

Chapitre : 02

Composants de l'architecture d'Oracle

Composants de l'architecture d'Oracle

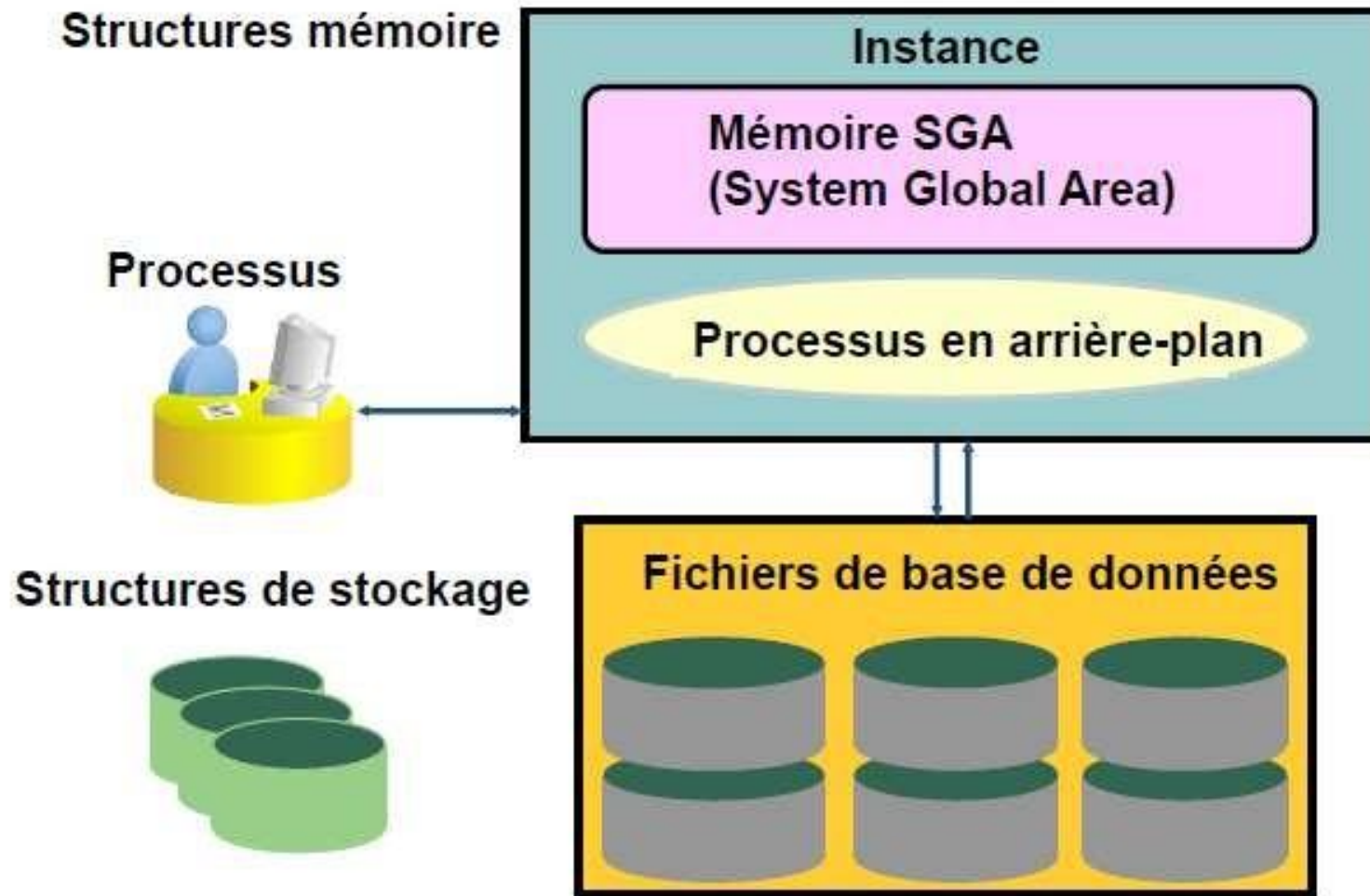
Architecture Oracle

L'architecture oracle est constituée de:

- Une instance
- Une base de données appelée database.

Composants de l'architecture d'Oracle

Architecture Oracle



Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

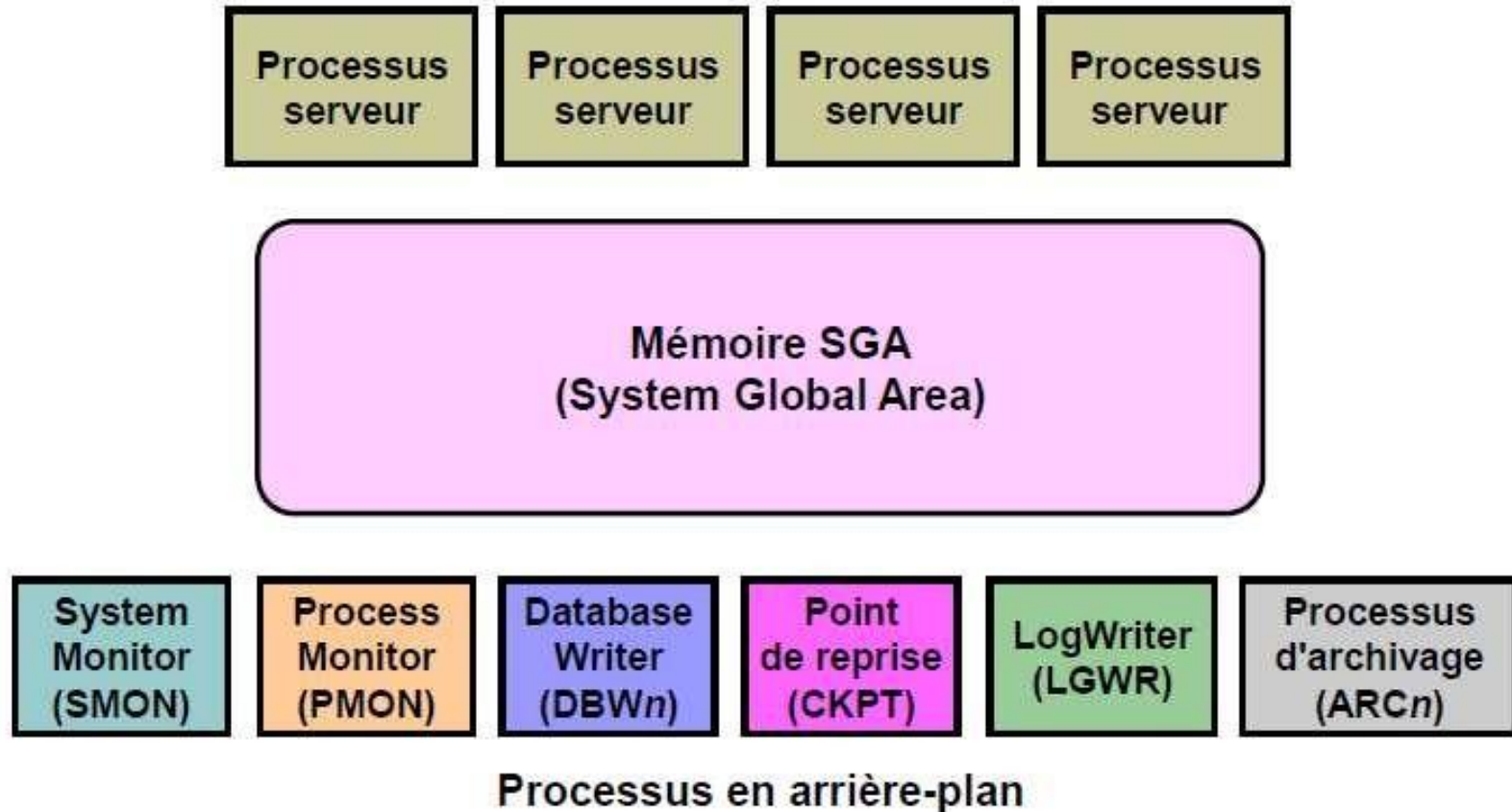
Une instance est un moyen pour accéder à une base de données Oracle (ouvre une unique base de données)

Elle est constituée de :

- Une zone de mémoire partagée appelée System Global Area (SGA)
- Un ensemble de processus d'arrière plan ayant chacun un rôle bien précis
- Un ensemble de processus serveur chargés de traiter les requêtes des utilisateurs

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle



Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Structures mémoire oracle

Les structures mémoire de base associées à une instance Oracle sont :

- Mémoire SGA (System Global Area) :
- Mémoire PGA (Program Global Area) :

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Structures mémoire oracle : Mémoire SGA

- Elle est partagée par tous les processus serveur et les processus en arrière-plan.
- C'est une zone de mémoire contenant les données et les informations de contrôle de l'instance.
- C'est elle qui assure le partage des données entre les utilisateurs.
- Elle est allouée au démarrage de l'instance et est libérée lors de l'arrêt de celle-ci.
- Oracle utilise la mémoire SGA comme buffer intermédiaire (plus rapide que le disque) pour l'échange de données entre processus

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Structures mémoire oracle : Mémoire SGA

Elle est divisée en:

- ✓ Trois composants obligatoires :
 - Shared pool
 - Database buffer cache (Le cache de données)
 - redo log buffer (Le cache de reprise)
- ✓ Trois composants optionnels
 - Java pool
 - Large pool
 - Streams pool

Ces différentes zones mémoires sont configurées à l'aide du paramètre contenu dans le fichier de paramètres SPFILE.

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Structures mémoire oracle : Mémoire SGA

➤ Database Buffer cache

- Est utilisé pour stocker des blocs de données en mémoire afin d'accélérer l'interrogation et/ou la modification
- Aucune modification est faite directement sur les données du disque
- Oracle lit les données suite à la demande d'un *processus utilisateur* et ensuite valide les modifications sur le disque
- Il utilise un algorithme nommé LRU moins récemment utilisés (Least-Recently Used) pour déterminer les données à libérer du cache

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Structures mémoire oracle : Mémoire SGA

➤ **Database Buffer cache** : Organisation du buffer cache

Le buffer cache est organisé en 2 listes :

1. La “dirty list” : contient les buffers qui ont été modifiés mais ne sont pas encore écrits sur le disque
2. La liste LRU : contient
 - les buffers libres (qui n'ont pas été modifiés),
 - les “pinned buffers” (qui sont en cours de modification),
 - les “dirty buffers” (qui n'ont pas encore été déplacés dans la “dirty list”)

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Structures mémoire oracle : Mémoire SGA

➤ Shared Pool:

- C'est une zone de partage des requêtes et du dictionnaire Oracle.
- Elle est globalement dimensionnée par le paramètre `SHARED_POOL_SIZE` ;
- La répartition entre le Library Cache et le Dictionary Cache est assurée par Oracle.

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Structures mémoire oracle : Mémoire SGA

➤ Shared Pool:

- Elle permet de stocker plusieurs éléments cruciaux pour la gestion des données :
 - Library cache : permet d'analyser l'ordre d'exécution d'une requête SQL et de définir un plan d'exécution.
 - Dictionary cache : stocke les données en provenance du dictionnaire afin d'accélérer l'accès au dictionnaire (nom d'utilisateurs, privilèges, etc.).
 - SQL area : stocke les requêtes SQL les plus récemment utilisées par les utilisateurs de base de données.
 - Si la même requête est ré-exécutée, le serveur n'analyse pas son ordre. Cela permet d'améliorer la performance des applications

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Structures mémoire oracle : Mémoire SGA

➤ Buffer Redo Log

- C'est un buffer circulaire (quand le buffer est plein, il repart au début ... après avoir été écrit sur disque dans les fichiers de Redo Log)
- Il stocke les modifications réalisées sur la base de données avec les opérations: insert, delete, update, create, alter et drop.
- Il permet à oracle de reconstruire les modifications des données en cas de panne
- L'information Redo reste dans le buffer Redo log jusqu'à ce qu'oracle la stocke sur le disque
- Sa taille est définie par LOG_BUFFER

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Structures mémoire oracle : Mémoire SGA

- **Zone de mémoire LARGE POOL** : zone fournissant d'importantes allocations mémoire pour certains processus utilisant beaucoup de mémoire, tels que
 - les opérations de sauvegarde et de récupération Oracle
 - les processus serveur d'entrée-sortie
- **Zone de mémoire Java** : zone utilisée pour l'ensemble du code Java et des données propres à la session dans la JVM (Java Virtual Machine)
- **Zone de mémoire Streams** : zone réservée notamment lors de la réplication de données entre bases de données distantes. Elle est dimensionnée par le paramètre `STREAMS_POOL_SIZE`.

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Structures mémoire oracle : Mémoire SGA

La vue dynamique **V\$MEMORY_TARGET_ADVICE**

Cette vue dynamique de performances, permet de suivre l'allocation dynamique et visualiser les différentes valeurs de l'allocation dynamique de la mémoire.

Cette vue contient les colonnes :

- Memory size : taille réelle de la mémoire totale allouée à l'instance
- Size_factor : coefficient de taille
- Estd_db_time : taille de l'instance utilisée en mémoire en moyenne par rapport aux facteurs size-factor et time_factor.
- Time_factor : coefficient de temps
- Version :

```
Select * from v$memory_target_advice ;
```

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Structures mémoire oracle : Mémoire SGA

La vue **V\$MEMORY_DYNAMIC_COMPONENTS**, permet de visualiser les différentes valeurs de chaque pool, entre autre la shared_pool, le database buffer cache, le large pool, etc ...

```
Select component, current_size, min_size, max_size  
from v$memory_dynamic_components;
```


Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Structures mémoire oracle : Mémoire SGA

Le fichier de paramètres (init.ora ou SPFILE.ORA)

Au démarrage, l'instance lit un fichier de paramètres qui contient des paramètres d'initialisation.

Ce fichier est géré par le DBA.

Les paramètres d'initialisation permettent notamment à l'instance :

- D'allouer la mémoire souhaitée aux différentes structures de la SGA
- De trouver le nom et l'emplacement des fichiers de contrôle de la base

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Structures mémoire oracle : Mémoire SGA

Le fichier de paramètres (init.ora ou SPFILE.ORA)

Il existe 2 types de paramètres, les paramètres dynamiques et les paramètres statiques.

Les paramètres dynamiques sont modifiables sans avoir besoin d'arrêter la base de données.

Les vues V\$SYSTEM_PARAMETER et V\$SYSTEM_PARAMETER2 (idem V\$SYSTEM_PARAMETER avec une mise en forme des paramètres) permettent de connaître la valeur des paramètres de l'instance en cours de fonctionnement.

```
Select * from V$SYSTEM_PARAMETER ;
```

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Structures mémoire oracle : Mémoire SGA

Le fichier de paramètres (init.ora ou SPFILE.ORA) : Règles :

- Les paramètres sont spécifiés sous la forme `nom_paramètre = valeur`
- Tous les paramètres sont optionnels et ont une valeur par défaut
- Des commentaires peuvent être inclus et commencent par le caractère `#`
- La valeur peut être spécifiée entre des guillemets doubles si elle contient des caractères spéciaux (égal, espace, ...)
- Les valeurs multiples sont spécifiées entre parenthèses, séparées par des virgules

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

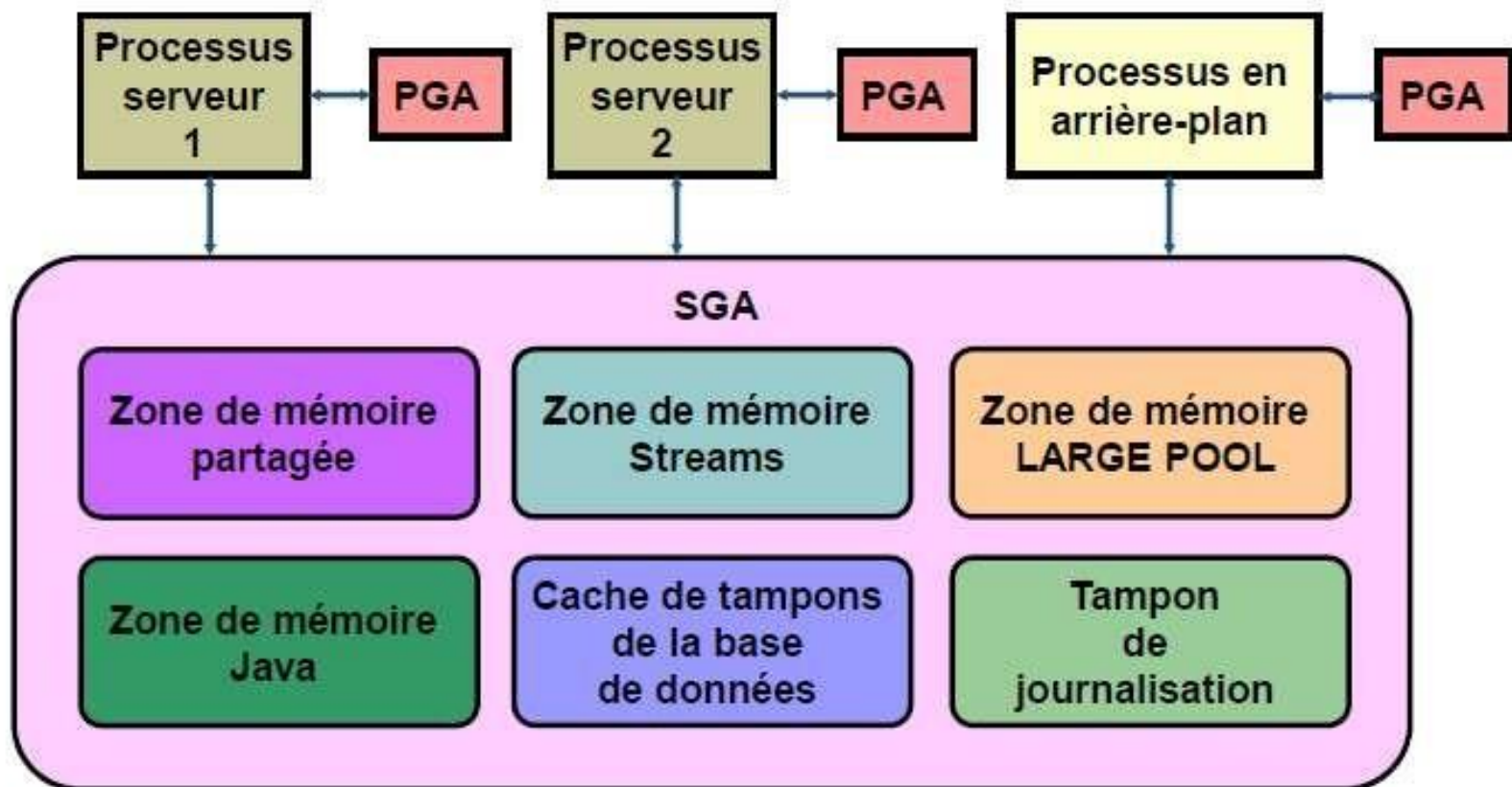
Structures mémoire oracle : **Mémoire PGA**

- Mémoire privée des différents processus distribuée au moment de la connexion d'un client.
- propre à chaque processus serveur et processus en arrière-plan.
- il existe une mémoire PGA pour chaque processus

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Structures mémoire oracle : Mémoire PGA



Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Processus en arrière plan

Le fonctionnement d'une base Oracle est assuré par un ensemble de processus imbriqués qui réalisent de nombreuses actions.

Pour plus de simplicité, nous avons regroupé les processus en deux familles :

- les indispensables: ils sont présents dès qu'une base Oracle fonctionne. Ils sont requis pour en assurer le fonctionnement minimal. Si l'un d'eux s'arrête, la base de données n'est plus opérationnelle.
- les optionnels: ils peuvent être lancés pour assurer des fonctions complémentaires, comme l'archivage des redo log. Si l'un de ces processus optionnels n'est pas démarré, cela ne met pas en cause le fonctionnement global de la base de données. Seule la tâche assurée par ce processus optionnel ne sera pas réalisée.

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Structures mémoire oracle : Mémoire PGA - Processus en arrière plan

- Les processus indispensables:
 - DBWR (*DatabaseWriter*) ;
 - LGWR (*Log Writer*) ;
 - CKPT (*Checkpoint*) ;
 - PMON (*ProcessMonitor*) ;
 - SMON (*System Monitor*).
- Les processus optionnels :
 - Listener ou listenerNet ;
 - Archiver (ARCn) : en base « archivée » il archive des fichiers de Redo Log chaque fois qu'un fichier Redo est plein.
 - Recover (RECO) : gère les bases de données distribuées.
 - Dispatcher (Dnnnn) : présent en serveur partagé.
 - ...

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Processus en arrière plan

1. DBWR (*Database Writer*)

- Ecrit les blocs modifiés dans le cache de données sur les disques.
- **DBWn** se déclenchera lors des événements suivants :
 - Lorsque le nombre de bloc *dirty* atteint une certaine limite
 - Lorsqu'un processus sera à la recherche de blocs libres dans le *Database Buffer Cache*, et qu'il ne sera pas en mesure d'en trouver.
 - Lors de timeouts (environ toutes les 3 secondes par défaut)
 - Lors d'un *checkpoint*.

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Processus en arrière plan

2. LOGWR (LoGWRiter)

- Ecrit dans les fichiers Redo Log le contenu du cache Redo Log
- Le processus LGWR transcrit les informations contenues dans le *REDO LOG Buffer* vers les fichiers *REDO LOG FILE* quand :
 - une transaction s'est terminée avec un COMMIT
 - le REDO LOG Buffer est au 1/3 plein
 - Avant que DBWn n'écrive le contenu du Database Buffer Cache dans les fichiers du disque dur

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Processus en arrière plan

3. Checkpoint (CKPT) (Point de reprise) :

- enregistre les checkpoints dans l'en-tête des fichiers de données.
- Lorsque qu'un Checkpoint a lieu, toutes les informations qui se trouvent en mémoire sont enregistrées sur disque à l'emplacement prévu.
- Cet évènement correspond à un « jalon » permettant la restauration des données jusqu'à ce point précis dans le temps.
- CKPT peut à son tour déclencher DBWR et LGWR.

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Processus en arrière plan

4. Process Monitor (PMON) :

chargé du nettoyage lors de l'échec d'un processus utilisateur. Il libère les ressources de sessions qui se sont mal terminées.

5. System Monitor (SMON) :

restauration de l'instance après un arrêt anormal. C'est le gardien de la cohésion des données. Une instance cohérente est établie chaque fois que la base est démarrée.

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Processus Serveurs

- Le rôle des processus serveur est fondamental. Dès que l'application demande l'accès à l'instance, ce sont eux qui vérifient que les données sont présentes en mémoire.
 - Dans l'affirmative, le processus serveur traite vos ordres SQL.
 - Dans la négative, les processus serveur lisent les fichiers de la base de données pour les « monter » en mémoire SGA, avant de traiter l'ordre SQL.
- Ce n'est pas un processus permanent de l'instance
- Le processus serveur est créé lors de chaque connexion
- Il assure le lien entre le processus utilisateur, les processus d'arrière plan, la SGA et les fichiers qui composent la base.

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Processus utilisateurs

- Processus client exécutant le code d'une application (PRO*C, FORMS, ...) ou d'un Outil Oracle (SQL*PLUS, ENTREPRISE MANAGER, ...)
- Processus souvent exécuté sur une machine différente de celle où réside le serveur Oracle
- Processus qui établit une communication avec Oracle via un protocole de communication et SQLNET
- La communication est gérée via le User Programme Interface (UPI)

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Connexion a un serveur Oracle

- ❑ Une connexion est un chemin de communication entre un processus utilisateur et un processus serveur. Il existe trois types de connexions grâce auxquelles un utilisateur peut accéder à un Serveur Oracle :
- **Connexion locale** : Selon cette méthode, un utilisateur est directement connecté sur la machine faisant office de Serveur Oracle.
- **Connexion Deux Tiers** : Ce type de connexion est couramment nommé "Connexion 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.

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Connexion a un serveur Oracle

- Session: une session est une connexion spécifique d'un utilisateur à un serveur oracle.
- La session démarre lorsque l'utilisateur est authentifié par le serveur oracle et se termine lorsque l'utilisateur se déconnecte ou en cas de déconnexion anormale.

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Connexion a un serveur Oracle

Oracle supporte deux mode de fonctionnement:

1. Serveur dédié : chaque fois qu'un utilisateur se connecte, il est pris en charge par un processus serveur.

- Si 100 utilisateurs se connectent, 100 processus serveurs sont créés de même
- Meilleure configuration, si les ressources matérielles le permettent.
- ✓ **Avantage:**
 - Une commande SQL est tout de suite et directement prise en compte par un processus serveur
- ✓ **Inconvénient:**
 - Chaque processus serveur occupe une zone mémoire et utilise la CPU

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

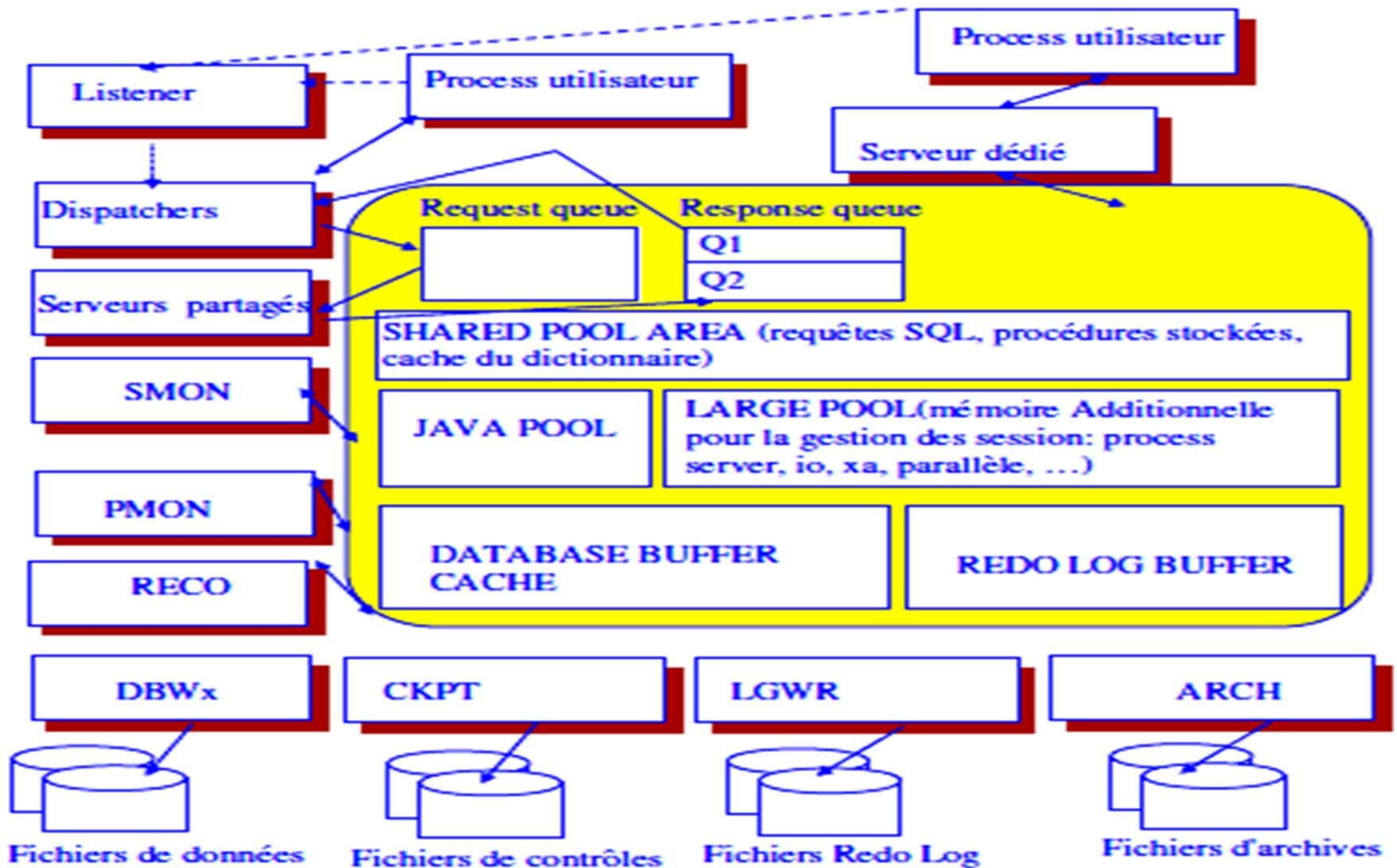
Connexion a un serveur Oracle

2.Serveur Partagé : c'est un groupe de processus serveurs qui s'occupent d'un grand nombre de processus utilisateurs.

- Les processus utilisateurs sont alloués à un processus DISPATCHER, celui-ci met les requêtes utilisateurs dans une file d'attente, et le processus serveur exécute toutes les requêtes, une par une
- ✓ **Avantage:**
 - Réduire la charge de la CPU et utilise moins de mémoire
- ✓ **Inconvénient:**
 - Lors de forte utilisations de la BDD, il risque d'y avoir des temps d'attente (moins de performance)

Composants de l'architecture d'Oracle

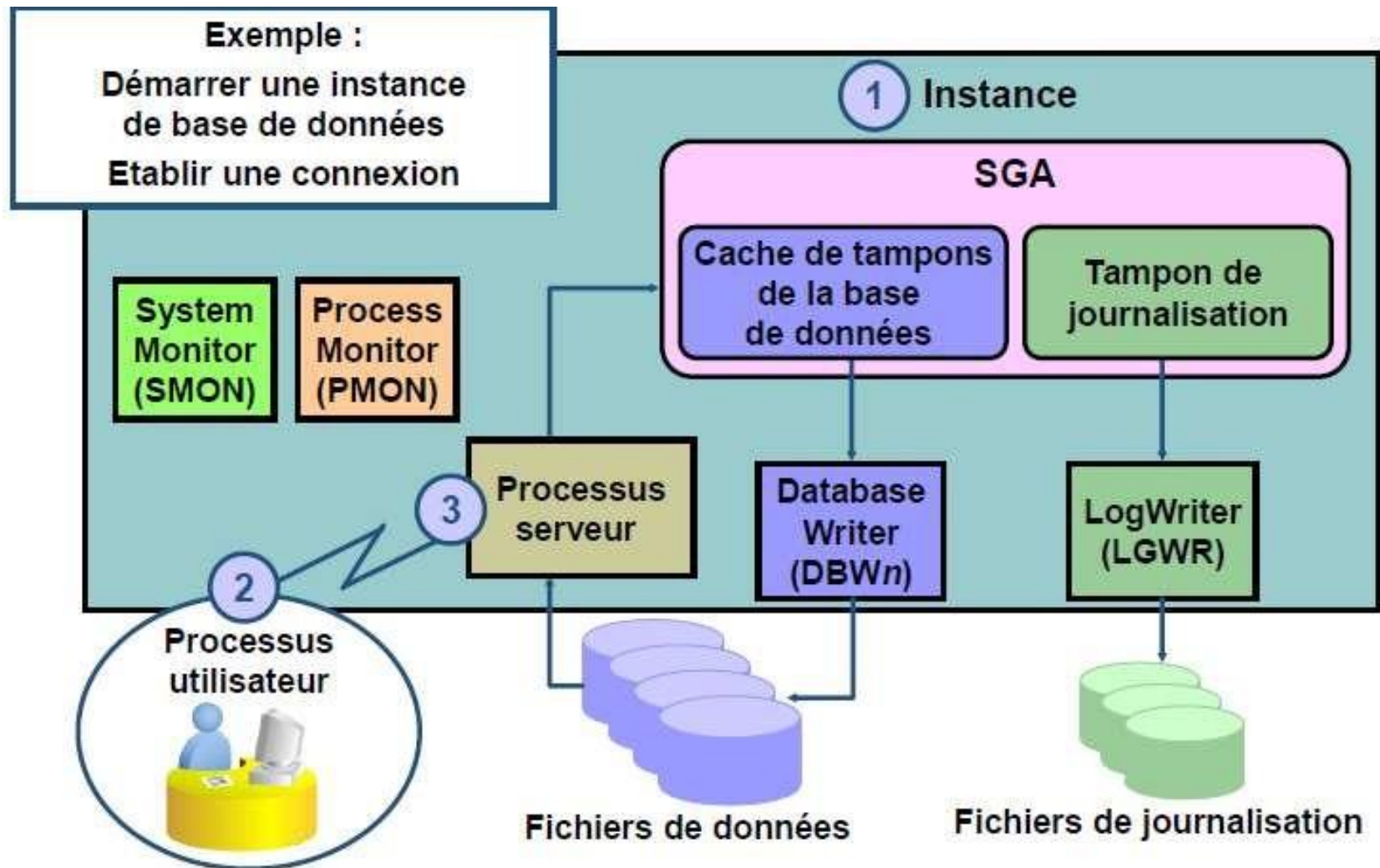
Architecture Oracle: 1. Instance Oracle



Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Traitement d'une requête SQL



Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

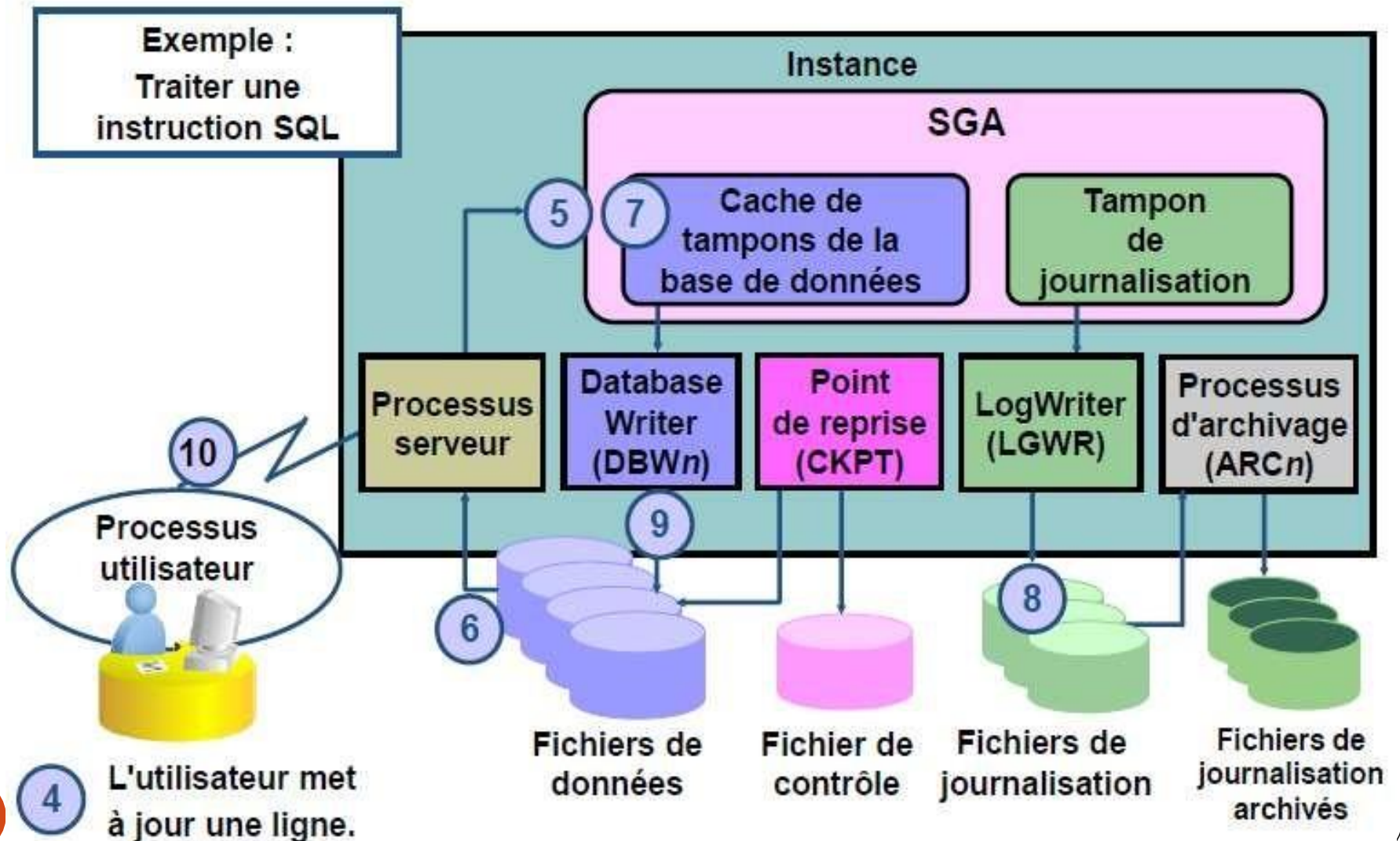
Traitement d'une requête SQL

- L'exemple illustre une configuration Oracle dans laquelle l'utilisateur et les processus serveur associés utilisent des ordinateurs distincts (connectés entre eux via un réseau)
 - 1) Une instance a été démarrée sur l'ordinateur exécutant Oracle (souvent appelé hôte ou serveur de base de données).
 - 2) L'ordinateur exécutant l'application (ordinateur local ou poste client) utilise un processus utilisateur. L'application client tente d'établir une connexion avec l'instance en utilisant le pilote Oracle Net Services.
 - 3) L'instance détecte la demande de connexion émanant de l'application et se connecte à un processus serveur pour le compte du processus utilisateur.

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Traitement d'une requête SQL



Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Traitement d'une requête SQL

- ...traiter une instruction SQL
- 4) L'utilisateur met à jour une ligne
- 5) Le processus serveur reçoit l'instruction et vérifie si elle se trouve déjà dans la zone de mémoire partagée de la mémoire SGA
 - Si une zone SQL partagée est détectée, le processus serveur vérifie les privilèges d'accès de l'utilisateur par rapport aux données demandées et la zone SQL partagée existante est utilisée pour le traitement de l'instruction.
 - Si l'instruction ne se trouve pas dans la zone de mémoire partagée, une nouvelle zone SQL partagée est allouée pour celle-ci afin qu'elle puisse être analysée et traitée.
- 6) Le processus serveur extrait les valeurs des données nécessaires du fichier de données (table) ou des blocs de données stockés dans la mémoire SGA

Composants de l'architecture d'Oracle

Architecture Oracle: 1. Instance Oracle

Traitement d'une requête SQL

➤ ...

- 7) Le processus serveur modifie les données de la table dans la mémoire SGA.
- 8) Lorsque la transaction est validée (commit), le processus LGWR enregistre immédiatement la transaction dans le fichier de journalisation (fichier redo log).
- 9) Le processus DBWn *écrit les blocs modifiés sur le disque* lorsque cela s'avère utile.
- 10) Le processus serveur envoie un message de succès ou d'erreur à l'application via le réseau.

Composants de l'architecture d'Oracle

Architecture Oracle: 2. Base de données Oracle

La base de données est l'ensemble des fichiers qui permettent de gérer les données de la base.

Elle a deux Structure:

1. Structure physique:
2. Structure logique

Architecture Oracle: 2. Base de données Oracle

1. Structure physique:

- **Un fichier de contrôle**, contenant les informations sur tous les autres fichiers de la base (nom, emplacement, taille).
- **Fichiers de Redo Log**, contenant l'activité des sessions connectées à la base. Ce sont des journaux de transactions de la base. Ils sont organisés en groupe possédant le même nombre de membres.
- Et éventuellement, de fichiers de **Redo Log archivés** contenant les archives d'anciens fichiers de Redo Log.
- un ou plusieurs **fichiers de données** qui contiennent les données des tables de la base.
- Autres fichiers importants: (fichier de paramètres, fichier de mots de passe)

Composants de l'architecture d'Oracle

Architecture Oracle: 2. Base de données Oracle



Architecture Oracle: 2. Base de données Oracle

2. Structure logique

- Tablespace,
- segment,
- extent,
- bloc

Composants de l'architecture d'Oracle

Une instance Oracle
=
Ensemble des zones mémoires
+
Processus alloués à une base de données

Objectif du cours

- Concrétiser la **conception physique** des bases de données
- Démystifier les ordres SQL simples
 - `create table EMP ( ... );`
 - `insert into EMP values (... );`
 - `select * from EMP where ...;`
- Comprendre les choix de conception physique
- Les objectifs techniques sous Oracle
 - Comprendre l'architecture d'un serveur de BD Oracle
 - Démarrage et arrêt d'une instance et d'une base de données Oracle
 - Création d'une base de données opérationnelle
 - Gestion des fichiers d'une base de données Oracle
 - Gestion de la structure logique (tablespaces, segments, extents et blocs)
 - Gestion des utilisateurs, des privilèges et des ressources
 - Sauvegarde/restauration/réorganisation des données

Pré-requis du cours

- Autour des BD
 - Modèle relationnel
 - Langage de requêtes
 - Transactions
 - Langage de programmation et BD
- Autour des systèmes d'exploitation
 - Hiérarchie de la mémoire
 - Gestion du cache de données
 - Politique de défaut de pages

Introduction

Historique (1)

- Software Development Laboratories a été créé en 1977. En 1979, SDL change de nom en devenant Relational Software, Inc. (RSI) et introduit son produit Oracle V2 comme base de données relationnelle. La version 2 ne supportait pas les **transactions** mais implémentait les fonctionnalités SQL basiques de requête et jointure. Il n'y a jamais eu de version 1, pour des raisons de marketing, la première version a été la version 2. Celle-ci fonctionnait uniquement sur les systèmes Digital VAX/VMS.
- En 1983, RSI devient Oracle Corporation pour être plus représentative de son produit phare. La version 3 d'Oracle, entièrement ré-écrite en langage de programmation C, est publiée. Celle-ci supportait les transactions grâce aux fonctionnalités de commit et rollback. C'est aussi à partir de cette version que la plate-forme Unix est supportée.
- En 1984, la version 4 d'Oracle apparaît, supportant la cohérence en lecture (read consistency).
- Début 1985, Oracle commence à intégrer le modèle client-serveur, avec l'arrivée des réseaux au milieu des années 80. La version 5 d'Oracle supporte donc les **requêtes distribuées**.
- En 1988, Oracle met sur le marché son ERP - Oracle Financials basé sur la base de données relationnelle Oracle. Oracle version 6 supporte le PL/SQL, le **verrouillage de lignes** (row-level locking) et les **sauvegardes à chaud** (hot backups, lorsque la base de données est ouverte).

Historique (2)

- En 1992, la version 7 d'Oracle supporte les contraintes d'intégrité, les procédures stockées et les déclencheurs (triggers).
- En 1995, acquisition d'un puissant moteur multidimensionnel, commercialisé sous le nom d'Oracle Express.
- En 1997, la version 8 introduit le développement orienté objet et les applications multimédia.
- En 1999, la version 8i est publiée dans le but d'affiner ses applications avec Internet. La base de données comporte nativement une machine virtuelle Java.
- En 2001, Oracle 9i ajoute 400 nouvelles fonctionnalités et permet de lire et d'écrire des documents XML.
- À partir de la version 9i, intégration du moteur OLAP au sein d'Oracle : le moteur Oracle express est dorénavant référencé au sein de l'option Oracle OLAP. Les données multidimensionnelles sont accessibles à partir du langage SQL.
 - En 2003, la version 10g est publiée. Le g signifie « grid » ; un des atouts marketing de la 10g est en effet qu'elle supporte le « grid computing ».
 - En 2005, vers la fin novembre, une version complètement gratuite est publiée, la « Oracle Database 10g Express Edition ».
 - En 2007, rachat d'Hyperion Solutions et sortie de la version 11g.

Administration des bases de données ?

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

La face cachée des bases de données

Tâches de l'administrateur de BD

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

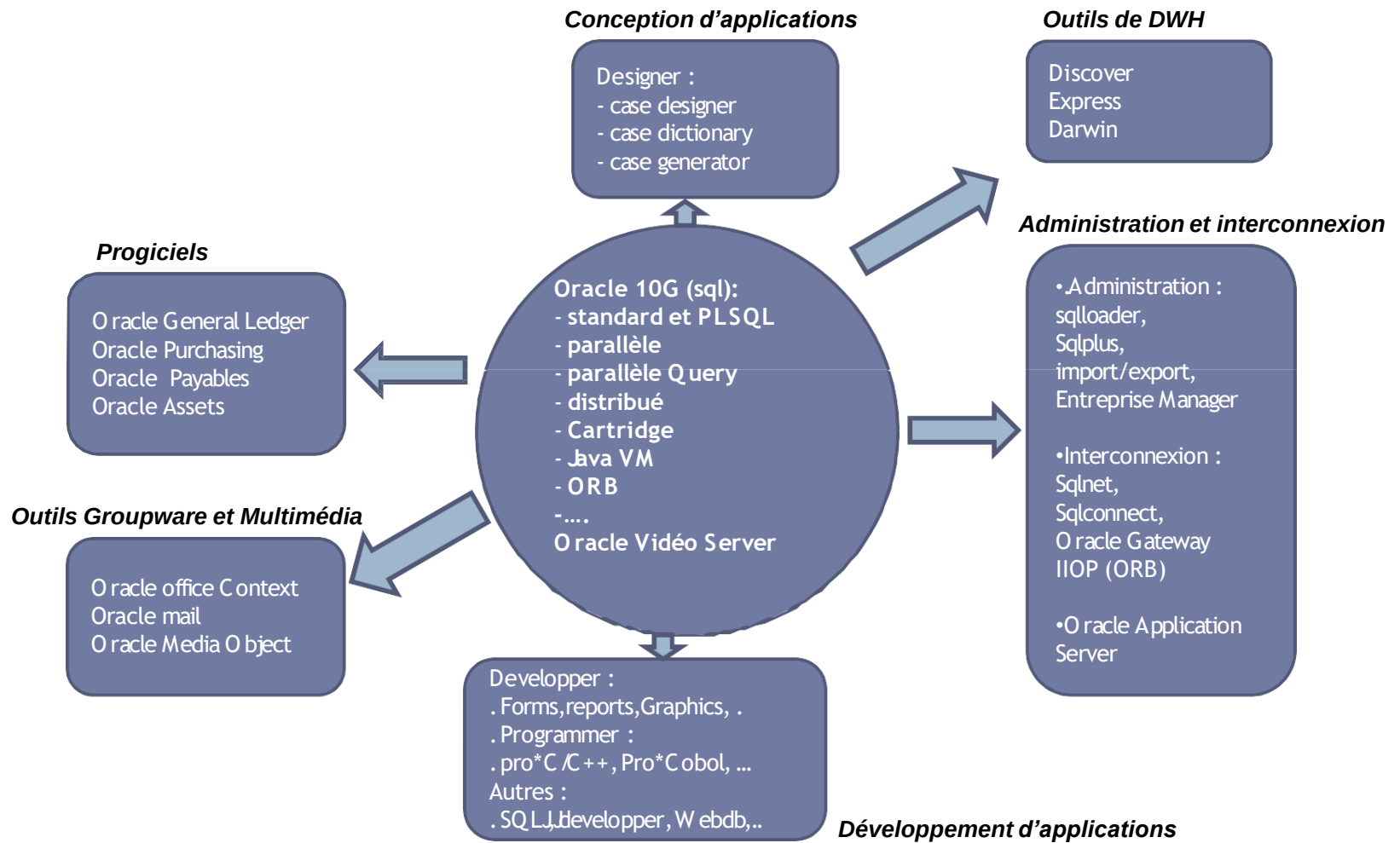
Rôles d'un DBA

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

Les outils d'administration

- Dépend du système d'exploitation sur lequel Oracle est installé
- Tendence actuelle : automatiser les tâches d'administration
 - avec des outils graphiques
 - des assistants
- Identification des applications d'administrations d' *Oracle Enterprise Manager*(O EM)
- Objectif :
 - Utilisation
 - des composants d'O EM
 - du mode ligne de commande

L'offre d'Oracle



Architecture d'Oracle

Architecture d'Oracle

Composants de l'architecture d'Oracle

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

Architecture d'Oracle

1. Les clients et les serveurs d'Oracle

1 serveur <--> plusieurs clients
Oracle Oracle

- Utilisateurs
 - Connexion directe à l'hôte
 - avec telnet + sql*plus
 - Connexion client/serveur ou deux tiers
 - avec les outils OEM : Oracle Enterprise Manager (SQL Worksheet)
 - ou avec une application développée en Client/Serveur (avec Developer 2000)
- Administrateurs
 - chargés de l'entretien du serveur Oracle

Architecture d'Oracle

1. Les clients et les serveurs d'Oracle

Connexion à une base de données

- via SQL*Plus ou application tierce => lancement d'un processus utilisateur sur le poste du client (qui peut être le serveur)
- lors de la connexion
(login/passwd/service BD)
=> lancement d'un processus serveur sur le serveur
- communication
 - inter-processus si les 2 processus s'exécutent sur le même poste
 - via un logiciel de réseau sinon
- Session: connexion spécifique entre un utilisateur et un serveur prêt à l'emploi (un serveur démarré).

Architecture d'Oracle

1. Les clients et les serveurs d'Oracle

Processus Utilisateur

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

Architecture d'Oracle

1. Les clients et les serveurs d'Oracle

Processus Serveur

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

Architecture d'Oracle

1. Les clients et les serveurs d'Oracle

Types d'architecture

- Architecture à serveur dédié

- Architecture

- la plus simple
- la plus épandue
- Inadapté si beaucoup de clients connectés de façon simultanée



- Architecture à serveur partagé

- Serveur partagé ou Multi-Thread Server (MTS)



- Gestion d'une file de requêtes des processus clients et de réponses à retourner
- Dispatcher + Listener

Architecture d'Oracle

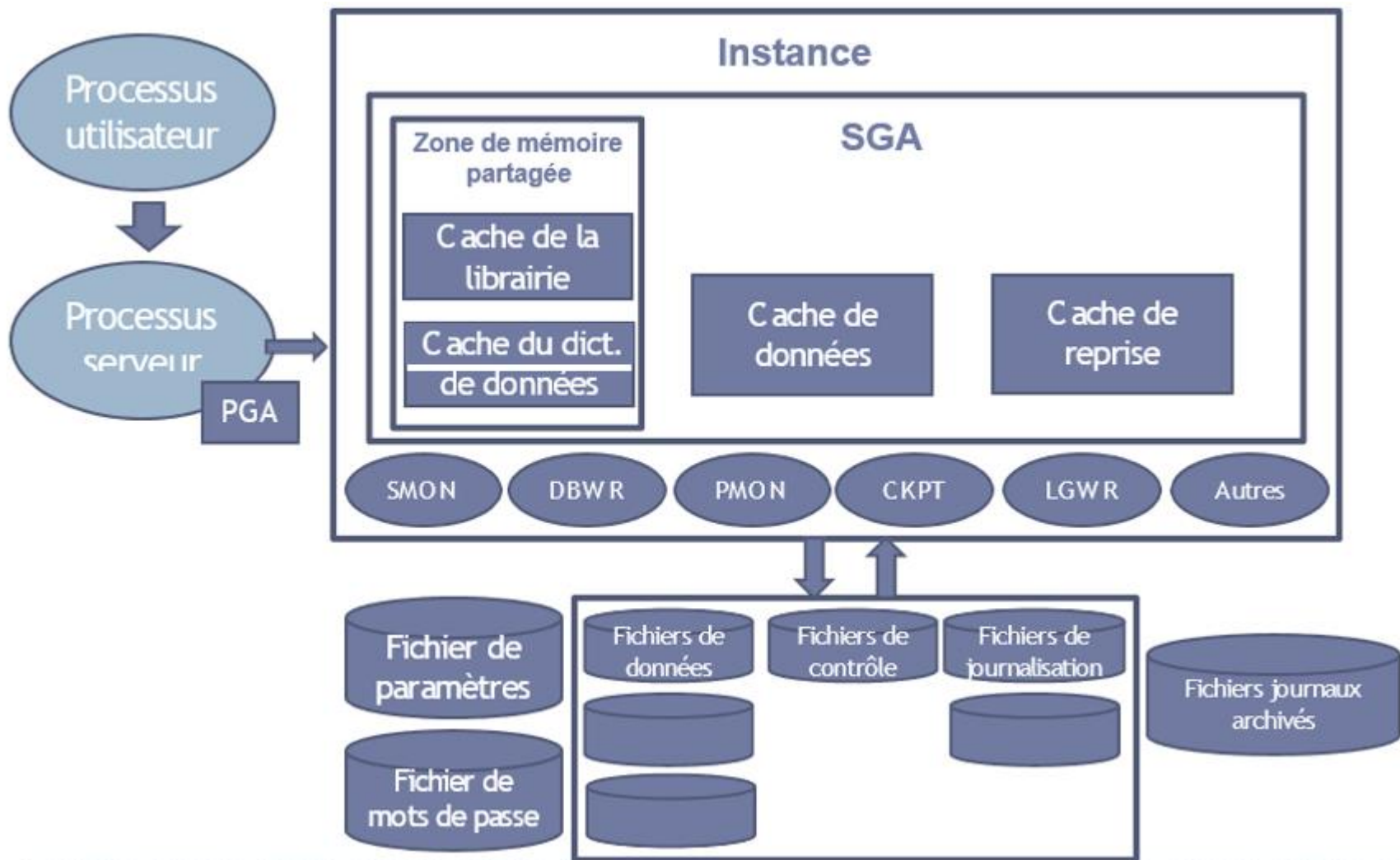
2. Serveur Oracle

Serveur Oracle = instance Oracle + base de données Oracle

- Le serveur Oracle
 - Il s'agit du système installé sur une machine qui va permettre la gestion de toutes les bases de données disponibles sur la machine.
- Instance Oracle
 - C'est un moyen pour accéder à une base de données Oracle
 - Ouvre une **unique** base de données
- Instance Oracle = SGA + des processus en arrière plan pour gérer la base + un processus serveur

Architecture d'Oracle

2. Serveur Oracle



Architecture d'Oracle

2. Serveur Oracle

Instance Oracle : Structure de la mémoire d'Oracle

- Principales composantes sont :
 - la SGA (System Global Area)
 - la PGA (Program Global Area)
 - la zone de Tri (Sort Area Size)
- Gérer essentiellement selon le principe dit LRU (*Last Recent Used*)
- La taille de ces zones est déterminée grâce à des paramètres d'initialisation
- Un sous-dimensionnement peut entraîner des pertes importantes de performances

Architecture d'Oracle

2. Serveur Oracle

Instance Oracle : la SGA

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

Architecture d'Oracle

2. Serveur Oracle

SGA : Le cache de données (Data buffer cache)

- Objectif :
 - minimiser les coûts d'entrées/sorties entre la mémoire centrale et la mémoire secondaire
- Zone de chargement et de mise à jour en mémoire des blocs de données (blocs les plus récemment utilisés)
- Ces blocs proviennent des fichiers de données
- Les blocs concernés peuvent être :
 - des blocs de tables et clusters
 - des blocs d'index (B-tree, Bitmap, Reverse Key, ...)
 - des blocs des rollback segments
- Le buffer de données est géré selon le mécanisme LRU (Last Recent Used) :
 - Seul les blocs les plus récemment utilisés sont maintenus en mémoire. Les blocs les moins récemment utilisés sont éjectés du Buffer de données, ceux modifiés écrits dans les fichiers de données

Architecture d'Oracle

- 2. Serveur Oracle
- SGA : Le cache de données (Data buffer cache)
- Paramètres d'initialisation influençant sa taille :
 - DB_BLOCK_BUFFERS nombre de blocks
 - DB_CACHE_SIZE : nombre de blocs du buffer de données par défaut.
 - DB_BLOCK_SIZE : taille du bloc par défaut
 - DB_16K_CACHE_SIZE : nombre de blocs du buffer de données de blocs de 16K
 - DB_2K_CACHE_SIZE : nombre de blocs du buffer de données de blocs de 2K
 - DB_32K_CACHE_SIZE : nombre de blocs du buffer de données de blocs de 32K
 - DB_4K_CACHE_SIZE : nombre de blocs du buffer de données de blocs de 4K
 - DB_8K_CACHE_SIZE : nombre de blocs du buffer de données de blocs de 8K
- Oracle peut gérer plusieurs buffers de données avec des tailles de blocs différents si DB_CACHE_SIZE et au moins un DB_xK_CACHE_SIZE sont posés
- DB_CACHE_SIZE et DB_xK_CACHE_SIZE sont modifiables dynamiquement via ALTER SYSTEM

Architecture d'Oracle

2. Serveur Oracle

SGA : Le cache de données (Data buffer cache)

- Les Buffers Pools multiples (3 pools)
 - Les paramètres d'initialisations BUFFER_POOL_KEEP et BUFFER_POOL_RECYCLE permettent de définir deux buffers pools supplémentaires. Le premier étant celui par défaut
 - La zone définie par BUFFER_POOL_KEEP permet de définir un espace ou fixer les objets en mémoire
 - La zone définie par BUFFER_POOL_RECYCLE permet de définir un espace ou fixer les objets qui s'y trouvent libérés aussitôt qu'on ne les utilise plus
- **NOTE:** Seule le buffer de données par défaut définit par le paramètre d'initialisation DB_CACHE_SIZE peut être géré de la sorte.

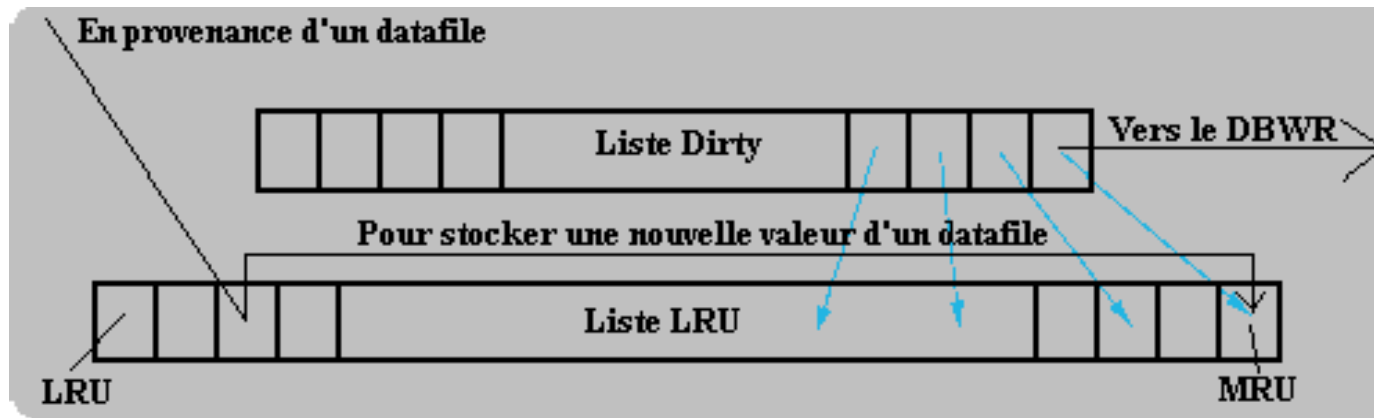
Architecture d'Oracle

- 2. Serveur Oracle
- SGA : Le cache de données (Data buffer cache)
- Les buffers de données sont organisés en deux listes :
 - La Dirty List
 - contient la liste des blocs en modification. La connaissance de cette Liste permet d'accélérer l'écriture des blocs modifiés dans les fichiers de données
 - référence l'ensemble des **tampons notés dirty** de la liste LRU
 - La LRU list
 - contient une série d'éléments (tampons) qui sont triés dans un ordre bien précis; des plus anciennement utilisés (Least Recently Used) aux plus récemment utilisés (Most recently Used :MRU).
 - Les éléments de cette liste (les tampons) peuvent prendre 3 états :
 - l'état free (libre) : il faut interpréter cela dans le sens où ce tampon n'est pas référencé par la liste dirty et donc les données contenues ne nécessitent pas d'être mises à jour dans un datafile,
 - l'état pinned : qui indique que des données de ce tampon sont en cours d'utilisation pour un traitement quelconque,
 - l'état dirty : une donnée de ce tampon est plus récente que son homologue sur le datafile requis, en conséquence de quoi il faudra mettre à jour ce tampon sur le datafile.

Architecture d'Oracle

2. Serveur Oracle

- SGA : Le cache de données (Data buffer cache)



Fonctionnement du cache de données

Architecture d'Oracle

2. Serveur Oracle

SGA : Le cache de données (Data buffer cache)

- les performances sont bonnes si le ratio R est ≥ 60 ou 70%

$$R = 1 - \frac{\text{Physical read}}{\text{db block gets} + \text{consistent gets}}$$

- **Physical read** : nombre de lecture sur disque
- **db block gets + consistent gets** : nombre total de lecture sur disque ou en mémoire.
- **Exemple** : La table v\$sysstat contient les statistiques utiles :
 - SELECT name, value FROM v\$sysstat
 - WHERE name IN ('db block gets', 'consistent gets', 'physical reads');
- **Résultats** :

Name	Value
db block gets	85792
consistent gets	278888
physical reads	13182

Architecture d'Oracle

2. Serveur Oracle

SGA : La shared Pool (ou Zone de partage des ordres SQL)

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

Architecture d'Oracle

2. Serveur Oracle

SGA : La shared Pool (ou Zone de partage des ordres SQL)

- Se compose des données suivantes :
 - les plans d'exécution et les résultats d'analyse des ordres venant des processus utilisateurs
 - les procédures stockées (PL/SQL)
 - les requêtes récursives (requêtes sur le dictionnaire)
- Condition de partage
 - le plan d'exécution et les résultats d'analyse sont encore dans le buffer
 - les objets composants la requête n'ont pas évolués
 - le texte de la requête est identique au caractère prêt y compris le code PL/SQL

- Exemple:

<code>SELECT * FROM DEPT</code>	est différent de	<code>Select * FROM DEPT</code>
<code>SELECT * FROM . DEPT</code>	différent de	<code>SELECT * FROM .. DEPT</code>

- **NOTE** : le dba peut nettoyer le buffer via la commande `Alter System Flush Shared Pool`

Architecture d'Oracle

2. Serveur Oracle

SGA : La shared Pool (ou Zone de partage des ordres SQL)

- **Informations sur les requêtes**

- v\$sqlarea (texte des requêtes)
- v\$sqlibrarycache (tuning de requêtes partagées)
- v\$rowcache (tuning du dictionnaire d'Oracle)

Architecture d'Oracle

2. Serveur Oracle

SGA : La shared Pool (ou Zone de partage des ordres SQL)

- **Optimisation du cache de la librairie**

```
SELECT sum(pins) "Executions",  
sum(reloads) "Défaut de cache",  
sum(reloads) / (sum(pins) + sum(reloads))*100 "R"  
FROM v$librarycache ;
```

- reloads : défaut de lecture dans le cache de librairie d'exécutions
- pins : nombre d'exécutions sans défaut de cache
- si $R \geq 1\%$ alors augmenter **SHARED_POOL_SIZE**

- **Optimisation du cache du dictionnaire**

```
SELECT sum(gets) "DC Gets", sum(getmisses) "DC cache get  
Misses",  
sum(getmisses) / (sum(gets)+sum(getmisses))*100 "R" FROM v$rowcache ;
```

- R doit être $\leq 10\%$ ou 15% sinon accroître **SHARED_POOL_SIZE**

Architecture d'Oracle

2. Serveur Oracle

La PGA

- La PGA (Program Global Area) est une zone de mémoire **privée**
 - Un PGA est associé à un processus (et inversement).
- Zone mémoire utilisée par un seul processus serveur
 - à la différence de la SGA qui est partagée par tous les processus serveurs

Architecture d'Oracle

2. Serveur Oracle

La PGA

- Ce PGA sert à temporiser les données que manipule le processus, toujours dans un souci d'optimisation.
 - Buffer contenant des données et des informations de contrôle pour un process serveur.
- PGA contient :
 - une zone de tri
 - des informations sur la session
 - l'état du curseur
 -

Architecture d'Oracle

2. Serveur Oracle

La PGA

- Les tables v\$sesstat, v\$statname, permettent de déterminer la taille de la PGA pour une session

```
Select ss.sid, ss.value, sn.name  
FROM v$sesstat ss, v$statname sn, v$session se WHERE  
ss.statistic#=sn.statistic#  
and sn.name in ('session pga memory')  
and se.sid=ss.sid and type != 'BACKGROUND';
```

Architecture d'Oracle

2. Serveur Oracle

La PGA

- La taille maximum de la PGA est influencée en plus par les paramètres d'initialisations suivants :
 - `sort_area_size`
 - `hash_area_size`
 - `bitmap_merge_area_size` et `create_bitmap_area_size`
- D'autres paramètres influence aussi la taille de la PGA d'une session
 - `OPEN_LINKS` : nombre de databases link ouverts
 - `DB_FILES` : nombre de fichiers de données pouvant être ouverts
- En mode serveur dédié il est difficile de gérer l'allocation des paramètres *_area_size.
- Depuis la 9i le DBA peut fixer sa PGA maximale grâce au paramètre :
 - `PGA_AGGREGATE_TARGET`

Architecture d'Oracle

2. Serveur Oracle

PGA : la zone de Tri (Sort Area Size)

- Une zone de tri est associée à un Serveur (dédié ou non) pour traiter des ordres nécessitant des tris (Group by, Order by, Join, ...)
- la taille de la zone de tri est déterminée par le paramètre `SORT_AREA_SIZE` (en bytes)
 - Par défaut cette taille est de 65000 bytes
 - Si cette zone est pleine un Segment temporaire est généré
- `SORT_AREA_RETAINED_SIZE` (exprimée en byte, 0 min, Sort_area_size par défaut et max) : espace à ne pas libérer en cas d'écriture dans le segment temporaire
- tuning de la zone de tri ; table `v$sysstat`

```
SELECT name, value FROM v$sysstat
WHERE name in ('sorts (memory)', 'sorts (disk)');
```
- NOTE : Si le nombre de tris sur disque croît, augmenter `Sort_area_size`. Mais attention au Swapping de l'OS.

Architecture d'Oracle

2. Serveur Oracle

Les process autour d'Oracle

- Pour que le système puisse fonctionner, il faut que des processus (des programmes) se chargent de manipuler toutes ces données (ainsi que celles stockées sur les fichiers).
- Deux classes de process autour d'Oracle
 - Les process utilisateurs (liés à l'exécution d'un outil, d'un programme d'application, ...)
 - Les process Oracle
- Process Oracle
 - Les process tâches de fond (SMON, PMON, LGWR, DBWR, CKPT, ARCH, RECO, ...)
 - Les process serveurs
 - autres process
- **Note** : Certains de ces process sont facultatifs (ARCH, CKPT)

Architecture d'Oracle

2. Serveur Oracle

Les process utilisateurs

- Ces processus sont créés lors de toutes connexions de la part des utilisateurs.
- **Attention** ces processus ne sont pas obligatoirement sur la machine locale : ils n'utilisent donc pas directement des ressources de la machine. (ils n'appartient pas au serveur Oracle)
 - Si le serveur fonctionne en mode multi-thread les processus utilisateurs dialoguent avec les processus **D**ispatchers.

Architecture d'Oracle

2. Serveur Oracle

Les Processus Serveurs

- Un process serveur peut être dédié ou non
- Il est aussi appelé **shadow process**
- Son rôle consiste :
 - à assurer la communication directe ou indirecte avec les process utilisateurs
 - à analyser et exécuter les requêtes
 - à lire les blocs de données dans les fichiers de données
 - à restituer directement ou indirectement le résultat au process utilisateur
 - à déplacer les blocs modifiés dans la DIRTY LIST

Architecture d'Oracle

2. Serveur Oracle

Instance Oracle : Processus en arrière plan

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

Architecture d'Oracle

2. Serveur Oracle

Processus en arrière plan : DBWR

- (DBWR) Le process DataBase WRiter : c'est processus d'écriture des blocs de données)
- Ecrit les blocs de données modifiés de la SGA vers les fichiers de données
 - est optimisé pour minimiser les accès disques
- Objectifs :
 - Limiter les coûts d'entrées/sorties en "retardant" (de façon transparente à l'utilisateur) l'écriture des blocs modifiés

Architecture d'Oracle

2. Serveur Oracle

Processus en arrière plan : D B W R

- Quand s'active D B W R ?
 - Lors d'un CHECKPOINT ; point de synchronisation quand LGWR ou CKPT l'avertit.
 - Pour libérer de la place dans le Buffer de données à la demande du process serveur
 - sur un TIMEOUT (toute les 3 s).
- Il peut avoir un ou plusieurs D B W R (DBW0...DBWn)
(DB_WRITER_PROCESSES=n allant de 0 à 9)

Architecture d'Oracle

2. Serveur Oracle

Processus en arrière plan : LGWR

- (LGWR) Le Process Log Writer : C'est le processus d'écriture du fichier de reprise
- Objectifs
 - Enregistrer toutes les modifications apportées aux données afin d'assurer la reprise après panne
- Trace le contenu du buffer REDO LOG dans les fichiers REDO LOG
 - Contrôle la mise à jour des données des Redo log files à partir du tampon Redo log contenu dans le SGA.
- En cas de checkpoint (CKPT absent) LGWR réveille DBWR et modifie l'entête des fichiers de données et de contrôles

Architecture d'Oracle

2. Serveur Oracle

Processus en arrière plan : LGWR

- Quand s'active LGWR ?
 - Si un COMMIT à été passé ; une transaction est validée
 - Sur un timeout toute les 3 secondes
 - Si le buffer REDO LOG est plein au 1/3
 - Quand DBWR libère des blocs de données du buffer de données (en cas de TIMEOUT ou de checkpoint)
- Optimisation
 - activer le process CHECKPOINT (CKPT)

Architecture d'Oracle

2. Serveur Oracle

Processus en arrière plan : CKPT

- (CKPT) Le Process Check Point : point de synchronisation
- Ce processus est chargé de coordonner la mise à jour des fichiers de données et de contrôle, ce à partir des informations contenues dans le SGA.
 - Cette mise à jour se fait à des moments précis appelés check points.
- Ce processus est cependant optionnel.
 - Dans le cas où il n'existe pas, c'est le processus LGWR qui se charge de réaliser la sauvegarde.
- Intérêt d'un checkpoint
 - Permet de forcer l'écriture dans les fichiers de données des blocs de données restant en mémoire car fréquemment modifiés (Mécanisme LRU)
 - Permet d'accélérer le recouvrement de données : les données avant le checkpoint dans le fichier Redo log ne seront plus appliquées au fichiers de données car elles y sont déjà présentes.

Architecture d'Oracle

2. Serveur Oracle

Processus en arrière plan : CKPT

- Un CKPT s'obtient en fixant le paramètre : `CHECKPOINT_PROCESS=TRUE`
- S'il est présent informe DBWR qu'un CHECKPOINT est intervenu
- Note le Checkpoint dans l'entête des fichiers de données et de contrôles
- Un checkpoint intervient :
 - Si le `TIMEOUT` a été atteint (`LOG_CHECKPOINT_TIMEOUT : 0` par défaut)
 - Si la fin d'un groupe de fichiers Redo log est atteint
 - Si une taille correspondant à `LOG_CHECKPOINT_INTERVAL` (en blocs OS) a été écrite dans le fichier Redo Log
 - Si les commandes suivantes sont passées :
 - `ALTER SYSTEM CHECKPOINT` (pas de changement de REDO LOG)
 - `ALTER SYSTEM SWITCH LOGFILE`
- Optimisation
 - Favoriser le recouvrement ou les performances en dimensionnant mieux `LOG_CHECKPOINT_INTERVAL` et `LOG_CHECKPOINT_TIMEOUT`

Architecture d'Oracle

2. Serveur Oracle

Processus en arrière plan : SMON

- (SMON) System MONitor : Ce processus, **père** de tous les processus de l'instance.
- Il se charge d'optimiser l'utilisation de la mémoire dans le système.
- Et d'assurer la reprise du système lors de tout démarrage d'une instance.
- Il s'occupe notamment de plusieurs autres tâches.
 - **Répare** l'instance au démarrage en cas d'arrêt brutal
 - **Libère** les segments temporaires
 - Compacte les extensions libres pour les rendre contiguës (Alter Tablespace nomtablespace Coalesce)
 - Recouvre les process suspendus suite à un crash

Architecture d'Oracle

2. Serveur Oracle

Processus en arrière plan : PMON

- (PMON) Process MONitor
- Ce processus se charge notamment de libérer toutes les ressources acquises par un processus client, lorsque celui-ci se termine.
- Il est aussi chargé de surveiller les processus serveurs et les processus dispatchers :
 - si l'un d'eux s'arrêtait **anormalement**, le PMON se chargerait de libérer les ressources de ce processus et de le relancer.
- Fait le ménage en cas de disparition brutale d'un process utilisateur.
 - Supprime au niveau Oracle les process en erreur
 - Annule les transactions en cours
 - Libère les verrous
 - Et contrôle les process **dispatchers et serveurs**. S'ils ne sont plus présents, il les redémarre.

Architecture d'Oracle

2. Serveur Oracle

Processus en arrière plan : ARCH

- (ARCH)ARCHiver
- Ce processus sert à effectuer un archivage des fichiers de Redo log :
 - En effet, si vous n'avez que deux redo log files (leur taille est fixe) et que le tampon Redo log est plein, un des deux fichiers doit être écrasé pour contenir les nouvelles données.
 - Ce processus, s'il est activé, permet de palier ce problème. En effet il archivera (sauvegarde) l'ancien redolog file.
- Pour que le processus existe, l'instance doit être en **ARCHIVELOG mode**.
 - C-à-d la base doit être démarrée en mode avec Archive.
- C'est un process facultatif. Jusqu'à 10 process ARCH peuvent être activés.
 - Le paramètre `log_archive_max_processes` permet de fixer le nombre maximum.
 - Il est conseillé de garder la valeur par défaut car Oracle alloue lui-même de nouveaux process en cas de besoin.

Architecture d'Oracle

2. Serveur Oracle

Processus en arrière plan : Dispatcher (Dnnn)

- Les processus dispatchers sont optionnels.
- Ils n'existent que si le serveur Oracle est configuré en **mode multi-threads**.
- S'il existe, il sert d'**interprète** entre les processus utilisateurs et les processus serveurs.
 - Réceptionne les requêtes des process utilisateurs et les met à disposition d'un process serveur
 - Lit les résultats d'une requête et les redirige vers le process utilisateur concerné
- Au moins un dispatcher par type **de protocole réseau**
 - Permet de partager les process serveurs

Architecture d'Oracle

2. Serveur Oracle

Processus en arrière plan : (RECO)

- (RECO) RECOOverer : Ce processus, très complexe, est utilisé uniquement dans le cadre d'une **BD distribuée**.
- Termine les transactions distribuées en suspens dû à une erreur réseau ou système.
- Se réveille par intervalles réguliers pour finir ou annuler des transactions suspendues

Architecture d'Oracle

2. Serveur Oracle

Processus en arrière plan : (LCKn)

- (LCKn) LoCK : Ces processus ne sont utilisés que dans un usage du serveur en **mode parallèle**.
- Ils servent à gérer des verrous entre les diverses instances de la BD.
- Jusqu'à 10 Lock process peuvent être démarrés

Architecture d'Oracle

2. Serveur Oracle

Base de Données Oracle

- désignée par un nom (DB_NAME), souvent pris identique à celui de l'instance correspondante.
- représente les structures physiques des données
- la BD est divisée en plusieurs sous unités.
 - Certaines sont physiques (ont une réelle existence tel un datafile),
 - d'autres sont dites logiques dans le sens ou leur existence est essentiellement liée à des conventions.

Architecture d'Oracle

2. Serveur Oracle

Base de Données Oracle

- se compose de fichiers du système d'exploitation
 - fichiers de données (≥ 1), (10g ≥ 2)
 - fichiers de reprise (redo log) (≥ 2)
 - fichiers de contrôle (≥ 1)
 - fichier de paramètres
 - fichier mot de passe
 - fichiers de reprise archivés

Architecture d'Oracle

2. Serveur Oracle

BD : Structure logique

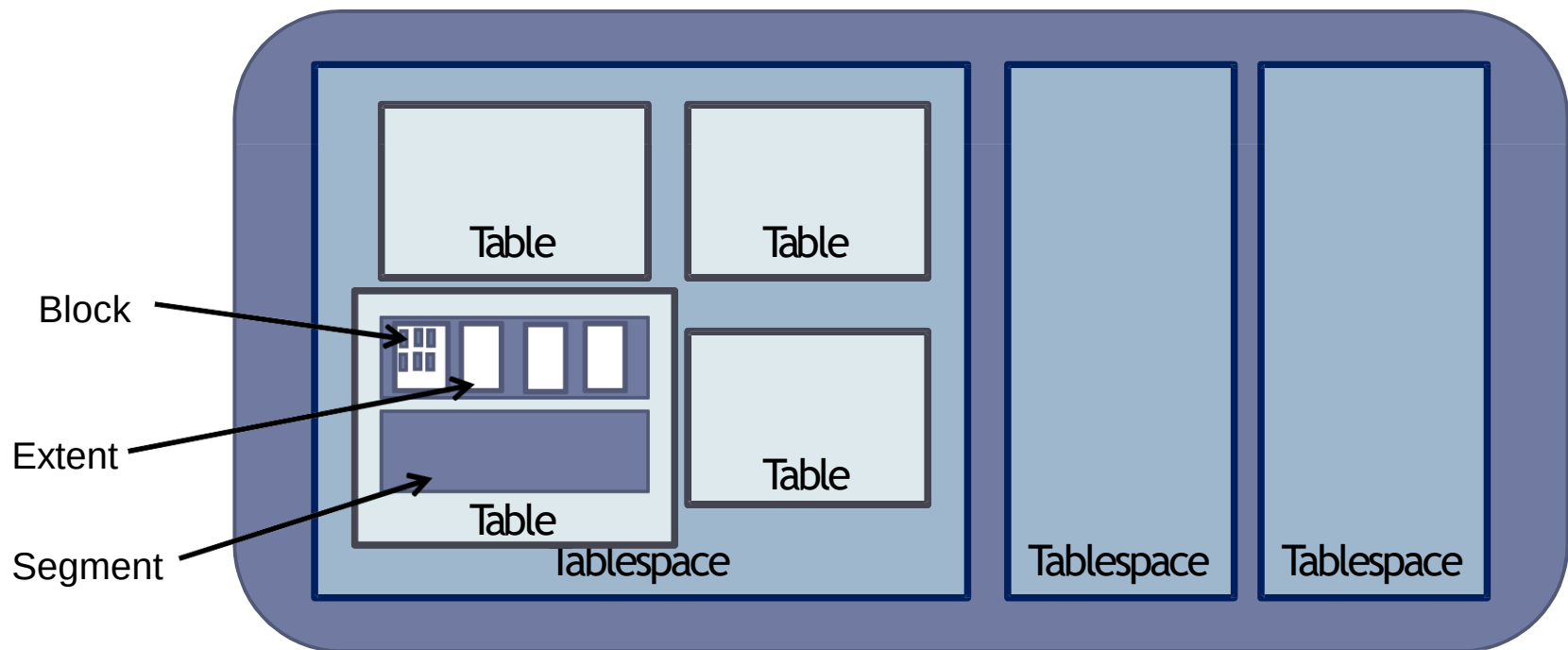
- Il existe plusieurs niveaux de structures logiques allant du **schema object** (la structure la plus importante) au **datablock** (la plus petite structure, indépendamment des données, sur laquelle on puisse apposer un contrôle). Et qui sont :
 - Les schema objects
 - Les tablespaces
 - Les tables
 - Les segments
 - Les extents
 - Les blocks

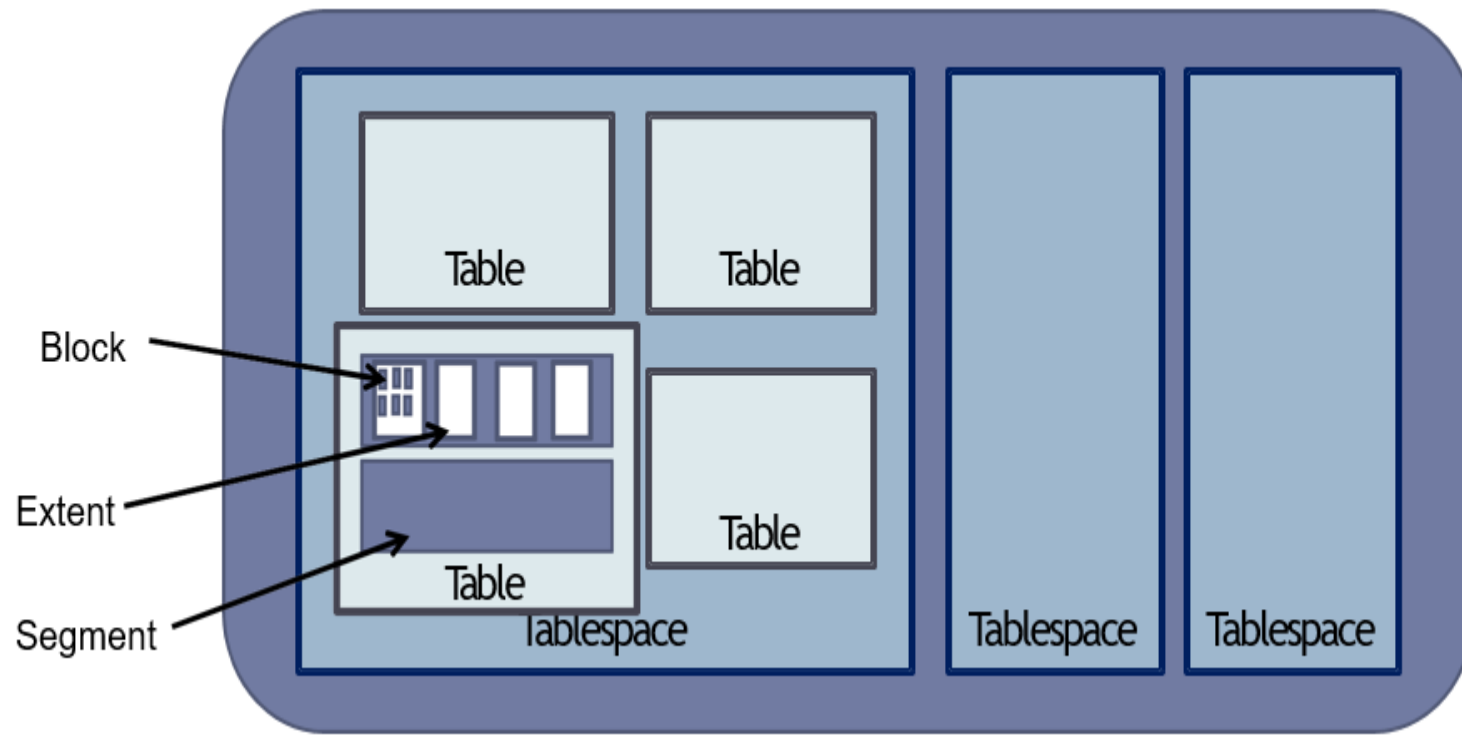
Architecture d'Oracle

2. Serveur Oracle

BD : Structure logique

- Place des différentes unités logiques existantes, les unes en rapport aux autres.





Architecture d'Oracle

2. Serveur Oracle

Structure logique : Les schema objects

- Par Schema objet on entend un moyen d'accès à la BD.
- On y trouve
 - les tables,
 - les vues,
 - les index,
 - les clusters,
 - les liens,
 - les synonymes,
 - les procédures PL/SQL
 - et les packages PL/SQL.
- L'ensemble de tous les schema objects pour un utilisateur est appelé **user's schema**.

Architecture d'Oracle

2. Serveur Oracle

Structure logique : Les schema objects

- **Les tables** : permettent directement d'accéder aux données
- **Les vues** : permettent de donner accès
 - à un sous-ensemble d'une table (par exemple, car on peut créer des vues sur n'importe quel schema object : pourquoi pas une vue sur une vue)
 - ou de plusieurs tables (jointes)
 - ces vues peuvent être utilisées pour cloisonner le champ d'action d'un utilisateur (au lieu de lui donner accès à une table complète, on ne lui accorde que le sous-ensemble requis).
- **Les index** : un index, similairement à l'index d'un ouvrage, permet à une instance du serveur d'accéder plus rapidement à des éléments.

Architecture d'Oracle

2. Serveur Oracle

Structure logique : Les schema objects

- **Les clusters** : permettent aussi un accès plus rapide aux données. L'astuce consiste à supprimer des données doubles et donc à avoir moins de données à charger à partir des disques.
- **Les liens** : permettent d'accéder des données sur une DB distante.
- **Les synonymes** : ils consistent en un nom de remplacement sur un autre schema object.
- **Les procédures et les packages** :
 - une procédure est un ensemble d'ordres PL/SQL permettant de réaliser une action sur des données.
 - Un package est un ensemble de procédures.
 - Pour qu'elles puissent être utilisées, ces unités de stockage ont besoin d'être stockées sur la BD et comme elles permettent la manipulation des données, ce sont des schema objects.

Architecture d'Oracle

2. Serveur Oracle

Structure logique : Les tablespaces

- Cette unité logique rentre dans la constitution de la BD.
- Une BD est constituée **d'au moins** un tablespace nommé SYSTEM.
 - Celui-ci contient le data dictionary (dictionnaire de données (qui contient des informations relatives au système Oracle)).
- Quand vous allez créer une nouvelle table, celle-ci sera contenue soit dans un **tablespace** existant soit dans un nouveau que vous créerez.

Architecture d'Oracle

2. Serveur Oracle

Structure logique : Les tables

- La table est l'élément constitutif d'un tablespace.
- La table est constituée de segments.

Architecture d'Oracle

2. Serveur Oracle

Structure logique : Les segments

- Un segment est constitué d'extents et rentre dans la constitution de la table.
- Une table est constituée **d'au moins deux** segments :
 - les data segments
 - et le rollback segment.
- Deux autres segments peuvent apparaître dans une table :
 - l'index segment
 - et le temporary segment.

Architecture d'Oracle

2. Serveur Oracle

Structure logique : Les segments

- **Le data segment** : rentre dans la constitution de la table, sert à stocker toutes les données (les valeurs des différentes lignes) que contient la table.
- **Le rollback segment** : ce segment stocke des données relatives aux transactions.
 - Si une transaction ne peut aboutir, la transaction doit être annulée par la commande `ROLLBACK`. Dans ce cas, on doit être capable de restituer la base dans l'état initiale ou elle était (au départ de la transaction en cours).

Architecture d'Oracle

2. Serveur Oracle

Structure logique : Les segments

- **L'index segment** : ce segment optionnel sert à stocker les informations relatives aux index créés sur la table.
 - les index servent notamment à optimiser les temps d'accès aux données.
- **Le temporary segment** : utilisé pour stocker les résultats temporaires d'une requête PL/SQL ne pouvant directement d'exécuter en mémoire.
 - Pour ce faire un segment est alloué pour les traitements intermédiaires puis d'ésalloué directement à la fin de la transaction (d'ou son nom).

Architecture d'Oracle

2. Serveur Oracle

Structure logique : Les extents

- L'extent est un ensemble de blocks consécutifs, et rentre dans la constitution du segment.
- Autrement dit, un segment est constitué de plusieurs extents.
- Tout comme il existe différents types de segments, il existe différents types d'extents.
- L'allocation des extents, constituant le segment, est dynamique.
 - C'est-à-dire que lorsque qu'un extent, contenant par exemple des données, est plein, et que des nouvelles données doivent être ajoutées, un nouvel extent est alloué et ce, s'il y a suffisamment de place sur le tablespage courant.
 - La taille du nouvel extent doit au moins être de la même taille que l'extent précédent.

Architecture d'Oracle

2. Serveur Oracle

Structure logique : Les blocks

- Le **block** est la plus petite unité logique de stockage que peut manipuler le système.
- Tout les blocs constituant la BD ont tous la même taille.
 - Cette taille peut soit être celle par défaut (fixée par le SGBD), soit être fixée par l'administrateur de la base. Dans ce dernier cas, le paramètre `DB_BLOCK_SIZE` du fichier d'initialisation doit contenir cette valeur (utilisation du fichier d'initialisation).

Architecture d'Oracle

2. Serveur Oracle

Structure logique : Les blocks

- Un bloc est principalement divisé en trois sous-parties :
 - la première contient des informations sur le bloc : c'est le header. Il indique par exemple à quel type de segment il appartient, la table et la ligne, ...
 - La seconde est une zone vide pouvant être utilisée pour ajouter de nouvelles données
 - La troisième contient les données sur les lignes de la table à laquelle le block appartient.
- **Note** : La zone vide est très importante, en contrôlant la taille de cette dernière, on peut optimiser ou non les temps d'accès à la BD.

Architecture d'Oracle

2. Serveur Oracle

BD : Le data dictionary

- Ce dictionnaire est un ensemble de tables et de vues contenant des informations sur la BD.
- En effet, comment se pourrait-il qu'un **administrateur** puisse se connecter à la base pour créer la première BD, s'il n'y avait déjà des informations stockées quelque part ?
- La BD stocke toutes données sous forme de tables. Ce dictionnaire est généré lors de la création de la BD.
- Ce dictionnaire contient par exemple :
 - des informations sur les utilisateurs,
 - sur les privilèges qu'ont ces utilisateurs,
 - sur les schema objects définis sur la BD,
 - ...

Architecture d'Oracle

2. Serveur Oracle

BD :Le data dictionary

- Vous ne pouvez en aucun cas modifier ce dictionnaire.
- Les données qui y sont stockées sont automatiquement mises à jour lors de l'utilisation de commandes DDL (data definition language) sql.
- Si vous créez une nouvelle table, les données seront automatiquement mises à jour dans la BD.
- On ne peut donc qu'accéder le dictionnaire que pour des **consultations** (SELECT).
- Afin d'en simplifier cette consultation, des vues sont définies sur les tables du dictionnaire.
- Il est déconseillé d'accéder ce dictionnaire directement par les tables.

Architecture d'Oracle

2. Serveur Oracle

BD :Le data dictionary

- Voici quelques exemples de requêtes que l'on peut être amené à réaliser.
 - Demande l'ensemble des liens utilisables sur la BD.

```
SELECT * FROM all_db_links;
```
 - Demande tous les datafiles de la BD.

```
SELECT * FROM V$DBFILE;
```
 - Demande l'ensemble des processus dispatchers.

```
SELECT * FROM V$DISPATCHER;
```
 - Encore quelques vues.

```
V$LOGFILE V$CONTROLFILE V$QUEUE ...
```
- Si vous désirez avoir une vue exhaustive de tous les éléments contenus dans ce data dictionary, il vous est possible de lancer le programme **Schema manager (BDA)**. Celui ci donne accès aux éléments de ce dictionnaire.

Architecture d'Oracle

2. Serveur Oracle

BD : Structure physique

- Ont distincte :
 - Les fichiers de données (datafile)
 - Les fichiers de reprise ou de journalisation (Redo log file)
 - Les fichiers de contrôles (Control files)

Architecture d'Oracle

2. Serveur Oracle

Structure physique : datafile

- Ces fichiers contiennent l'ensemble des données constituant la BD.
- Autrement dit l'ensemble des données accessibles par les schémas objets.
- Pour faire le lien avec les structures logiques,
 - nous pouvons dire qu'un tablespace doit l'ensemble de ses informations stockées dans un ou plusieurs datafiles.
 - Plus précisément, un schéma objet peut être stocké dans un ou plusieurs datafiles.
- Il est possible de dupliquer une même données dans plusieurs datafiles, ce pour des raisons de sécurité.
 - Si c'est le cas, alors les différents datafiles "clones" devront être dispersés sur des disques physiquement séparés (pas des partitions sur un même disque), sans quoi, cela ne servirait à rien : si le disque se crash, les deux copies seraient de toute façon détruites.

Architecture d'Oracle

2. Serveur Oracle

Structure physique : datafile

- Exemple :

```
SQL> select * from V$DBFILE;

      FILE#
-----
NAME
-----
          4
C:\ORACLEXE\ORADATA\XE\USERS.DBF
          3
C:\ORACLEXE\ORADATA\XE\SYSAUX.DBF
          2
C:\ORACLEXE\ORADATA\XE\UNDO.DBF

      FILE#
-----
NAME
-----
          1
C:\ORACLEXE\ORADATA\XE\SYSTEM.DBF

SQL>
```

Architecture d'Oracle

2. Serveur Oracle

Structure physique : Redo log file

- Ce type de fichiers contient l'ensemble des changements effectués sur la DB
 - l'ensembles des transactions sur la BD.
- L'utilité de tels fichiers est de pouvoir reconstruire au mieux la base, si une panne apparaît :
 - en effet dans ce cas, le serveur s'arrêterait sans avoir sauvegardé l'ensemble des transactions non validées (par un COMMIT) présentes en mémoire.
 - Ces fichiers permettraient alors de reconstituer (de façon consistante) au mieux la BD.

Architecture d'Oracle

2. Serveur Oracle

Structure physique : Redo log file

- Exemple :

```
SQL> select * from V$LOGFILE; GROUP# STATUS          TYPE
MEMBER

IS_

2 STALE ONLINE C:\ORACLEXE\APP\ORACLE\FLASH_RECOVERY_AREA\XE\ONLINELOG\01_MF_2_1XHPQXCQ_.LOG YES
1          ONLINE C:\ORACLEXE\APP\ORACLE\FLASH_RECOVERY_AREA\XE\ONLINELOG\01_MF_1_1XHPQVNZ_.LOG YES
GROUP# STATUS          TYPE MEMBER
IS_ SQL>
```


Architecture d'Oracle

2. Serveur Oracle

Structure physique : Control files

- Ces fichiers permettent à une instance de serveur d'utiliser les fichiers précédemment cités :
 - en effet, ils stockent les emplacements physiques des fichiers, leur nature, ...
- Ce n'est pas à vous de contrôler le contenu de ces fichiers. Le serveur se charge de le faire de manière automatique.

Architecture d'Oracle

2. Serveur Oracle

Structure physique : Redo log file

- Exemple :

```
SQL> select * from V$CONTROLFILE;
```

```
STATUS    NAME
```

```
IS_ BLOCK_SIZE      FILE_SIZE_BKLS
```

```
C:\ORACLEXE\ORADATA\XE\CONTROL.DBF  NO    16384    430
```

```
SQL>
```

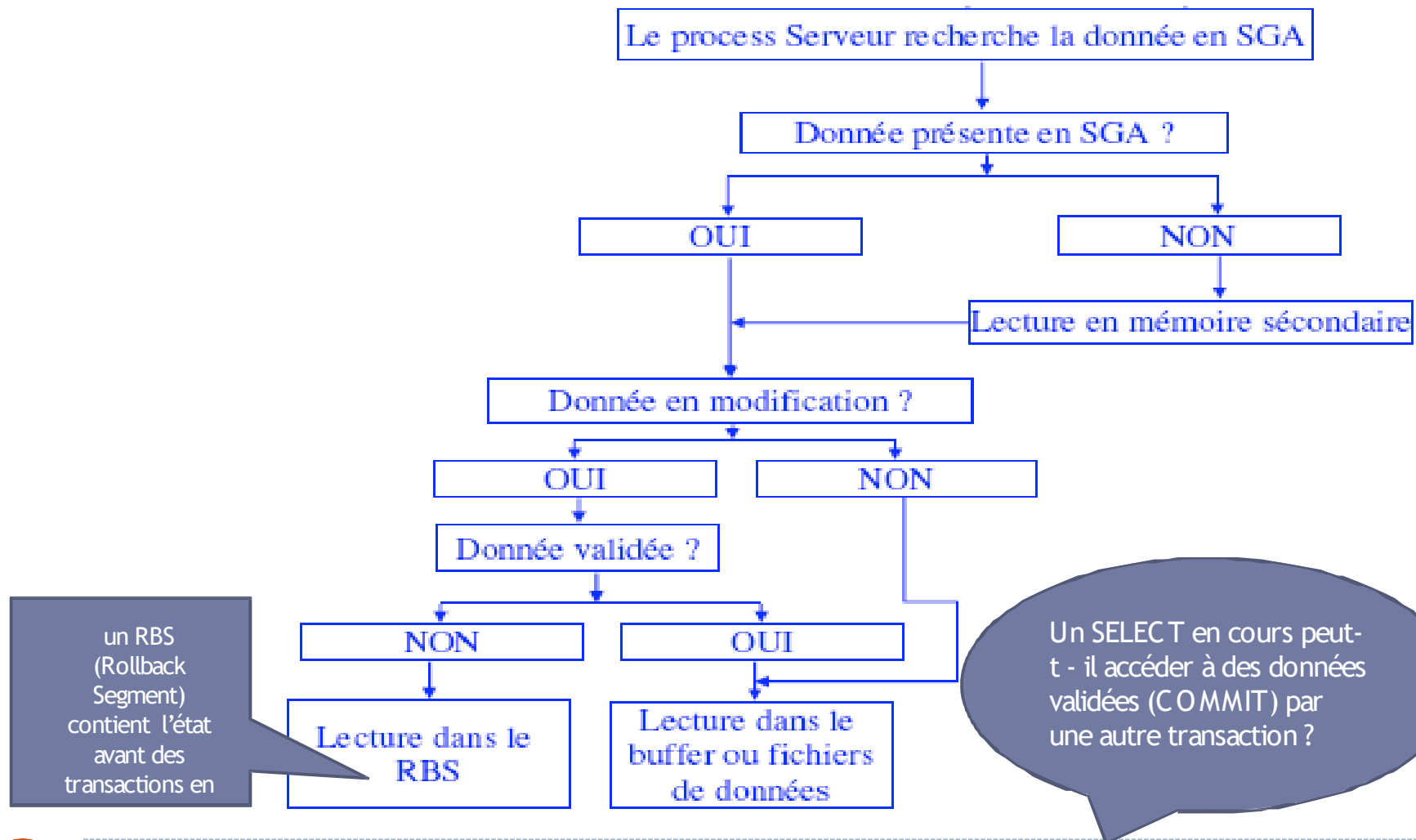
Architecture d'Oracle

3. Traitement d'une requête d'interrogation (SELECT)

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

Architecture d'Oracle

3. Traitement d'une requête d'interrogation (SELECT)



Architecture d'Oracle

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

- Similaire à une requête d'interrogation pour la phase d'analyse
- Phase d'exécution différente :
 1. • le processus serveur lit des blocs de données et de rollback à partir
 - des fichiers de données
 - du buffer cache de données
 - Si des blocs ne sont pas déjà dans le cache, copie des blocs du disque dans le cache

2. Mise à jour de verrou sur les données

3.4. et 5.

- Enregistrement des modifications
 - à apporter au rollback (image avant)
 - et aux données (nouvelles valeurs) dans le cache de reprise (redo log)
- Mêmes opérations dans le cache de données
 - Ces blocs sont marqués comme modifiés (différents des blocs stockés sur disque)

Architecture d'Oracle

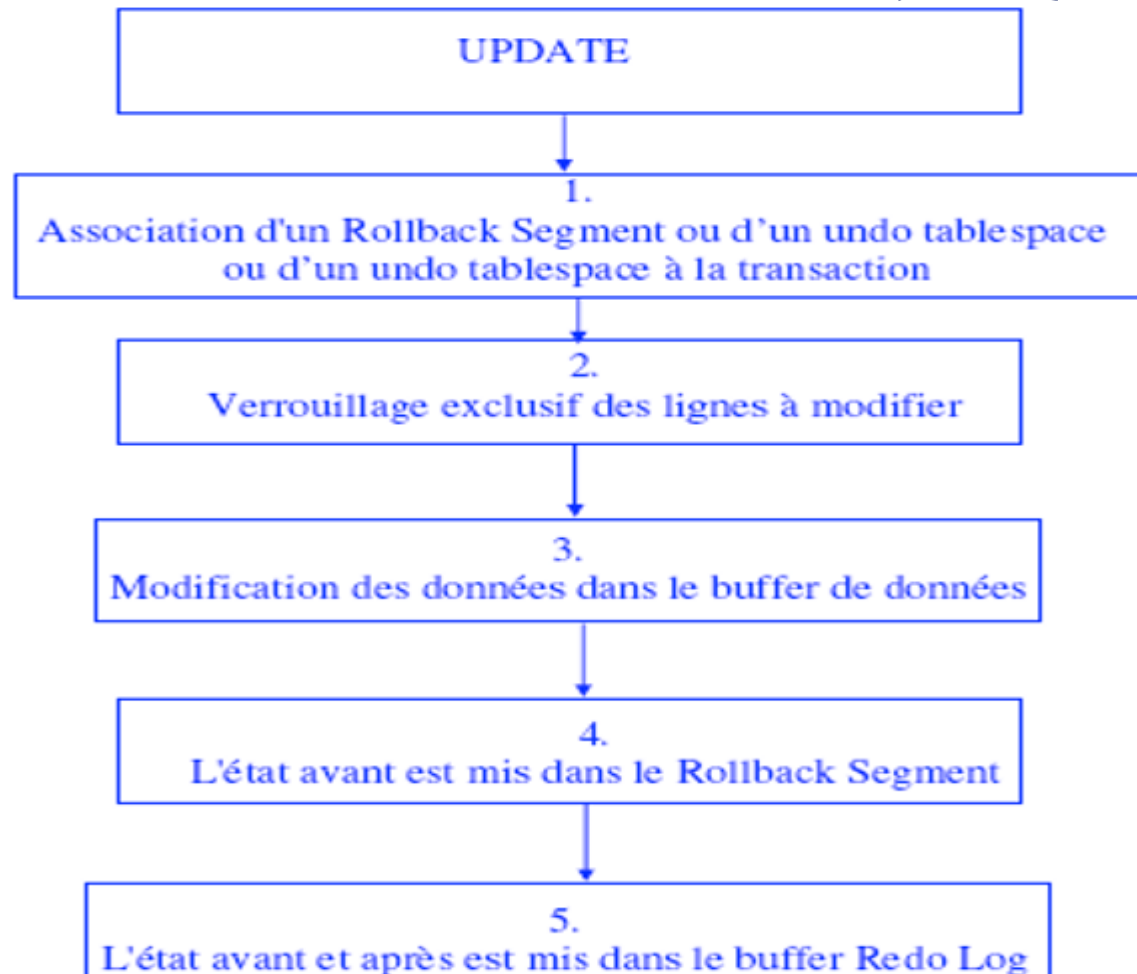
4. Traitement d'un ordre de mise à jour (DML)

Rappel : Rollback Segment

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

Architecture d'Oracle

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



Architecture d'Oracle

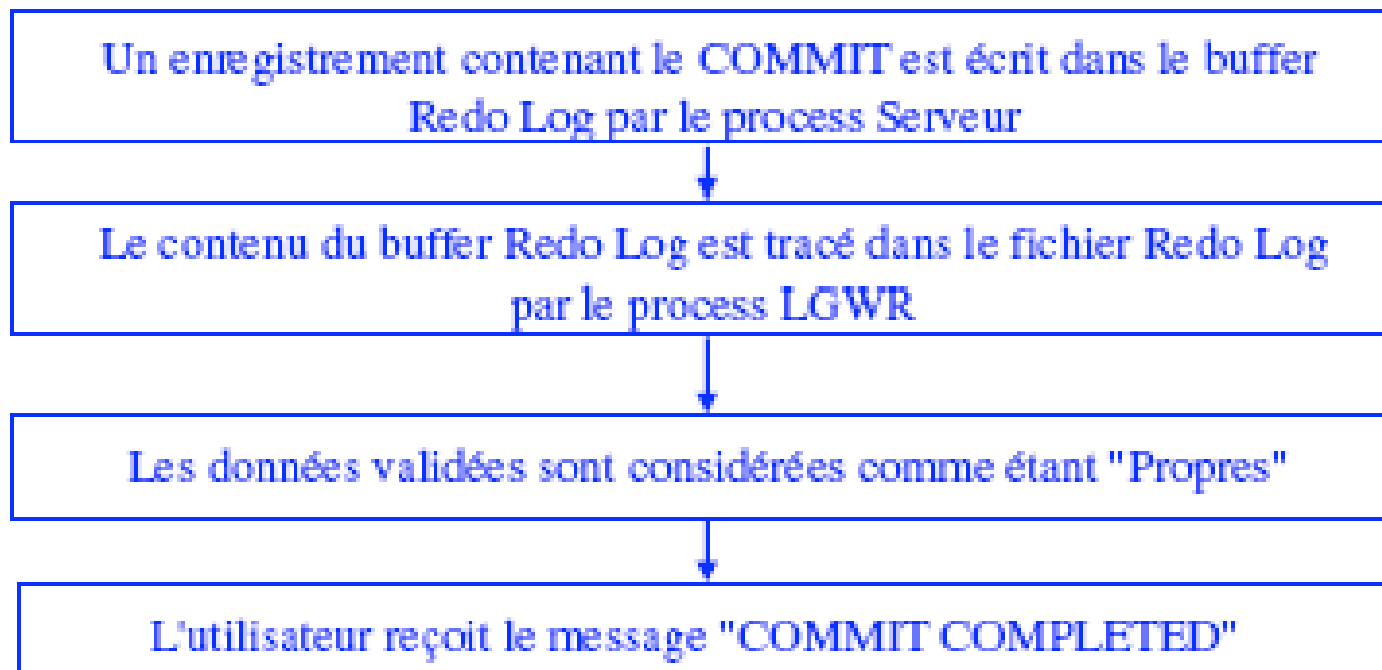
5. Traitement des validations et Invalidation

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

Architecture d'Oracle

5. Traitement des validations et Invalidation

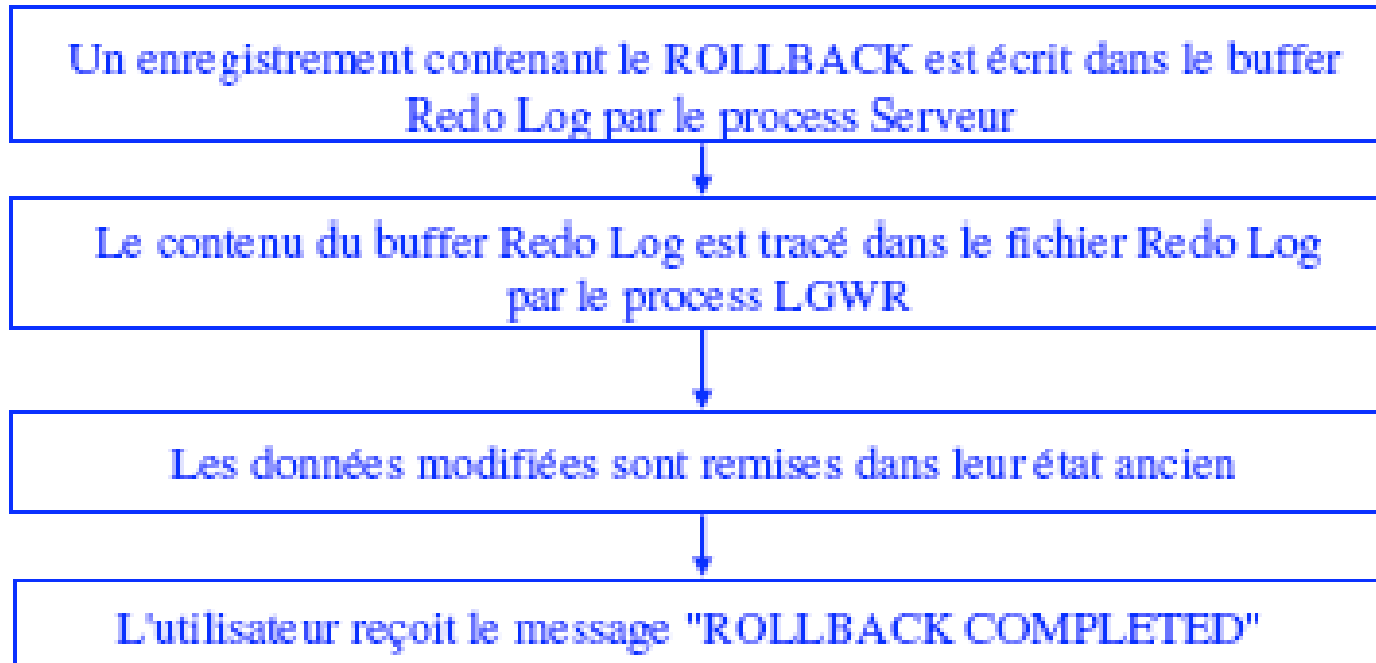
- **Validation** COMMIT



Architecture d'Oracle

5. Traitement des validations et Invalidation

- **Validation** ROLLBACK



L'espace alloué dans le ROLLBACK SEGMENT est -t- il désalloué immédiatement ?

Architecture d'Oracle

5. Traitement des validations et Invalidation

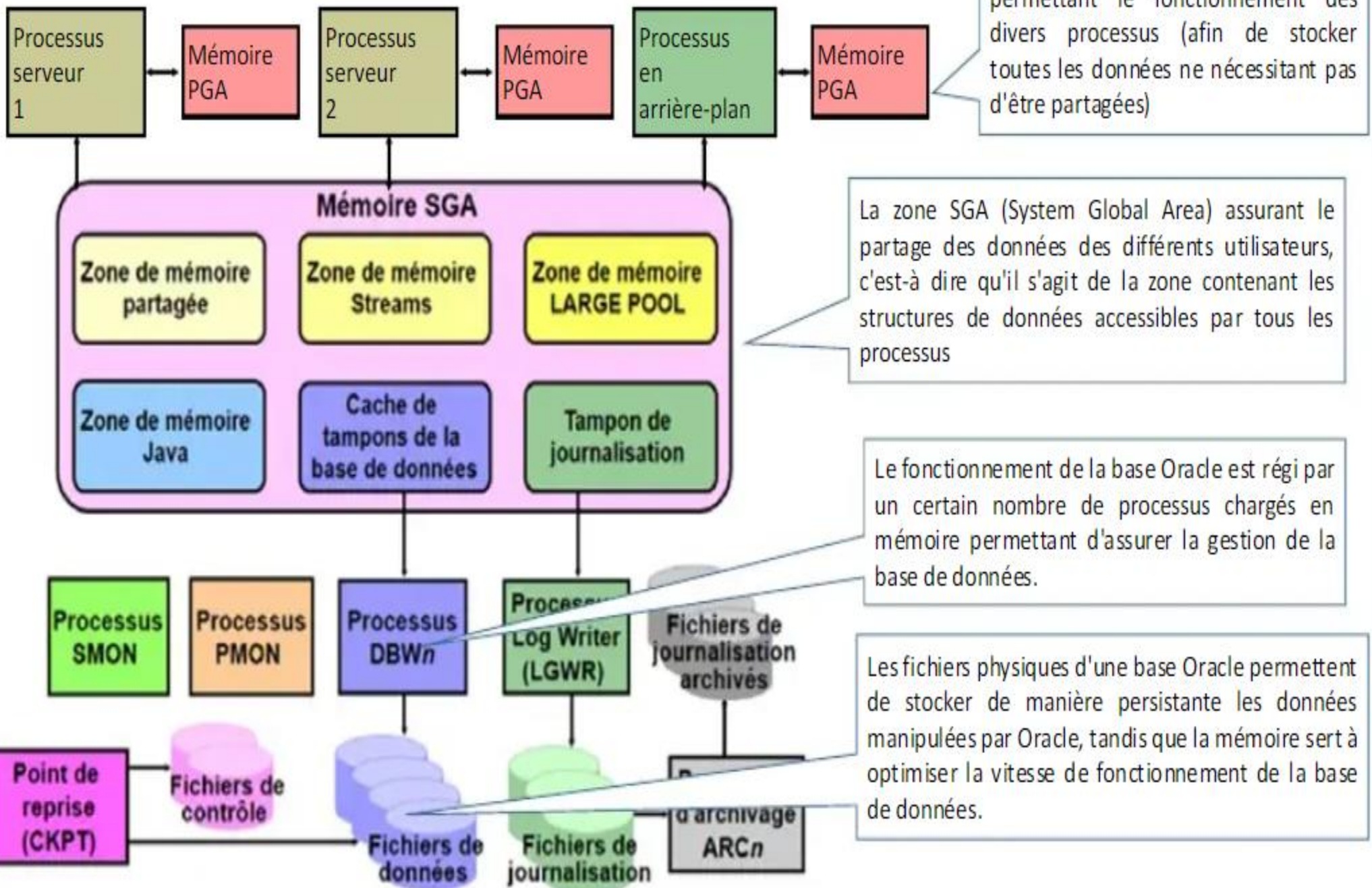
Mise à jour et validation

- **Mise à jour**
 - Transaction associée à un segment d'annulation
 - Verrou exclusif placé sur les n-uplets concernés
 - Données initiales enregistrées dans le segment d'annulation
 - Données modifiées enregistrées dans le cache de journalisation
- **Validation**
 - Écriture du cache de journalisation dans la base de données
 - Nettoyage du segment d'annulation

Architecture Oracle



Architecture SGBD Oracle



Structures mémoire Oracle

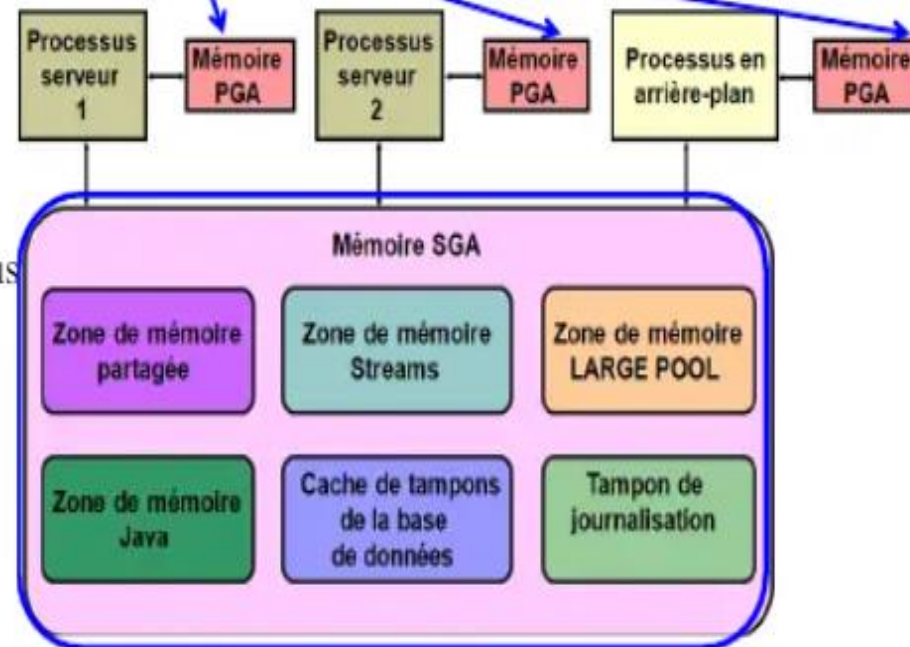
Mémoire PGA (Program Global Area)

- Propre à chaque processus serveur et processus en arrière-plan
- Alloué au lancement de chaque processus
- Zone d'exécution des processus

Structures de base de données
> Mémoire
Processus
Stockage

Mémoire SGA (System Global Area)

- Zone Partagée par tous les processus serveur et les processus en arrière-plan
- Alloué au démarrage de l'instance en mémoire principale
- Comprend des informations de contrôle et des données relatives à l'instance



Mémoire SGA

Mémoire SGA

Zone de Mémoire
Partagée

Zone de Mémoire
Streams

Zone de Mémoire
LARGE POOL

Zone de Mémoire
Java

Cache de Tampons
De la BD

Tampon de
Journalisation



Structures mémoire Oracle

Les structures mémoire de base associées à une instance Oracle sont les suivantes:

- **Mémoire SGA(System Global Area):** partagée par tous les processus serveur et les processus en arrière-plan.
- **Mémoire PGA (Program Global Area):** Propre à chaque processus serveur et processus en arrière-plan. Chaque processus dispose de sa mémoire PGA.
- La zone de mémoire SGA comprend des informations de contrôle et des données relatives à l'instance.
- La mémoire SGA contient les structures de données suivantes:
 - **Cache de tampons(buffer cache)de la base de données :** Met en mémoire cache les blocs de données extraits de la base.
 - **Tampon de journalisation (redo log buffer):** 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(redolog) physiques stockés sur le disque.
 - **Zone de mémoire partagée :** met en mémoire cache diverses structures pouvant être partagées par les utilisateurs.
 - **Zone de mémoire LARGEPOOL :** 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 d'E/S.
 - **Zone de mémoire Java:** Utilisée pour l'ensemble du code Java et des données propres à la session, dans la JVM(Java Virtual Machine).
 - **Zone de mémoire Streams:** utilisée par Oracle Streams. Il fournit les Outils pour capturer, traiter et gérer les événements de la BD



Structures mémoire Oracle

- Lorsque vous démarrez l'instance via Enterprise Manager ou SQL*Plus, la quantité de mémoire allouée pour la mémoire SGA s'affiche.
- Une mémoire PGA (Program Global Area) est une zone de mémoire contenant des données et des informations de contrôle pour chaque processus serveur.
- Un processus serveur Oracle traite les demandes d'un client.
- Chaque processus serveur dispose de sa propre mémoire PGA privée, qui est créée lors du démarrage du processus serveur.
- L'accès à la mémoire PGA est exclusivement réservé à ce processus serveur, qui peut lire le contenu de la mémoire et écrire dans celle-ci par l'intermédiaire du code Oracle.



Mémoire SGA : Sous Mémoires

Zone de Mémoire *Partagée*

Zone d'exécution partagée destinée pour les commandes SQL et PL/SQL.

Met en mémoire cache les diverses structures pouvant être partagées par les utilisateurs

Zone de Mémoire *Streams*

Utilisée par *Oracle Streams*

Oracle Streams est un produit d'*Oracle Corporation*. Il fournit les Outils pour capturer, traiter et gérer les événements de la BD

Zone de Mémoire *LARGE POOL*

Zone facultative qui fournit des **espaces importants d'allocation de mémoire à des processus volumineux**

(par exemple, les opérations de sauvegarde et de récupération Oracle et les processus serveurs d'E/S)



Mémoire SGA : Sous Mémoires

Zone de Mémoire *Java*

Utilisée pour l'ensemble du code Java et des données propres à la session, la JVM

Cache de Tampons De la BD

Met en mémoire cache les blocs de données extraits de la base

Tampon de Journalisation

Met en mémoire cache les informations de journalisation afin de récupérer l'instance. Ceci étant fait, jusqu'à ce qu'elles soient écrites dans les fichiers de journalisation (redo log) physiques stockés sur le disque



Structure de processus

- 1 Un **Processus utilisateur** est créé quand un utilisateur lance une application cliente

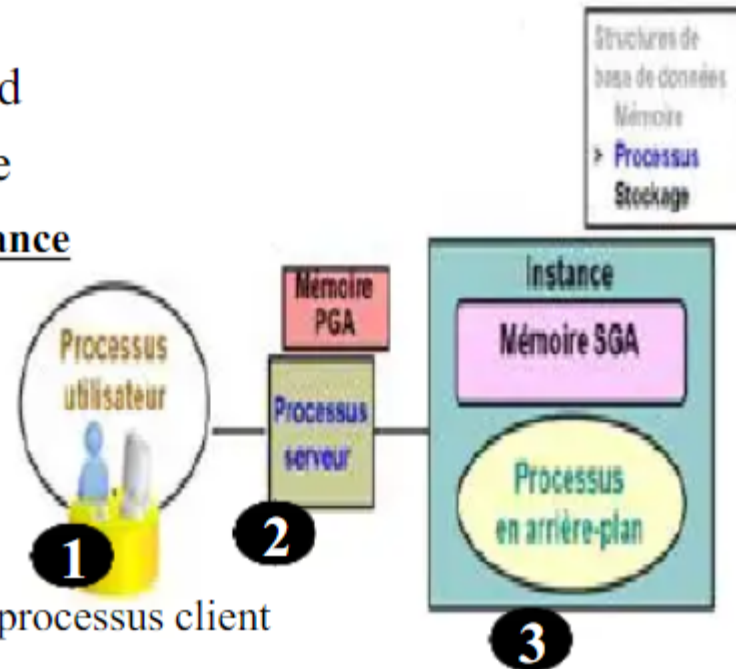
- Une **connexion** va être créée avec l'instance Oracle, l'utilisateur va ouvrir une session

- 2 Un **Processus serveur** va analyser et exécuter les requêtes et va retourner les données

- Mode dédié : un processus serveur pour un processus client
- Mode partagé : les clients partagent un groupe de processus serveurs

- 3 Des **Processus en arrière-plan** qui vont exécuter de façon asynchrone des opérations d'E/S et de contrôle d'autres processus ORACLE

- Ces Processus sont démarrés en même temps qu'une instance Oracle



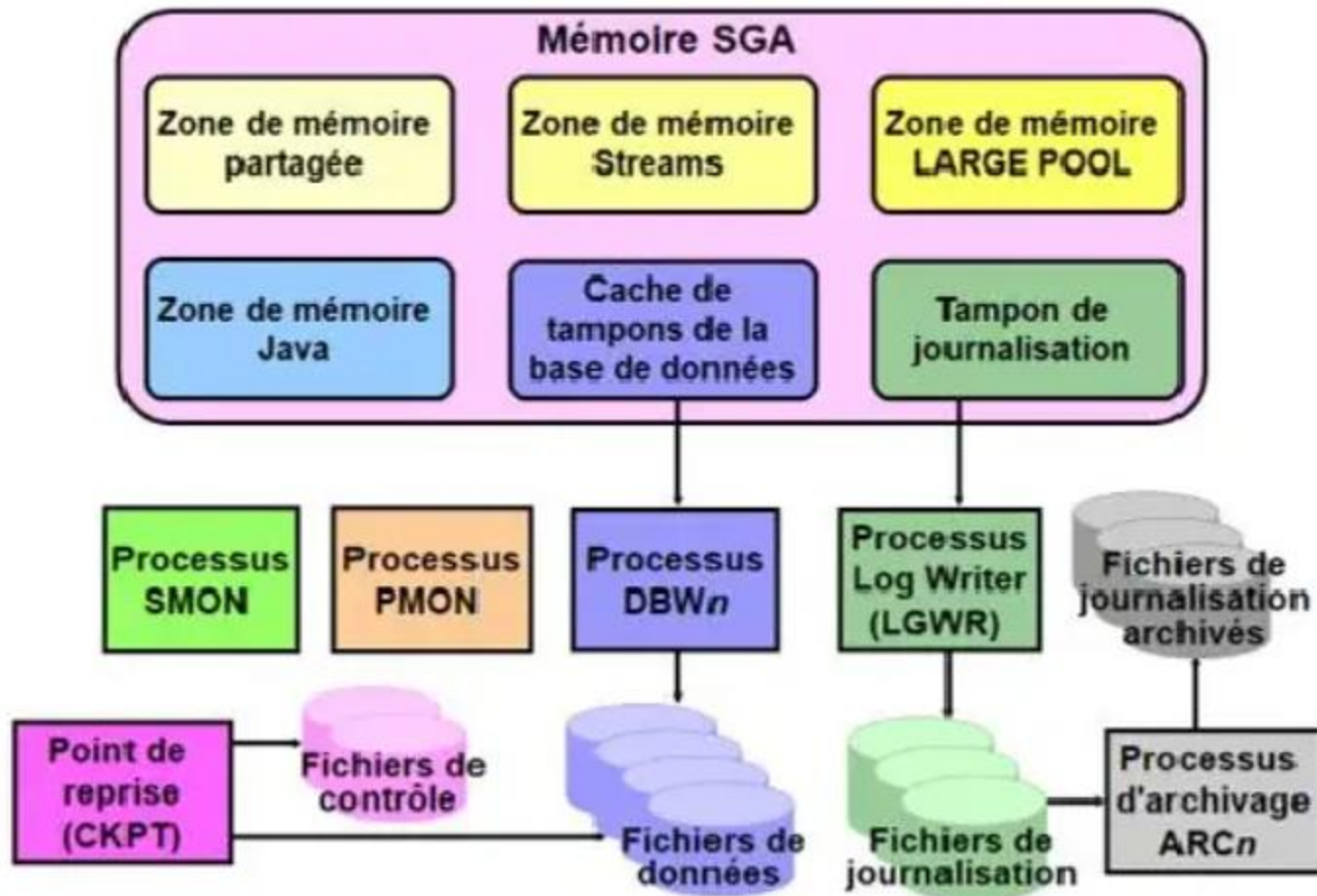
Structures de processus

- Lorsque vous appelez une application ou un outil Oracle tel qu'Enterprise Manager, le serveur Oracle crée un processus serveur afin d'exécuter les commandes émises par l'application.
- Le serveur Oracle crée également pour une instance un ensemble de processus en arrière-plan qui interagissent les uns avec les autres d'une part, et avec le système d'exploitation d'autre part, pour gérer les structures mémoire, effectuer des opérations d'E/S asynchrones afin d'écrire des données sur le disque, ainsi que pour réaliser d'autres tâches nécessaires.
- Les processus en arrière-plan qui sont disponibles dépendent des fonctionnalités utilisées dans la base de données



Gestion des instances Oracle

Gestion des instances Oracle



Gestion des instances Oracle

Un serveur de base de données Oracle est constitué d'une base de données Oracle et d'une instance Oracle. Une instance Oracle est composée de structures mémoire, appelées mémoire SGA (System Global Area), et de processus en arrière-plan qui gèrent la plupart des opérations qui ont lieu "en coulisses" lors de l'exécution d'une instance. Les processus en arrière-plan les plus courants sont les suivants:

- **Processus SMON(System Monitor):**effectue la récupération après panne lorsque l'instance est démarrée suite à une défaillance.
- **Processus PMON(Process Monitor):**procède au nettoyage des processus utilisateur en cas d'échec.
- **Processus Database Writer (DBWn):**Écrit les blocs modifiés du cache de tampons(buffer cache) de la base de données vers les fichiers de données présents sur le disque.
- **Point de reprise(CKPT):**Met à jour l'ensemble des fichiers de données et de contrôle de la base de donnée sa fin d'indiquer le point de reprise (checkpoint)le plus récent.
- **Processus LGWR (Log Writer):**Écrit les entrées de journalisation sur le disque.
- **Processus d'archivage(ARCn):**Copie les fichiers de journalisation(fichier redo log) dans l'emplacement de stockage d'archivage en cas de changement de fichier de journalisation.

Composants d'une BD : Structure Physique

Une base de donnée Oracle est composée des fichiers suivants :

Structures de base de données
Mémoire
Processus
> **Stockage**



Fichiers de contrôle



Fichiers de données



Fichiers de journalisation en ligne



Fichier de paramètres



Fichiers de sauvegarde



Fichiers de journalisation archivés



Fichier de mots de passe



Fichiers trace et fichiers d'alertes



Composants d'une BD : Structure Physique

- **Un fichier de contrôle (control files)**
 - Contient les informations relatives à la structure physique de la BD (nom de la base, localisation des fichiers de données et de reprise, etc.)
- **Des fichiers de données(data files)**
 - Stockent les données (données de gestion de la base, données utilisateurs de la base)
 - Un fichier de donnée est associé à une et une seule base de données
- **Des fichiers de reprise (redo log files)**
 - Contiennent la trace des dernières modifications de données
 - Permettent en cas de panne (matérielle ou logicielle) de remettre la BD dans un état cohérent
- **Des fichiers d'archivage**
 - Archivent les fichiers de contrôle
- **Un fichier de paramètres**
 - Stocke tous les paramètres de configuration de la base
- **Des fichiers de trace (d'alerte)**
 - Répertorie toutes les tâches et les erreurs effectuées



Composants d'une BD : Structure Physique

- **Un fichier de mots de passe**
 - Permet aux utilisateurs de se connecter à distance à la BD et d'effectuer des tâches DBA.
- **Des fichiers de sauvegarde**
 - Permettent la récupération de la base de données lorsqu'une défaillance physique ou une erreur utilisateur a endommagé ou supprimé le fichier d'origine.
- **Des fichiers de journalisation archivés**
 - Contiennent un historique des modifications des données générées par l'instance.
 - À l'aide de ces fichiers et d'une sauvegarde de la BD, vous pouvez récupérer un fichier de données perdu.
- **Des fichiers de journalisation en ligne**
 - Permettent la récupération d'une instance de la base de données. Si la base de données connaît une défaillance et qu'aucun fichier de données n'est perdu, l'instance peut récupérer la base de données à partir des informations de ces fichiers



1

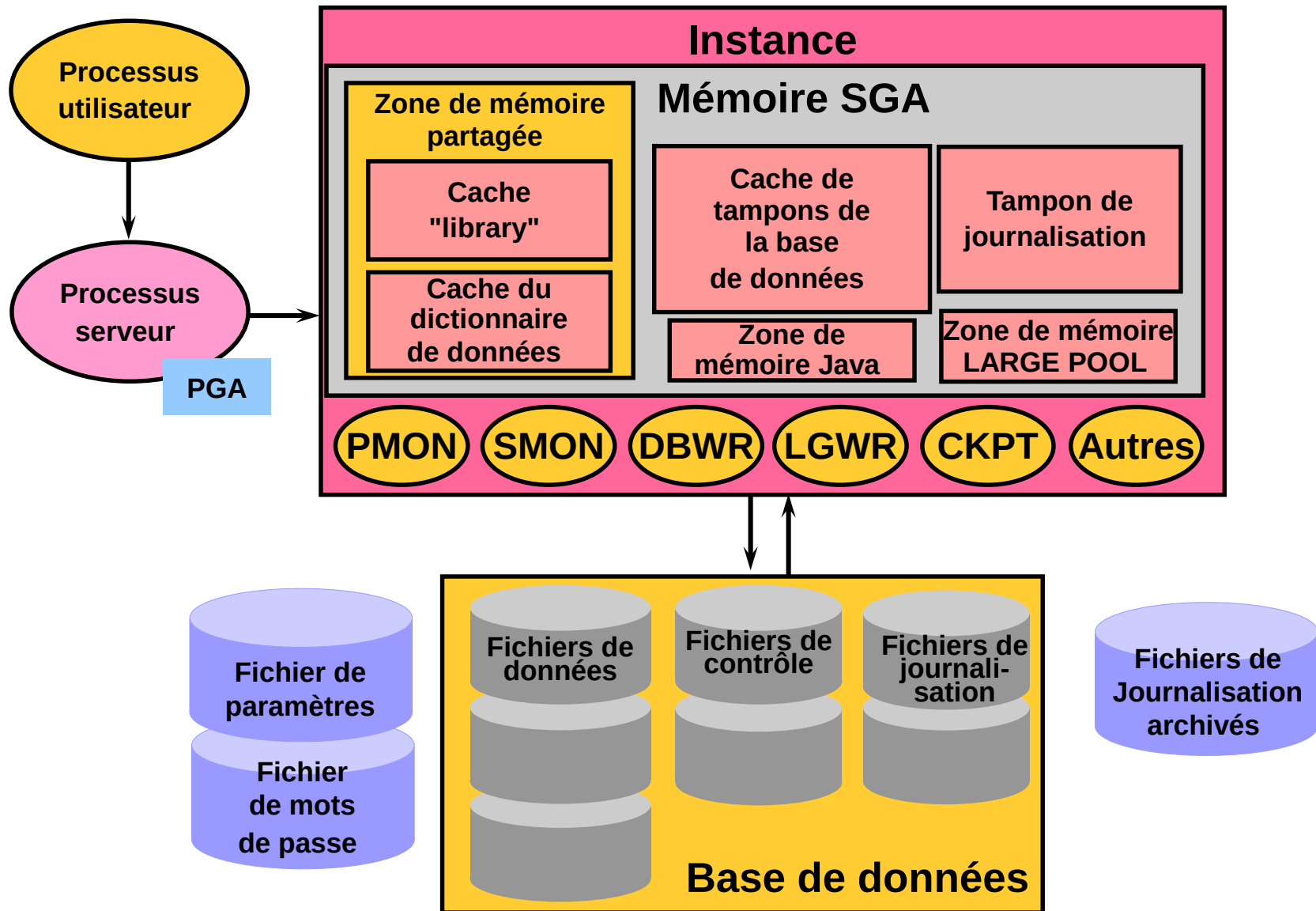
Composants de l'architecture Oracle

Objectifs

A la fin de ce chapitre, vous pourrez :

- **décrire l'architecture Oracle et ses principaux composants**
- **répertorier les structures utilisées dans la connexion d'un utilisateur à une instance Oracle**

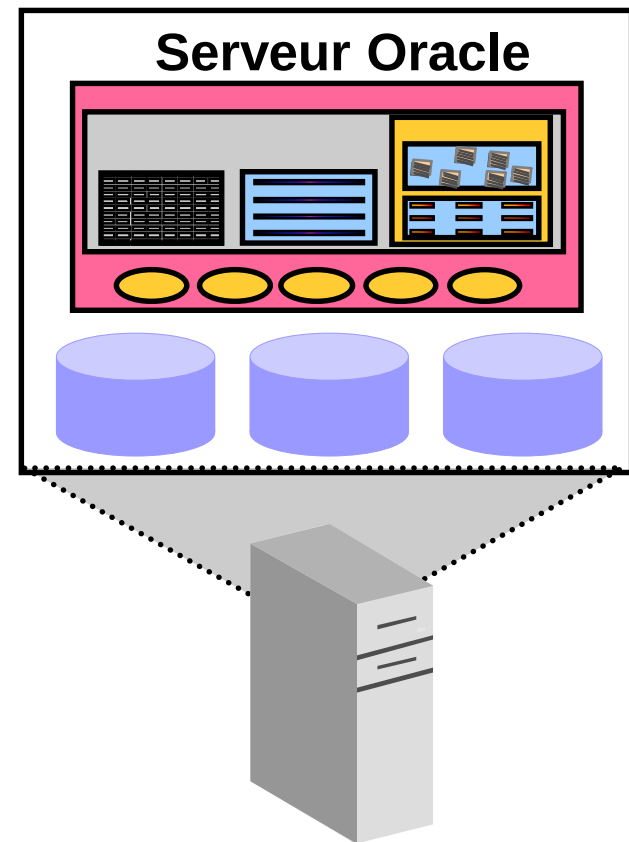
Présentation des principaux composants



Serveur Oracle

Un serveur Oracle :

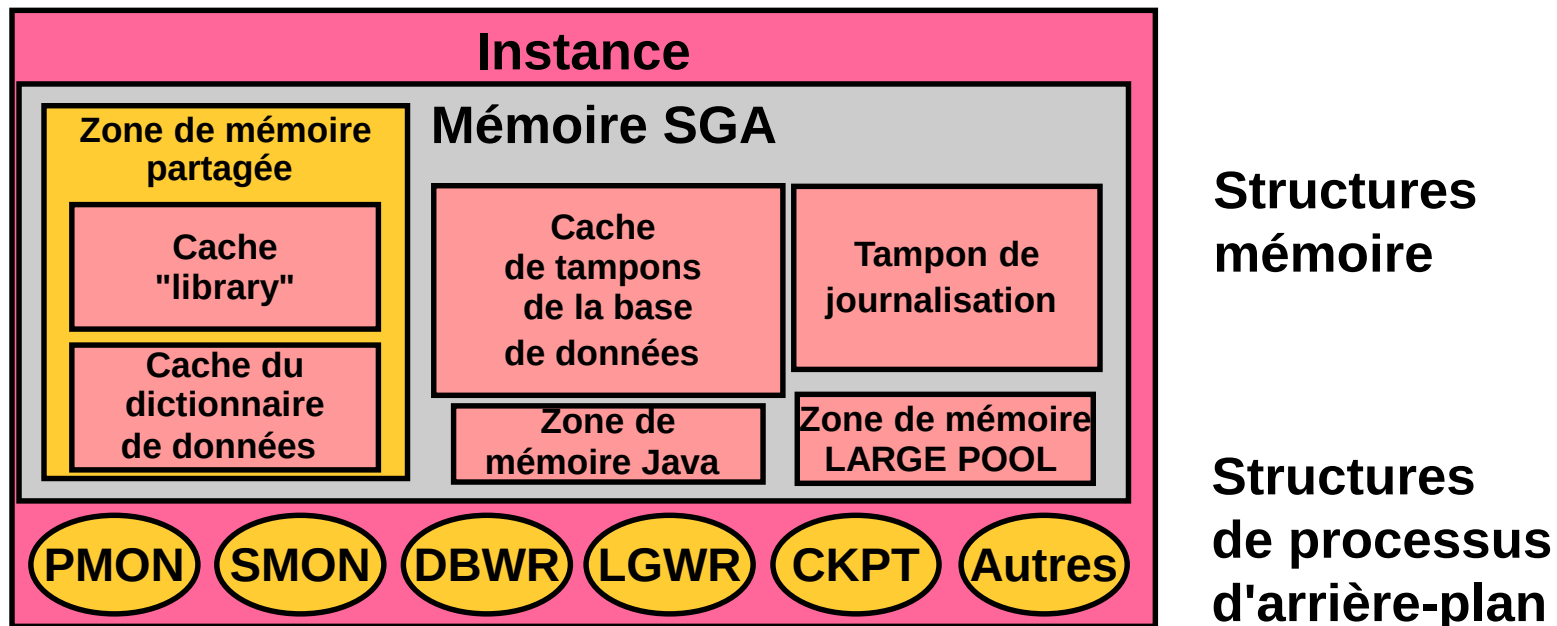
- **est un système de gestion de base de données qui offre une méthode de gestion des informations ouverte, complète et intégrée,**
- **est constitué d'une instance et d'une base de données Oracle.**



Instance Oracle

Une instance Oracle :

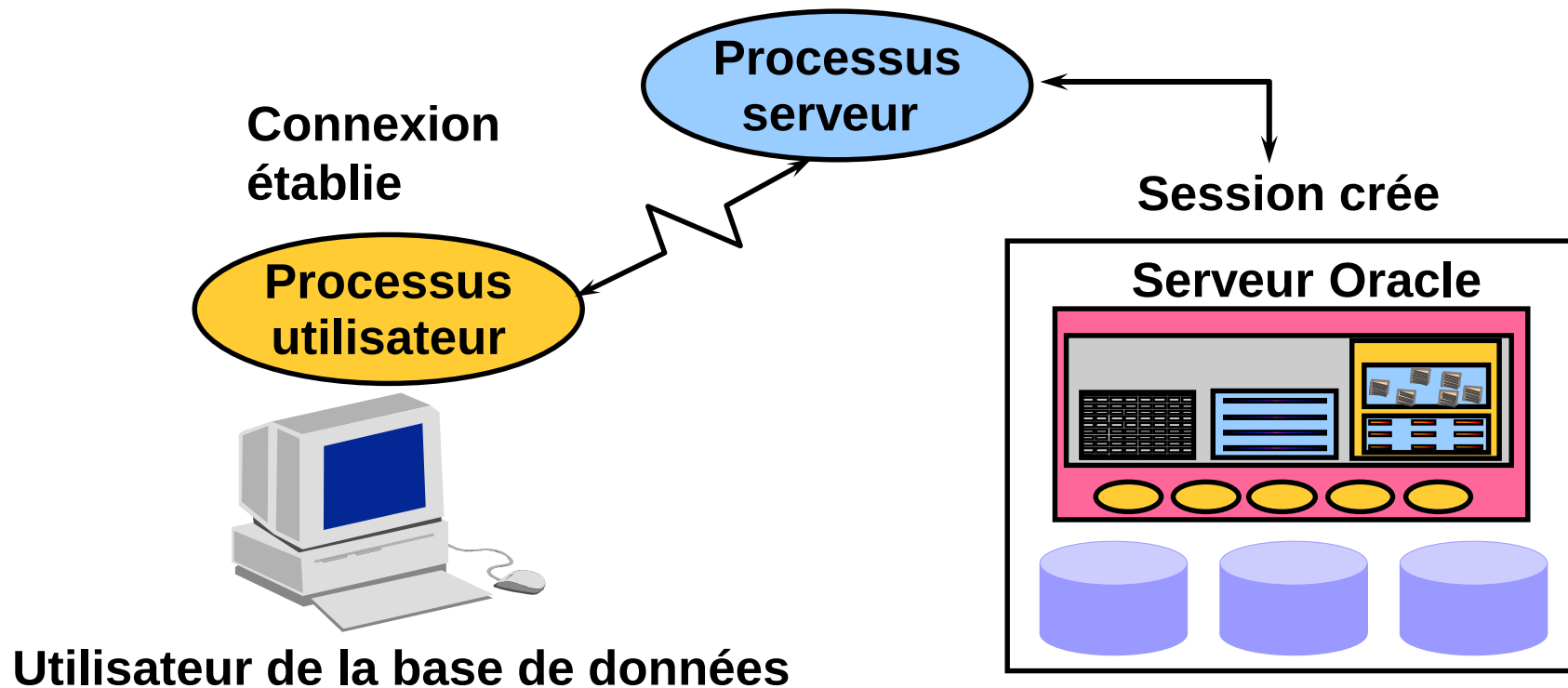
- permet d'accéder à une base de données Oracle,
- n'ouvre qu'une seule base de données,
- est constituée de structures de processus d'arrière-plan et de structures mémoire.



Etablir une connexion et créer une session

Se connecter à une instance Oracle :

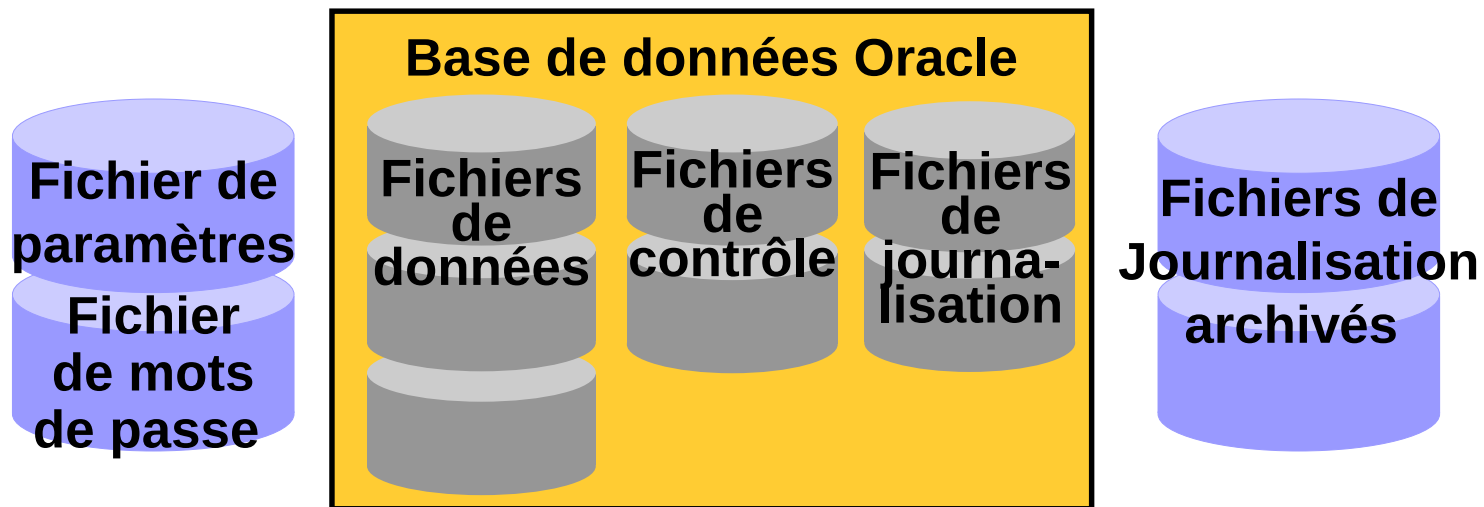
- Etablir une connexion utilisateur
- Créer une session



Base de données Oracle

Une base de données Oracle :

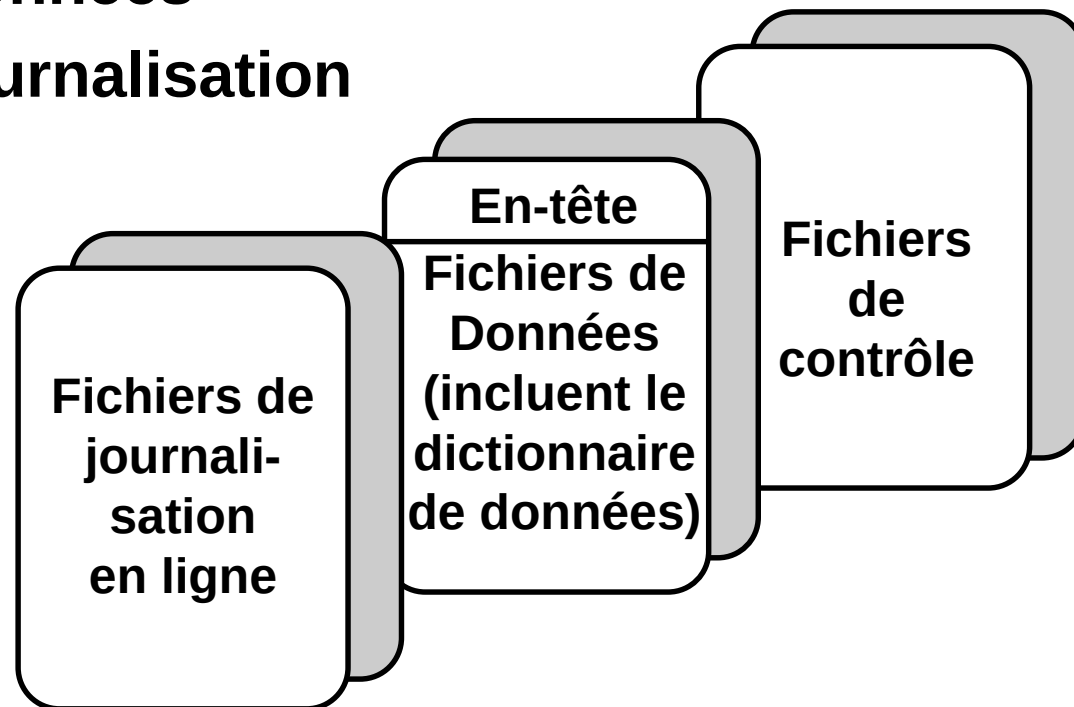
- **est un ensemble de données traitées comme une seule et même entité,**
- **est constituée de trois types de fichier.**



Structure physique

La structure physique 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 Oracle**
- **la mémoire PGA, qui est allouée au démarrage du processus serveur**

Mémoire SGA

- **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.

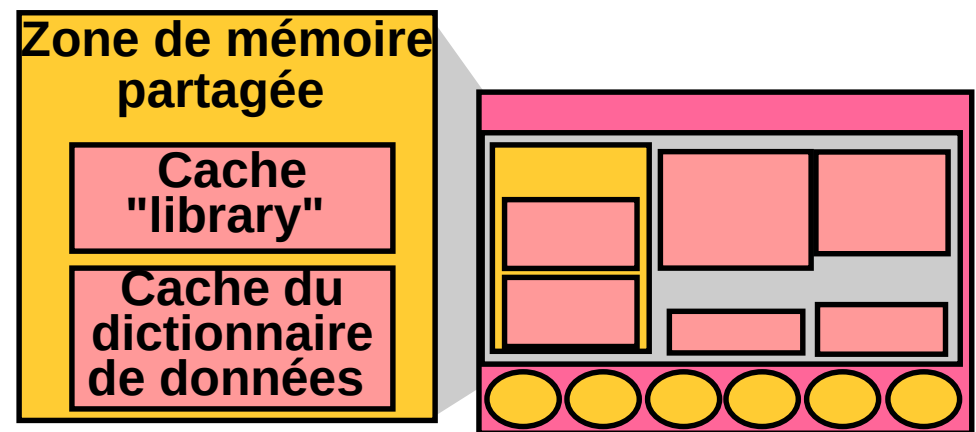
Mémoire SGA

- **Dynamique**
- **Taille définie à l'aide du paramètre `SGA_MAX_SIZE`**
- **Allocation et suivi sous forme de granules par les composants de la mémoire SGA**
 - **Allocation de mémoire virtuelle contiguë**
 - **Taille des granules définie en fonction de la valeur totale estimée de `SGA_MAX_SIZE`**

Zone de mémoire partagée

- Elle permet de stocker :
 - les dernières instructions SQL exécutées,
 - les dernières définitions de données utilisées.
- Elle est constituée de deux structures mémoire clés liées aux performances :
 - Cache "library"
 - Cache du dictionnaire de données
- Sa taille est définie par le paramètre `SHARED_POOL_SIZE`.

```
ALTER SYSTEM SET  
SHARED_POOL_SIZE = 64M;
```



Cache "library"

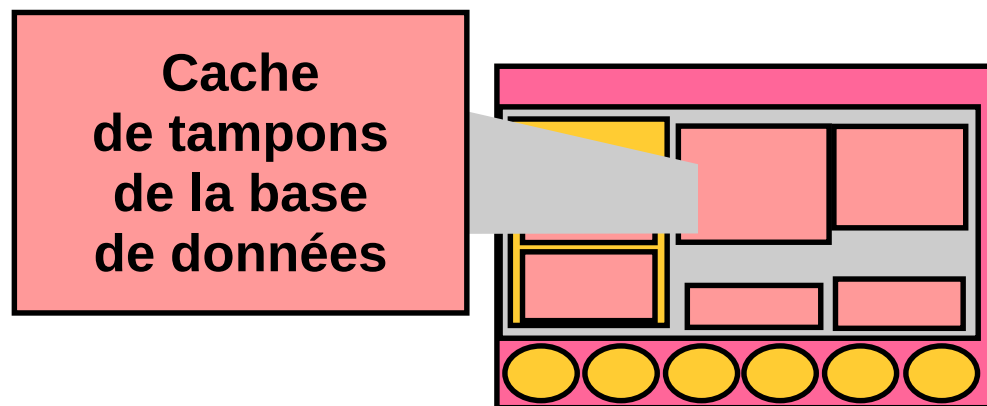
- Le cache "library" conserve des informations sur les dernières instructions SQL et PL/SQL utilisées.
- Il permet le partage des instructions fréquemment utilisées.
- Il est géré par un algorithme LRU.
- Il est composé de deux structures :
 - la zone SQL partagée,
 - la zone PL/SQL partagée.
- Sa taille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

Cache du dictionnaire de données

- Le cache du dictionnaire de données contient les dernières définitions utilisées 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

- 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.



Cache de tampons de la base de données

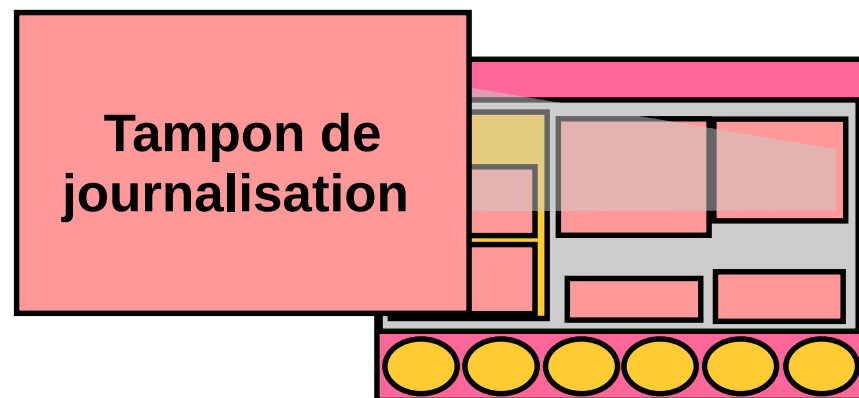
- Ce cache est composé de sous-caches indépendants :
 - DB_CACHE_SIZE
 - DB_KEEP_CACHE_SIZE
 - DB_RECYCLE_CACHE_SIZE
- Il peut être redimensionné dynamiquement :

```
ALTER SYSTEM SET DB_CACHE_SIZE = 96M;
```

- Le paramètre DB_CACHE_ADVICE peut être défini pour collecter des statistiques permettant de prévoir le comportement du serveur en fonction de différentes tailles de cache.
- La vue V\$DB_CACHE_ADVICE affiche les statistiques collectées.

Tampon de journalisation

- 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

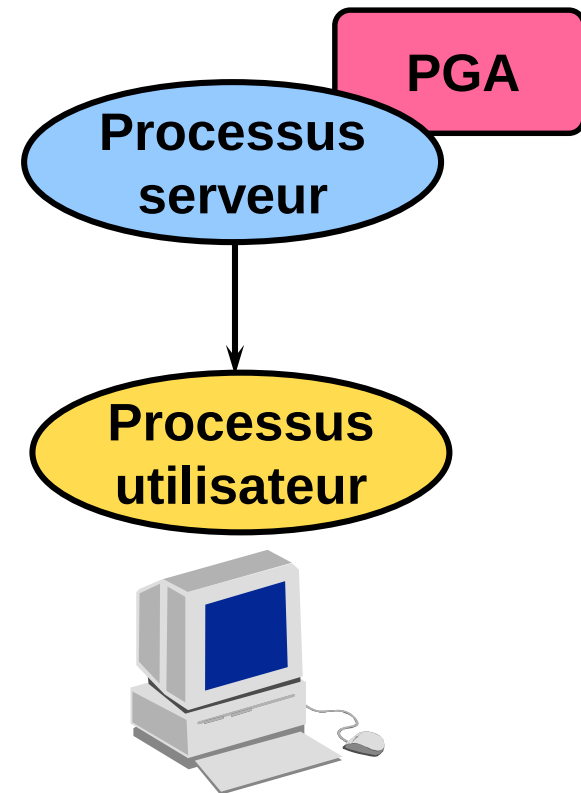
- Zone facultative de la mémoire SGA
- Elle réduit la charge de la zone de mémoire partagée.
 - la mémoire allouée par session (UGA) au serveur partagé
 - les processus serveur d'E/S
 - les opérations de sauvegarde et de restauration ou RMAN
 - les mémoires tampon des messages d'exécution en parallèle
 - `PARALLEL_AUTOMATIC_TUNING = TRUE`
- Elle n'utilise pas de liste LRU.
- Sa taille est définie par le paramètre `LARGE_POOL_SIZE`.

Zone de mémoire Java

- 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 PGA

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



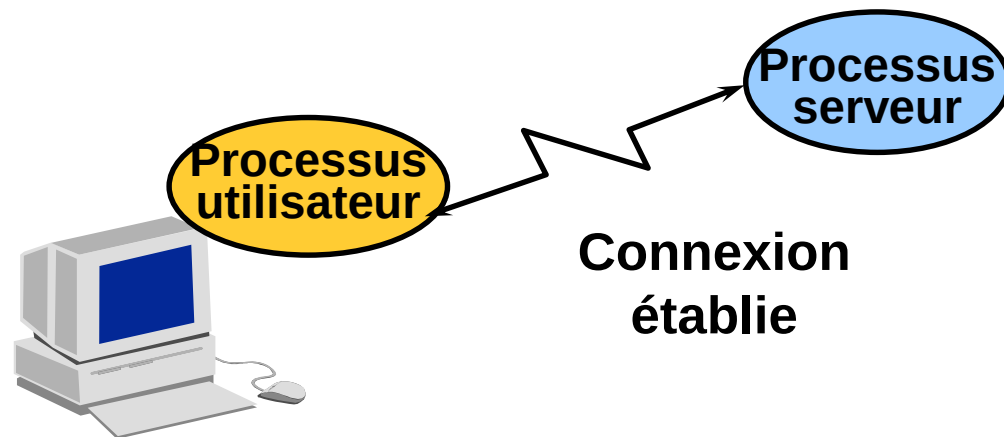
Structure de processus

Oracle utilise différents types de processus :

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

Processus utilisateur

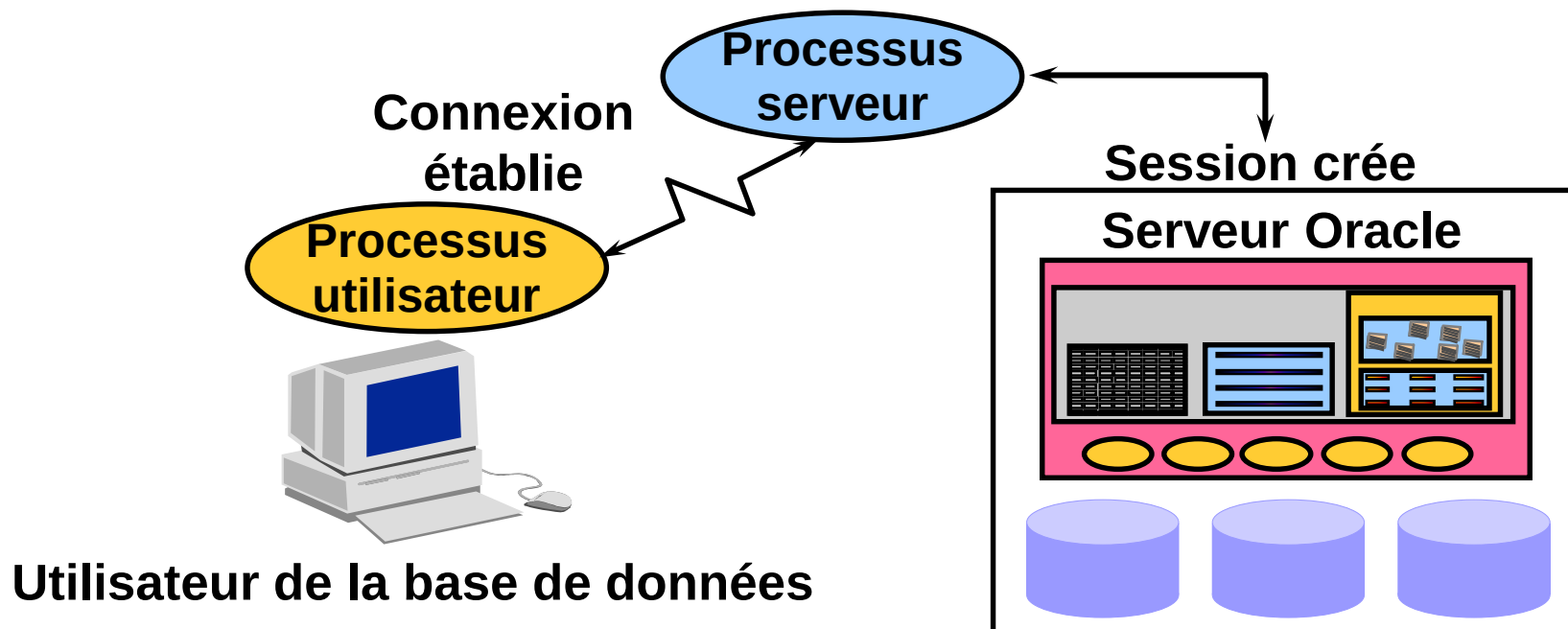
- Programme qui demande une interaction avec le serveur Oracle.
- Ce processus doit d'abord établir une connexion.
- Il n'entre pas directement en interaction avec le serveur Oracle.



Utilisateur de la base de données

Processus serveur

- Programme qui entre directement en interaction avec le serveur Oracle.
- 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

Gèrent et appliquent les relations entre les structures physiques et les structures mémoire.

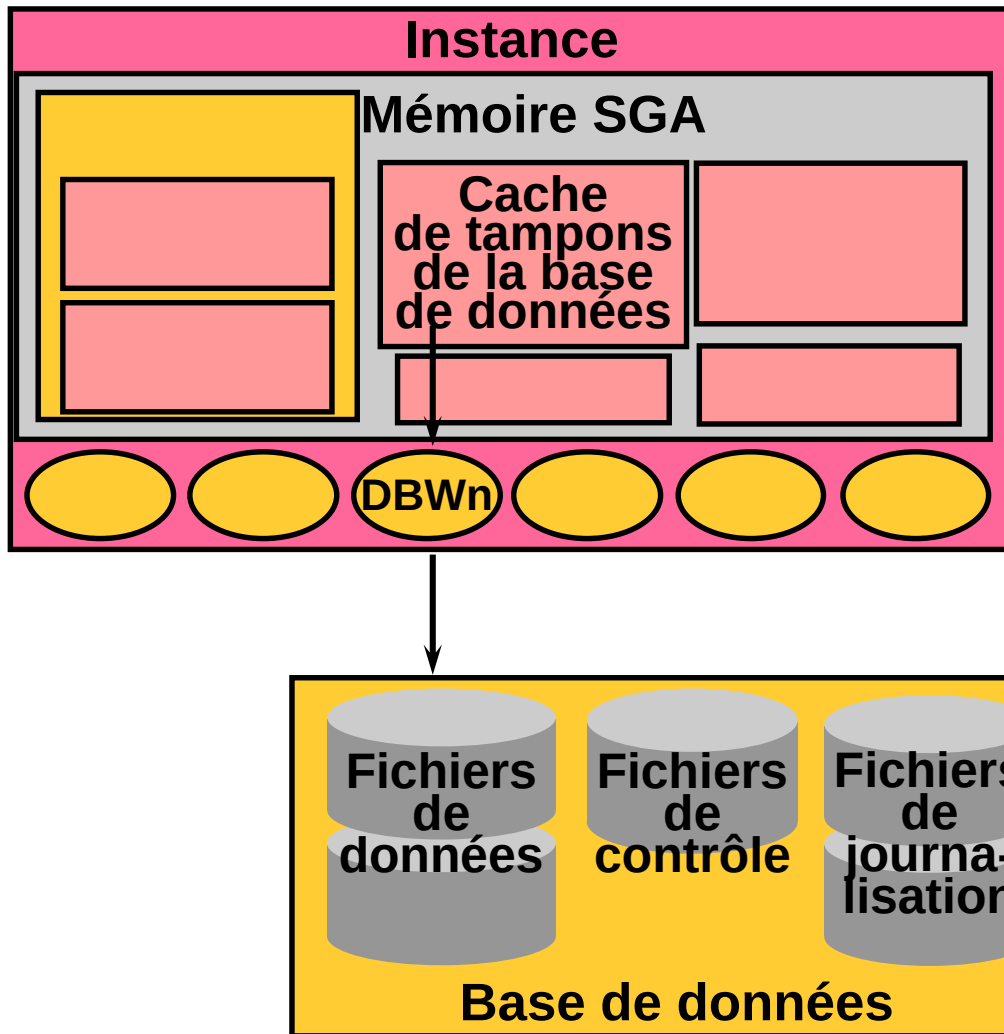
- **Processus d'arrière-plan obligatoires**

- DBWn PMON CKPT
- LGWR SMON

- **Processus d'arrière-plan facultatifs**

- ARCN LMDn RECO
- CJQ0 LMON Snnn
- Dnnn Pnnn
- LCKn QMNn

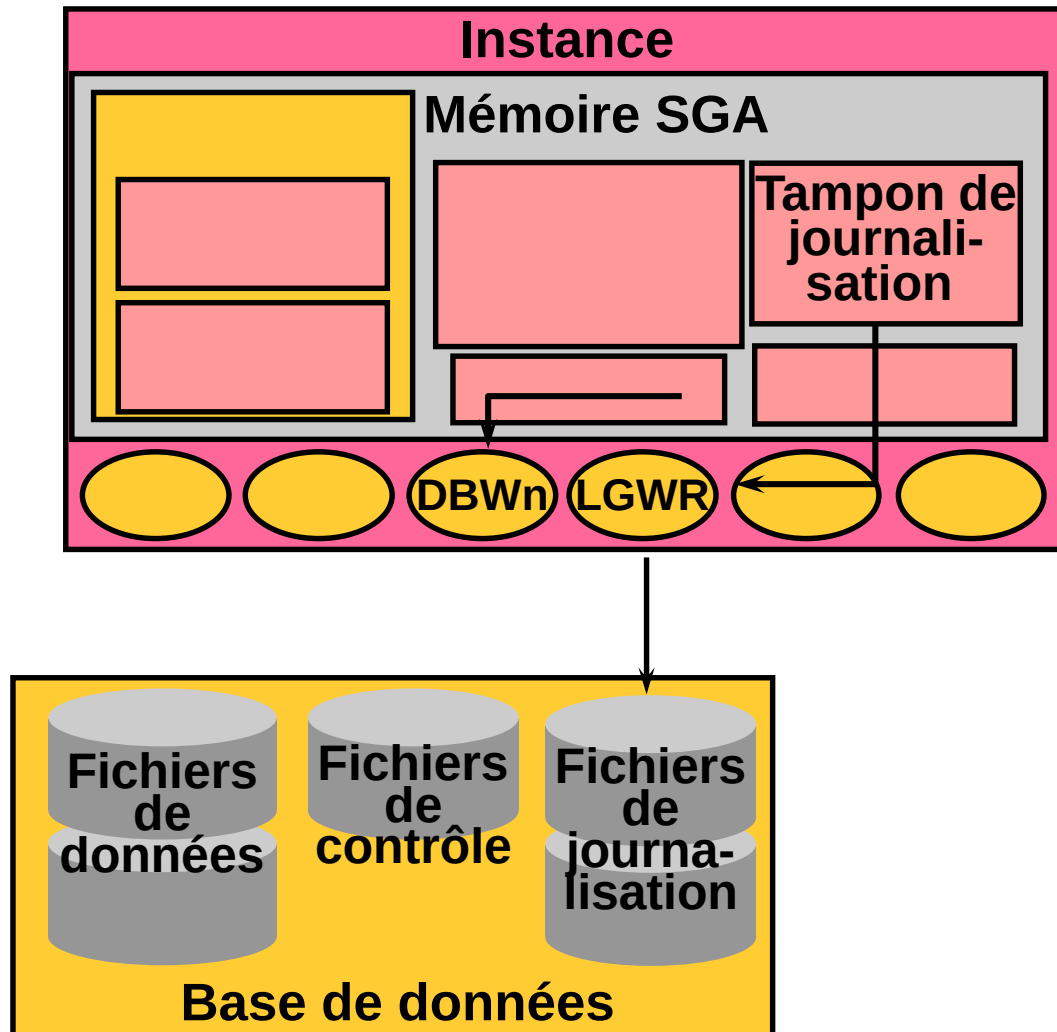
Processus database writer (DBWn)



DBWn écrit dans les cas suivants :

- point de reprise
- seuil des tampons "dirty" atteint
- aucune mémoire tampon disponible
- temps imparti dépassé
- demande de ping RAC
- tablespace hors ligne
- tablespace en lecture seule
- DROP ou TRUNCATE sur une table
- BEGIN BACKUP sur un tablespace

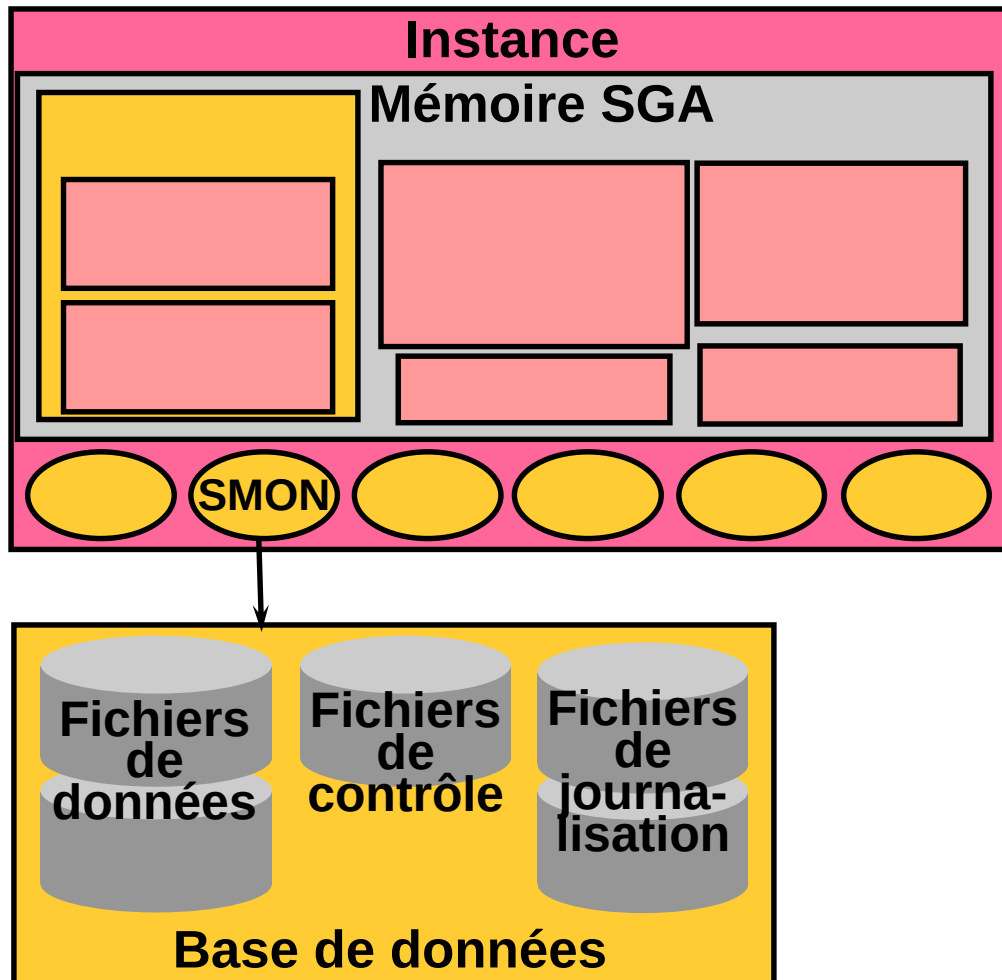
Processus LGWR (Log Writer)



LGWR écrit dans les cas suivants :

- validation
- un tiers du cache est occupé
- la journalisation atteint 1 Mo
- toutes les trois secondes
- avant que le processus DBWn ne procède à une opération d'écriture

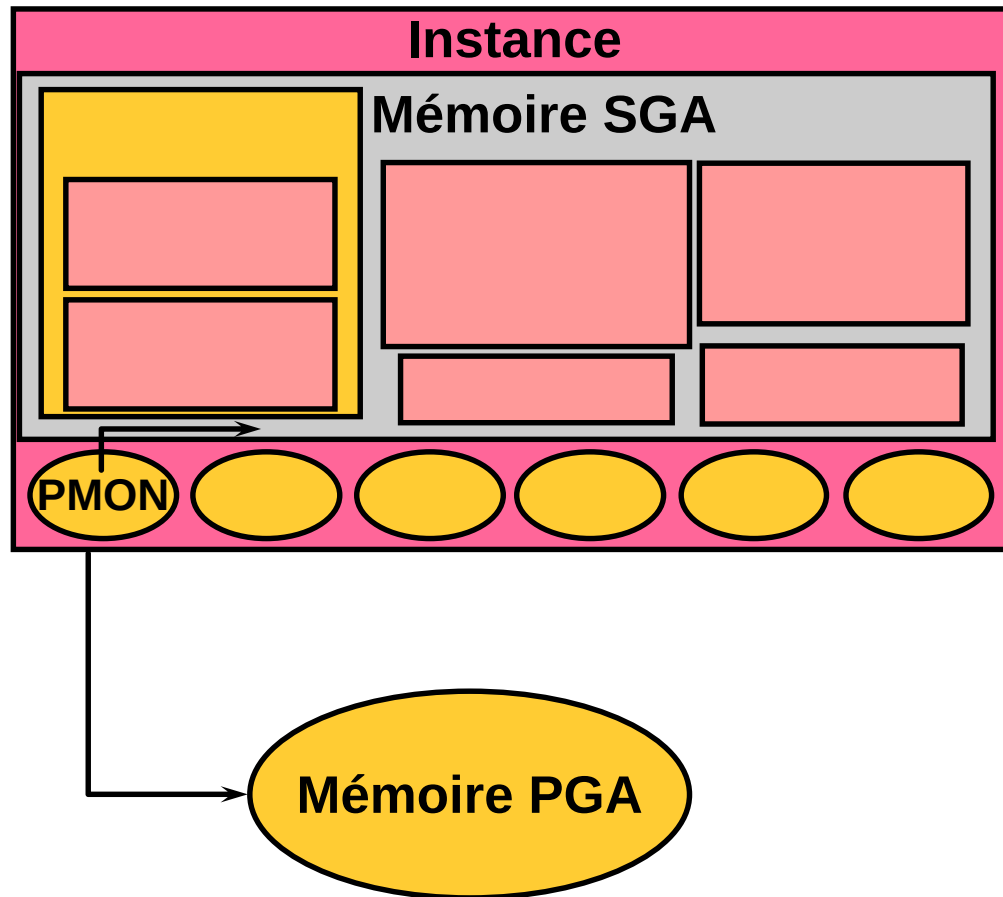
Processus SMON (System Monitor)



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 segments

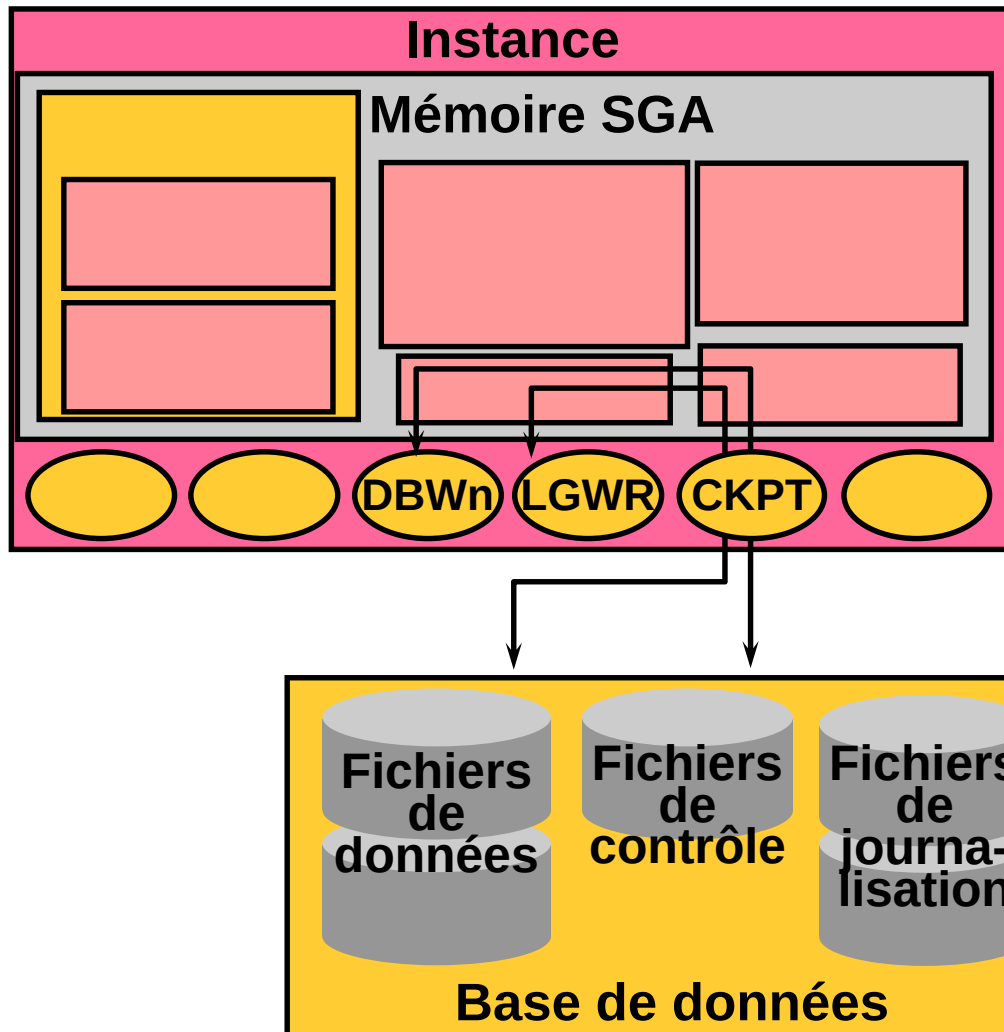
Processus PMON (Process Monitor)



Suite à l'échec de processus, PMON exécute des opérations de nettoyage :

- annule la transaction
- libère des verrous
- libère d'autres ressources
- redémarre les répartiteurs interrompus

Processus CKPT (Checkpoint)

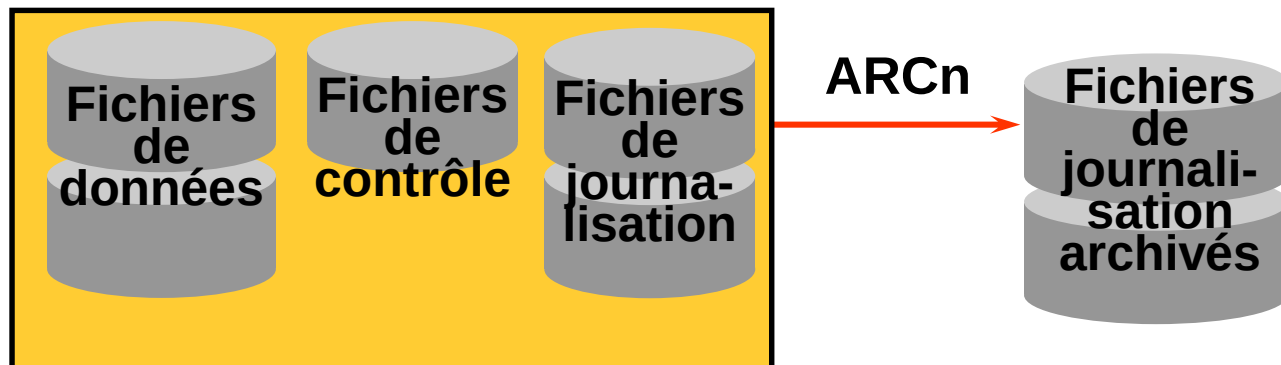


Ce processus est chargé :

- de signaler DBWn aux points de reprise,
- de mettre à jour les en-têtes 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.

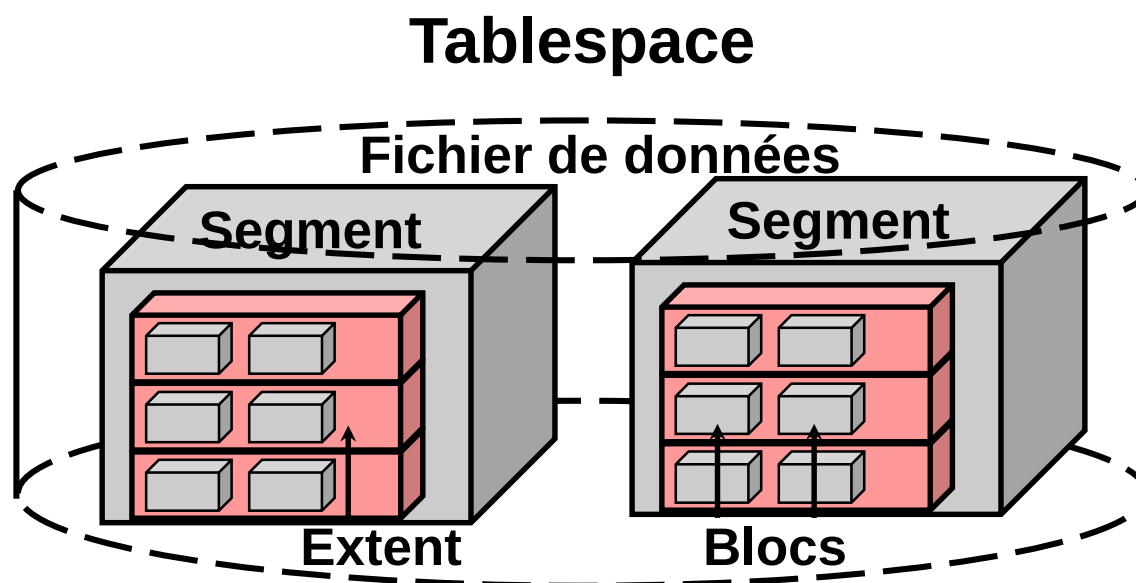
Processus ARCn (processus d'archivage)

- Processus d'arrière-plan facultatif
- En mode ARCHIVELOG, il archive automatiquement les fichiers de journalisation en ligne
- Il enregistre toutes les modifications apportées à la base de données



Structure logique

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



Traiter les instructions SQL

- **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.**

Synthèse

Ce chapitre vous a présenté :

- **les fichiers de base de données : fichiers de données, fichiers de contrôle, fichiers de journalisation en ligne**
- **les structures mémoire SGA : cache de tampons de la base de données, zone de mémoire partagée et tampon de journalisation**
- **les principaux processus d'arrière-plan : DBWn, LGWR, CKPT, PMON et SMON**
- **l'utilisation du processus d'arrière-plan ARCn**
- **les processus d'arrière-plan facultatifs et conditionnels**
- **la hiérarchie logique**