

Administration de bases de données Oracle

Cycle Ingénieur Applications Web et Mobiles -Semestre 08

Ecole d'Ingénierie Digitale et d'Intelligence Artificielle
Université Euro-Méditerranéenne de Fès

Pr. Abderrahim El Qadi
Département Mathématique Appliquée et Génie Informatique
ENSAM, Université Mohammed V de Rabat

A.U. 2023/2024

Références

- www.oracle.com
- Oracle 10g Administration, Olivier Heurtel, 2005, 489p.
- Oracle11g Administration, Razvan Bizoï, Eyrolles, 2011, ISBN :
978-2-212-12898-7
- Oracle 11g Exploitation, Clotilde Attouche, Société TELLORA.
- Oracle11g Sauvegarde et sécurité. Razvan Bizoï, Eyrolles, 2011.
- Oracle 11g New Features for Administrators Summary Sheets, Version 1.2, Ahmed Baraka.
- Administration Oracle 10G Partie I. G. Mopolo-Moké, prof. MBDS / UNSA NICE 2005/ 2006

Plan

Partie 1

1. Introduction

- Rappel sur le modèle relationnel, et le langage SQL
- Présentation d'Oracle
- Historique, versions et licences d'Oracle
- Les outils d'administration : SQL*Plus, Oracle SQL Developer, Oracle Enterprise Manager
- Principales étapes de l'installation.
- Configuration Post-installation
- Architectures du Serveur Oracle
- Configuration Oracle Net

2. Organisation d'une base de données oracle

- Architecture générale
- Notion d'instance
- Structure mémoire : SGA
- Les buffers DATABASE, et Le buffer REDO LOG.
- La zone de partage des requêtes SQL
- Les process : USER, SERVER, Process de fond
- Mécanismes de lecture et écriture des requêtes SQL.

3. Architecture de base

- Les fichiers de la base : data files, redo log files, control files, et init.ora
- Dictionnaire de données

4. Création de la base de données et gestion d'instance

- Étapes de création d'une nouvelle base de données
- Différentes méthodes de création d'une nouvelle base de données
- Gestion des paramètres d'initialisation.
- Différents statuts d'une base

5. Gestion de l'espace physique

- Les types de tablespace.
- La création d'un tablespace permanent, temporaire.
- Mise à jour d'un tablespace.
- Segment et leurs composants

6. Gestion des utilisateurs

- Création d'un utilisateur.
- Gérer les droits / privilèges
- Gérer les rôles
- Gestion des profils et mots de passe.
- Gestion de l'allocation des tables
- Les vues et autres objets

Partie 2 :

7. Audit d'une base Oracle

- Types d'audits
- Autorisation de Mise en route de l'audit
- Déclenchement effectif de l'audit
- Désactivation de l'audit sur les Ordres SQL et les privilèges systèmes

8. Gestion des fichiers de contrôle et des fichiers de journalisation

- Gérer les fichiers de contrôle.
- Gérer les fichiers de journalisation.
- Le rôle des fichiers de journalisation.

9. Sauvegarde et restauration.

- Solutions de sauvegarde et récupération
- Stratégies de sauvegarde
- Utilitaire RMAN
- Archivage des fichiers de journalisation

10. Transfert et chargement de données

- Les différents modes d'export et d'import
- Sqlloader et chargement de données
- Transfert de données Oracle-MySQL

11. Optimisation SQL

- Génération d'un Plan d'exécution
- Éléments d'optimisation : Index, vues matérialisées
- Optimiser les performances de l'instance

1. Introduction : Système de gestion de Base de Données (SGBD) Oracle

- **Base de données (BD)** : ensemble structuré de données cohérentes.

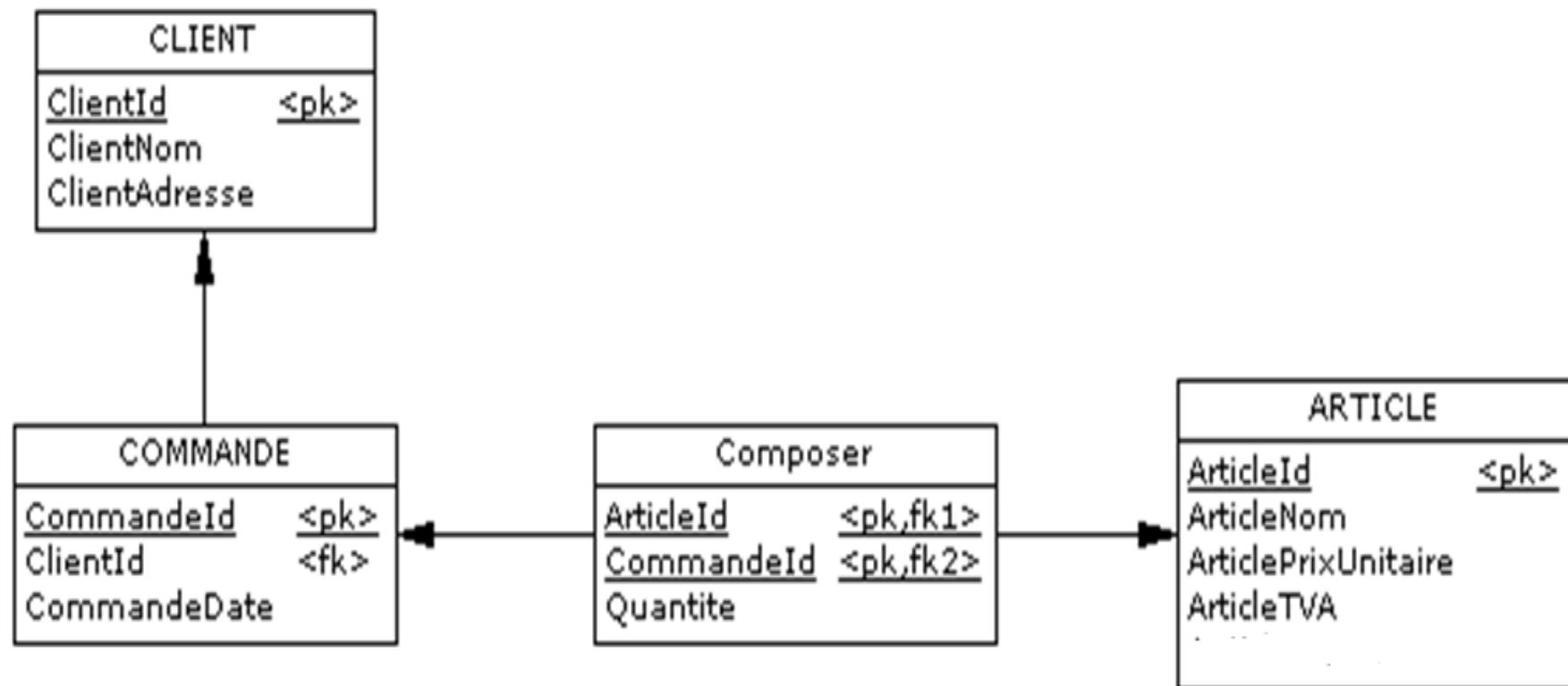
Une BD est un ensemble de relations, Chaque relation est un sous-ensemble de données caractérisé par des attributs et visualisable sous forme de table.

The diagram illustrates the structure of a table. It features a table with four columns and three rows. The columns are labeled 'N°', 'Nom', 'Prénom', and 'Age'. The rows contain numerical IDs, names, first names, and ages respectively. Above the table, the word 'Colonnes' is written with arrows pointing to each of the four columns. To the right of the table, the word 'Lignes' is written with arrows pointing to each of the three rows. The entire diagram is enclosed in a rectangular box.

N°	Nom	Prénom	Age
1	MBOUZANG	Franck	18
2	KENFACK	Stéphane	22
3	DONGO	Jules	19

Structure d'une table

— Modèle relationnel



- **SGBDR** : Système de Gestion de Bases de Données Relationnel (ou RDBMS en Anglais) : SGBD basé sur le modèle relationnel.

DBMS - Most Popular Database Management Systems

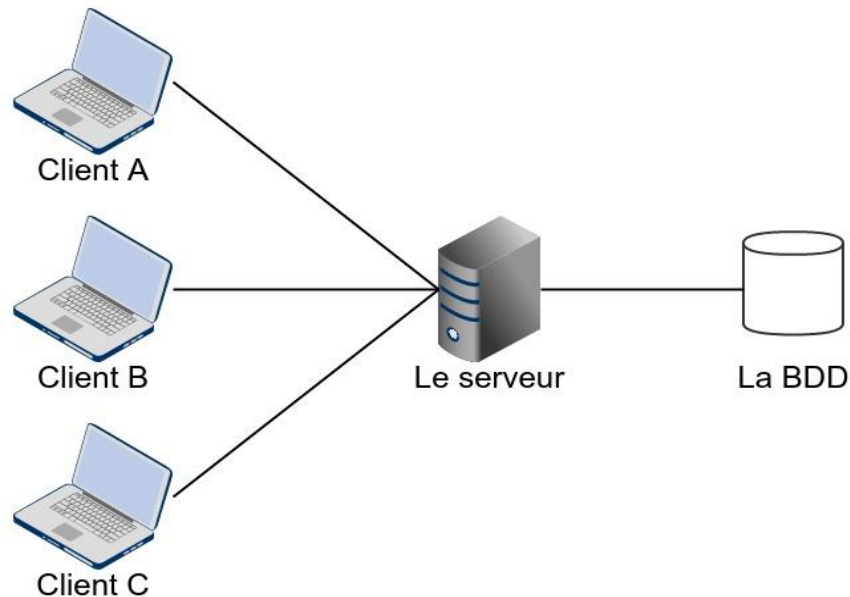




Oracle (<https://www.oracle.com>) est un **SGBD** édité par la société Oracle Corporation – leader mondial des bases de données. Sa technologie de base de données est largement utilisée dans le monde entier.

- *Oracle Database : Le système de gestion de base de données relationnelle phare d'Oracle.*
- *Oracle Exadata : Une plateforme de base de données optimisée pour les charges de travail à haute performance.*
- *MySQL : Une base de données open source, également propriété d'Oracle.*

- La force principale d'Oracle est la portabilité de sa base de données : oracle disponible sur de nombreuses plates-formes matérielles : (Solaris (Sun) / HP/UX (Hewlett Packard) / Aix (IBM) et Windows NT (Microsoft)
- Oracle a fait sa réputation comme serveur de données : les programmes fonctionnent sur les clients et interagissent avec la base de données sur le serveur.



— Les versions d'Oracle :

Oracle Corporation est une entreprise de technologie qui propose une gamme étendue de produits et services dans plusieurs versions. Voici quelques-uns des versions :

- Base de données Oracle 11g :

- Cette version a été publiée pour la première fois en 2007. Elle a introduit des fonctionnalités telles que la gestion automatique des espaces de stockage, la compression de tables, et des améliorations dans la gestion des performances.

- Base de données Oracle 12c :

- Publiée en 2013, la version "c" dans 12c signifie "Cloud". Cette version met l'accent sur le cloud computing et permet de gérer plusieurs bases de données en tant que conteneurs uniques.

- Base de données Oracle 18c :

- Il s'agit de la version 12.2.0.2 mais avec la nouvelle convention de version.
- Cette version, publiée en 2018, a introduit des fonctionnalités liées à la gestion autonome des bases de données et a marqué un changement dans la numérotation des versions, en passant de la numérotation classique "g" à une numérotation basée sur l'année de sortie.

- Base de données Oracle 19c :

- Sortie en 2019, Oracle Database 19c met en avant des fonctionnalités telles que la gestion autonome, la sécurité renforcée et des améliorations de performances.

- Base de données Oracle 21c :

- Livrée en premier dans le cloud public, la version 21c de la base de données d'Oracle **étend les types de données qu'elle gère, avec l'ajout de tables blockchain et un support étendu de JSON**. Elle optimise sa gestion de la mémoire sur le in-memory et avec le support de la mémoire persistante.

— **Détail d'un numéro de version (12.2.0.2)**

"Oracle Database 12.2.0.2" représente la deuxième version mineure de la version majeure 12, avec une version de maintenance initiale, et le deuxième correctif appliqué à cette version.

- **12** : Numéro de version majeure.
- **2** : Numéro de version mineure - il s'agit de la deuxième version mineure de la version majeure 12.
- **0** : Numéro de maintenance (0) - Cela représente le niveau de maintenance ou de patch. Dans ce cas, il est indiqué comme 0, ce qui signifie qu'il s'agit de la version initiale ou de la première version de maintenance de la version mineure.
- **2** : Numéro du correctif - cela peut indiquer un correctif spécifique ou une mise à jour mineure au sein de la version de maintenance.




— Secteurs d'activité

- Oracle propose une gamme étendue de produits et services dans plusieurs secteurs d'activité. Voici quelques-uns des principaux secteurs où Oracle est activement impliqué :

Informatique d'Entreprise (Enterprise IT)	Oracle fournit des bases de données et des solutions de gestion de données pour les entreprises.
Cloud Computing	Oracle propose des services de cloud computing, notamment des solutions de cloud public (Oracle Cloud Infrastructure), de cloud privé et des services de cloud hybride.
Logiciels d'Entreprise (Enterprise Software)	Oracle offre une suite complète de logiciels d'entreprise, y compris des applications ERP (Enterprise Resource Planning), CRM (Customer Relationship Management), SCM (Supply Chain Management), et HCM (Human Capital Management).
Services Financiers	Oracle propose des solutions logicielles spécifiques pour les institutions financières, y compris des applications de gestion des risques, de gestion de la conformité, et des solutions de paiement
Santé	Solutions pour la gestion des dossiers médicaux électroniques (DME), la gestion des soins de santé, la recherche clinique, etc.

Éducation	Oracle fournit des solutions technologiques pour les institutions éducatives, y compris des systèmes de gestion des campus, des solutions d'apprentissage en ligne, et des outils d'analyse de données éducatives.
Industrie Hôtelière et Touristique	Solutions technologiques pour l'industrie hôtelière et touristique, y compris des systèmes de gestion hôtelière, de réservation en ligne et de gestion de la relation client
Télécommunications	Solutions pour les opérateurs de télécommunications, y compris des logiciels de facturation, des solutions de gestion des réseaux, et des services cloud pour les télécommunications
Industrie des Transports et de la Logistique	Oracle propose des solutions pour la gestion des opérations de transport, la logistique, et la gestion des chaînes d'approvisionnement

– Les Outils et environnement de développement Oracle

 <p>SQL*Plus</p>	<p>Utilitaire en ligne de commande développé par Oracle qui permet aux utilisateurs d'exécuter interactivement des commandes SQL et PL/SQL</p>
<p>Oracle SQL Développeur</p> 	<p>Outil de développement de bases de données pour les développeurs SQL et PL/SQL</p>
<p>Oracle JDeveloper</p>	<p>Environnement de développement intégré (IDE) pour Java.</p>
<p>Oracle SQL Developer Data Modeler</p>	<p>les utilisateurs peuvent créer, parcourir et modifier des modèles logiques, relationnels, physiques, multidimensionnels et de type de données.</p>
<p>Oracle WebLogic Server</p>	<p>Serveur d'applications Java EE</p>
<p>Oracle Fusion Middleware</p>	<p>Ensemble d'outils et de services pour le développement, l'intégration et la gestion des applications</p>
<p>Oracle Analytics Cloud</p>	<p>Plateforme d'analyse et de business intelligence dans le cloud.</p>
<p>Oracle Business Intelligence Enterprise Edition (OBIEE)</p>	<p>Suite complète pour l'analyse des données.</p>
<p>Oracle APEX</p> 	<p>est la plate-forme d'applications low-code d'entreprise la plus populaire au monde qui permet de créer des applications Web et mobiles évolutives et sécurisées ;</p>

– Installation Oracle Database 19c sous Windows

– Oracle Server Standard Edition :

- Cette édition est conçue pour répondre aux besoins de base des petites et moyennes entreprises. Elle offre les fonctionnalités essentielles de la base de données Oracle, mais avec des limitations par rapport à l'édition Enterprise. Par exemple, il peut y avoir des restrictions sur le nombre de processeurs ou de sockets pris en charge.

– Oracle Server Enterprise Edition :

- Il s'agit de l'édition la plus complète d'Oracle Database. Elle offre toutes les fonctionnalités avancées, outils et capacités de gestion de base de données d'Oracle. L'édition Enterprise est conçue pour répondre aux besoins des entreprises de grande envergure, offrant une évolutivité maximale, des fonctionnalités avancées de sécurité, de gestion des performances et d'intégration.

- L'installation d'Oracle Database 19c sous Windows peut être un processus complexe, mais Oracle propose un assistant d'installation graphique qui simplifie la tâche.

Voici une procédure générale pour installer Oracle Database 19c sous Windows:

Prérequis :

1. Configuration du Système :

- Vérifiez que votre système Windows répond aux spécifications requises pour Oracle Database 19c.
- Assurez-vous que le compte utilisateur a les privilèges d'administrateur.

2. Téléchargement des Fichiers d'Installation :

- Téléchargez le programme d'installation d'Oracle Database 19c depuis le site officiel d'Oracle.

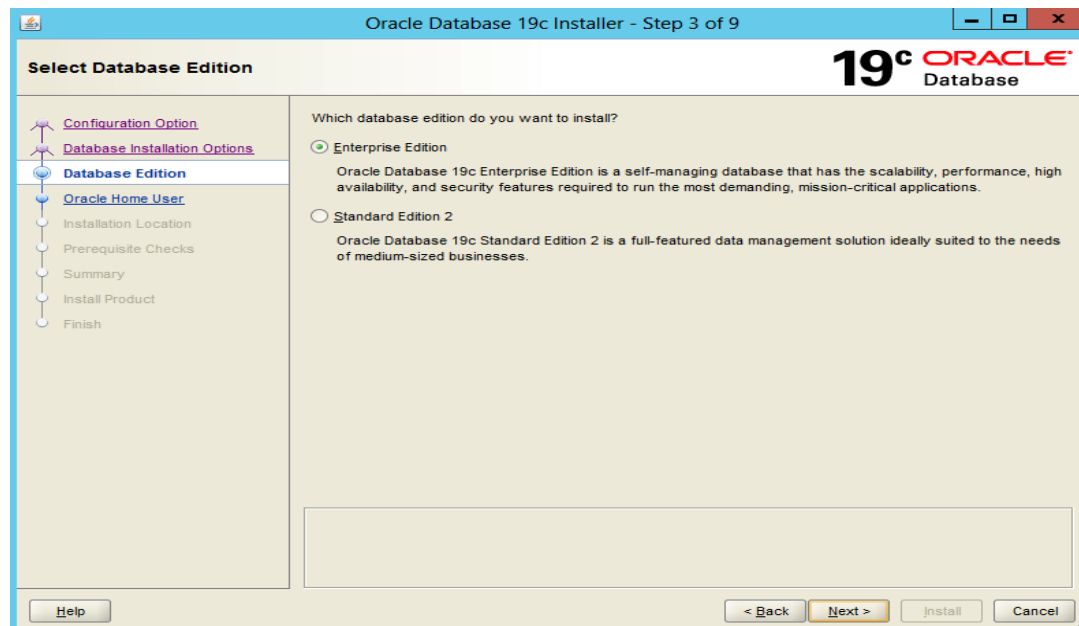
Étapes d'Installation :

1.Extraction des Fichiers :

- Extrayez les fichiers du programme d'installation téléchargé dans un répertoire sur votre système.

2.Lancement de l'Assistant d'Installation :

- Allez dans le répertoire où les fichiers ont été extraits.
- Localisez le fichier exécutable "setup.exe" et lancez-le.



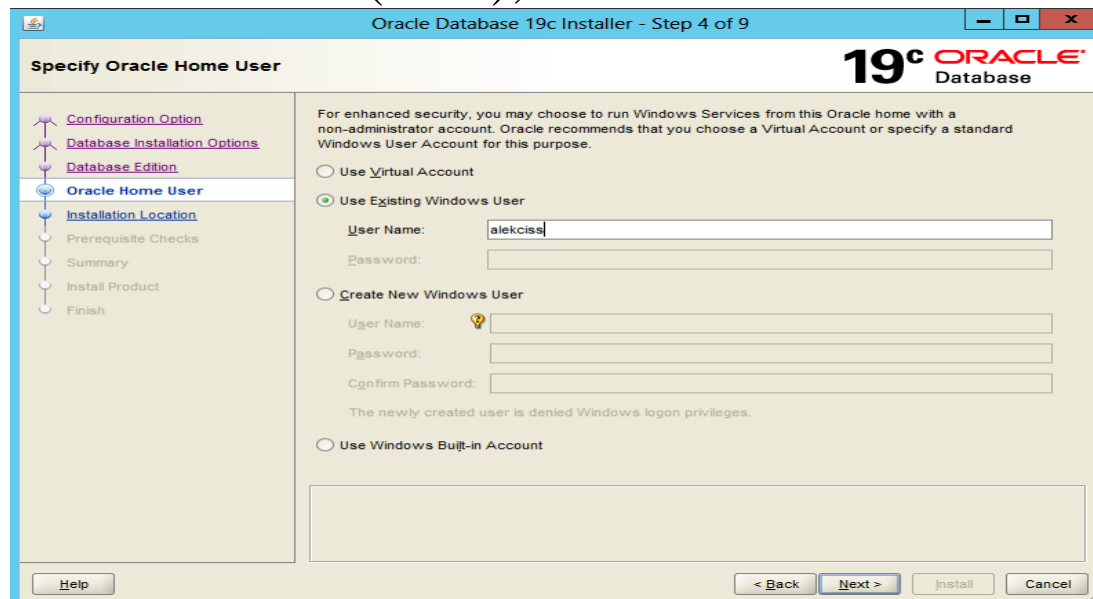
3.Étapes d'installation :

Les options d'installation de la base de données

- Installer le logiciel Oracle et créer une base de données en même temps (Créer et configurer une base de données à instance unique).
 - Cette option permet d'installer une base de données à instance unique sur un seul serveur.
 - Avec l'option Oracle Real Application Clusters (RAC) permet à plusieurs instances s'exécutant sur des serveurs différents d'accéder à la même base de données physique stockée sur un stockage partagé.
 - Suivez les étapes de l'assistant, en fournissant les informations requises telles que le type d'installation (typique, personnalisé, etc.), le chemin d'installation, et les paramètres de la base de données.

4. Configuration des Options d'Installation :

- Configurez les options spécifiques telles que le type de base de données (OLTP, Data Warehouse), le jeu de caractères, le mot de passe du compte administrateur (SYS), etc.



5. Configuration des Composants Additionnels :

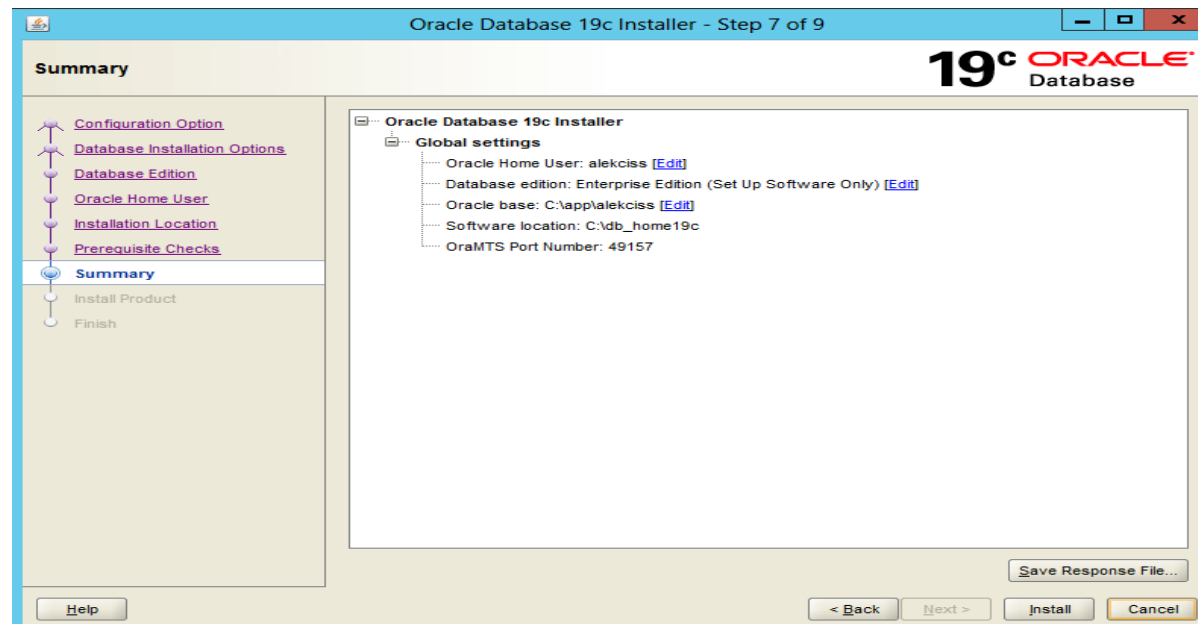
- Configurez les composants additionnels selon vos besoins, comme Oracle Text, Oracle Spatial, etc.

6. Configuration du Réseau :

- Configurez les paramètres de réseau, tels que le nom de la base de données (SID), le port listener, etc.

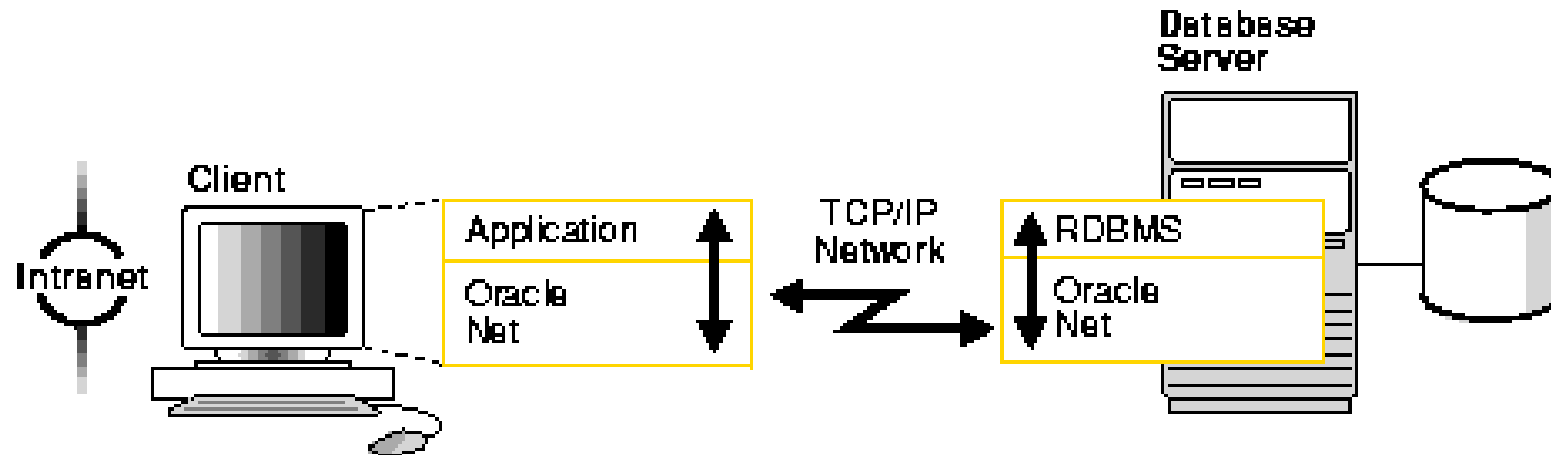
7. Vérification des exigences minimales, résumé et fin de l'installation

- Après ces phases initiales de configuration, vérifions les prérequis d'installation. Si, toutefois, il y a des erreurs, essayez de réajuster les exigences minimales d'installation et recommencez.
- Cliquez sur Installer si aucune erreur ne se produit.

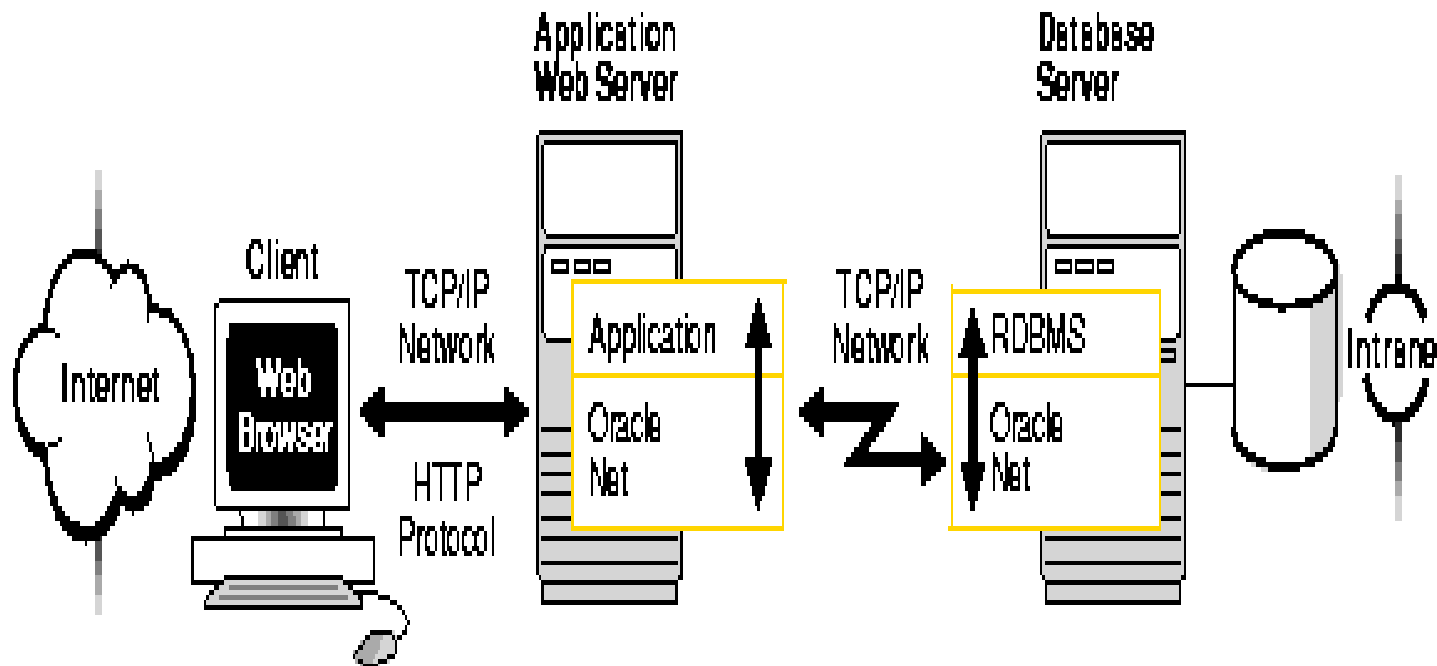


– Architectures du Serveur Oracle

- 3 types d'architecture globale de systèmes d'information basés sur Oracle :
 - Architecture locale : Tout est sur le même serveur matériel, programme client et serveur de données Oracle.
 - Architecture client/serveur : On a un programme client (un exécutable) sur le poste client, dit alors 'client lourd'. Le PC communique avec le serveur de données sur un serveur distant, via le réseau, et la couche Oracle Net.

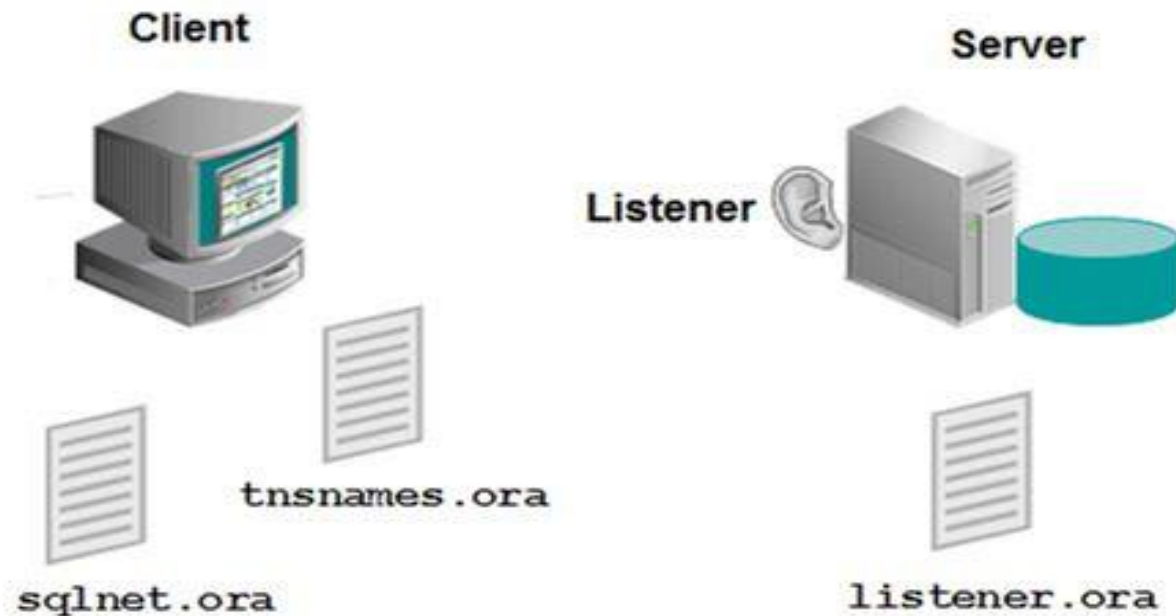


- Architecture 3 tiers : Pas de programme client. Un navigateur suffit sur le poste de travail (dit alors ‘client léger’). Il dialogue avec le serveur de données distant via http.



– Configuration Oracle Net

- Oracle utilise trois fichiers (listener.ora, tnsnames.ora et sqlnet.ora) pour la configuration du réseau
- Les fichiers de configuration coté client et/ou serveur se trouvent dans le répertoire ORACLE_HOME/network/admin



—Le programme client doit notamment pouvoir identifier le serveur de données Oracle sur le réseau. Il utilise pour ce faire une résolution de nom de serveur.

—Les méthodes de résolution de nom

- 4 modes de nommage :
 - Local : on utilise un fichier de configuration situé sur le poste client : TNSNAMES.ORA
 - Via un annuaire : on utilise un serveur d'annuaire compatible LDAP : Oracle Internet Directory par exemple...
 - Basé uniquement sur TCP/IP (exceptionnellement pas de couche spécifique Oracle Net nécessaire coté client) : On utilise une chaîne de connexion explicite, qui devra préciser le serveur (machine) cible, et éventuellement le port d'écoute du logiciel serveur Oracle Net, et un nom de service de données.

Exemple :

CONNECT user1/pwd@serveur1:1521/orcl

Le port 1521 est le port par défaut du listener Oracle Net.

- Externe : on utilise un service de nom externe, non Oracle mais compatible avec ce dernier (NIS, DCE)

—Fichier de configuration client : TNSNAMES.ORA

- Ce fichier contient le nom de service de réseau
- Un nom de service net est un alias mappé à une adresse réseau de base de données contenue dans un descripteur de connexion.
- Un descripteur de connexion contient l'emplacement de l'auditeur via une adresse de protocole et le nom de service de la base de données à laquelle se connecter.

Exemple d'une entrée 'ORCL_NET' qui pointe sur la BD 'ORCL' du serveur 'LocalHost'

```
ORCL_NET = (DESCRIPTION =  
  (ADDRESS = (PROTOCOL = TCP)(HOST = LOCALHOST)(PORT = 1521))  
  (CONNECT_DATA = (SERVER = DEDICATED)  
    (SERVICE_NAME = ORCL)  
  )  
)
```

—Fichier de configuration serveur : LISTENER.ORA

- Ce fichier contient des paramètres de configuration du réseau côté serveur

Exemple :

```
LISTENER
(DESCRIPTION-LISTE
  (DESCRIPTION)
    (ADDRESS - (PROTOCOL et IPC) (KEY et EXTPROC1))
    (ADDRESS - (PROTOCOL et TCP)(HOST - myserver.example.com) (PORT: 1521))
  )
)
```

- Après la modification du fichier "listener.ora", l'admin doit être redémarré ou rechargé pour permettre à la nouvelle configuration de prendre effet.

C:\>lsnrctl stop

C:\>lsnrctl start

—Fichier **SQLNET.ORA**

- Ce fichier contient des paramètres de configuration du réseau côté client.
- Ce fichier sera également présent sur le serveur si des connexions de style client sont utilisées sur le serveur lui-même, ou si une configuration de connexion de serveur supplémentaire est nécessaire

Exemple :

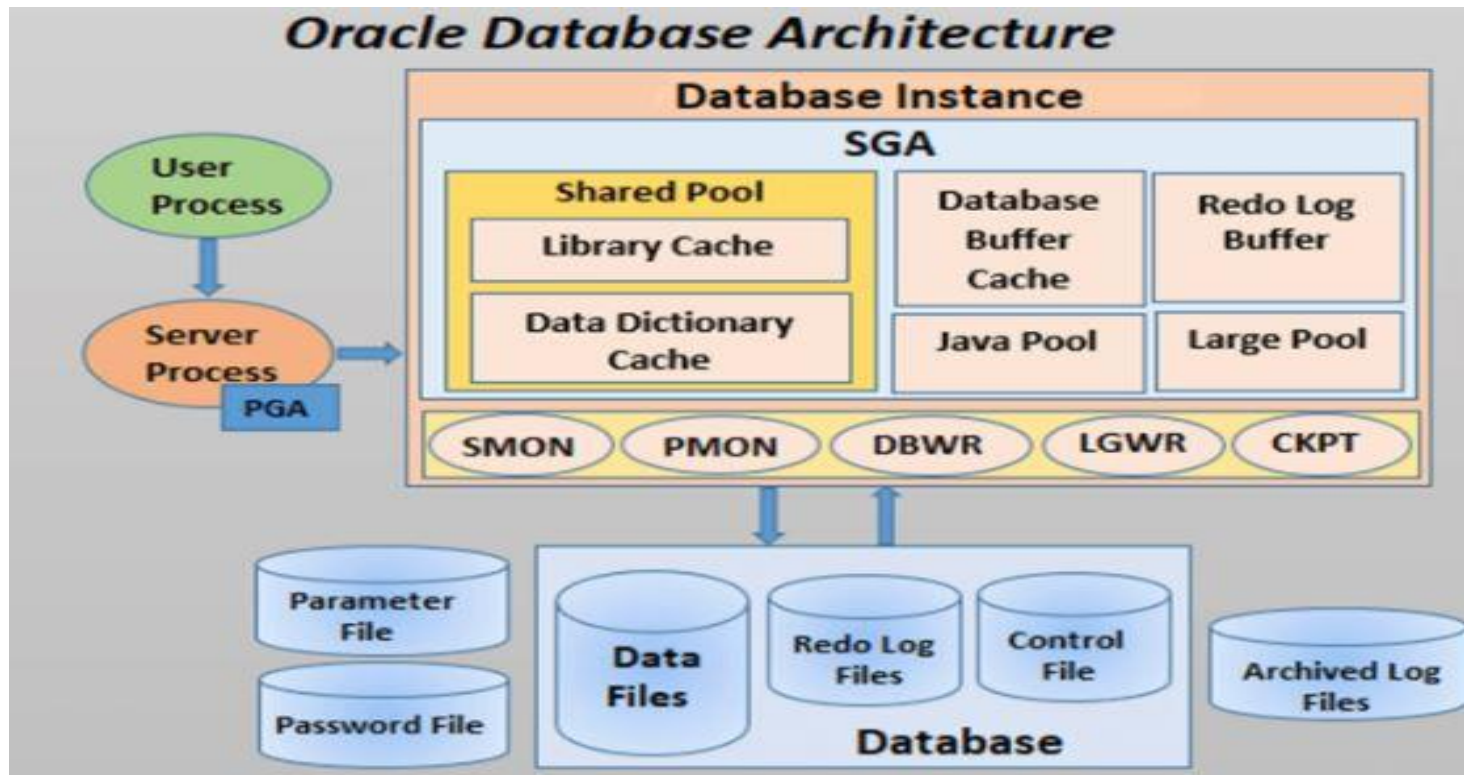
```
NAMES.DIRECTORY_PATH= (TNSNAMES, EZCONNECT)
```

--La rubrique suivante est nécessaire sur Windows si une authentification du système d'exploitation est requise.

```
SQLNET.AUTHENTICATION_SERVICES= (NTS)
```

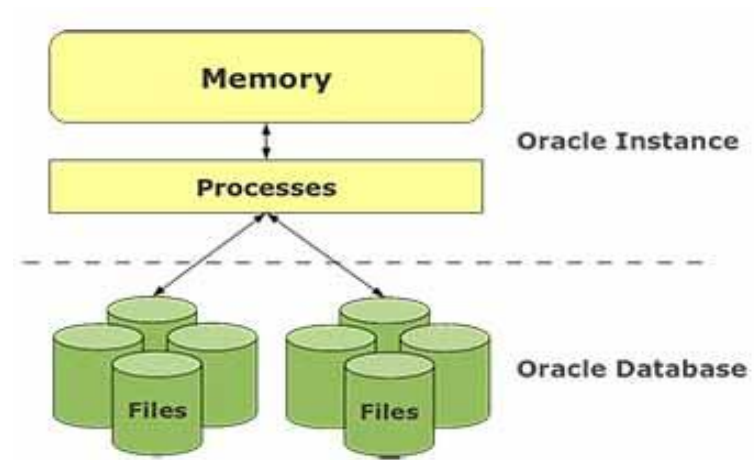
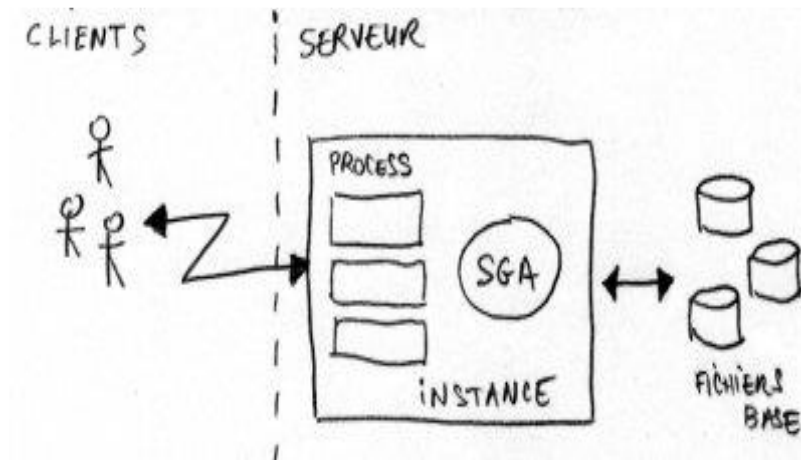
2. Architecture standard d'une BD Oracle

- Une base de données Oracle est un objet riche et complexe composé physiquement :
 - D'une INSTANCE (zone mémoire partagée + processus de fond)
 - D'un certain nombre de FICHIERS PHYSIQUES, totalement indépendants du nombre de table.



— Instance Oracle

- A chaque démarrage d'une base, **une mémoire SGA** (System Global Area) est allouée et un ensemble de **BACKGROUND PROCESS** est démarré.



- La combinaison SGA+BACKGROUND PROCESS est appelée :
« **Instance** ».
- Le nom de l'instance est dépendant d'une variable d'environnement **ORA_SID** (ou ORACLE_SID).
- Pour changer d'instance, il suffira de changer la valeur de cette variable.

C:> **set oracle_sid=info**

— Structure mémoire Oracle

- La structure mémoire d'ORACLE permet de conserver toutes les informations de façon à ce qu'elles puissent être partagées entre tous les PROCESS.
- Elle est constituée essentiellement d'une zone appelée **SGA** (System Global Area), et des zones mémoires dédiées aux utilisateurs : les **PGAs** (Private Global Area).
- La SGA est allouée (sa taille dépend des paramètres d'initialisation) au démarrage de l'instance. On voit ci-après les infos affichées lors du démarrage :
SQL> startup

- Modes de gestion de taille de la SGA :

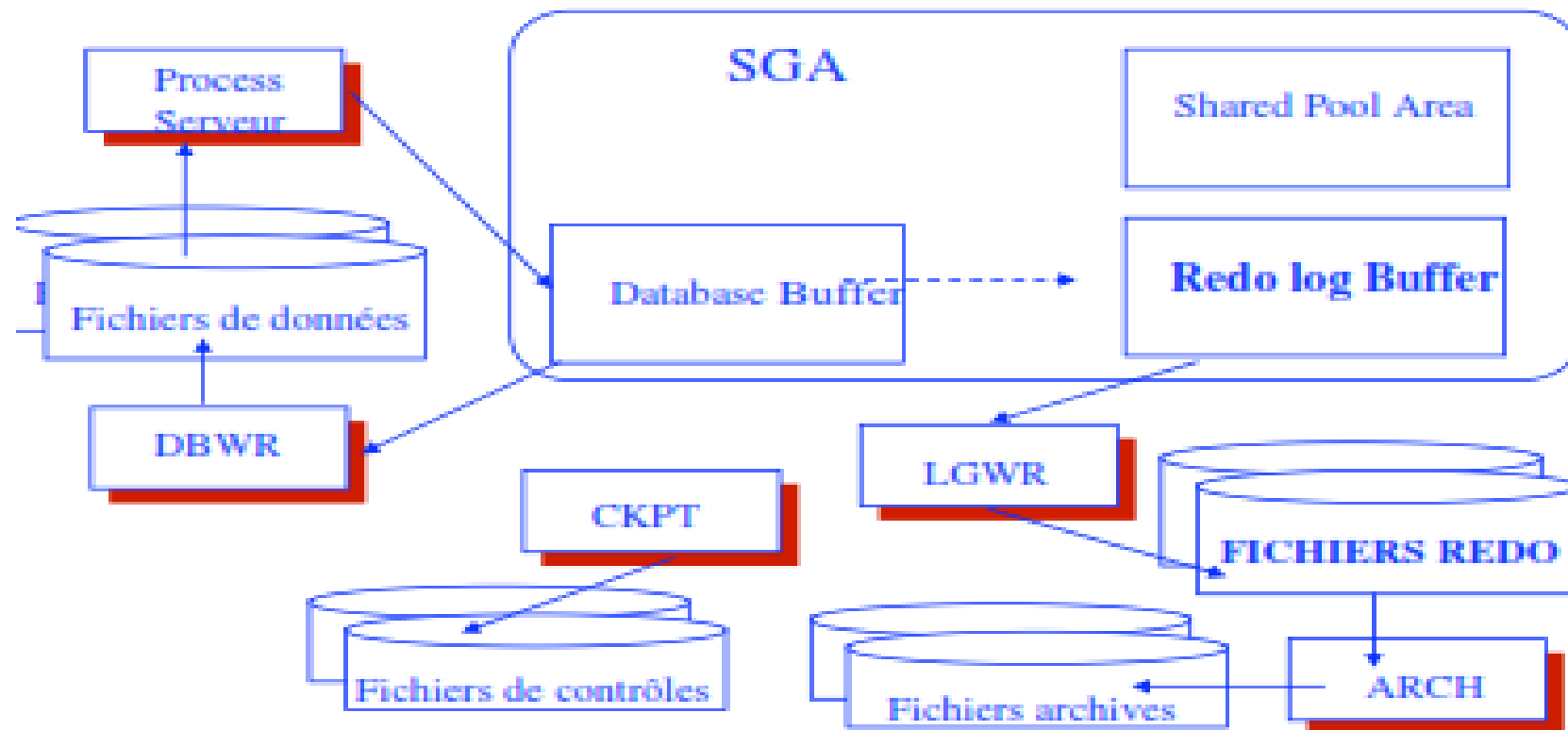
Deux modes de gestion de taille de la SGA, la gestion manuelle et la gestion automatique.

- Dans le mode manuel, c'est le DBA qui analyse le fonctionnement de la base et détermine quelle valeur affecter à chacun des paramètres de taille des composants.
- Dans le mode automatique, c'est Oracle qui calcule et ajuste automatiquement, en fonction de règles et de résultats statistiques, les tailles des différents composants de la SGA et conséquemment sa taille globale.
- Pour passer en mode automatique le paramètre SGA_TARGET doit être positionné à une valeur différente de 0, ou en calculant la taille réellement utilisé à l'instant T :

```
SQL>SELECT SUM(value) FROM v$SGA;
```

```
SQL>SELECT current_size FROM  
V$SGA_DYNAMIC_FREE_MEMORY;
```

- La SGA est un ensemble de buffers qui contient des données utilisateurs et des données système.



- **Les buffers DATABASE** : 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, ...)
- des blocs des rollback segments

– **Le buffer REDO LOG** contient :

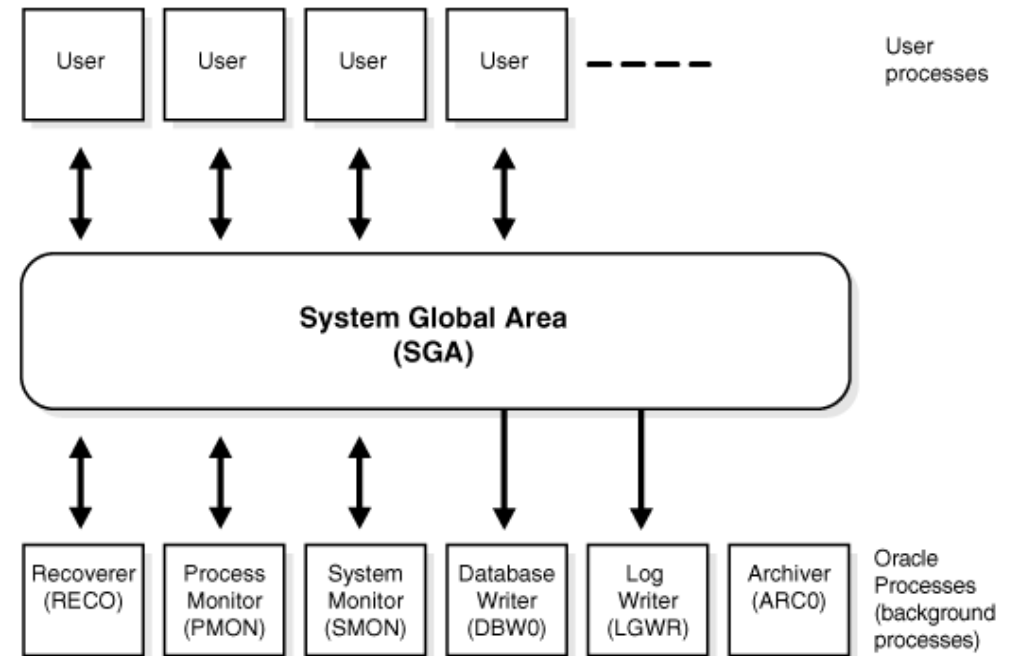
- toutes les données avant leur mise à jour,
- toutes les modifications effectuées sur ces données,
- la trace de toutes les transactions validées ou non encore validées.

- **La zone de partage des requêtes 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).

NOTE : le dba peut nettoyer le buffer via la commande :

Alter System Flush Shared Pool

— Les process



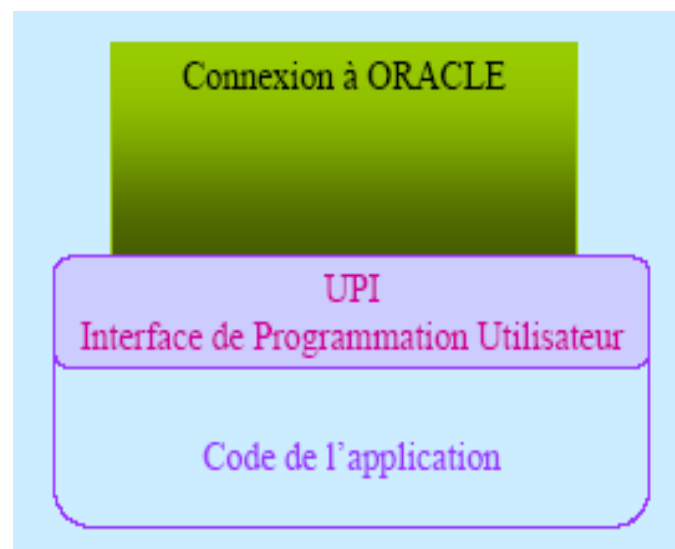
Deux types de PROCESS

- Les PROCESS USER : sont créés pour exécuter le code d'une application ou d'un outil ORACLE.
- Les PROCESS du noyau sont activés automatiquement par l'appel d'autres PROCESS et divisés en deux catégories :
 - Les PROCESS SERVER qui prennent en charge les demandes des utilisateurs,
 - Les BACKGROUND PROCESS qui prennent en charge la gestion des mécanismes internes ORACLE.

1. Les process USER

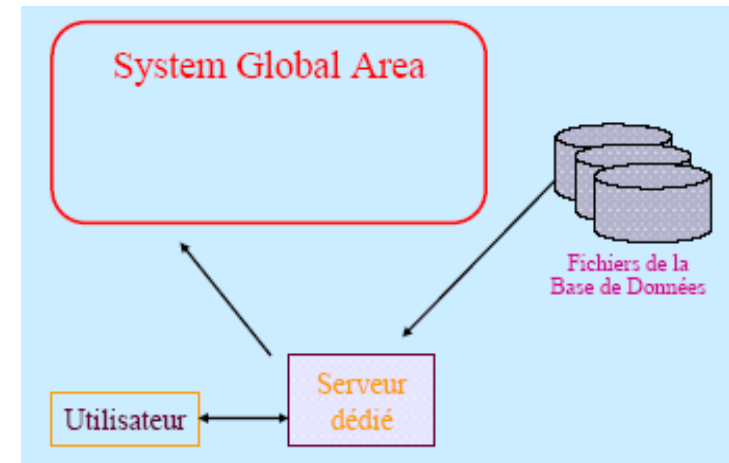
Une session utilisateur est une connexion à la base de données par l'intermédiaire d'un PROCESS USER

Exemple : Quand un utilisateur se connecte à travers SQL*Plus, cet utilisateur doit fournir son nom et son mot de passe. Une session est alors ouverte.



2. Les process SERVER

Le noyau crée un PROCESS SERVER pour prendre en charge les demandes des PROCESS USER.



Le PROCESS SERVER est responsable de la communication entre la mémoire SGA et le PROCESS USER.

- *analyse et exécute les ordres SQL,*
- *lit les fichiers DATABASE et ramène les blocs de données en SGA,*
- *retourne le résultat au PROCESS USER.*

3. Les process de fond

Ils permettent de paralléliser et désynchroniser les accès multi-utilisateur à la base.

Les principaux :

- **DBWR** (*DataBase Writer* ou *Dirty Buffer Writer*), le processus chargé d'écrire le contenu des buffers dans les fichiers de données
- **LGWR** (*Log Writer*), le processus chargé d'écrire le contenu des buffers dans les fichiers Redo Log
- **PMON** (*Process Monitor*), le processus chargé de nettoyer les ressources, les verrous et les processus utilisateurs non utilisés
- **SMON** (*System Monitor*), le processus chargé de vérifier la cohérence de la base de données et éventuellement sa restauration lors du démarrage si besoin

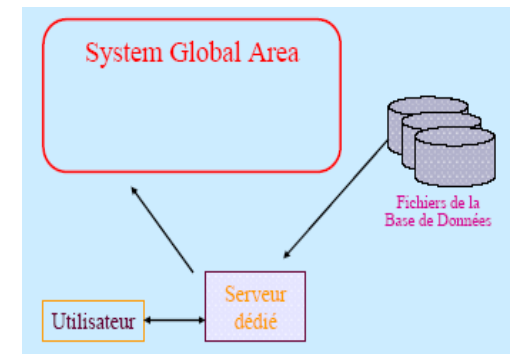
— Traitement de requêtes SQL

- Lorsqu'un utilisateur lance une déclaration ou une requête SQL, le process utilisateur l'envoie au process serveur qui la traite.

Ce traitement est découpé en trois étapes :

Parcours (Parse) :

- Le process serveur reçoit la requête, analyse la syntaxe et vérifie les privilèges de l'utilisateur pour les objets référencés à accéder.
- La section de mémoire du pool partagé de la SGA est utilisée pour compiler la requête et créer un arbre de parcours.
- Le statut de réussite ou non de l'analyse est renvoyé au process utilisateur.



Exécution (Exécution) :

- Préparation de la récupération des données.
- Si l'ordre SQL est un UPDATE ou un DELETE, les lignes affectées sont verrouillées par le process serveur pour être rendues non accessibles aux autres utilisateurs.

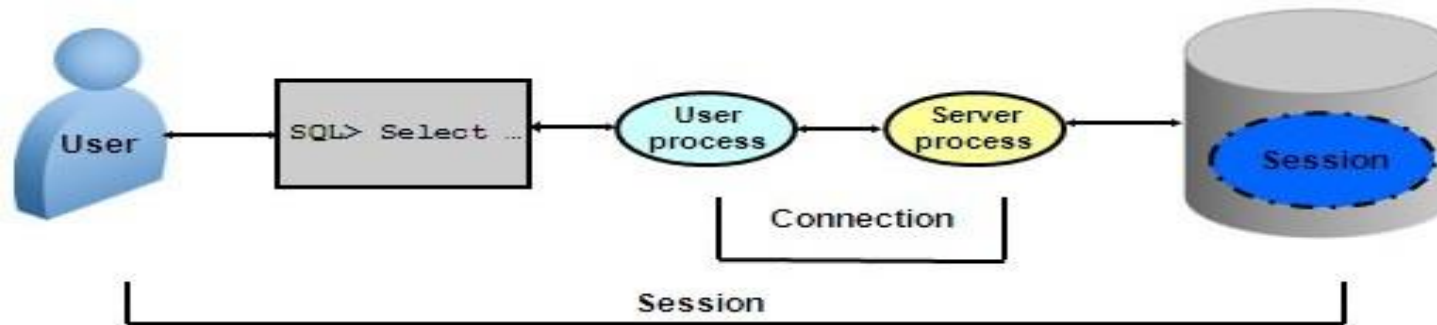
Récupération des données (Fetch) :

- Les données sont récupérées et envoyées au process utilisateur.
- Selon la quantité de mémoire disponible, un ou plusieurs fetch peuvent être nécessaires

— **Traitement d'une requête DML se compose de deux phases :**

- **Parcours (Parse) :** Contrôle des erreurs et placement dans une mémoire partagée SQL.
- **Exécution :** Si la phase de parcours s'est effectuée avec succès, l'exécution se fait en plusieurs étapes.
 - Lecture des blocs de données et de roll back dans le database buffer cache.
 - Mise en place de verrous sur les données qui vont être changées.
 - Ecriture de l'image-avant dans les segments de roll back et enregistrement des changements du buffer cache dans le redo log buffer pour protéger l'image-avant en cas de défaillance.
 - Finalement, les données sont enregistrées dans le buffer cache et les changements sont également enregistrés dans le redo log buffer protégeant ainsi l'image-après.
 - Les blocs de données altérées dans le buffer cache sont marqués comme " sale (Dirty) ", c'est-à-dire qu'ils ne correspondent plus aux blocs dans les fichiers de données.

- **Session utilisateur** : est une connexion à la base de données par l'intermédiaire d'un PROCESS USER



Exemple : Quand un utilisateur se connecte à travers SQL*Plus, cet utilisateur doit fournir son nom et son mot de passe. Une session est alors ouverte.

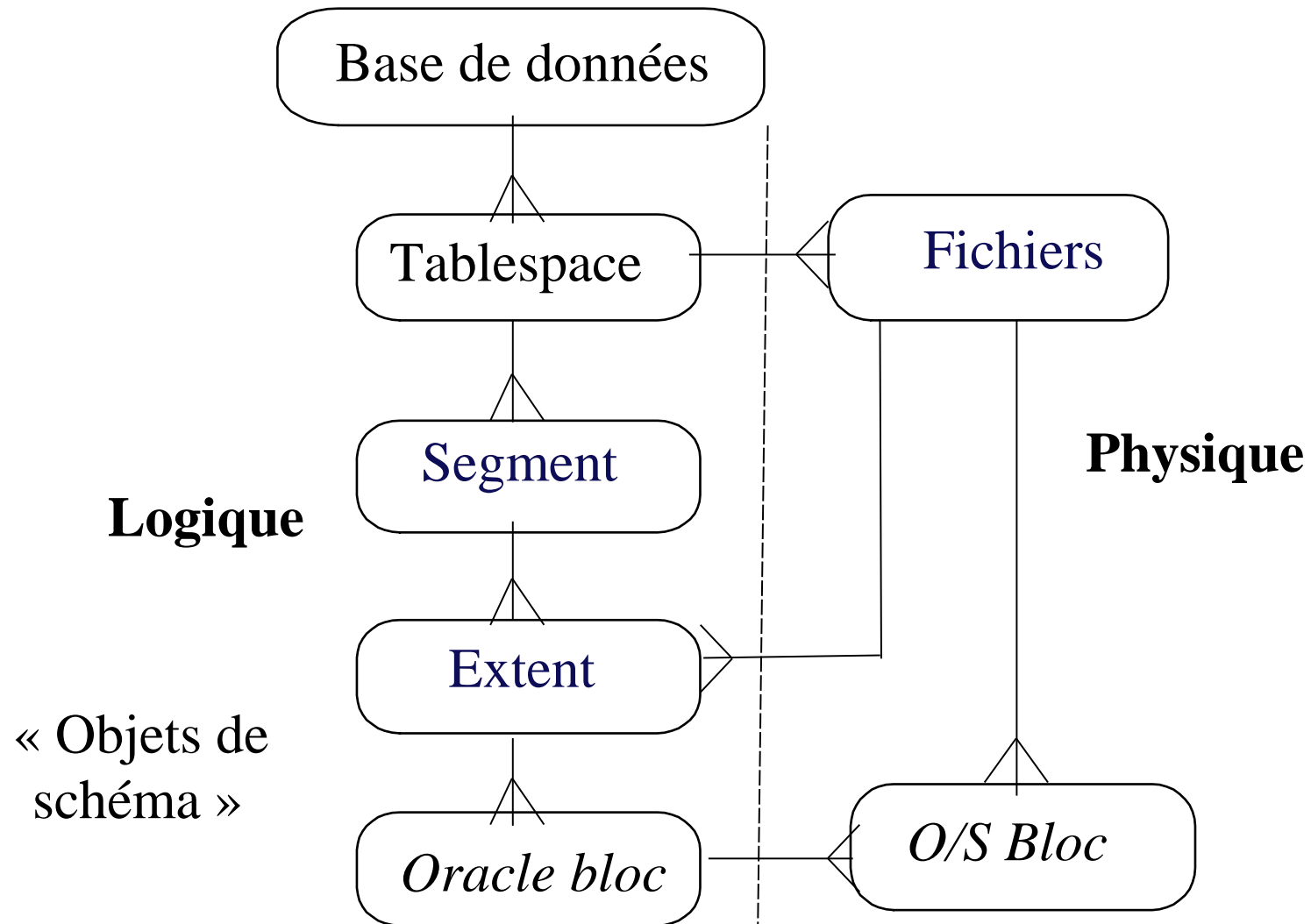
SQL> Entrer le nom d'utilisateur : **scott**

SQL> Entrer le mot de passe : **tiger**

Pour changer l'utilisateur :

SQL> **connect system/manager**

3. Architecture de base



3.1. Niveau Physique

- Ce niveau constitue d'une série de fichiers qui contiennent l'ensemble des données de la base :
- **Les fichiers de données (data files) :**

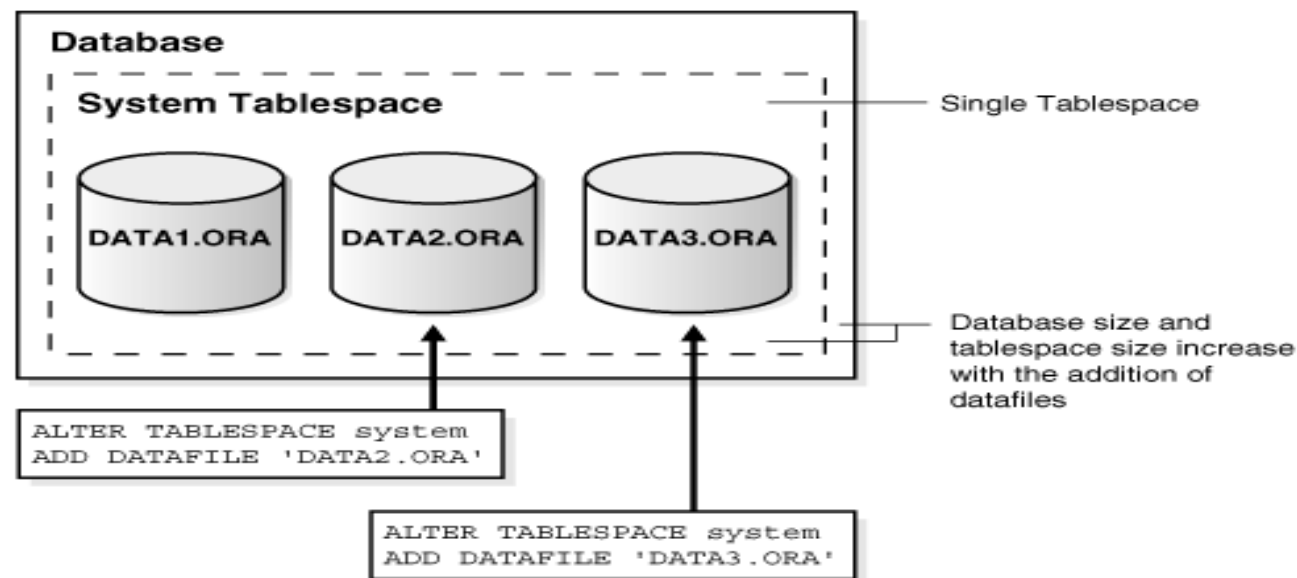
- Ils contiennent toutes les données de la base :

Exemple : données utilisateurs : informations clients (nom, prénom, adresse...)

Données systèmes : taille d'une table, nom et mot passe d'utilisateur...etc.

- Toutes les structures logiques et physiques y sont stockées : tables, index,

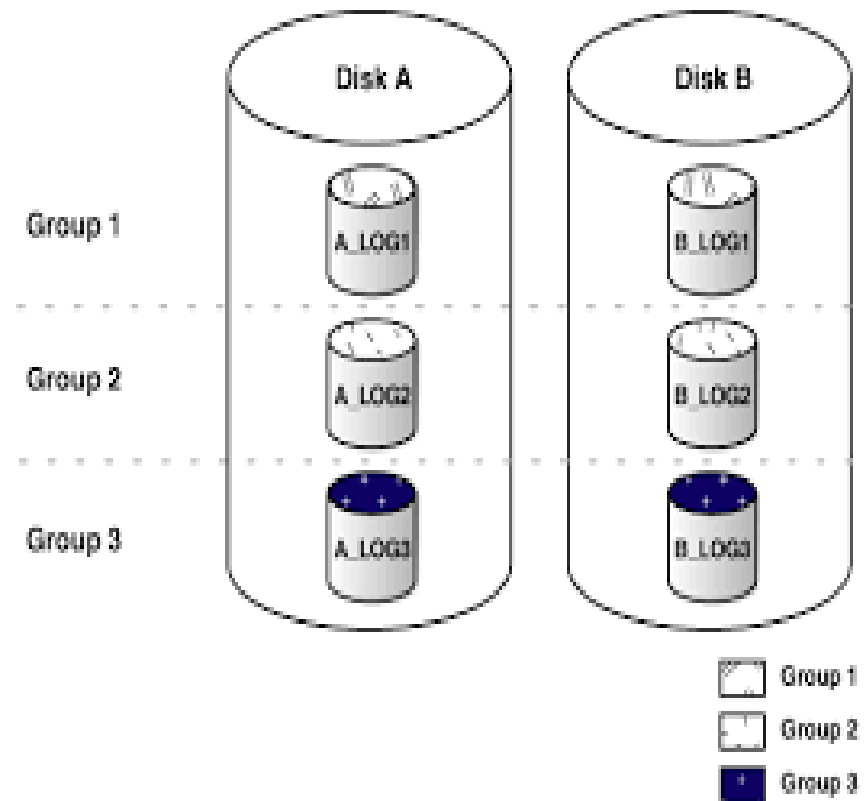
...



– Les fichiers de journalisation (redo log files) :

Ces fichiers contiennent la trace de l'activité en termes de mise à jour sur la base

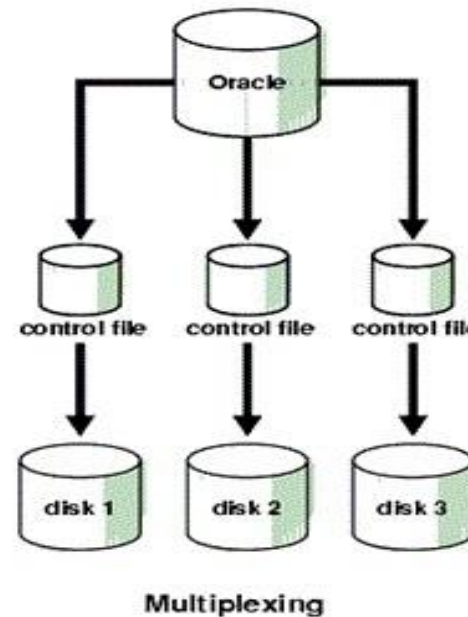
- Deux groupes au moins avec chacun au moins un fichier
- Ils doivent être multiplexés pour plus de sécurité
- Ne sont utiles qu'en cas de perte des fichiers de données ou d'arrêt anormal de la base
- Fonctionnent de façon cyclique



– Les fichiers de contrôle (control files) :

Contiennent la description physique de la base :

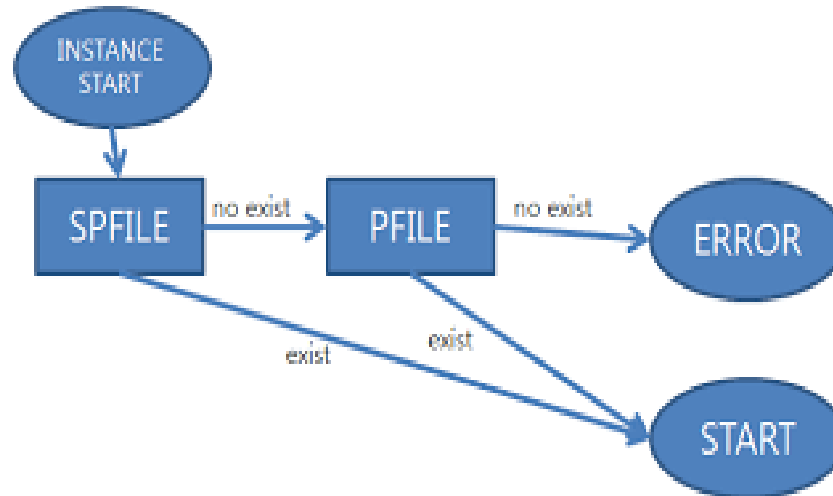
- Le nom de la base,
- Le nom et le chemin d'accès de chacun des fichiers,
- La date et l'heure de création de la base,
- Ils sont obligatoires au moment du démarrage et de l'ouverture de la base.
- Il est recommandé d'en avoir plusieurs et de les localiser physiquement sur des disques différents.



**How to
Multiplex
Control
Files in
Oracle**

Fichier INIT.ORA

Est le fichier de paramétrage de la base de données.



- Contient des paramètres de fonctionnement d'une instance et un paramètre identifiant le ou les fichiers CONTROL
- Sans ce fichier la base ne pourra pas démarrer.
- L'emplacement par défaut est ORACLE_HOME\db.

Quelques Paramètres du fichier INIT.ORA

- **CONTROL_FILES** : Noms des fichiers de contrôle.
- **DB_NAME** : Identifiant de la base de données de 5 caractères ou moins. (Seul paramètre nécessaire à la création d'une base).

– Dictionnaire de données

- C'est un ensemble de tables et de vues contenant des informations sur la BD.

Il fournit des informations sur :

- Structure physique et logique de la base de données
 - Noms, descriptions, implantation des objets
 - Contraintes d'intégrité
 - Utilisateurs et privilèges.
- On peut accéder au dictionnaire que par l'intermédiaire de vues.
- Il existe 4 catégories de vues (reconnaissables par leur préfixe) :
 - **Vues_dynamiques : v\$_nomvue** : donnent les informations systèmes de la base.

Exemple : v\$instance : donne le nom d'instance.

- **USER_XXX** : décrit les objets appartenant à l'utilisateur connecté
- **ALL_XXX** : décrit les objets accessibles à l'utilisateur connecté
- **DBA_XXX** : décrit tous les objets (vues autorisées aux DBAs seulement...)

Chaque XXX est en général remplacé par un nom (en anglais) significatif.

- Un (méta) description du dictionnaire est donnée par la vue DICT.

```
Select * from DICT where table_name= 'DBA_USERS';
```

Exemples de vues:

- Tables qui contiennent un attribut Name

```
SELECT TABLE_NAME FROM USER_TAB_COLUMNS  
WHERE COLUMN_NAME='NAME';
```

- Attributs de la table DEPT

```
SELECT COLUMN_NAME FROM USER_TAB_COLUMNS  
WHERE TABLE_NAME='DEPT';
```

3.2. Niveau logique

– Ce niveau se compose des objets suivants :

Tablespaces ; segments ; extensions (extents) ; blocs ; objet de schéma (schema object).

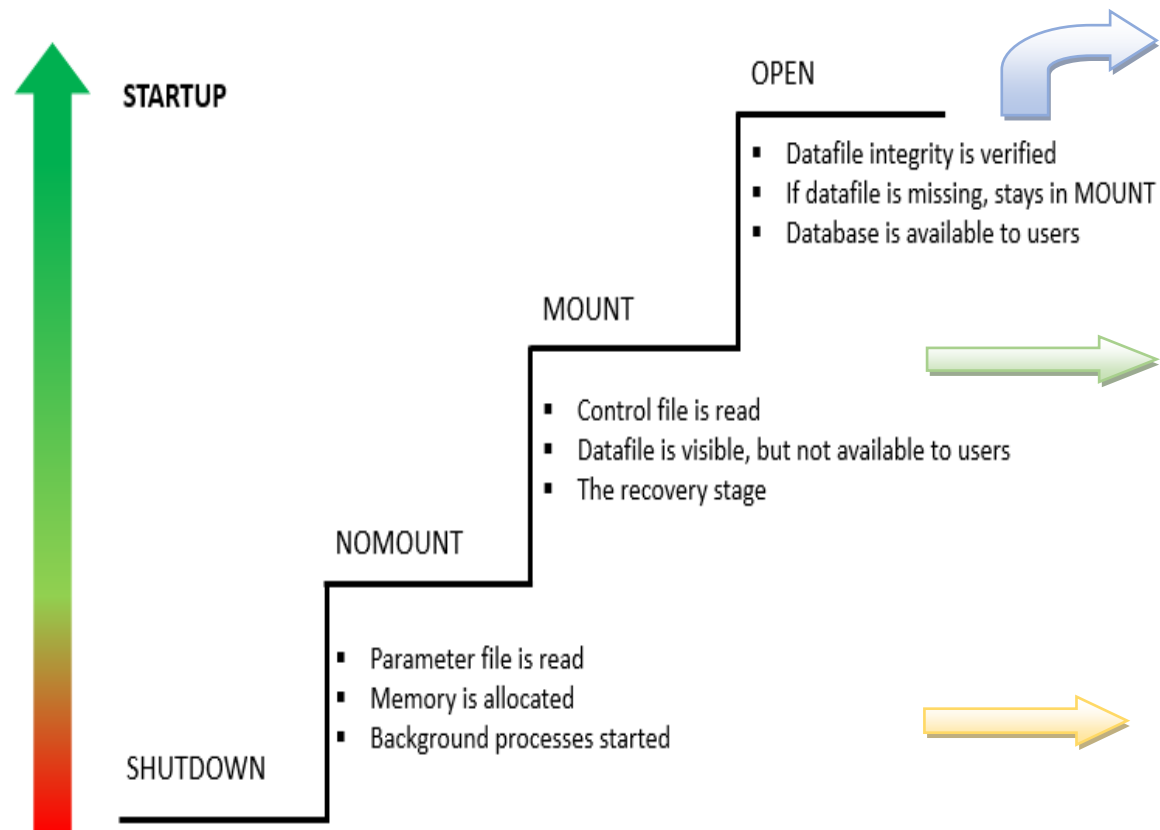
Les objets de schéma comprennent :

Les tables ; les vues ; les index ; les clusters ; les séquences ; les procédures stockées ; les fonctions ; les packages ; les déclencheurs (triggers).

4. Administration d'une base de données

4.1. Gestion d'une base

- Les différents statuts



C'est l'état classique, de la base disponible à tous les utilisateurs

L'instance est lancée, la base n'autorise pas de transactions, ni les connexions utilisateurs normales, mais des opérations d'exploitations portant généralement sur les fichiers. Ce type est nécessaire pour des actions de « maintenance ».

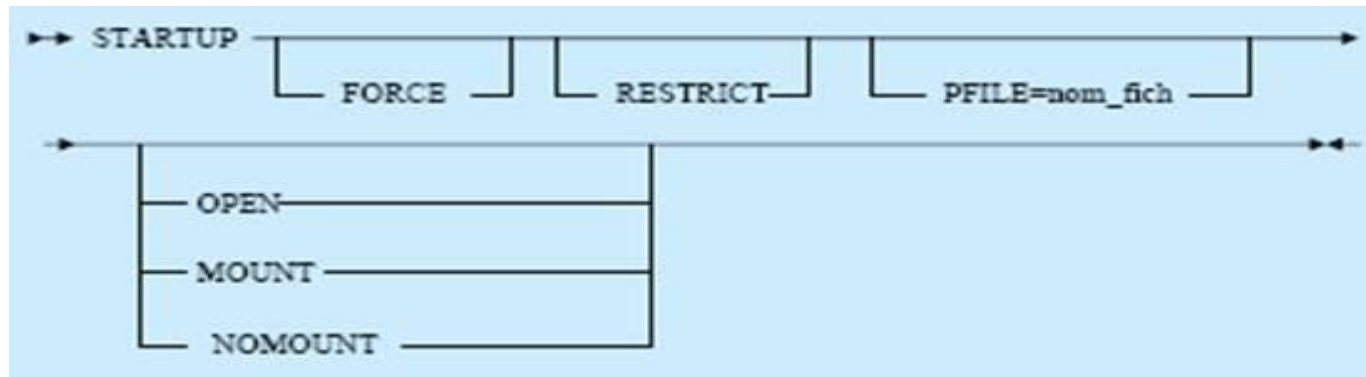
Il sert dans les cas suivants : renommer des data files, ajouter, supprimer, renommer des fichiers de redo log, activer/désactiver l'archivage des fichiers de redo log, effectuer une restauration.

L'instance seule est démarrée. On en peut déclencher que des pseudo connexions de type 'SYSDBA' ou 'SYSOPER'.

Les opérations sont assez limitées. Ce type n'est utilisé que lors de la création d'une base de données.

FORCE	L'administrateur demande au système de forcer un démarrage (l'instance fut probablement mal arrêtée). utilisée qu'exceptionnellement, et nécessaire si les autres options de démarrage n'aboutissent pas, car un fichier est perdu, par exemple.
EXCLUSIVE	L'administrateur désire être en « single-user » sur l'instance (nous verrons ultérieurement que seuls certains utilisateurs « privilégiés » peuvent se connecter, ils sont alors pourvus du privilège appelé « restricted session »). Des opérations de maintenance sont prévues : reconstruction d'index, effectuer un import ou un export.
‘en veille’ (QUIESCED)	C'est un état particulier, qui permet au DBA de travailler tranquillement, sans transactions concurrentes, mais sans avoir à redémarrer la base pour la passer en accès restreint. Dans ce mode, les transactions actives sont momentanément suspendues, jusqu'à l'éveil de la base.

- Démarrage de la base



- FORCE : permet de forcer un démarrage quel que soit l'état de la base, en forçant un arrêt brutal puis un startup.
- RESTRICT : démarre la base en accès restreint seulement pour les DABS. Aucune autre connexion n'est acceptée.
- PFILE=: permet de spécifier un fichier d'initialisation autre que le défaut.
- SPFILE=: permet de spécifier un fichier SPFILE autre que le défaut,
- READ : en lecture seule, pas d'update, delete, insert, ...

Exemples :

un démarrage (très) progressif en 3 temps :

```
SQL> STARTUP NOMOUNT PFILE=init_XXXX.ora;
```

```
SQL> ALTER DATABASE MOUNT;
```

```
SQL> ALTER DATABASE OPEN;
```

un démarrage en comité restreint

```
SQL> STARTUP RESTRICT
```

Démarrage automatique

```
SQL> STARTUP
```

- Arrêt de la base

- Nécessite des privilèges au niveau du système d'exploitation.
- Se fait sous SQL*DBA avec la commande SHUTDOWN.
- Ne peut pas se faire en plusieurs étapes.

Arrêt normal SQL>SHUTDOWN	Les nouvelles connexions ne sont pas permises, Oracle attend que les utilisateurs aient fini leurs travaux pour fermer l'instance
Arrêt immédiat SQL>SHUTDOWN IMMEDIATE	Les utilisateurs sont déconnectés, les opérations en cours annulées (rollback), Oracle n'attend pas la fin des travaux des utilisateurs. Il force la déconnexion et ne valide pas leurs échanges
Arrêt brutal SQL>SHUTDOWN ABORT	Forcer l'arrêt de la base de données de manière brutale (abrupte), Cette action peut être effectuée lors d'un blocage de l'instance, peut entraîner la perte de données non validées, et recommander de l'utiliser avec précaution
transactional	Oracle attend la fin des transactions des utilisateurs connectés. Aucune nouvelle connexion n'est possible, seuls les ordres de type commit ou rollback sont autorisés

– Création d'une base

- Création de l'instance

Etapes à suivre pour créer une base manuellement

Etape 1. Définir l'arborescence de la base

C:\app\...\dbtest\admin, ...

Étape 2: Création d'une variable DOS ORACLE_SID

C:> SET ORACLE_SID=BTEST

Etape 3 : Création d'un Service Windows qui gère l' instance.

Nous créons ce service en utilisant ORADIM.EXE.

C:> oradim -new -sid BTEST -intpwd mypass -startmode manual

Etape 4 : Création du fichier de paramètres Init.ora

Ce fichier est un fichier binaire, appelé SPFILE. Ce fichier binaire est généré à partir d'un fichier ASCII INIT.ORA que nous devons créer.

Paramètres du fichier Init.ora :

control_files= liste des fichiers contrôles files

db_name=btest

Etape 5 : Démarrage de l'instance

C:> **sqlplus /nolog** -- pour ouvrir une session sql plus

SQL> **connect / as sysdba** -- pour se connecter en tant qu'admin

- création du fichier des paramètres serveur :

SQL> **Create spfile from pfile='C:\app\vaio\admin\Btest\pfile\\init.ora'**

démarrage de l'instance :

SQL> **startup nomount**

• Résultat :

- la SGA est créée,
- les **BACKGROUND PROCESS** sont démarrés.

- Création de la Base

La création d'une nouvelle base de données se fait par l'ordre CREATE DATABASE.

```
CREATE DATABASE nom_base  
[CONTROLFILE REUSE]  
[LOGFILE fichier, ...]  
[MAXLOGFILES valeur]  
[MAXLOGMEMBERS valeur]  
[DATAFILES valeur]  
[MAXINSTANCES valeur];
```

Paramètres :

- nom_base : nom donné à la BD, composé de 8 caractères au maximum.
- CONTROLFILE : utilisé uniquement en cas de création d'une base suite à la suppression d'une base existante ; permet de réutiliser les fichiers de contrôle de la base précédente.
- MAXLOGMEMBERS valeur : définit le nombre maximal de copies pour un fichier de reprise

Exemple:

CREATE DATABASE B_TEST

DATAFILE 'system.dbs' SIZE 5M

LOGFILE 'log01.rdo' SIZE 200K, 'log02.rdo' SIZE 200K ;

Résultat :

- un fichier **DATABASE** contenant le dictionnaire de données est créé,
- 2 fichiers **REDO LOG** sont créés,
- la base a le statut **OPEN**,
- 2 utilisateurs existent : **SYS/CHANGE_ON_INSTALL** et

SYSTEM/MANAGER.

- Création des tables système

- Création des vues du dictionnaire de données : Exécuter les scripts **SQL** fournis.

- Sous **SYS**:

- **CATALOG.SQL**: vues et synonymes publics

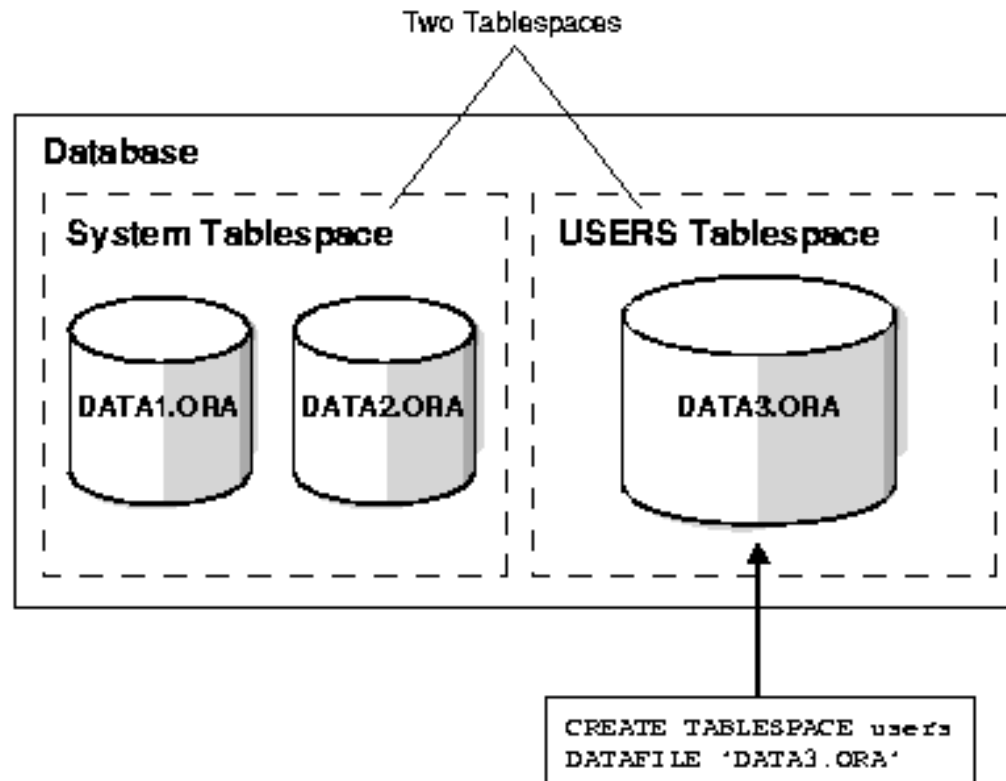
- **UTLMONTR.SQL**: synonymes publics pour les tables virtuelles

- Sous **SYSTEM**, et pour chaque administrateur de la base :

- **CATDBSYN.SQL**: synonymes sur les vues **DBA_***

5. Gestion de l'Espace physique d'une base

5.1. Tablespace



- Unité logique de stockage des données,
- Lié à un ou plusieurs fichiers, et un fichier est associé à un et un seul Tablespace,

- Un Tablespace peut être mis **OFFLINE**, sauf le Tablespace **SYSTEM** (créé en même temps que la BD).
- Un tablespace peut être TEMPORARY ou PERMANENT.
- Un tablespace temporaire contient des données temporaires liés au TRI par exemple.
- Il existe toujours deux tablespace baptisés SYSTEM et SYSAUX.
 - SYSTEM : contient le dictionnaire de données et segment d'annulation SYSTEM
 - SYSAUX : contient les informations nécessaires aux composants et outils supplémentaires et traditionnellement.

Exemples d'utilisation du Tablespace

- Limitation d'utilisation des espaces pour les utilisateurs (quota),
- disponibilité des informations,
- sauvegarde et restauration des données,
- distribution des informations sur différents disques.

1. Création

```
CREATE [UNDO] [TEMPORARY] TABLESPACE nom_TS  
DATAFILE fichier [, fichier] ...  
DEFAULT STORAGE ([INITIAL valeur]  
[NEXT valeur] [MINEXTENTS nombre] [MAXEXTENTS nombre]  
[PCTINCREASE 0])  
[ONLINE / OFFLINE];
```

- INITIAL : taille de la 1ère extension du segment (par défaut 5 blocs)
- NEXT : taille 2ème extension du segment (par défaut 5 blocs)
- MINEXTENTS : nombre d'extensions allouées à la création du segment (par défaut : 1)
- MAXEXTENTS : nombre maximal d'extensions pouvant être allouées au segment (par défaut dépend de la taille du bloc, min : 1)
- PCTINCREASE : pourcentage d'accroissement de la taille des extensions à partir de la 3^{ème} (taille extension i+1 = (taille extension i) * (1+p/100)) (par défaut : 50%).

Example:

```
CREATE TABLESPACE app_data  
DATAFILE 'app01.dbf' SIZE 1M,  
        'app02.dbf' SIZE 1M  
DEFAULT STORAGE (INITIAL 500K NEXT 500K MINEXTENTS 1  
MAXEXTENTS 500 PCTINCREASE 0);
```

2. Ajout des fichiers à un Tablespace

Exemple

```
ALTER TABLESPACE app_data  
ADD DATAFILE 'app03.dbf' SIZE 2M;
```

3. Extension Automatique

Exemple :

```
ALTER TABLESPACE app_data  
ADD DATAFILE 'app04.dbf' SIZE 2M  
AUTOEXTEND ON NEXT 1M MAXSIZE 50M;
```

4. Modification de la taille d'un fichier

Exemple :

```
ALTER DATABASE DATAFILE 'app02.dbf' RESIZE 2M;
```

5. Modification des paramètres de Stockage

Exemple :

```
ALTER TABLESPACE app_data  
DEFAULT STORAGE  
(INITIAL 2M NEXT 2M MAXEXTENTS 900);
```

6. Statut d'un Tablespace

- En mode **OFF LINE**, les données ne sont pas accessibles
- En mode **ON LINE**, les données sont accessibles
- Le tablespace SYSTEM ne peut pas être mis OFF LINE.

Exemple: ALTER TABLESPACE app_data OFFLINE;

7. READ-ONLY Tablespace

Exemple:

```
ALTER TABLESPACE app_data READ ONLY;
```

Les données du Tablespace ne sont accessibles qu'en consultation.

8. Suppression d'un Tablespace

La suppression d'un tablespace entraîne la disparition de tous les objets qu'il contient.

Exemple:

```
DROP TABLESPACE app_data  
INCLUDING CONTENTS;
```

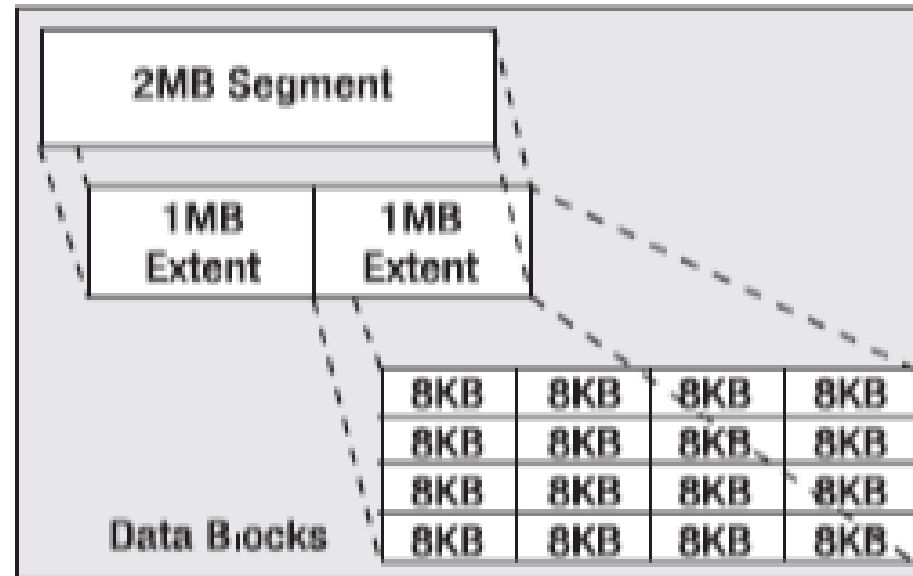
L'ordre DROP ne supprime pas les fichiers du disque ; il faut ensuite détruire ces fichiers par les commandes du système d'exploitation. Avant de supprimer un tablespace, il faut le mettre hors service (OFFLINE).

9. Vues associées aux Tablespaces

- Vues liées aux extensions : User_extents, dba_extents
- Vues liées aux Segments: User_segments, dba_segments, v\$sort_segment
- Vues liées aux espaces libres dans les fichiers : user_free_space, dba_free_space
- vues liées aux utilisateurs : dba_users, v\$sort_user
- vues liées aux quotas : dba_ts_quotas
- vues liées au tablespaces : user_tablespaces, dba_tablespaces, v\$tablespace
- Vues liées aux fichiers permanents ou temporaires : dba_data_files, v\$datafiles, v\$tempfile, dba_temp_files, v\$temp_extent_map, v\$temp_extent_pool

5.2. Segment et leurs composants

i. Segment et Extent



- Un segment est un ensemble d'extensions
- Une extension est un ensemble de blocs contiguës
- Les paramètres de stockage de données et les paramètres de dimensionnement des blocs de données permettent de contrôler l'allocation d'espaces à un segment.
- Un segment ne peut pas s'étendre sur plusieurs tablespaces.

- Il y'a plusieurs types de segments :
 - Segment de données = table
 - Segment d'index = index
 - Segment d'annulation = rollback segment, qui stocke l'image avant modification des données, il permet d'effectuer des lectures consistantes, d'annuler les transactions et de recouvrer la base de données.
 - Les entrées rollback sont stockées aussi dans le buffer redo log puis dans le fichier Redo Log.
 - Segment temporaire, utilisé en interne par Oracle, si la zone mémoire de tri est insuffisante
- Remarque : seuls les objets 'physiques' peuvent être des segments. Ainsi une vue ou un synonyme n'est pas un segment...
- On peut forcer les segments de données et d'index à s'implanter dans un tablespace particulier :
 - explicitement à la création du segment
 - implicitement en affectant un tablespace par défaut à l'utilisateur qui va créer le segment.

ii. Segment de données de type Table

1. Création d'une Table

Au moment de sa création, une table est allouée dans un tablespace et est constituée :

- d'un segment de données.
- de zéro, un ou plusieurs segments Index

- Exemple de création d'un segment géré dans le dictionnaire

```
CREATE TABLE scott.emp (  
  empno NUMBER PRIMARY KEY,  
  ename VARCHAR2(10) NOT NULL CHECK (ename =  
    UPPER(ename)),  
  job VARCHAR2(9),  
  mgr NUMBER CONSTRAINT fk_mgr REFERENCES  
    scott.emp(empno),  
  hiredate DATE DEFAULT SYSDATE,  
  sal NUMBER(10,2) CHECK (sal > 500),  
  comm NUMBER(9,0) DEFAULT NULL,  
  deptno NUMBER(2) NOT NULL  
    CONSTRAINT fk_deptno REFERENCES scott.dept(deptno)  
)  
  
STORAGE (initial 100K next 50K minextents 3 maxextents 5 pctincrease 50)  
TABLESPACE USERS  
EXTENT MANAGEMENT DICTIONARY;
```

—Extensions allouables au segment de la table scott.emp :

<u>Nr. Extension</u>	<u>Taille extension</u>	<u>Valeur de l'extension suivante</u>
1 (initial/min 1)	100K	50K
2 (min 2)	50K	$75K = 1,5 * 50K$
3 (min 3)	75K	$112,5K = 1,5 * 75K$
4	112,5K	$168,75 = 1,5 * 112,5K$
5	168,75	$253,125 = 1,5 * 168,75K$

2. Modification d'Allocation d'une Table

Exemple

```
ALTER TABLE scott.emp  
STORAGE( NEXT 300K MAXEXTENT 10 PCTINCREASE 10)  
ALLOCATE EXTENT (SIZE 400K DATAFILE 'user_data1.dbf') ;
```

3. Suppression d'une table

- Trois approches pour supprimer les données d'un segment :
 - DROP TABLE : supprime les données et le segment

DROP TABLE [schema.] table [CASCADE CONSTRAINTS]

- TRUNCATE TABLE : supprime les données et libère à la demande l'espace alloué au segment

TRUNCATE {TABLE [schema] table | CLUSTER [schema] CLUSTER } [{
DROP | REUSE } STORAGE]

- DELETE FROM : supprime toutes les données d'une table mais l'espace alloué au segment est conservé.

iii. Segment Rollback

- Ce segment est une structure temporaire mémoire stockée sur disque dans un tablespace.

1. Création d'un RBS

Exemple

```
create rollback segment r1;
```

```
CREATE PUBLIC ROLLBACK SEGMENT r5  
TABLESPACE ts_rbs  
STORAGE (INITIAL 100K MINEXTENTS 3 MAXEXTENTS 10 NEXT 100K  
OPTIMAL 400K);
```

- PUBLIC : segment accessible par toutes les instances d'une même base
- OPTIMAL : Taille optimale du RBS obtenue par Oracle par désallocation dynamique

NOTE : par défaut un RBS créé est inactif.

2. Modification d'un RBS manuel

- Il est possible d'activer ou désactiver un RBS et de modifier ses paramètres de stockage
- un Rollback segment actif ne peut être mis OFFLINE

Exemple:

```
ALTER ROLLBACK SEGMENT r5 ONLINE;
```

3. Suppression d'un RBS

- Un segment avec des transactions actives ne peut être supprimé
- il est utile de séparer les segments RBS des segments de données
- Le RBS SYSTEM ne peut être ni arrêté ni supprimé.

Exemple:

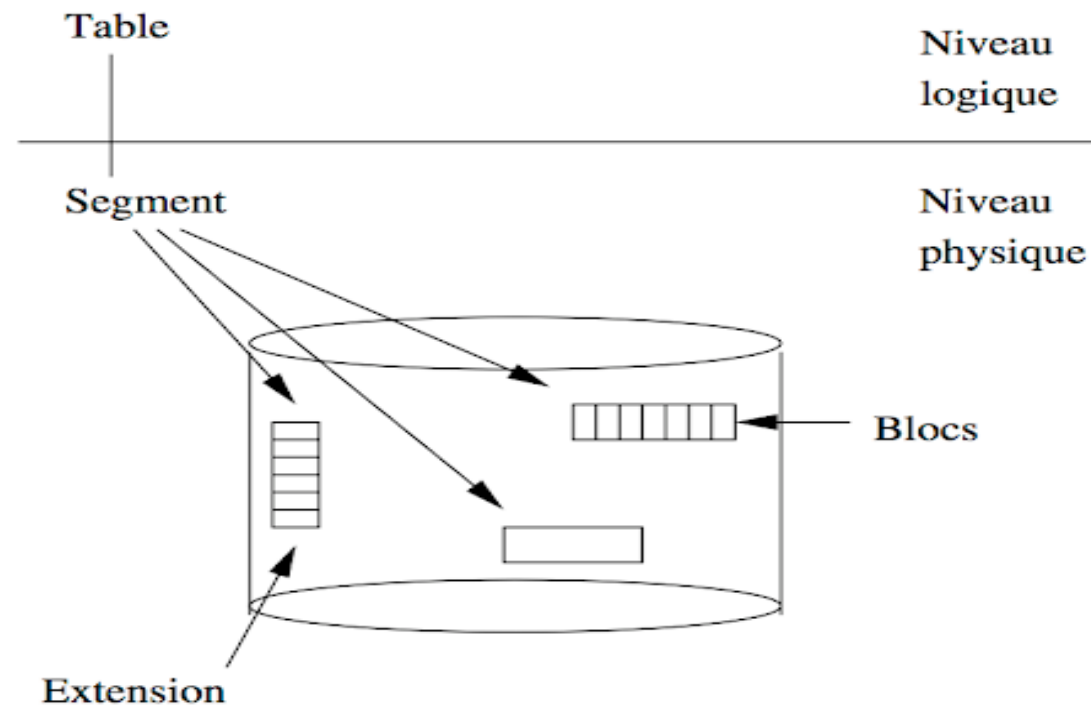
```
DROP ROLLBACK SEGMENT r5;
```

4. Vue DBA_ROLLBACK_SEGS

```
select * from DBA_ROLLBACK_SEGS;
```

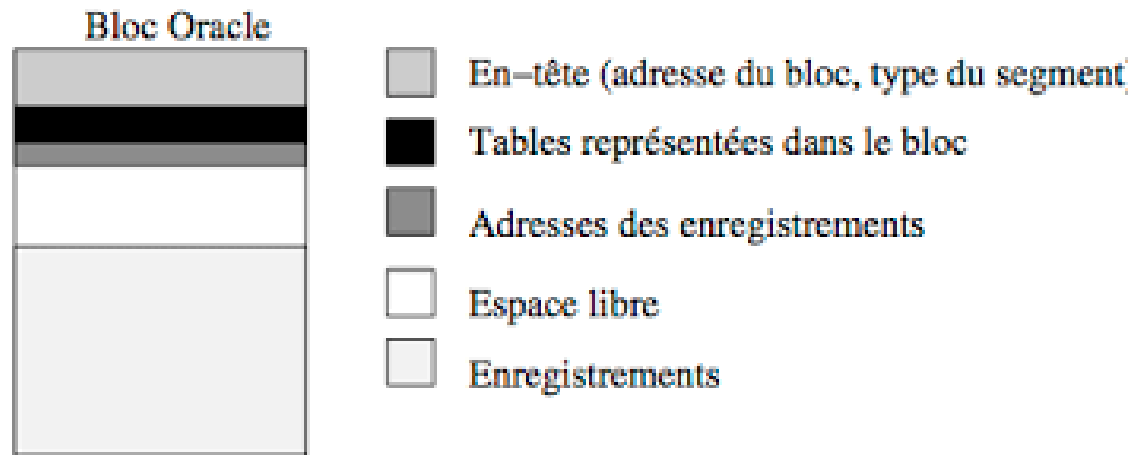
5.3. Bloc

- Le bloc est l'unité logique de transfert d'informations entre disque et mémoire centrale.
- La taille du bloc peut soit être fixée par l'administrateur de la base (paramètre DB_BLOCK_SIZE du fichier INIT.ORA).



- Les paramètres suivants permettent de contrôler la taille d'un bloc en insertion ou en mise à jour :
 - **PCTFREE** : 10 % par défaut, réserve l'espace pour les mises à jour, permet un meilleur remplissage en insertion, ...
 - **PCTUSED** : 40% par défaut, lorsqu'un bloc est plein, les insertions ne peuvent y reprendre que si $PCTUSED \leq 40\%$
 - **INITRANS** : Nombre de transactions pré-allouées pouvant être actives dans un bloc, par défaut 1 pour un segment de table et 2 pour un d'index ou de cluster.

- Un bloc est principalement divisé en trois sous-parties :



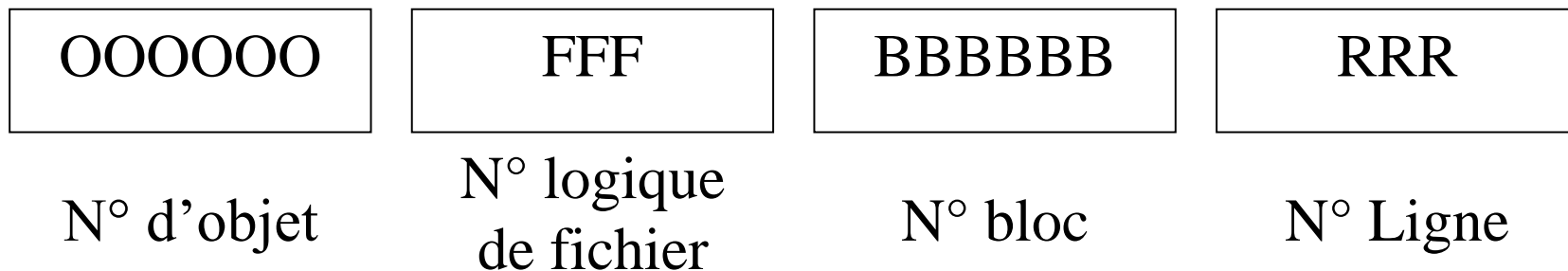
- En-tête de bloc : contient des informations sur le bloc : numéro de bloc, la date de création et le type de segment auquel il appartient.
- Espace libre : est une zone vide pouvant être utilisée pour ajouter de nouvelles données ;
- Espace de lignes : contient les données sur les lignes de la table à laquelle le bloc appartient.

5.4. Structure d'une ligne

Chaque enregistrement est identifié par un **ROWID**, comprenant trois parties :

- Le numéro du bloc au sein du fichier
- Le numéro de l'enregistrement au sein du bloc
- Enfin l'identifiant du fichier.

Format de ROWID



Package DBMS_ROWID : permet d'interpréter la valeur du ROWID

Exemple:

```

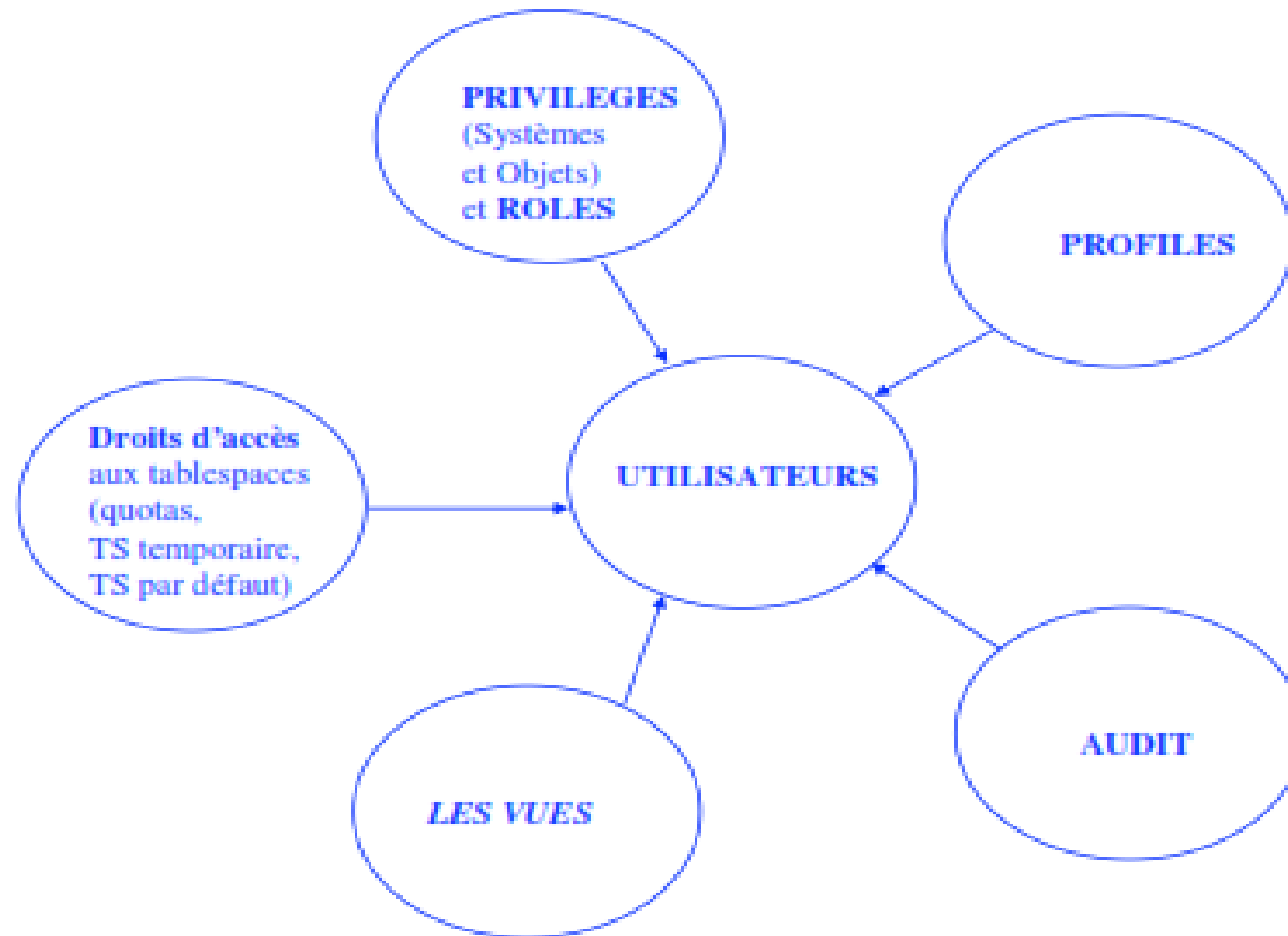
SET SERVEROUTPUT ON
SELECT DBMS_ROWID.ROWID_OBJECT (rowid) "n° objet",
       DBMS_ROWID.ROWID_RELATIVE_FNO (rowid) "n° fichier",
       DBMS_ROWID.ROWID_BLOCK_NUMBER (rowid) "n° bloc",
       DBMS_ROWID.ROWID_ROW_NUMBER (rowid) "n° ligne"
FROM Employe;
/

```

Résultats :

<u>n° objet</u>	<u>n° fichier</u>	<u>n° bloc</u>	<u>n° ligne</u>
2430	4	2638	0
2430	4	2638	1
2430	4	2638	2

6. Gestion de la sécurité et des ressources



6.1. Contrôle des accès

L'accès à la base de données s'effectue par l'intermédiaire de la notion utilisateur.

6.1.1. Utilisateur

Chaque utilisateur, ou USER est définit par :

- Un nom d'utilisateur + mot de passe ;
- Liste de tablespaces ;
- Un ensemble de privilèges et rôles ;
- Un profil.

La vue **DBA_USERS** permet de consulter la liste des caractéristiques des utilisateurs.

1. Création d'un Utilisateur

La création d'un utilisateur s'effectue par l'ordre Create user :

Exemple:

```
CREATE USER user1  
IDENTIFIED BY info  
DEFAULT TABLESPACE app_data  
QUOTA 15m ON app_data  
PASSWORD EXPIRE  
PROFILE default;
```

- Default tablespace : tablespace qui sera utilisé par défaut lors de la création d'une table.
- Quota [nombre/unlimited] on tablespace : limite l'espace qui pourra être alloué pour les tables de l'utilisateur.
- Profile nom_profile : affecter un profil particulier à l'utilisateur
- Password Expire : demande le changement de passe.

2. Modification du mot de passe

```
ALTER USER user1  
IDENTIFIED BY deptinfo  
PASSWORD EXPIRE;
```

3. Modification du quota

```
ALTER USER user1  
QUOTA 10M ON app_data;
```

4. Suppression d'un Utilisateur

```
DROP USER user1;
```

Suppression des Objets associés : Clause CASCADE :

```
DROP USER user1 CASCADE;
```

6.1.2. Privilège

Deux types de privilèges :

- les privilèges système autorisent l'accès à la BD.
- les privilèges objets sur les données (tables, vues, procédures, fonctions, séquences) qui donnent les autorisations d'accès aux données.
 - La commande **GRANT** permet d'attribuer un privilège à un utilisateur ou à un groupe d'utilisateurs.
 - La commande **REVOKE** permet de supprimer les privilèges.

1.Privilège système

Exemple : Privilèges SYSTEM

CREATE SESSION : connexion à la BD

CREATE TABLE, ALTER TABLE, DROP TABLE

a- Attribution d'un privilège SYSTEM :

GRANT privilège TO [utilisateur | rôle | PUBLIC][WITH ADMIN OPTION]

- WITH ADMIN OPTION : autorise celui qui a reçu le privilège ou le rôle à le transmettre à un autre utilisateur ou rôle.
- Public permet d'affecter le privilège ou le rôle à tous les utilisateurs.

Exemples:

GRANT CREATE SESSION, CREATE TABLE TO user1;

GRANT CREATE SESSION TO scott WITH ADMIN OPTION;

b- Suppression d'un privilège SYSTEM

```
REVOKE privilège FROM utilisateur| rôle | PUBLIC;
```

Exemples:

REVOKE CREATE TABLE FROM user1;

REVOKE CREATE SESSION FROM scott;

c- Les vues du dictionnaire

*SELECT * FROM DBA_SYS_PRIVS ORDER BY grantee, privilege ;*

<i>GRANTEE</i>	<i>PRIVILEGE</i>	<i>ADM</i>
<i>CONNECT</i>	<i>ALTER SESSION</i>	<i>NO</i>
<i>CONNECT</i>	<i>CREATE CLUSTER</i>	<i>NO</i>

2.Privilège objet

Un utilisateur qui crée un objet à tous les droits sur celui-ci, les autres utilisateurs (sauf DBA) n'ont aucun droit.

a- Création d'un privilège OBJET

```
GRANT [privilège | ALL] ON objet TO [utilisateur | rôle | PUBLIC] [WITH  
GRANT OPTION];
```

Exemple :

```
GRANT UPDATE(ename, sal) ON emp TO user1  
WITH GRANT OPTION;
```

b- Suppression d'un privilège

```
REVOKE [privilège | ALL] ON objet FROM [utilisateur | rôle | PUBLIC] ;
```

c- Visualisation des privilèges objets

*DBA_TAB_PRIVS, DBA_COL_PRIVS, ALL_TAB_PRIVS,
ALL_COL_PRIVS, ALL_TAB_PRIVS_MADE, ALL_COL_PRIVS_MADE,
ALL_TAB_PRIVS_RECD, ALL_COL_PRIVS_RECD, USER_TAB_PRIVS,
USER_COL_PRIVS, USER_TAB_PRIVS_MADE,
USER_COL_PRIVS_MADE, USER_TAB_PRIVS_RECD,
TABLE_PRIVILEGES, COLUMN_PRIVILEGES*

-

Principales Colonnes de vues ci-dessus

GRANTEE : utilisateur ayant reçu le privilège

OWNER : propriétaire de la table

TABLE_NAME : nom de la table

COLUMN_NAME : Nom de la colonne concerné

GRANTOR : Utilisateur ayant affecté le privilège

PRIVILEGE : privilège affecté

GRANT : privilège reçu

6.1.3. Rôle

C'est un groupe nommé de privilèges qui peut être accordé à un utilisateur.

1. Création

```
CREATE ROLE nom_de_rôle [NOT IDENTIFIED]  
IDENTIFIED BY[mot_de_passe]
```

Exemples :

```
CREATE ROLE role1;
```

```
GRANT CREATE TABLE, CREATE SESSION TO role1;
```

```
GRANT SELECT ON emp TO role1;
```

2. Modification

```
ALTER ROLE nom_rôle [NOT IDENTIFIED]  
IDENTIFIED BY [mot_de_passe]
```

3. Activation et désactivation des Rôles

- Désactiver un rôle enlève les privilèges associés aux utilisateurs.
- Activer un rôle affecte les privilèges associés aux utilisateurs.
- La commande SET ROLE permet d'activer et de désactiver les rôles
- Les rôles par défaut sont affectés à l'utilisateur à sa connexion.
- Un mot de passe peut être nécessaire

```
SET ROLE {rôle [IDENTIFIED BY mot_de_passe] | ALL | EXCEPT rôle  
[,rôle] ...};
```

Exemples :

```
SET ROLE role1;
```

```
SET ROLE ALL EXCEPT role1;
```

4. Suppression d'un rôle pour un utilisateur

```
REVOKE rôle [,rôle] ... FROM {utilisateur | rôle | PUBLIC};
```

Exemple:

```
REVOKE role1 FROM PUBLIC;
```

5. Suppression d'un rôle

Exemple :

```
DROP ROLE role1;
```

6. Informations sur les rôles

- Les vues suivantes contiennent des informations sur les rôles :
dba_roles, user_role_privs, dba_role_privs, role_role_privs,
role_sys_privs, role_tab_privs, session_roles

Exemple 1 : liste de tous les rôles de la base

```
sql> SELECT * FROM sys.dba_roles ;
```

6.1.4. Profil

- C'est un objet géré par la BD qui définit l'ensemble de ressources mises à la disposition des utilisateurs par le SGBDR, et les paramètres de gestion des mots de passe.

1.Création

– Syntaxe partie limite des ressources :

```
CREATE PROFILE profile LIMIT  
[ SESSIONS_PER_USER { integer | UNLIMITED | DEFAULT } ]  
[ CPU_PER_SESSION { integer | UNLIMITED | DEFAULT } ]  
[ CPU_PER_CALL { integer | UNLIMITED | DEFAULT } ]  
[ CONNECT_TIME { integer | UNLIMITED | DEFAULT } ]  
[ IDLE_TIME { integer | UNLIMITED | DEFAULT } ]  
[ LOGICAL_READS_PER_SESSION {integer | UNLIMITED/DEFAULT}]  
[ LOGICAL_READS_PER_CALL {integer | UNLIMITED/DEFAULT}]  
[ COMPOSITE_LIMIT { integer | UNLIMITED | DEFAULT } ]  
[ PRIVATE_SGA {integer [K | M] | UNLIMITED | DEFAULT}]
```

- Mots clés et paramètres :

- **Session_per_user** : Nombre maximum de sessions par utilisateur
- **Logical_read_per_session** : Nombre de blocs de données à lire pour une session
- **cpu_per_session** : temps CPU max par session en % de secondes
- **cpu_per_call** : temps CPU pour un appel (en cas de parse, execute ou fetch) en % de secondes
- **connect_time** : temps écoulé maximum (en minutes)
- **idle_time** : temps maximum d'inactivité.
- **private_sga** : taille privée de la SGA allouée à un utilisateur
- **unlimited** : limite de la ressource illimitée
- **default** : prend la limite par défaut de la ressource

Exemple :

```
CREATE PROFILE etud LIMIT  
SESSIONS_PER_USER 1  
CONNECT_TIME 480;
```

– Syntaxe partie password :

```
CREATE PROFILE profile LIMIT  
[FAILED_LOGIN_ATTEMPTS {expr | UNLIMITED | DEFAULT}]  
[PASSWORD_LIFE_TIME {expr | UNLIMITED | DEFAULT}]  
[PASSWORD_REUSE_TIME {expr | UNLIMITED | DEFAULT}]  
[PASSWORD_REUSE_MAX {expr | UNLIMITED | DEFAULT}]  
[PASSWORD_LOCK_TIME {expr | UNLIMITED | DEFAULT}]  
[PASSWORD_GRACE_TIME {expr | UNLIMITED | DEFAULT}]  
[PASSWORD_VERIFY_FUNCTION {function, NULL, DEFAULT}]
```

– Mots clés et paramètres :

- **Failed_login_attempts**: nombre d'échecs avant le blocage du compte
- **password_life_time**: durée en jours avant l'expiration du mot de passe
- **password_reuse_time**: durée en jours avant la réutilisation d'un password
- **password_reuse_max**: nombre de modif du password avant réutilisation
- **password_lock_time**: durée en jours du verrouillage d'un compte
- **password_grace_time**: délai de tolérance du password avant son expiration
- **password_verify_function**: fonction de contrôle des mots de passes

2. Attribution Profile à un utilisateur

```
CREATE USER user1  
  IDENTIFIED BY info  
  PROFILE etud;
```

Ou: ALTER USER user1
 PROFILE etud;

La vue **DBA_PROFILE** permet de consulter la liste des caractéristiques des utilisateurs.