

# Partiel1: I-Introduction aux SGBDs

2<sup>ème</sup> Ingénieurs info

Année Universitaire 2020-2021

R.MGHIRBI

## Chapitre 1: -Présentation d'un SGBD

# Plan

1. Objectifs d'un SGBD
2. Vue simplifiée d'un SGBD
3. Fonctions d'un SGBD
4. Exemples d'SGBDs
5. SGBD Oracle
  - Organisation
  - Dictionnaire de données
6. Plan du module SGBD

## 1. Objectifs d'un SGBD

- ◆ Indépendance physique des programmes aux données
- ◆ Indépendance logique des programmes aux données
- ◆ Manipulation des données par des langages non procéduraux
- ◆ Administration facilitée des données.
- ◆ Efficacité des accès aux données
- ◆ Cohérence des données
- ◆ Sécurité des données

# 1. Objectifs d'un SGBD

## Objectifs

- ◆ Indépendance physique des programmes aux données
  - Insérer, supprimer, modifier des données directement sur l'image logique.
  - Le SGBD va s'occuper de faire le travail sur les fichiers physiques
    - Dans le cadre d'un SGBDR:
      - => Modèle conceptuel (entité/association)
      - => Génération du modèle logique de données (modèle relationnel)
      - => Langages de Définition de Données SQL2
      - => Création des structures logiques de la base

# 1. Objectifs d'un SGBD

## Objectifs:

- ◆ Indépendance physique des programmes aux données
- ◆ Indépendance logique des programmes aux données
  - Un même ensemble de données peut être vu différemment par des utilisateurs différents.
  - Notion des vues (à voir dans le cours) : Les vues permettent d'implémenter l'indépendance logique en permettant de créer des **objets virtuels: chacun voit ce qui l'intéresse**
  - Besoin de protection avec:
    - Droits et devoirs des utilisateurs
    - Notion de rôles et de privilèges
  - Objectif dérivé: **Sécurité des accès et confidentialité**

# 1. Objectifs d'un SGBD

## Objectifs:

- ♦ Indépendance physique des programmes aux données
- ♦ Indépendance logique des programmes aux données
- ♦ Manipulation des données par des langages

Interface de communication SQL/JAVA (JDBC)  
recours éventuel à des programmes stockés (procédures/ fonctions PLSQL  
par exemples)

# 1. Objectifs d'un SGBD

## Objectifs:

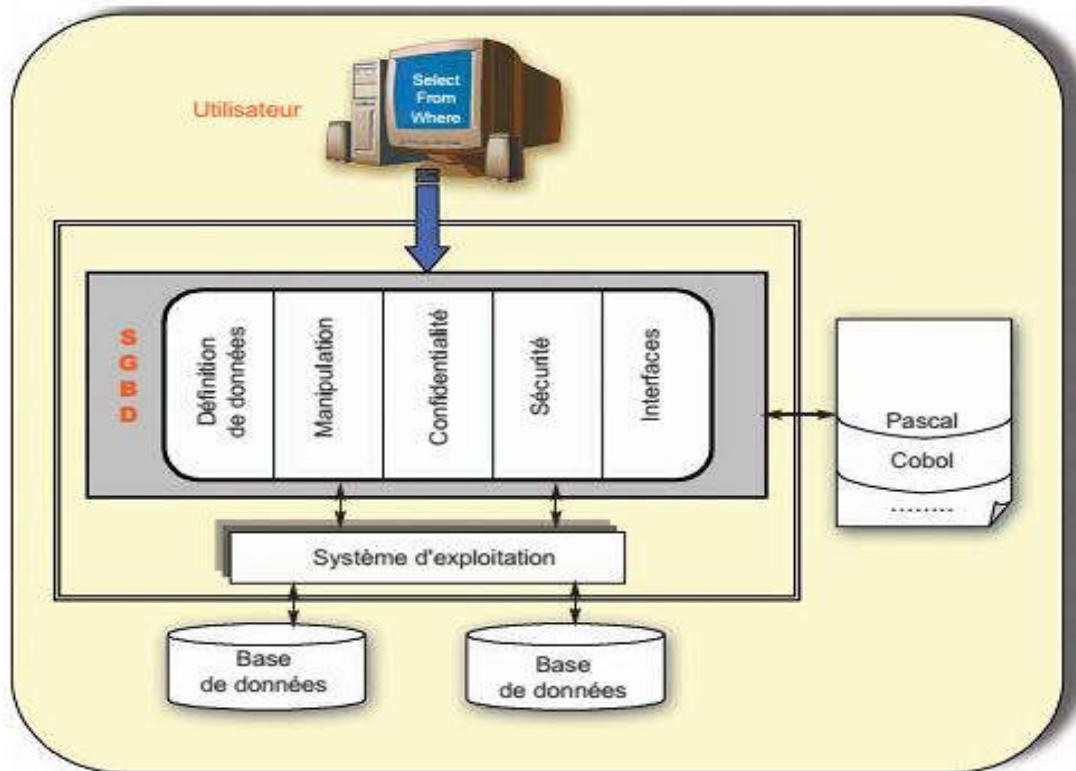
- ♦ Indépendance physique des programmes aux données
- ♦ Indépendance logique des programmes aux données
- ♦ Manipulation des données par des langages
- ♦ Administration facilitée des données.

Des visions différentes des données (entre autres) se résolvent plus facilement si les données sont administrées de façon centralisée.

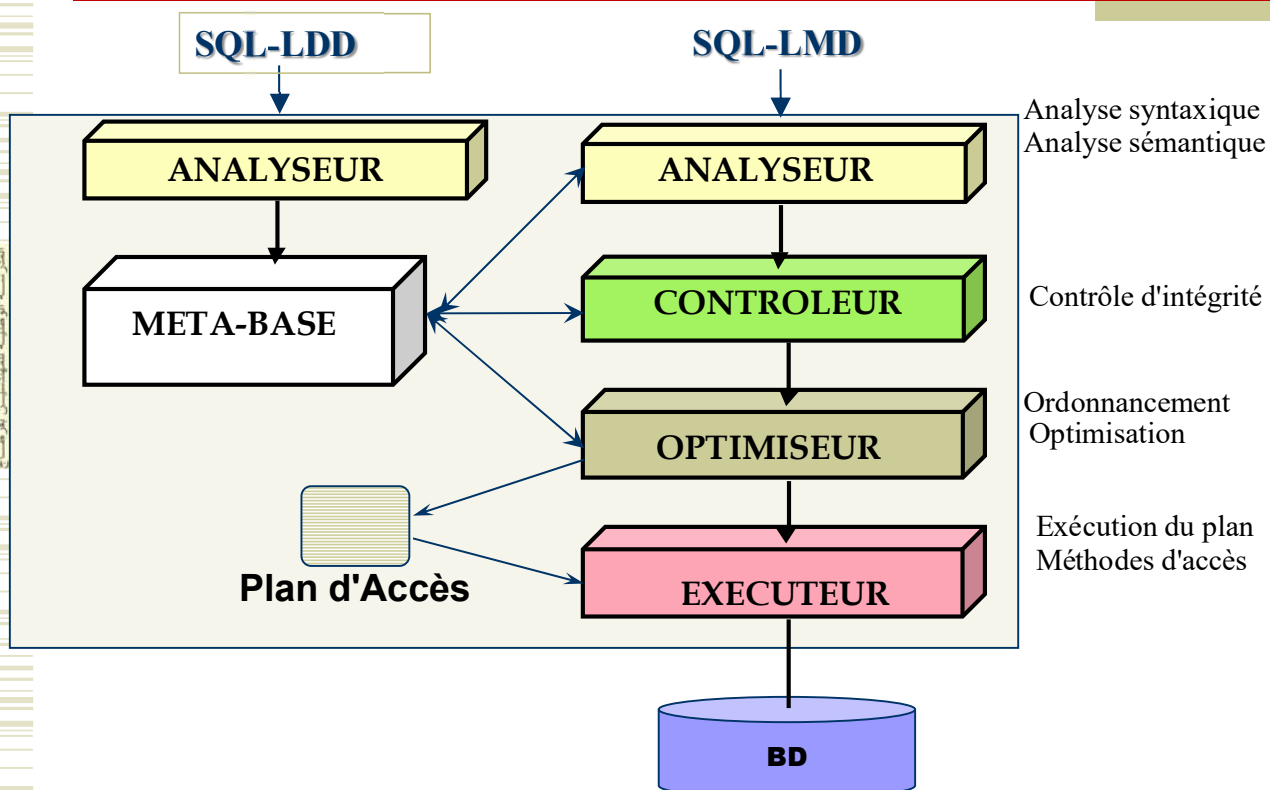
Pbm: évolutivité? => fournir des outils permettant de décentraliser la description de données (**le dictionnaire**), tout en assurant une cohérence entre les diverses descriptions partielles

➔ Gestion de la meta\_base

## 2. Vue simplifiée



## Architecture d'un SGBD



# 3. Fonctionnalités d'un SGBD

## 1. Décrire l'information

- Création des objets avec leurs contraintes
- Modification des structures et des contraintes
- Gestion de l'espace disque

## 2. Respecter l'Intégrité de l'information

- Intégrité logique: contrôle de contraintes lors des ordres LMD
- Intégrité physique: contrôle de transactions (annulation ou confirmation)
- Reprise après incident

## 3. Partager l'information

- Concurrency (verrouillage)

## 4. Garantir la Confidentialité des informations

- Tout le monde ne peut pas voir et faire n'importe quoi
- Droits et devoirs des utilisateurs
- Notion de rôles et de privilèges

# 4. Exemples des SGBDs

## ◆ Les grands SGBD

- Oracle
- IBM DB2
- Microsoft SQL Server
- Sybase SQL Server
- Ingres
- Informix

## ◆ Les open sources

- MySQL
- PostgreSQL

## ◆ Les SGBD personnels

- Borland Paradox
- Filemaker
- Interbase
- Microsoft Access
- Microsoft FoxPro

## ◆ Les SGBD objets

- Objectivity
- Object Store
- Versant
- O2

## 5-A SGBD Oracle: Organisation

### Pourquoi Oracle

- ♦ Au Top 3 des SGBD: Oracle → MySQL → SQLserver
- ♦ Leader en mars 2018. Leader en mars 2019.
- ♦ C'est le SGBD le plus « portable », que ce soit en matière de machines ou d'architectures ;
- ♦ C'est le SGBD le plus utilisé sur le marché mondial (48% du parc global), ce qui se traduit par une documentation dense et une communauté active ;
- ♦ C'est le SGBD le plus « adaptable », puisqu'il permet de créer des applications ultra-personnalisées pour éliminer les fonctionnalités superflues et gagner en performance ;
- ♦ Il est accessible à tous les langages et à toutes les technologies ;
- ♦ Il se démarque de ses concurrents par des fonctionnalités intéressantes comme la gestion avancée des pools de sessions et connexions, un support de migrabilité de session avancé, la gestion de Large Objects (LOB) de 128 TetraOctets, etc.
- ♦ engouement des professionnels pour Oracle → les 10 plus grandes banques à l'échelle mondiale et les départements de la défense de plusieurs pays font d'Oracle DB leur principal SGBD
  - <http://e-mag.xxci.fr/sghd-plus-populaires/>

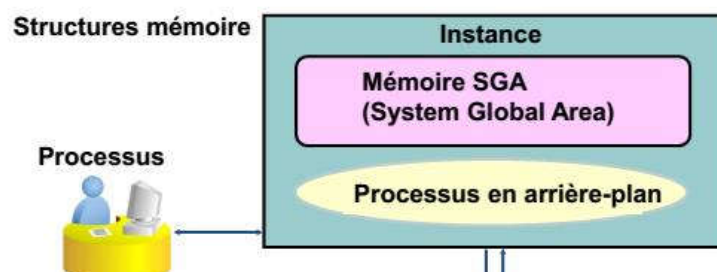


## 5.1. Serveur Oracle

- ♦ Un serveur de bases de données est composé de/d':
  - Une instance = plusieurs processus et une zone de mémoire
  - Une base de données:
    - Structures de stockage physiques et logiques
  - Plusieurs **schémas**, assimilés à des **utilisateurs**

## 5.2. Qu'est ce qu'une instance oracle?

- ♦ Une instance Oracle est un ensemble de **processus** et de zones mémoires (mémoire principale) qui permettent de gérer la base de données.
- ♦ Elle est caractérisée par son **System ID (SID)** appelé **ORACLE\_SID** ;
- ♦ Une instance ne peut être associée qu'à une et une seule base de données, par contre une base de données peut utiliser plusieurs instances.





## 5.3. Les processus

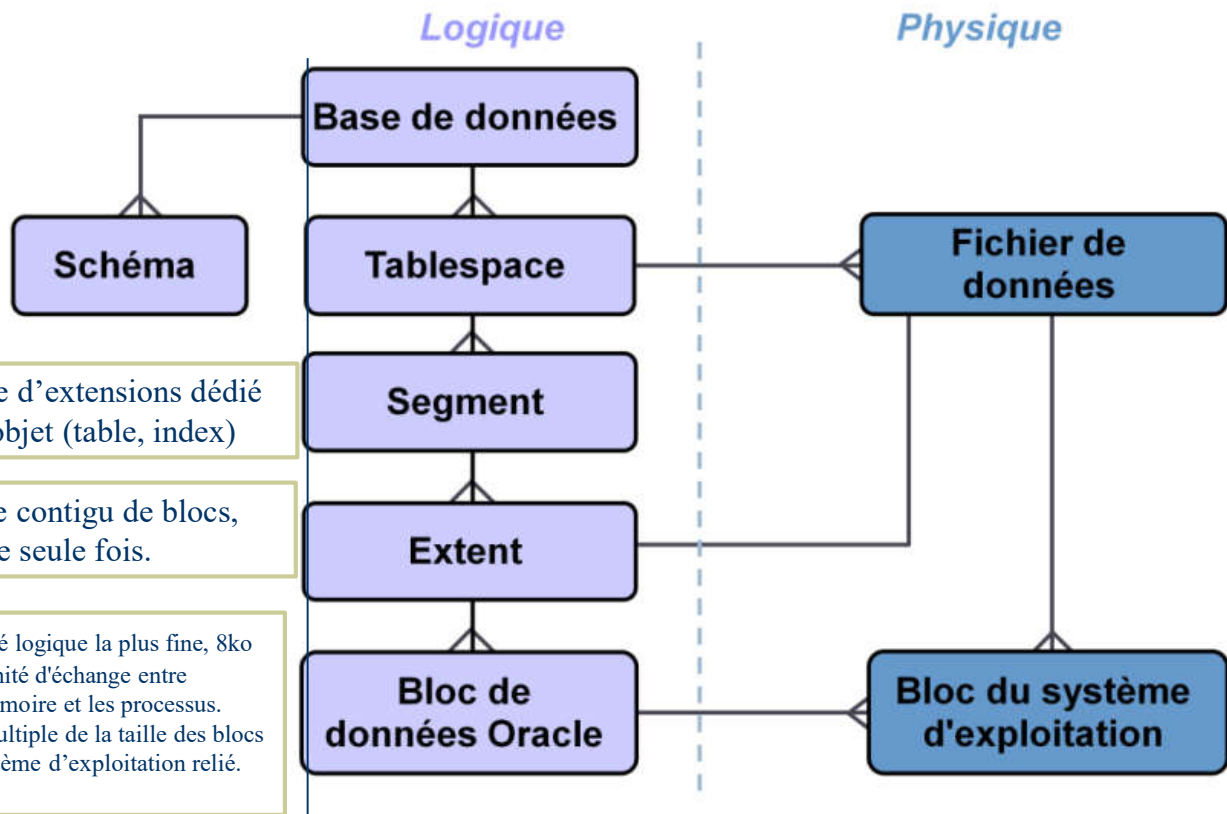
- ♦ Ils permettent une interaction entre les différentes composantes du serveur ainsi qu'avec les utilisateurs.
- ♦ Il existe trois types de processus
  1. Les processus utilisateur: s'exécutent au niveau client.
  2. Les processus serveur: s'exécutent au niveau serveur.
  3. Les processus d'arrière plan: Assurent le bon fonctionnement du serveur.
- ♦ L'interaction entre le serveur et les clients se fait en réalité grâce à ces deux processus, chacun de son côté.
- ♦ Lorsque le client est lié au serveur, on parle d'une **connexion**.
- ♦ Lorsque l'utilisateur **s'identifie**, il ouvre une **session**.
- ♦ Plusieurs sessions peuvent être ouvertes en même temps.

2<sup>ème</sup> Ing.Inf

## 5.4. Structures de stockage

- ♦ Une base de données Oracle est un ensemble de données traitées comme une seule et même unité.
- ♦ Une base de données comporte des structures logiques et des structures physiques.

## 5.5. Structures de stockage



## Les structures physiques

### ◆ Des fichiers sur le disque:

- Plusieurs **fichiers de données** pour stocker les données (.dbf)
- un **fichier de contrôle** qui spécifie le nom et l'emplacement des fichiers, le nom de la base,... (.ctl)
- au moins deux **fichiers de reprise ou de journalisation** après panne qui contiennent les modifications récentes (.rdo ou .log)
- des **fichiers d'archivage** (optionnel) pour archiver les fichiers de contrôle
- un **fichier de paramètres** (optionnel) qui stocke tous les paramètres de la base
- des **fichiers de trace** pour répertorier toutes les tâches et erreurs effectuées

# Les structures physiques

## ♦ Fichiers de données (Data Files)

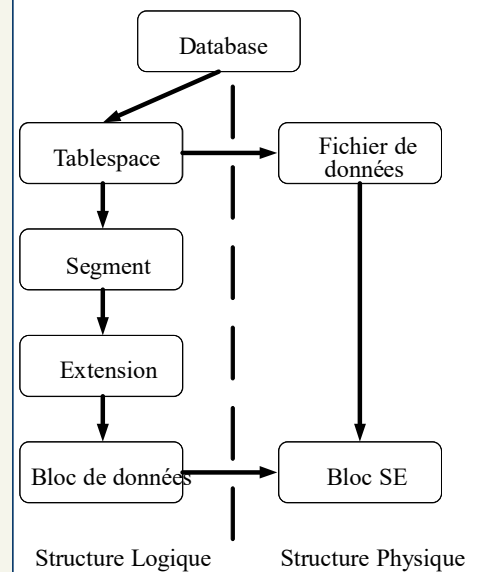
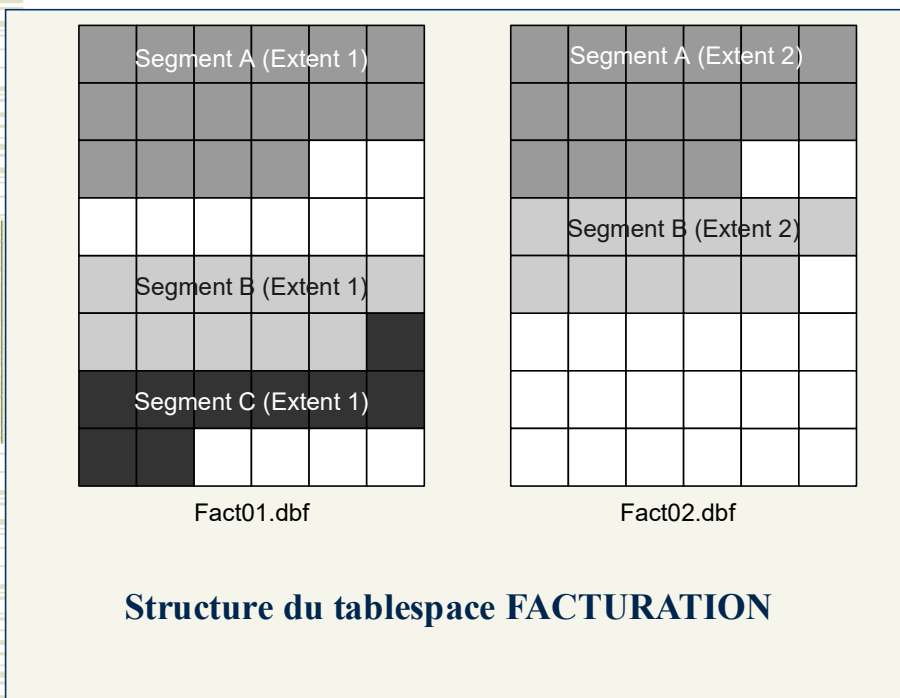
- Stockent les données sous un format spécial à Oracle (format propriétaire):
- Ce sont des fichiers binaires, ce qui signifie qu'ils sont inexploitable avec un éditeur de texte.
- Seules les requêtes SQL permettant un accès implicite à ces fichiers.
- Ces fichiers contiennent l'ensemble des données de la base (les tables, les vues, les procédures stockées, ...).
- La lecture de ces fichiers de données est faite à l'aide des processus utilisateurs.
- L'écriture est assurée par le processus DBWR (Database Writer).

# Les structures logiques

## TABLESPACE

- ♦ Chaque base de données est divisée de manière logique en un ou plusieurs tablespaces.
  - Un tablespace est une unité logique qui correspond physiquement à un ou plusieurs fichiers de données.
  - Un tableSpace renseigne sur la nature des fichiers de données qui vont être créés physiquement: Un tablespace TEMPORARY contient un fichier temporaire et non un fichier de données.
  - Il regroupe un ensemble d'objets (=segments) pour faciliter leur administration.
    - Peut atteindre qq exabytes !
    - EX: On peut créer plusieurs tablespaces dans une BD d'une ESE commerciale qui gère la FACTURATION, les RH, et le PARC INFO.
      - ♦ Chaque tablespace va gérer ses propres objets et sera enregistré sur plusieurs fichiers de données

# Exemple



2<sup>ème</sup> Ing.Inf

## 5.6 Schéma

- ◆ Ensemble d'objets qui ont été créés par un utilisateur.
  - Correspond à un utilisateur : les deux portent le même nom
- ◆ Chaque utilisateur est le propriétaire d'un unique schéma.
- ◆ Les principaux types d'objets d'un schéma sont les **tables**, les **index**, les **vues**, les **synonymes**, les **séquences**, les **déclencheurs**, les **fonctions et procédures stockées** et les **packages PL/SQL**.
- ◆ Seuls les **tables** et les **index** sont stockés sous forme de segments (dans un ou plusieurs tablespace)
- ◆ Les autres objets correspondent à des définitions dans le **dictionnaire de données**.
- ◆ Pas de correspondance schema/tablespace: Les objets d'un schéma peuvent être stockés sur différents tablespaces tandis qu'un unique tablespace peut inclure plusieurs schémas.

2<sup>ème</sup> Ing.Inf

## 5-B Dictionnaire de données d'oracle (DD)

### 1. Présentation

- ◆ Un schéma utilisateur est créé et géré par celui-ci
- ◆ le dictionnaire de données (DD), lui, est créé et géré par Oracle.
- ◆ Il constitue la description d'une base de données. :
- ◆ C'est un référentiel créé dans le tablespace **SYSTEM**, et appartient au super utilisateur **SYS**
  - Il stocke des métadonnées (infos) sur les tables, colonnes de tables, les contraintes, les index ...etc
  - Il permet de stocker également :
    - les objets utilisateurs tels que les vues, synonymes, séquences, procédures, fonctions, packages, déclencheurs (triggers)...etc
    - Les noms des utilisateurs, leurs privilèges et rôles ...etc.
  - Par exemple, si un utilisateur USER1 veut afficher la structure d'une de table TABLE1 qui lui appartient, il lance la commande "DESC TABLE1". →
    - Le résultat de cette commande est obtenu à partir du dictionnaire de données.

# 1. Présentation

- ♦ Lors de la création ou de la modification d'un objet, le DD est mis à jour afin de refléter les changements opérés.
- ♦ Les informations du dictionnaire sont stockées dans les tables de base gérées par la base de données Oracle.
- ♦ Vous ne lisez pas directement ces tables mais vous y accédez par le biais de vues prédéfinies.
- ♦ Il existe deux types de vues:
  - ♦ les vues statiques et
  - ♦ les vues dynamiques.
- ♦ La vue **DICTIONARY** inclut des informations sur les vues statiques et dynamiques du dictionnaire de données: C'est la vue d'entrée du dictionnaire - Raccourci : DICT'
- ♦ ou le synonyme DICT' correspondant, contient le nom et la description de tous les éléments inclus dans le dictionnaire de données.
  - ♦ Il suffit de taper `select * from dictionary` pour savoir le nom de toutes les vues prédéfinies

## 2. Les vues statiques du DD

- ♦ Elles sont basées sur des tables créées réellement dans le dictionnaire de données.
- ♦ Les vues **USER** (dont le nom commence par *USER\_*xxx)
  - donnent des informations sur tous les objets logiques dont l'utilisateur connecté est propriétaire (tables, index, vues, procédures, ...)
- ♦ Les vues **ALL** (dont le nom commence par *ALL\_*xxx)
  - fournissent des informations sur les objets pour lesquels l'utilisateur a un droit d'accès, c'est-à-dire les objets de la base créés par l'utilisateur ainsi que tous les objets accessibles par cet utilisateur.
- ♦ Les vues **DBA** (dont le nom commence par *DBA\_*xxx).
  - Ces vues sont réservées à l'administrateur de la base (DBA), afin de lui fournir des informations sensibles sur tous les objets de la base de données.



# Principales vues statiques

- ♦ ALL\_CATALOG Tous les objets (tables, vues,...) accessibles à l'utilisateur -  
USER\_CATALOG Tous les objets qui sont propriétés de l'utilisateur
- ♦ ALL\_CONSTRAINTS Toutes les contraintes d'intégrité (dont référentielles) sur les objets accessibles à l'utilisateur
- ♦ ALL\_TAB\_PRIVS Droits sur les objets accessibles par l'utilisateur. Ce sont les droits qui concernent l'utilisateur, celui qui a accordé les droits, le propriétaire ou le rôle PUBLIC (cf infra GRANT) –
- ♦ ALL\_USERS Informations sur tous les utilisateurs de la base de données. -  
USER\_USERS A votre avis ?

## 3. Les vues dynamiques du DD

- ♦ Préfixées par V\$ (dont le nom commence par V\$\_):
  - Relatives au suivi des performances
  - Sont des vues dynamiques permettant d'avoir des informations sur l'état courant de l'instance de la base de données de son démarrage à son arrêt.
  - Elles permettent par exemple de connaître les fichiers physiques actuellement utilisés par la base (logs, rollback segments, ...).
  - Ne sont accessibles que par les administrateurs.



# Utilisateurs SYSTEME

- ◆ Deux utilisateurs sont remarquables dès l'installation d'Oracle
  - SYS (mot de passe par défaut : CHANGE\_ON\_INSTALL): propriétaire des tables du dictionnaire de données. Il est préférable de ne jamais se connecter sous SYS en ligne.
  - SYSTEM (mot de passe par défaut : MANAGER): le DBA qu'Oracle vous offre.
    - Il vous permettra d'effectuer vos tâches administratives en ligne ou par la console Enterprise Manager (créer des utilisateurs par exemple).
  - Si vous voulez qu'un autre utilisateur prenne cette fonction, il faudra que SYSTEM lui affecte les privilèges à retransmettre (ou le rôle DBA qui contient tous les privilèges système (à voir dans le chapitre contrôle des données))

## Références

### Livres :

1. G. Gardarin, O. Gardarin , Le Client-Serveur , Eyrolles, 2e édition, 1996.
2. Marc Frappier, **Exploitation de bases de données relationnelles et orientées objet**
3. Christian SOUTOU, **SQL pour Oracle, Edition 3 et 7**

### Supports de cours :

1. G. Gardarin, XML et les BD
2. Odile PAPINI, Bases de données. Cours 2 : Architecture pour les bases de données.
3. M. Nakechbnadi, Première partie : Administration d'une BD, Chapitre III Architecture de Base de Données Oracle
4. Virginie Sans , BASE DE DONNÉES OBJET.
5. Cours SGBD 1. Concepts et langages des Bases de Données Relationnelles
6. *Miriano Romualdi* . Introduction aux Systèmes de Gestion de Bases de données. U N I V E R S I T E D E G E N E V E
7. Riadh HADJ M'TIR , *Systèmes de Gestion des Bases de Données*; Institut Supérieur d'Informatique de Mahdia AU : 2011-2012

# Plan du module

## ♦ Partie 1- I. Introduction aux SGBDs

- Chapitre 1: Présentation des SGBDs
- Chapitre 2: Définition et Evolution des données
- Chapitre 3: Contrôle des données
- Chapitre 4: Gestion des objets utilisateurs
  - (Vues, séquences et Index)

## ♦ Partie 2- II. Langage procédural: PL/SQL

## ♦ Partie 3- III. Les Transactions

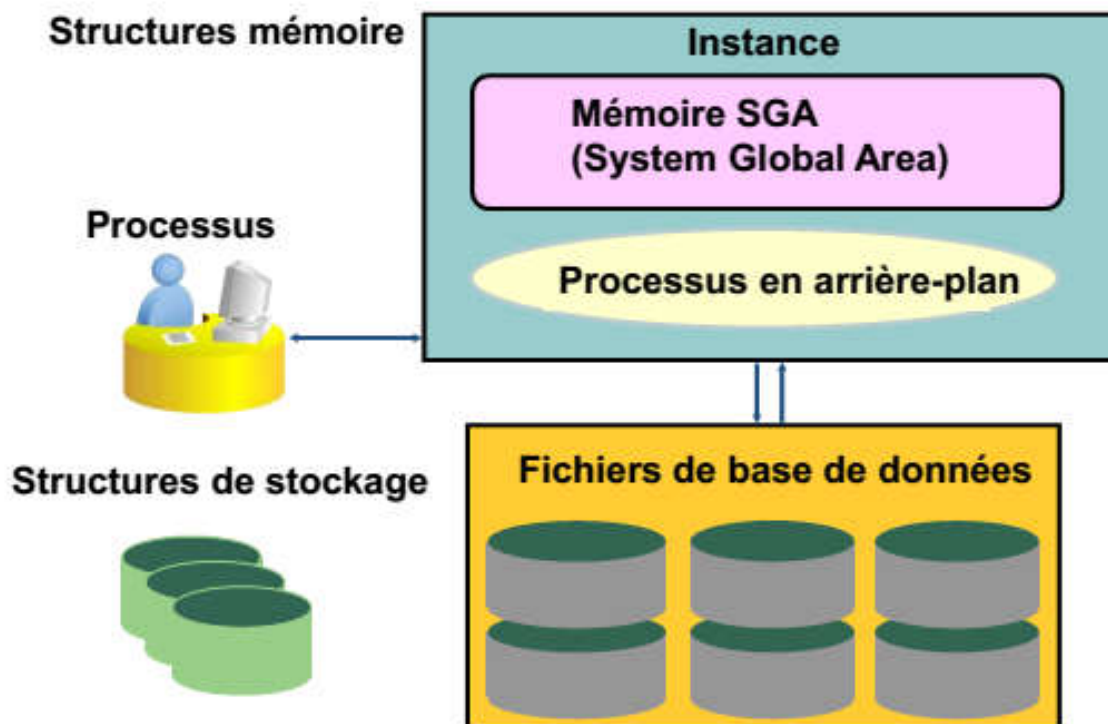
## Partie 4- IV. L'interface JDBC

Connexion à une base et état de connexion

Interaction avec la base et extraction des données

## Annexe: IV- Oracle

# 1- Architecture d'une BD Oracle



# 1- Architecture d'une BD Oracle

## ♦ Structures mémoire

- **Mémoire SGA** (System Global Area) : La SGA est une zone mémoire qui est utilisée par la base de données pour partager les informations entre les différents processus Oracle.
- **Mémoire PGA** (Program Global Area): propre à chaque serveur et processus en arrière-plan: Il existe une mémoire PGA pour chaque processus

## ♦ Processus

- **Processus utilisateur** (Un processus utilisateur est créé pour chaque programme exécuté par un utilisateur )et **processus serveur**
- **Processus en arrière-plan** : SMON, PMON, DBWn, CKPT, LGWR, ARCn, etc.

## ♦ Structures de stockage

- **Structures logiques** : base de données, schémas, tablespaces, segments, extents et blocs Oracle.
- **Structures physiques** : fichiers de données, fichiers de paramètres, fichiers de journalisation et blocs du système d'exploitation

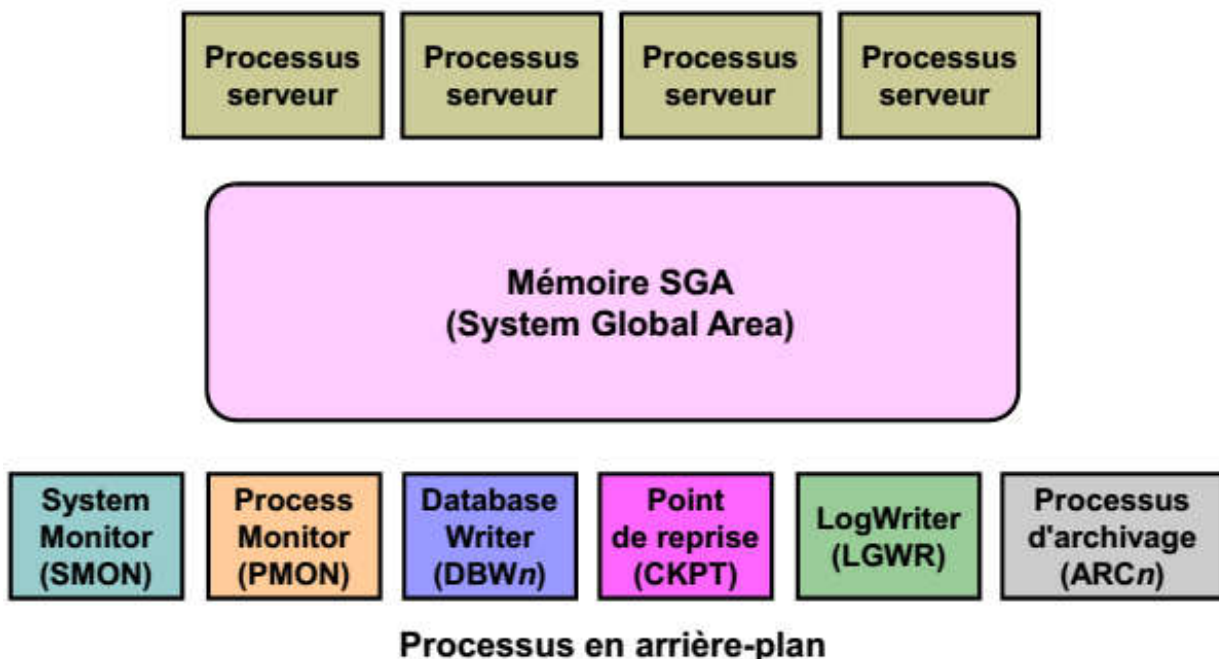
## 2- La mémoire SGA

- ♦ La mémoire SGA contient plusieurs structures dont les plus importantes sont:
  - **cache des blocs de donnée (buffer cache) :**  
est un espace mémoire contenant toutes les données transitant de ou vers la BD.
  - **Tampon de journalisation (redo log buffer) :**
  - contient les blocs de données à modifier et les modifications à effectuer sur ces données, avant que l'ensemble de ces mises à jour de la base ne soient archivées dans les fichiers Redo-log.
    - L'ensemble des tailles des caches peut être modifié (augmentée ou diminuée) grâce aux paramètres du fichier d'initialisation (init.ora).
  - **Zone de mémoire partagée (Shared Pool) :** cette zone sert à mémoriser et traiter les requêtes SQL provenant des divers utilisateurs.
  - **Zone de mémoire LARGE POOL :** zone facultative fournissant d'importantes allocations mémoire pour certains processus utilisant beaucoup de mémoire;

2<sup>ème</sup> Ing.Inf

46

## 3- Processus Oracle



2<sup>ème</sup> Ing.Inf

47

## 3- Processus Oracle

- ♦ **System Monitor (SMON)** : effectue une récupération après panne lorsque la BD est démarrée après une défaillance.
- ♦ **Process Monitor (PMON)** : effectue un nettoyage lorsqu'un processus utilisateur échoue (il supprime les processus en erreur, annule les transactions n'ayant pas été validées....)
- ♦ **Database Writer (DBWn)** : transfère les blocs modifiés du cache de tampons de la base (RedoLog buffer de l'SGA) suites à des requêtes SQL dans des fichiers de données stockés sur disque.
- ♦ **Point de reprise (CKPT)** : met à jour l'ensemble des fichiers de données et de contrôle de la base afin d'indiquer le point de reprise le plus récent.
- ♦ **LogWriter (LGWR)** : écrit les entrées de journalisation sur le disque.
- ♦ **Processus d'archivage (ARCn)** : copie les fichiers de journalisation (fichiers redo log) dans le lieu destiné au stockage des archives lorsqu'un changement de fichier de journalisation se produit.

48

## 4- Gestion d'une instance Oracle

Une instance Oracle

=

Ensemble des zones mémoires (SGA)

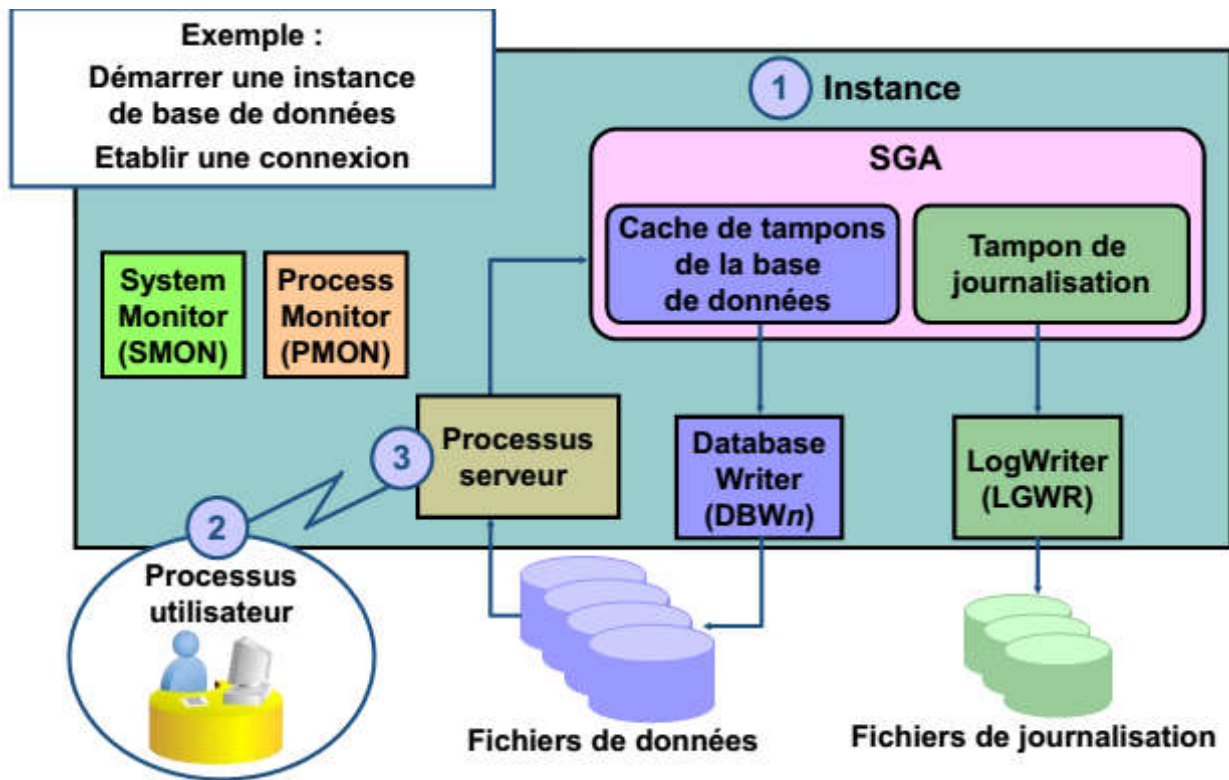
+

Processus alloués à une base de données

49



## 4- Gestion d'une instance Oracle



## 4- Gestion d'une instance Oracle

- ◆ 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.

## 4- Gestion d'une instance Oracle

### ♦ 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 stockées dans la mémoire SGA

## 4- Gestion d'une instance Oracle

- 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 DBW<sub>n</sub> 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.



## 5- Structure physique d'une BD oracle



## 5- Structure physique d'une BD oracle

- ◆ Les fichiers constituant une base de données Oracle sont organisés de la façon suivante :
  - **Fichiers de contrôle** : contiennent des données sur la base elle-même (informations sur la structure physique de la base de données). Ces fichiers sont d'une importance capitale pour la base. Sans eux, vous ne pouvez pas ouvrir de fichiers de données pour accéder aux données de la base.
  - **Fichiers de données** : contiennent les données utilisateur ou les données d'application de la base.
  - **Fichiers de journalisation en ligne** : permettent la récupération d'une instance de base de données. S'il se produit une panne de la base sans perte des fichiers de données, l'instance peut récupérer la base grâce aux informations contenues dans ces fichiers.
  - **Le fichier d'initialisation**  
Ce fichier est un fichier au **format texte** de paramètres de démarrage de la base (il est généralement nommé **initSID.ora**. Ce fichier peut être facilement **reconstruit** car **un nombre limité de paramètres est réellement utile**.

# 5- Structure physique d'une BD oracle

- Exemple : (init.ora) :
  - db\_name = DB01
  - db\_files = 20
  - control\_files =  
/home/oracle/ORADBA/DB01/DATABASE/ctl1db01.ora
  - db\_block\_buffers = 200
  - db\_block\_size = 2048