



## 1 - Architecture du SGBD Oracle

---

# Plan

---

- ❑ Introduction
- ❑ Architecture d'un serveur de BD Oracle
- ❑ Structure de la mémoire
- ❑ Structure des processus
- ❑ Structure de stockage de données
- ❑ Interagir avec une base Oracle : Mémoire, processus et stockage

# Plan

---

- ❑ **Introduction**
- ❑ Architecture d'un serveur de BD Oracle
- ❑ Structure de la mémoire
- ❑ Structure des processus
- ❑ Structure de stockage de données
- ❑ Interagir avec une base Oracle : Mémoire, processus et stockage

# Introduction

---

## Base de données (DataBase)

- Ensemble structuré d'éléments d'information, agencés sous forme de Tables, dans lesquels les données sont organisées selon certains critères en vue de permettre leur exploitation.



## Système de gestion de Base de Données ( DataBase Management System )

- Système qui gère et contrôle l'accès à une base de données, assurant ainsi une interface normalisée entre les applications et les bases de données



# Tâches d'un administrateur de BD

---

L'administrateur de base de données (appelé **DBA**) est responsable du contrôle et de la configuration du système et des utilisateurs. Il a pour tâches :

- Installer le serveur Oracle et les outils applicatifs, créer les bases de données.
- Assurer la maintenance et la disponibilité de la base de données.
- Créer la structure logique de stockage, c'est à dire les tablespaces, les tables, les vues et les indexes.
- Manager la structure physique de stockage, comprenant les fichiers de données, les fichiers de contrôle et les fichiers de redo log.
- Allouer et prévoir l'espace disque système de stockage nécessaire aux spécifications de la base de données.
- Maintenir, contrôler et monitorer les accès des utilisateurs.
- Assurer la sécurité du système.
- Monitorer les performances de la base de données.
- Créer un plan de sauvegarde et de récupération.

# Produits Oracle

---

- **Oracle Database** : La base de données Oracle est la première base de données conçue pour le grid computing (calcul distribué) d'entreprise, qui est le moyen le plus flexible et le moins coûteux pour gérer les informations et les applications.
- **Oracle WebLogic Application Server** : Le serveur Oracle certifié J2EE (Java 2 Enterprise Edition) intègre tous les éléments nécessaires au développement et au déploiement des applications basées sur le Web. Le serveur d'applications déploie les portails e-business, les services Web et les applications transactionnelles, telles que les applications PL/SQL, Oracle Forms et Java EE.
- **Oracle Applications** : Oracle E-Business Suite est un ensemble complet d'applications métier pour la gestion et l'automatisation des processus au sein d'une organisation.
- **Oracle Collaboration Suite** : est un système intégré unique pour toutes les données de communication d'une organisation (voix, email, fax, sans fil, informations calendaires et fichiers).
- **Oracle Developer Suite** : est un environnement complet et intégré qui associe des outils de développement d'applications et des outils décisionnels.

# Présentation d'Oracle Database 11g

---

- Oracle Database 11g est un SGBD relationnel.
- Disponible pour la plateforme Unix et Windows
- Il permet d'assurer :
  - La définition et la manipulation des données
  - La cohérence des données
  - La confidentialité des données
  - L'intégrité des données
  - La sauvegarde et la restauration des données
  - La gestion des accès concurrents

# Présentation d'Oracle Database 11g

---

- Oracle Database 11g existe dans plusieurs éditions :
  - **Oracle Database 11g Enterprise Edition (EE)** assure une gestion efficace, fiable et sécurisée des données pour les applications transactionnelles (OLTP) les plus critiques, les datawarehouses traitant beaucoup de requêtes et les applications de gestion des contenus ou Web2.0. Cette édition est disponible sur serveurs isolés ou des serveurs en cluster
  - **Oracle Database 11g Standard Edition One (SE1)** est conçue pour un déploiement dans de petites et moyennes sociétés. Elle offre une facilité d'utilisation.
  - **Oracle Database 11g Personal Edition (PE)** supporte le développement et le déploiement pour un seul utilisateur, avec une compatibilité totale avec Oracle Database Standard Edition One, Oracle Database Standard Edition et Oracle Database Enterprise Edition.
  - **Oracle Database Express Edition (XE)** est une base de données légère basée sur le même code que l'Enterprise Edition (EE). Elle est utilisable gratuitement pour le développement, le déploiement et la distribution.

# Plan

---

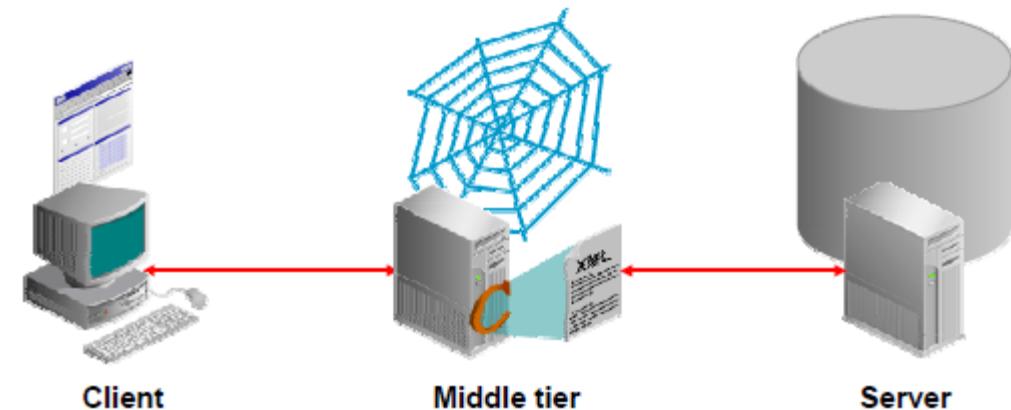
- ❑ Introduction
- ❑ **Architecture d'un serveur de BD Oracle**
- ❑ Structure de la mémoire
- ❑ Structure des processus
- ❑ Structure de stockage de données
- ❑ Interagir avec une base Oracle : Mémoire, processus et stockage

# Connexion à un Serveur Oracle (1/3)

---

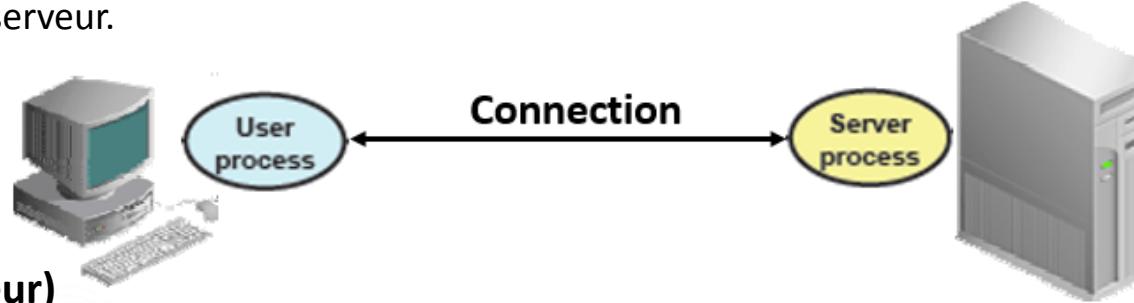
Il existe trois types de connexions grâce auxquelles un utilisateur peut accéder à un Serveur Oracle :

- **Connexion locale** : un utilisateur est directement connecté sur la machine faisant office de Serveur Oracle.
- **Connexion Deux Tiers (Client/Serveur)** : un utilisateur se connecte à partir d'une machine directement connectée à un Serveur Oracle.
- **Connexion Multi Tiers** : Dans une architecture multi tiers, la machine de l'utilisateur se connecte à un Serveur applicatif (Par exemple un Serveur Web) qui lui-même va se connecter au serveur Oracle pour récupérer les données issues de la base de données.



# Connexion à un Serveur Oracle (2/3)

Lorsque qu'un utilisateur est connecté à une machine sur laquelle réside un Serveur Oracle, deux processus sont invoqués: Le processus utilisateur et le processus serveur.



## User Process (Processus utilisateur)

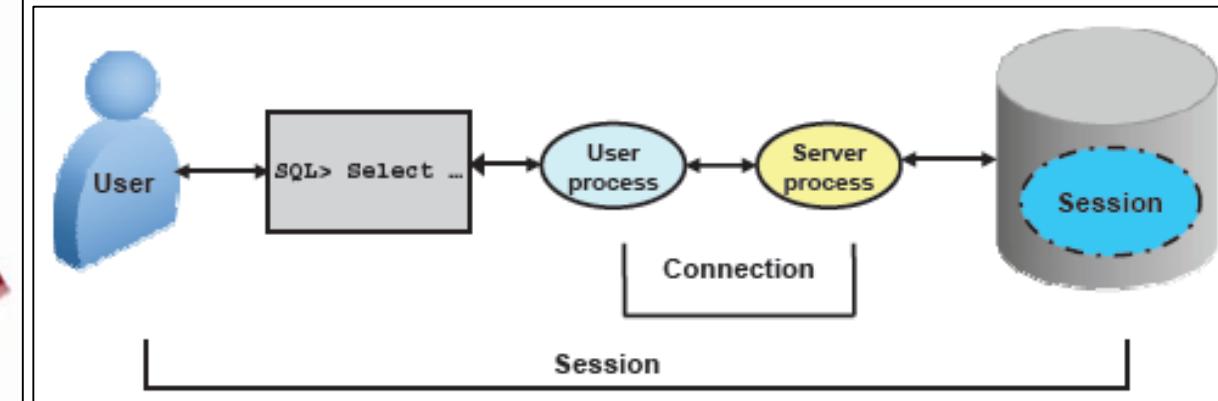
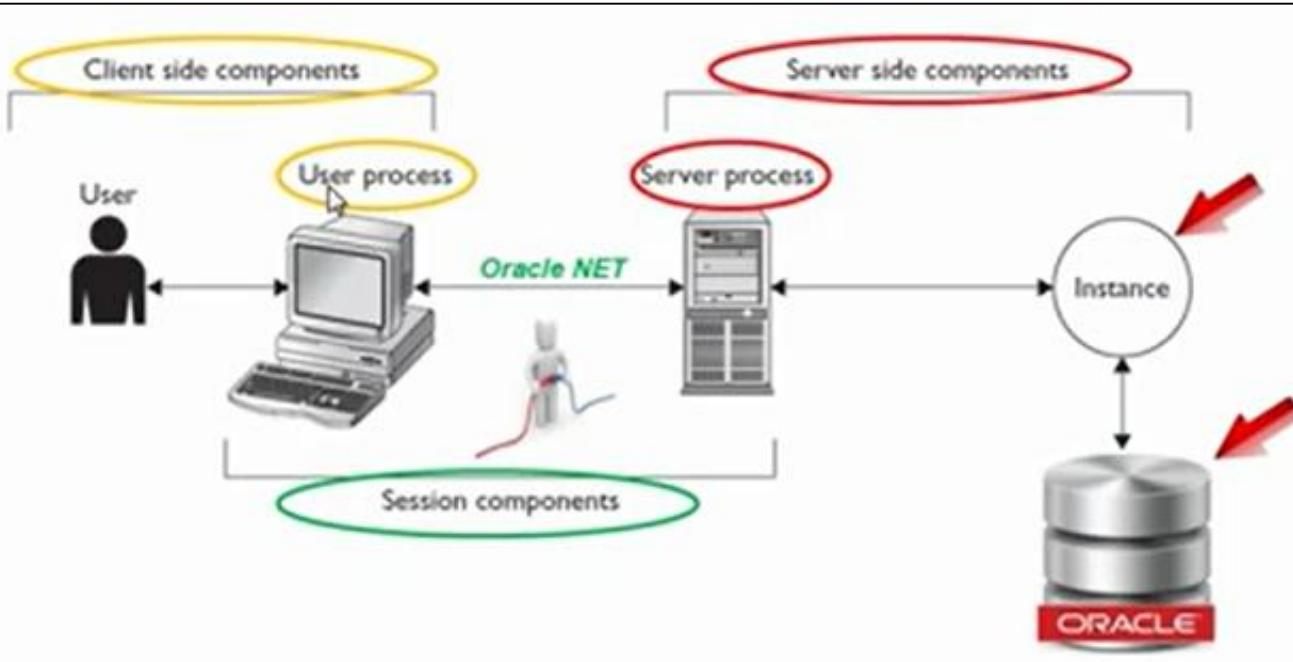
- Un processus utilisateur représente une application ou un outil (outil d'administration, une application, SQL\*Plus) qui se connecte à la base de données Oracle.
- Un processus utilisateur est un mécanisme qui exécute le code d'une application ou d'un outil Oracle.
- Il peut être sur la même machine que la base, ou il peut exister sur un client distant et utiliser un réseau pour accéder à la base.

**Server Process (Processus Serveur)** : permet l'interaction entre le **processus utilisateur** et le serveur Oracle. Il a pour tâches :

- Analyse (parse) et exécution des instructions SQL émises via l'application.
- Lecture des blocs de données nécessaires à partir des fichiers de données sur disque et écriture dans des tampons de base de données partagés (Buffer Cache de l'instance Oracle)
- Renvoi des résultats de telle sorte que l'application puisse traiter les informations.

# Connexion à un Serveur Oracle (3/3)

- Le lien entre le **processus utilisateur** et le **processus serveur** est appelé une **connexion**.
- Une connexion spécifique entre un utilisateur et un serveur Oracle est appelé une **Session**.
- La **session** démarre lorsque la connexion de l'utilisateur est validée par le serveur Oracle et se termine lorsqu'il se déconnecte ou lorsqu'une fin de connexion prématuée se produit.



# Architecture du serveur de BD oracle

L'architecture d'un **serveur de base de données Oracle** se base sur deux composants :

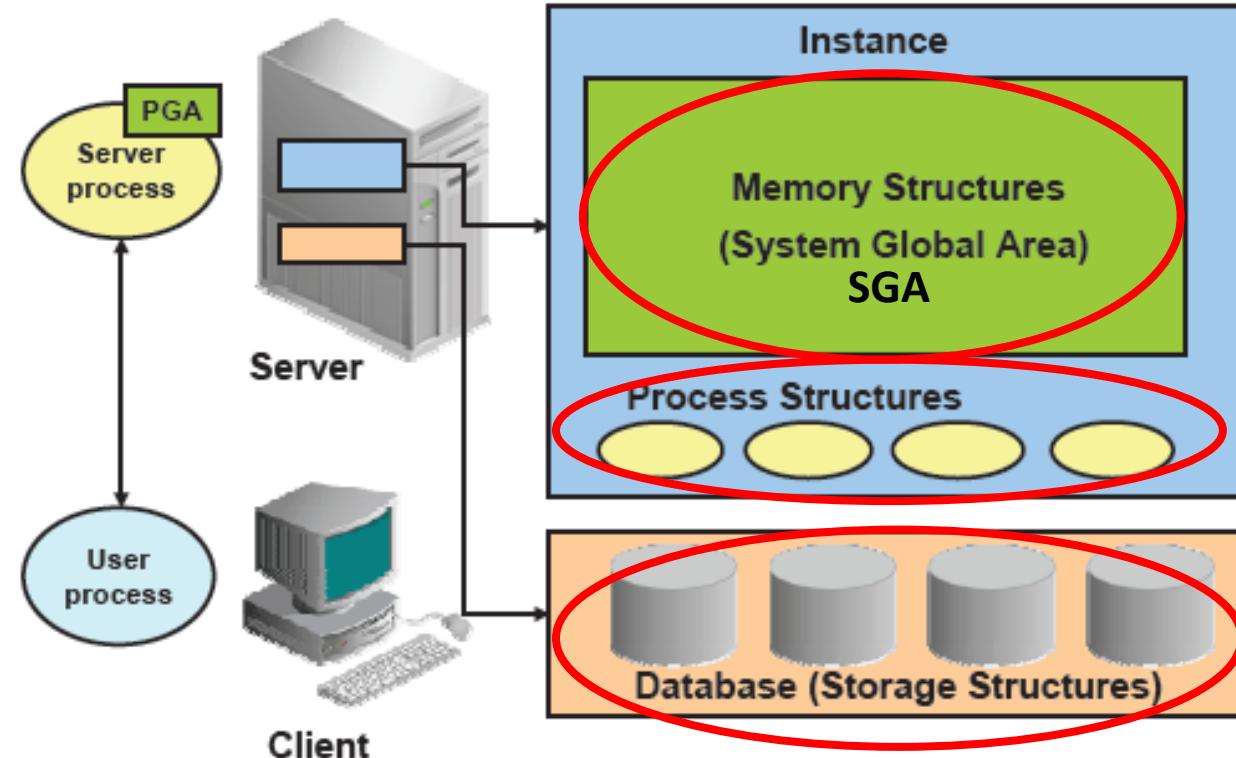
1. Instance
2. Stockage de Base de Données

Elle comprend trois principaux types de structure :

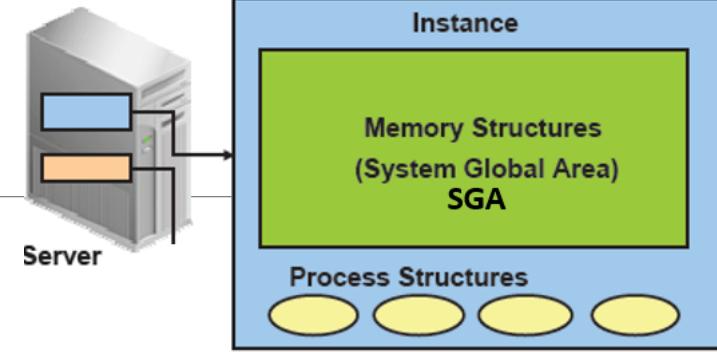
- **Memory Structures (Structures mémoire)**
- **Process Structures (Structures de processus)**
- **Storage Structures (Structures de stockage)**

Il y a deux mémoires principales :

- **System Global Area (SGA)**
- **Program Global Area (PGA)**



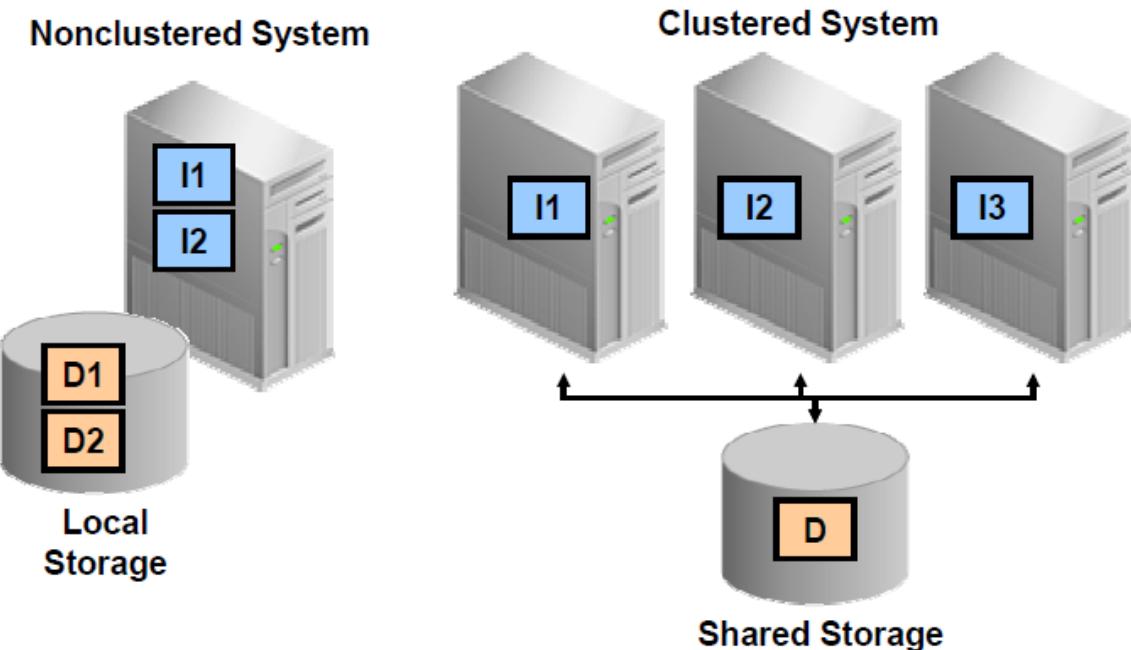
# Instance (1/2)



- Instance Oracle est un programme chargé sur la RAM du serveur
- Elle existe uniquement dans la mémoire et créé à chaque fois que la base de donnée démarre
  - A chaque démarrage d'une instance, une zone de mémoire partagée appelée mémoire **SGA (System Global Area)** est allouée et les **processus en arrière-plan** sont lancés.
- L'instance n'est pas persistante et disparaît à chaque fois que la base de données redémarre
- Instance Oracle contient :
  - **Des structures de mémoire partagées** : des zones spéciales utilisées pour stocker dans la mémoire les données **les plus importantes** ainsi que leurs métadonnées
    - **Métadonnées** est une donnée qui décrit d'autre donnée. Exemple, dans une table « Employé », les lignes représentent les données et les colonnes représentent la structure de la table. Cette structure est la métadonnée.
  - **Des processus oracle** : qui effectuent des opérations de base de données très adaptées. Ces opérations comprennent l'écriture de données sur le disque, la cohérence de la base de données et plus encore

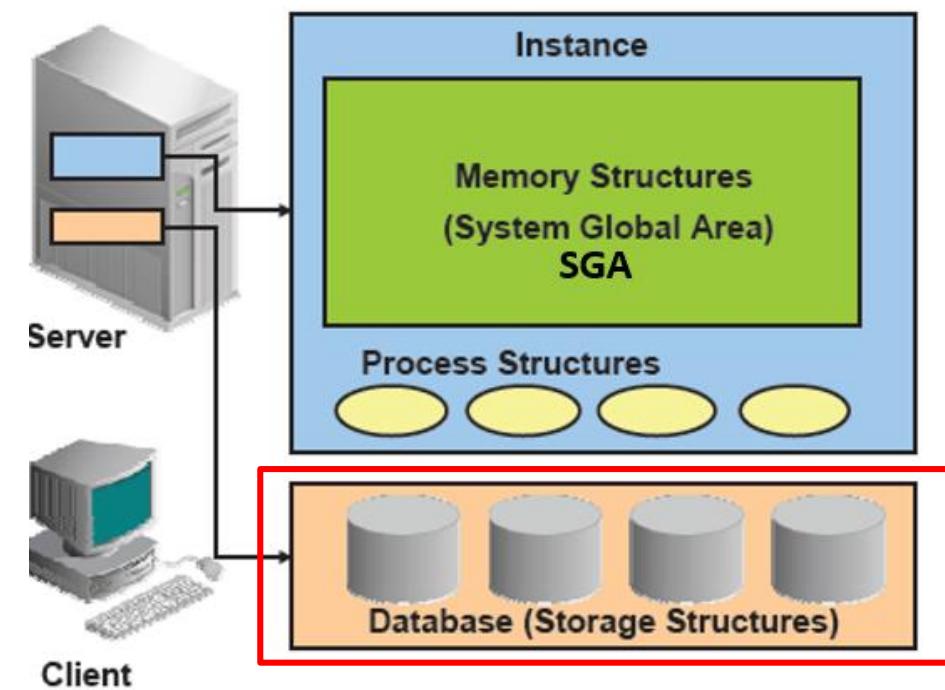
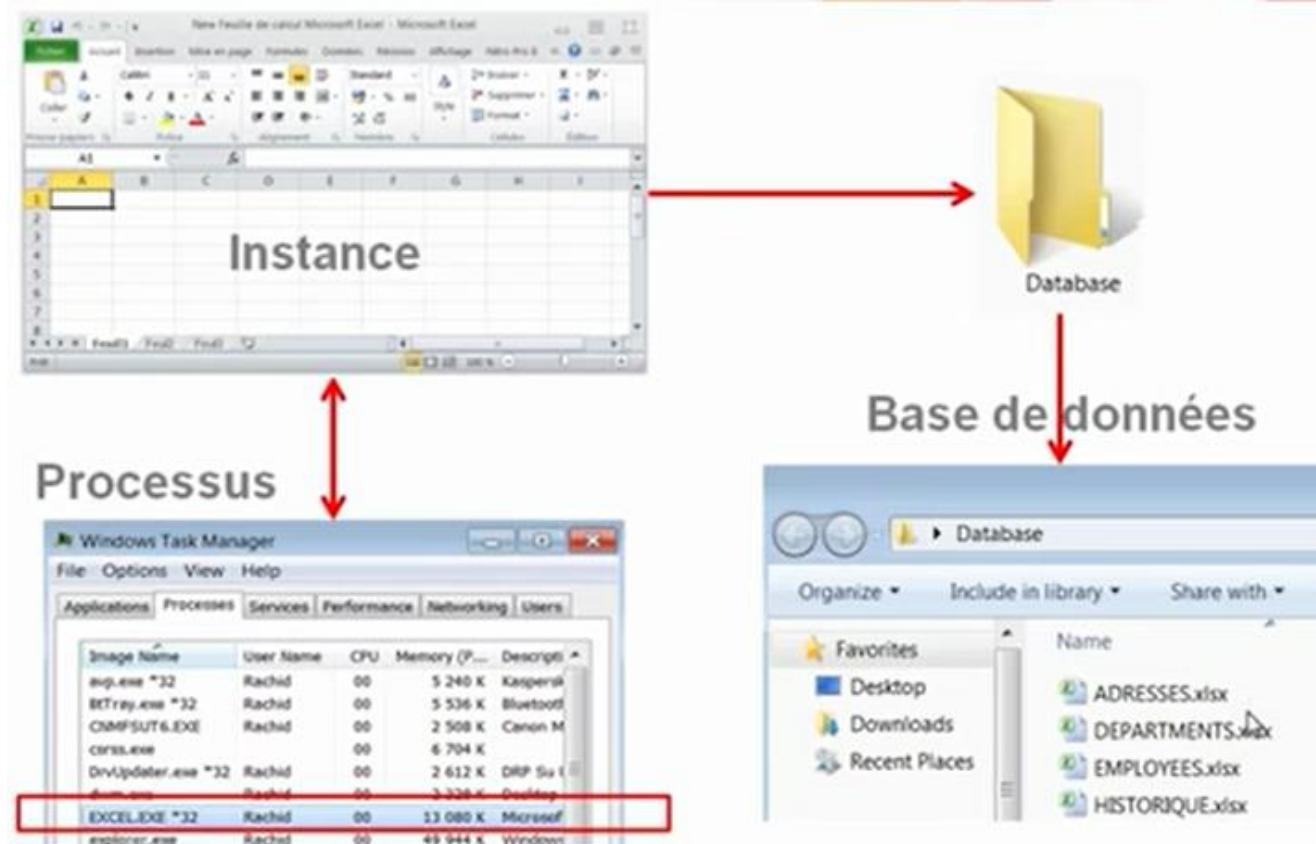
# Instance (2/2)

- Chaque instance de base de données est associée à une seule base.
  - S'il existe plusieurs bases sur le même serveur, il existe une instance distincte pour chacune.
- Une instance de base de données ne peut pas être partagée.
- Mais avec la mise en place de **RAC (Real Application Cluster)**, une base de données peut être ouverte par plusieurs instances qui sont situées dans des nœuds distincts
  - On parle de la haute disponibilité (clustering)



# Stockage de base de données

- Le stockage de la base de données oracle est une collection de fichiers utilisés pour préserver sur un disque physique les données à partir de l'instance oracle



# Plan

---

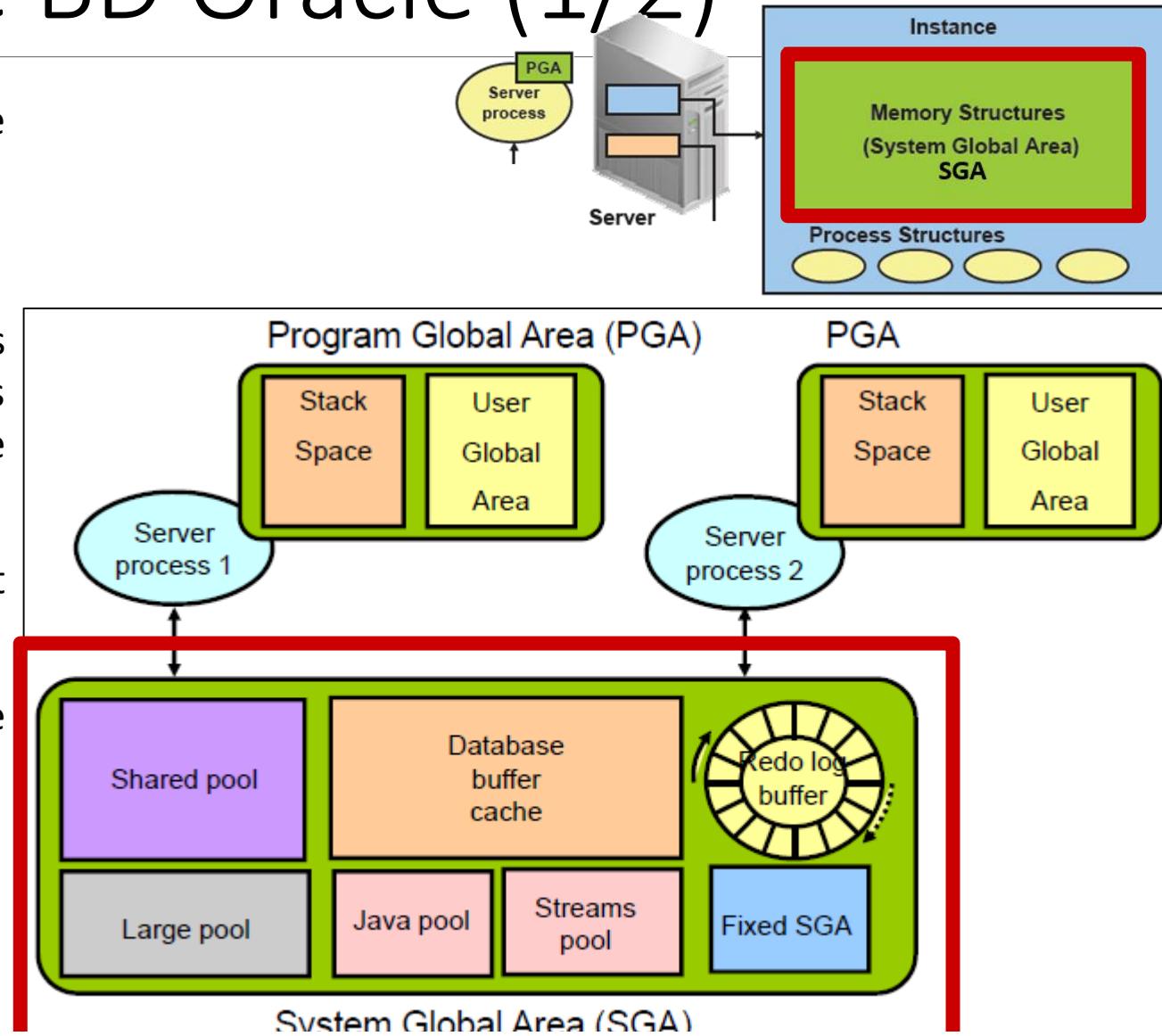
- ❑ Introduction
- ❑ Architecture d'un serveur de BD Oracle
- ❑ **Structure de la mémoire**
  - ❖ Mémoire SGA
  - ❖ Mémoire PGA
- ❑ Structure des processus
- ❑ Structure de stockage de données
- ❑ Interagir avec une base Oracle : Mémoire, processus et stockage

# Structures mémoire BD Oracle (1/2)

Deux **structures mémoire** élémentaires sont associées à une **instance** :

## Mémoire SGA (System Global Area) :

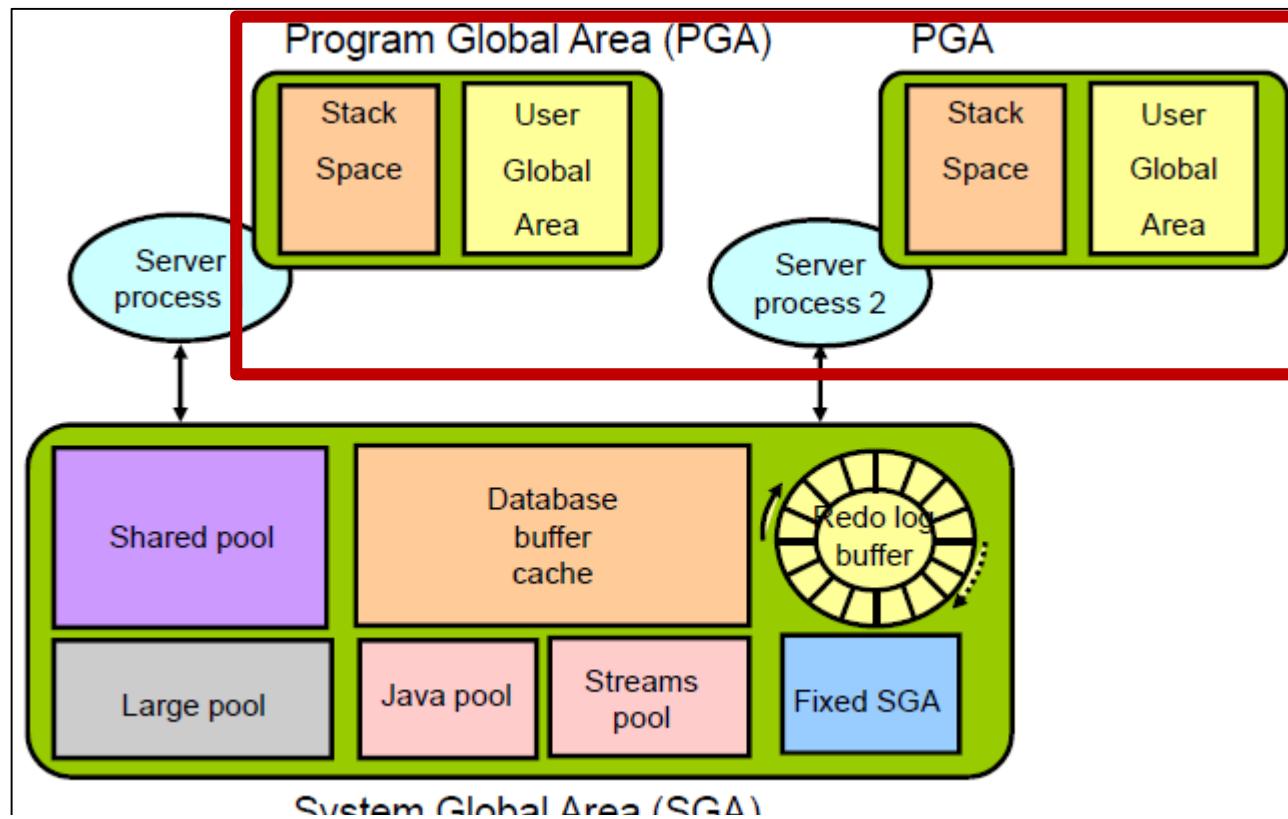
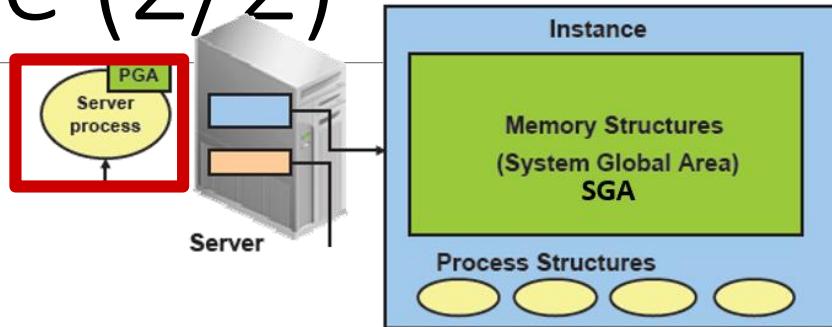
- Groupe de structures **mémoire partagées** (appelées composants SGA) qui contiennent les données et les informations de contrôle correspondant à une instance Oracle Database.
- La mémoire SGA est partagée par les processus serveur et par les processus en arrière-plan.
- Elle contient notamment les blocs de données en mémoire cache et les zones SQL partagées.



# Structures mémoire BD Oracle (2/2)

## Mémoire PGA (Program Global Area) :

- Région de la mémoire qui contient des données et des informations de contrôle pour un **processus serveur**
- Il s'agit d'une **mémoire non partagée** qui est créée par Oracle Database au démarrage d'un processus serveur
- Chaque processus serveur a sa propre mémoire PGA

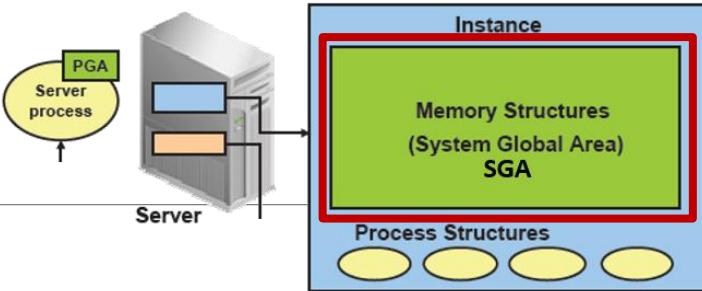


# Plan

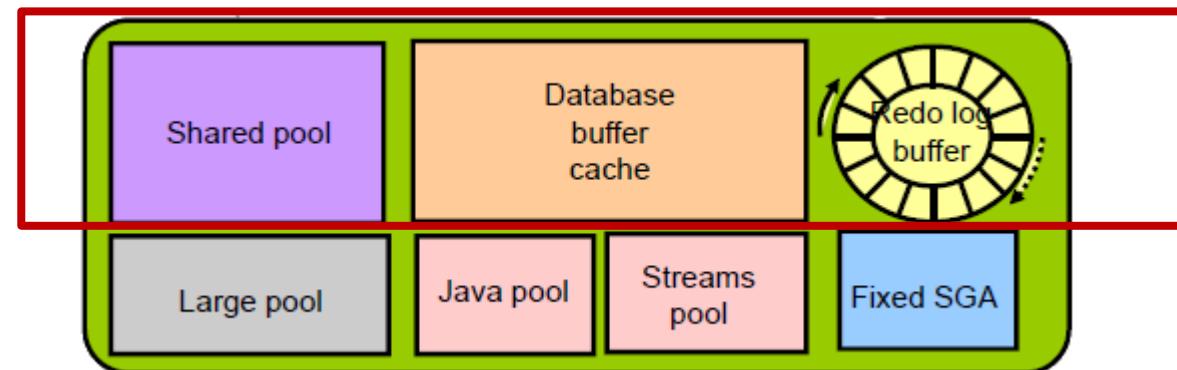
---

- ❑ Introduction
- ❑ Architecture d'un serveur de BD Oracle
- ❑ **Structure de la mémoire**
  - ❖ Mémoire SGA
  - ❖ Mémoire PGA
- ❑ Structure des processus
- ❑ Structure de stockage de données
- ❑ Interagir avec une base Oracle : Mémoire, processus et stockage

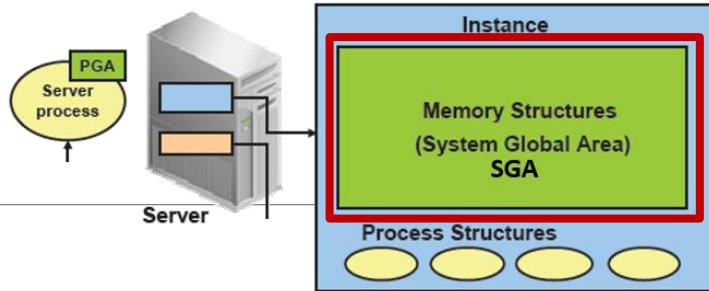
# Structures mémoire SGA (1/2)



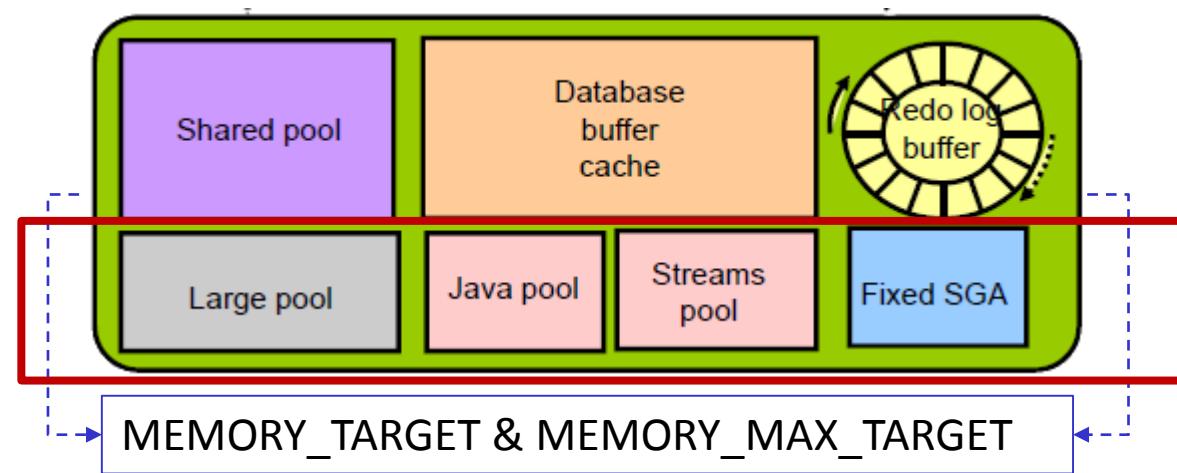
- **Shared pool (zone de mémoire partagée)** : met en mémoire cache diverses structures pouvant être partagées par les utilisateurs
- **Database buffer cache (cache de tampons) de la base de données** : met en mémoire cache les blocs de données extraits de la base de données
- **Redo log buffer (Tampon de journalisation)** : met en mémoire cache les informations de journalisation (utilisées pour la récupération d'instance) jusqu'à ce qu'elles puissent être écrites dans les fichiers de journalisation (fichiers redo log) physiques stockés sur le disque



# Structures mémoire SGA (2/2)



- **Large pool (zone de mémoire large)** : une zone facultative qui fournit des espaces importants d'allocation de mémoire à des processus volumineux, tels que les opérations de sauvegarde et de récupération Oracle et les processus serveur
- **Java pool (zone de mémoire Java)** : utilisée pour l'ensemble du code Java et des données propres à la session, dans la JVM (Java Virtual Machine)
- **Streams pool (zone de mémoire Streams)** : utilisée pour stocker les informations nécessaires aux opérations de capture et d'application des modifications
- **Fixed SGA** : contient des informations générales sur l'état de la base de données, l'instance et la communication entre les processus



Il est possible de modifier la taille de la **mémoire SGA** en définissant les paramètres d'initialisation **MEMORY\_TARGET** et **MEMORY\_MAX\_TARGET**

# Mémoire SGA : Shared pool (1/2)

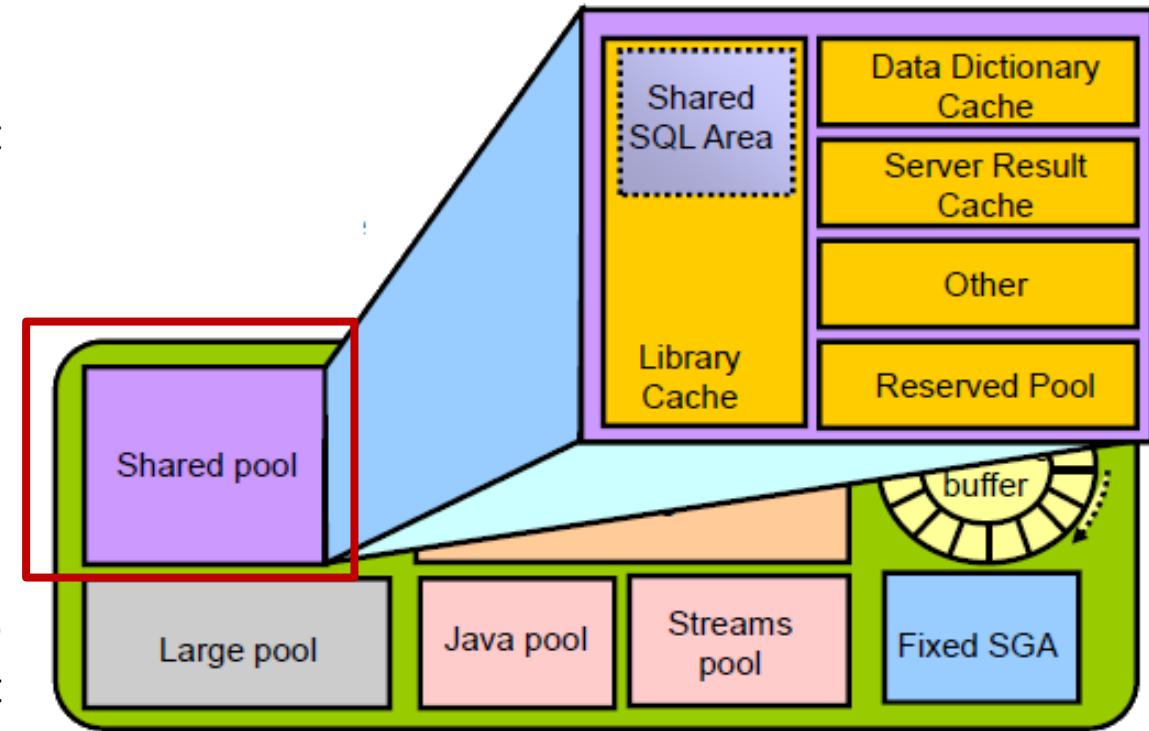
**Shared pool (zone de mémoire partagée)** contient :

- les requêtes SQL le plus récemment exécuté
- l'information du dictionnaire de données la plus récemment utilisée

**Dictionnaire de données**

- un ensemble de tables et de vues de la base de données.
- Il présente des informations de référence concernant la base, ses structures et ses utilisateurs.
- Oracle Database accède fréquemment au dictionnaire de données lors de l'analyse des instructions SQL. Cet accès est essentiel pour un fonctionnement continu.

Le dictionnaire de données est stocké dans « **Library Cache** » & « **Data Dictionary Cache** »



# Mémoire SGA : Shared pool (2/2)

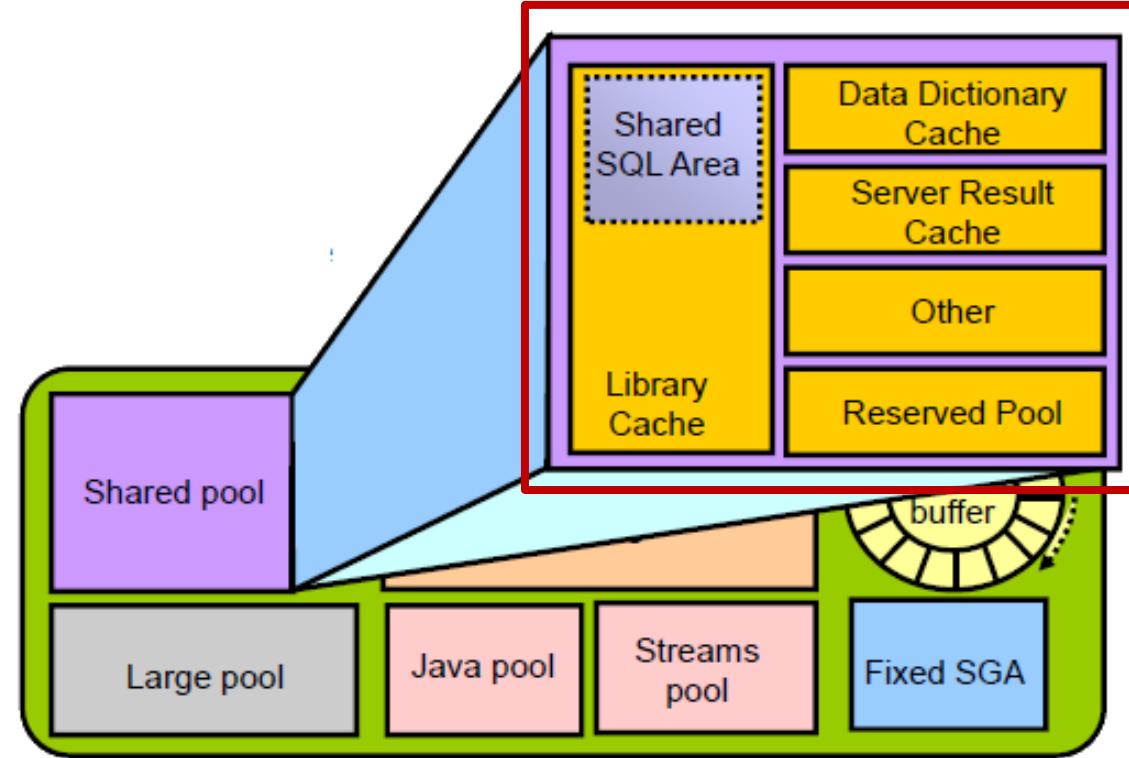
## Library Cache (cache de librairie)

- Contient les informations sur les requêtes SQL récemment utilisées. Le plan d'exécution de ces requêtes ainsi que l'arbre de parcours et le texte des requêtes
- Chaque instruction SQL exécutée est représentée par **Shared SQL Area** (zone SQL partagée). Si la même instruction est exécutée par un autre utilisateur, Oracle réutilise la même zone SQL partagée

## Data Dictionary Cache (cache du dictionnaire de données)

Contient des informations sur :

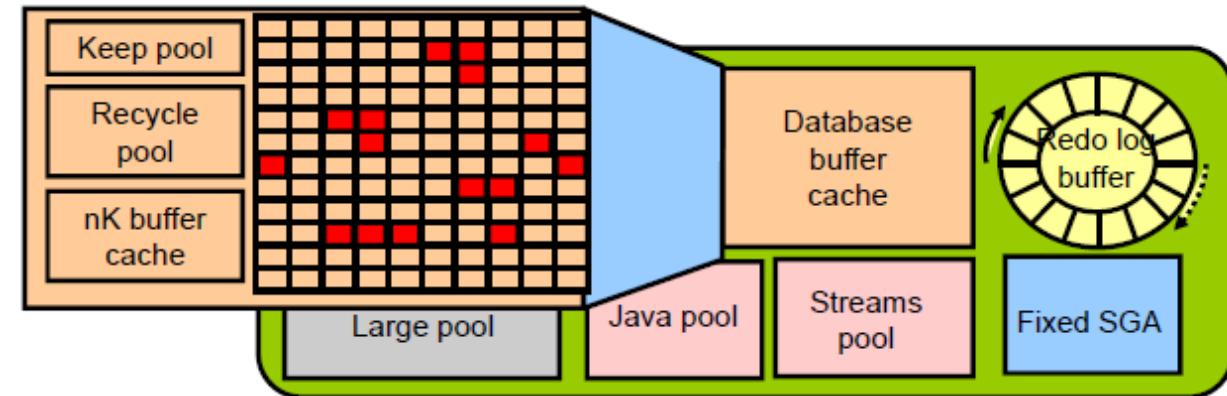
- le dictionnaire de données
- les objets de la base de données et sa structure et la définition des colonnes
- les utilisateurs valides, leur mot de passe et leurs privilèges



# Mémoire SGA : Database Buffer Cache (1/2)

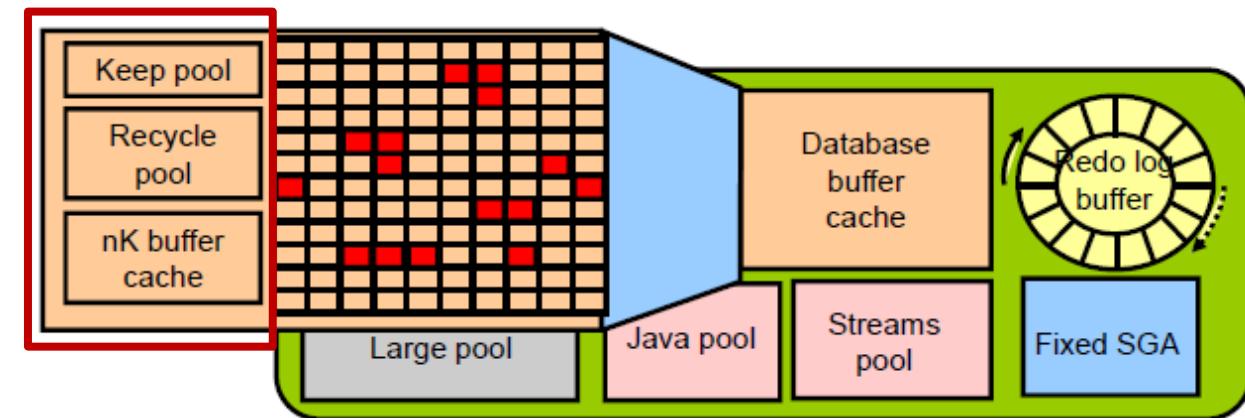
## Database Buffer Cache (Cache de tampons de la base de données) :

- Contient les **images de blocs** lues à partir des fichiers de données ou construites dynamiquement pour satisfaire au modèle de cohérence en lecture. Son accès est partagé par tous les utilisateurs connectés simultanément à l'instance.
- La première fois qu'un processus utilisateur a besoin de données particulières, il va les chercher dans le **Buffer Cache**.
  - S'il les trouve dans le cache (succès en mémoire cache), il peut les lire directement à partir de la mémoire.
  - S'il ne les trouve pas (échec en mémoire cache), il doit les copier depuis un fichier de données sur disque vers un cache tampon afin de pouvoir y accéder.
- L'accès aux données est plus rapide lorsqu'elles se trouvent déjà dans le cache.



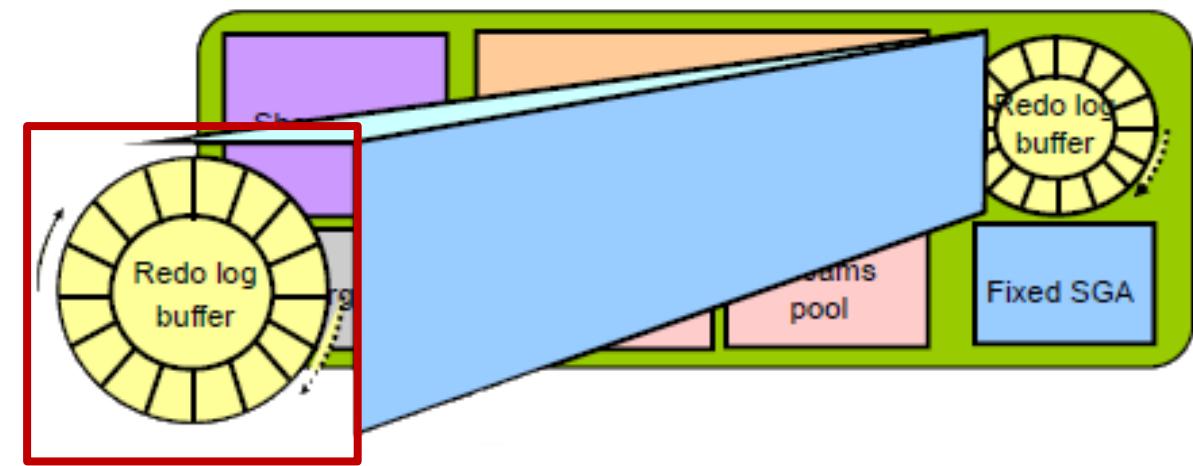
# Mémoire SGA : Database Buffer Cache (2/2)

- **Keep pool (espace de tampons à conserver)** : Type particulier de cache de tampons qui est configuré pour le stockage de blocs de données pendant de longues périodes.
- **Recycle pool (espace de tampons à recycler)** : Type particulier de cache de tampons qui est configuré pour un recyclage ou une suppression rapide de blocs.
- **nK buffer cache (cache de tampons de blocs de taille nK)** : Type particulier de cache de tampons conçu pour le stockage des blocs qui présentent une taille différente de la taille de bloc par défaut de la base de données.



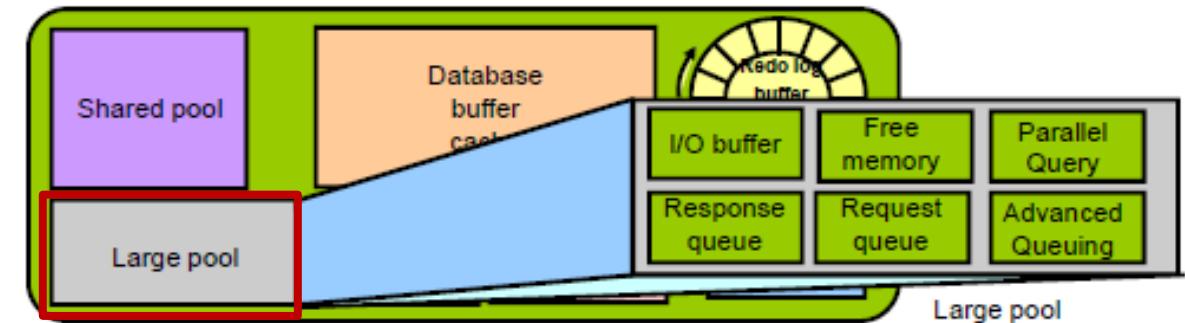
# Mémoire SGA : Redo Log Buffer

- **Redo Log Buffer (tampon de journalisation)** est une mémoire tampon réutilisable contenant des informations sur les modifications effectuées dans la base.
- Les données modifiées dans le buffer cache lors d'une transaction sont stockées dans le redo log buffer pour être récupérées si une erreur est survenue
- Elle contiennent les informations nécessaires à l'**annulation (redo)** des modifications apportées à la base par des opérations :
  - LMD (langage de manipulation de données)
  - LDD (langage de définition de données) ou internes.
    - Elles sont utilisées pour la récupération de la base.
- Lorsque le **processus serveur** effectue des modifications dans le **Buffer Cache**, des entrées de journalisation sont générées et écrites dans le **Redo Log Buffer**.



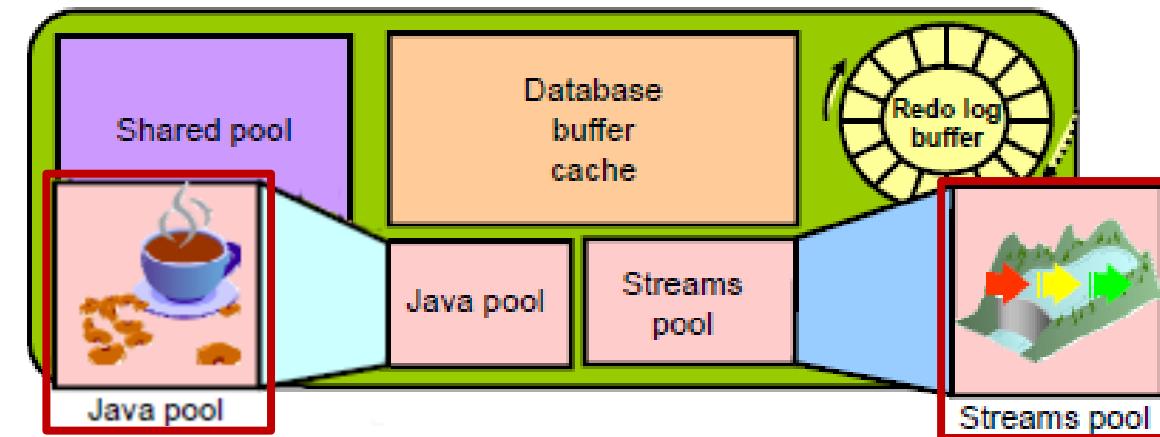
# Mémoire SGA : Large Pool

- **Large Pool (Zone de mémoire LARGE POOL)** fournit des allocations de mémoire de grande taille pour les éléments suivants :
  - Mémoire de session pour le serveur partagé et l'interface Oracle
  - Processus serveur d'E/S
  - Opérations de sauvegarde et de restauration d'Oracle
- En allouant de l'espace mémoire de session à partir de la mémoire LARGE POOL, Oracle peut réserver la zone de mémoire partagée (**Shared pool**) à la mise en cache des interrogations SQL partagées
  - Cela évite la perte de performances causée par la réduction du cache SQL partagé (**Shared SQL Area**).



# Mémoire SGA : Java Pool & Streams Pool

- **Java Pool (zone de mémoire Java)** est utilisée pour l'ensemble du code Java et des données propres à la session dans la JVM (Java Virtual Machine). Elle est utilisée de différentes manières, en fonction du mode d'exécution d'Oracle.
- **Streams Pool (zone de mémoire Streams)** est utilisée par Oracle Streams exclusivement.
  - Elle contient les messages en file d'attente tampon
  - Elle fournit la mémoire nécessaire aux processus **Oracle Streams** de capture et d'application des modifications.
    - Oracle Streams permet de partager l'information (données) entre les base de données
- La taille de la zone de mémoire Streams est initialement égale à zéro (sauf configuration spécifique). Elle augmente dynamiquement en fonction des besoins quand Oracle Streams est utilisé.

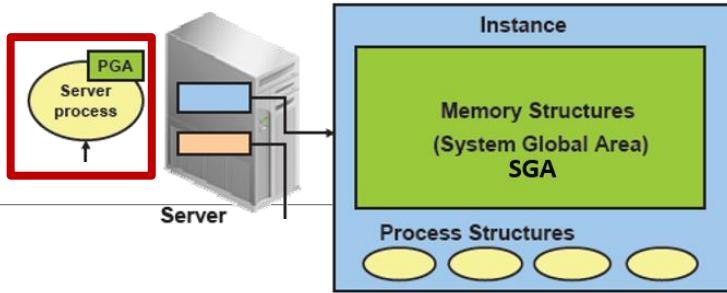


# Plan

---

- ❑ Introduction
- ❑ Architecture d'un serveur de BD Oracle
- ❑ **Structure de la mémoire**
  - ❖ Mémoire SGA
  - ❖ **Mémoire PGA**
- ❑ Structure des processus
- ❑ Structure de stockage de données
- ❑ Interagir avec une base Oracle : Mémoire, processus et stockage

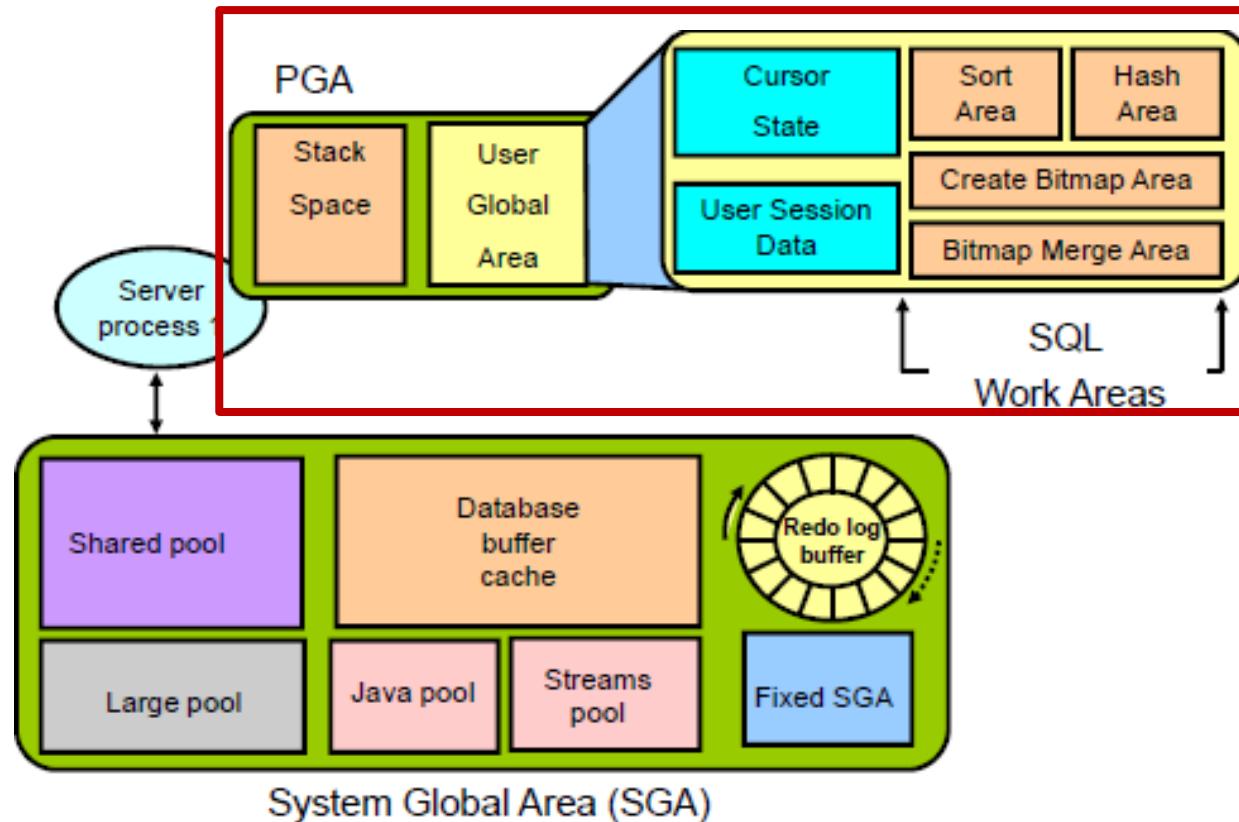
# Structures mémoire PGA (1/3)



- **Program Global Area (mémoire PGA)** est une zone de mémoire contenant des données et des informations de contrôle pour un processus serveur.
- Chaque **processus serveur** dispose de sa propre mémoire PGA privée, qui ne peut être lue que par le code Oracle agissant pour son compte.

Dans un environnement avec serveur dédié :

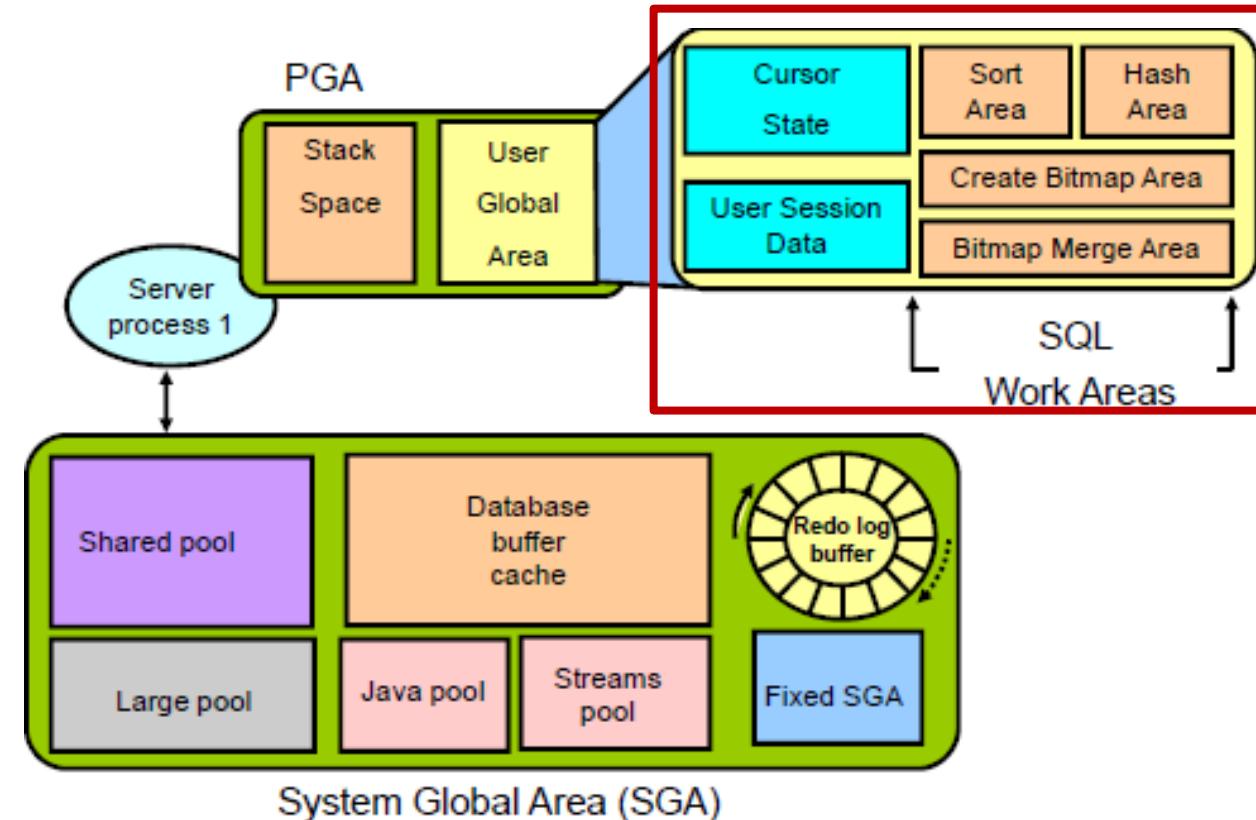
- Chaque utilisateur qui se connecte à l'instance de base de données comporte un processus serveur distinct.
- Chaque PGA contient un **Stack Space (espace de pile)**
- Pour ce type de connexion, la mémoire PGA contient une subdivision de mémoire connue sous le nom de zone **UGA (User Global Area)**.



# Structures mémoire PGA (2/3)

La mémoire UGA comprend les éléments suivants :

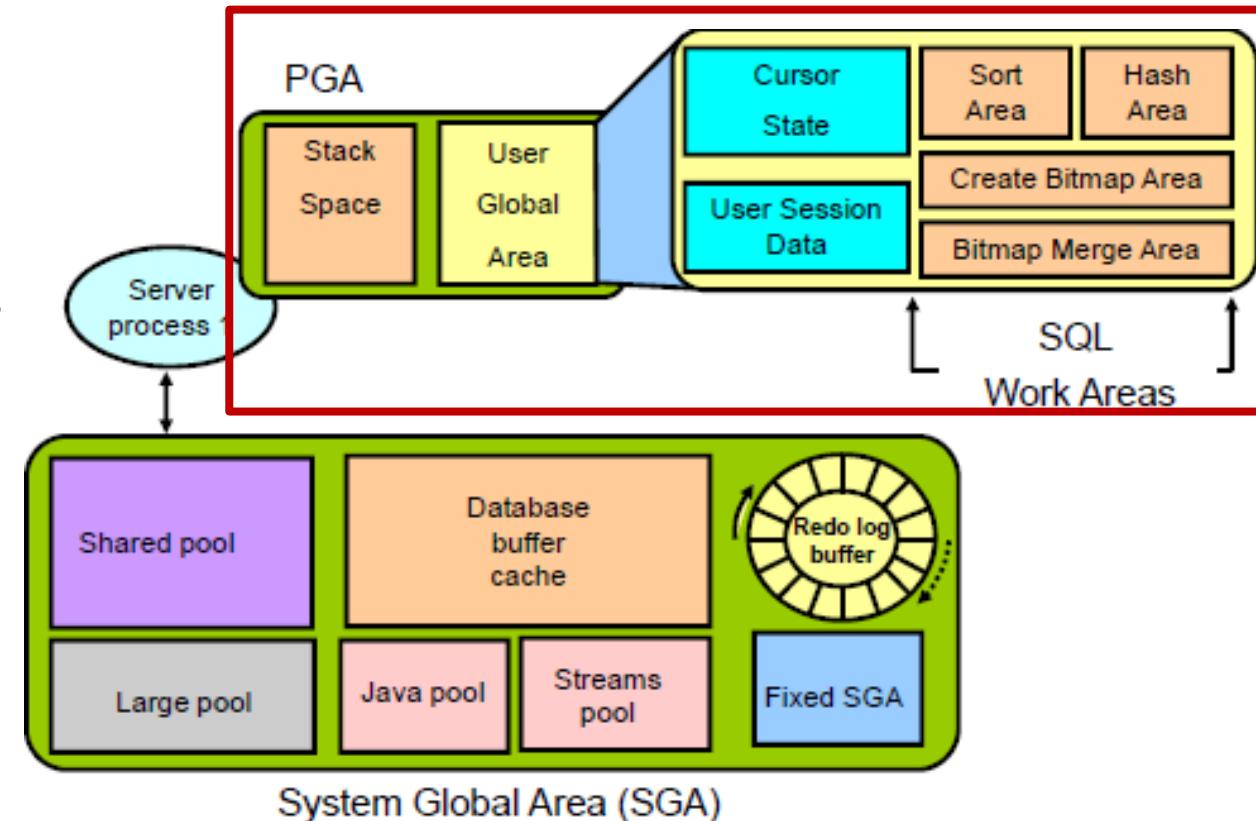
- **Cursor State (zone de curseurs)** pour stocker les informations d'exécution sur les curseurs.
- **User Session Data (zone de mémoire de session utilisateur)** pour des informations concernant une session.
- **SQL Work Areas (zones de travail SQL)** pour le traitement des instructions SQL :
  - **Sort Area (zone de tri)** pour les fonctions qui classent les données telles que ORDER BY et GROUP BY.
  - **Hash Area (zone de hachage)** pour exécuter les jointures de hachage entre les tables.
  - **Create Bitmap Area (zone de création d'index bitmap)** commune aux data warehouses utilisée pour la création d'index bitmap.
  - **Bitmap Merge Area (zone de fusion d'index bitmap)** utilisée pour définir les plans d'exécution basés sur un index bitmap.



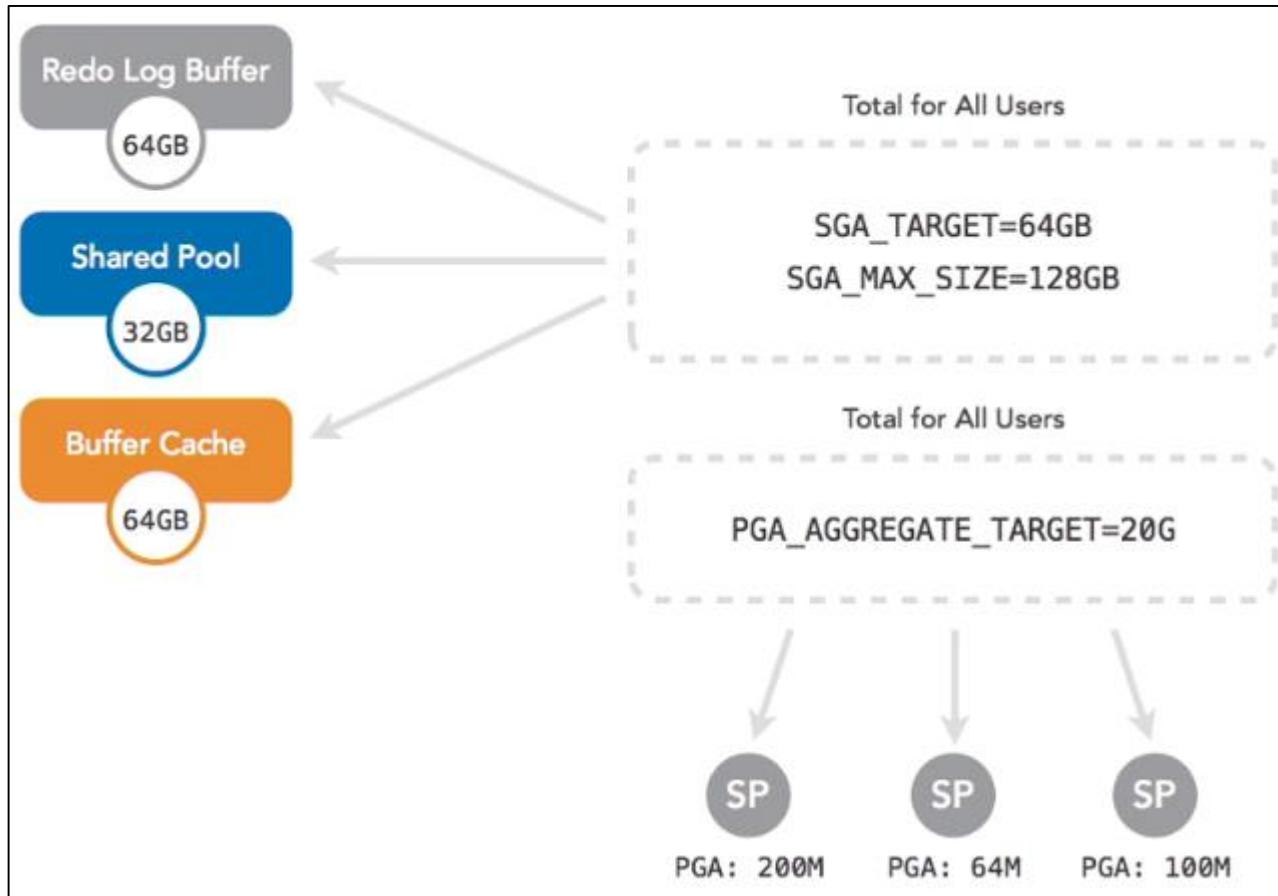
# Structures mémoire PGA (3/3)

Dans un environnement avec serveur partagé :

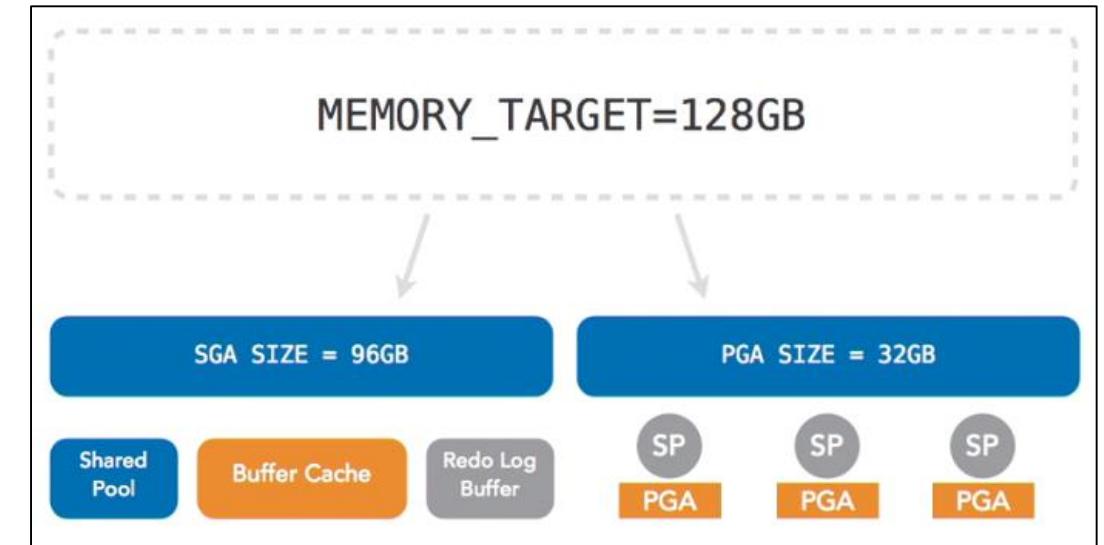
- Plusieurs clients partagent un même processus serveur
- Dans ce modèle, la **mémoire UGA** est transférée dans la mémoire SGA (dans la **Shared pool** ou dans la **LARGE POOL** si elle est configurée)
  - de sorte que la mémoire PGA contient uniquement **l'espace de pile (Stack Space)**



# Définir la taille de la mémoire



Allouer de la mémoire manuellement



Allouer de la mémoire dynamiquement

SP : Server Process

# Plan

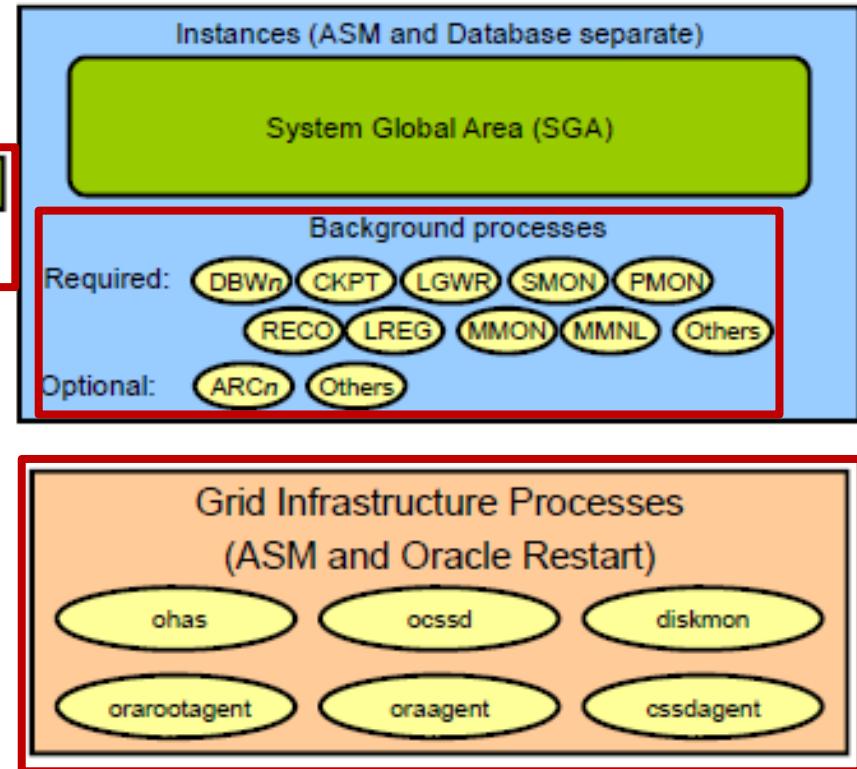
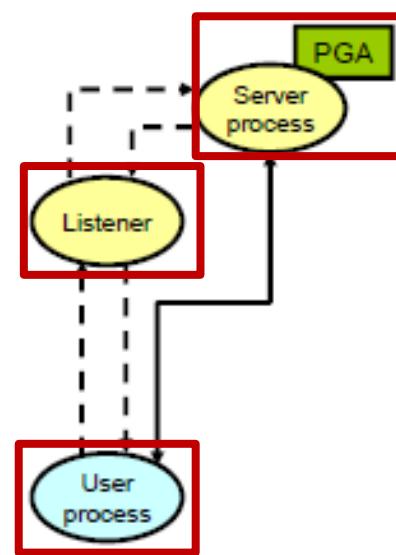
---

- ❑ Introduction
- ❑ Architecture d'un serveur de BD Oracle
- ❑ Structure de la mémoire
- ❑ **Structure des processus**
  - ❖ Listener process (processus d'écoute)
  - ❖ Background processes (processus d'arrière plan)
- ❑ Structure de stockage de données
- ❑ Interagir avec une base Oracle : Mémoire, processus et stockage

# Architecture des processus

Les processus d'un système Oracle Database peuvent être classés en trois grandes catégories :

- **User Process (processus utilisateur)** : Application ou outil qui se connecte à la base de données Oracle
- Processus de base de données
  - **Server Processes (processus serveur)** : Se connecte à l'instance Oracle et démarre lorsqu'un utilisateur ouvre une session
  - **Background Processes (processus en arrière-plan)** : Démarrant en même temps qu'une instance Oracle
- Démon / processus applicatifs
  - **Listener Processes (processus d'écoute réseau)**
  - **Grid Infrastructure Processes (processus de l'infrastructure de grille)**

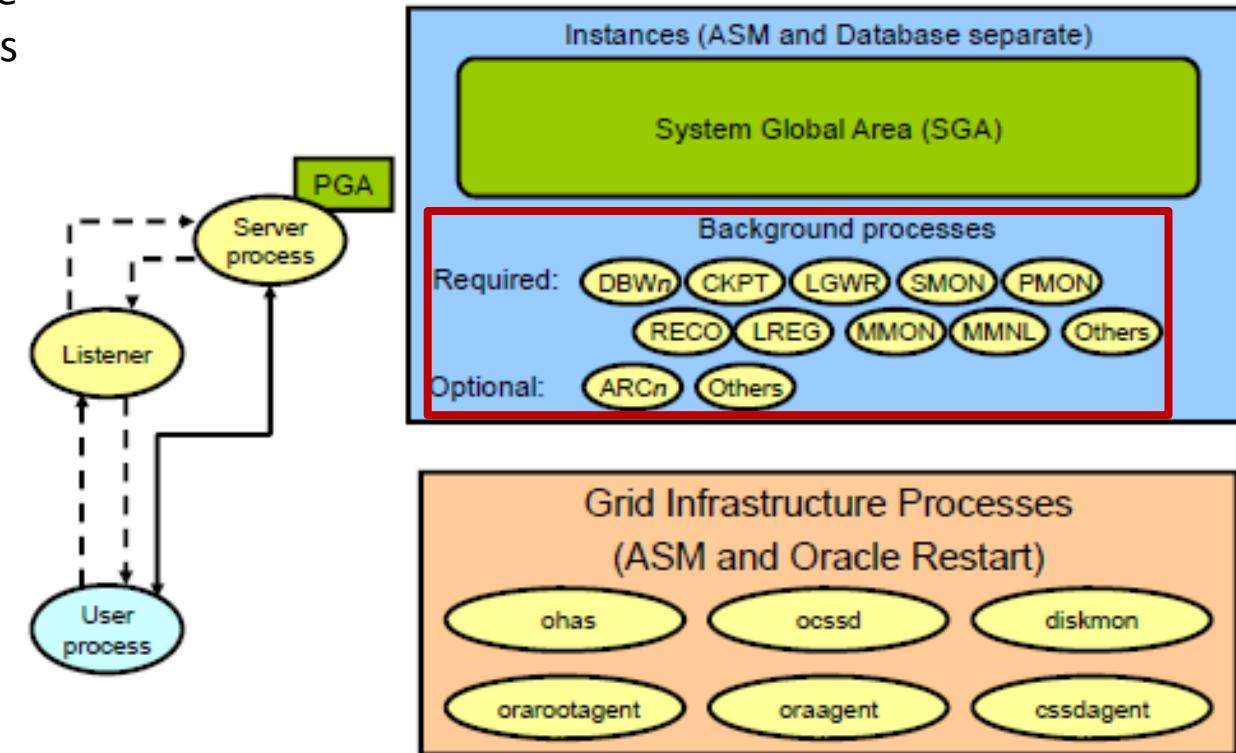


**Unix** : chaque processus s'exécute individuellement

**Windows**: Oracle.exe = Master Process + Les Processus  
Oracle s'exécute comme des fils (Threads)

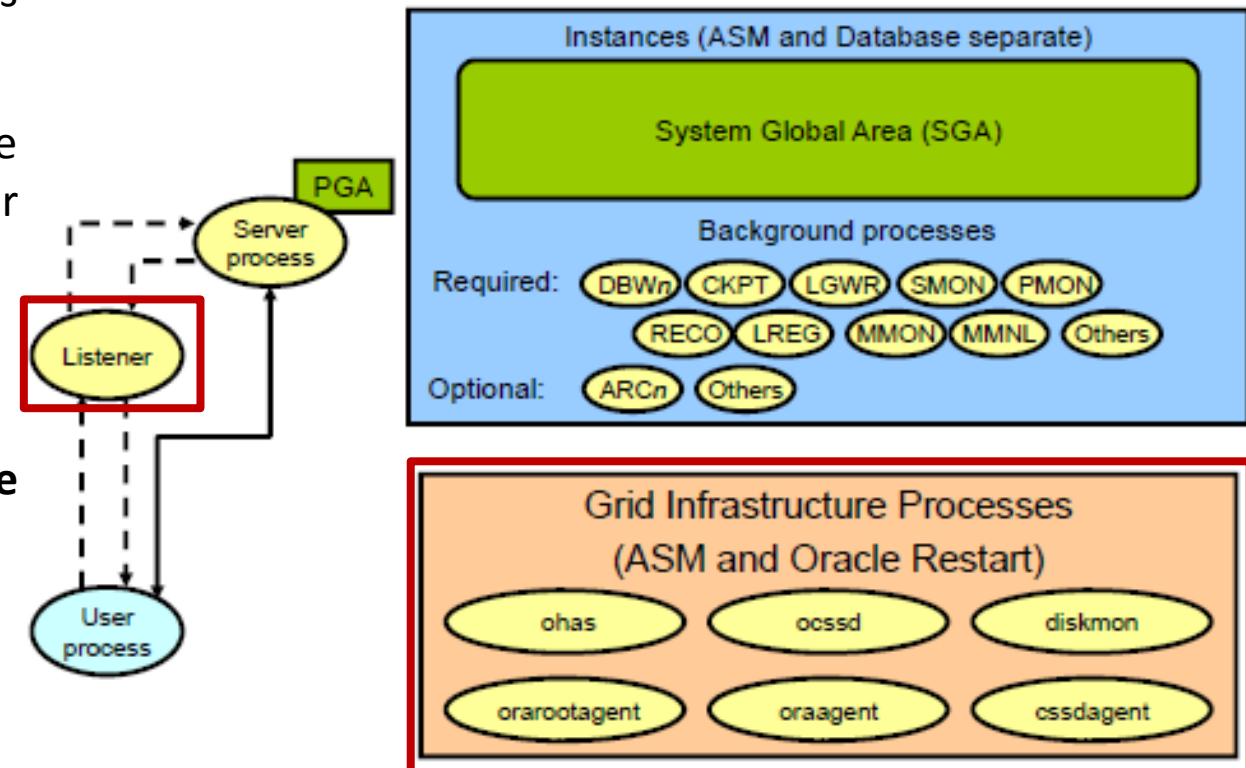
# Structure des processus (1/2)

- Dans les environnements qui n'utilisent ni RAC (Real Application Clusters), ni ASM (Automatic Storage Management), les processus en arrière-plan les plus courants sont les suivants :
  - DBWn (Database Writer Process)
  - LGWR (Log Writer)
  - CKPT (Checkpoint process) : processus de point de reprise
  - SMON (System Monitor)
  - PMON (Process Monitor)
  - RECO (Recoverer process) : processus de récupération
  - LREG (Listener registration process)
  - MMON (Manageability monitor process)
  - MMNL (Manageability monitor lite process)
  - CJQ0 (coordonnateur de file d'attente de travaux)
  - Jnnn (processus esclaves de travail)
  - ARCh (Archiver processes) : processus d'archivage
  - QMNN (processus de surveillance de file d'attente)



# Structure des processus (2/2)

- Certains processus en arrière-plan sont créés automatiquement lorsqu'une instance est démarrée, tandis que d'autres sont lancés en cas de besoin.
- D'autres structures de processus ne sont pas propres à une base de données unique, mais peuvent être partagées par plusieurs bases hébergées sur le même serveur.
- Il s'agit notamment des :
  - **Listener Processes (processus d'écoute réseau)**
  - **Grid Infrastructure Processes (processus de l'infrastructure de grille)**



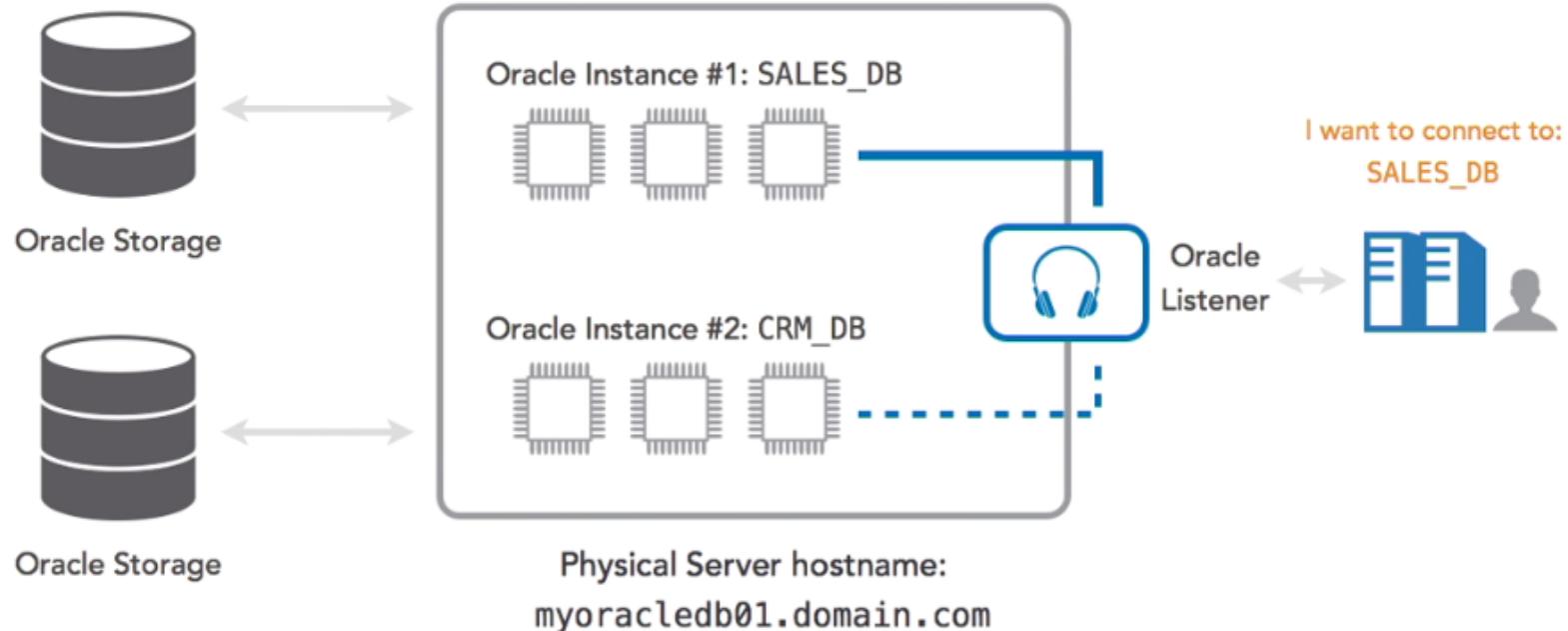
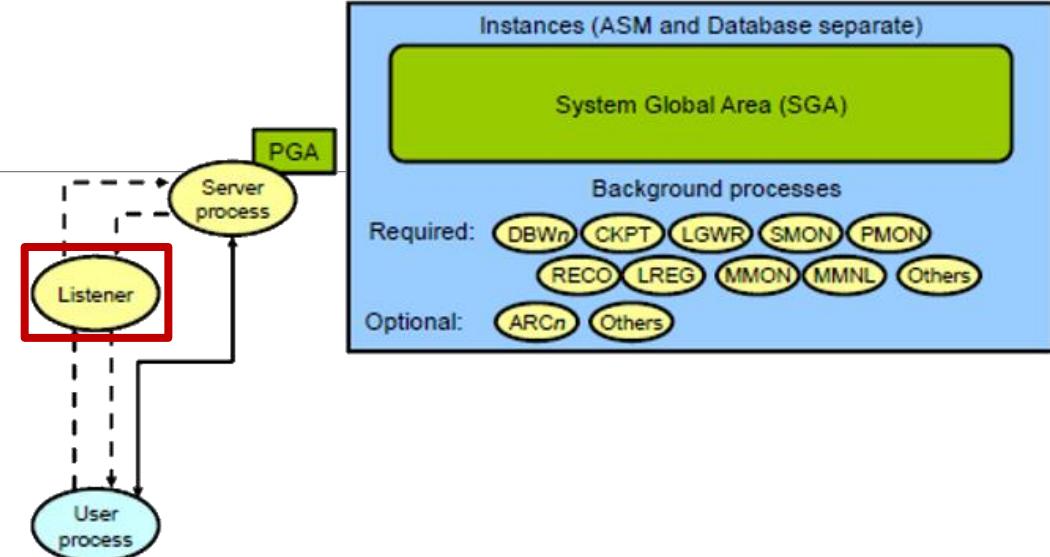
# Plan

---

- ❑ Introduction
- ❑ Architecture d'un serveur de BD Oracle
- ❑ Structure de la mémoire
- ❑ **Structure des processus**
  - ❖ **Listner process (processus d'écoute)**
  - ❖ Background processes (processus d'arrière plan)
- ❑ Structure de stockage de données
- ❑ Interagir avec une base Oracle : Mémoire, processus et stockage

# Listener Processes

- Listener Processes (processus d'écoute réseau) représente le pont (gateway) pour accéder à la base de données oracle
- Il a pour tâche de coordonner les connexions entre la base de données et l'application



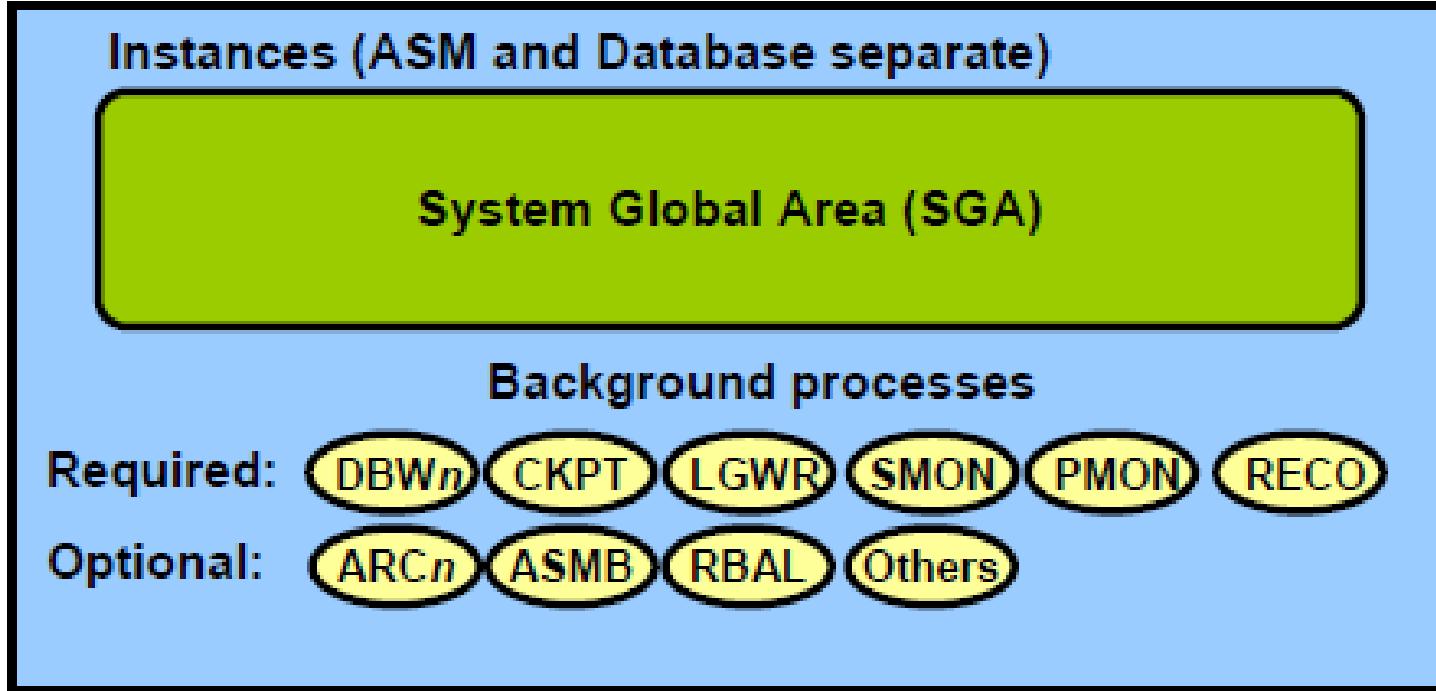
# Plan

---

- ❑ Introduction
- ❑ Architecture d'un serveur de BD Oracle
- ❑ Structure de la mémoire
- ❑ **Structure des processus**
  - ❖ Listener process (processus d'écoute)
  - ❖ **Background processes (processus d'arrière plan)**
- ❑ Structure de stockage de données
- ❑ Interagir avec une base Oracle : Mémoire, processus et stockage

# Processus d'arrière plan

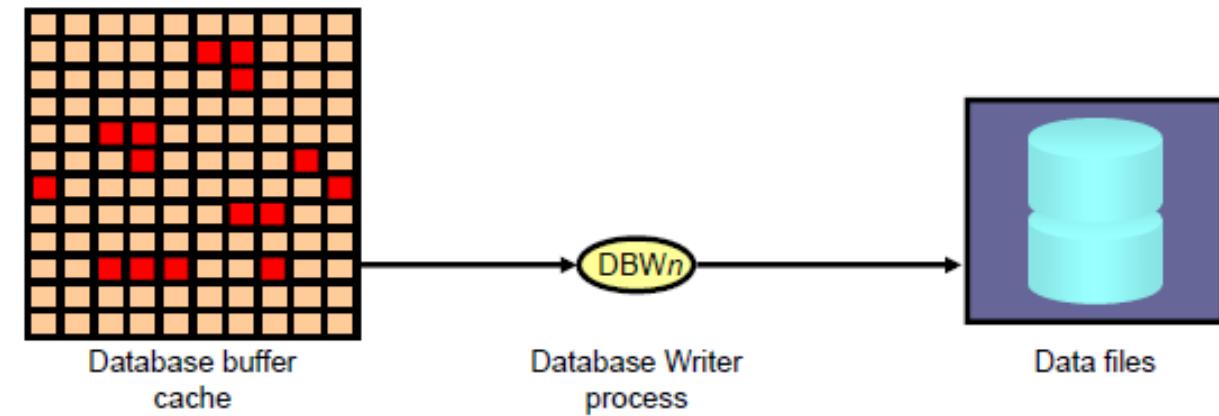
---



# Processus DBWn (Database Writer)

Le **processus DBWn** écrit sur le **disque** les tampons qui ont été modifiés ("dirty") dans le **DataBase Buffer Cache** (cache de tampons de la base de données) :

- de manière asynchrone pendant l'exécution d'autres traitements
- pour faire avancer le point de reprise



Les blocks « **dirty** » sont les blocks qui n'ont pas encore été enregistré sur le disque

# Processus DBWn (Database Writer)

---

Le processus DBWn écrit les tampons "dirty" sur disque dans les cas suivants :

- Quand un **processus serveur** ne trouve pas de tampon réutilisable "clean" après avoir passé en revue un nombre seuil de tampons, il demande au processus DBWn de transférer des tampons sur disque. DBWn écrit les tampons "dirty" de manière asynchrone tout en effectuant d'autres traitements.
  - Si le Buffer cache est plein, le DBWn enregistre les données sur le disque (Data files)
- DBWn transfère périodiquement sur disque les données des tampons afin de faire avancer le **point de reprise**, qui détermine le point de départ d'une récupération d'instance dans le fichier de journalisation (redo log). Ce point correspond au plus ancien tampon "dirty" du cache de tampons.
- Un seul processus Database Writer (DBW0) est suffisant pour la plupart des systèmes, mais il est possible de configurer d'autres (DBW1 à DBW9 et DBWa à DBWz) pour optimiser les performances d'écriture dans un système qui modifie les données de manière intensive. Ces processus **DBWn supplémentaires ne sont pas nécessaires** sur les systèmes dotés **d'un seul processeur**.
- Le **nombre maximal de processus DBWn est 36**. S'il n'est pas précisé par l'utilisateur au démarrage, Oracle Database détermine sa valeur en **fonction du nombre de CPU** et de groupes de processeurs.

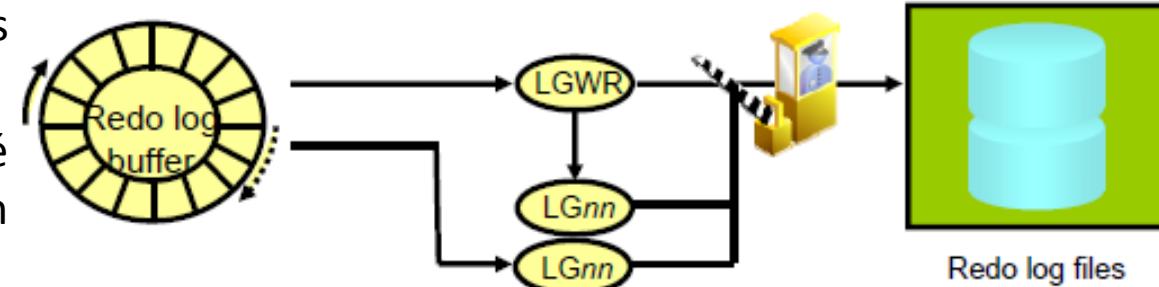
# Processus LGWR (Log Writer Process)

Le **processus Log Writer (LGWR)** est chargé de gérer la mémoire tampon de journalisation (**redo log buffer**) en écrivant ses entrées dans un fichier de journalisation sur le disque (**redo log files**).

- Il écrit toutes les entrées de journalisation qui ont été copiées dans le tampon depuis la dernière opération d'écriture.
- Le tampon de journalisation est réutilisable. Une fois que le processus LGWR a écrit son contenu dans un fichier de journalisation sur le disque, il peut recevoir de nouvelles données provenant des processus serveur.

## LGWR effectue une opération d'écriture :

1. Quand un processus utilisateur valide une transaction.
2. Quand un tiers du tampon de journalisation est plein.
3. Avant qu'un processus DBWn écrive des tampons modifiés sur le disque (si nécessaire).
4. Toutes les 3 secondes.



- **LGWR** gère les opérations qui sont très rapides ou doivent être coordonnées et délègue des opérations à **LGnn**
- **LGWR** coordonne les processus **LGnn** et assure l'ordre correct pour toutes les opérations

# Processus CKPT (Checkpoint)

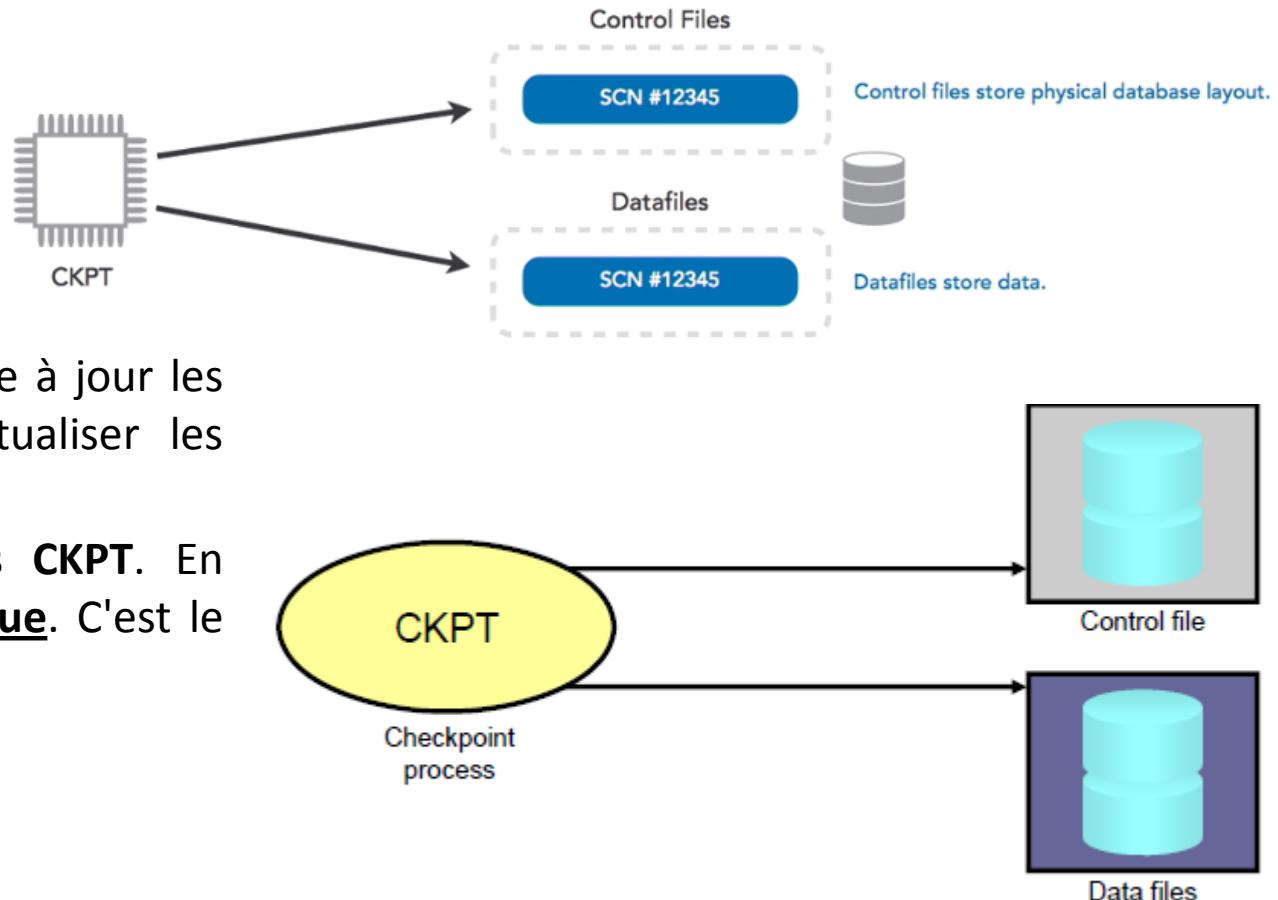
- **Un point de reprise (checkpoint)** est une structure de données qui définit un numéro **SCN (System Change Number)** dans le thread de journalisation d'une base.

- Les points de reprise sont enregistrés dans :

- le fichier de contrôle
- dans chaque en-tête de fichier de données

- Quand un point de reprise est créé, Oracle doit mettre à jour les en-têtes de tous les fichiers de données pour actualiser les informations correspondantes.

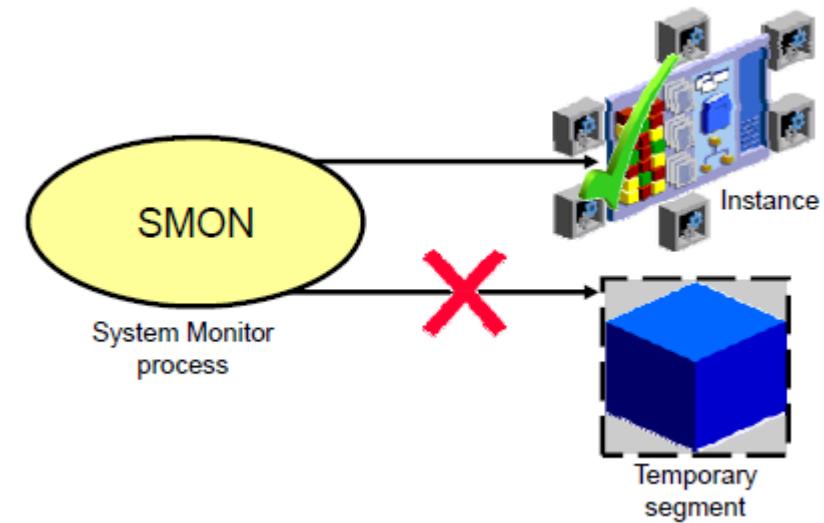
- Cette opération est effectuée par le **processus CKPT**. En revanche, celui-ci n'écrit pas les blocs sur le disque. C'est le rôle du processus **DBWn**.



# Processus SMON (System Monitor)

## Processus SMON (System Monitor) :

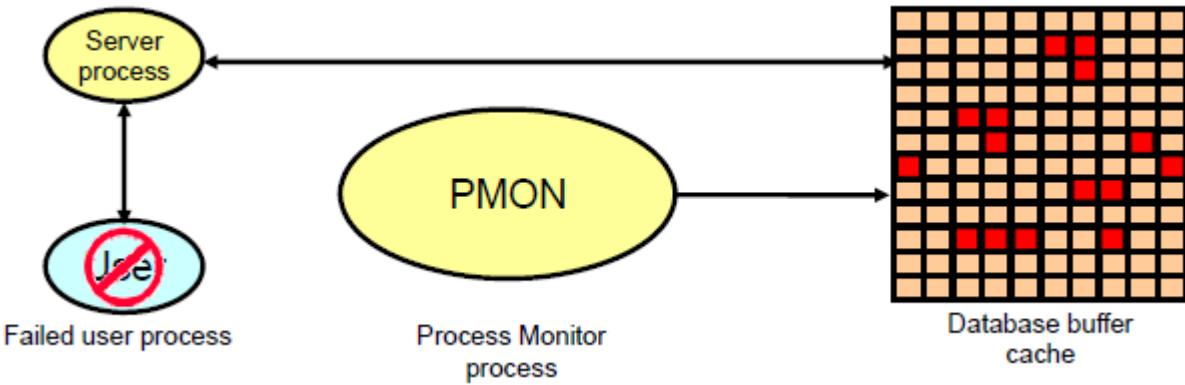
- effectue les opérations de récupération des données au démarrage de l'instance
- nettoie les segments temporaires inutilisés
- Si des transactions terminées ont été ignorées pendant la récupération d'instance à cause d'erreurs liées aux fichiers ou d'erreurs de mise hors ligne (offline), le processus SMON les récupère quand le tablespace ou le fichier correspondant est remis en ligne (online).



# Processus PMON (Process Monitor)

## Processus PMON (Process Monitor) :

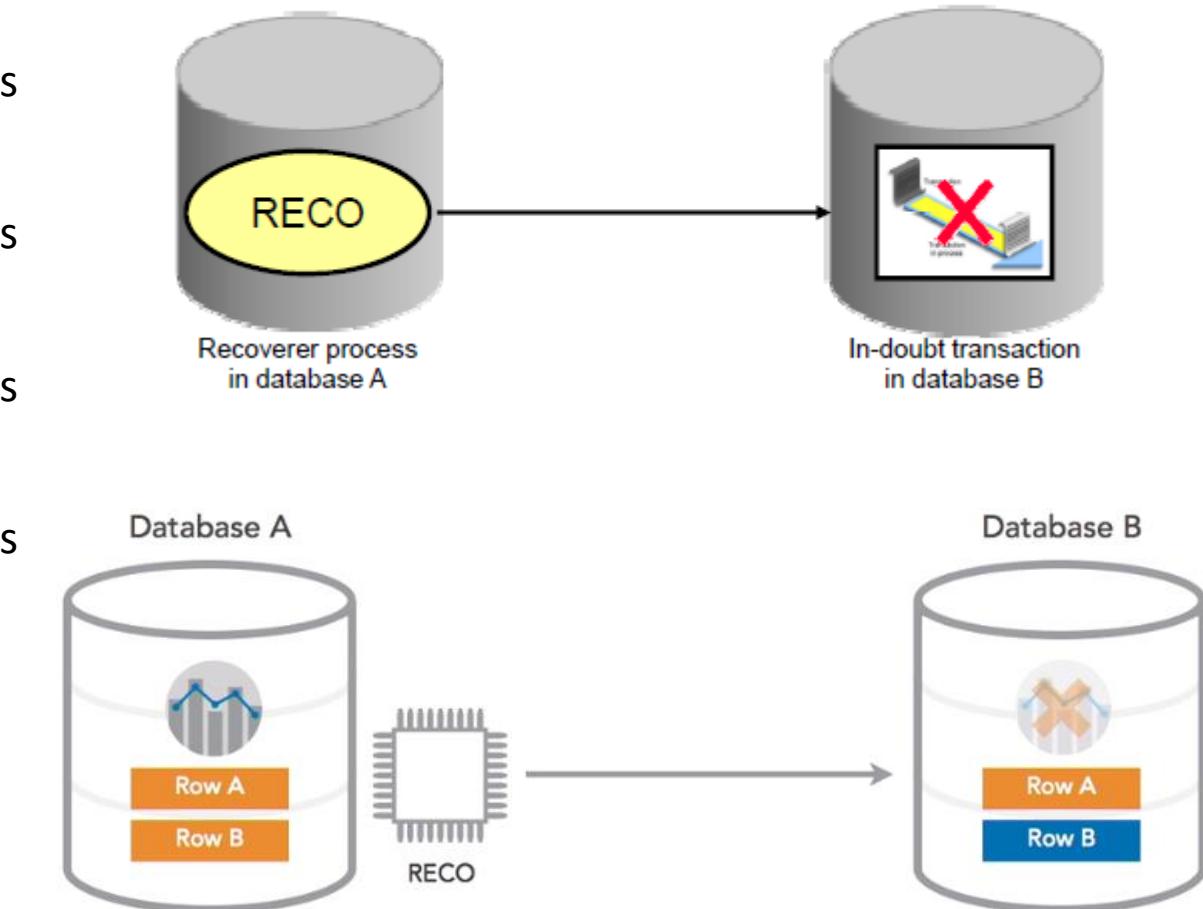
- Il assure la récupération des processus utilisateur qui ont échoué.
  - Nettoyage du cache de tampons de la base de données
  - Libération des ressources utilisées par le processus utilisateur en échec
- Il surveille les sessions pour détecter tout dépassement du délai d'inactivité.



# Processus RECO (Recoverer)

## Processus RECO (Recoverer) :

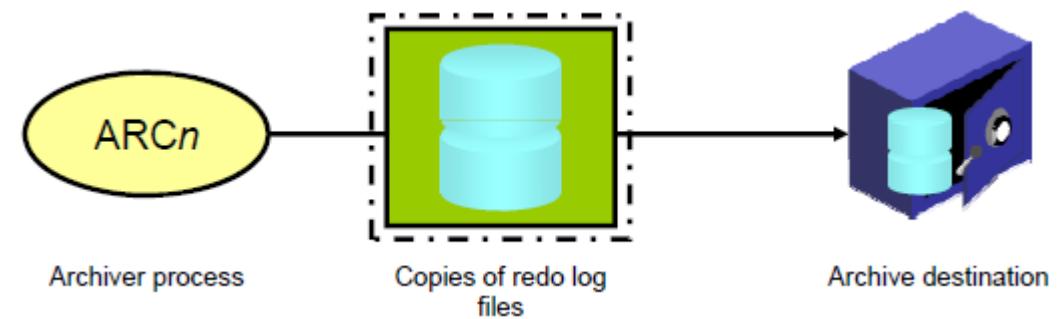
- Intervient dans une configuration de base de données distribuée.
- Se connecte automatiquement aux autres bases impliquées dans des transactions distribuées équivoques.
- Résout automatiquement toutes les transactions équivoques.
- Supprime toutes les lignes correspondant à des transactions équivoques.



# Processus ARCn (Archiver)

## Processus ARCn (Archiver) :

- Copie les fichiers de journalisation (redo log files) sur le périphérique de stockage désigné après un changement de fichier de journalisation.
- Peut collecter des données de journalisation sur les transactions et les transmettre à des destinations de secours.

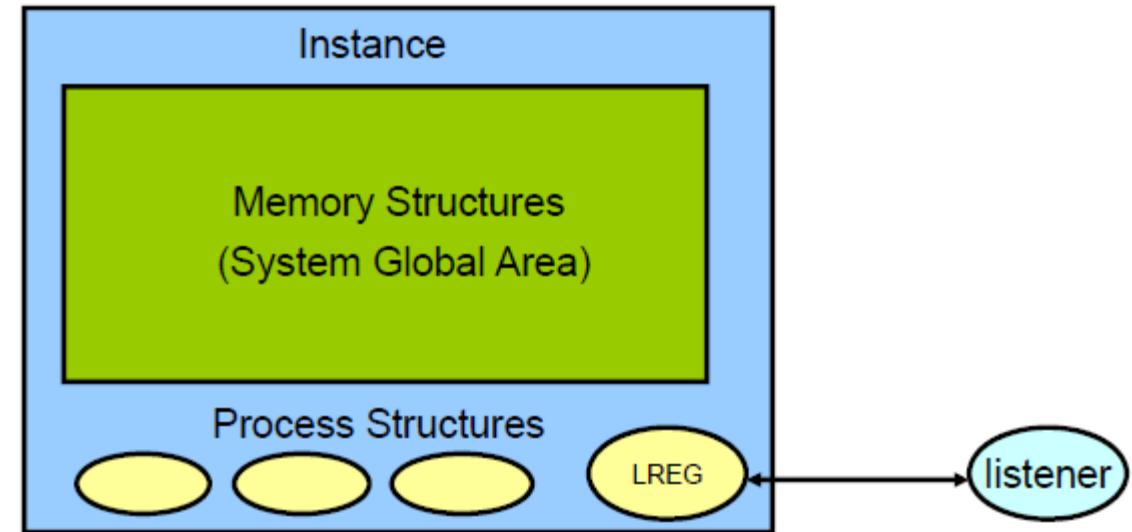


# Processus LREG

---

## Processus LREG (Listener Registration Process) :

- Enregistre les informations sur l'instance de la base de données et dispatche les processus avec le listener



# Plan

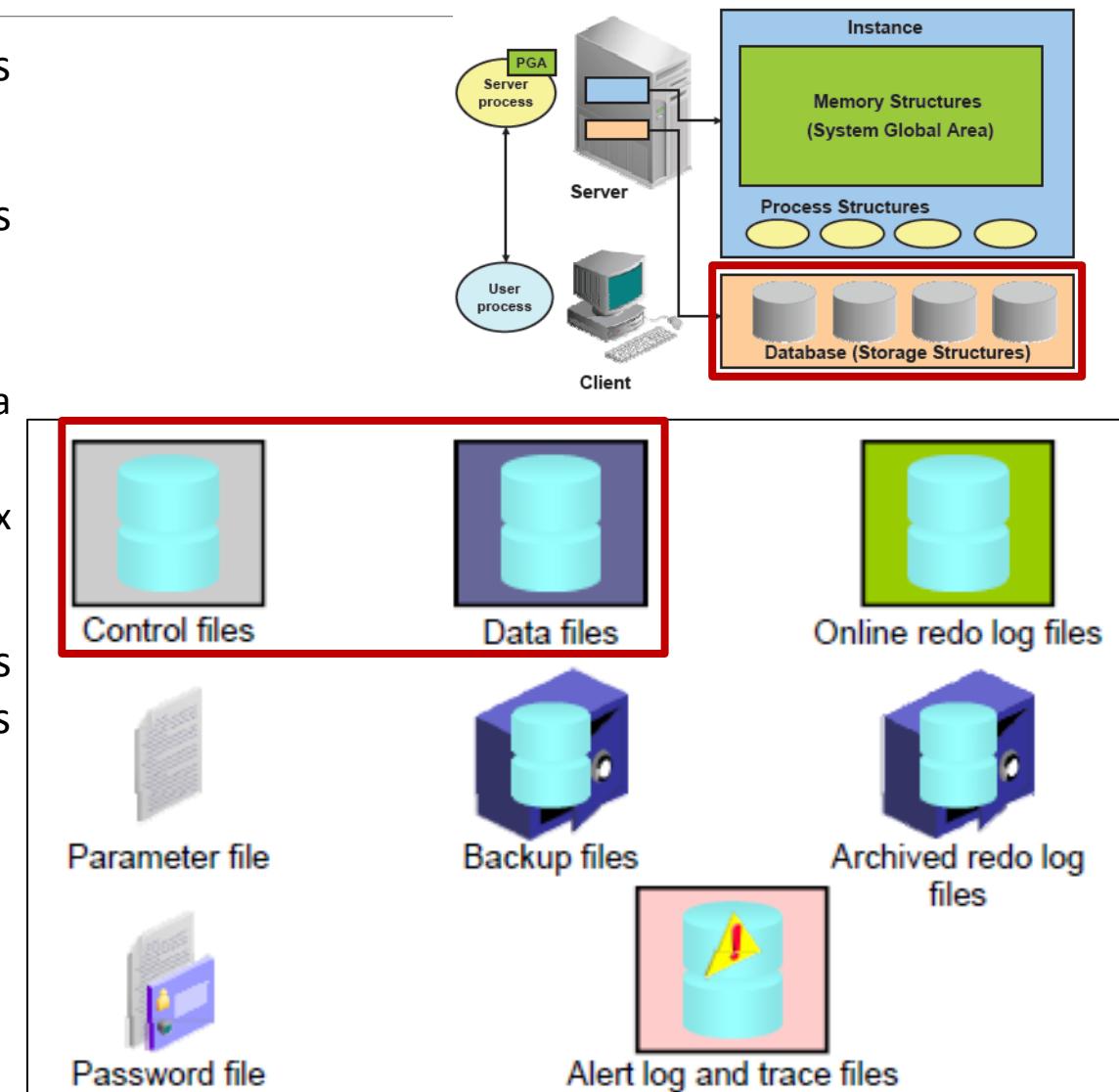
---

- ❑ Introduction
- ❑ Architecture d'un serveur de BD Oracle
- ❑ Structure de la mémoire
- ❑ Structure des processus
- ❑ **Structure de stockage de données**
  - ❖ **Architecture de stockage de la BD**
  - ❖ Structures logiques et physiques de la BD
- ❑ Interagir avec une base Oracle : Mémoire, processus et stockage

# Architecture de stockage de la base de données (1/3)

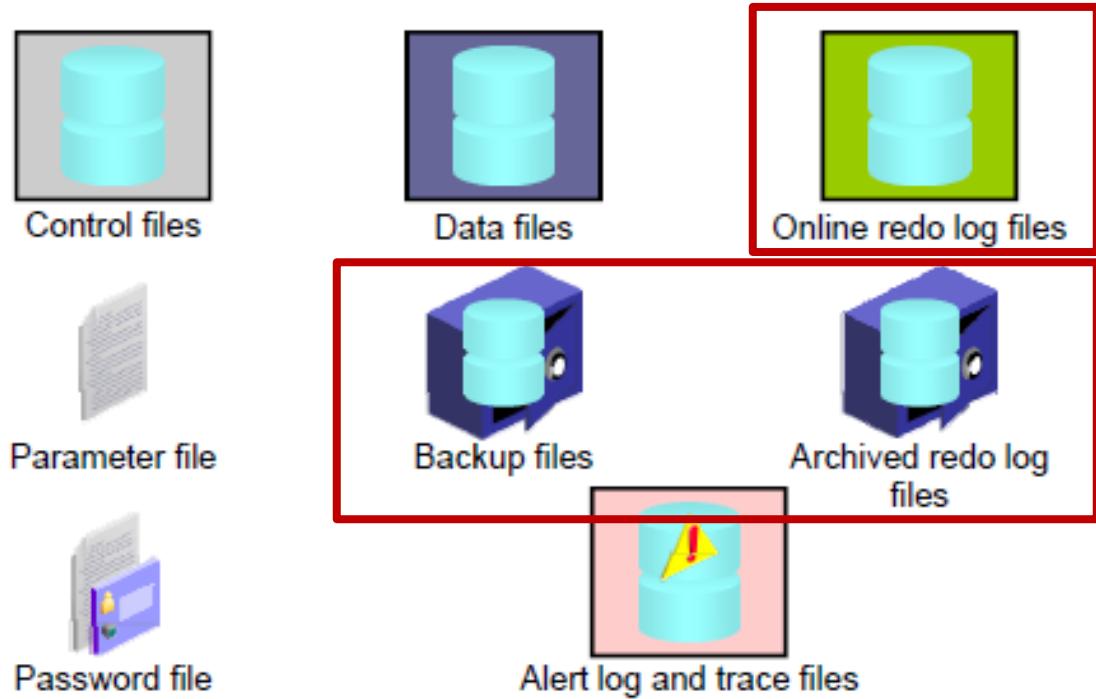
Les fichiers constituant une base de données Oracle sont organisés de la façon suivante :

- **Fichiers de contrôle (Control files)** : contiennent des données sur la base elle-même (informations sur sa structure physique).
  - Ces fichiers sont d'une importance capitale pour la base.
  - Sans eux, il n'est pas possible d'ouvrir les fichiers de données de la base.
  - Ils peuvent également contenir des métadonnées relatives aux sauvegardes.
- **Fichiers de données (Data files)** : contiennent les données utilisateur ou les données des applications, ainsi que des métadonnées et le dictionnaire de données.

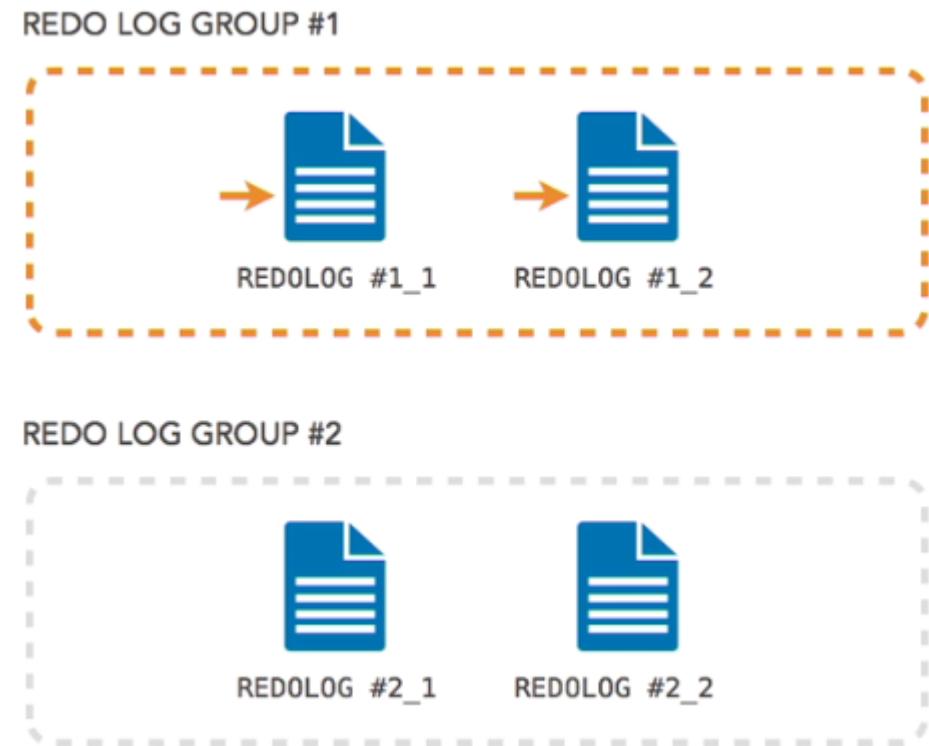


# Architecture de stockage de la base de données (2/3)

- **Fichiers de journalisation en ligne (Online redo log files)** : ils permettent la récupération d'une instance de base de données. Si le serveur de base de données connaît une défaillance et qu'aucun fichier de données n'est perdu, l'instance peut récupérer la base à l'aide des informations contenues dans ces fichiers.
- **Fichiers de sauvegarde (Backup files)** : Ils sont utilisés pour la récupération de la base de données. Les fichiers de sauvegarde sont généralement restaurés lorsqu'une défaillance physique ou une erreur utilisateur a endommagé ou supprimé les fichiers d'origine.
- **Fichiers de journalisation archivés (Archived redo log files)** : Ils contiennent l'historique complet des modifications de données (informations de journalisation) générées par l'instance.
  - A l'aide de ces fichiers et d'une sauvegarde de la base, il est possible de restaurer un fichier de données perdu.
  - En d'autres termes, les fichiers de journalisation archivés permettent de récupérer des fichiers de données restaurés.

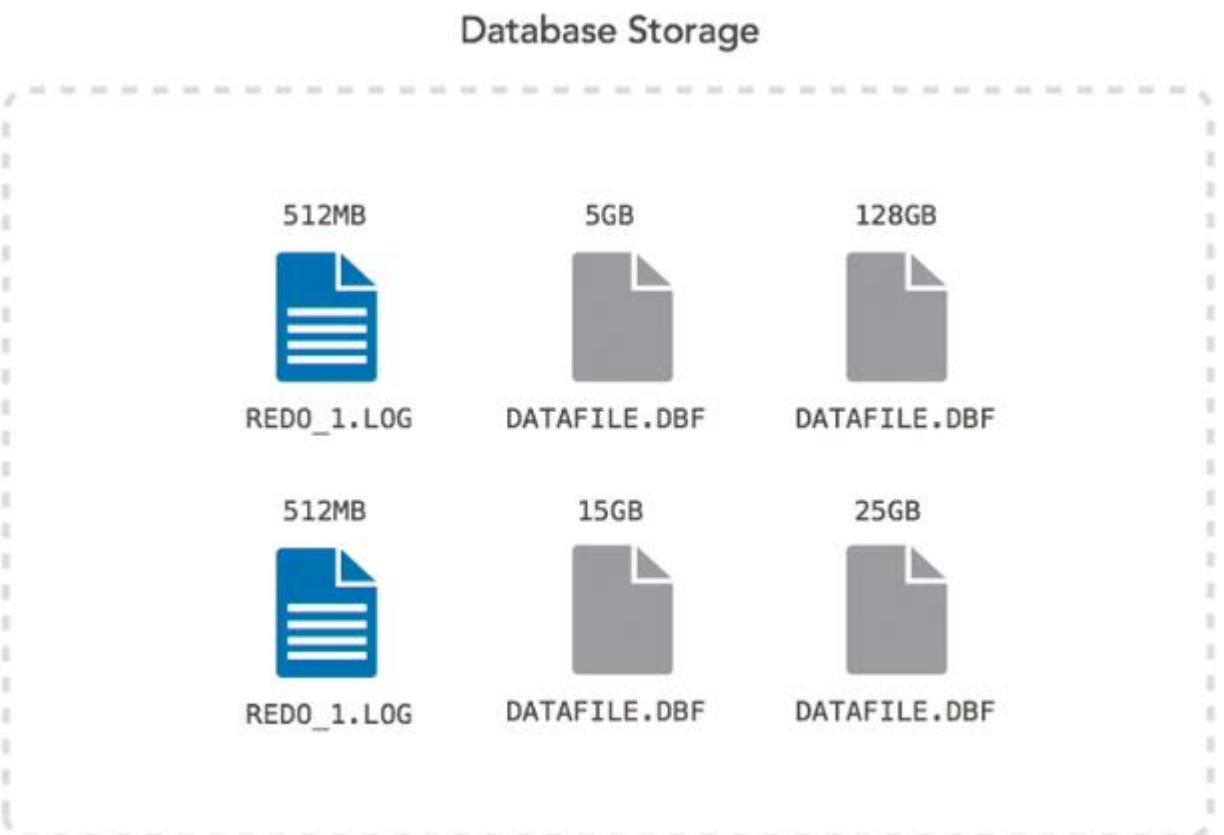


# Redo log file



- Redo log file sont cyclique : càd une fois un Redo log est plein le redo log vide stocke les données, et vice-versa
- On peut mettre les redo log dans un même groupe

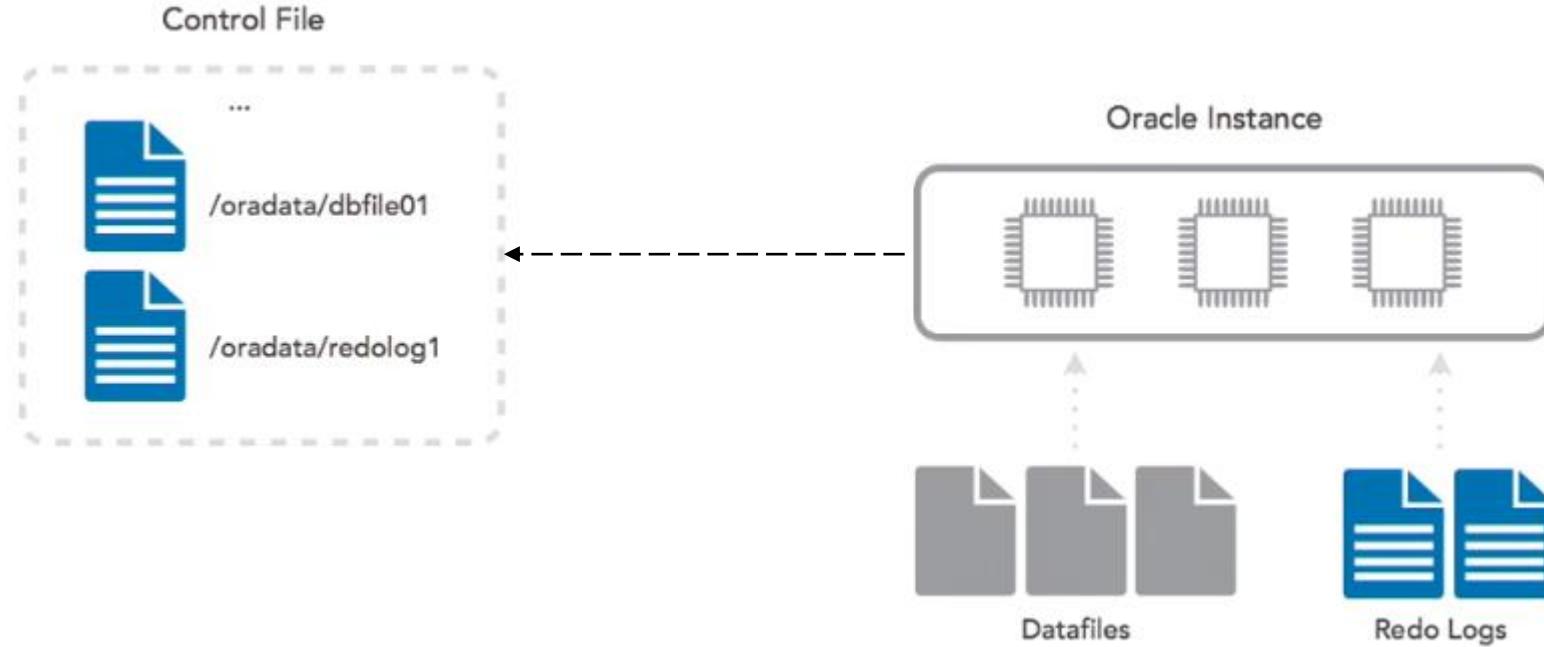
# Redo log file



**Attention :** Redo log file a comme extension LOG  
ne pas confondre avec les Log Files

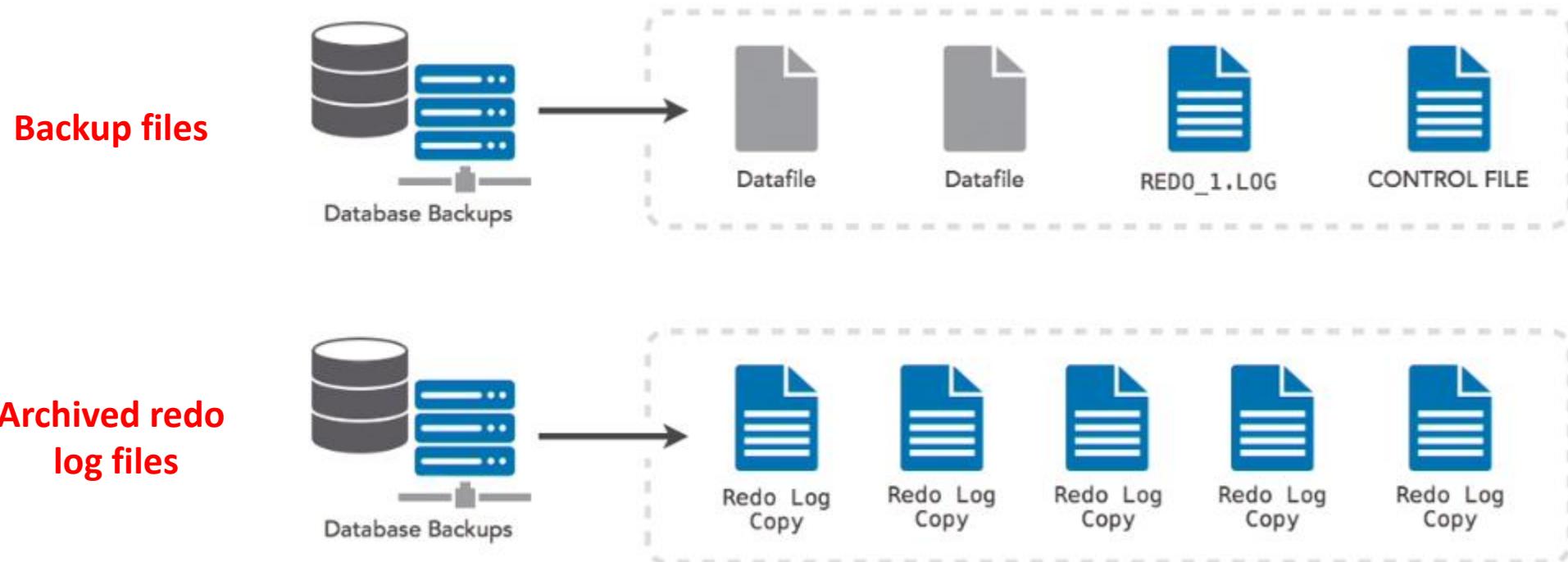
# Control file

---



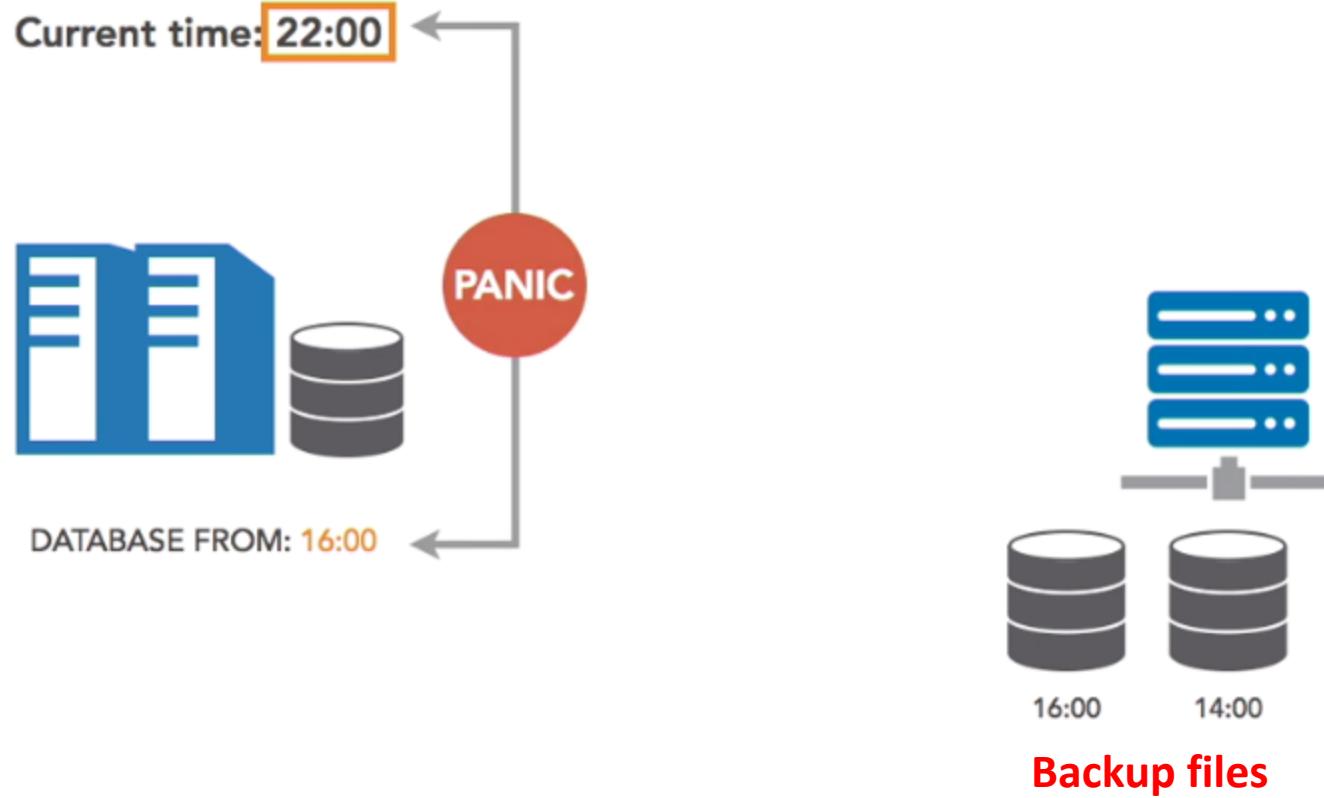
- **Control file** contient le chemin d'accès aux Datafiles et Redo Logs

# Backup files & Archived redo log files



# Backup files & Archived redo log files

---



# Backup files & Archived redo log files

---

Current time: 22:00



DATABASE FROM: 16:00

Contain list of changes  
to database:  
18:05 INSERT INTO...  
19:00 DELETE FROM...  
20:05 UPDATE...



Archived Logs



Backup files

# Backup files & Archived redo log files

---

Current time: 22:00



DATABASE FROM: 22:00



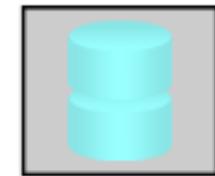
Redo Log Files

*"Roll-forward changes"*

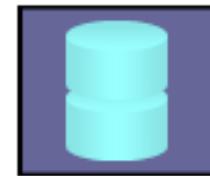


# Architecture de stockage de la base de données (3/3)

- **Fichier de paramètres (Parameter file)** : Il est utilisé pour définir la façon dont l'instance est configurée au démarrage.
- **Fichier de mots de passe (Password file)** : Il permet aux utilisateurs **sysdba**, **sysoper** et **sysasm** de se connecter à distance à la base de données et d'effectuer des tâches d'administration.
- **Fichiers trace (trace files )** : Chaque processus serveur ou en arrière-plan peut écrire dans un fichier trace associé. Lorsqu'un processus détecte une erreur interne, il effectue un dump des informations sur l'erreur dans son fichier trace. Certaines de ces informations sont destinées à l'administrateur de base de données, et d'autres au support technique Oracle.
- **Fichier d'alertes (Alert log)** : Il contient des entrées de trace spéciales. Le fichier d'alertes d'une base de données est un journal chronologique des messages et des erreurs.



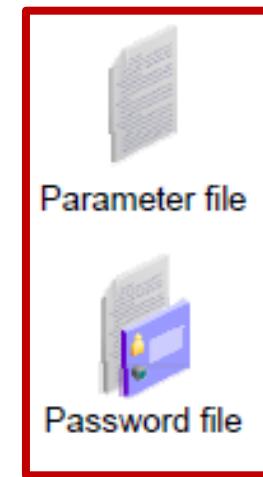
Control files



Data files



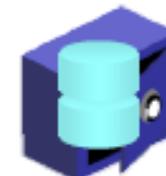
Online redo log files



Parameter file



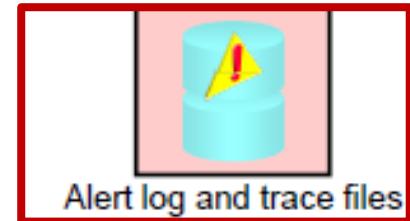
Password file



Backup files



Archived redo log files



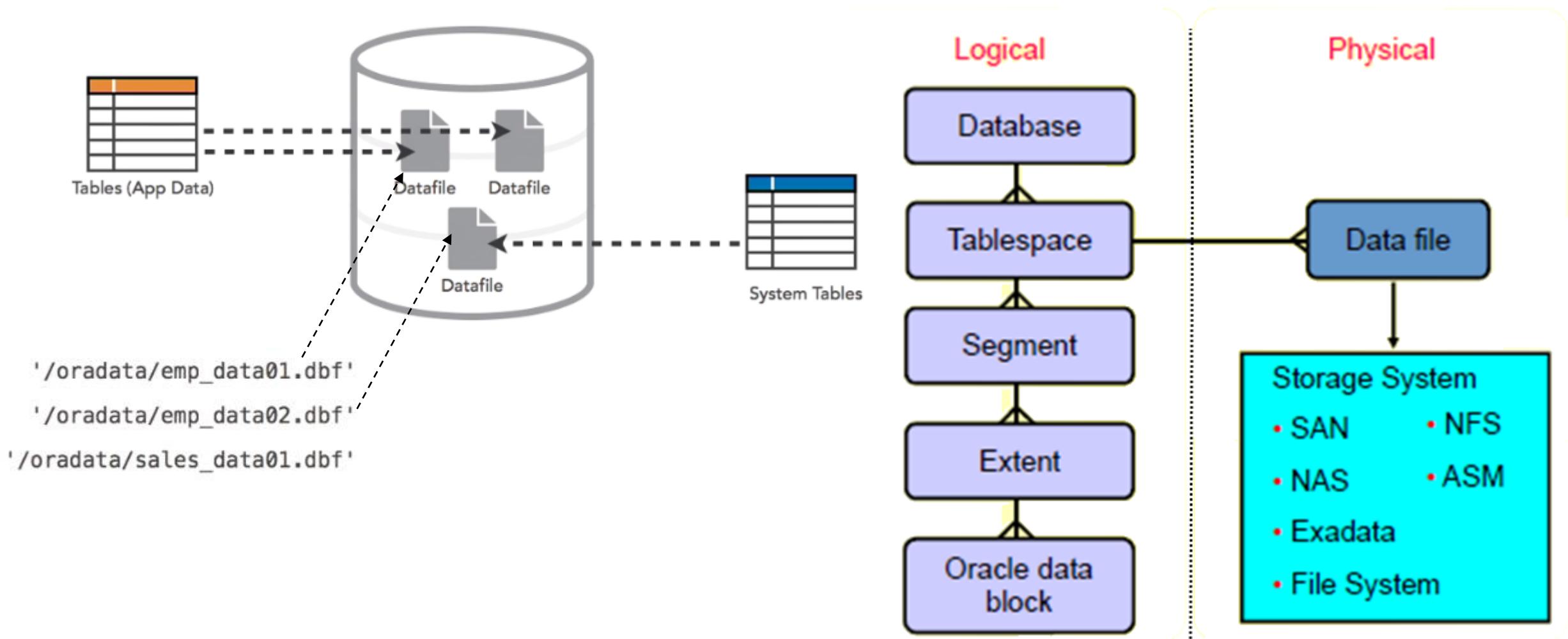
Alert log and trace files

# Plan

---

- ❑ Introduction
- ❑ Architecture d'un serveur de BD Oracle
- ❑ Structure de la mémoire
- ❑ Structure des processus
- ❑ **Structure de stockage de données**
  - ❖ Architecture de stockage de la BD
  - ❖ **Structures logiques et physiques de la BD**
- ❑ Interagir avec une base Oracle : Mémoire, processus et stockage

# Structures logiques et physiques d'une Base de Données

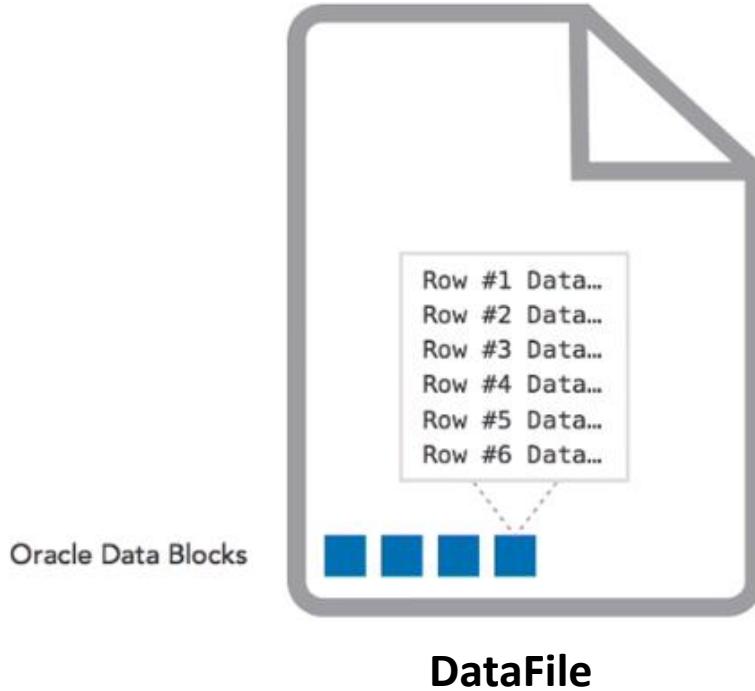


# Structures logiques et physiques d'une Base de Données

---

## Blocs de données (Data Block) :

- les données d'une base Oracle sont stockées dans les blocs de données.
- Un bloc de données correspond à un nombre d'octets spécifique d'espace physique sur le disque.
- La taille des blocs est définie lors de la création de chaque tablespace.
  - Taille peut être : 2, 4, 8, 16, 32 Ko (par défaut 8 ko)
- La base de données utilise et alloue l'espace disponible dans les blocs de données Oracle

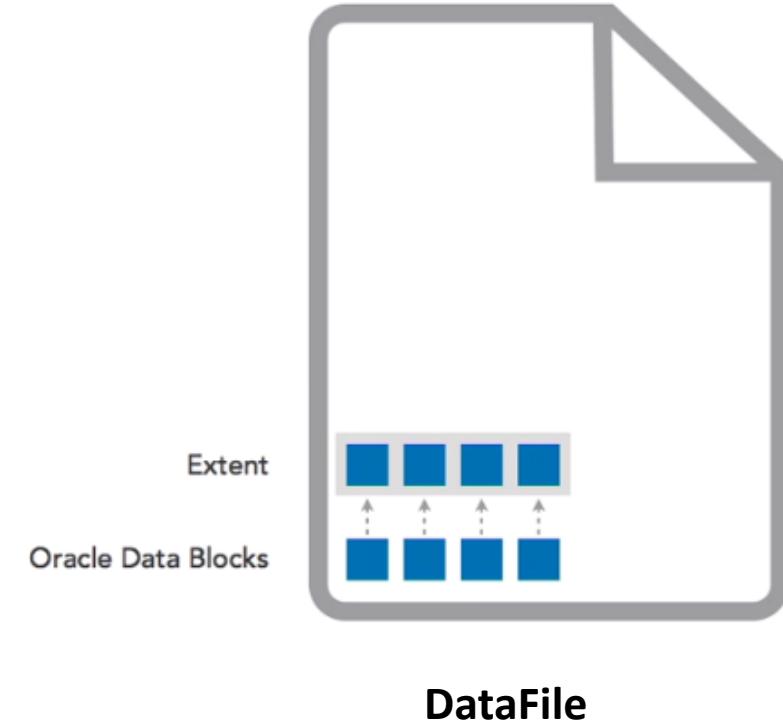


# Structures logiques et physiques d'une Base de Données

---

## Extent :

- Le terme **extent** (ensemble de **blocs contigus**) désigne le niveau suivant de l'organisation logique d'une base de données.
- Un extent consiste en un nombre précis de blocs de données contigus (obtenus dans le **cadre d'une allocation unique**) qui permettent de stocker un type d'information particulier.
  - ce qui signifie qu'il ne peut se trouver que dans un seul fichier de données.



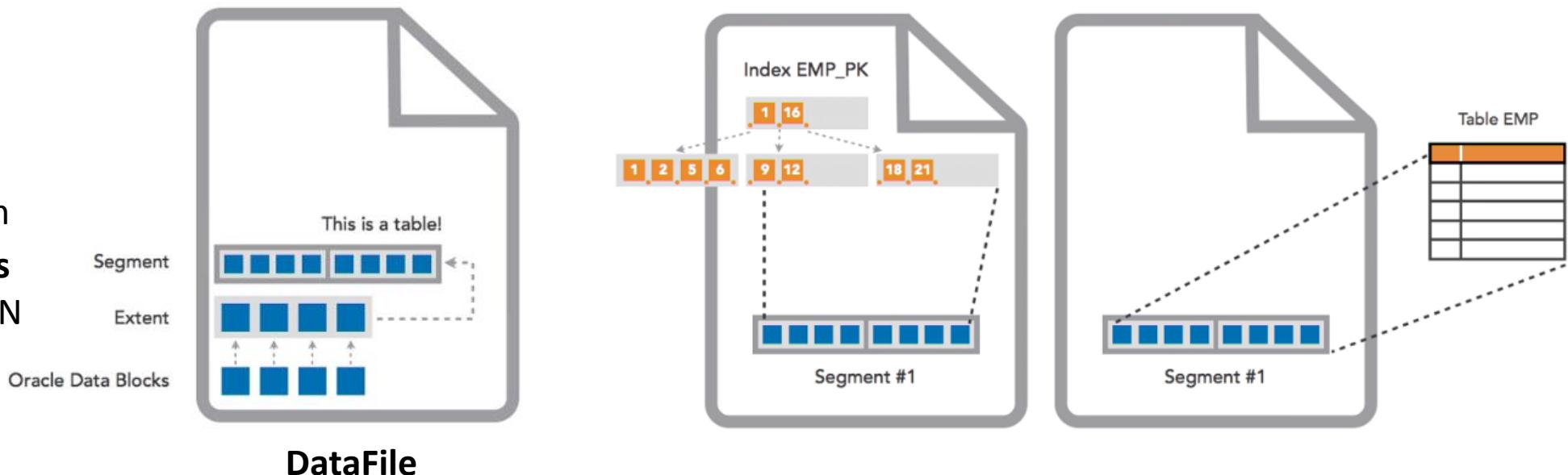
# Structures logiques et physiques d'une Base de Données

## Segment :

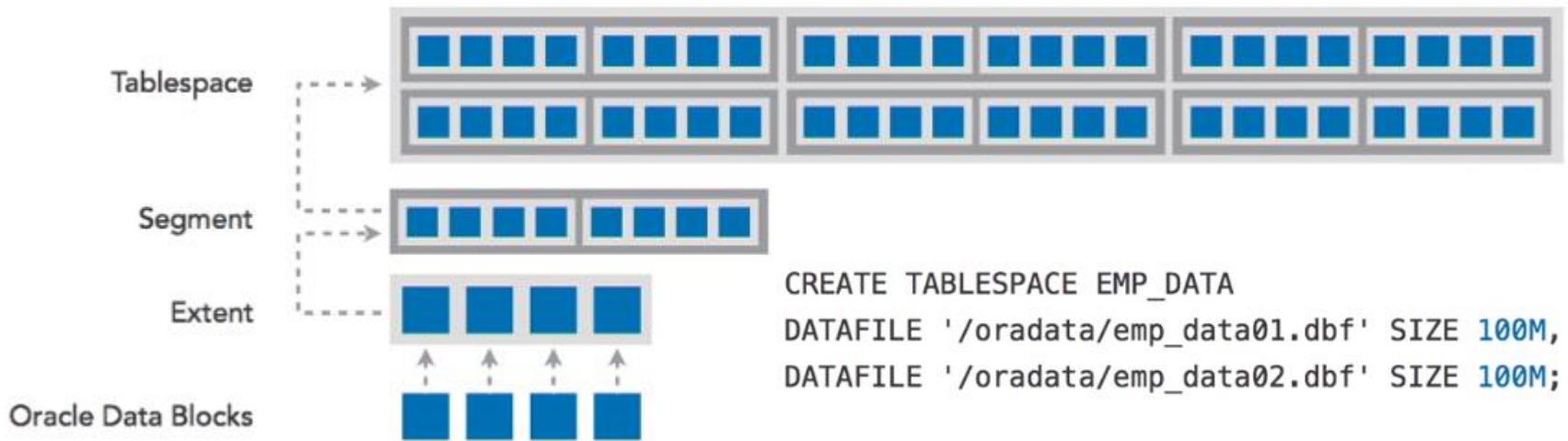
Segment est un ensemble d'extents alloués à une structure logique.

Exemples :

- Segments de données
- Segments d'index
- Segments d'annulation
- **Segments temporaires**
  - Nettoyer par SMON



# Structures logiques et physiques d'une Base de Données



```
CREATE TABLESPACE EMP_DATA  
DATAFILE '/oradata/emp_data01.dbf' SIZE 100M,  
DATAFILE '/oradata/emp_data02.dbf' SIZE 100M;
```

Tablespace: EMP\_DATA



## Tablespaces :

- les tablespaces regroupent généralement tous les objets d'une application.

Exemple, les tables, index, ...

# Structures logiques et physiques d'une Base de Données

---

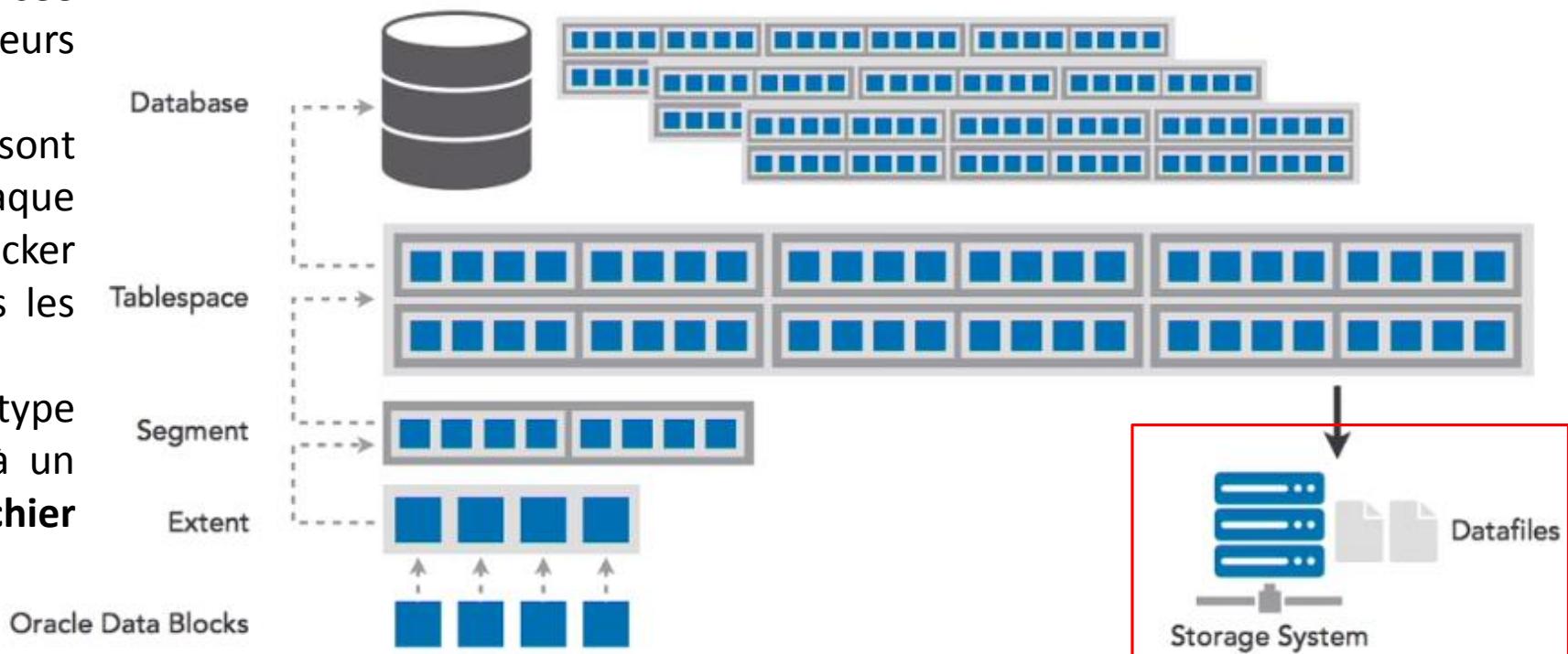
```
create table employees (
    emp_id      number not null,
    emp_name    varchar2(30) not null)
tablespace emp_data;
```



# Structures logiques et physiques d'une Base de Données

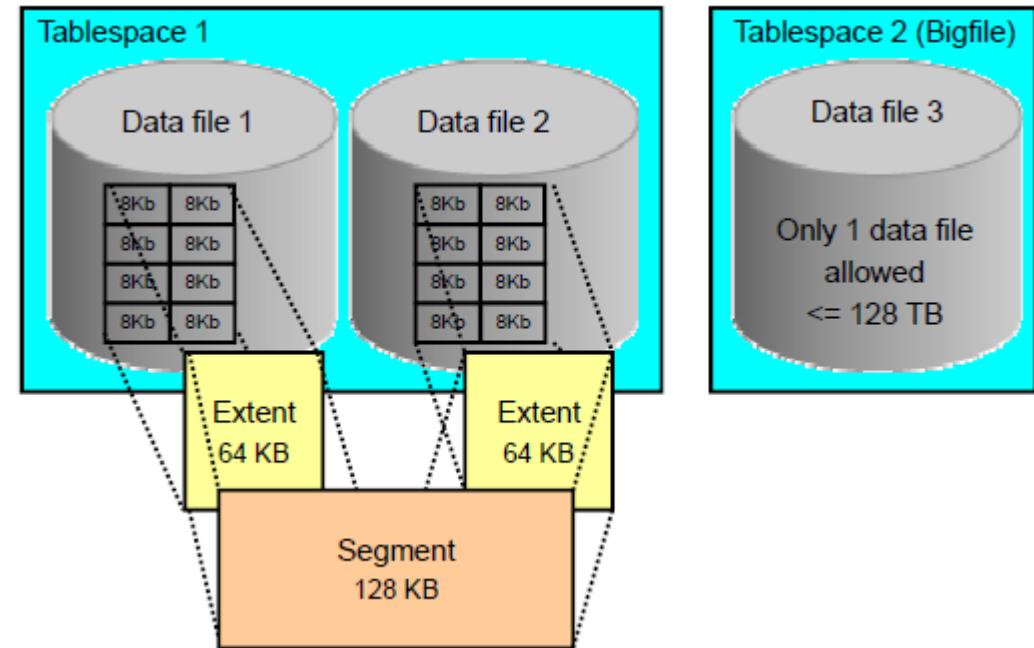
## Database (base de données) :

- Chaque base de données est divisée d'un point de vue logique en plusieurs **tablespaces**.
- Des fichiers de données (**Data files**) sont créés explicitement pour chaque tablespace afin de stocker physiquement les données de tous les **segments**.
- Si le tablespace est de type **TEMPORARY**, il n'est pas associé à un fichier de données mais à un **fichier temporaire (Temp File)**.

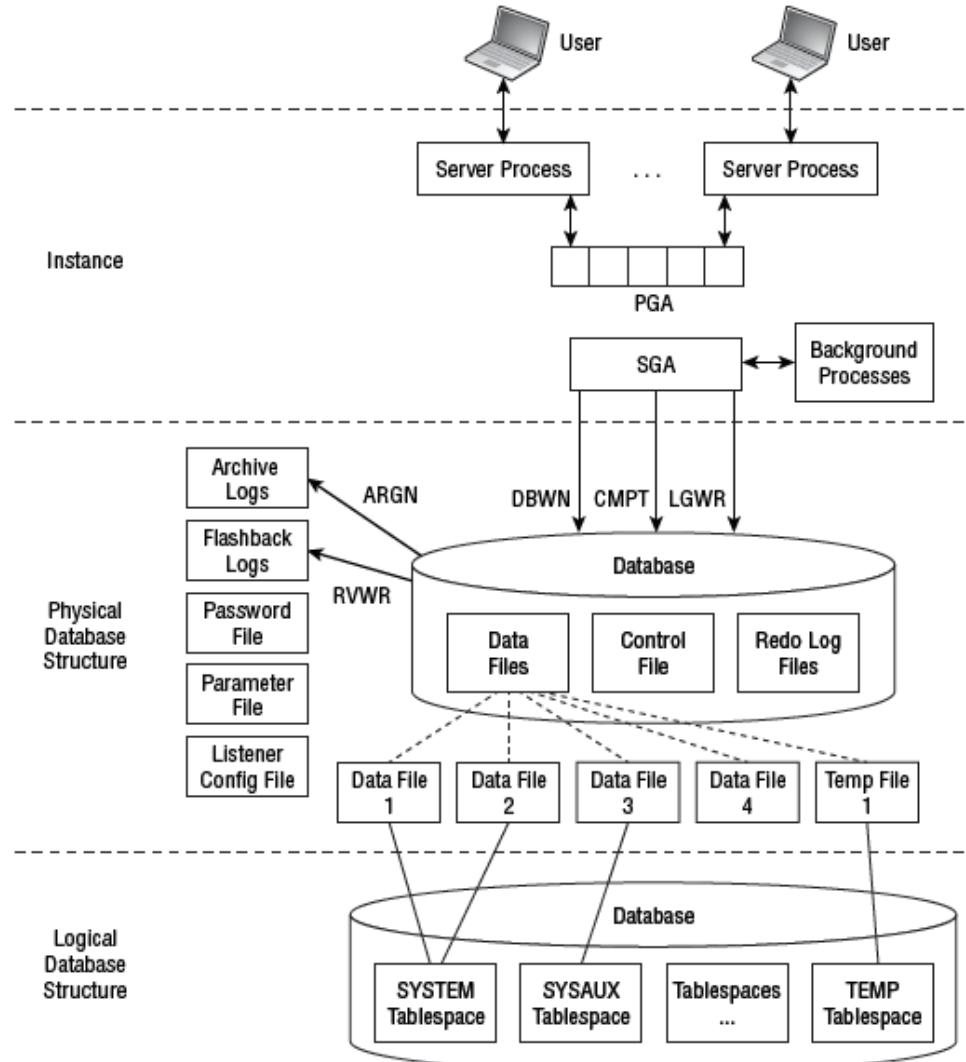


# Tablespaces spéciaux

- Une base de données est divisée en unités de stockage logiques, appelées tablespaces, qui peuvent être utilisées pour regrouper des structures logiques liées.
- Chaque base est divisée logiquement en plusieurs tablespaces :
  - Tablespace **SYSTEM** : est utilisé pour les fonctionnalités principales (par exemple, les tables du dictionnaire de données)
  - Tablespace **SYSAUX** sert aux composants de base de données supplémentaires (tels que le référentiel Enterprise Manager)
  - Tablespace **Undo** permet de faire backup des données si une transaction échoue
  - Tablespace **Temp** permet de stocker des segments temporaire quand la mémoire de PGA est pleine
- Il est déconseillé d'utiliser les tablespaces spéciaux pour stocker les données des applications



# Architecture de la BD Oracle



# Plan

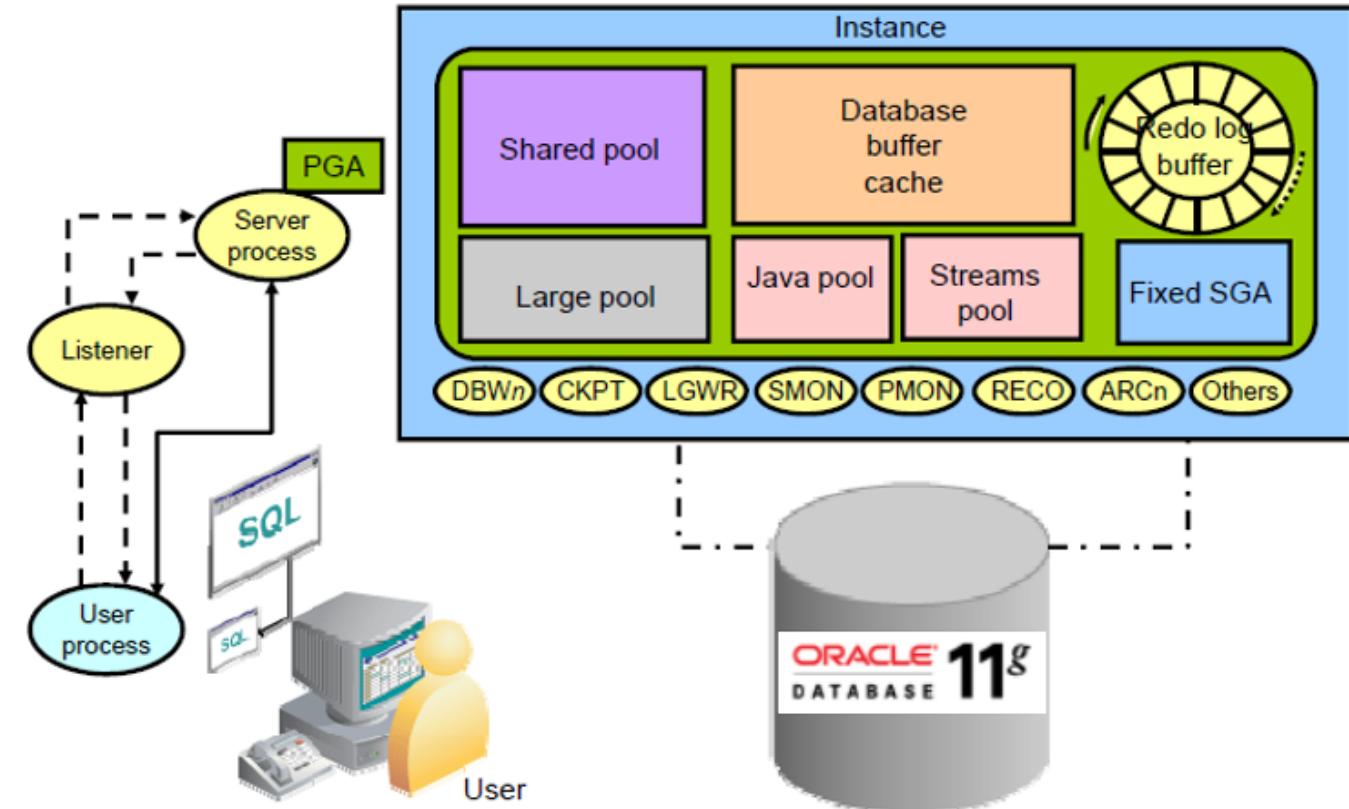
---

- ❑ Introduction
- ❑ Architecture d'un serveur de BD Oracle
- ❑ Structure de la mémoire
- ❑ Structure des processus
- ❑ Structure de stockage de données
- ❑ Interagir avec une base Oracle : Mémoire, processus et stockage**

# Interagir avec une base Oracle : Mémoire, processus et stockage

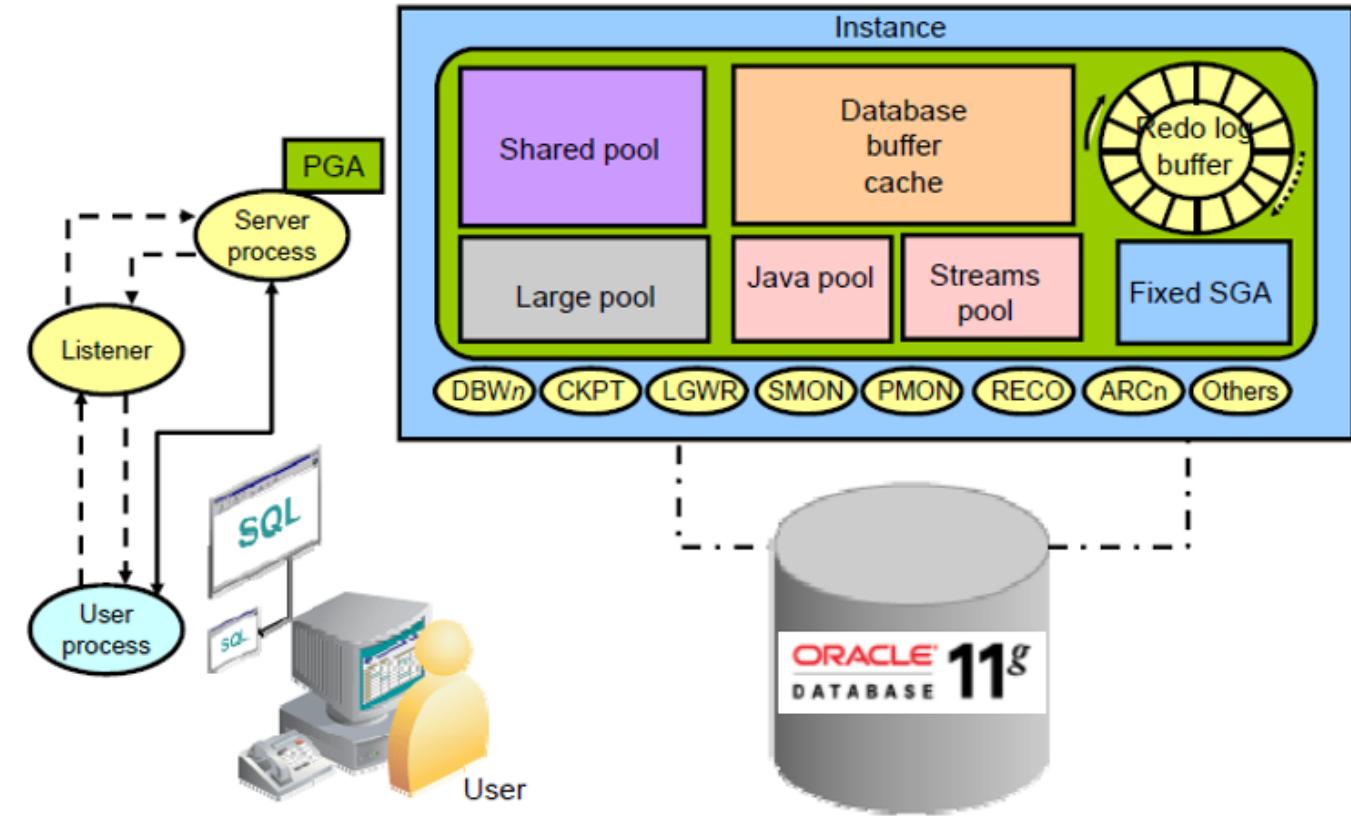
## Interagir avec une base de données Oracle

1. Une **instance** démarre sur un nœud où Oracle Database est installé. Ce nœud est souvent appelé **hôte** ou **serveur de base de données**.
2. Un utilisateur démarre une application qui lance un **processus utilisateur**. L'application essaie d'établir une *connexion* au serveur. (Il peut s'agir d'une connexion locale, client/serveur ou d'une connexion à trois niveaux (3-tiers) fournie par un niveau intermédiaire)
3. Le serveur exécute un **processus d'écoute (listener)**. Le serveur détecte la demande de connexion émise par l'application et crée un **processus serveur** dédié pour le compte du **processus utilisateur**.



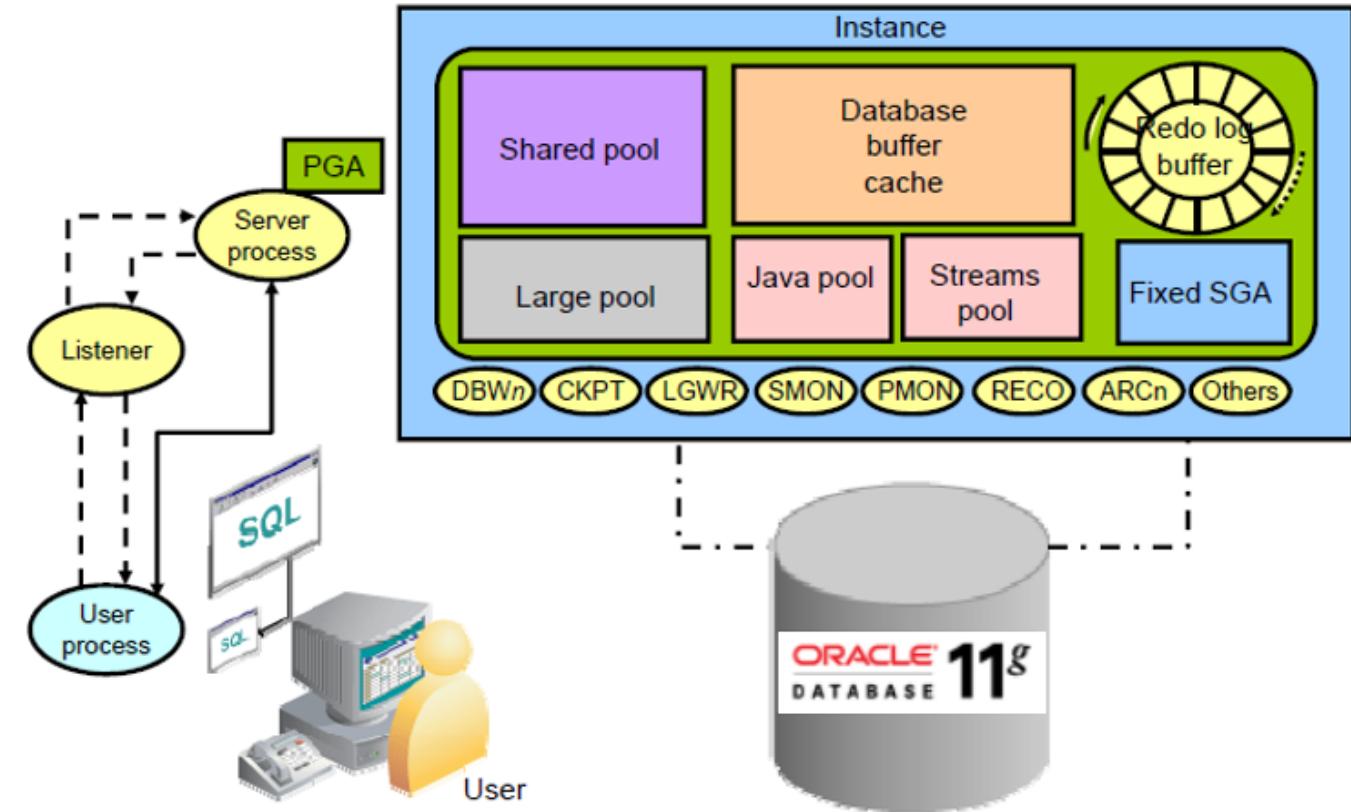
# Interagir avec une base Oracle : Mémoire, processus et stockage

4. L'utilisateur exécute une **instruction SQL** de type **LMD (langage de manipulation de données)** et valide la transaction. Par exemple, il change l'adresse d'un client dans une table et valide la modification dans la base.
5. Le **processus serveur** reçoit l'instruction et recherche dans la **zone de mémoire partagée** (composant SGA) une zone **SQL partagée** contenant une instruction SQL semblable. S'il trouve cette zone SQL partagée, le processus serveur vérifie les privilèges de l'utilisateur pour l'accès aux données demandées et utilise la zone SQL partagée existante pour traiter l'instruction. A défaut, une nouvelle zone SQL partagée est allouée à l'instruction pour que celle-ci puisse être analysée et traitée.



# Interagir avec une base Oracle : Mémoire, processus et stockage

6. Le processus serveur extrait les valeurs de données nécessaires, à partir du fichier de données réel (table) ou à partir du cache de tampons de la base.
7. Le processus serveur modifie les données dans la mémoire SGA. Comme la transaction est validée, le **processus Log Writer (LGWR)** enregistre immédiatement la transaction dans le **fichier de journalisation (fichier redo log)**. Le **processus Database Writer (DBWn)** écrit les blocs modifiés sur le disque au moment opportun.



# Interagir avec une base Oracle : Mémoire, processus et stockage

8. Si la transaction réussit, le processus serveur envoie un message à l'application via le réseau. Si elle échoue, un message d'erreur est transmis.
9. Tout au long de cette procédure, les autres processus en arrière-plan s'exécutent pour détecter les conditions nécessitant leur intervention. Par ailleurs, le serveur de base de données gère les transactions des autres utilisateurs et empêche les problèmes de contention entre les transactions qui demandent les mêmes données.

