

Didier LESTRAT

Consultant indépendant en Système d'information

Architecture d'un SI d'entreprise

- ▶ Les Principes
- ▶ Architecture 1 tier (stand alone)
- ▶ Architecture 2 tier
- ▶ Architecture 3 tier
- ▶ Architecture n tier

Définition d'un serveur d'application

Ce serveur est un ensemble de logiciel qui sert à séparer la logique métier de la logique présentation et de la logique base de données.

la base de données peut être séparée physiquement de la logique métier.

Les difficultés à surmonter pour implémenter un serveur d'application

- ▶ Accès simultanés des clients
- ▶ Surcharge des réseaux
- ▶ Autorisation d'accès aux bases de données
- ▶ Tolérance aux pannes, performances
- ▶ Equilibrage des charges
- ▶ Sécurité des données

Le serveur d'application doit donner une souplesse pour le développement et la mise en production des données.

Architecture d'un SI

Architecture 2-tiers

L'architecture à deux niveaux se compose de plusieurs clients (100 user) et d'un serveur.

Le client se connecte au serveur via des protocoles de réseaux (TCP/IP).

Le serveur contient la logique métier et stocke les données.

Le processus client collecte les informations au serveur (logique métier).

Le serveur implémente la logique métier et valide les données.

Client lourd : une partie de la logique métier est déplacé sur le poste client.

Client léger : le serveur gère toute la partie métier.

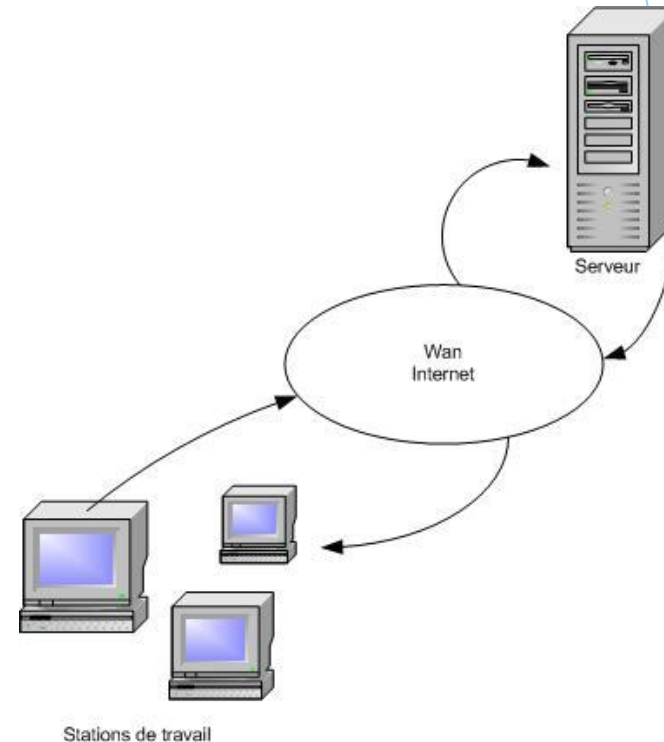
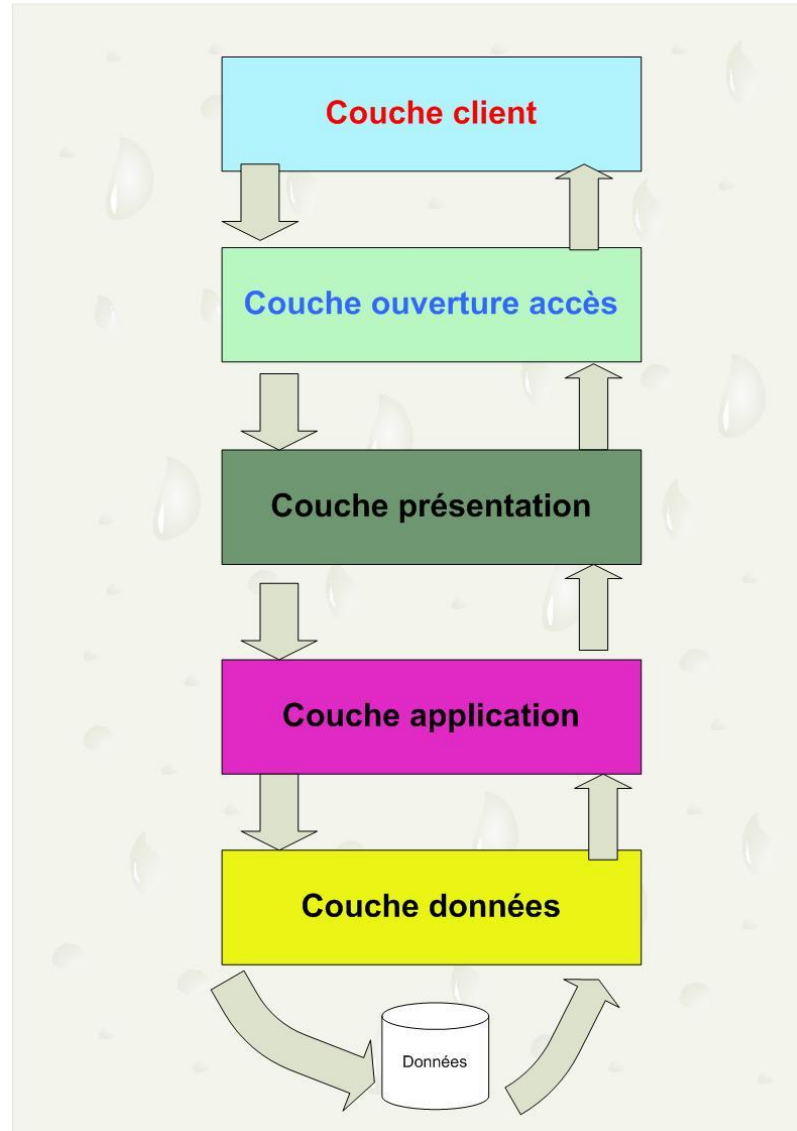
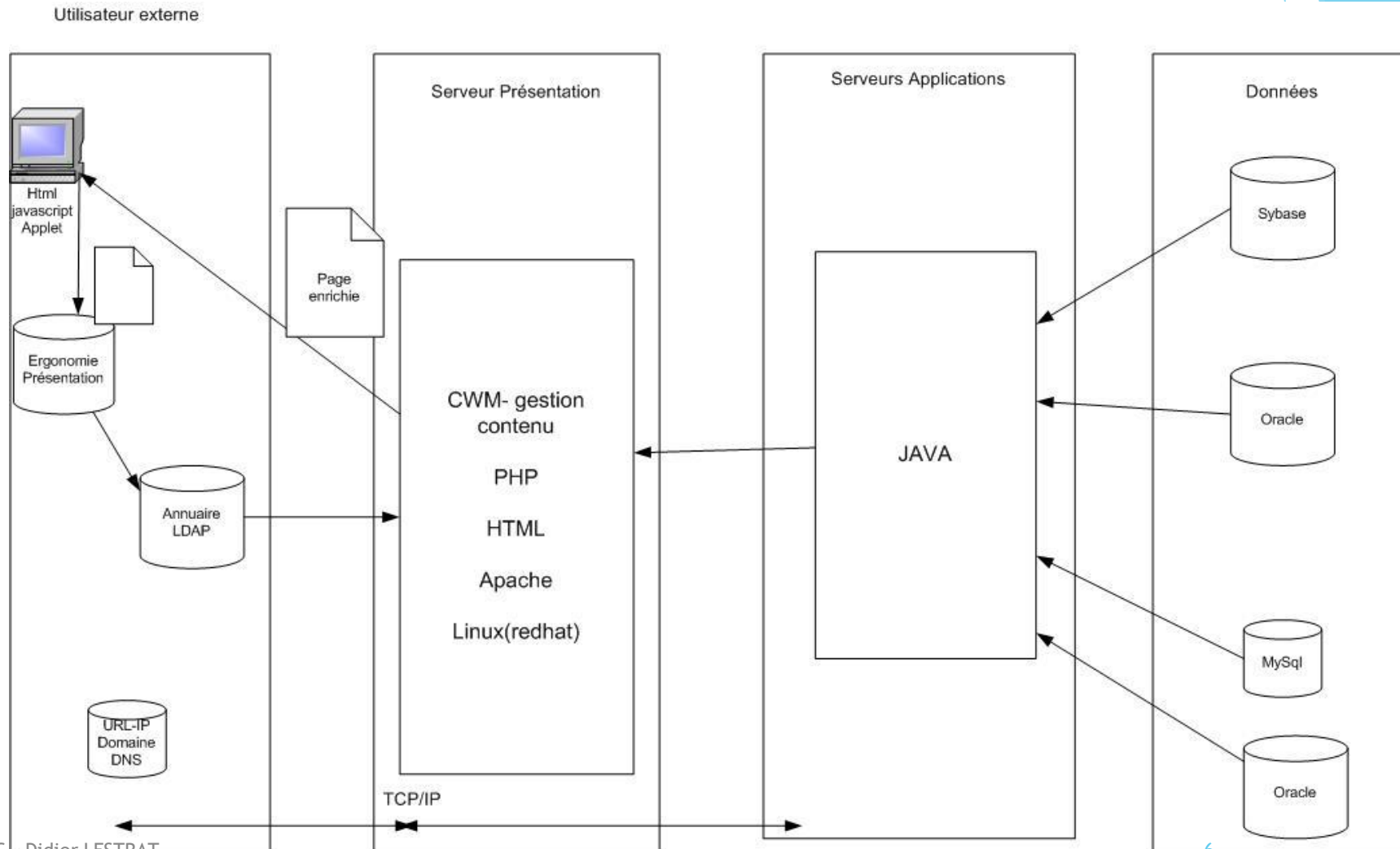


Figure3 : architecture 2-tiers

Architecture logique



Architecture physique



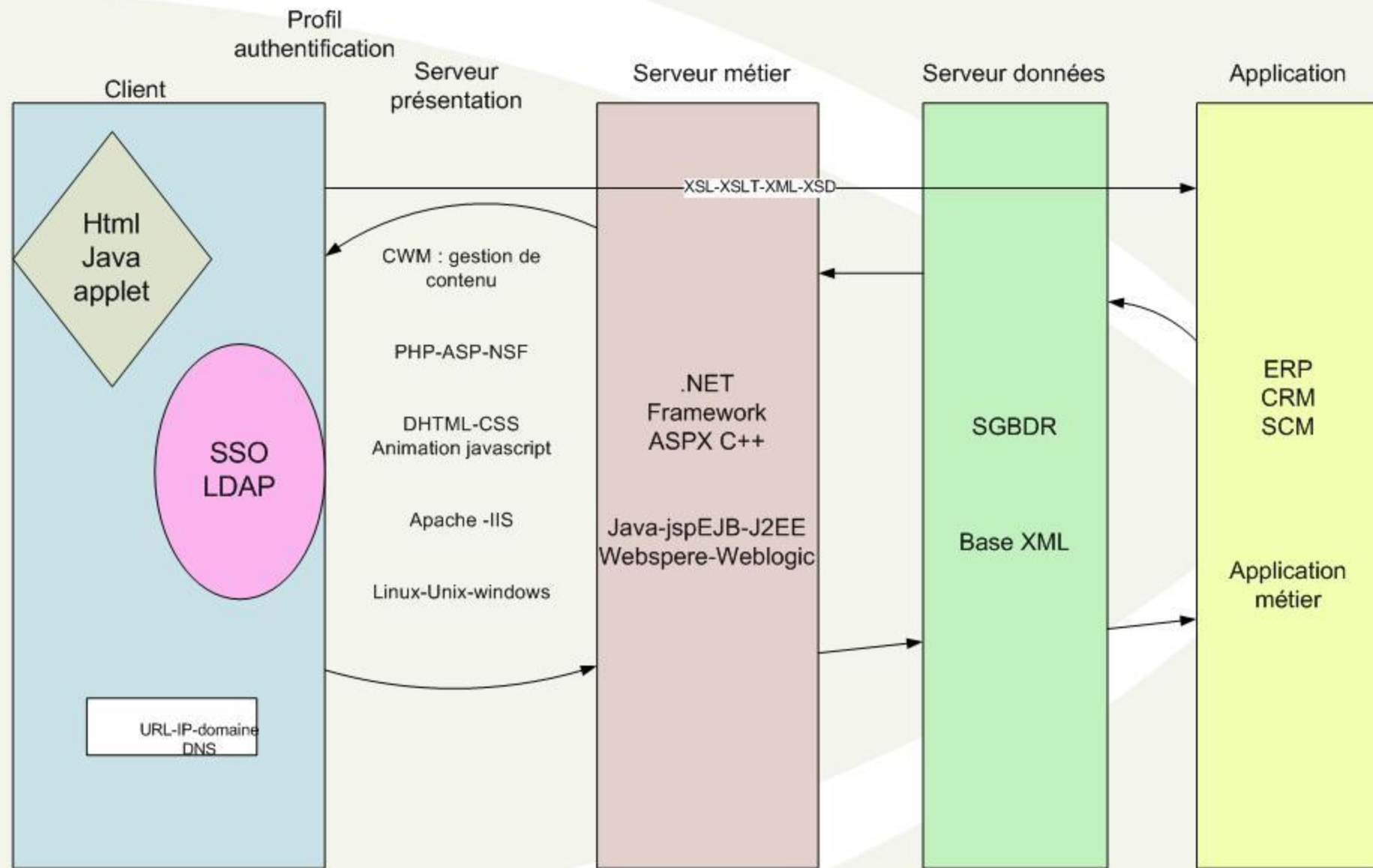


Figure 8 : Topologie d'un portail

SGBD ?

Objectifs

- ▶ Indépendance physique des données
- ▶ Indépendance logique
- ▶ Manipulation des données par des non informaticiens
- ▶ Efficacité d'accès
- ▶ Administration cohérente
- ▶ Pas de redondance
- ▶ Cohérence des données
- ▶ Partage des données
- ▶ Sécurité des données

Différence entre BDD et SGBD

- ▶ BDD : ensemble structuré de l'information
- ▶ SGBD : c'est la suite logicielle pour administrer l'ensemble de l'information

Catégorie d'instruction

DATA DEFINITION LANGUAGE (DDL)

- ▶ langage de manipulation des objets
 - ▶ Créer, manipuler, supprimer des objets
 - ▶ Autoriser ou interdire l'accès aux objets
 - ▶ Les instructions : Create, Alter, Drop, Grant, Revoke, Truncate
- ▶ interroger et mettre à jour les données sans préciser d'algorithme d'accès

▶ DATA MANIPULATION LANGUAGE (DML)

langage de manipulation des objets

- ▶ Ajout, suppression, modification de lignes
- ▶ Visualisation du contenu des tables
- ▶ Verrouillage des tables
- ▶ Les instructions : Insert, Update, Delete, Select, Lock Table.

Catégorie d'instruction

Transaction Control Language

- ▶ Gère les modifications des manipulations des données DML
 - ▶ Caractéristiques des transactions
 - ▶ Validation, annulation des modifications
 - ▶ Les instructions : Commit, Rollback, Savepoint

▶ Session Control Language

Gérer la session des utilisateurs

- ▶ Activation, désactivation des privilèges des utilisateurs
- ▶ Les instructions : Create user, Alter user, Grant

Objectifs et avantages des SGBD

Que doit permettre un SGBD ?

- ▶ **Contrôler les données**

- ▶ **Intégrité** : vérification de contraintes d'intégrité

ex.: le salaire doit être compris entre 4000 euros et 20000 euros

- ▶ **confidentialité** contrôle des droits d'accès, autorisation

⇒ langage de contrôle des données : DATA CONTROL LANGUAGE (DCL)

- ▶ **Partage**

une BD est partagée entre plusieurs utilisateurs en même temps

⇒ contrôle des accès concurrents notion de transaction

L'exécution d'une transaction doit préserver la cohérence de la BD

- ▶ **Sécurité**

- ▶ reprise après panne, journalisation

- ▶ Performances d'accès

- ▶ index (hashage, arbres balancés ...)

Objectifs et avantages des SGBD

Que doit permettre un SGBD ?

▶ Indépendance physique

- ▶ Pouvoir modifier les structures de stockage ou les index sans que cela ait de répercussion au niveau des applications
- ▶ Les disques, les méthodes d'accès, les modes de placement, le codage des données ne sont pas apparents

▶ Indépendance logique

- ▶ Permettre aux différentes applications d'avoir des vues différentes des mêmes données
- ▶ Permettre au DBA de modifier le schéma logique sans que cela ait de répercussion au niveau des applications

Modèle relationnel

A remplacer le modèle hiérarchique

- ▶ Le modèle de données relationnel est fondé sur la notion de relation : un tableau à deux dimensions qui contient un ensemble de n-uplets (les lignes).

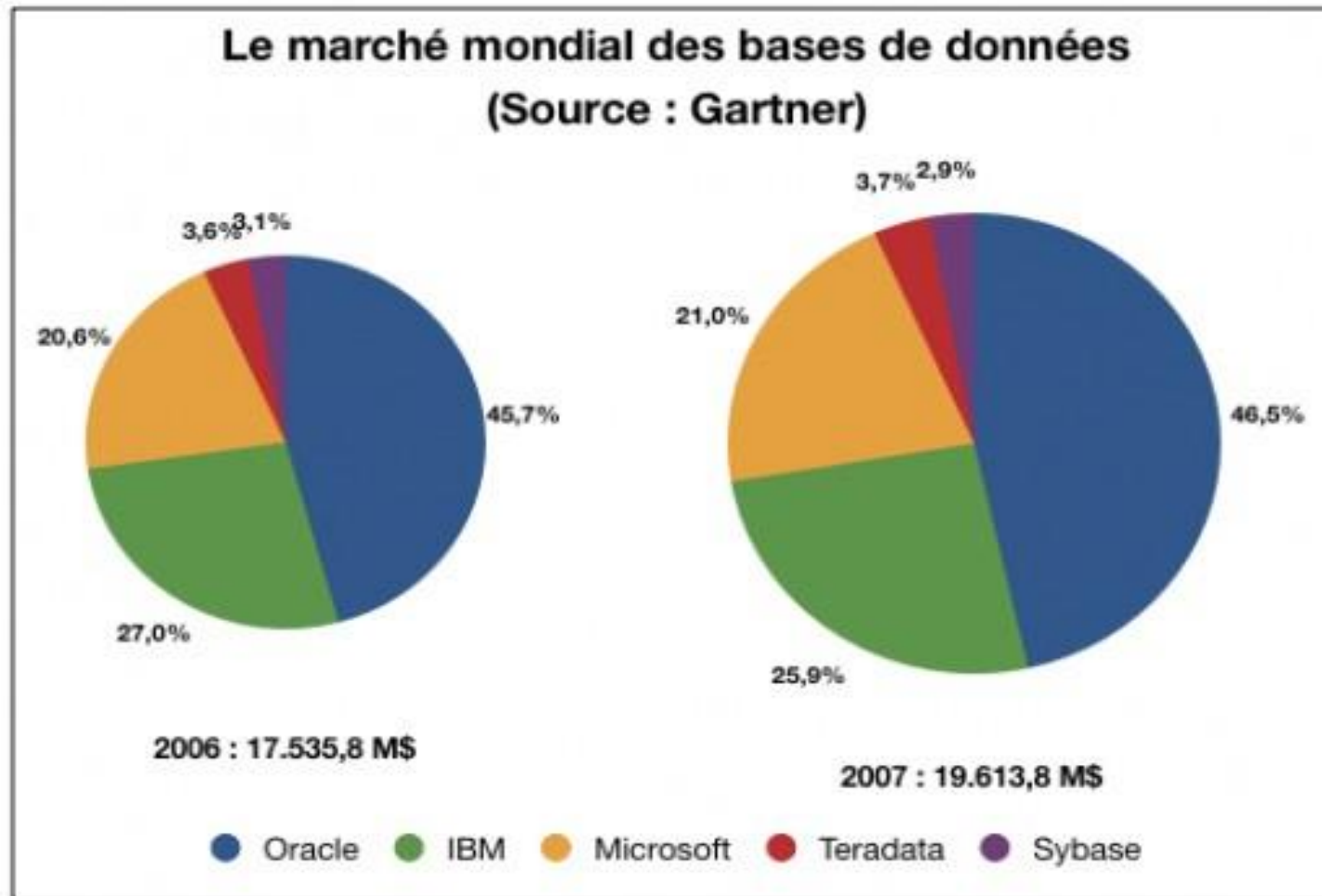
Quand on se focalise plus sur le stockage, les relations sont souvent appelées des tables et **les n-uplets** des enregistrements.

- ▶ Les entrées dans les tables sont appelées des valeurs.
- ▶ Les relations représentent les entités du monde réel (comme des personnes, des objets, etc.) ou les associations entre ces entités
- ▶ Notions de clés primaires et étrangères

Modèle relationnel

- ▶ SIMPLICITE DE PRÉSENTATION :
représentation sous forme de tables
- ▶ OPÉRATIONS RELATIONNELLES
algèbre relationnelle
- ▶ INDEPENDANCE PHYSIQUE
 - ▶ optimisation des accès
 - ▶ stratégie d'accès déterminée par le système
- ▶ INDEPENDANCE LOGIQUE
 - ▶ concept de VUES
- ▶ MAINTIEN DE L'INTEGRITÉ
 - ▶ contraintes d'intégrité définies au niveau du schéma

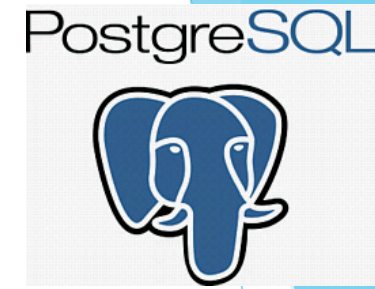
Les SGBD du marché



Evolution des SGBD du marché



PostgreSql



- ▶ Conforme à la norme SQL
- ▶ Fonctionne sous Solaris, Linux, SunOS, Windows

Outils d'administration

- ▶ Psql : outil de commande
- ▶ PgAdmin : Outil d'administration
- ▶ PhpPgAdmin : Outil d'administration sous interface web
- ▶ Pgpool : réplication

MySQL



Très développé sur toutes les applications web et chez les hébergeurs

- ▶ Conforme à la norme SQL
- ▶ Différents moteurs
 - ▶ Mysam : ne support pas les clés étrangères et les transactions
 - ▶ InnoDB : supporte les clés étrangères et les transactions
- ▶ Il gère la réplication et le clustering schéma

Fonctionne sous Solaris, Linux, SunOS, Windows

Edition MySQL 6.5

Outils d'administration

- ▶ MySQL administrator
- ▶ MySQL query browser
- ▶ Mysql Migration toolkit
- ▶ Mysql cluster

Sql server

- ▶ Architecture proche d'Oracle
- ▶ Beaucoup de demande depuis la mise en place des outils collaboratifs (SHAREPOINT)

Outils d'administration

- ▶ SQL Server Management Studio
- ▶ Assistant Paramétrage de base de données SQL Server.
- ▶ Outils d'invite de commandes, tels que sqlcmd.exe et osql.exe.

SyBase



Très stable

- ▶ Edition 15.5

Outils d'administration

- ▶ Adaptive server Enterprise
- ▶ Sql Anywhere

Oracle

- ▶ Fonctionne sur toutes les plateformes
- ▶ Edition 11G 12C 18C
- ▶ Bien implanté dans les entreprises
- ▶ Outils d'administration
 - ▶ Oracle entreprise manager
 - ▶ Oracle net Manager
 - ▶ Oracle installer

Administration SGBD Oracle

- ▶ Architecture d'un serveur ORACLE
- ▶ Préparation aux outils d'Oracle
- ▶ Création d'une base de données
- ▶ Administration d'une instance Oracle

Les métiers

- ▶ DBA
- ▶ Administrateur
- ▶ Responsable sécurité
- ▶ Administrateur réseau et système
- ▶ Développeur d'application (MCO MOE TMA)
- ▶ Utilisateurs métiers :
 - ▶ modifier le logiciel métier,
 - ▶ modifier les données,
 - ▶ créer des rapports
- ▶ Métiers du Big Data

L'administrateur de bases de données

Principales tâches

- ▶ Installation des produits
 - ▶ Un serveur, des applications clientes
 - ▶ Composants réseaux Oracle
- ▶ Création/démarrage/arrêt des bases de données
- ▶ Gestion des structures de stockage
- ▶ Gestion des utilisateurs (et de leurs droits) non métiers
- ▶ Sauvegarde/restauration.
- ▶ Export Import
- ▶ Création de script d'exploitation
- ▶ Optimisation et Tuning des bases
- ▶ Analyse des données

L'administrateur de bases de données

- ▶ Documentation d'exploitation
- ▶ Administration courante : Surveillance des traitements d'exploitation, performance, fluidité des flux, disponibilité
- ▶ AMOE sur l'architecture des bases

En général, le DBA se situe dans le groupe d'exploitation production avec l'équipe systèmes et réseaux.

Il est très rarement en MOE sauf dans les grandes structures.

Certaines entreprises externalisent l'administration des ses SGBD.

Architecture Oracle

Objectif de ce module

- ▶ Connaître les composants d'une architecture
- ▶ Expliquer la structure de la mémoire
- ▶ Expliquer les processus
- ▶ Expliquer la gestion de stockage physique et logique

Les concepts

Deux architectures possibles :

- Client/serveur : des applications clientes envoient des requêtes SQL ou PL/SQL à un serveur
- Architecture n-tiers : des serveurs d'application allègent la charge serveur

Un serveur de bases de données est composé :

- ▶ Une instance = Zone mémoire et des processus
- ▶ D'une base de données
- ▶ De plusieurs schémas, assimilés à des utilisateurs : SYS, SCOTT

Dans le cas d'un cluster de machines, Oracle peut associer plusieurs Instances

Architecture ORACLE

Connexion : lien de communication entre un processus utilisateur et l'instance de la base

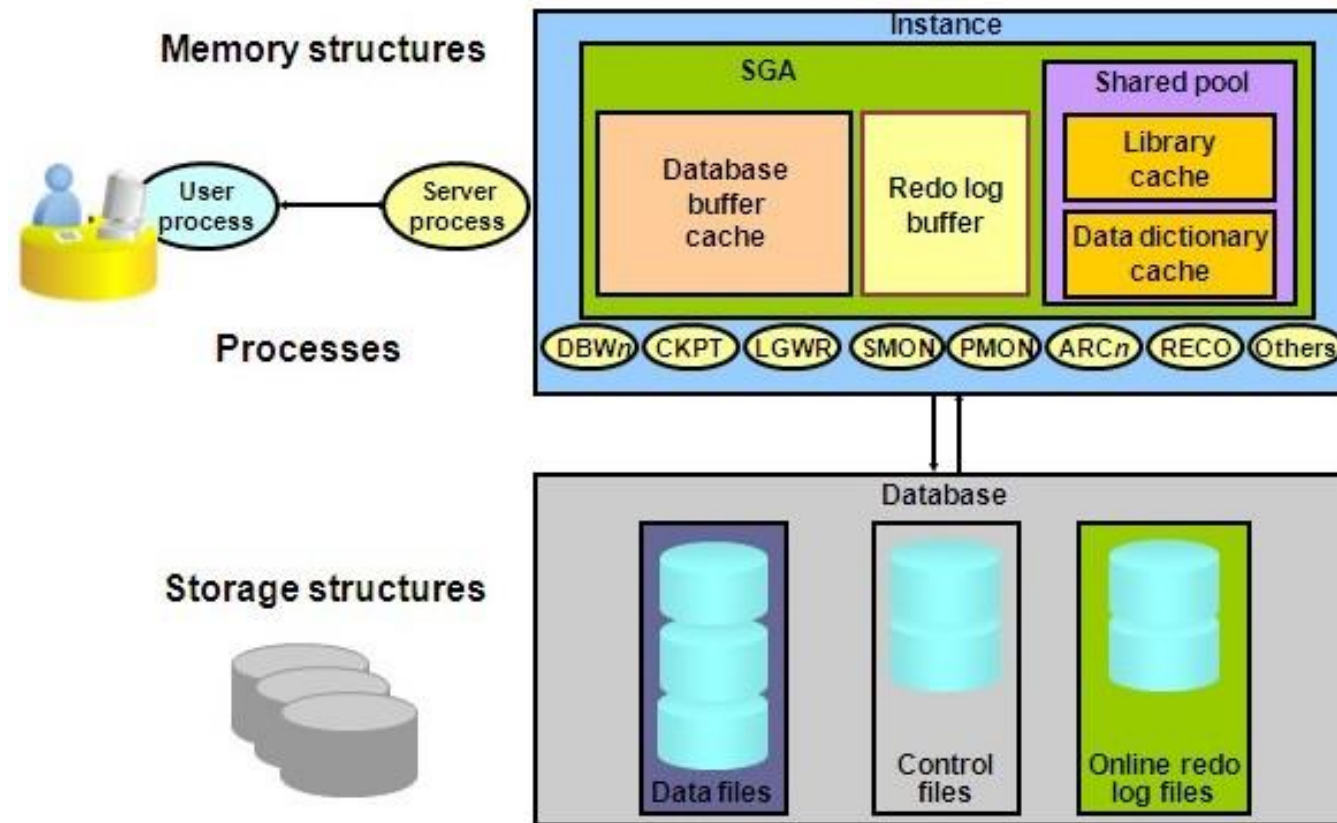
- ▶ Session : c'est le statut d'un profil utilisateur qui se logue sur une instance (plusieurs instances simultanée)
- ▶ Communication avec une instance Oracle

Processus serveur et un processus utilisateur

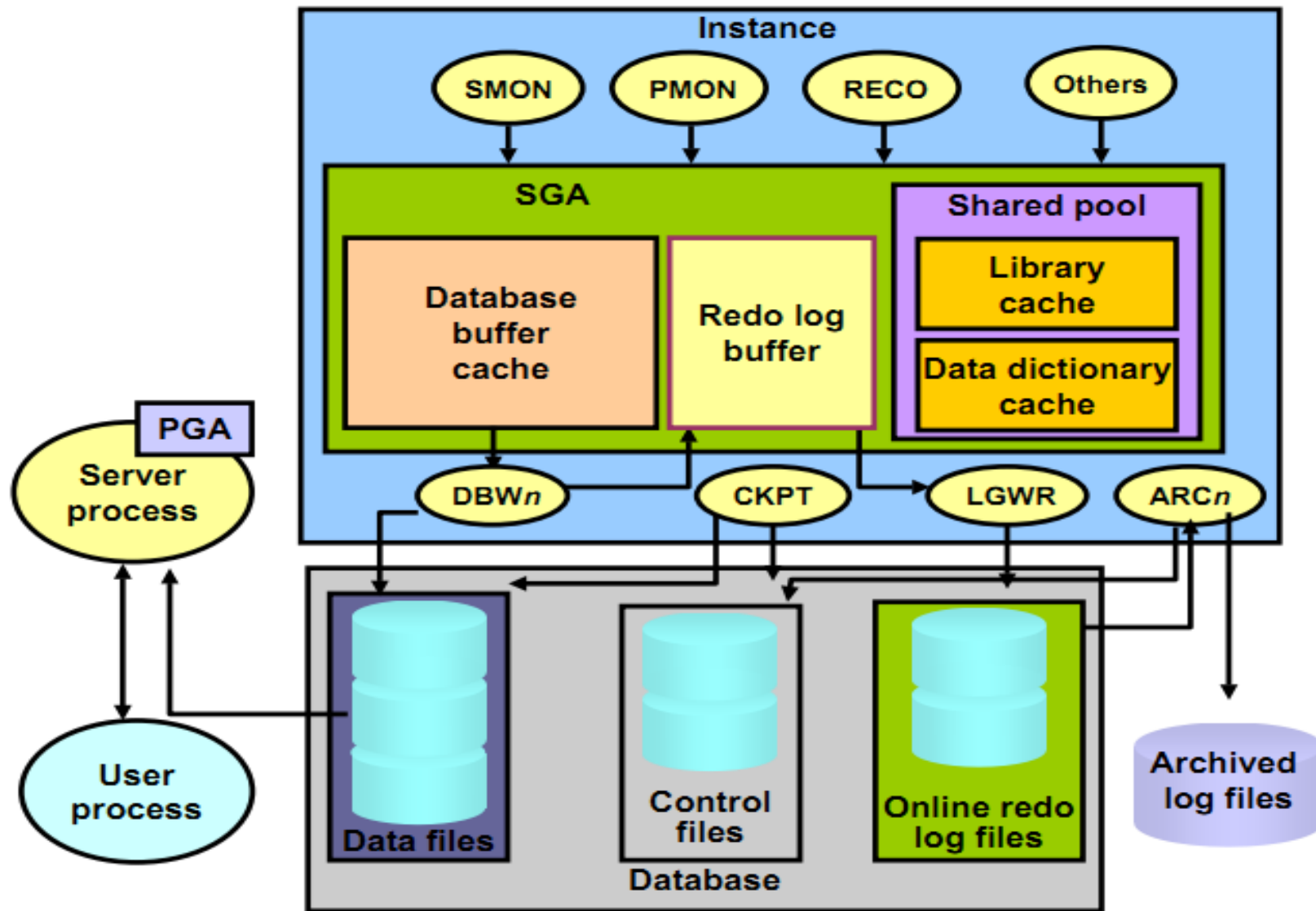
Principe :

- ▶ Le serveur récupère les valeurs des données et sont stockées dans le Buffer Cache
- ▶ Le processus serveur modifie les données dans le Buffer cache
- ▶ Si la transaction est correcte, le lien se fait par un retour de message, sinon un message d'erreur est activé. Ora 1020 ... une liste de descriptions d'erreur à visualiser pour analyser l'erreur

Process server et process user



Architecture logique et physique



Structure de la mémoire : SGA (System Global Area)

Elle contient les données et les informations de contrôle de l'instance.

Une instance est nommée SID

- ▶ Zone partagée par tous les utilisateurs
- ▶ La SGA est allouée au démarrage de l'instance et libérée lors de l'arrêt de l'instance.
- ▶ Elle est dimensionnée par un ensemble de paramètres définis dans le fichier de paramètres.
- ▶ La taille maximale de la SGA est limitée par le paramètre SGA_MAX_SIZE.

La SGA inclue les structures suivantes :

1. Database Buffer Cache : cache de données et de blocs
2. Redo Log Buffer : mémoire tampon pour l'enregistrement des modifications apportées à la base de données
3. Shared Pool : zone de partage des requêtes que peut partager plusieurs utilisateurs, et où sont exécutées les requêtes les plus utilisées
4. Large Pool : zone de mémoire optionnelle utilisée par différents processus comme les sauvegarde des opérations de reconstruction, les I/O...
5. Java Pool : Zone mémoire utilisée pour des sessions spécifiques Java et le virtuelle Java intégrée (JVM)
6. Streams Pool : zone de mémoire utilisée pour stocker de l'information des séquences spécifiques
7. *Result Cache (nouveau en version 11) : cache pour le résultat des requêtes SQL ou des fonctions PL/SQL.*

Structure de la mémoire SGA (System Global Area)

La SGA inclut les structures suivantes :

1. Database Buffer Cache : cache de données et de blocs
2. Redo Log Buffer : mémoire tampon pour l'enregistrement des modifications apportées à la base de données
3. Shared Pool : zone de partage des requêtes que peut partager plusieurs utilisateurs, et où sont exécutées les requêtes les plus utilisées
4. Large Pool : zone de mémoire optionnelle utilisée par différents processus comme les sauvegarde les opérations de reconstruction, les I/O...
5. Java Pool : Zone mémoire utilisée pour des sessions spécifiques Java et le virtuelle Java intégrée (JVM)
6. Streams Pool : zone de mémoire utilisée pour stocker de l'information des séquences spécifiques
7. *Result Cache (nouveau en version 11) : cache pour le résultat des requêtes SQL ou des fonctions PL/SQL.*

Quand vous lancez SQL plus ou OEM, la SGA alloue un espace dédié et unique. Paramètres SGA : `SGA_MAX_SIZE`

Database Buffer Cache

Il détient les copies de blocs de données qui sont les plus lus

Ce cache contient les résultats des requêtes les plus utilisés.

Au moment de l'exécution d'une requête, le processus regarde si le résultat n'est pas contenu dans le buffer, sinon il récupère les données dans le fichier de données pour l'exécuter au sein du Buffer.

- ▶ C'est une zone de partage des utilisateurs

Paramètres utilisés :

- ▶ DB_CACHE_SIZE : taille du cache pour la taille de bloc
- ▶ DB_nK_CACHE_SIZE : taille des blocs de nKo (valeurs de n 2, 4, 8, 16, 32)

Redo Log Buffer

- ▶ Ce sont des journaux où sont stockées toutes les modifications apportées à la base de données.
- ▶ Ce sont des opérations comme les DML (manipulation de données) et les DDL

Paramètre du Redo Log Buffer : Log_Buffer

Shared Pool

Une zone mémoire appartenant à la SGA qui contient :

- ▶ Library Cache : espace de partage SQL, espace où les ordres SQL et PL/SQL sont les plus utilisés
- ▶ Données du dictionnaire : les vues référentes à la base de données comme la description des tables, profils des utilisateurs, structures de la base...
- ▶ Des structures de contrôle

Paramètre utilisé : SHARED_POOL_SIZE

Large Pool

L'administrateur peut configurer un espace optionnel de mémoire afin d'allouer plus d'espace pour :

- ▶ I/O
- ▶ La restauration ou la sauvegarde

PGA (Program Global Area)

C'est un espace mémoire qui contient les données et les informations de contrôle du processus serveur.

Le processus serveur est un service pour répondre aux requêtes clients.

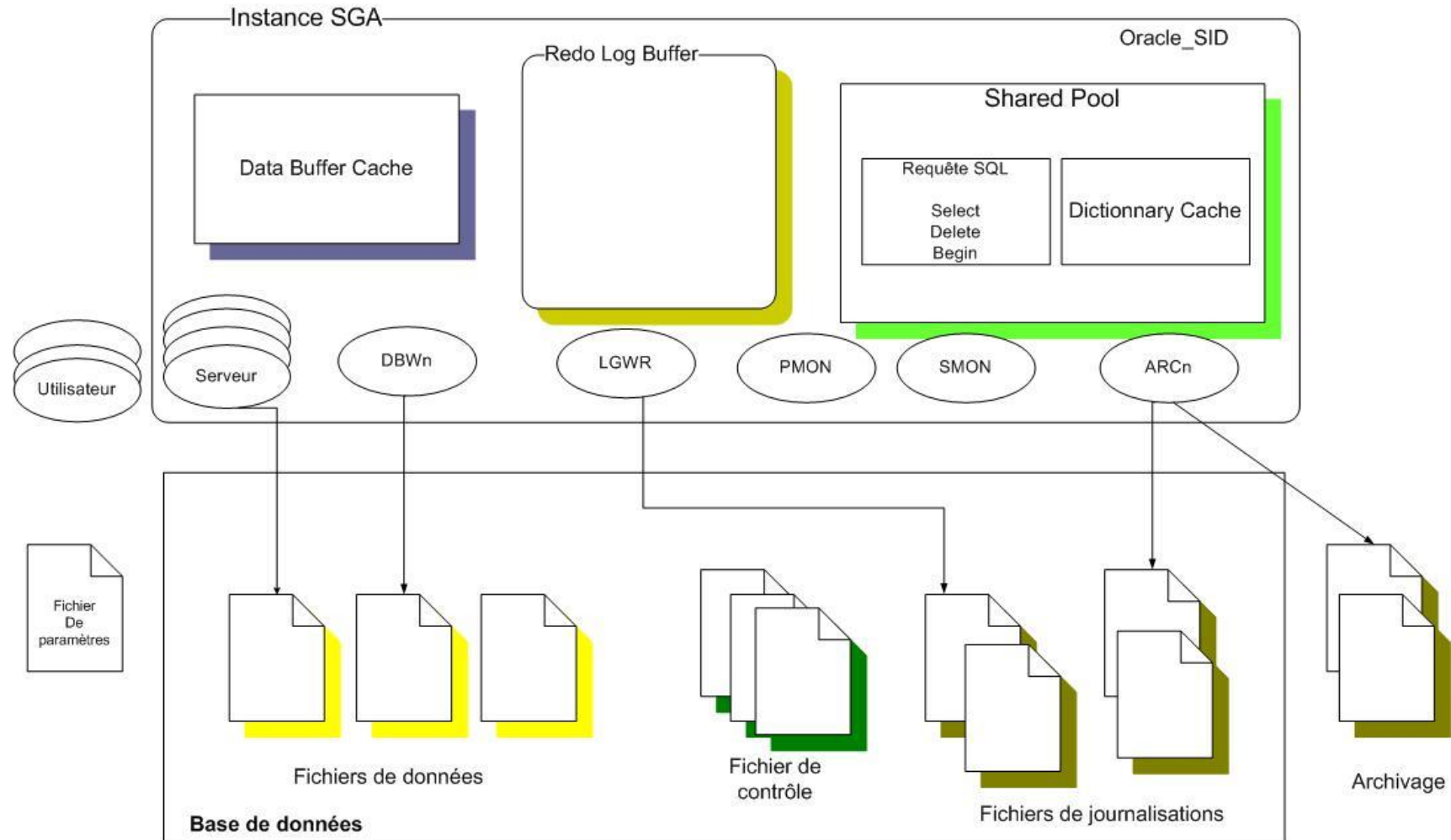
L'architecture des traitements

- ▶ Le processus utilisateur : est démarré au moment où un utilisateur se connecte à une base de données via un script, accès distant, bureau distant

Les processus de la base de données :

- ▶ Les processus du serveur : connexion à l'instance et démarre lorsqu'une requête utilisateur arrive
- ▶ Les traitements de fonds : ils sont démarrés au moment où l'instance est démarrée

Architecture des traitements



Cartographie de la Structure des traitements

▶ Data writer process (DBWn) :

Ecriture des modifications faites au niveau du Buffer Cache vers les fichiers de données sur le disque

Il se déclenche :

- ▶ Quand il manque de place dans le buffer
- ▶ De façon périodique avec le processus checkpoint

▶ Log Writer (LGWR): les journaux d'écritures

Il gère l'écriture du Redo Log Buffer vers le redo Log file sur le disque

Il se déclenche :

- ▶ Au moment d'un Commit
- ▶ Le buffer est au tiers plein
- ▶ Avant que le processus DBWn écrit les modifications dans le fichier de données sur le disque
- ▶ Tous les 3 secondes

Cartographie de la Structure des traitements

Checkpoint :

C'est le mécanisme de synchronisation pour les écritures des en-têtes vers les fichiers de données ou les fichiers de contrôles.

Il caractérise le point de reprise d'une transaction (donnée de structure avec un numéro unique SCN).

Il est important pour toute reconstruction d'une base suite à incident.

SMON (system monitor) :

- ▶ Il récupère une instance après un arrêt anormal.
- ▶ Il nettoie les segments temporaires
- ▶ Roll Back : suppression des données de transactions non validées
- ▶ Roll forward mise à jour des données suite à une transaction validée avant l'arrêt de l'instance

PMON (process monitor) :

- ▶ Il nettoie suite à un plantage d'un processus utilisateur ou après un délai trop long.
- ▶ Il libère les verrous.

Cartographie de la Structure des traitements

RECO (Recover Process)

ARCn (Archiver process)

- ▶ C'est le traitement d'écriture afin d'archiver les fichiers journaux.
- ▶ La base est soit en mode archive log ou en noarchivelog.
- ▶ Le mode Archivelog prend beaucoup d'espace sur les disques, il existe le paramètre LOG_ARCHIVE_MAX_PROCESS pour une gestion automatique des suppressions des archives les plus anciennes.

Architecture du stockage de la base de données

Fichiers de contrôle : il comprend toutes les informations sur l'intégrité d'une base de données et sa structure

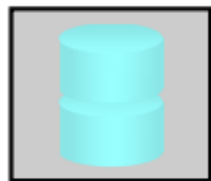
Fichiers de données : Stockent les données

il contient les objets des utilisateurs ou les données d'application de la base de données, ils stockent le dictionnaire de données.

D'autres catégories de fichiers physiques sont important aussi afin que la base de données fonctionne correctement.

- ▶ Fichier de paramètres (parameter file) : on retrouve les informations
 - ▶ D'une instance.
 - ▶ Nom de la base de données
 - ▶ Les chemin des fichiers de contrôle de la base

DB structures
- Memory
- Process
→ **Storage**



Control files



Data files



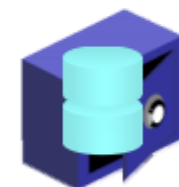
Online redo log files



Parameter file



Backup files



Archived redo log files



Password file



Alert log and trace files

Architecture du stockage de la base de données

- ▶ Fichier des mots de passe (Password file) : Privilèges sysdba, sysoper and sysman
- ▶ Fichiers de sauvegarde (Backup files) : utilisé pour la restauration d'une base de données.
- ▶ Fichiers des journaux archivés (archived redo log files)
- ▶ Fichiers de trace (Trace files) : Inscrit toutes les erreurs de traitements, important pour le DBA.
- ▶ Fichiers d'alerte (Alert log File) : chronologie des erreurs, un fichier à vérifier régulièrement

Atelier 1

Visualiser l'arborescence du logiciel Oracle

- ▶ Retrouver les fichiers suivant :
 - ▶ Redo log Files
 - ▶ Datafiles
 - ▶ Control files
 - ▶ Parameters files
 - ▶ Trace files
 - ▶ Init.ora
- ▶ Donner le chemin complet d'Oracle Home et Oracle BASE
- ▶ Répertoire contenant les outils Oracle
 - ▶ DBCA, NETCA, SQL DEVELOPPER
 - ▶ Lancer les outils DBCA, NETCA

Structure logique et physique d'une base de données

Un schéma est un ensemble d'objets de la base données qui sont détenues par l'utilisateur de la base de données.

Les objets du schéma comprennent les tables, les vues, les procédures, indexes...

Tablespaces

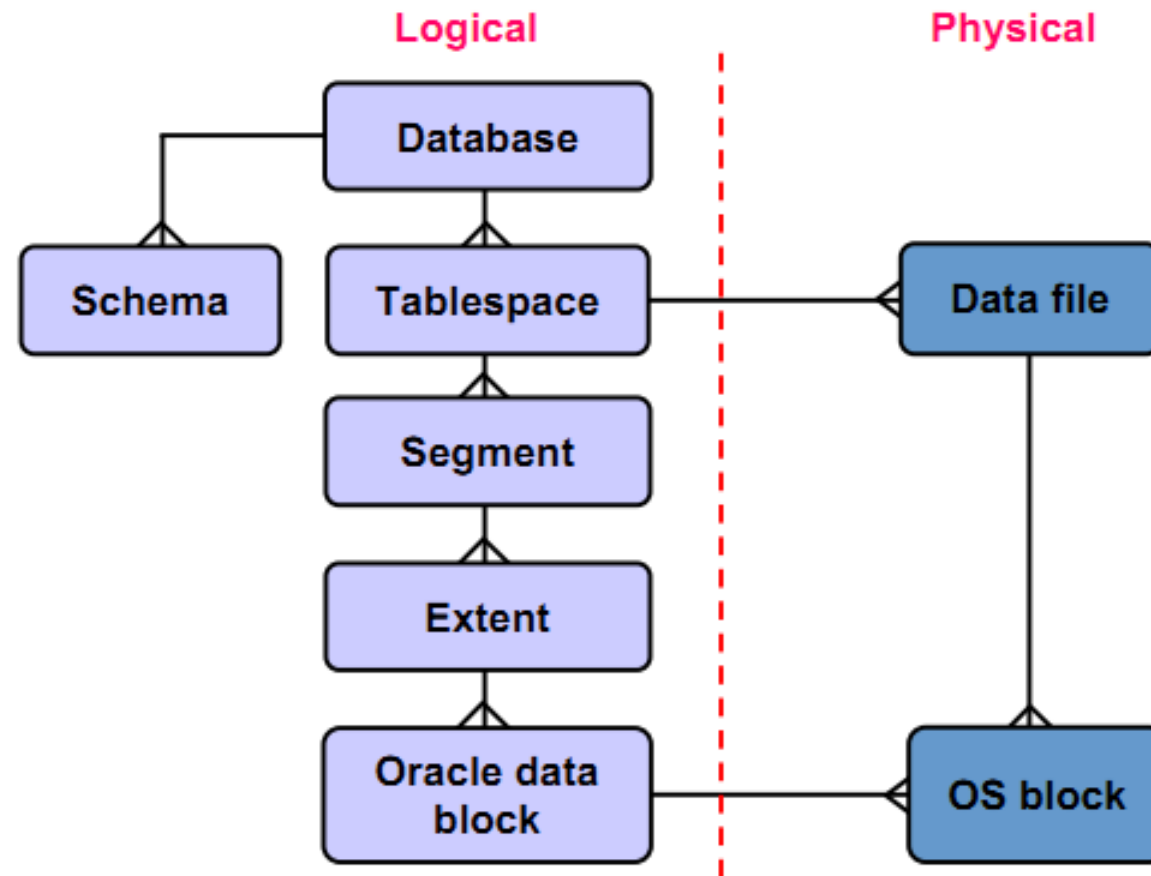
- ▶ Une base de données est divisée en unité de stockage logique appelée Tablespaces. Efficace pour les performances d
- ▶ Vous devez avoir un tablespace pour les données et un autre pour les indexes, les utilisateurs,....

Tablespaces system and sysaux

- ▶ System tablespace : utilisé pour les fonctionnalités fondamentales, données du dictionnaire
- ▶ Sysaux : utilisé pour rajouter des composants de la base de données(comme le référentiel d' OEM)

C'est deux tablespaces sont automatiquement créés au moment ou une base de données est créée.

Structure logique et physique d'une base de données



Structure logique

- ▶ **Block de données** est l'unité logique la plus fine , 8 K0 par défaut

- ▶ **L'extent** est un ensemble contiguë de blocks

Nota : On ajoute des extents lorsque le segment est plein. C'est un automatisme d'Oracle.

- ▶ **Tablespace** : regroupe un ensemble d'objets pour faciliter l'administration
- ▶ Il peut atteindre 8 exabytes
- ▶ Une tablespace peut regrouper un ou plusieurs fichiers de données

Structure logique

Le Segment : est un ensemble d'extents dédié à un même objet

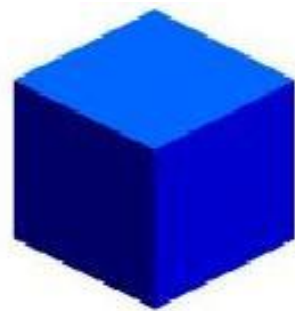
C'est l'espace utilisé par une structure logique : tables, indexes,...

- ▶ Les différents types de segments sont les suivants :
 - ▶ Table
 - ▶ Partition de table
 - ▶ Cluster
 - ▶ Table organisée en index
 - ▶ Segment LOB
 - ▶ Table imbriquée
 - ▶ Index
 - ▶ Partition d'index
 - ▶ Index lob
 - ▶ Rollback segment
 - ▶ Segment temporaire
 - ▶ Segment de démarrage

Structure physique

Les fichiers de données : c'est l'espace physique sur le disque du serveur qui stocke toutes les données de la base

- ▶ Un ou plusieurs fichiers de données sont créés pour chaque tablespace sur un stockage physique..
- ▶ La taille maximum est la taille du bloc pour la tablespace multiplié par 2^{36} , soit 128TB pour une taille de bloc de 32KB.



Segment



Extents



**Data
blocks**



**Disk
blocks**

Contenu d'un schéma

Appartient à un utilisateur

Ensemble d'objets de l'utilisateur manipulable en SQL

Tables, index, clusters, triggers, vues, dimensions

Un objet correspond à

- ▶ Plusieurs extents
- ▶ Un tablespace
- ▶ Un ou plusieurs fichiers physiques sur le disque.

Stockage des tables

- ▶ Pour des tables de moins de 256 attributs

Un tuple est entièrement dans un bloc

Sinon : chaînage inter-blocs (augmente les E/S)

- ▶ Pour des tuples de plus de 256 attributs

Chaque partie de 255 tuples est chaînée en intra-bloc

Les tuples sont chaînés en utilisant le ROWID

A la création d'une table on spécifie :

- ▶ La tablespace
- ▶ Spécifier la taille du segment
- ▶ La quantité d'espace libre laissée dans chaque bloc

Compte DBA

Comptes Oracle d'administration

Il est souhaitable de créer plusieurs comptes administrateur

- ▶ L'administration quotidienne
- ▶ Export/import des bases
- ▶ Sauvegarde/restauration
- ▶ Administrateur restreint

Une base de données Oracle contient toujours deux comptes ayant les privilèges d'administrateur

- ▶ **SYS** (mot de passe par défaut : `change_on_install`)

SYS est le propriétaire du dictionnaire de données

- ▶ **SYSTEM**(mot de passe par défaut : `manager`).

SYSTEM peut être propriétaire de tables complémentaires utilisées par les outils Oracle.

Compte DBA

Depuis Oracle9i Release 2, ces mots de passe par défaut peuvent être changés lors de la création de la base de données.

Ces comptes peuvent être utilisés indifféremment pour l'administration courante (gestion des utilisateurs, du stockage, etc.) uniquement lorsque la base est démarrée.

Un **privilege** supplémentaire particulier (SYSDBA ou SYSOPER) est nécessaire pour certaines opérations (démarrage, arrêt, etc.). De plus, l'activation de ce privilege SYSDBA ou SYSOPER nécessite un mécanisme d'authentification particulier, puisque la base de données peut ne pas être disponible. Cette authentification s'effectue soit par le système d'exploitation, soit par un fichier de mot de passe.

Principe de connexion

Connexion système d'exploitation

- ▶ Salle blanche
- ▶ Outil de connexion réseau : VNC, Telnet, Putty
- ▶ Outils de commande : cmd
- ▶ Canal sécurisé ou non sécurisé : SSL, SSH
- ▶ Droits et rôle de connexion sur les serveurs

Principe de connexion

Connexion à l'instance Oracle sous SQLPLS

- ▶ Connexion direct via un outil de commande
`CONNECT / AS { SYSDBA | SYSOPER }`

- ▶ Connexion via un fichier de mots de passe

Pour utiliser l'authentification par un fichier de mot de passe, vous devez mettre le paramètre d'initialisation

`REMOTE_LOGIN_PASSWORDFILE` à `EXCLUSIVE`

(défaut) ou `SHARED`

et créer un fichier de mot de passe à l'aide de l'utilitaire **orapwd** fourni par Oracle.

Comptes DBA

- ▶ SYS et SYSTEM

Lors de la création d'une base de données, d'autres comptes Oracle peuvent être créés, notamment SYSMAN et DBSNMP.

- ▶ SYSMAN (mot de passe par défaut CHANGE_ON_INSTALL) est un compte qui peut être utilisé pour effectuer des tâches d'administration dans Oracle Enterprise Manager. SYSMAN est un compte DBA.
- ▶ DBSNMP (mot de passe par défaut DBSNMP) est un compte utilisé par l'agent d'Oracle Enterprise Manager pour superviser et gérer une base de données.
- ▶ De nombreux autres comptes "administratifs" peuvent être créés selon les options installées dans la base de données.

Privilèges SYSDBA et SYSOPER

- ▶ Le privilège SYSDBA permet toutes les opérations "lourdes" d'administration
 - ▶ la création d'une base de données
 - ▶ les arrêts et les démarrages
 - ▶ la création d'un fichier de paramètre serveur
 - ▶ les récupérations, etc.
 - ▶ Il donne un accès à toutes les données de la base de données.
 - ▶ La connexion s'effectue implicitement dans le schéma de SYS.
- ▶ Le privilège SYSOPER donne à peu près les mêmes droits que SYSDBA, à l'exception notable de la création de la base de données.
- ▶ Par contre, l'accès est restreint aux seules données du dictionnaire de données. La connexion s'effectue implicitement dans le schéma PUBLIC.

Installation du serveur et des logiciels

Pour bien commencer, il est bon de connaître

- ▶ les principes de bases d'une installation ORACLE
- ▶ De se fier aux recommandations de l'éditeur
 - ▶ sur le nommage des fichiers
 - ▶ l'arborescence sur le disque.
 - ▶ Paramétrages
 - ▶ Documentation oracle

Construire une base

- ▶ MCD : créer le modèle conceptuelle des données
- ▶ Réflexion sur les tables, indexes à venir, estimer leur taille
- ▶ Espace mémoire et espace physique
- ▶ Choisir l'encodage des caractères
- ▶ Déterminer la taille des blocs de données
- ▶ Choisir le mode de gestion d'annulation
- ▶ Tablespace dédié
- ▶ Segments d'annulations
- ▶ Stratégie de sauvegarde et archivage

OFA (optimal flexible architecture)

Se référer à OFA pour

- ▶ Une meilleure organisation
- ▶ Faciliter l'administration
- ▶ Faciliter la relation entre plusieurs bases de données ORACLE
- ▶ Administrer et gérer la montée en charge des bases
- ▶ Séparer les fichiers d'administration et les fichiers de données
- ▶ D'avoir plusieurs versions d'Oracle installées sur le serveur

Répertoires Oracle

Deux répertoires jouent un rôle particulier

Sous Windows

- ▶ Oracle Base : répertoire racine de l'arborescence Oracle
- ▶ Oracle Home : sous répertoire d'Oracle base contenant le logiciel
- ▶ dans le répertoire base, nous pouvons avoir plusieurs version d'ORACLE

Nota : vérification des nommages des fichiers dans la base de registre de Windows

Sous Unix ou Linux

- ▶ Oracle Base : /ora01/app/oracle
- ▶ Oracle Home : /ora01/app/oracle/product/11.2.0/db_1

(stockage des logiciels et des applications oracle)

- ▶ Oracle Base contient d'autres répertoires
 - ▶ Oradata : fichiers des bases de données
 - ▶ Admin : fichiers d'administration
 - ▶ Cfgtoollogs : journaux des installations
 - ▶ Diag : référentiel du diagnostic

Convention d'installation

- ▶ Arborescence et nommage

- ▶ /ORA01

- ▶ /disk01

- ▶ Dossiers

- ▶ Oracle_base : /ora01/app/oracle

- ▶ Oracle_home : /ora01/app/oracle/product/11.2.0/db_1

- ▶ Fichiers

- ▶ Fichiers de contrôle : /ora01/app/oracle/oradata-nom_instance/controln.ctl

- ▶ Redo log files : /ora01/app/oracle/oradata/nom_instance/redo.log

- ▶ Fichier de données :
/ora01/app/oracle/oradata/nom_instance/tn.dbf

(n est un numéro d'ordre)

Les variables d'environnement

- ▶ ORACLE_BASE
- ▶ ORACLE_HOME
- ▶ ORACLE_SID : initiale du nom de l'instance
- ▶ NLS_LANG : spécifie le langage supporté

Voir le regedit sous Windows

Et le bash_profile pour Linux

ORACLE Universal Installer (oui)

Il faut être administrateur système ou membre d'un groupe administrateur pour installer le logiciel Oracle et ses outils

Il se trouve dans le dossier products

et se lance avec `./runinstaller` avec le compte Oracle

Pré-requis Windows

- ▶ Os supporté : Window 2008 et w2012 server
 - ▶ Mémoire : 1GB /instance
 - ▶ Configuration minimum du serveur 1,5GB de mémoire
 - ▶ 400 MB espace disque
 - ▶ 1,5GB et 3,5 Gb pour les outils Oracle
 - ▶ 1, 2 Gb / base
 - ▶ 2,4 GB pour l'espace de restauration Vidéo adapter 256 MO

Pré-requis Unix- Linux

- ▶ OS supporté
 - ▶ Oracle enterprise 4 ou Redhat enterprise 4 (noyau2.6.9)
 - ▶ Oracle enterprise 5 ou Redhat enterprise 5 (noyau2.6.18)
 - ▶ SUSE enterprise linux 10 (noyau 2.6.16.21)
 - ▶ Cent OS 7 cent OS 6

Le serveur

- ▶ Mémoire : 1GB /instance
- ▶ Configuration minimum du serveur
- ▶ 1,5GB de mémoire
- ▶ 400 MB espace disque
- ▶ 1,5GB et 3,5 Gb pour les outils Oracle
- ▶ 1, 2 Gb / base
- ▶ 2,4 GB pour l'espace de restauration
- ▶ Vidéo adapter 256 MO

Configuration du réseau ORACLE

Installer un Listener en utilisant Oracle Net configuration

Rôle du Listener :

écouter et transmettre les données envoyées entre le client et le serveur

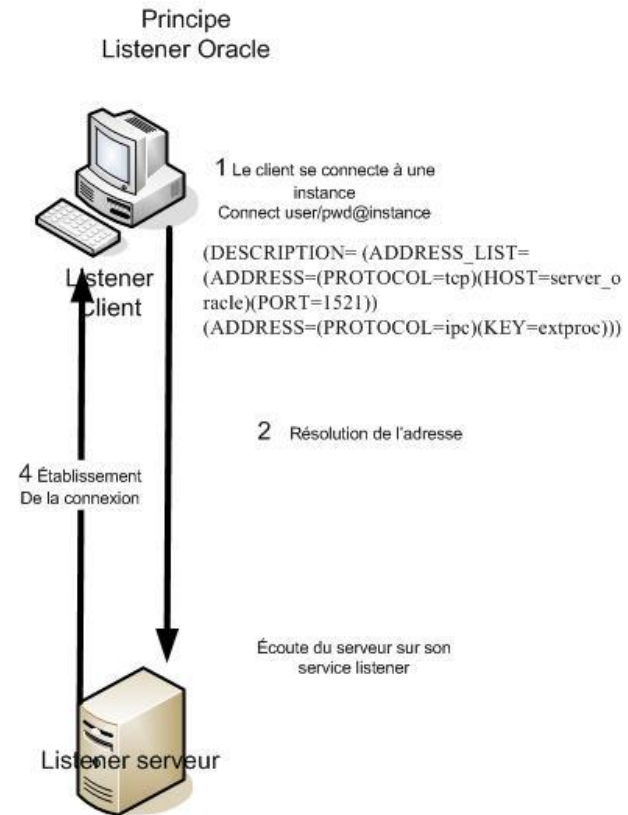
Par défaut le fichier listener.ora est dans le dossier

- ▶ \$ORACLE_HOME/network/admin sous Unix
- ▶ %ORACLE_HOME%\network\admin sous Windows

Fichier listener.ora : vous pouvez conserver le fichier et le déployer sur tous les clients Oracle étant donné qu'il n'est pas spécifique à un client.

Nota : si vous avez une identification via un LDAP : il faut avoir un outil tiers (non fourni par ORACLE)

Principe du Listener ORACLE



Configuration côté serveur : listener.ora

Installation avec Oracle net configuration

On peut aussi créer un script listener.ora
sous l'arborescence :

D:\oracle\app\oracle\product\11.2.0\server\network\ADMIN

Atelier : création du fichier listener.ora

Configuration côté serveur : listener.ora

Configuration côté client : tnsname.ora

Commande : >lsnrctl start(status-stop) LISTENER

LSNRCTL> help

LSNRCTL>

Configuration côté client

- ▶ sélectionner les méthodes de résolution, elles sont stockées dans le fichier sql.ora
- ▶ Lancer Oracle net configuration et suivre les instructions
- ▶ Visualiser le fichier tnsname.ora
- ▶ Lancer la commande tnsping pour visualiser la connexion réseau

Nota ; le fichier tnsname.ora n'est pas fixé à un client. Vous pouvez le dupliquer sur d'autres machines clients en le maintenant dans un dossier de distribution

Plusieurs connexion sont envisagées pour une connexion client/instance base

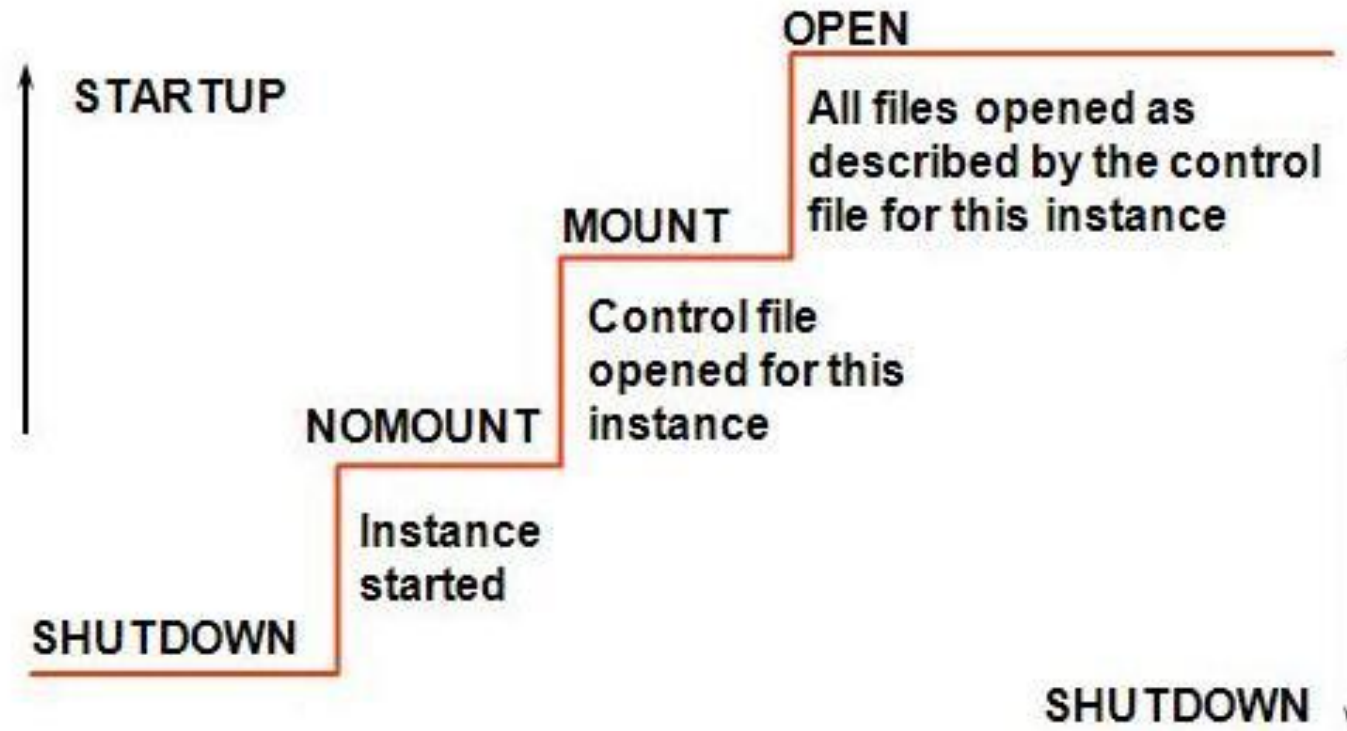
- ▶ Connexion locale
 - ▶ SQL> connect user/password@instance
 - ▶ SQL> connect sys/oracle2018@orcl
- ▶ Connexion réseau : TCP/IP
 - ▶ SQL> connect user/password@<hostname>:listenerport/servicename
 - ▶ SQL> connect user/password@server.esaip.org:1521/orcl

Tester avec le commande TNSPING

Local : >tnsping nom_instance

Via réseau >tnsping server.esaip.org:1521/orcl

Gestion de l'instance et d'une base



1. Démarrage de l'instance
2. Montage de la base de données
3. Ouverture de la base

Gestion de l'instance et d'une base

1. Démarrage de l'instance
2. Montage de la base de données
3. Ouverture de la base

STARTUP [NOMOUNT | MOUNT [nom_base] | OPEN [nom_base]]
[RESTRICT] [PFILE=nom_fichier] Avec

NOMOUNT | MOUNT | OPEN

\$ Export ORACLE_SID=orcl

\$ sqlplus /nolog

SQL> CONNECT /AS SYSDBA

SQL> STARTUP

Nota : ALTER DATABASE permet de passer d'un mode à l'autre

Arrêt de l'instance et de la base

Syntaxe

▶ SHUTDOWN [NORMAL | IMMEDIATE | TRANSACTIONAL | ABORT]Options :

▶ NORMAL

Oracle attend que tous les utilisateurs soient déconnectés (pas de nouvelle connexion autorisée) puis ferme proprement la base de données.

▶ IMMEDIATE

Oracle déconnecte tous les utilisateurs (en effectuant un ROLLBACK des éventuelles transactions en cours) puis ferme proprement la base de données.

▶ TRANSACTIONAL

Oracle attend que toutes les transactions en cours se terminent avant de déconnecter les utilisateurs (pas de nouvelle transaction autorisée) puis ferme proprement la base de données.

Arrêt de l'instance et de la base

► ABORT

Oracle déconnecte tous les utilisateurs (sans effectuer de ROLLBACK des éventuelles transactions en cours) puis ferme "brutalement" la base de données, sans effectuer de point de synchronisation (checkpoint). Une récupération de l'instance sera nécessaire lors du prochain démarrage

Nota : pour vérifier si des utilisateurs sont connectés.

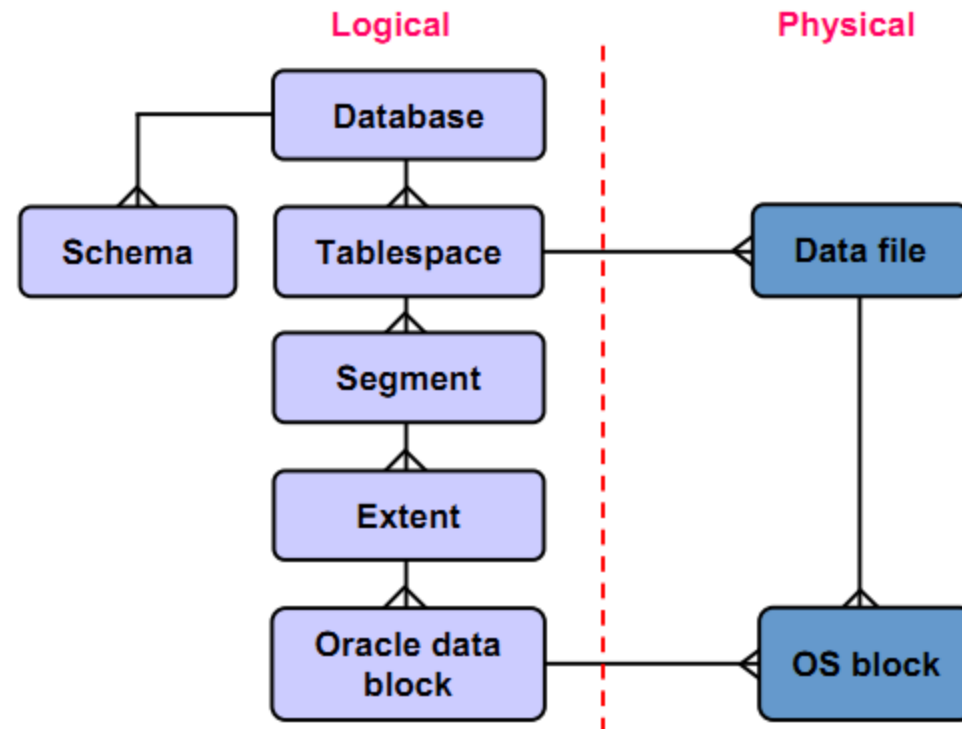
► `Sql>SELECT sid,serial#,username,DECODE(taddr,NULL,',','Oui') trans
FROM v$session`

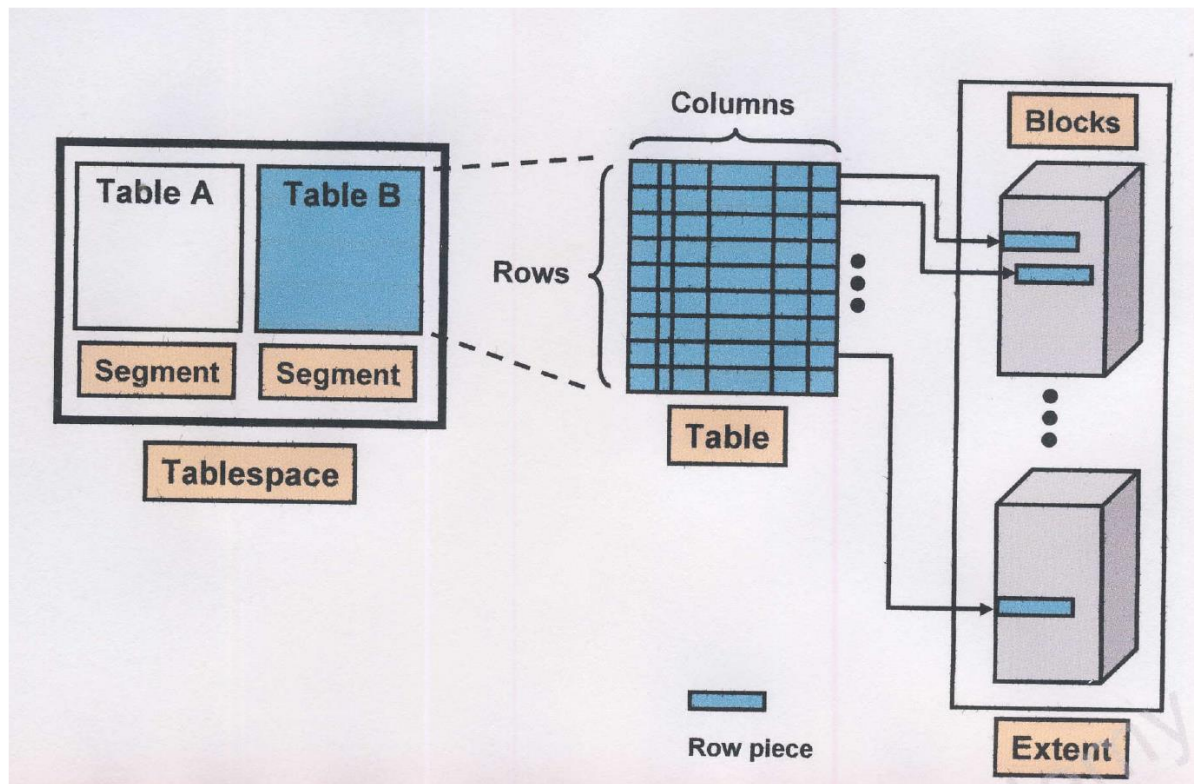
Database Behavior	ABORT	IMMEDIATE	TRANSACTIONAL	NORMAL
Nouvelles connections	No	No	No	No
Attendre les fins de sessions	No	No	No	Yes
Attendre les fins de transactions	No	No	Yes	Yes
Performs a checkpoint and closes open files	No	Yes	Yes	Yes

Atelier 3

- ▶ Shutdown de la base
 - ▶ Démarrer la base en mode nomount
 - ▶ Passer la base en Mount
 - ▶ Passer la base de Mount en Open
-
- ▶ Qualifier la nouvelle connexion de la base avec une requête d'une table ou vue système

Gestion de la structure de la base de données

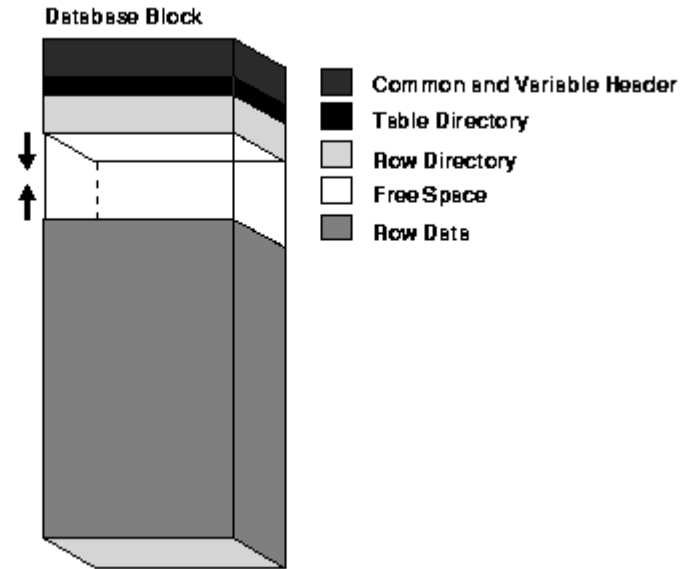




Comment sont stockées les données d'une table

- Un segment est créé à la création d'une table et inséré dans un tablespace
- Logiquement une table est constituée de lignes et de colonnes. Une ligne est stockée et rangée dans un bloc de données.

Contenu d'un bloc de données



Header

Il contient l'information générale : bloc de données, bloc d'index, rollback...

Table Directory : Informations sur la table

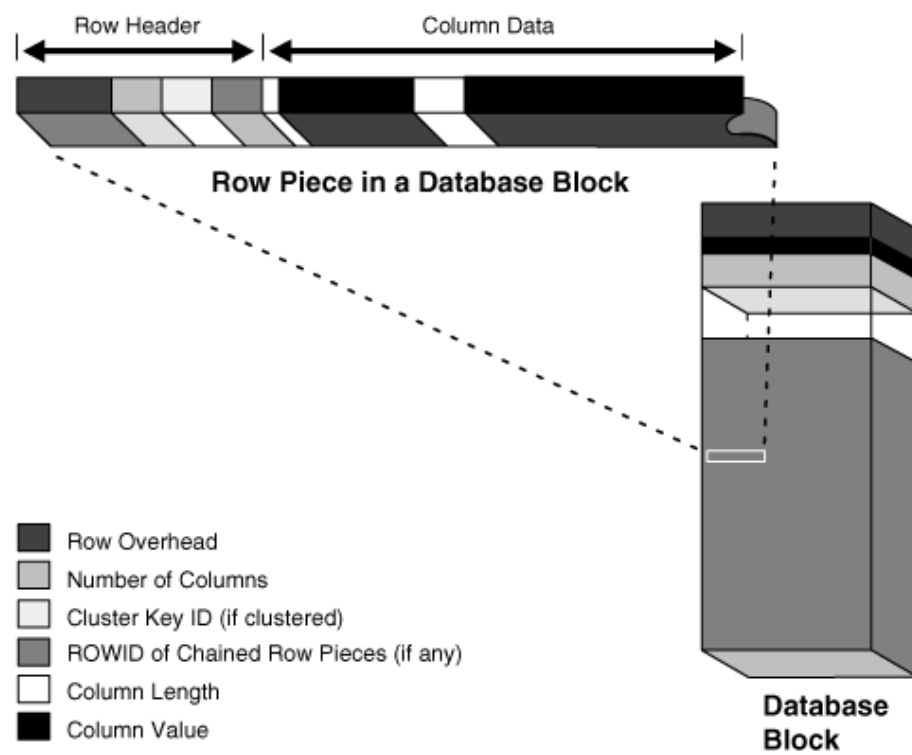
Row Directory : Cette partie du bloc de données contient des informations sur les lignes réelles dans le bloc

Free Space

Espace pour l'insertion de nouvelles données

Row Data

Contenu des données des tables ou index

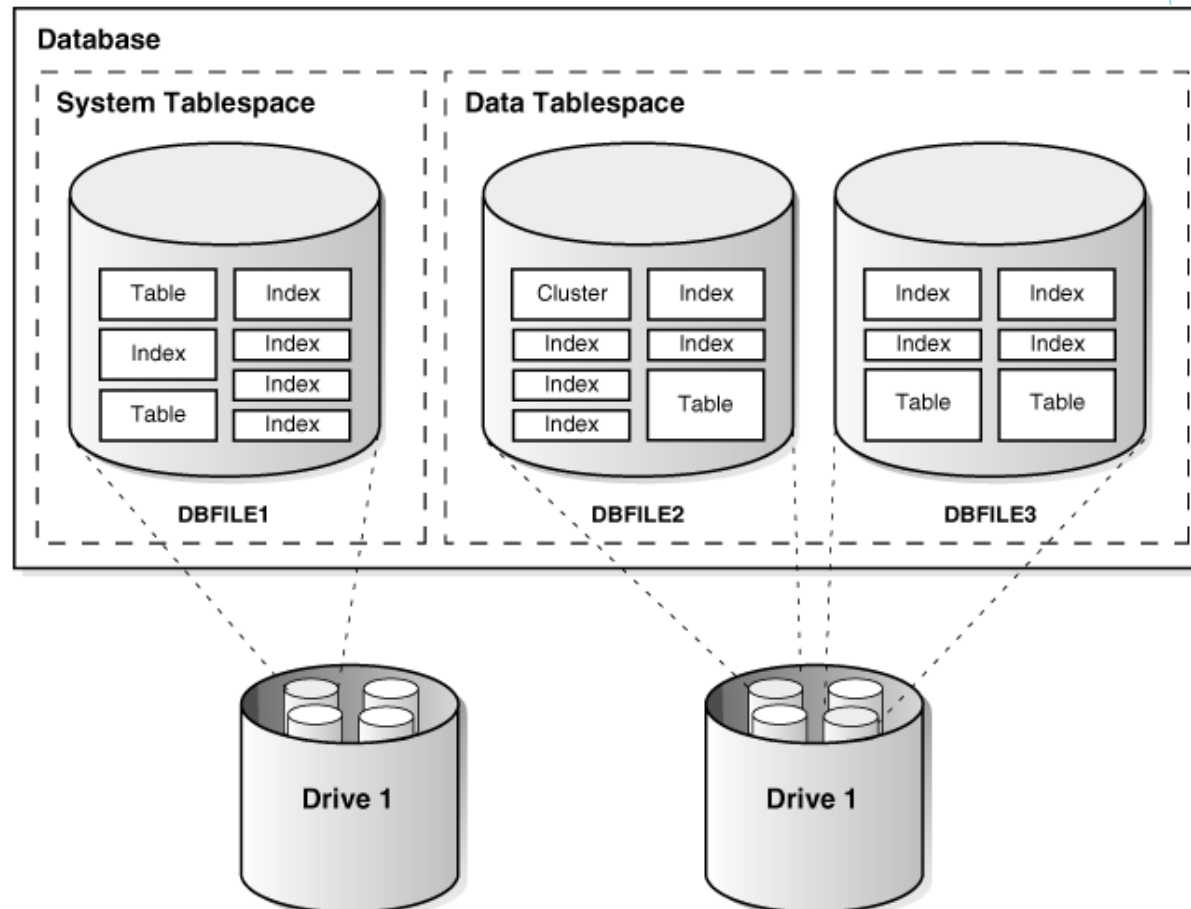


Les espaces logiques Tablespaces et les fichiers physiques de données

Un tablespace est une unité logique de stockage composée d'un ou plusieurs fichiers physiques.

Il y a au minimum deux tablespaces : 1 tablespace system et sysaux

- SYSAUX.DBF
- SYSTEM.DBF



Création d'un nouveau tablespace

Avant de créer le tablespace, vous devez créer un dossier et choisir l'emplacement physique de stockage

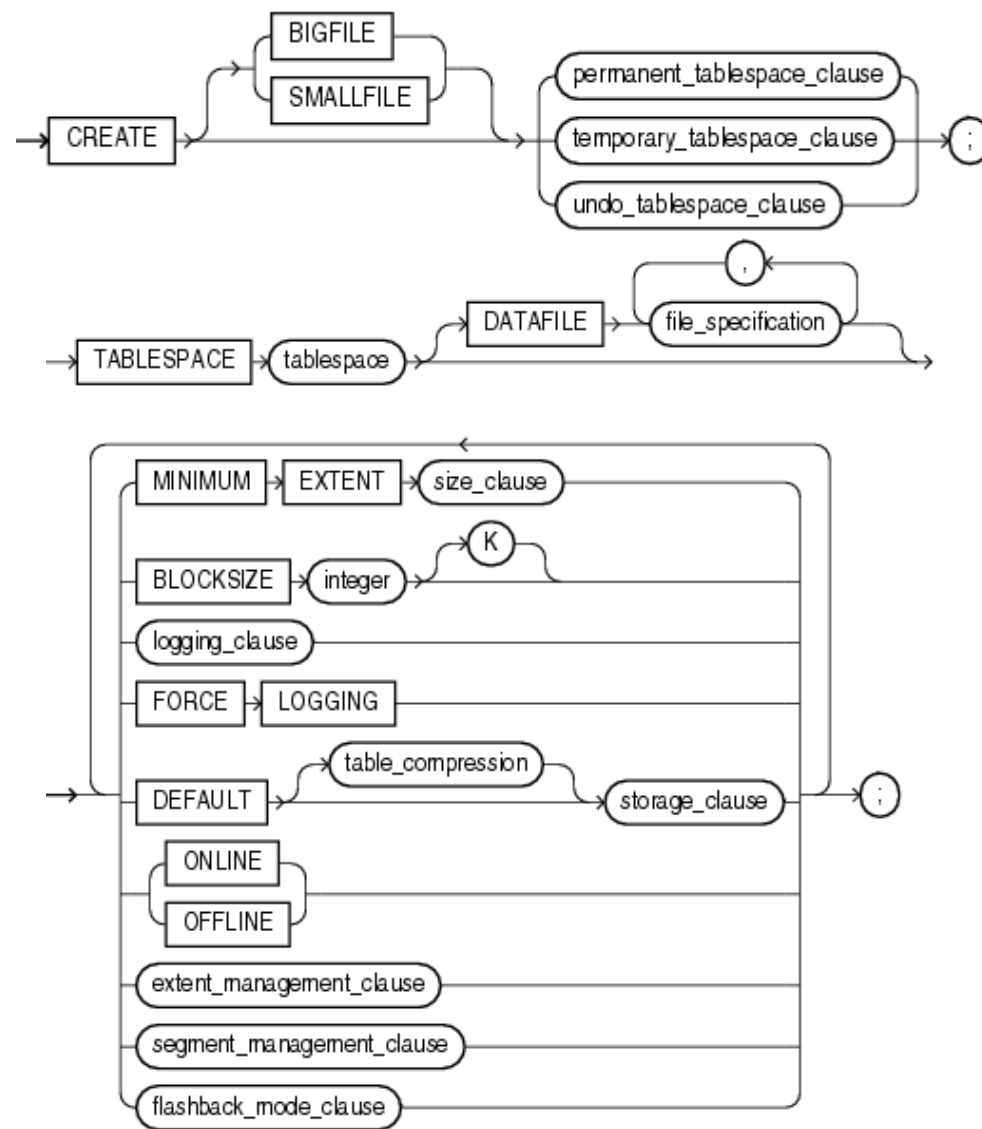
Commande

- ▶ CREATE TABLESPACE
- ▶ Exemple :
- ▶

```
CREATE TABLESPACE lmtbsb DATAFILE  
'/ora01/oracle/app/oracle/oradata/orcl/lmtbsb01.dbf' SIZE 50M  
AUTOEXTENT ON;
```

Informations relatives aux tablespaces et au data files

- ▶ Tablespaces : DBA_TABLESPACES, V\$TABLESPACE
- ▶ Data files : DBA_DATA_FILES, V\$DATAFILE



Atelier 4

- ▶ Création de deux tablespaces
- ▶ 1 tablespace pour les index tbs_index
- ▶ 1 tablespace pour les utilisateurs tbs_user

Modifier une tablespace

ALTER tablespace

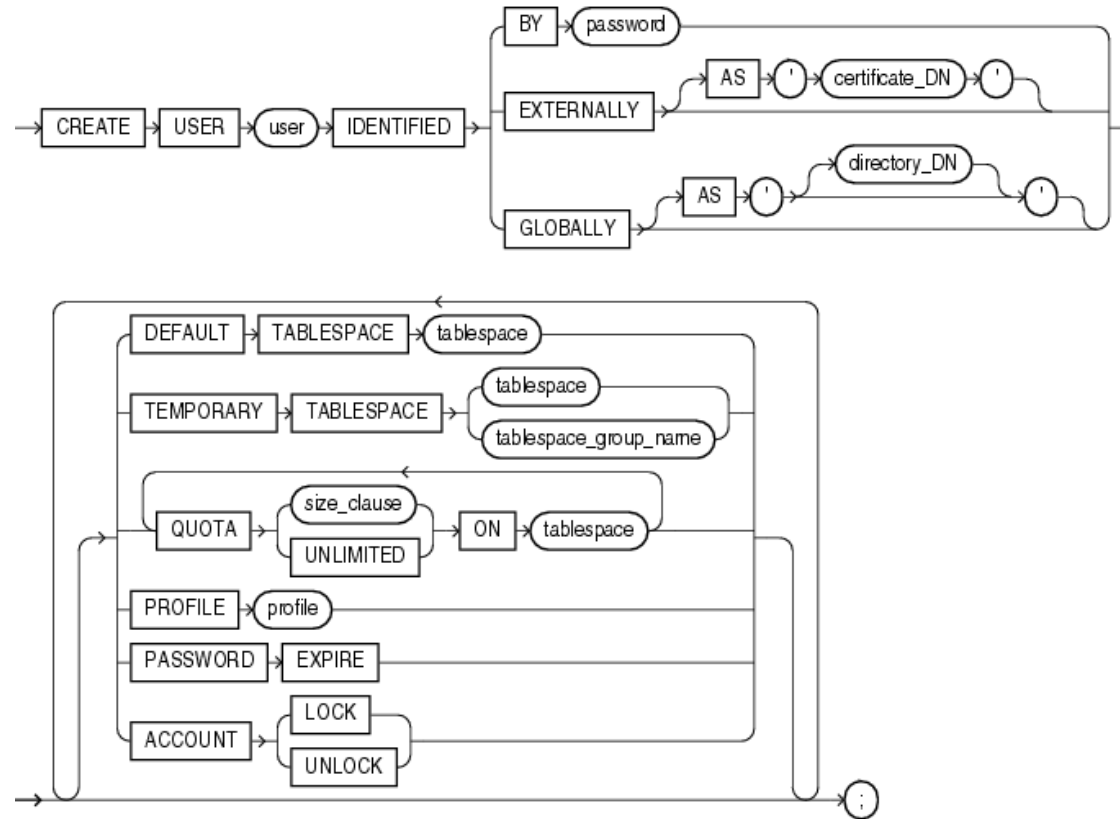
- ▶ Passer le tablespace en OFFLINE
- ▶ Modifier le TABLESPACE , le dossier de stockage
- ▶ Passer le tablespace en lecture ou lecture/écriture
- ▶ Supprimer les tablespaces créés au dessus

Gestion des utilisateurs et de leurs droits

Oracle permet de définir :

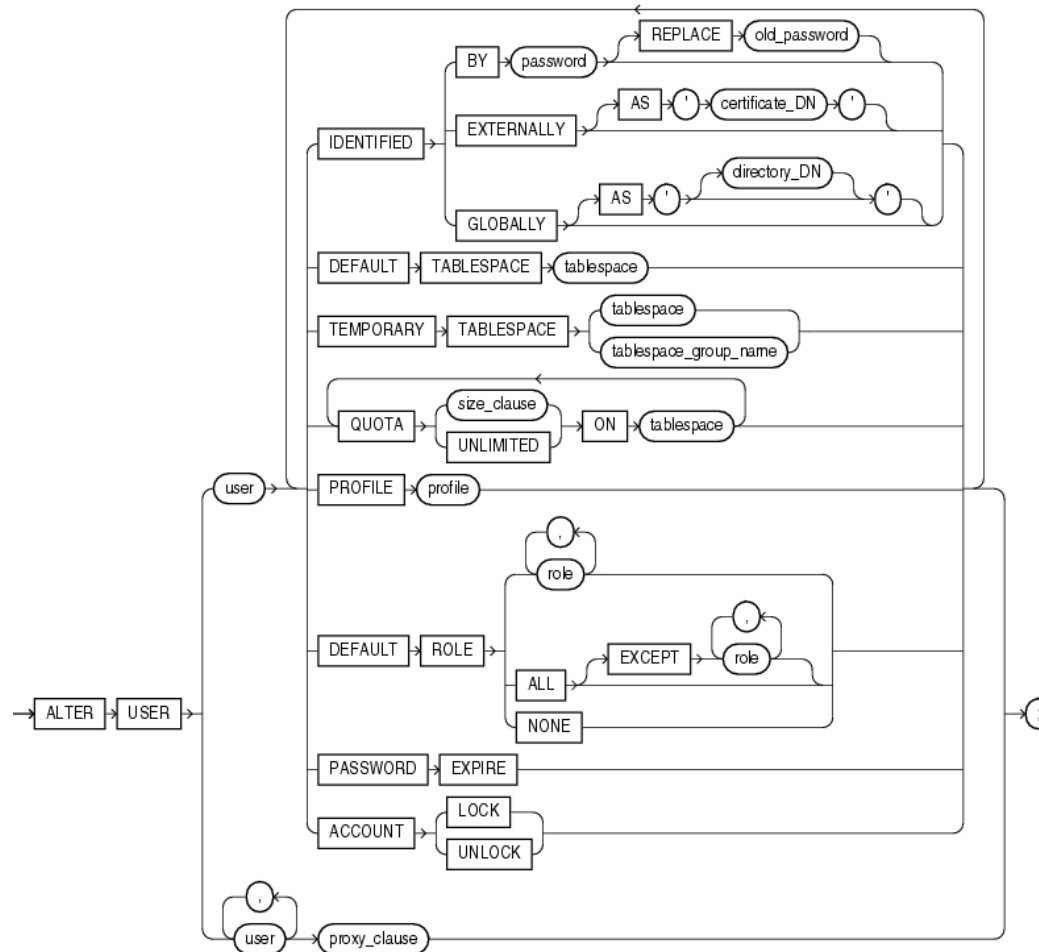
- ▶ Des utilisateurs pouvant se connecter à la base
- ▶ D'affecter des droits à des utilisateurs
- ▶ Des droits sur les objets et le système
- ▶ Oracle parle de privilèges pour les droits des utilisateurs
- ▶ Un rôle est un regroupement de privilèges

Commande : Création d'un utilisateur



CREATE USER metier
IDENTIFIED BY metier_esaip
DEFAULT TABLESPACE tbs_users
QUOTA 10M ON tbs_users
TEMPORARY TABLESPACE temp
PASSWORD EXPIRE;

Modification d'un utilisateur



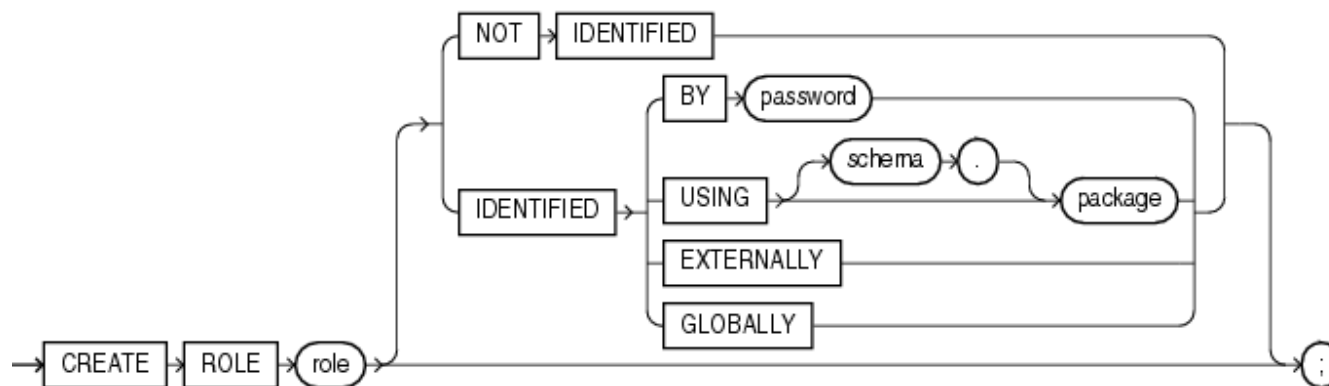
ALTER USER metier
IDENTIFIED BY new_passwd
DEFAULT TABLESPACE tbs_users2;

Atelier 5

- ▶ Création d'un user
- ▶ Modification du mot de passe
- ▶ Bloquer un compte
- ▶ Débloquer un compte
- ▶ Visualiser les paramètres
- ▶ Attribution d'un quota
- ▶ Changement de tablespace

Création de rôles

Regroupement d'utilisateurs



```
CREATE ROLE dba_exploit  
IDENTIFIED BY dba;
```

Les privilèges

Il y a deux sortes de privilèges :

- ▶ Droits sur les outils systèmes : system privileges
- ▶ Droits sur les objets de la base : objets privileges

Privilèges système (DDL)

- ▶ Create role : affecte un rôle à un utilisateur
- ▶ Create user : droit de créer un utilisateur

Table SYTEM_PRIVILEGE_MAP pour voir les types de privilège système

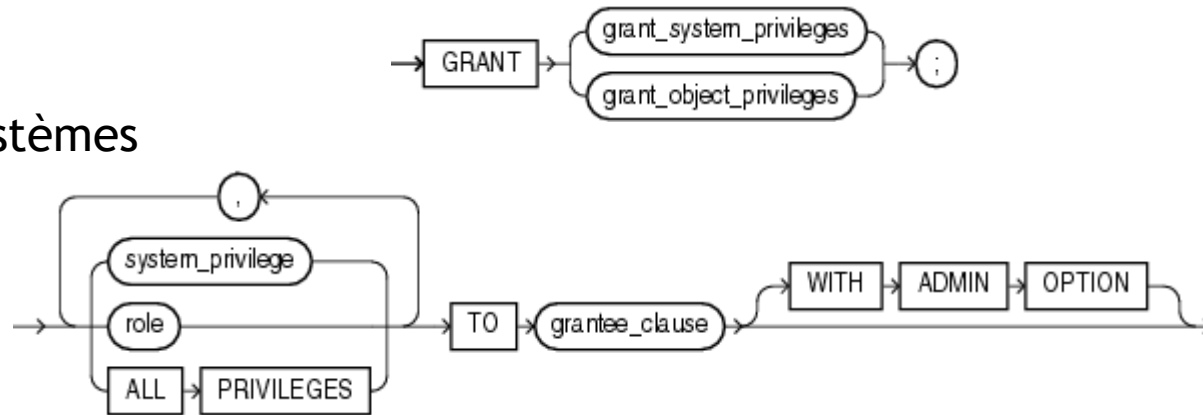
Privilèges objets (DML)

- ▶ Insert
- ▶ Delete
- ▶ Update

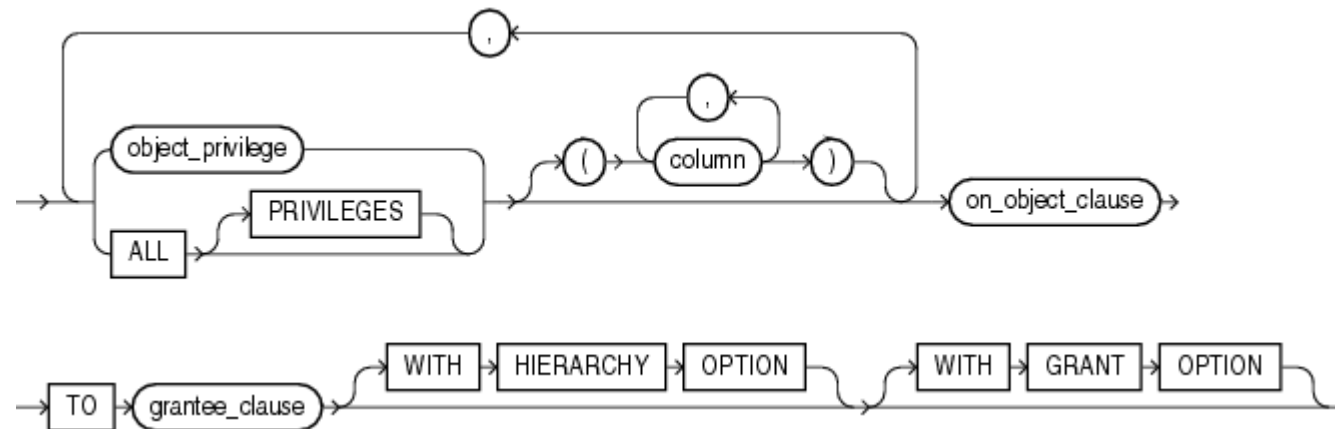
Table TABLE_PRIVILEGE_MAP pour voir les types de privilège objets

Affecter des privilèges

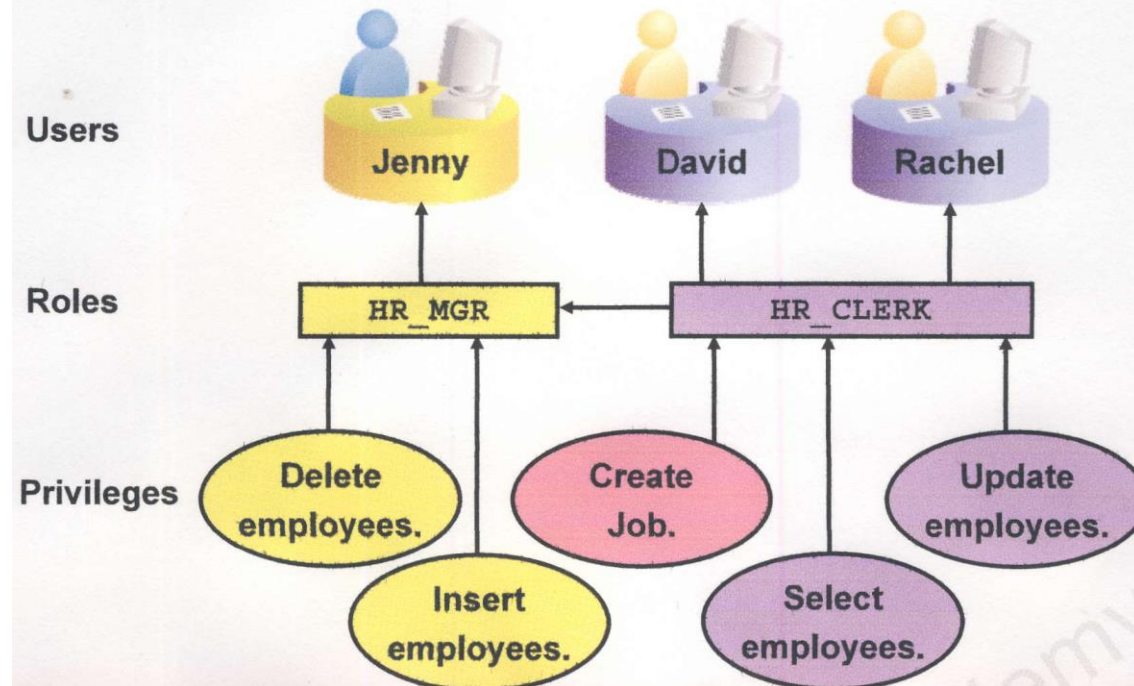
Systèmes



Objets



Assigning Privileges to Roles and Assigning Roles to Users



Atelier 6

- ▶ Création de quatre utilisateurs
- ▶ Création de deux rôles
 - ▶ Rôle_metier 1 (2 users) : delete et insert sur la table emp du schéma Scott
 - ▶ Rôle_metier 2 (2 users) : update, select sur la table emp du schema Scott
- ▶ Création d'un rôle dba_exploit
 - ▶ Privilèges système : création, modifier, supprimer des utilisateurs, donner des privilèges
- ▶ Création d'un rôle dba_exp : export et import des bases
 - ▶ Vérifier dans les tables et vues système et envoyer les résultats dans un fichier Texte
 - ▶ Faire une requête afin de superviser les utilisateurs connectés

Gestion des profils

Un ensemble qui limite les ressources attribuées à un utilisateur

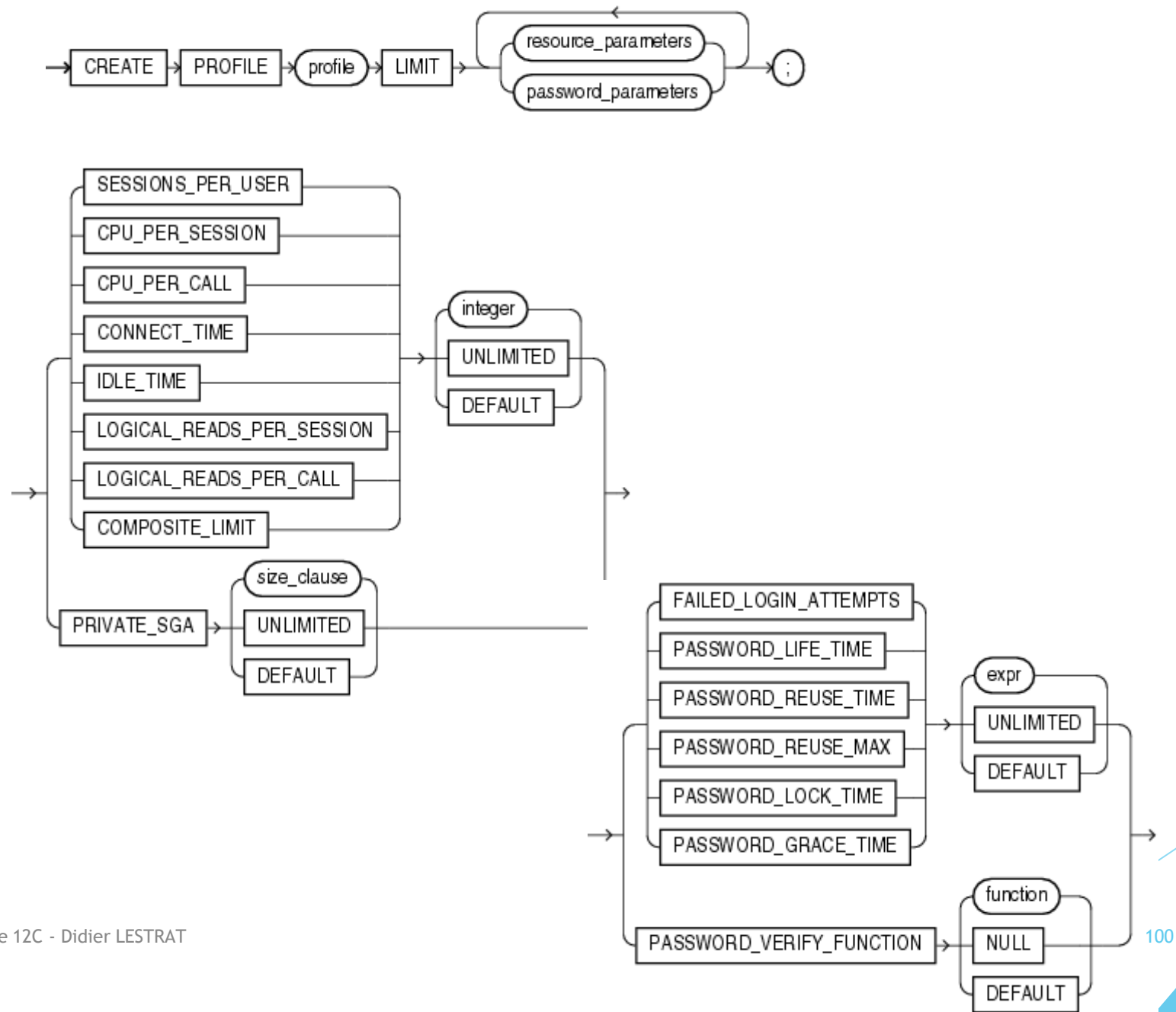
- ▶ Temps Cpu par appel ou par session
- ▶ Nbre de lecture logique
- ▶ Nbre d'ouverture de session limité pour un user
- ▶ Temps d'inactivité par session

Depuis la version 8 : mise en œuvre d'une politique des mots de passe

- ▶ Verrouillage des comptes
- ▶ Durée de vie des mots de passe
- ▶ Complexité du mot de passe

Ressource_parameters

- ▶ SESSIONS_PER_USER Specify the number of concurrent sessions to which you want to limit the user.
- ▶ CPU_PER_SESSION Specify the CPU time limit for a session, expressed in hundredth of seconds.
- ▶ CPU_PER_CALL Specify the CPU time limit for a call (a parse, execute, or fetch), expressed in hundredths of seconds.
- ▶ CONNECT_TIME Specify the total elapsed time limit for a session, expressed in minutes.
- ▶ IDLE_TIME Specify the permitted periods of continuous inactive time during a session, expressed in minutes. Long-running queries and other operations are not subject to this limit.
- ▶ LOGICAL_READS_PER_SESSION Specify the permitted number of data blocks read in a session, including blocks read from memory and disk.
- ▶ LOGICAL_READS_PER_CALL Specify the permitted number of data blocks read for a call to process a SQL statement (a parse, execute, or fetch).
- ▶ PRIVATE_SGA Specify the amount of private space a session can allocate in the shared pool of the system global area (SGA). Please refer to [size_clause](#) for information on that clause.



Atelier 7

- ▶ Création de 6 utilisateurs métiers, 3 rôles et des privilèges pour des rôles
- ▶ Création de trois rôles
 - ▶ Role 1 nommé admin_metier :
 - ▶ Create table, create user, lancer des procédures, create index
 - ▶ Role nommé user_metier1 :
 - ▶ Update les colonnes first_name et last_name de la table emp de Scott
 - ▶ Select sur la table emp de Scott
 - ▶ Role nommé user_metier2 :
 - ▶ Insert sur les colonnes phone_number et salary de la table emp de Scott
 - ▶ Select sur la table emp de scott
- ▶ Création de profil pour les utilisateurs métiers
 - ▶ Changement du mot de passe : Pas de réutilisation du même mot de passe avant 60 jours
 - ▶ Possibilité de réutiliser le même mot de passe après 5 nouveaux mots de passe
 - ▶ Délai max de connection 30 mn
 - ▶ Session simultanée 20
 - ▶ Espace dédié pour une session : 30K
- ▶ **Vérifier dans les tables et vues système et envoyer les résultats dans un fichier Texte**

Privilèges système

Table DBA_SYS_PRIVS

- ▶ GRANTEE non de l'utilisateur ou du rôle qui a reçu le privilège système
- ▶ PRIVILEGE privilège système reçu
- ▶ ADMIN_OPTION : privilège reçu avec la clause WITH ADMIN OPTION, notion d'héritage

Table SESSION_PRIVS

PRIVILEGE Nom du Privilège

Table SYTEM_PRIVILEGE_MAP

NAME Nom des Privilèges

Privilège objets

Table DBA_TAB_PRIVS

- ▶ GRANTEE Nom user qui a reçu le privilège
- ▶ OWNER nom du propriétaire de l'objet
- ▶ TABLE_NAME Nom de l'objet
- ▶ GRANTOR Nom de l'utilisateur qui a attribué le privilège
- ▶ PRIVILEGE privilège objet reçu
- ▶ GRANTABLE : privilège reçu avec la clause WITH ADMIN OPTION

DBA_COL_PRIVS

- ▶ GRANTEE Nom user qui a reçu le privilège
- ▶ OWNER nom du propriétaire de l'objet
- ▶ TABLE_NAME Nom de l'objet
- ▶ GRANTOR Nom de l'utilisateur qui a attribué le privilège
- ▶ PRIVILEGE privilège objet reçu
- ▶ GRANTABLE : privilège reçu avec la clause WITH ADMIN OPTION

Table TABLE_PRIVILEGE_MAP

NAME Nom du privilège

Un schéma

- ▶ Un schéma est un ensemble d'objets appartenant à un seul utilisateur

Exemple schéma sys, schéma system, schema Scott

D'autres schémas sont installés pendant l'installation d'Oracle :

- ▶ SCOTT : débloquent l'utilisateur SCOTT
- ▶ BI
- ▶ OE
- ▶ IX
- ▶ PM
- ▶ SH

Pour visualiser les schémas et les objets des schémas :
voir OEM

Les types de données dans les tables

TYPE	VALEURS
BINARY-INTEGER	entiers allant de -2^{31} à 2^{31})
POSITIVE / NATURAL	entiers positifs allant jusqu'à $2^{31} - 1$
NUMBER	Numérique (entre -2^{418} à 2^{418})
INTEGER	Entier stocké en binaire (entre -2^{126} à 2^{126})
CHAR (n)	Chaîne fixe de 1 à 32767 caractères (différent pour une colonne de table)
VARCHAR2 (n)	Chaîne variable (1 à 32767 caractères)
Chaîne variable (1 à 32767 caractères)	(maximum 2 gigaoctets)
DATE	Date (ex. 01/01/1996 ou 01-01-1996 ou 01-JAN-96 ...)
CLOB	Grand objet caractère. Objets de type long stockés en binaire (maximum 4 giga octets)
BLOB	Grand objet binaire. Objets de type long (maximum 4 giga octets)
NCLOB	Support en langage nationale (NLS) des grands objets caractères. Déclare une variable gérant un pointeur sur un grand bloc de caractères utilisant un jeu de caractères mono-octets, multi-octets de longueur fixe ou encore multi-octets de longueur variable et stocké en base de données

Les types de données dans les tables

ROWID	Composé de 6 octets binaires permettre d'identifier une ligne par son adresse physique dans la base de données.
UROWID	Le U de UROWID signifie Universel, une variable de ce type peut contenir n'importe quel type de ROWID de n'importe quel type de table
FLOAT(p)	Nombre flottant
BINARY_FLOAT	Nombre flottant sur 32 bytes
BINARY_DOUBLE	Nombre flottant sur 64 bytes
BFILE	Stocker vers un fichier en dehors de la base
TIMESTAMP	Etendre le type date en conservant les secondes – par défaut la précision est sur 6 chiffres et peut être étendue à 9 chiffres

Création et modification de tables

Via la commande

```
▶ CREATE TABLE scott.classes  
(  
  classe_id NUMBER(5),  
  name VARCHAR2(15) NOT NULL,  
  nombre_etud NUMBER(9) ENCRYPT,  
  Spe_classe VARCHAR2(30),  
  DATE DEFAULT ,  
  Photo_classe BLOB )  
TABLESPACE tbs_tables STORAGE ( INITIAL 50K);  
  
ALTER TABLE scott.classes ADD annee NUMBER (6);
```

Atelier PRACTICE 8

Gestion des tables et des index

Sur tous les objets d'une base, ce sont les tables et les index qui occupent de la place

Les tables et les index sont des segments et le stockage est géré par des extensions et par les caractéristiques des tablespaces

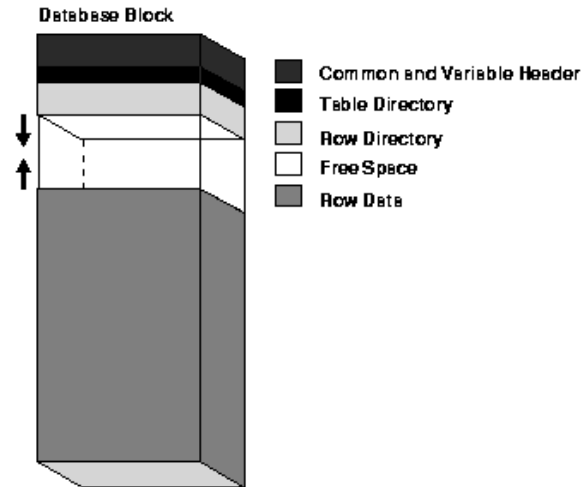
L'organisation du stockage dans les blocs est important

Écritures des lignes dans les blocs

Index : Il existe plusieurs types d'index

- ▶ Index B-TREE
- ▶ Index BITMAP
- ▶ Index à clé inversée
- ▶ Index basé sur les fonctions

Organisation du stockage dans les blocs de données



Header

Il contient l'information générale : bloc de données, bloc d'index, rollback...

Table Directory : Informations sur la table

Row Directory : Cette partie du bloc de données contient des informations sur les lignes réelles dans le bloc

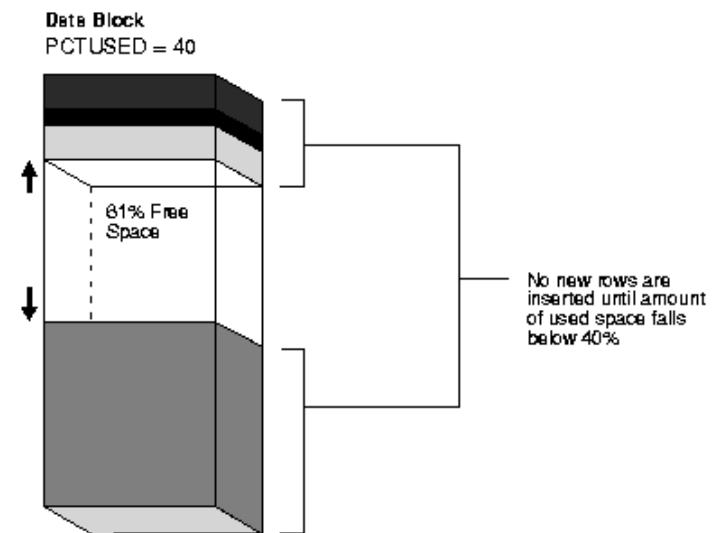
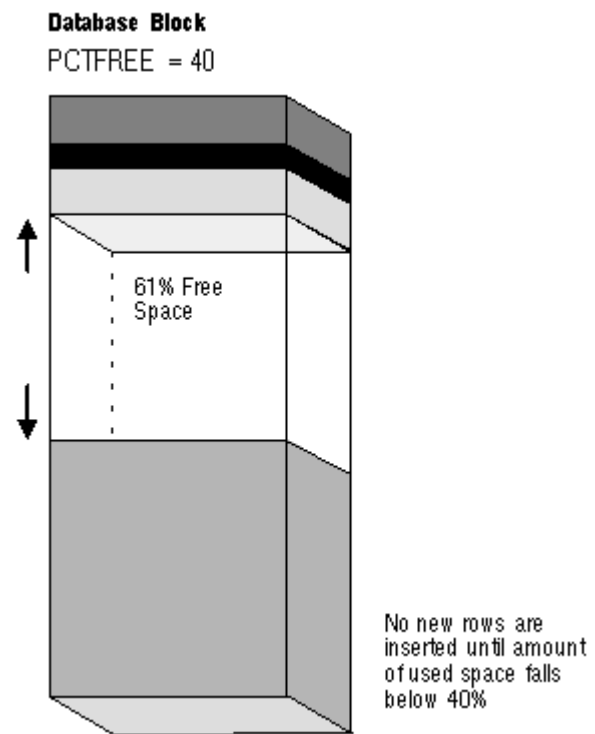
Free Space

Espace pour l'insertion de nouvelles données

Row Data

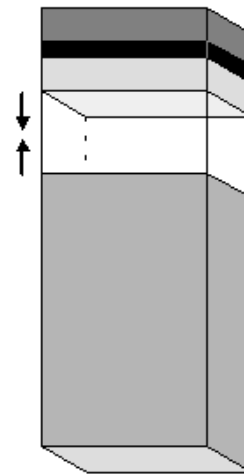
Contenu des données des tables ou index

HWM High Water Mark

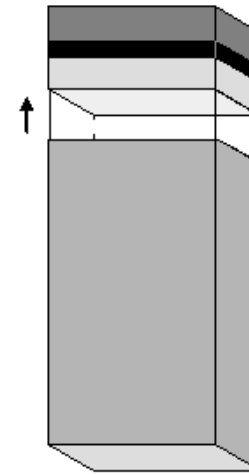


Database Block

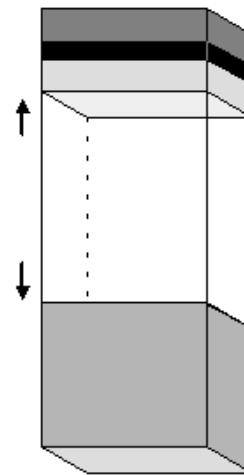
PCTFREE = 20, PCTUSED = 40



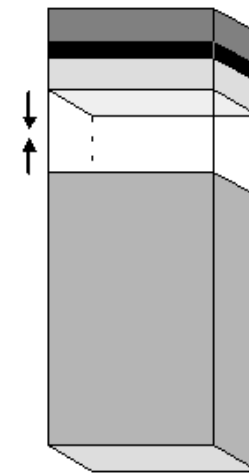
- 1** Rows are inserted up to 80% only, since PCTFREE says that 20% of the block must remain open for updates of existing rows.



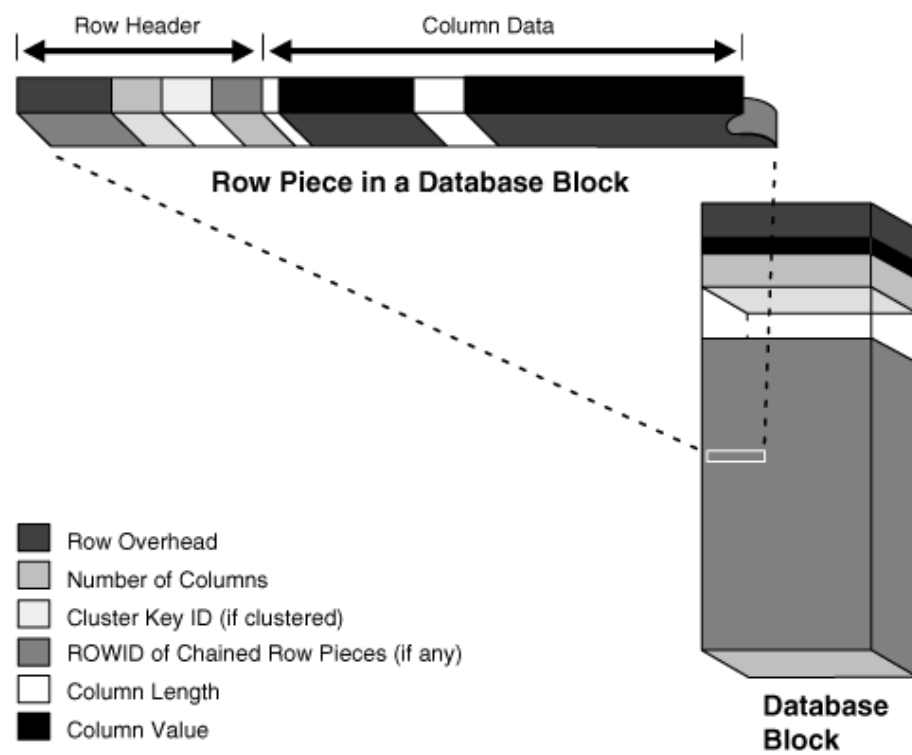
- 2** Updates to existing rows use the free space reserved in the block. No new rows can be inserted into the block until the amount of used space is 39% or less.



- 3** After the amount of used space falls below 40% new rows can again be inserted into this block.



- 4** Rows are inserted up to 80% only, since PCTFREE says that 20% of the block must remain open for updates of existing rows. This cycle continues ...



Structure d'un bloc

Header : adresse du bloc, le type de segment, un répertoire des tables, un répertoire des lignes

La taille est variable : de 100 à 200 octets

Structure d'une ligne : l'en-tête d'une ligne contient des informations sur la ligne et les données insérées

Taille variable 3 octets minimum

Chaque colonne est stockées avec un en-tête de colonne : 1 à 3 octets pour la longueur de la colonne et suivi de la valeur de la colonne

Gestion de l'espace dans les blocs

L'espace libre peut gérer soit automatiquement ou manuellement

Gestion manuelle avec deux paramètres PCTFREE et PCTUSED

PCTFREE : spécifie le pourcentage de l'espace libre pour les modifications de lignes stockées dans les blocs.

Il permet de ne pas remplir le bloc à 100% et de conserver de l'espace

PCTUSED : spécifie le pourcentage d'occupation dans le bloc

En gestion automatique Oracle utilise un BITMAP pour gérer ces 2 paramètres

- ▶ 0 à 25% d'espace libre
- ▶ 25 à 50% d'espace libre
- ▶ 50 à 75% d'espace libre

Gestion des tables

Le **ROWID** est une colonne virtuelle qui est présente dans chaque table

- ▶ Donne la localisation de l'adresse physique de stockage de la ligne

On peut requêter sur cet valeur

- ▶ SQL> select ROWID
- ▶ Ou sql> UPDATE Where ROWID = 'abc'
- ▶ Augmente la performance dans une application dans les clauses WHERE

Chaînage et migration

Une ligne complète est stockée en totalité dans un bloc

Si la ligne est trop grande, Oracle l'insère dans d'autres blocs

- ▶ Chaînage d'une ligne
- ▶ Lit plusieurs blocs
- ▶ Pointeur entre bloc
- ▶ Perte en performance

Spécifier le stockage d'une table

- ▶ Spécifier lors de la création d'une table
- ▶ Ordre SQL

SQL>CREATE TABLE

STORAGE INITIALE VALEUR

MAXEXTENTS

PCTFREE

- ▶ La clause PCTFREE par défaut : 10 par défaut
- ▶ La clause PCTUSED par défaut : 40 par défaut

La somme des deux doit strictement inférieure à 100

- ▶ La clause COMPRESS : compresse les données dans les blocs (DIRECT_LOAD), par défaut ALL
- ▶ La clause LOGGING : permet de ne pas journaliser certaines opérations

Atelier 8

```
SQL>CREATE TABLE scott.formation  
(ID_ent  NUMBER(6),  
Nom      VARCHAR2(40),  
Prenom VARCHAR2(40),  
Demande_form Varchar2(60))  
TABLESPACE tbs_form  
STORAGE (INITIAL 1M)  
PCTFREE 30  
PCTUSED 40;
```

Recommandations pour le stockage d'une table

- ▶ Stocker les tables dans un ou plusieurs tablespaces dédiées
- ▶ Utiliser des tablespaces gérées localement avec une gestion automatique des extensions (EXTENT MANAGEMENT LOCAL AUTOALLOCATE)
- ▶ Régler correctement le PCTFREE
- ▶ Allouer un espace initial à la table en prévoyant son évolution sur 1,2 ou 3 ans

Suivant les critères : Table statique, petite, volumineuse, à croissance régulière

Estimation de la volumétrie d'une table

- ▶ Estimer le nombre de lignes attendues : +/- 10%
- ▶ Créer la table en qualification (copie de la production)
- ▶ Charger un jeu de données
- ▶ Calculer le nombre de blocs utilisés (package DBMS_SPACE)

Sql > exec dbms_stats.gather_table_stats ('schéma', 'nom de la table')

Sql>select blocks FROM DBA_TABLES where TABLE-NAME='nom de la table'

150 BLOCKS

/* estimation règle de trois*/

/* insertion de 10 000 lignes, 10 000 000 lignes dans 3 ans

SQL>select 150*10 000 000 /10 000 estimation FROM DUAL;

Estimation de PCTFREE

- ▶ Valeur optimale
- ▶ $PCTFREE = 100 * (1 - Ti / Tr)$
- ▶ Ti taille moyenne initiale d'une ligne (à l'insertion)
- ▶ Tr taille moyenne finale d'une ligne (les mises à jour)
- ▶ Statistique à visualiser dans la table DBA_TABLES la colonne AVG_ROW_LEN

Surveillance des tables

- ▶ Analyser l'activité d'un ou plusieurs tables
- ▶ Le mécanisme de surveillance est installé par défaut
- ▶ Les informations des statistiques se trouvent dans la table
DBA_TAB_MODIFICATIONS

Nota : les statistiques ne sont pas versées au fil de l'eau dans le dictionnaire, voir la colonne TIMESTAMP

Pour forcer l'inscription du dictionnaire, on utilise la procédure
FLUSH_DATABASE_MONITORING_INFO du paquet DBMS_STATS

- ▶ Faire une analyse périodique des statistiques
- ▶ Le package DBMS_STATS propose plusieurs procédures pour superviser le stockage des tables
- ▶ Ex la procédure gather_table_stats permet de calculer les statistiques d'une table

Problèmes

- ▶ Espace inutilisé alloué à une table
BLOCKS dans la table DBA_TABLES
Lancer le package DBMS_SPACE
- ▶ Faible taux d'occupation moyen des blocs
- ▶ Lancer la procédure SPACE_USAGE du package DBMS_SPACE

Réorganiser le stockage d'une table

- ▶ Libérer de l'espace dans les blocs
- ▶ Améliorer le taux de remplissage des blocs
- ▶ Corriger un souci de migration de données
- ▶ Réorganiser le stockage de la table, changement de tablespace, modification du PCTFREE

Informations sur les tables

- ▶ DBA_TABLES
- ▶ DBA_TAB_COLUMNS
- ▶ DBA_SEGMENTS
- ▶ DBA_EXTENTS
- ▶ DBA_TAB_MODIFICATIONS

Réorganiser le stockage d'une table

Plusieurs techniques

- ▶ **Ordre SQL ALTER TABLE....DEALLOCATE UNUSED**

Libérer de l'espace au dessus de HWM

- ▶ **Recréer la table ou des lignes de tables**

Object dépendants supprimés (TRIGGER, CONTRAINTES, INDEX,...)

Une variante à rajouter TRUNCATE afin de libérer de l'espace sous HWM

- ▶ **Export/Import**

- ▶ **Ordre SQL ALTER TABLE....SHRINK SPACE**

Compacter les lignes d'une table (compacte le segment aussi)

- ▶ **Ordre SQL ALTER TABLE.... MOVE**

Réorganisation complète du stockage physique de la table sans la supprimer

Atelier 9

- ▶ Statistique sur une table

exec dbms_stats.gather_table_stats ('schéma', 'nom de la table')

Extraction des valeurs de la table DBA_TABLES

NUM_ROWS : nombre de ligne

BLOCKS : nombre de blocks en dessous de HWM

AVG_ROW_LEN : longueur moyenne d'une ligne

SAMPLE_SIZE : nombre de lignes utilisées dans l'échantillon

LAST_ANALYZED ; date de l'analyse

- ▶ Espace inutilisé alloué à une table

- ▶ Select t.blocks « occupés »,s.blocks »alloués » from
dba_tables t, dba_segments s where
s.segment_name=t.table_name and s.owner=t.owner and
t.table_name='emp' and t.owner='scott';

Ok : le mieux adapté
X : possible

	DEALLOCATE	Recréer	Export/import	SHRINK	MOVE
Libérer espace HWM	OK	X	X	X	X
Améliorer le taux de remplissage		X	X	OK	OK
Corriger migration de données		OK	X		OK
Réorganisation globale		X	X		OK

ATELIER 9

1 - Analyse des tables avec le package et la procédure suivante
`DBMS_STATS.GATHER_TABLE_STATS`

- ▶ Découvrir ce package
- ▶ Lancer la procédure
- ▶ Visualiser les résultats : `num_rows`, `blocks`, `avg_row_len`, `sample_size`
 - ▶ Nombre de lignes dans la table
 - ▶ Nombre de blocs en dessous HWM
 - ▶ Longueur moyenne d'une ligne
 - ▶ Nombre de ligne dans l'échantillon
 - ▶ Date de la dernière analyse
- ▶ 2 - Réorganisation complète du stockage physique de la table sans la supprimer

Avec l'Ordre SQL `ALTER TABLE.... MOVE`

Espace d'annulation

- ▶ Oracle stocke temporairement les données en cours de modifications dans des segments d'annulations.
- ▶ Ces segments d'annulations sont en attente de validation ou d'annulation (COMMIT ou ROLLBACK).
- ▶ Utile pour la notion de lecture cohérente des données pendant des opérations de mises à jour, pour la récupération de données (FLASHBACK) et le RECOVER.

La gestion des segments d'annulations est proposée en automatique pour ce type de Tablespace UNDO (conseillé par Oracle) et anciennement appelés : ROLLBACK SEGMENT.

Ce Tablespace est créé au moment de la création de la base de données ou après et est obligatoirement géré localement.

```
SQL> SELECT name, value, description FROM V$PARAMETER  
WHERE name LIKE '%undo%';
```


Espace d'annulation

- ▶ Structure : un segment d'annulation est stocké dans un tablespace dédié
- ▶ Les segments d'annulation correspondent à une structure physique (segment)
- ▶ Si l'instance a besoin d'espace dans le DATA BUFFER CACHE, les blocs de segments d'annulation peuvent être écrites sur le disque
- ▶ Les modifications apportées dans les segments d'annulations sont inscrits dans le REDO LOG.
- ▶ Segment d'annulation System
 - ▶ Créé lors de la création de la base
 - ▶ Stocké dans la tablespace SYSTEM

Principe

- ▶ Attribution d'un segment affectée au démarrage d'une transaction

Trois états :

- ▶ Actif
- ▶ Non expiré
- ▶ Expiré

Mise en œuvre de la gestion automatique

- ▶ Mettre le paramètre UNDO_MANAGEMENT à AUTO
- ▶ Créer une tablespace d'annulation lors de la définition de la base de données
- ▶ Gestion de la tablespace d'annulation
 - ▶ Gérer localement
 - ▶ Gestion automatique des extensions
 - ▶ Toujours en READ WRITE, LOGGING, PERMANENT

Atelier 11 : Réaliser 3 scripts

création, modification, suppression d'une tablespace d'annulation

Rôle du DBA

- ▶ Anticiper les destructions des bases
- ▶ Diminuer la moyenne MTBF : Echeecs
- ▶ Diminuer la moyenne MTTR : reconstruction
- ▶ Minimiser la perte des données

Les causes des défaillances des bases

- ▶ Processus utilisateur : session, connexion
- ▶ Réseaux défectueux : listener, connexion, carte réseau
- ▶ Erreur des utilisateurs : modifications de données, drop tables
- ▶ Echeec de l'instance : Hardware, processus
- ▶ Défaillance des supports : média disques...

Phases de la reconstruction de l'instance

- ▶ Synchronisation des fichiers de données
- ▶ Roll Forward (redo log)
- ▶ Roll Back (undo)
- ▶ Les données sont 'comittées' ou non 'comittées'
- ▶ La base était ouverte ou pas

Pendant la reconstruction de l'instance, les transactions entre le dernier checkpoint et la fin du redo log doit être validé dans le fichier de données.

Les dernières transactions sont écrites dans les redo log. Tous les 3 secondes le processus checkpoint surveille dans le fichier de contrôle la position du redo log (le dernier SCN).

Echec de l'Instance

- ▶ utiliser shutdown abort et startup force
- ▶ Visualiser les traces

Comprendre l'instance de reconstruction

- ▶ CKPT :
- ▶ Redolog et logWriter

Configurer correctement l'espace de reconstruction

- ▶ Copie des fichiers de contrôle sur des disques et contrôleur différents
(voir commande ajout d'un fichier de contrôle)
- ▶ Copie des fichiers redolog sur des disques et contrôleur différents
- ▶ Plannifier régulièrement les sauvegardes
- ▶ Archiver les fichiers Log (Database en mode archivelog)
 1. Shutdown de la base
 2. Startup mount
 3. Alter database archivelog;
 4. Alter database open;

Sauvegarde et récupération

Principes

- ▶ Assurer la sécurité des données est une des tâches principales de l'administrateur.
- ▶ Mise en œuvre d'une protection des fichiers sensibles de la base :
 - ▶ fichiers de contrôle
 - ▶ fichiers de journalisation.
 - ▶ Fichiers de paramètres
 - ▶ Fichiers de données

Adaptée aux contraintes de l'entreprise : testée et documentée.

- ▶ Cohérente ou incohérente

Les questions à se poser

- ▶ Est-il acceptable de perdre des données ?
- ▶ Est-il possible d'arrêter périodiquement la base ?
- ▶ Est-il possible de réaliser une sauvegarde complète de la base pendant l'arrêt ?
- ▶ Les données sont-elles mises à jour quotidiennement par les utilisateurs ?
C'est typiquement le cas dans une application transactionnelle.
- ▶ Les données sont-elles mises à jour périodiquement (toutes les nuits, toutes les semaines) et simplement consultées dans la journée ?
C'est typiquement le cas avec une application décisionnelle.

Principes

- ▶ Protection des fichiers sensibles :
 - ▶ Fichiers de contrôle
 - ▶ Fichiers de journalisation
 - ▶ Fichiers de paramètres serveur
 - ▶ Fichiers de données
- ▶ Une stratégie de sauvegarde/restauration
 - ▶ Adaptée à l'entreprise
 - ▶ Qualifiée et documentée
- ▶ Solutions :
 - ▶ outil RMAN
 - ▶ Commande système avec des requêtes SQL

Archivage des fichiers de journalisation

- ▶ Garantie aucune perte de données
- ▶ Mode ARCHIVELOG

Principe

- ▶ To : sauvegarde des fichiers de données
- ▶ Archivage activée
- ▶ T1 : incident se produit et le fichier de données est perdu

Récupération des données :

- ▶ Prendre la dernière sauvegarde
- ▶ Appliquer sur cette sauvegarde les fichiers de journalisation archivées

- ▶ Les scénarios
 - ▶ Sauvegarde complète base fermée (cohérente)
 - ▶ Sauvegarde complète base ouverte (incohérente)
 - ▶ Sauvegarde partielle base ouverte
 - ▶ Sauvegarde incrémentale

Quelle stratégie prendre : Archivelog ou Noarchivelog

		Perte de données	
Sauvegarde base fermée ?		oui	Non
	OUI	Archivelog Noarchivelog	Archivelog
	NON	Archivelog	Archivelog

Sauvegarde restauration

Utilitaire RMAN :

1. Shutdown de la base
2. Startup mount
3. Backup..(conditions)

commande :

- ▶ rman target /
- ▶ Rman>

La commande LIST permet d'interroger le référentiel RMAN

L'outil est installé dans EM ou utilisable en ligne de commande

Configurer RMAN

Configuration actuelle

► RMAN>show all

Configurer la destination de la sauvegarde

CONFIGURE CHANNEL DEVICE TYPE DISK options

```
Ram >CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT \home\oracle\sauve\%u  
MAXPIECESSIZE 4 G;
```

(taille des fichiers de sauvegarde est limité à 4 GO, le jeu comprendra plusieurs jeu)

Pour revenir à la configuration par défaut

```
Rman>CONFIGURE CHANNEL DEVICE TYPE DISK CLEAR
```

Configurer la politique de conservation

Permet de prévoir combien de jours, vous souhaitez revenir

Fenêtre de restauration

CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF n DAYS;

Redondance

CONFIGURE RETENTION POLICY TO REDUNDANCY n;

Configurer RMAN

Configuration de la sauvegarde automatique des fichiers de contrôles

CONFIGURE CONTROLFILE AUTOBACKUP ON;

CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO 'format';;

Utilisation de la zone de récupération rapide

- ▶ Conseillé
- ▶ Quota d'espace alloué : paramètre DB_RECOVERY_FILE_DEST_SIZE
- ▶ Répertoires définis par oracle ou sont stockés les fichiers générés

Commande VALIDATE

VALIDATE clause

DATABASE, DATAFILE_LIST, CURRENT, SPFILE.....

Détecter les problèmes de corruption ou de fichiers manquants

Sauvegarde

Base montée : fichiers de contrôle ouverts

Sauvegarde base ouverte si la base fonctionne en ARCHIVELOG

Fichiers de données, contrôle, fichiers de journalisation archivés, paramètres,...

Syntaxe

BACKUP mode quoi [option]

Mode :

- ▶ INCREMENTAL LEVEL n [CUMULATIVE]
- ▶ VALIDATE
- ▶ AS COPY AS [COMPRESSED] BACKUPSET
- ▶ Option : DATABASE, DATAFILE cible,....

Sauvegarde incrémentielle

Sauvegarde que les blocs modifiés

Les sauvegardes

Incrémentale de Niveau 0 : sauvegarde tous les blocs, correspond à une sauvegarde totale

Incrémentale différentielle de Niveau 1 : sauvegarde tous les blocs modifiés depuis la sauvegarde niveau 0 ou 1

Incrémentale cumulative de Niveau 1 : sauvegarde tous les blocs modifiés depuis la dernière sauvegarde de niveau 0

Scénario 1 : base ouverte (ARCHIVELOG)

Dimanche : BACKUP INCREMENTAL LEVEL 0 DATABASE;

Lundi au samedi : BACKUP INCREMENTAL LEVEL 1 CUMULATIVE DATABASE;

Scénario 2 : base fermé (NOARCHIVELOG)

On ajoute les commandes

Shutdown immediate;

Startup mount;

Backup....

SQL>alter database open;

Dimanche : BACKUP INCREMENTAL LEVEL 0 DATABASE;

Lundi au samedi : BACKUP INCREMENTAL LEVEL 1 CUMULATIVE DATABASE;

Lancer l'outil RMAN
\$ rman target/

SQL>shutdown immediate;
SQL>startup mount;
Sql>exit

Visualisation des paramètres RMAN
Rman> Show all;

Modification des paramètres RMAN
RMAN >CONFIGURE;

1. Modifier le chemin de la sauvegarde

CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT
'/home/oracle/sauvegarde/ora_df%t_s%s_s%p';

Faire une sauvegarde complete avec la base ouverte

Passage en mode ARCHIVELOG

connect /as sysdba

sql>shutdown;

startup mount;

Enable database ARCHIVELOG mode

alter database archivelog;

Shutdown and restart the database instance.

Sql>shutdown immediate

Sql>startup

RMAN> BACKUP DATABASE;

ATELIER 14 : sauvegarde incrémentale

Dimanche : sauvegarde incrémentale niveau 0

```
connect /as sysdba  
sql>shutdown immediate;  
Sql> startup mount;  
RMAN> Backup INCREMENTAL LEVEL 0 DATABASE;
```

Du lundi au samedi : Faire une sauvegarde incrémentale de niveau 1

```
connect /as sysdba  
sql>shutdown immediate;  
Sql> startup mount;  
  
RMAN> Backup INCREMENTAL LEVEL 1 DATABASE;
```

Trouver les informations sur les sauvegardes : commande LIST

- Lister les sauvegardes par jeu de sauvegarde
LIST BACKUP OF DATABASE
LIST BACKUPSET 8;

Supprimer des sauvegardes : commande DELETE

- Supprimer la sauvegarde complète

Récupération

Dépend de plusieurs facteurs

- ▶ De la nature du fichier endommagé ou perdu
 - ▶ Fichiers de données
 - ▶ Fichiers de contrôle
 - ▶ Fichiers de paramètre serveur
 - ▶ Fichiers de journalisation

Que faire en cas de problème ?

- ▶ Bien identifier la nature du problème
- ▶ Définir le mode opératoire

- ▶ Restauration : consiste à extraire d'une sauvegarde les fichiers nécessaires
- ▶ Récupération : appliquer les fichiers de journalisation aux fichiers récupérés dans la sauvegarde

Principes généraux de la récupération

En mode NOARCHIVELOG

Restaurer la dernière sauvegarde complète de la base;

Redémarrer la base

Inconvénients : toutes les données saisies après la sauvegarde sont perdues.

Si les fichiers de journalisation sont correctes, on lance la récupération comme en mode ARCHIVELOG

Les techniques de FLASHBACK

- ▶ Ensemble de fonctionnalités qui permettent de voir l'état passé des données ou de ramener une table ou la totalité d'une base dans le passé.

- ▶ Activer la fonctionnalité

Alter database flashback on;

Les outils

- ▶ Utiliser Flashback technology

- ▶ Flashback Query

- Lire les données dans un instant passé

- ▶ Flashback Version Query

- Voir toutes les versions d'une ligne

- ▶ Flashback Transaction Query

- Voir les modifications faites par plusieurs transactions

- ▶ Flashback Transaction DATABASE

- Annule les modifications d'une transaction et des transactions dépendantes

- ▶ Flashback Table

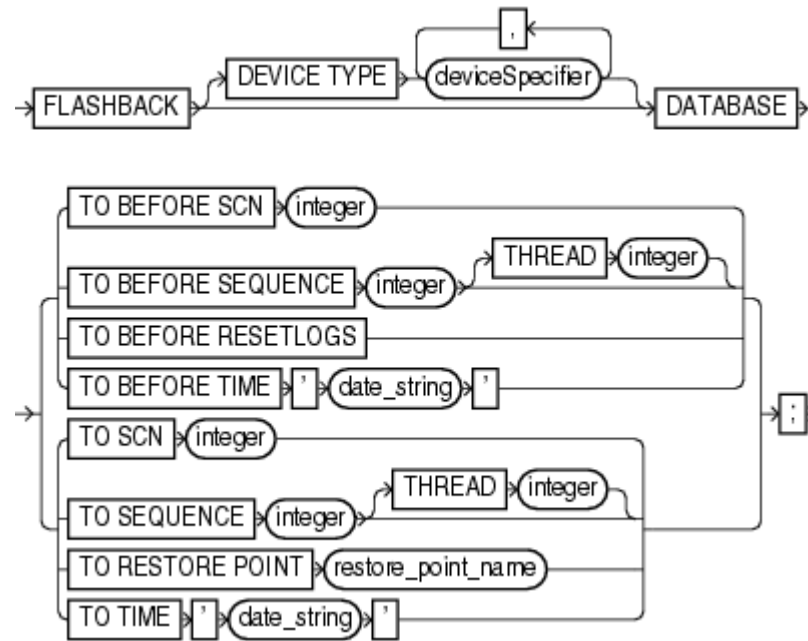
- Ramener une table dans l'état, à un certain moment dans le passé.

- ▶ Flashback Drop ramener la table dans l'état juste avant sa suppression.

- ▶ Flashback Database

- Ramener une base de données dans l'état, à un certain moment dans le passé.

Atelier 17 : Flashback de la base à un instant donné



Atelier 17 : Flashback QUERY

Regarder la documentation ORACLE

Faire un flashback avec l'attribut TO SCN

- Situation de départ
sql>Select ename from scott.emp where numero=7900

```
Sql>Select TO_CHAR(SYSDATE, 'DD/MM/YYYY HH24:MI:SS') " sysdate ",  
dbms_flashback.get_system_change_number " SCN" From dual;
```

Faire un UPDATE

```
Sql>Update scott.emp set ename='jean' and empno =7900;
```

```
Sql>Commit;
```

Vérifier le SCN nouvellement attribué

```
Sql>Select TO_CHAR(SYSDATE, 'DD/MM/YYYY HH24:MI:SS') " sysdate ",  
dbms_flashback.get_system_change_number " SCN" From dual;
```

```
sql>Select ename from scott.emp where numero=7900
```

```
Sql> alter table scott.emp enable row movement;
```

```
Sql flashback table scott.emp to scn 25800093;
```

```
Select ename from scott.emp where numero=7900;
```

Conclusion

Refaire un flashback avec l'attribut TO BEFORE TIME

Atelier 18 : Flashback DROP

Regarder la documentation ORACLE

- ▶ Supprimer la table emp
- ▶ Lancer le flashback avant la suppression.

Database sécurité

Une base de données est confidentielle

Toutes données entrées sortantes et entrantes sont sur la responsabilité de l'entreprise

Les aspects à respecter :

- ▶ Restreindre l'accès des données et des services
- ▶ Authentifier les utilisateurs
- ▶ Surveiller les activités
- ▶ Les utilisateurs avec le privilège DBA doivent être contrôlés
- ▶ Les responsabilités doivent être partagé
- ▶ Séparation des comptes DBA et system
- ▶ Séparer les comptes d'opération et les comptes DBA

Protéger les authentifications

- ▶ Utiliser un fichier de mots de passe forts
 - ▶ Renouvellement des mots de passe
 - ▶ Intégration à la politique de sécurité de l'entreprise
 - ▶ Utiliser les certificats
 - ▶ Utiliser les couches SSL
-
- ▶ Auditer les comptes avec privilèges SYSDBA et SYSOPER

Les utilitaires d'exportation et importation

- ▶ DATA PUMP

Déplacer des données ou métadonnées entre des bases Oracle

- ▶ EXPORT

Exporter dans un fichier binaire : complet, ou partiel

- ▶ IMPORT : complet, ou partiel

- ▶ SQL*LOADER

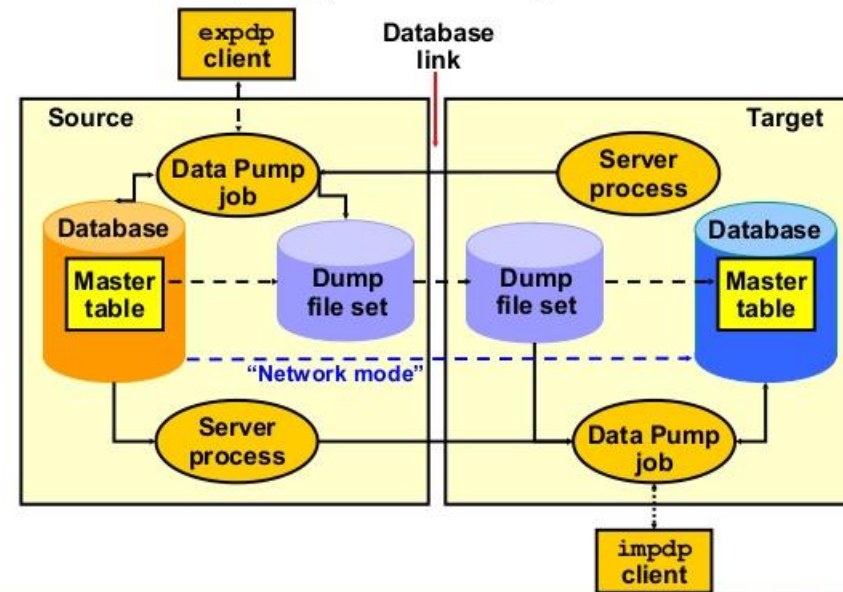
Permet de charger dans une table des données stockées dans des fichiers ACII, ou fichier plat

DATA PUMP

- ▶ Architecture DATA PUMP
- ▶ Un package PL/SQL DBMS_DATAPUMP
- ▶ Un package PL/SQL DBMS_METADATA
- ▶ Deux outils de commande
 - ▶ expdp
 - ▶ Impdp

Les deux outils de commande
sert d'interface avec le package
DBMS_DATAPUMP

Data Pump Export and Import: Overview



Les modes EXPORT ou IMPORT

- ▶ Complet
- ▶ Schéma
- ▶ Table
- ▶ Tablespace
- ▶ Pour chaque mode, on peut préciser plus finement les objets
 - ▶ expdp
 - ▶ Impdp

Mise en place

- ▶ Création d'un USER ; dba_exp ou dba_imp
- ▶ Privilèges : EXP_FULL_DATABASE ou IMP_FULL_DATABASE
- ▶ Avoir les droits pour accéder au répertoire et aux fichiers d'exploitation
- ▶ CREATE ou REPLACE DIRECTORY nom_repertoire AS 'chemin_repertoire'
- ▶ Ajouter un privilège à l'utilisateur sur le répertoire
- ▶ Grant privilège On DIRECTORY To user

Syntaxe expdp paramètres ou impdp

```
>Expdp dba_expl/oracle@nom_base DIRECTORY=dir_exp FULL=y  
DUMPFILE=exp.dmp
```

Atelier 20

Export de la base SCOTT

Export de 20% de la base SCOTT

Export de la table emp de la base SCOTT

Ateliers export -import

Lancer un export en s'appuyant sur un fichier de paramètres

```
> expdp scott parfile=emp_query.par
```

Fichier de paramètres : emp_query.par

```
QUERY=emp:"WHERE depno > 10 AND sal > 5000"
```

```
NOLOGFILE=y
```

```
DIRECTORY=dpump_dir1
```

```
DUMPFILE=exp1.dmp
```

Exporter des tablespaces

```
> expdp scott DIRECTORY=dpump_dir1 DUMPFILE=tbs.dmp
```

```
TABLESPACES=tbs_4, tbs_5, tbs_6
```

DATA PUMP Import

- ▶ Réaliser un export Full
- ▶ Supprimer une table
- ▶ Importer une table à partir du dump export_full
- ▶ Importer à partir d'une requête à partir du dump export_full

Principes généraux de la récupération

En mode ARCHIVELOG

- ▶ Restaurer la dernière sauvegarde de chaque fichier perdu
- ▶ Appliquer les fichiers de journalisations : archives puis ceux en lignes)
- ▶ Redémarrer la base

Identifier la nature du problème

- ▶ Message erreur concernant les fichiers de contrôle
- ▶ Message erreur concernant les fichiers de journalisation
- ▶ Message erreur concernant les fichiers de données

Les commandes RMAN

Commande RESTORE : restaurer les fichiers à partir des sauvegardes, les fichiers nécessaires

Commande RECOVER : récupération complète ou incomplète, appliquer les redo log aux fichiers récupérés

Atelier :

supprimer la base de données

Et recréer avec la sauvegarde

Scénario de récupération

Récupération du fichier de paramètres serveur

Récréer votre fichier de paramètres

Ou le récupérer dans la sauvegarde

RMAN> startup nomount

```
RMAN>restore spfile from autobackup db_recovery_file_dest  
'\ora01\oracle\app\oracle\oradata\flash_recover db_name 'orcl';
```

Redémarrer l'instance

RMAN>SHUTDOWN

RMAN>STARTUP

Scénario de récupération

Récupération du fichier de contrôle

Vérifier dans le fichier d'alerte quel fichier est endommagé

Dupliquer un des fichiers de contrôle pour remplacer les fichiers perdus (sans passer par la sauvegarde)

Récupération complète de la totalité de la base

Restaurer la base

```
Rman>STARTUP MOUNT;
```

```
RMAN>restore database;
```

Récupérer la base

```
Rman>STARTUP MOUNT;
```

```
RMAN>recover database;
```

Atelier 16 : récupération base de données ouverte

Réaliser un script

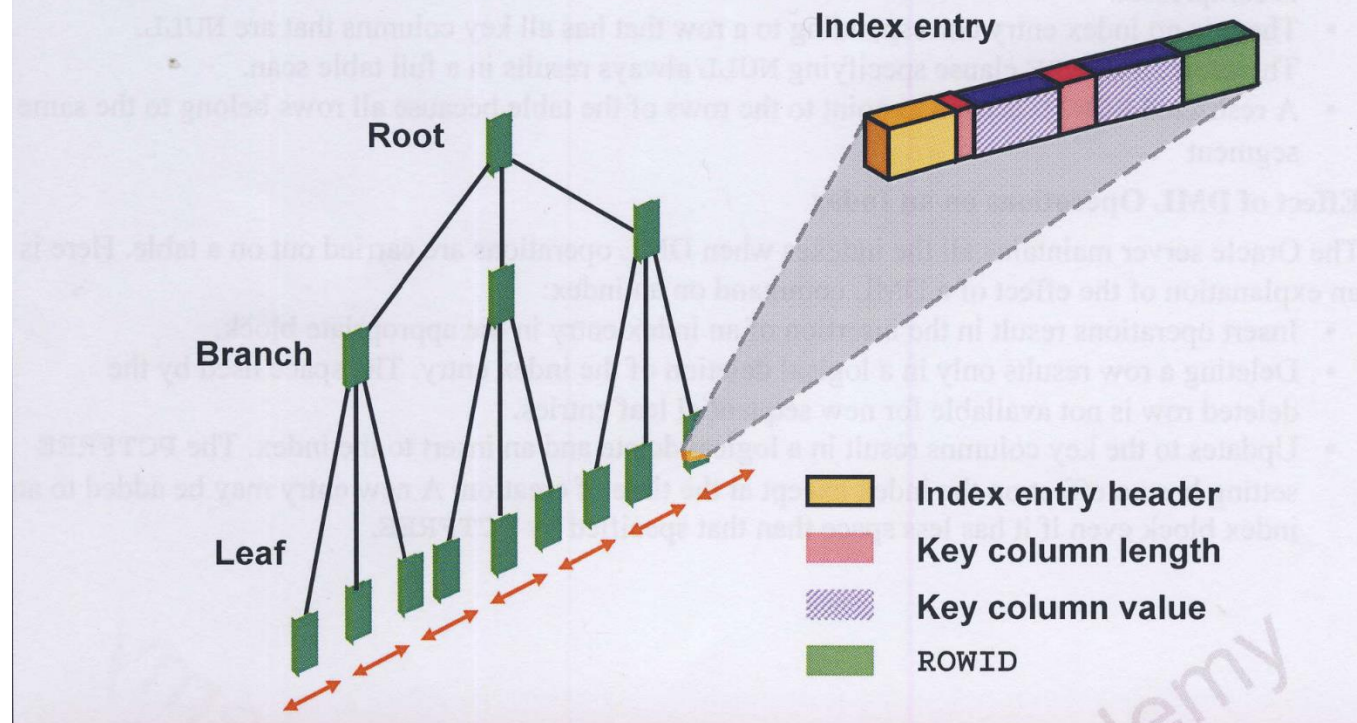
Le fichier de données INDX est perdu et le fichier de données est le numéro 6

1. Monter la base de données
2. Mettre OFFLINE les fichiers de données perdus
3. Ouvrir la base de données
4. Passer OFFLINE les tablespaces concernés
5. Restaurer les fichiers de données
6. Récupérer les fichiers de données
7. Passer ONLINE les tablespaces concernés

Gestion des index

- ▶ C'est un objet dédié sur une ou plusieurs colonnes
- ▶ Accès rapide aux lignes basées sur la clé de l'index
- ▶ Indépendant de la table
- ▶ Unique et non unique
- ▶ Tablespaces spécifiques

B-Tree Index



Les index

Les types d'index :

- ▶ B-tree Index
- ▶ Bitmap Index

But d'indexer une table

- ▶ Performance des requêtes
- ▶ Accès plus rapide dans les colonnes
- ▶ Eviter le FULL Scan des tables

Les index peut également améliorer dans l'application des clés primaires et uniques.

Création d'index

```
CREATE INDEX employees_idx1 ON employees (last_name, job_id);
```

```
CREATE INDEX employees_idx2 ON employees (job_id, last_name);
```

Structure d'un index B-TREE

- ▶ Les données des index sont stockés dans des blocs réservés aux index

Structure d'un bloc :

- ▶ Un bloc d'index comprend les données et des informations de contrôle
- ▶ Un bloc peut contenir plusieurs index
- ▶ Le remplissage du bloc est contrôlé par le paramètre PCTFREE
- ▶ Pas de PCTUSED pour les index

Avantages :

- ▶ Améliore la performance des requêtes (SELECT, UPDATE et DELETE) qui utilisent la clé d'index.
- ▶ Oracle maintient l'arborescence au fur et à mesure des entrées des index

Inconvénient :

- ▶ Dégrade les performances pour les mises à jour

Création d'index B-TREE

- ▶ Les colonnes souvent sollicitées par la clause WHERE
 - ▶ La requête ramène moins de 5 à 10% de lignes de la table
 - ▶ Pas d'indexation sur les petites tables
 - ▶ S'assurer que les requêtes soient bien écrites
 - ▶ Les valeurs NULL, LIKE (si le début de chaîne n'est pas connu '%tel'), NOT IN SUBSTR, ne sont pas indexées
-
- ▶ Stockage des index et tablespace : même attribut que pour les tables
 - ▶ Attribut pour le STORAGE : INITIAL, NEXT, MINEXTENTS, MAXEXTENTS, PCTINCREASE

```
CREATE INDEX emp_nomidx ON emp(nom,prenom)
    TABLESPACE tbs_index
    STORAGE (INITIAL 20K
    NEXT 20k
    PCTFREE 20);
```

Index d'une contrainte de clé primaire ou unique

- ▶ Le stockage de l'index d'une clé primaire ou unique est spécifié à la création de la contrainte ou en ajoutant une contrainte

Spécifier un Index déjà existant

- ▶ Avec l'attribut USING INDEX
- ▶ Avec les attributs de l'index

USING INDEX ;

```
ALTER TABLE emp ADD CONSTRAINT emp_index1 PRIMARY KEY(numero)
```

```
USING INDEX
```

```
TABLESPACE indx
```

```
PCTFREE 10
```

```
STORAGE INITIALM 2 M;
```

La vue DB_CONSTRAINTS a deux colonnes : INDEX_OWNER et INDEX_NAME qui fait le lien entre l'index et la contrainte

Index d'une contrainte de clé primaire ou unique

Définition du stockage de l'index

```
Alter table adherent ADD CONSTRAINT adherentpk PRIMARY KEY (numero)  
USING INDEX TABLESPACE tbs_indx PCTFREE 0 STORAGE (INITIAL 2M);
```

Spécification d'un index déjà existant

```
ALTER TABLE adherent ADD CONSTRAINT adherentuq1  
UNIQUE (nom, prenom, telephone)  
USING INDEX adherentindx;
```

Création complete

```
ALTER TABLE adherent ADD CONSTRAINT adherentuq1  
UNIQUE (nom, prenom, telephone)  
USING INDEX  
(CREATE INDEX emp_nomidx ON emp(nom, prenom)  
TABLESPACE tbs_index  
STORAGE (INITIAL 20K  
PCTFREE 20);
```

Recommandation

Même principe que les tables

- ▶ Estimer le nombre de lignes attendues
- ▶ Créer les index dans les conditions de production
- ▶ Charger la table avec un jeu de données représentatives
- ▶ Calculer le nombre de blocs d'index utilisés par ce jeu d'essai
- ▶ Estimer le nombre de blocs sur 1, 2 ou 3 ans (règle de trois)

`ANALYSE INDEX nom_index VALIDATE STRUCTURE`

`Select lf_blks+br_blks FROM index_stats where name='nom_index'`

Estimation de PCTFREE : $100 * (1 - Ni/Nf)$

Ni : nombre initial et final de blocs utilisés

Surveiller l'espace occupé

Les vues DBA_SEGMENTS DBA_EXTENTS, DBA_INDEXES

Calculer avec le package DBMS_SPACE

- ▶ Se servir aussi du package DBMS_STATS
- ▶ Se servir aussi SQL ANALYSE ... VALIDATE STRUCTURE
- ▶ L'ordre ANALYSE INDEX nom_index VALIDATE STRUCTURE vérifie l'intégrité de l'index et le stockage dans les blocs

Le résultats à consulter dans la vue INDEX_STATS

- ▶ Name : nom de l'index
- ▶ HEIGHT ; nombre de l'arbre
- ▶ BLOCKS : nombre de blocs
- ▶ LF_BLKs : nombre de blocs feuille dans l'index
- ▶ BR_BLKs : nombre de blocs branche dans l'index
- ▶ LF_ROWS : nombre de lignes de l'index
- ▶ PCT_USED : % alloué dans le bloc pour l'index
- ▶ DEL_LF_ROWS : nombre de lignes supprimées dans l'index

Réorganiser le stockage de l'index

- ▶ Libérer l'espace libre au dessus de la HWM
- ▶ Réorganiser un index
- ▶ Modifier les valeurs des attributs : tablespace, extents

Plusieurs techniques

- ▶ **Ordre SQL ALTER INDEX....DEALLOCATE UNUSED**

Libérer de l'espace au dessus de HWM

- ▶ **Ordre SQL ALTER INDEX....COALESCE**

Fusionner le contenu de blocs feuille

- ▶ **Ordre SQL ALTER INDEX....SHRINK SPACE**

Compacter les lignes d'une index (compacte le segment aussi)

- ▶ **Ordre SQL ALTER INDEX....REBUILD**

Reconstruire un index segment compris

Informations sur les index : DBA_INDEXES et DBA_IND_COLUMNS
INDEX_STATS, DBA_SEGMENTS, DBS_EXTENTS

Ok : le mieux adapté
X : possible

	DEALLOCATE	COALESCE	SHRINK	REBUILD
Libérer espace HWM	OK		X	X
Améliorer le taux de remplissage		OK	OK	OK
Réorganisation globale				OK

ATELIER 17

- 1 - Vérifier les index positionnés sur les tables de Scott
- 2 - Réaliser une requête : création d'un Index sur 3 colonnes sur une table de SCOTT
- 3 - Réaliser une requête : création d'un Index sur une clé primaire d'une table de SCOTT
- 4 - Une simple Analyse des index avec ANALYSE INDEX ...VALIDATE STRUCTURE
 - ▶ Découvrir l'outil
 - ▶ Visualiser les résultats : height, lf_blks, bl_blks, blocks, pct_used, lf_rows
- 5 - Calculer avec le package DBMS_SPACE
 - ▶ Se servir aussi du package DBMS_STATS
- 6 - Libérer de l'espace au dessus de HWM
 - ▶ 7 - Augmentation de la performance des index avec l'ordre suivant

Ordre SQL ALTER INDEX...DEALLOCATE UNUSED

SQL*LOADER

Charger des données

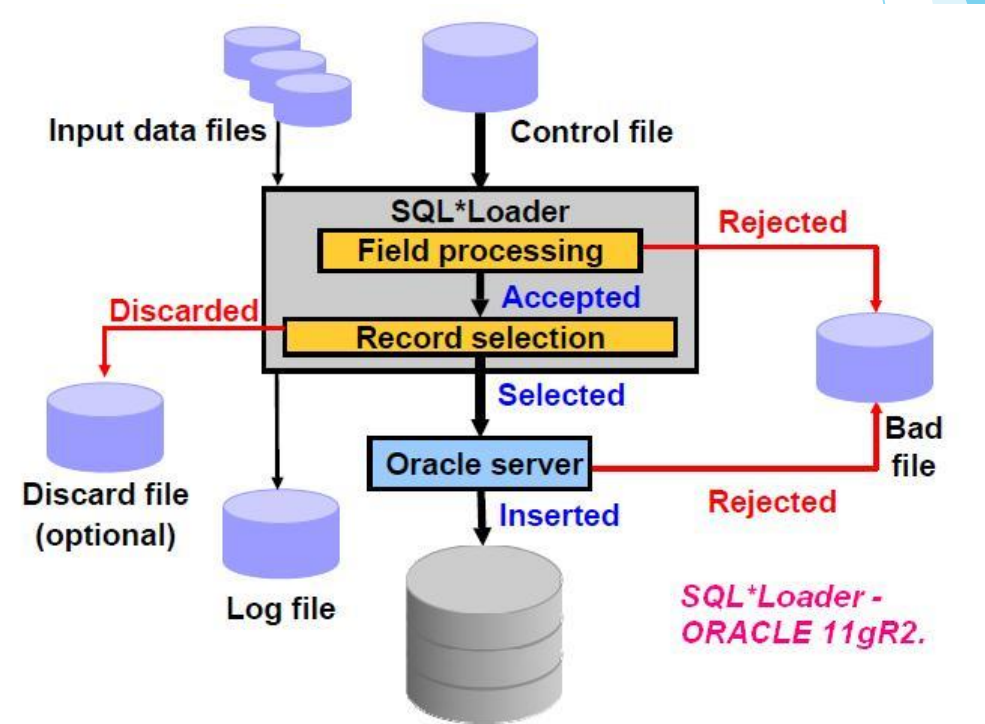
à partir de un ou plusieurs fichiers externes

Avec des enregistrements de longueur fixe ou variable

En appliquant des contrôles, des filtres

Résultats dans des journaux :

- ▶ Rejets
- ▶ Refus



SQL*LOADER

Mise en œuvre : seulement en ligne de commande

```
$sqlldr userid=scott/oracle control=charge.ctl log=charge.log
```

```
$sqlldr PARTFILE=charge.par
```

Certains paramètres sont inclus dans le fichier de contrôle

OPTIONS : permet de spécifier dans le fichier de contrôle les options de la ligne de commande : OPTIONS (SKIP=10,LOAD=900, ERRORS=100) Paramètres de la ligne de commande

BAD : nom du fichier BAD (.bad) , fichier de rejets

Control : nom du fichier de contrôle

DATA : nom du fichier de données (.dat)

DISCARDFILE : nom du fichier DISCARD(.dsc)

DISCARDMAX : nombre de rejets acceptés

LOAD : nombre d'enregistrements à charger

LOG : nom du fichier journal

PARTFILE : nom du fichier de paramètres

SKIP : nombre d'enregistrement à éliminer avant le chargement

SQL*LOADER

LOAD DATA

La clause INFILE : Fichiers de données à traiter et son chemin

Si * : les données sont dans le fichier de contrôle

INSERT : ajout uniquement pour une table vide

APPEND : ajout dans la table (vide ou non vide)

REPLACE : Remplace le contenu de la table (DELETE exécuté avant)

TRUNCATE : remplace tout le contenu de la table

Exemple de fichiers de paramètres

Usersid=system/oracle2018

Control=transfert.ctl

Data=transfert_esaip.dat

Log=transfert_esaip.log

Bad=transfert_esaip.basd

Discardfile=transfert_esaip.dsc

SQL*LOADER ; table de destination

INTO TABLE : donne le nom de la table à charger et décrit la façon de la faire
INSERT|APPEND|REPLACE|TRUNCATE , mode de l'import de la table

Insert : dans une table vide

APPEND : ajout à la table

REPLACE remplace tout le contenu de la table, (un ordre DELETE est lancé avant)

TRUNCATE Remplace tout le contenu de la table

WHEN : indique une condition, définit la position de son caractère, début de position, et fin

FIELDS TERMINATED BY 'x' : enregistrement de longueur variable, indique comment sont délimités les champs

X: séparateur : virgule, point virgule

Y : guillemets, apostrophes

TRAILING NULLCOLS : les colonnes sont présentes dans le fichier sont mises à NULL

Colonne POSITION (x,y) type clause SQL : nom de la colonne à alimenter dans la table

BEGINDATA : marque le début des données insérées dans le fichier de contrôle

Fichier de control

Load data

infile *

INTO TABLE scott.emp

APPEND

FIELDS BY TERMINATED BY','

TRAILING NULLCOLS

(nom,prenom,dept)

▶ BEGINDATA

VALJEAN, Jean, 56

Hugo,Victor,44

Enregistrements à longueur variable avec séparateur

SQL*LOADER

Atelier 19 :

Insertion de données dans la table EMP de SCOTT

Créer 1 fichier .csv avec 3 lignes d'insertion

Insérer les données dans la table