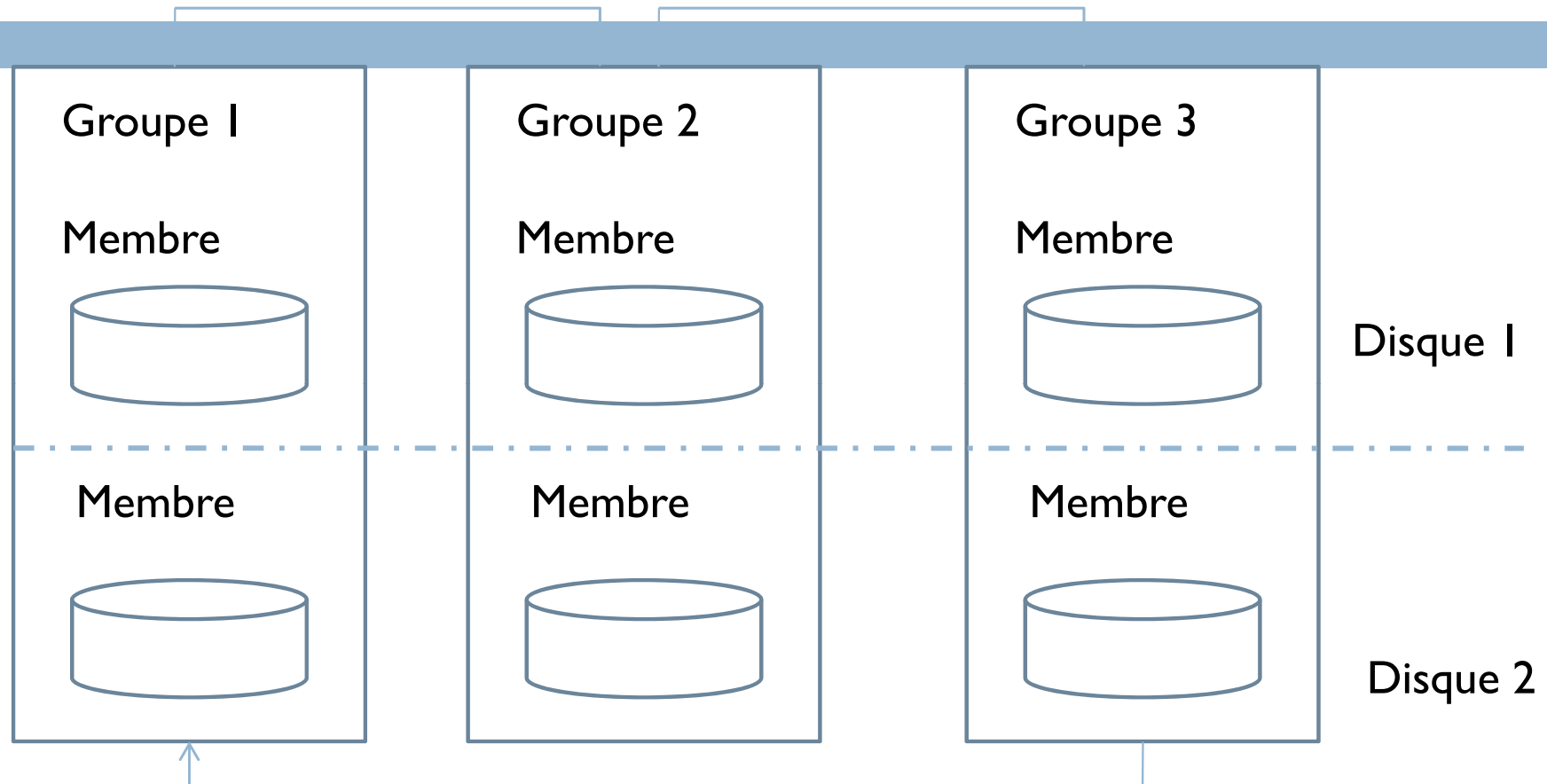
9

Les fichiers de journalisation (redo log) permettent de journaliser les transactions en cas de panne de BD et présentent les caractéristiques suivantes:

- Il enregistrent toutes les modifications apportées au données.
- Ils offrent un mécanisme de récupération(ex. transactions qui n'ont pas encore validées)
- Ils peuvent être organisés en groupes
- Deux groupes au moins sont requis

# Structures des fichiers de journalisation

10



L'administrateur de DB peut configurer la base pour gérer des copies de fichiers de journalisation en ligne afin d'éviter de perdre des données en cas d'incident.

Le nombre de groupe est limité par le paramètre MAXLOGFILES

Le nombre de membre est limité par le paramètre MAXLOGMEMBERS

# Mode de fonctionnement de fichiers de journalisation

11

- Le serveur enregistre de manière séquentielle toutes les modifications apportées à la BD dans le tampon de journalisation.
- Les entrées de journalisations sont écrites par le processus LGWR dans l'un des fichiers de journalisation appelé groupe de fichiers de journalisation en cours dans les cas suivants:
  - Lorsqu'une transaction est validée,
  - Lorsqu'un tiers de tampon de journalisation est occupé
  - Lorsque le tampon de journalisation contient plus d'un mégaoctet d'enregistrement modifiés.
  - Avant que le processus DBWn n'écrive les blocs modifiés du cache de tampons (buffer cache) de la base de données dans les fichiers de données.
- Les fichiers de journalisations sont écrite d'une façon cyclique. Chaque groupe est identifié par un numéro de séquence.
- On parle alors de changement de fichier de journalisation

# Mode de fonctionnement de fichiers de journalisation

12

- Une opération de point de reprise se produit.
  - Plusieurs tampons « dirty » de la BD gérés par le fichier de journalisation faisant l'objet d'un point de reprise( checkpoint) sont écrits par le DBWRn dans les fichiers de données. Leur nombre est déterminé par le paramètre FAST\_START\_MTTR\_TARGET
  - Le processus CKPT met à jour le fichier de contrôle. Si le point de reprise est lancé par un changement de fichier de journalisation, ce processus met également à jour les en-têtes des fichiers de données.
  - Un point de reprise peut se produire également à chaque changement de fichier de journalisation
  - Lors de l'arrêt d'une instance (Normal, Transactional, Immediate).
  - Exécuté par l'administrateur



# Imposer des changements de fichiers de journalisation et des points de reprise

13

- Imposer un changement de fichier de journalisation

```
ALTER SYSTEM SWITCH LOGFILE;
```

- Vous pouvez forcer l'application de points de reprise à l'aide :  
Du paramètre FAST\_MTTR\_TARGET,

```
FAST_MTTR_TARGET=600; //600 secondes
```

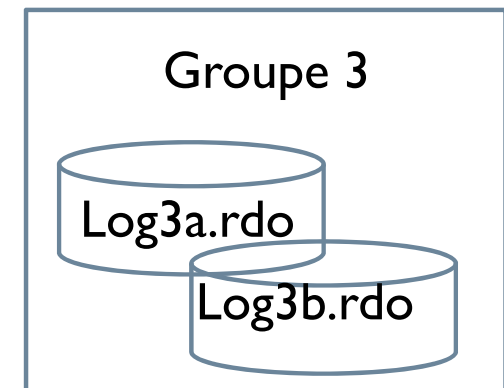
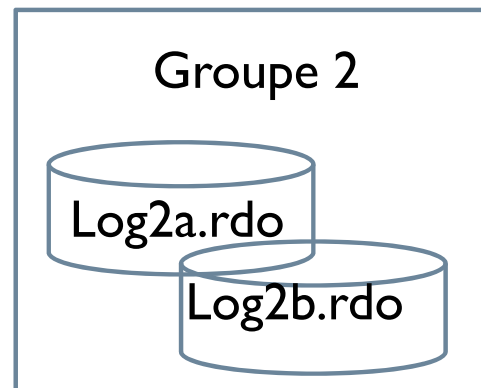
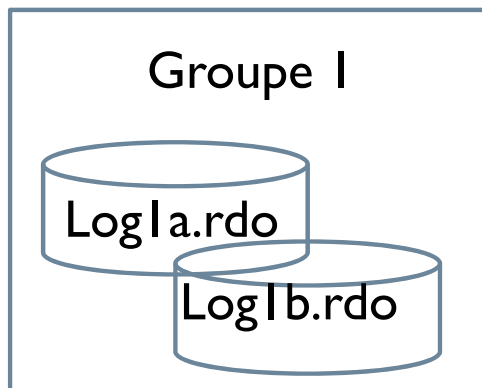
- De la commande ALTER SYSTEM CHECKPOINT

```
ALTER SYSTEM CHECKPOINT;
```

# Ajouter des groupes de fichiers de journalisation en ligne

14

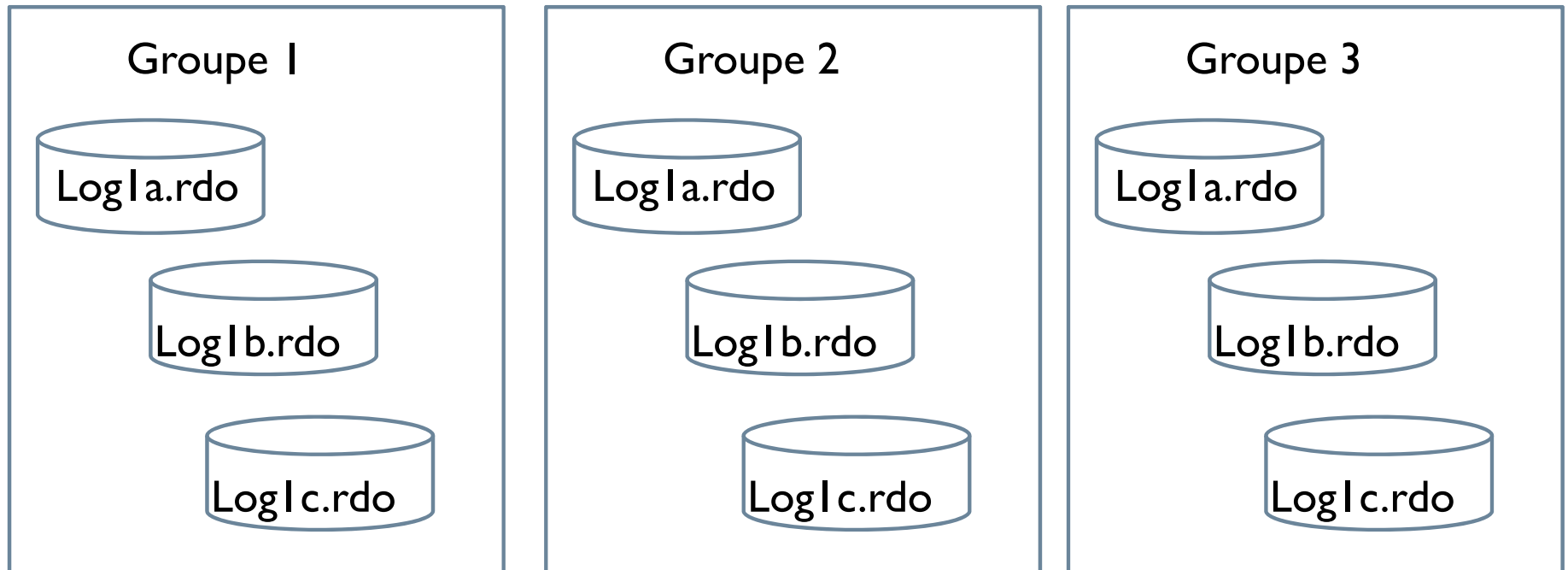
```
ALTER DATABASE ADD LOGFILE GROUP 3  
('$HOME/ORADATA/U01/log3a.rdo',  
 '$HOME/ORADATA/U02/log3b.rdo')  
SIZE 1M;
```



# Ajouter des membres à des fichiers de journalisation en ligne

15

```
ALTER DATABASE ADD LOGFILE MEMBER  
'$HOME/ORADATA/U04/log1c.rdo' TO GROUP 1,  
'$HOME/ORADATA/U04/log2c.rdo' TO GROUP 2,  
'$HOME/ORADATA/U04/log3c.rdo' TO GROUP 3;
```



# Supprimer des groupes de fichiers de journalisation en ligne

16

- Pour augmenter ou réduire la taille des groupes de fichiers de journalisation en ligne, ajouter-en de nouveaux (ayant la nouvelle taille) puis supprimez les anciens

**ALTER DATABASE DROP LOGFILE GROUP 3;**

- Restrictions:
  - Une instance nécessite aux moins deux groupes de fichiers de journalisation en ligne.
  - Vous ne pouvez pas supprimer un groupe actif ou en cours.
  - Lorsque vous supprimez un groupe de fichiers de journalisation en ligne, les fichiers du système d'exploitation sont conservés.

# Supprimer des membres de fichiers de journalisation en ligne

17

- Vous pouvez supprimer un membre parce qu'il n'est plus valide.
- `ALTER DATABASE DROP LOGFILE MEMBER  
'$HOME/ORADATA/U04/log3c.rdo';`
- Restrictions:
  - Vous devez imposer un changement de fichier de journalisation pour supprimer un membre du groupe en cours
  - La base fonctionne en mode ARCHIVLOG et que le groupe de fichiers de journalisation du membre n'est pas archivé, vous ne pouvez pas supprimer le membre.
  - Lorsque vous supprimez un membre d'un groupe de fichiers de journalisation en ligne, les fichiers du système d'exploitation sont conservés.

# Obtenir des informations sur les groupes et les membres

18

- Pour obtenir des informations sur les groupes et les membres, interrogez les vues suivantes:
- V\$LOG
- V\$LOGFILE

```
SELECT group#, sequence#, byte members, status from v$log;
```

La colonne STATUS affiche des éléments qui correspondent à:

- UNUSED (fichiers viennent d'être ajoutés)
- CURRENT (ce groupe est actif)
- ACTIVE (groupe actif mais il n'est pas du groupe en cours)
- CLEARING (journal recréé sous la forme d'un fichier vide)
- CLEARING\_CURRENT (erreur de changement de fichier)
- INACTIVE (n'est plus utile à la récupération d'instance)

# Fichiers de journalisation archivés

19

- L'une des principales décisions d'un administrateur de BD consiste à déterminer si la base doit être configurée pour fonctionner en mode ARCHIVLOG
- L'exécution de la DB en mode ARCHIVELOG et l'archivage des fichiers de journalisation présentent deux avantages:
  - Récupération: le sauvegarde de la BD et des fichiers de journalisation en ligne et archives peuvent garantir la récupération de toutes les transactions validées.
  - Sauvegarde: Peut s'effectuer lorsque la BD est ouverte
- Par défaut une base de données est créée en mode NOARCHIVELOG

# Utiliser les vues du dictionnaire de données et les vues dynamiques des performances



# objectifs



A la fin de ce chapitre, vous pourrez :

- identifier les objets de base de données intégrés
- identifier le contenu et les types d'utilisation du dictionnaire de données.
- comprendre comment sont créées les vues du dictionnaire de données
- interroger le dictionnaire de données et les vues dynamiques des performances
- comprendre les conventions d'appellation des scripts d'administration.

# Objectifs de base de données intégrés

Autres objectifs créés avec la base de données:

- dictionnaire de données
  - Décrit les objets de BD
- Tables des performances
  - Contiennent des informations permettant au DBA de surveiller et de régler la BD et l'instance correspondante
- Packages PL/SQL
  - Sont des unités de programme ajoutant une fonctionnalité à la BD
- Déclencheurs d'événements de base de données.

# Dictionnaire de données

- Il est vital pour toute base de données Oracle.
- Il décrit la base de données et ses objets.
- Il contient des tables et des vues en lecture seule.
- Il est stocké dans le tablespace SYSTEM.
- Le propriétaire est l'utilisateur SYS.
- Sa mise à jour est effectuée par le serveur (commande LDD).
- Il est accessible avec le privilège SELECT.

# Tables de base et vues du dictionnaire de données

Le dictionnaire de données est composé de deux parties:

- table de base
- Contiennent la description de la base de donnée (Seul le serveur oracle peut écrire dans ces tables)
- Sont créées à l'aide de la commande CREATE DATABASE et contient des informations sur les index de la BD.
- Vues du dictionnaire de données.
- Simplifient les informations contenues dans les tables de base (synthétisent les informations des tables de base et les affiches dans un format qui facilite leurs exploitation, les vues utilisent des noms alors que les tables utilisent des numéros).
- sont créées à l'aide du script catalog.sql qui est

# Créer des vues du dictionnaire de données

Script	Tâche effectuée
Catalog.sql	Crée des vues du dictionnaire de données couramment utilisées et des synonymes
Catproc.sql	Exécute des scripts requis par PL/SQL coté serveur

Exécuter manuellement ces scripts lorsque vous créez manuellement une nouvelle base de données  
%ORACLE\_HOME%\rdbms\admin

# Contenu du dictionnaire de données

Le dictionnaire de données fournit des informations sur:

- La structure logique et la structure physique de la base de données.
- La définition d'objets de schéma de la BD et l'espace alloué aux objets. (Tables, vues, index, clusters, synonymes, séquences, procédures, fonctions, packages, déclencheurs, etc)
- Les contraintes d'intégrité.
- Les utilisateurs.
- Les rôles
- Les privilèges.
- La fonction d'audit (ex. les utilisateurs qui ont consulté ou mis à jour des objets de schéma).

# Utilisation du dictionnaire de données

Principaux types d'utilisation:

(le serveur de BD lit le dictionnaire de données pour s'assurer que les objets de schéma existent et que les utilisateurs disposent des privilèges d'accès appropriés sur ces derniers).

- Le serveur Oracle l'utilise pour rechercher des informations sur:
  - Les utilisateurs
  - Les objets de schéma
  - les structures de stockage

Le serveur Oracle le modifie lorsqu'une instruction LDD est exécutée

- Les utilisateurs et les administrateurs de base de données l'utilisent comme référence en lecture seule

# Catégories de vue du dictionnaire de données

- Trois principaux ensembles de vues statiques
- Se distinguent par leur portée:
  - DBA: contenu de tous les schémas

Tout utilisateur doté du privilège système **SELECT ANY TABLE** peut interroger les vues DBA du dictionnaire de données

**ALL: éléments auxquels l'utilisateur a accès**  
DBA\_XXX: tous les objets de la base de données

- **USER: contenu du schéma de l'utilisateur**  
ALL\_XXX: objets accessibles à l'utilisateur

USER\_XXX: objets appartenant à l'utilisateur en cours



# Catégories de vue du dictionnaire de données

L'administrateur peut exécuter une interrogation sur tous les objets de la base de données.

```
SELECT owner, object_name, object_type  
FROM dba_objects;
```

L'interrogation suivante renvoie des informations sur tous les objets auxquels l'utilisateur a accès:

```
SELECT owner, object_name, object_type  
FROM all_objects;
```

L'interrogation suivante renvoie des informations sur tous les objets dans le schéma utilisateur:

```
SELECT owner, object_name, object_type  
FROM user_objects;
```

# Catégories de vue du dictionnaire de données

- Les vues du dictionnaire de données sont des vues statiques qui répondent aux questions du type:
- L'objet-a-t-il été créé?
- D'où provient-il?
- Qui est son propriétaire?
- De quel privilège les utilisateurs disposent-ils?
- Quelles sont les restrictions relatives à l'objet?

# Exemples de vues du dictionnaire de données

- **Présentation générale**

- DICTIONNARY, DIC\_COLUMNS

- **Objets de schéma**

- DBA\_TABLES, DBA\_INDEXES,  
DBA\_TAB\_COLUMNS, DBA\_CONSTRAINTS

- **Location d'espace**

- DBA\_SEGMENTS, DBA\_EXTENTS

- **Structure de base de données**

- DBA\_TABLESPACES, DBA\_DATA\_FILES

# Exemples de vues du dictionnaire de données

Présentation de toutes les vues du DD

```
SELECT * FROM DICT
```

```
WHERE table_name LIKE 'dba_seg%'
```

Utilisez le mot clé DESCRIBE pour obtenir la liste des colonnes d'une vue:

```
DESCRIBE dba_users;
```

```
SELECT * FROM dba_users;
```

# Tables dynamiques des performances

- Le serveur de BD enregistrent l'activité en cours de la base de données dans des tables virtuelles
- Sont constamment mises à jour lorsque la base de données est active (ces tables ne sont présent en mémoire que lorsque la base est active)
- Pointent sur des informations réelle figurant dans la mémoire et dans le fichier de contrôle
- Permettent de surveiller et de régler la base de données
- Le propriétaire est l'utilisateur SYS
- Les synonymes commencent par le préfixe V\$\_

Sont répertoriées dans la vue V\$FIXED\_TABLE

(contient des information sur toutes les tables et vues

# Tables dynamiques des performances

- Les tables dynamiques de performances permettent de répondre à des questions de type:
  - L'objet est-il en ligne et disponible?
  - L'objet est-il ouvert?
  - Quels verrous externes (locks) sont appliqués?
  - La session est-elle active?

# Exemples de tables dynamiques des performances

- **V\$CONTROLFILE** (répertorie les noms des fichiers de contrôle);
- **V\$DATABASE** (contient les informations du fichier de contrôle sur la base de données)
- **V\$DATAFILE** (contient les informations du fichier de contrôle sur le fichier de données)
- **V\$INSTANCE** (affiche l'état de l'instance en cours)
- **V\$PARAMETER** (valeurs des paramètres utilisées dans la session)
- **V\$SESSION** (information sur la session en cours)
- **V\$SGA** : information de synthèse sur la mémoire SGA
- **V\$SPPARAMETER** : contenue de SPFILE
- **V\$TABLESPACE**:
- **V\$THREAD**: information du fichier de contrôle sur le THREAD

# Tables dynamiques des performances

Pour afficher des informations sur la date de création de la base.

```
SELECT STATUS,INSTANCE,ENABLE_TIME  
FROM V$THREAD;
```

Liste et commentaire de tous les colonnes des vues dynamique de performances.

```
SELECT *FROM DICT_COLUMNS;
```

Liste des SESSIONS ouvertes:

```
SELECT USERNAME,SERVER, MACHINE FROM  
V$SESSION;
```



# Convention d'appellation des scripts d'administration

Convention	description
Cat*.sql	Informations du catalogue et du dictionnaire de données
dbms*.sql	Spécifications de package de la base de données
pryt*.plb	Code de package de base de données crypté
utl*.sql	Vues et tables des utilitaires de base de données

# Synthèse

- Ce chapitre vous a permis d'apprendre à:
- identifier les objets de base de données intégrés
- identifier le contenu et les types d'utilisation du dictionnaire de données
- comprendre comment sont créées les vues du dictionnaires de données
- identifier les catégories des vues du dictionnaire de données
- interroger le dictionnaire de données et les vues dynamiques des performances
- comprendre les conventions d'appellations des scripts d'administration

# GÉRER LES DONNÉES D'ANNULATION

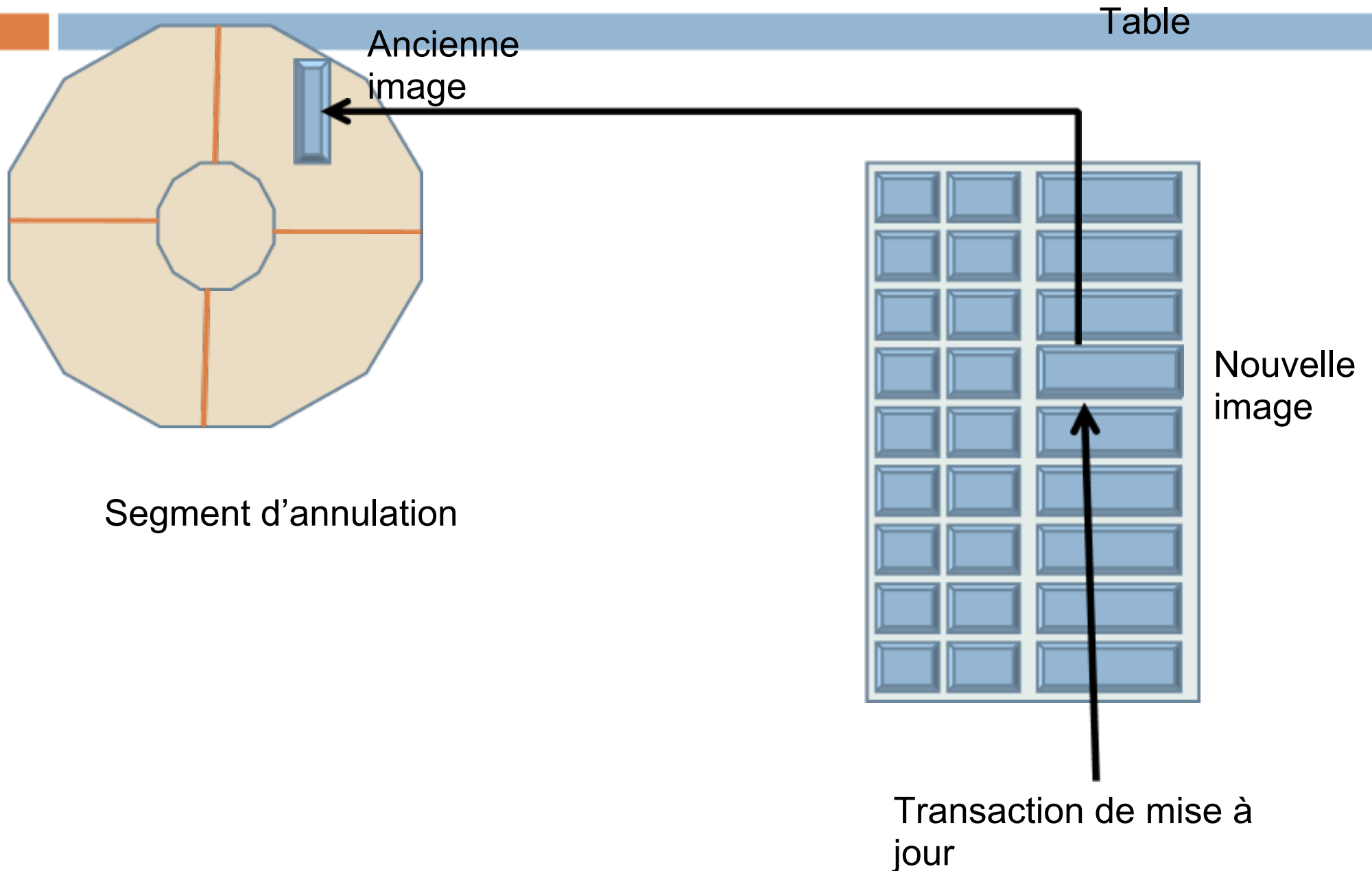


# Segment d'annulation

Un segment d'annulation permet d'enregistrer l'ancienne valeur (données d'annulation) lorsqu'un processus modifie les données d'une base. Il enregistre l'emplacement des données et leur valeur avant modification.

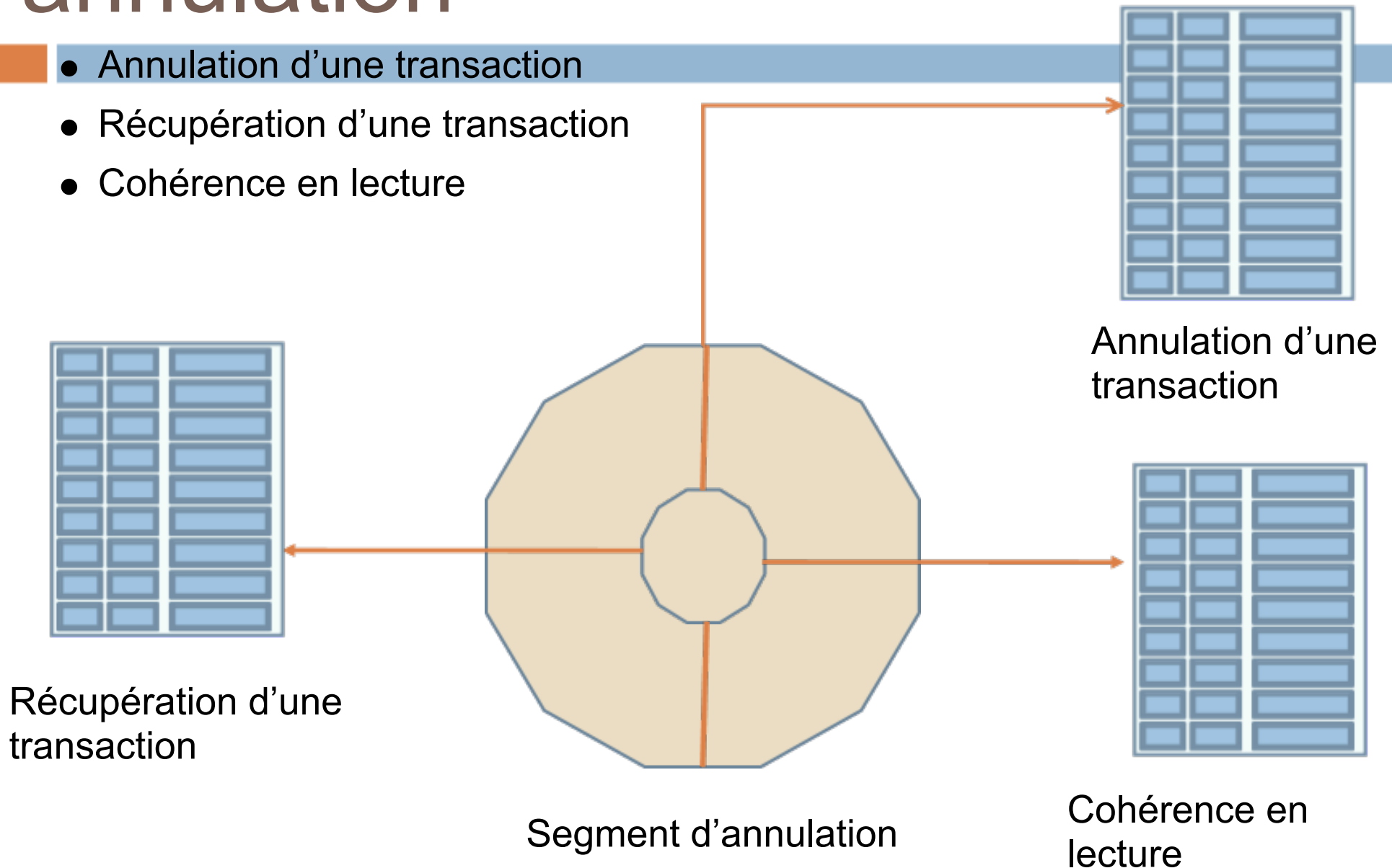
Dans l'entête d'un segment d'annulation contient une table des transactions dans laquelle sont stockées des informations sur les transactions en cours associées au segment d'annulation.

# Segment d'annulation



# Fonctions des segments d'annulation

- Annulation d'une transaction
- Récupération d'une transaction
- Cohérence en lecture



# Récupération d'une transaction

- Si l'instance échue tandis que des transactions sont en cours, le serveur doit annuler les modifications non validées lors de la réouverture de la base de données. L'annulation fait partie de la récupération de la transaction. La récupération est possible uniquement dans la mesure où les modifications effectuées dans le segment d'annulation sont également protégées par les fichiers de journalisations (fichiers redo log).

# Annulation d'une transaction

- Lorsqu'une transaction modifie une ligne dans une table, l'ancienne image des colonnes modifiées (données d'annulation) et enregistrée dans le segment d'annulation. En cas d'annulation (ROLLBACK) de la transaction, le serveur restaure les valeurs d'origine en les réécrivant dans la ligne du segment d'annulation.



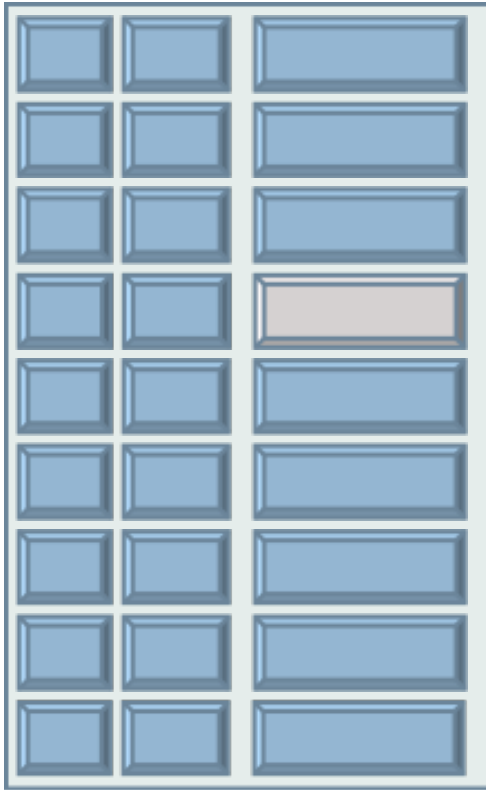
# Cohérence en lecture

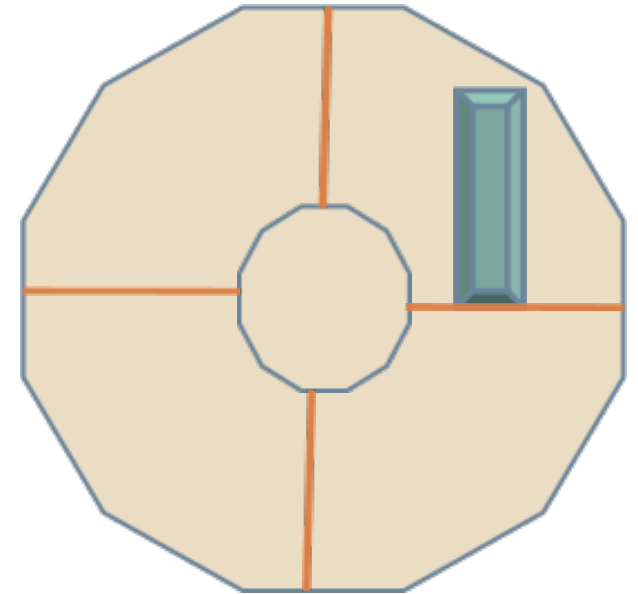
- Lorsque des transactions sont en cours, les autres utilisateurs de la base de données ne doivent pas avoir connaissance des modifications non validées effectuées par ces transactions. En outre une instruction ne doit pas avoir connaissance des modifications validées après le début de son exécution. Les anciennes valeurs (données d'annulation) des segments d'annulation sont également utilisées pour fournir aux lecteurs une image cohérente d'une instruction donnée.

# Cohérence en lecture

SELECT \* FROM Table

Table



Nouvelle image

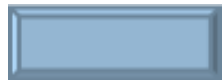


Image au démarrage de l'  
instruction

# Types de segment d'annulation

- Le segment SYSTEM est utilisé pour les objets du tablespace SYSTEM

- Créé dans le tablespace SYSTEM à la création d'une base de données. Il ne peut être utilisé que pour les modifications apportées aux objets du tablespace SYSTEM

Le segment d'annulation non SYSTEM utilisé pour les autres tables spaces

- Mode automatique: nécessite un tablespace d'annulation (UNDO)
- Mode manuel: acquis par une instance spécifique
- Public: acquis par une instance quelconque (constituent un pool de segments d'annulation disponibles dans une base de données).

# Concepts de la gestion automatique des annulations

- Les données d'annulations sont gérées à l'aide d'un tablespace d'annulation (UNDO)
- Allouez un tablespace d'annulation (UNDO) par instance en prévoyant suffisamment d'espace pour la charge globale de l'instance.
- Le serveur Oracle gère automatiquement les données d'annulation dans le tablespace d'annulation.

# Configuration de la gestion automatique des annulations

Configurer les deux paramètres suivants dans le fichier d'initialisation

- UNDO\_MANAGEMENT=AUTO/MANUEL
- UNDO\_TABLESPACE=UNDOTBS

Créer au moins un tablespace d'annulation (UNDO).



# Gestion automatique des annulations: tablespace d'annulation (UNDO)

- Pour créer le tablespace d'annulation (UNDO) en même temps que la base de données. Ajoutez une clause dans la commande CREATE DATABASE.

```
CREATE DATABASE db01  
  
...  
UNDO TABLESPACE undo1  
DATAFILE '/u01/oradata/undo1db01.dbf'  
SIZE 20M;
```

Vous pouvez également créer le tablespace ultérieurement à l'aide de la commande CREATE UNDO TABLESPACE

```
CREATE UNDO TABLESPACE undo1  
DATAFILE '/u01/oradata/undo1db01.dbf'  
SIZE 20M;
```

# Gestion automatique des annulations: modifier un tablespace d'annulation (UNDO)

- La commande ALTER TABLESPACE permet de modifier des tablespaces d'annulation (UNDO).
- Dans l'exemple ci-dessous, un fichier de données est ajouté au tablespace d'annulation (UNDO):

```
ALTER TABLESPACE undo1  
ADD DATAFILE '/u01/oradata/undo1tbs2.dbf'  
SIZE 30M  
AUTOEXTEND ON;
```

# Gestion automatique des annulations: changer de tablespace d'annulation (UNDO)

- Vous pouvez passer d'un tablespace d'annulation (UNDO) à un autre.
- Vous ne pouvez affecter à une base de données qu'un tablespace d'annulation (UNDO) à la fois.
- Une instance peut contenir plusieurs tablespaces d'annulation (UNDO). Mais un seul peut être actif.
- La commande ALTER SYSTEM permet de changer dynamiquement de tablespace d'annulation.

```
ALTER SYSTEM SET  
UNDO_TABLESPACE=UNDOTBS2;
```



# Gestion automatique des annulations: supprimer un tablespace d'annulation (UNDO)

- La commande DROP TABLESPACE permet de supprimer un tablespace d'annulation.

```
DROP TABLESPACE UNDOTBS2;
```

- Vous ne pouvez pas supprimer un tablespace d'annulation (UNDO) s'il est en cours d'utilisation par une instance.
- Pour supprimer un tablespace d'annulation actif:
- Changer de tablespace d'annulation.
- Supprimer le tablespace une fois que toutes les transactions en cours sont terminées.

# Statistiques relatives aux données d'annulation

```
SELECT end_time, begin_time, undoblks  
FROM V$UNDOSTAT;
```

END_TIME	BEGIN_TIME	UNDOBLKS
14-NOV-08	14-NOV-08	0
14-NOV-08	14-NOV-08	2
14-NOV-08	14-NOV-08	3
14-NOV-08	14-NOV-08	2
14-NOV-08	14-NOV-08	32
14-NOV-08	14-NOV-08	1
14-NOV-08	14-NOV-08	5
14-NOV-08	14-NOV-08	2
14-NOV-08	14-NOV-08	2
14-NOV-08	14-NOV-08	2
14-NOV-08	14-NOV-08	43
14-NOV-08	14-NOV-08	4
14-NOV-08	14-NOV-08	3
14-NOV-08	14-NOV-08	0
14-NOV-08	14-NOV-08	3
14-NOV-08	14-NOV-08	4
14-NOV-08	14-NOV-08	10

Cette vue affiche un histogramme des données statistiques relatives au bon fonctionnement de la base de données. Chaque ligne contient les statistiques collectées toutes les 10 minutes dans l'instance. Cette vue vous permet d'estimer l'espace d'annulation nécessaire pour la charge globale en cours .

# Gestion automatique d'annulations: dimensionner un tablespace d'annulation

- Pour déterminer la taille d'un tablespace d'annulation (UNDO) vous devez disposer des éléments d'informations suivants:
- UNDO\_RETENTION en secondes (UR)
- Nombre de blocs de données d'annulation générés par seconde (UPS)
- L'espace supplémentaire varie en fonction des extents et des fichiers (db\_block\_size) (DBS)

$$\text{UndoSpace} = [\text{UR} * (\text{UPS} * \text{DBS})] + (\text{DBS} * 24)$$

# Gestion automatique d'annulations: dimensionner un tablespace d'annulation

```
SELECT (UR* (UPS * DBS))+(DBS * 24 ) AS "Bytes"
FROM ( SELECT value AS UR
FROM v$parameter
WHERE name='undo_retention'),
(SELECT (SUM(undoblks)/SUM( ( ( end_time-begin_time)
*86400) ) ) AS UPS
FROM v$undostat),
(SELECT value AS DBS
FROM v$parameter
WHERE name='db_block_size');
```

Pour convertir des octets en mégaoctets,  
divisez-les par 1 048 576.

# Gestion automatique d'annulations: dimensionner un tablespace d'annulation

```
SELECT ROUND((Bytes/1048576),2) AS MEGAOCTETS
FROM (

SELECT (UR* (UPS * DBS))+(DBS * 24 ) AS Bytes
FROM ( SELECT value AS UR
FROM v$parameter
WHERE name='undo_retention'),
(SELECT (SUM(undoblks)/SUM( ( ( end_time-begin_time)*86400) ) ) AS UPS
FROM v$undostat),
(SELECT value AS DBS
FROM v$parameter
WHERE name='db_block_size')
```

```
);
```

# Gestion automatique d'annulations: quota d'annulation

- Les transactions longues ou incorrectes peuvent utiliser des ressources considérables.
- Les quotas d'annulation permettent de regrouper les utilisateurs et de leur affecter un espace d'annulation maximum défini.
- La directive UNDO\_POOL de Resource Manager définit l'espace alloué à un groupe de ressources.
- Lorsqu'un groupe dépasse la limite définie, il ne peut plus exécuter de nouvelle transaction jusqu'à ce que les transactions en cours se terminent ou échouent, libérant ainsi de l'espace d'annulation.

# Obtenir des informations sur les segments d'annulation

- Interrogez les vues suivantes pour obtenir des informations sur les segments d'annulation:
- - BDA\_ROLLBACK\_SEGS
- Vues dynamiques de performances
- V\$ROLLNAME
- V\$ROLLSTAT
- V\$UNDOSTAT
- V\$SESSION
- V\$TRANSACTION

# Obtenir des informations sur les segments d'annulation

```
SELECT segment_name, owner, tablespace_name,  
status  
FROM dba_rollback_segs;
```

SEGMENT NAME	OWNER	TABLESPACE_NAME	STATUS
SYSTEM	SYS	SYSTEM	ONLINE
_SYSSMU1\$	PUBLIC	UNDOTBS1	ONLINE
_SYSSMU2\$	PUBLIC	UNDOTBS1	ONLINE
_SYSSMU3\$	PUBLIC	UNDOTBS1	ONLINE
_SYSSMU4\$	PUBLIC	UNDOTBS1	ONLINE
_SYSSMU5\$	PUBLIC	UNDOTBS1	ONLINE
_SYSSMU6\$	PUBLIC	UNDOTBS1	ONLINE
_SYSSMU7\$	PUBLIC	UNDOTBS1	ONLINE
_SYSSMU8\$	PUBLIC	UNDOTBS1	ONLINE
_SYSSMU9\$	PUBLIC	UNDOTBS1	ONLINE
_SYSSMU10\$	PUBLIC	UNDOTBS1	ONLINE



# Obtenir des informations sur les segments d'annulation

```
SELECT n.name, s.extents, s.rssize, s.hwmsize, s.xacts,  
s.status  
FROM V$ROLLNAME n, V$ROLLSTAT s  
WHERE n.usn=s.usn;
```

NAME	EXTENTS	RSSIZE	HWMSIZE	XACTS	STATUS
SYSTEM	6	385024	385024	0	ONLINE
_SYSSMU1\$	4	253952	319488	0	ONLINE
_SYSSMU2\$	3	1171456	1171456	0	ONLINE
_SYSSMU3\$	4	253952	253952	0	ONLINE
_SYSSMU4\$	3	1171456	1171456	0	ONLINE
_SYSSMU5\$	4	2220032	2220032	0	ONLINE
_SYSSMU6\$	4	253952	253952	0	ONLINE
_SYSSMU7\$	4	2220032	2220032	0	ONLINE
_SYSSMU8\$	4	253952	385024	0	ONLINE
_SYSSMU9\$	4	2220032	2220032	0	ONLINE
_SYSSMU10\$	4	253952	385024	0	ONLINE

# Obtenir des informations sur les segments d'annulation

Joignez les vues V\$TRANSACTION et V\$SESSION pour vérifier l'utilisation d'un segment d'annulation par les transactions en cours.

- `SELECT s.username, t.xidusn, t.ubafil, t.ubablk, t.used_ublk`
- `FROM V$SESSION s, V$TRANSACTION t`
- `WHERE s.saddr=t.ses_addr;`

# Synthèse

- Ce chapitre vous a permis d'apprendre à:  
Configurer la gestion automatique des annulations  
Créer un tablespace d'annulation (UNDO)  
Dimensionner correctement un tablespace d'annulation (UNDO).

## Chapitre 7

# GÉRER LA SÉCURITÉ DES MOTS DE PASSE ET LES RESSOURCES



# Introduction

---

Vous pouvez définir des limites de ressources système disponible pour chaque utilisateur comme étant une partie de sécurité du domaine de l'utilisateur.

Pour ce faire, vous pouvez priver les consommations incontrôlables des ressources systèmes comme par exemple le temps CPU.

La consommation excessive de certain ressources par un ou plusieurs utilisateurs peut perturber les autres utilisateurs de la base de données.

# Profils

- Un profil est un ensemble nommé contenant les limites relatives aux mots de passe et aux ressources (durée de vie et expiration des mots de passe, historique des mots de passe, vérification de la complexité des mots de passe, verrouillage d'un compte, temps CPU, opération d'entrée/sortie, durée d'inactivité, durée de connexion, espace mémoire, sessions simultanées).
- La commande `CREATE USER` ou `ALTER USER` permet d'affecter des profils aux utilisateurs.
- Les profils peuvent être activés ou désactivés.

Par défaut, affectation du profil `DEFAULT`

# Fonction de profils

- Empêcher les utilisateurs d'exécuter des opérations qui utilisent beaucoup de ressources.
- Garantir que les utilisateurs sont déconnectés de la base de données lorsque leur session reste inactive pendant un certain temps.
- Activer des limites relatives aux ressources de groupe pour les utilisateurs similaires.
- Affecter facilement aux utilisateurs des limites relatives aux ressources.
- Gérer l'utilisation des ressources dans les systèmes de base de données multiutilisateurs complexes et volumineux
- Contrôler l'utilisation des mots de passe.

# Gestion des mots de passe

- Pour mieux contrôler la sécurité des bases de données, les administrateurs de base de données gèrent les mots de passe à l'aide de profils.
- La gestion de mots des passes comprend:
  - Verrouillage d'un compte: (lorsque l'utilisateur ne parvient pas à se connecter au système après un nombre défini de tentative)
  - Durée de vie et expiration de mots de passe.
  - Historique des mots de passe: un nouveau mot de passe ne sera pas réutilisé avant un certain temps ou avant un certain nombre.
  - Vérification de la complexité de mots de passe.



# Activer la gestion des mots de passe

- Configurer la gestion des mots de passe à l'aide de profils que vous affectez aux utilisateurs.
- Verrouillez, déverrouillez et faites expirer des comptes à l'aide de la commande CREATE USER ou ALTER USER.
- Les limites relatives aux mots de passe sont toujours appliquées.
- Pour activer la gestion des mots de passe, exécutez le script utlpwdmg.sql sous le nom utilisateur SYS.

# Verrouillage d'un compte

Paramètre	Description
FAILED_LOGIN_ATTEMPTS	Nombre d'échecs de connexion avant verrouillage du compte
PASSWORD_LOCK_TIME	Durée, en jours, de verrouillage du compte après le nombre d'échecs de connexion défini.

Le compte est déverrouillé automatiquement après un certain délai défini par le paramètre `PASSWORD_LOCK_TIME`

# Durée de vie et expiration des mots de passe

Paramètre	Description
PASSWORD_LIFE_TIME	Durée de vie en jours du mot de passe avant expiration
PASSWORD_GRACE_TIME	Période de grâce, en jours pendant laquelle l'utilisateur peut changer de mot de passe après la première connexion établie une fois le mot de passe expiré

L'utilisateur reçoit un message qui commence après la première tentative de connexion à la base de données, une fois le mot de passe expire, jusqu'à la fin de la période de grâce.

# Historique des mots de passe

Paramètre	Description
PASSWORD_REUSE_TIME	Période en jours pendant laquelle un mot de passe ne peut pas être réutilisé
PASSWORD_REUSE_MAX	Nombre maximum de réutilisations d'un mot de passe.

## Vérifier les mots de passe

Paramètre	Description
PASSWORD_VERIFY_FUNCTION	Fonction PL/SQL qui vérifie la complexité d'un mot de passe avant que celui-ci ne soit affecté

# Fonction de vérification de mot de passe `VERIFY_FUNCTION`

- Longueur minimale de quatre caractères
- Le mot de passe doit être différent du nom utilisateur
- Il doit comporter au moins une lettre, un caractère numérique et un caractère spéciale.
- Il doit comprendre au moins trois lettres différentes par rapport à l'ancien mot de passe.
- Le serveur fournit une fonction de vérification de complexité `VERIFY_FUNCTION`

# Créer un profil: paramètre de mot de passe

```
CREATE PROFILE grace_5 LIMIT  
FAILED_LOGIN_ATTEMPTS 3  
PASSWORD_LOCK_TIME UNLIMITED  
PASSWORD_LIFE_TIME 60  
PASSWORD_GRACE_TIME 10  
PASSWORD_REUSE_TIME 30  
PASSWORD_REUSE_MAX UNLIMITED
```

```
PASSWORD_VERIFY_FUNCTION
```

verify\_function:  
PASSWORD\_GRACE\_TIME: Nombre  
de  
jours durant les quels, une fois la  
période

de grâce commence un avertissement

PASSWORD\_VERIFY\_FUNCTION:  
permet et la connexion est autorisé.

de transmettre une fonction PL/SQL de  
vérification de complexité de mot de

FAILED\_LOGIN\_ATTEMPTS

Définit le nombre d'échecs de  
connexion au compte utilisateur  
avant le verrouillage du compte.

PASSWORD\_LIFE\_TIME:

nombre de jours durant lesquels  
le même mot de passe peut être  
utilisé avant son expiration

PASSWORD\_LOCK\_TIME

durée du verrouillage d'un  
compte utilisateur après échecs  
consécutifs de connexion.

PASSWORD\_REUSE\_MAX : le  
nombre de changement de mot  
de passe nécessaires avant que  
le mot de passe puisse être  
utilisé.

# Modifier un profil: paramètres de mot de passe

```
ALTER PROFILE default LIMIT  
FAILED_LOGIN_ATTEMPTS 3  
PASSWORD_LIFE_TIME 60  
PASSWORD_GRACE_TIME 10
```

# Supprimer un profil: paramètres de mot de passe

La commande DROP PROFILE permet de supprimer un profil  
Vous ne pouvez pas supprimer le profil DEFAULT  
L'option CASCADE retire le profil à l'utilisateur auquel il a été affecté.

```
DROP PROFILE developer_prof;
```

```
DROP PROFILE developer_prof CASCADE;
```

# Gestion des ressources

Les limites relatives à la gestion des ressources peuvent s'appliquer au niveau session, au niveau appel ou aux deux.

Les limites peuvent être définies par des profils via la commande CREATE PROFILE.

Vous pouvez activer les limites relatives aux ressources à l'aide:

- Du paramètre d'initialisation RESOURCE\_LIMIT,
- De la commande ALTER SYSTEM



# Activer les limites relatives aux ressources

- Affecter la valeur TRUE au paramètre d'initialisation RESOURCE\_LIMIT
- Activer le paramètre à l'aide de la commande ALTER SYSTEM pour appliquer les limites relatives aux ressources.

```
● ALTER SYSTEM SET  
  RESOURCES_LIMIT=TRUE;
```

# Définir des limites relatives aux ressources au niveau session

Ressource	Description
CPU_PER_SESSION	Temps CPU total calculé en centièmes de secondes
SESSION_PER_USER	Nombre de sessions simultanées autorisées pour chaque nom utilisateur
CONNECT_TIME	Temps de connexion calculées en minutes
IDLE_TIME	Période d'inactivité calculées en minutes
LOGICAL_READS_PER_SESSION	Nombre de blocs de données (lecture physiques et logiques)
PRIVATE_SGA	Espace privé de la mémoire SGA mesuré en octets (dans le cas d'un serveur partagé uniquement)

# Définir des limites relatives aux ressources au niveau session

- La durée d'inactivité (IDLE\_TIME) n'est calculé que pour le processus serveur.
- LOGICAL\_READS\_PER\_SESSION limite le nombre total d'opérations de lecture en mémoire et sur disque. Vous pouvez limiter ces opérations pour empêcher les instructions qui effectuent de nombreuses opérations d'E/S de saturer la mémoire et de monopoliser le disque.

# Définir les limites relatives aux ressources au niveau appel

Ressource	Description
CPU_PER_CALL	Temps CPU par appel en centième de secondes
LOGICAL_READS_PER_CALL	Nombre de blocs de données pouvant être lus par appel

Les limites au niveau appel s'applique pour chaque appel effectué pendant l'exécution d'une instruction SQL

Lorsqu'une limite au niveau appel est dépassée:

1. Le traitement de l'instruction est interrompu
2. L'instruction est annulée
3. Toutes les instructions précédentes restent inchangées
4. La session utilisateur reste connectée

# Créer un profil: limites relatives aux ressources

```
CREATE PROFILE developer_prof LIMIT  
SESSION_PER_USER 2  
CPU_PER_SESSION 1000  
IDLE_TIME 60  
CONNECT_TIME 480;
```

# Database Resource Manager

Resource Manager donne à l'administrateur de la base de données plus de contrôle sur la décision d'aménagement des ressources, de telle sorte que l'allocation de ressource peut favoriser les affaires objectives de l'entreprise. Il donne l'habileté de données des priorités pour certains travaux sur d'autres avec la disponibilité de la base de données implique la fonctionnalité et la performance.

# Gérer les ressources à l'aide de Database Resource Manager

- Le serveur Oracle exerce un contrôle accru sur les décisions relatives à la gestion des ressources
- Éléments de Database Resource Manager
  - Groupe de consommateurs de ressources
  - Plan d'allocation des ressources
  - Méthode d'allocation des ressources
  - Directives du plan d'allocation de ressources
- Le package DBMS\_RESOURCE\_MANAGER permet de créer et de gérer des éléments.
- Le privilège ADMINISTER\_RESOURCE\_MANAGER est requis.

# Eléments de Database Resource Manager

- Groupe de consommateurs de ressources: Groupe d'utilisateurs ou de sessions constitué en fonction des besoins liés au traitement des ressources.
- Plan d'allocation de ressources: Contient des directives indiquant le mode d'allocation des ressources aux groupes de consommateurs de ressources
- Méthodes d'allocation des ressources: méthode utilisée par Database Resource Manager pour l'allocation d'une ressource spécifique.
- Directives du plan d'allocation de ressources: permet aux administrateurs d'associer des groupes de consommateurs de ressources à des plans spécifiques et d'allouer des ressources à ces groupes



# Gérer les ressources à l'aide de Database Resource Manager

- Les plans d'allocation de ressources indiquent quels groupes de consommateurs de ressources leur appartient.
- Ils contiennent des directives précisant le mode d'allocation des ressources aux groupes de consommateurs.

# Directives du plan d'allocation de ressources

- Database Ressource Manager offre plusieurs méthodes d'allocation des ressources:
- Méthode CPU
- Pool de sessions actif et mise en file d'attente
- Limite du degré de parallélisme
- Changement automatique de groupe de consommateurs
- Durée d'exécution maximale estimée
- Quota d'annulation

# Directives du plan de ressources

- **Méthode CPU:** Permet d'indiquer le mode d'allocation des ressources de la CPU aux groupes de consommateurs.
- **Pool de sessions actif et mise en file d'attente:** vous pouvez contrôler le nombre maximal de sessions actives simultanées autorisées au sein d'un groupe de consommateurs. Lorsqu'une session active ne peut pas être lancée en raison de la saturation de la pool, elle est placée dans une file d'attente.
- **Limite de degré de parallélisme:** indique une limite de degré de parallélisme pour toutes les opérations d'un groupe de consommateurs.
- **Changement automatique de groupe de consommateurs:** permet de gérer les ressources en fonction de certains critères. Si les critères ne sont pas satisfaits, les sessions passent automatiquement à un autre groupe de consommateurs.
- **Durée d'exécution maximale estimé:** le DBA peut définir la durée maximale estimée d'exécution d'une opération. Si l'estimation de l'opération est supérieur à la valeur de MAX\_ESTIMATED\_EXEC\_TIME, l'opération ne démarre pas, ce qui permet d'éviter les travaux particulièrement lourds qui consomment trop de ressources systèmes.
- **Pool d'annulation:** un pool d'annulation peut être défini pour chaque groupe de consommateurs de ressources.

# Obtenir des informations sur les limites relatives aux mots de passe et aux ressources

- Vous pouvez interroger les vues suivantes pour obtenir des informations sur les limites relatives aux mots de passe et aux ressources
- DBA\_USERS
- DBA\_PROFILES

## Chapitre 8

# GERER LES UTILISATEURS, LES PRIVILEGES ET LES ROLES



# GÉRER LES UTILISATEURS



# Objectifs

A la fin de ce chapitre, vous pourrez :

- Créer des utilisateurs de base de données
- Modifier et supprimer des utilisateurs existants
- Surveiller les informations relatives aux utilisateurs existants.
- Identifier les privilèges systèmes et objets
- Accorder et révoquer des privilèges
- Créer et modifier des rôles
- Gérer la disponibilité des rôles
- Supprimer des rôles
- Utiliser des rôles prédéfinis
- Afficher des informations sur les rôles à partir du dictionnaire des données

# Objectifs

---

- A la fin de ce chapitre vous pourrez:
- Créer et modifier des rôles
- Gérer la disponibilité des rôles
- Supprimer des rôles
- Utiliser des rôles prédéfinis
- Afficher des informations sur les rôles à partir du dictionnaire des données



# Utilisateurs et sécurité

- **Domaine de sécurité**

Le DBA définit le nom des utilisateurs autorisés à accéder à la BD. Un domaine de sécurité définit les paramètres qui s'appliquent à l'utilisateur.

- **Mécanisme d'authentification:**

- Par le dictionnaire de données

- Par le système d'exploitation

- **Quotas de tablespace**

Les quotas de tablespace contrôle la quantité d'espace physique de stockage alloué à un utilisateur dans les tablespace de la BD.

# Utilisateurs et sécurité

- **Tablespace par défaut**

Le tablespace par défaut définit l'emplacement de stockage des segments créés par un utilisateur si celui-ci n'indique pas explicitement de tablespace lors de la création des segments.

- **Tablespace temporaire :**

Définit l'emplacement d'allocation des extends (ensembles de blocs contigus) par le serveur lorsque l'utilisateur exécute une opération dans laquelle les données triées sont écrites sur disque.

# Utilisateurs et sécurité

- **Verrouillage de compte**

Vous pouvez verrouiller les comptes pour empêcher les utilisateurs de se connecter à la BD.

- **Limites relatives aux ressources**

Vous pouvez définir des limites pour l'utilisation de ressources telles que le temps CPU. Les entrées/sorties et le nombre de sessions ouvertes par l'utilisateur.

# Utilisateurs et sécurité

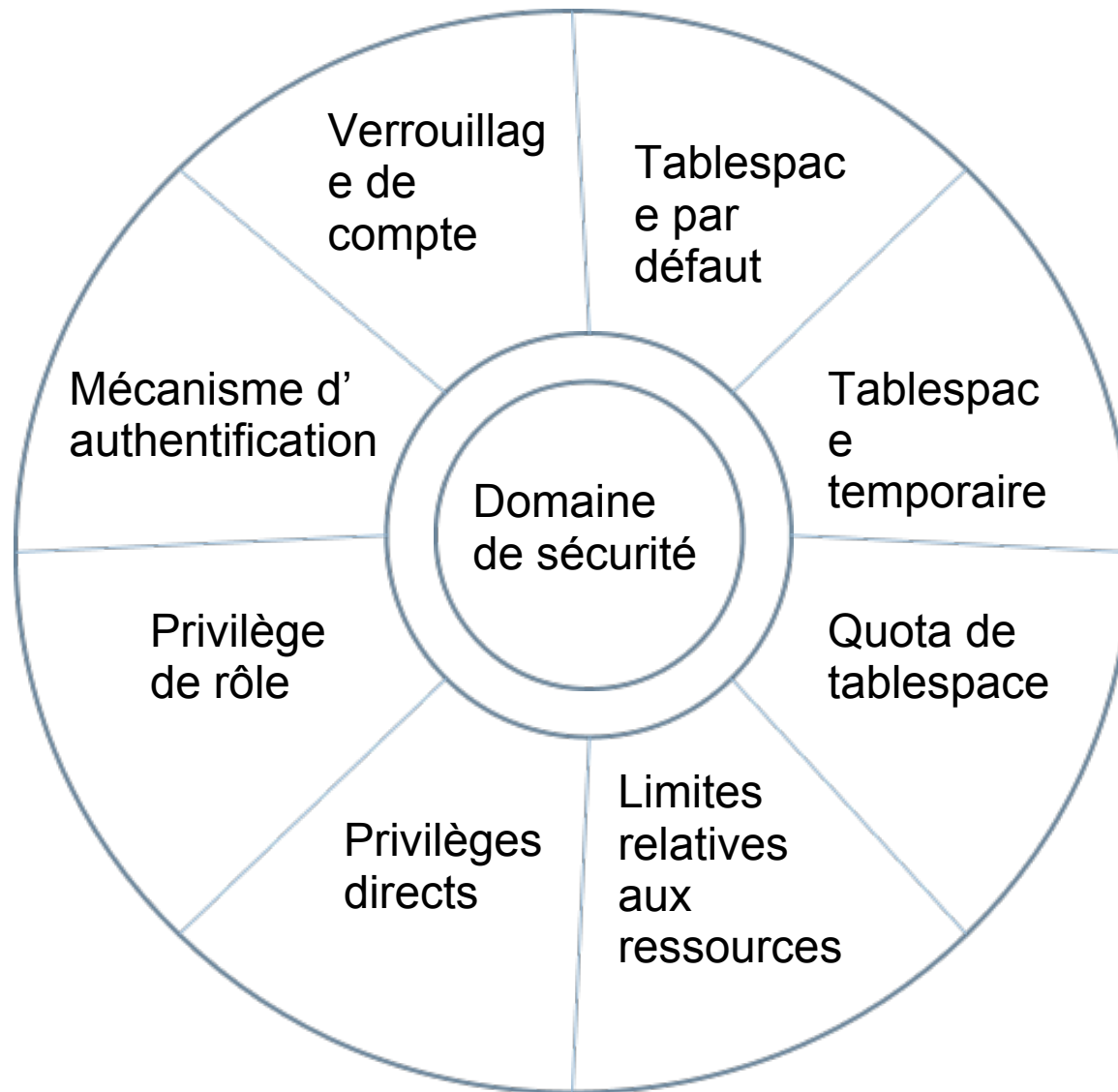
- **Privilège de rôle**

Vous pouvez accorder indirectement des privilèges à un utilisateur à l'aide de rôle.

- **Privilèges directs**

Les privilèges directs permettent de contrôler les actions qu'exécute l'utilisateur dans la BD.

# Utilisateurs et sécurité



# Schéma de base de données

- Un schéma est un ensemble nommé d'objets associés à un utilisateur.
  - Lorsqu'un utilisateur est créé, un schéma correspondant est également créé.
  - Un utilisateur ne peut être associé qu'à un seul schéma.
  - Le nom utilisateur et le nom de schéma sont souvent utilisés indifféremment.
- Objets de schéma
    - Tables
    - Déclencheurs
    - Contraintes
    - Index
    - Vues
    - Séquences
    - Programmes stockés
    - Synonymes
    - Types de données définis par l'utilisateur

# Liste de contrôle pour la création d'utilisateurs

- Identifiez les tablespaces dans lesquels l'utilisateur a besoin de stocker des objets.
- Déterminez les quotas applicables pour chaque tablespace.
- Affecter un tablespace par défaut et un tablespace temporaire.
- Créez un utilisateur.
- Accordez des privilèges et des rôles à l'utilisateur.

# Créer un utilisateur: authentification par la base de données

Définissez le mot de passe initial:

```
CREATE USER aaron  
IDENTIFIED BY soccer  
DEFAULT TABLESPACE data  
DEFAULT TEMPORARY TABLESPACE temp  
QUOTA 15M ON data  
QUOTA 10M ON USERS  
PASSWORD EXPIRE;
```



# Créer un utilisateur: authentification par le système d' exploitation

Le paramètre d'initialisation

OS\_AUTHENT\_PREFIX indique le format des noms utilisateurs. Sa valeur par défaut est OPS\$

```
CREATE USER aaron
```

```
IDENTIFIED EXTERNALLY
```

```
DEFAULT TABLESPACE USERS
```

```
DEFAULT TEMPORARY TABLESPACE temp
```

```
QUOTA 15M ON data
```

```
PASSWORD EXPIRE;
```

# Modifier les quotas de tablespace d'un utilisateur

- Vous pouvez modifier les quotas de tablespace d'un utilisateur dans les cas suivants:
  - Lorsque la taille des tables appartenant à l'utilisateur augmente de manière imprévue,
  - Lorsqu'une application est étendue et nécessite des tables ou des index supplémentaires,
  - Lorsque les objets sont réorganisés et placés dans des tablespaces différents.
  - Procédez comme suit pour modifier le quota de tablespace d'un utilisateur:

```
ALTER USER aaron  
QUOTA 0 ON USERS;
```

Si vous définissez un quota de 0, les objets de l'utilisateur sont conservés dans le tablespace révoqué, mais aucun nouvel espace ne peut leur être alloué.

# Supprimer un utilisateur

La clause **CASCADE** permet de supprimer tous les objets d'un schéma avant de supprimer l'utilisateur.

```
DROP USER aaron;
```

**Vous ne pouvez pas supprimer les utilisateurs qui sont connectés au serveurs Oracle.**

```
DROP USER aaron CASCADE;
```

# Obtenir des informations sur les utilisateurs

- Interroger les vues suivantes pour obtenir des informations sur les utilisateurs:
  - DBA\_USERS
  - DBA\_TS\_QUOTAS

```
SELECT username,  
default_tablespace  
FROM dba_users;
```

# GÉRER LES PRIVILÈGES



# Créer les privilèges

- Un privilège est un droit d'exécution d'un type donné d'instruction SQL ou un droit d'accès à l'objet d'un autre utilisateur.
- Il existe deux types de privilèges utilisateur Oracle:
- Système: permet aux utilisateurs de réaliser certaines actions dans la base de données (Ex. privilège lié à la création de tablespace).
- Objet: permet aux utilisateurs d'accéder à un objet donné et de le manipuler

# Privilèges système

- Il existe plus de 100 privilèges système différents.
- Le mot clé ANY signifie que les utilisateurs disposent du privilège de gestion d'objets dans n'importe quel schéma.
- La commande GRANT permet d'accorder un privilège à un utilisateur ou un groupe d'utilisateurs.
- La commande REVOKE supprime les privilèges.

# Privilèges système: exemples

Catégorie	Exemples
INDEX	CREATE ANY INDEX ALTER ANY INDEX DROP ANY INDEX
TABLE	CREATE TABLE CREATE ANY TABLE ALTER ANY TABLE DROP ANY TABLE SELECT ANY TABLE UPDATE ANY TABLE DELETE ANY TABLE
SESSION	CREATE SESSION ALTER SESSION RESTRICTED SESSION
TABLESPACE	CREATE TABLESPACE ALTER TABLESPACE DROP TABLESPACE UNLIMITED TABLESPACE



# Accorder des privilèges système

- Utilisez la commande GRANT pour accorder des privilèges système.
- Le bénéficiaire peut accorder le privilège système à d'autres utilisateurs grâce à l'option ADMIN

```
GRANT CREATE SESSION TO emi;
```

```
GRANT CREATE SESSION TO emi WITH ADMIN OPTION ;
```

# Révoquer des privilèges système

- Utiliser la commande REVOKE pour révoquer un privilège système accordé à un utilisateur.
- Les utilisateurs qui disposent d'un privilège système avec l'option ADMIN OPTION peuvent révoquer des privilèges systèmes.
- Seuls les privilèges accordés via la commande GRANT peuvent être révoqués.

```
REVOKE CREATE TABLE FROM emi;
```

# Privilèges objet

Priv.objet	Table	Vue	Séquence	Procédure
ALTER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DELETE	<input type="checkbox"/>	<input type="checkbox"/>		
EXECUTE				<input type="checkbox"/>
INDEX	<input type="checkbox"/>	<input type="checkbox"/>		
INSERT	<input type="checkbox"/>	<input type="checkbox"/>		
REFERENCES	<input type="checkbox"/>			
SELECT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
UPDATE	<input type="checkbox"/>	<input type="checkbox"/>		

Un privilège objet est un privilège ou droit autorisant la réalisation d'une action donnée sur une table, une vue, une procédure, une fonction ou un package spécifique.

# Accorder et révoquer des privilèges objet

- Utilisez la commande GRANT pour accorder des privilèges objet.
- L'objet doit se trouver dans votre schéma ou vous devez avoir reçu le privilège avec l'option GRANT OPTION.

```
GRANT EXECUTE ON dbms_output TO jeff;
```

```
GRANT UPDATE ON emi.customers TO jeff WITH GRANT OPTION;
```

- Utiliser la commande REVOKE pour révoquer des privilèges objet.
- Seul l'utilisateur qui a accordé un privilège objet peut le révoquer.

```
REVOKE SELECT ON emi.orders FROM jeff ;
```

# Obtenir des informations sur les privilèges

Interrogez les vues suivantes pour obtenir des informations sur les privilèges:

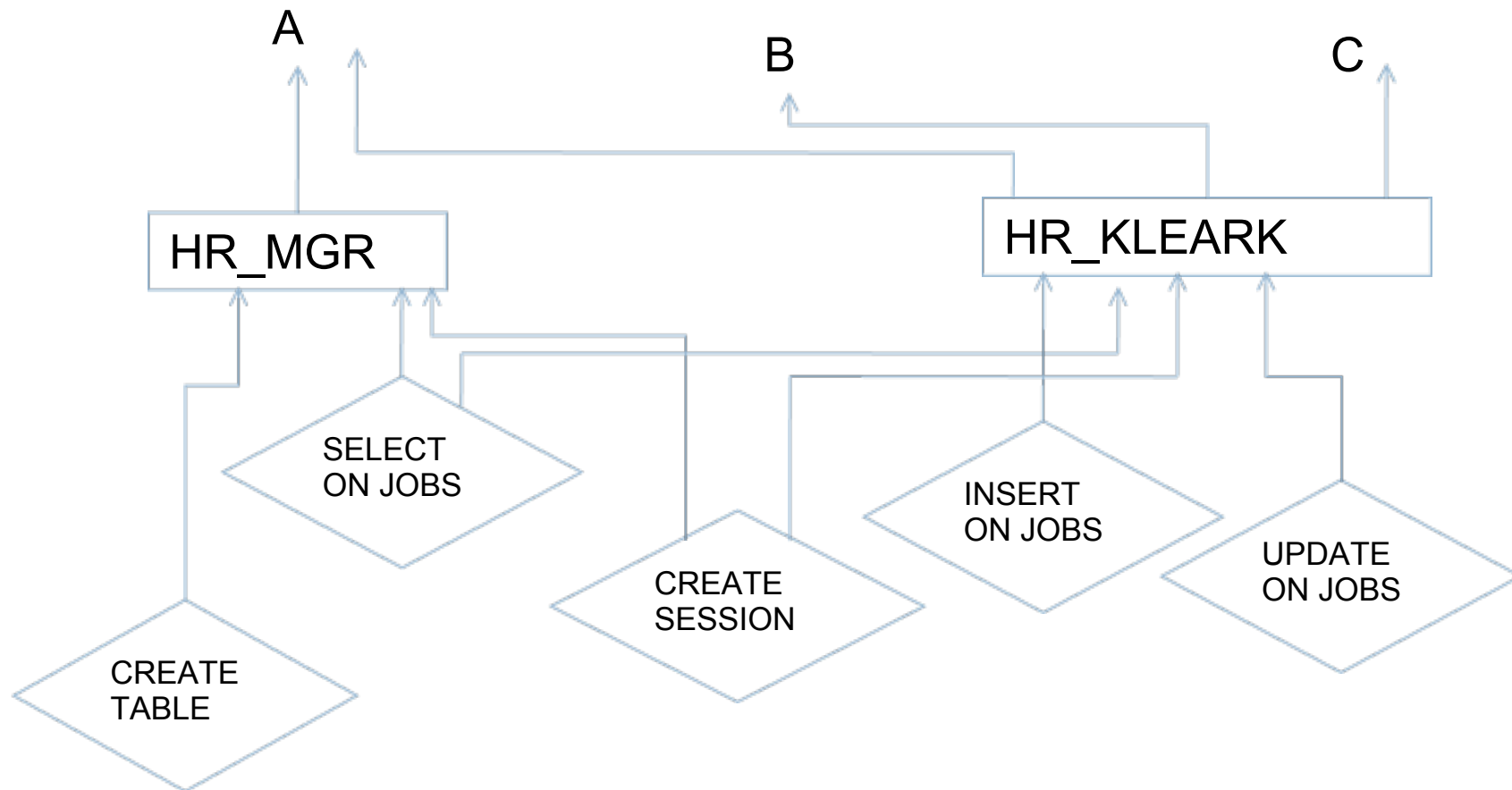
- **DBA\_SYS\_PRIVS**: Affiche la liste des privilèges système accordés aux utilisateurs et aux rôles
- **SESSION\_PRIVS**: affiche la liste des privilèges auquel l'utilisateur a accès.
- **DBA\_TAB\_PRIVS**: Affiche la liste de tous les octrois de privilèges sur les objets de la base de données.
- **DBA\_COL\_PRIVS**: Décrit tous les octrois de privilèges sur les colonnes de la base de données

# GÉRER LES RÔLES



# Rôles

- Les rôles sont des groupes nommées de privilèges associés qui sont accordés à des utilisateurs ou à d'autres rôles.
- Ils facilitent l'administration des privilèges dans une base de données.





# Avantages des rôles

- Gestion simplifiée des privilèges
  - Au lieu d'accorder les mêmes privilèges à plusieurs utilisateurs. Vous pouvez accorder les privilèges à un rôle et associer ce rôle à chaque utilisateur.
- Gestion dynamique des privilèges
  - En cas de modification des privilèges associés à un rôle, tous les utilisateurs auxquels ce rôle a été accordé bénéficient automatiquement et immédiatement des nouveaux privilèges.

# Avantages des rôles

- Disponibilité sélective des privilèges
  - Vous pouvez activer et désactiver les rôles pour activer et désactiver temporairement les privilèges.
- Octroi possible via le système d'exploitation
  - Vous pouvez utiliser des commandes ou des utilitaires du système d'exploitation pour accorder des rôles aux utilisateurs dans la base de données.

# Créer des rôles

- Non identifier:

```
CREATE ROLE oe_clerk;
```

- Identifier par mot de passe

```
CREATE ROLE oe_clerk  
IDENTIFIED BY bonus;
```

- Identifier de manière externe

```
CREATE ROLE oe_clerk  
IDENTIFIED EXTERNALLY;
```

**EXTERNALLY** indique que l'utilisateur doit avoir reçu une autorisation d'un service externe (tel que le OS ou un service tiers) pour activer le rôle.

# Role prédéfinis

Rôles	Description
CONNECT, RESSOURCE, DBA	Fournis pour garantir une compatibilité descendante avec les versions précédentes d'oracle
EXP_FULL_DATABASE	Privilège d'export de la base de données
IMP_FULL_DATABASE	Privilège d'import de la base de données
DELETE_CATALOG_ROLE	Privilège DELETE sur les tables du dictionnaire de données
EXECUTE_CATALOG_ROLE	Privilège EXECUTE sur les packages du dictionnaire de données
SELECT_CATALOG_ROLE	Privilège SELECT sur les tables du dictionnaire de données

# Modifier des rôles

- Utiliser ALTER ROLE pour modifier la méthode d'authentification.
- Cette commande requiert l'option ADMIN ou le privilège ALTER ANY ROLE.

```
ALTER ROLE oe_clerk  
IDENTIFIED BY ORDER;
```

```
ALTER ROLE hr_clerk  
IDENTIFIED EXTERNALLY;
```

```
ALTER ROLE oe_clerk  
NOT IDENTIFIED ;
```

NOT IDENTIFIED indique qu'aucune vérification n'est nécessaire lorsque le rôle est activé.

# Accorder des rôles

- Pour accorder un rôle, utilisez la commande GRANT:

```
GRANT oe_clerk TO scott;
```

```
GRANT hr_clerk TO hr_manager;
```

```
GRANT hr_manager TO scott WITH ADMIN OPTION;
```

- WITH ADMIN OPTION permet au bénéficiaire d'accorder le rôle à d'autres utilisateurs ou rôles. Si vous accordez un rôle avec cette option, le bénéficiaire peut l'accorder à d'autres utilisateurs ou le révoquer, le modifier ou le supprimer.
- Le privilège système GRANT ANY ROLE permet d'accorder des rôles aux autres utilisateurs ou révoqué.

# Etablir des rôles par défaut

- Un utilisateur peut se voir accorder un grand nombre de rôles.
- Un utilisateur peut se voir accorder un rôle par défaut
- Vous pouvez limiter le nombre de rôle par défaut d'un utilisateur.

```
ALTER USER scott  
DEFAULT ROLE hr_clerk, oe_clerk;
```

```
ALTER USER scott DEFAULT ROLE ALL;
```

```
ALTER USER scott DEFAULT ROLE ALL EXCEPT  
Hr_clerk;
```

```
ALTER USER scott DEFAULT ROLE NONE;
```

# Rôles d'application

- Seuls les packages PL/SQL autorisés peuvent activer des rôles d'application
- La clause de package USING permet de créer un rôle d'application
- Les développeurs d'applications non pas besoin d'intègre des mots de passe aux applications pour sécuriser un rôle. Ils peuvent en effet créer un rôle d'application et désigner le package PL/SQL autorisé à l'activer.

```
CREATE ROLE admin_role  
IDENTIFIED USING hr8.employee;
```



# Activer et désactiver les rôles

- Désactiver un rôle accordé à un utilisateur pour le révoquer temporairement les privilèges associés.
- Activez un rôle pour l'accorder temporairement
- La commande SET ROLE permet d'activer et de désactiver les rôles.
- Les rôles par défaut d'un utilisateur sont activés à la connexion
- Un mot de passe peut être requis pour activer des rôles (celles activé par la commande SET ROLE).

# Activer et désactiver les rôles

- Utiliser la commande `ALTER USER ... DEFAULT ROLE ..;` pour indiquer les rôles qui seront activés à la connexion de l'utilisateur. Tous les autres seront alors désactivés.
- Les rôles par défaut accordé à un utilisateur ne nécessitent pas de mot de passe; ils sont activés à la connexion, comme les rôles sans mot de passe.
- Vous ne pouvez pas **activer** un rôle à partir d'une procédure stockée.

# Activer et désactiver les rôles

- La commande SET ROLE active et désactive tous les autres rôles accordés à l'utilisateur.

```
SET ROLE { role [ IDENTIFIED BY password ]  
[ , role [ IDENTIFIED BY password ] ] ...  
| ALL [ EXCEPT role [, role ] ...]  
| NONE  
}
```

- Vous ne pouvez pas utiliser la clause ALL pour activer les rôles nécessitant des mots de passes.

# Activer et désactiver les rôles

```
SET ROLE hr_clerk;
```

```
SET ROLE oe_clerk IDENTIFIED BY order;
```

```
SET ROLE ALL EXCEPT oe_clerk ;
```

# Révoquer des rôles accordés à des utilisateurs

- La révocation d'un rôle accordé à un utilisateur requiert l'option ADMIN OPTION ou le privilège GRANT ANY ROLE.
- Pour révoquer un rôle, utilisez la syntaxe suivante:

```
REVOKE oe_clerk FROM scott;
```

```
REVOKE hr_manager FROM PUBLIC;
```

# Supprimer des rôles

- Lorsque vous supprimer un rôle:
  - Il est retiré à tous les utilisateurs et rôles auxquels il était accordé,
  - Il est supprimé de la base de données.
- La suppression d'un rôle requiert l'option ADMIN OPTION ou le privilège DROP ANY ROLE.
- Pour supprimer un rôle, utilisez la syntaxe suivante:

```
DROP ROLE hr_manager ;
```

# Instructions relatives à l'én

## rôles



utilisateurs

5

3

Rôles utilisateur

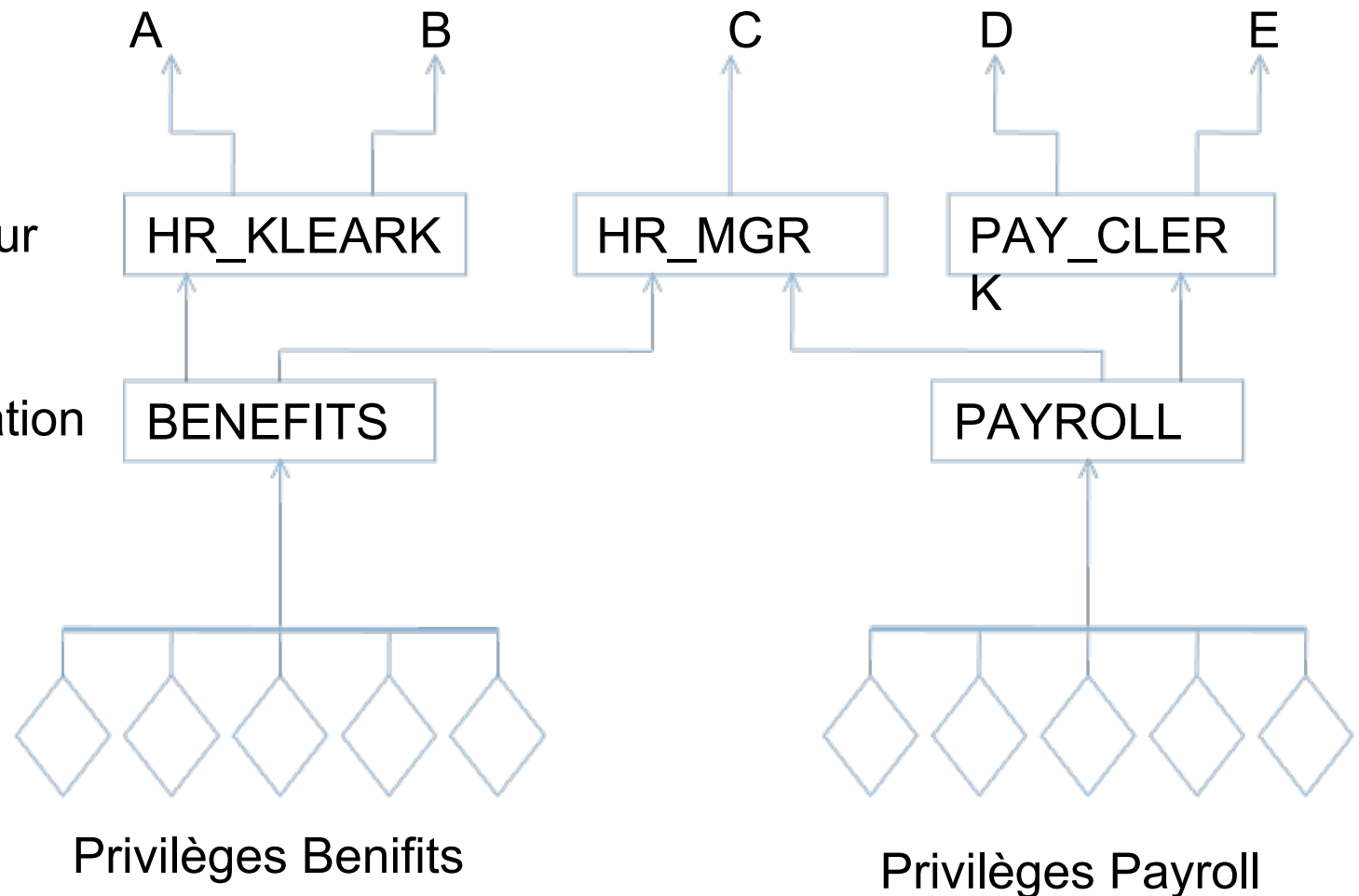
4

1

Rôles d'application

2

Privilèges d'  
application

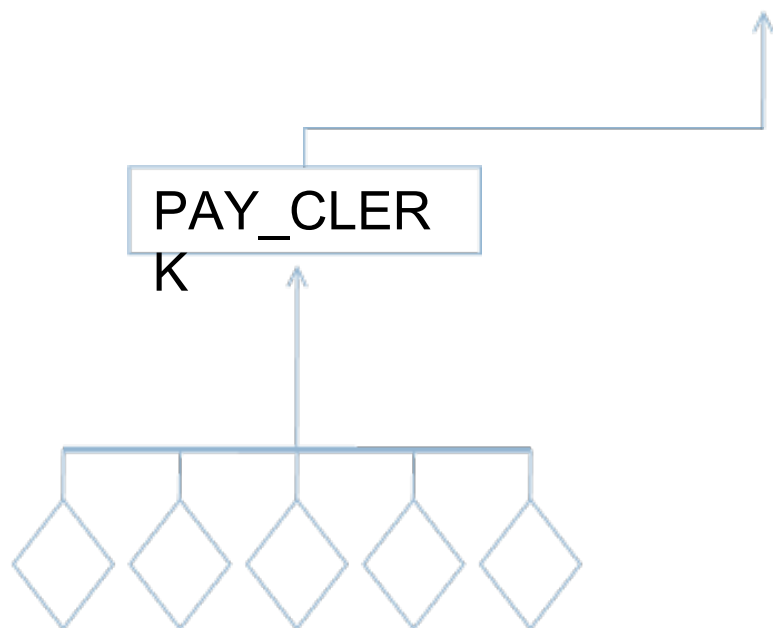


# Règles d'utilisation des mots de passe et des rôles par défaut

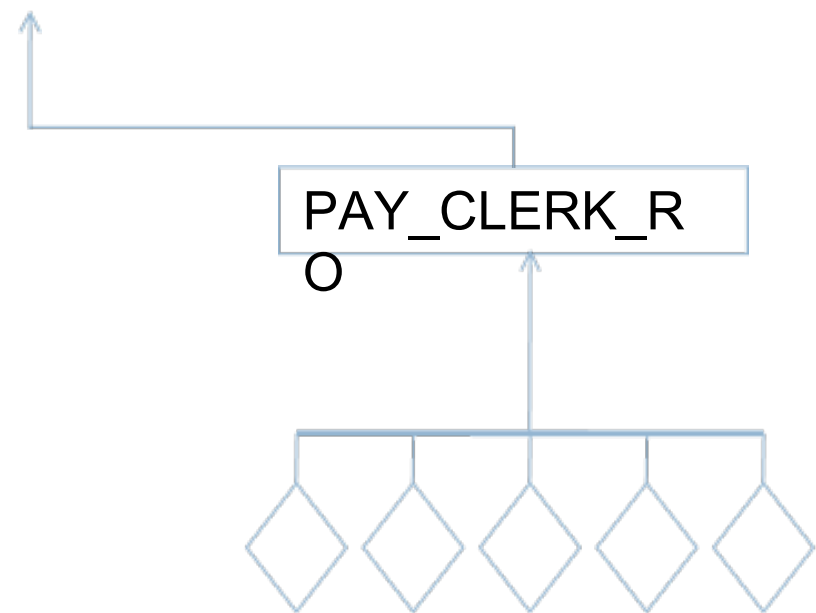


Protection par mot de passe  
(pas la valeur par défaut)

Rôle par défaut



Privilèges INSERT,  
UPDATE, DELETE et  
SELECT



Privilèges SELECT



# Pour obtenir des informations sur les rôles

Pour obtenir des informations sur les rôles, interrogez les vues suivantes du dictionnaire de données:

- **DBA\_ROLES**: Tous les rôles qui existent dans la base de données.
- **DBA\_ROLE\_PRIVS**: Rôles accordés à des utilisateurs et des rôles.
- **ROLE\_ROLE\_PRIVS**: Rôles accordés à des rôles
- **DBA\_SYS\_PRIVS**: Privilèges systèmes accordés à des rôles
- **ROLE\_TAB\_PRIVS**: Privilèges objet accordés à des rôles
- **SESSION\_ROLES**: Rôles activés par l'utilisateur.

# Synthèse

- Ce chapitre vous a permis d'apprendre à:
  - Créer des utilisateurs en indiquant le mécanisme de contrôle par mot de passe approprié
  - Contrôler l'utilisation de l'espace.
  - Identifier les privilèges systèmes et objets
  - Accorder et révoquer des privilèges
  - Créer des rôles
  - Accorder des privilèges à des rôles
  - Accorder des rôles à des utilisateurs ou à des rôles.
  - Etablir des rôles par défaut