Architecture du serveur de base de données Oracle

Rôles d'un administrateur de BD :

- Installer le serveur de BD, la créer, la démarrer, l'arrêter et manipuler les paramètres d'initialisation.
- Créer et configurer les espaces de stockage physiques et logiques.
- Créer des utilisateurs et leur affecter des privilèges et des rôles.
- Auditer et sécuriser la BD.
- Maintenir les ressources et assurer la continuité de fonctionnement et contrôler les indices de performance.
- Assurer l'export et l'import des données interbase.
- Assurer la sauvegarde et la récupération en cas de pannes.

Serveur de base de données Oracle :

Système de gestion de base de données qui fournit une approche intégrée, complète et ouverte de la gestion des informations. Il se compose d'une instance Oracle et des fichiers de base de données.

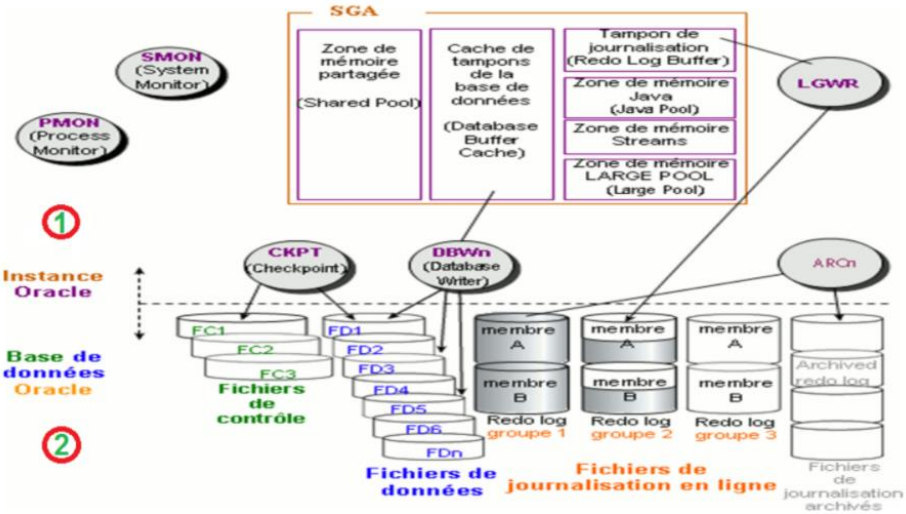
Scénario de connexion à une BD Oracle :

1- Le client contacte le listener Oracle et choisit une instance (demande d'un nom de service).
2- Le listener démarre un processus dédié appelé processus serveur.
3- Le listener envoie un accusé de réception au client avec l'adresse du processus serveur dédié.
4- Le client établit une connexion avec le processus serveur dédié.
5- Le processus serveur se connecte à l'instance Oracle pour le compte du processus utilisateur (création d'une session utilisateur)

Le processus utilisateur n'interagit pas directement avec le serveur Oracle. C'est le processus serveur qui le fait.

Architecture : un serveur Oracle se compose d'une instance Oracle et d'une BD Oracle.

Une instance Oracle est composée d'une SGA (System Global Area), structure de mémoire partagée par tous les processus serveurs et les processus en arrière-plan, et de plusieurs processus oracle en arrière-plan ayant chacun un rôle.



System Global Area (SGA) : constitué de

Zones de mémoire obligatoires :

- Zone de mémoire partagée (Shared Pool).
- Cache de tampons de la BD (DB buffer cache).
- Tampon de journalisation (Redo log buffer).

Zones mémoire non obligatoires :

- Zone de mémoire LARGE POOL.
- Zone de mémoire Java (Java Pool).
- Zone de mémoire Streams (Streams pool).

Zone de mémoire partagée (Shared Pool) :

Elle est constituée de 2 structures mémoire liées aux performances :

- Cache du dictionnaire de données (row cache) :

Quand un utilisateur soumet une requête SQL, le processus serveur extrait des informations stockées dans les tables du dictionnaire et les place dedans pour des besoins de réutilisation.

- Cache "Library" :

Il conserve, à des fins de partage, des informations sur les commandes SQL et le code PL/SQL récents soumis par des utilisateurs de la BD → Optimiser les ressources.

Cache de tampons de la BD (DB buffer cache) :

Géré par un algorithme LRU, il conserve des copies des blocs de données extraits des fichiers.

+ Gain de performances lors de l'obtention et de la mise à jour de données.

DB_BLOCK_SIZE : détermine la taille du bloc principal.

Tampon de journalisation (Redo log buffer) :

Il récupère les données et enregistre toutes les modifications apportées aux blocs de données de la BD. Ces modifications constituent des entrées de journalisation.

LOG_BUFFER : définit la taille du tampon.

Les processus en arrière-plan :

Processus DBWn (DB Writer) : écrit les blocs modifiés du cache tampon de la BD dans les fichiers de données dans les cas suivants :

- Point de reprise
- Seuil des tampons "dirty" atteint
- Aucune mémoire tampon disponible
- Temps imparti dépassé
- Demande de ping RAC
- Tablespace hors ligne ou en lecture seule
- DROP ou TRUNCATE sur une table
- BEGIN BACKUP sur un tablespace

Processus LGWR (Log Writer) : écrit les entrées de journalisation sur le disque dans les cas suivants :

- Validation
- Un tiers du cache est occupé
- La journalisation atteint 1 Mo toutes les trois secondes avant que le processus DBWn ne procède à une opération d'écriture.

Processus SMON (System Monitor) : assure la récupération en cas de panne, la fusion de l'espace libre et la libération des segments temporaires.

Processus CKPT (Check Point) : informe le processus DBWn des points de reprise pour lancer la mise à jour, met à jour les entête de fichiers de données et de contrôle avec les informations sur le point de reprise.

PMON (Process Monitor) : assure le nettoyage des processus utilisateur en cas d'échec (session abandonnée) : annule la transaction, redémarre les répartiteurs interrompus, etc.

Program Global Area (PGA) :

C'est une structure de mémoire créée pour chaque utilisateur connecté. Elle stocke des informations de contrôle spécifiques à la session de l'utilisateur (Ex : zones privées pour le traitement des curseurs, variables attachées (bind), informations sur la session, etc.)

Chaque processus serveur ou en arrière-plan dispose de sa propre mémoire PGA privée qui lui est exclusivement réservée.

Lorsque le processus utilisateur se déconnecte, le processus serveur associé prend fin et la m mémoire PGA est libérée.

Le dictionnaire de données :

Le dictionnaire de données Oracle est la description d'une BD, créé et mis à jour en même temps qu'elle. Il contient le nom et les attributs de tous les objets de la base et ses informations sont stockées dans des tables de base.

Il appartient à l'utilisateur SYS, mais les autres utilisateurs peuvent y accéder par le biais de vues prédéfinies :

Préfixe **USER** : vue de l'utilisateur (ce que contient son schéma).

Préfixe **ALL** : vue étendue de l'utilisateur (ce à quoi il peut accéder).

Préfixe **DBA** : vue de l'administrateur de la BD (ce que contient le schéma de chaque user).

Préfixe **V$** : fichier de données de performances (taille mémoire, user connectés, etc.).

La colonne **LAST_NUMBER** affiche le prochain numéro de séquence disponible.

Exemples :

SELECT * FROM dictionary WHERE table_name = USER_OBJECTS ;

SELECT table_name FROM user_tables;

SELECT constraint_name FROM user_cons_columns WHERE table_name = EMPLOYEES ;

Les vues V$:

Ce sont des vues dynamiques qui enregistrent l'activité en cours de la BD. Elles sont constamment mises à jour lorsque la BD est active et accessibles par un DBA.

Les informations sont lues à partir de la mémoire et du fichier de contrôle.

Exemples :

V$session : affiche les sessions en cours.

V$logfile : affiche la liste des fichiers journaux.

V$log : affiche le statut des groupes de fichiers journaux.

Démarrage de la BD Oracle :

Paramètres d'initialisation :

Les paramètres d'initialisation permettent entre autres de spécifier le nom de la BD, l'emplacement des fichiers de contrôle, le répertoire de destination des fichiers de données (datafiles), la destination des fichiers de journalisation (redo log files), etc.

Il existe deux types de paramètres d'initialisation :

Paramètres statiques :

Ils ne peuvent être modifiés que dans le fichier de paramètres uniquement à l'aide des commandes **ALTER SYSTEM** avec l'option **SCOPE='SPFILE'**.

Un redémarrage de l'instance est nécessaire pour que les modifications prennent effet.

Paramètres dynamiques :

Ils peuvent être modifiées tant que la BD est en ligne et sont valides pour la durée de la session ou dans les limites établies par le paramètre **SCOPE**.

Les modifications se font en deux niveaux : niveau session et niveau système.

Les modifications sont effectuées à l'aide des commandes **ALTER SESSION** et **ALTER SYSTEM.**

Fichiers de paramètres d'initialisation :

Une BD oracle n'est disponible à l'utilisateur que lorsque le DBA a démarré une instance et ouvert la BD.

Pour démarrer une instance, le serveur Oracle doit lire le fichier de paramètres d'initialisation qui contient les paramètres d'initialisation pour allouer la SGA et démarrer les processus d'arrière-plan. Il existe deux types de fichier de paramètres d'initialisation :

Fichier de paramètres statique PFILE **initSID.ORA** :

C'est un fichier texte, indispensable dans les postes clients, qui ne peut être ouvert que lors du démarrage de l'instance. S'il est modifié (manuellement ou pas) en cours, l'instance doit être interrompue et redémarrée pour que les nouvelles valeurs des paramètres soient effectives.

Créer un fichier au format texte à partir d'un fichier de paramètre au format binaire :

create pfile [='nom pfile'] From Spfile ['nom spfile'] ;

Fichier de paramètres persistant SPFILE **spfileSID.ORA** :

C'est un fichier binaire, accessible par le serveur de BD qui présente un moyen dynamique de gérer les paramètres.

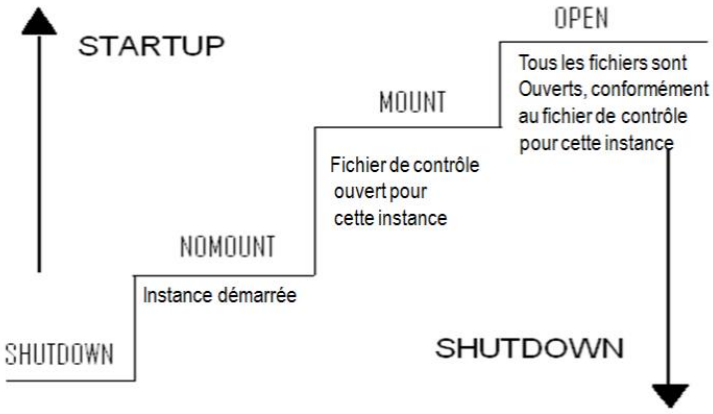
Les modifications apportées par le serveur et ne nécessitent pas une installation en copie dans les postes clients en cas d'une connexion à distance.

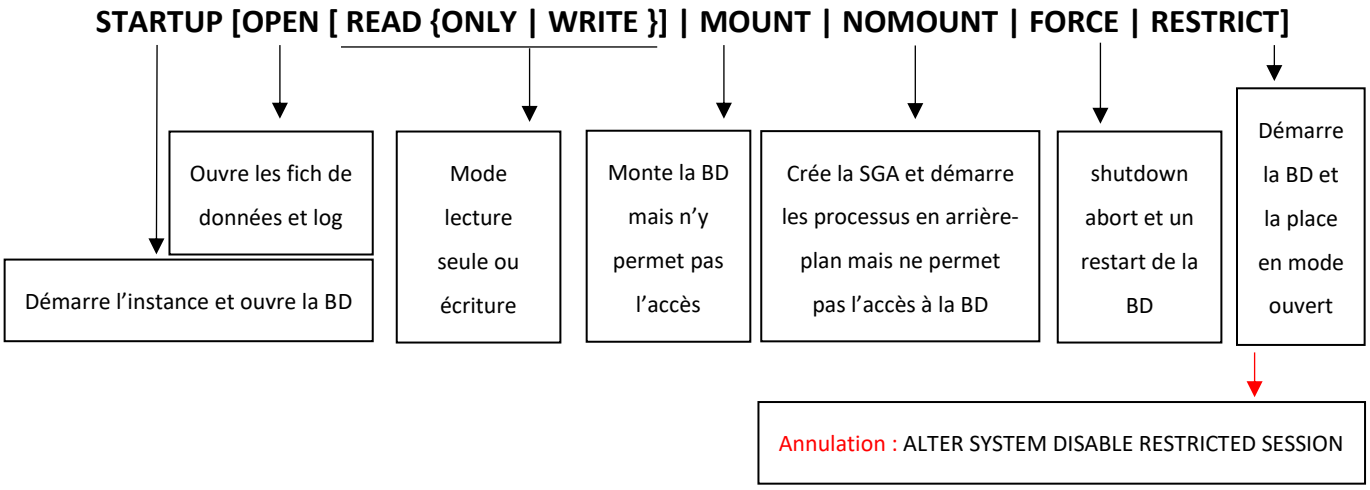
Créer un fichier de paramètre binaire à partir d'un fichier de paramètre texte :

create pfile ['nom pfile'] From Spfile ['nom spfile'] ;

Démarrage de la BD :

Lors du démarrage, les structures mémoires et les processus sont initialisés et démarrés. Différents modes de démarrages sont possibles :





La commande ALTER :

Permet de faire passer la BD du statut NOMOUNT à MOUNT ou du statut MOUNT à OPEN :

- ALTER DATABASE { NOMOUNT | MOUNT }
- ALTER DATABASE { MOUNT | OPEN }

Pour que les données ne soient pas modifiées, on peut ouvrir la base en mode read-only :

- ALTER DATABASE OPEN [READ WRITE| READ ONLY]

Arrêt de la BD :

Fermer la BD :

- Quand la BD se ferme, Oracle écrit les changements de la BD buffer cache et les entrées du buffer redo log dans les data files et redo log files.
- Les control files restent ouverts.

Démonter la BD d'une instance

- Fermeture des control files.

Arrêt de l'instance

- Libération de la SGA.
- Arrêt des background processes.
- Fermeture des fichiers trace et ALERT (journaux chronologiques des messages d'erreur).

SHUTDOWN [NORMAL | TRANSACTIONNEL | IMMEDIATE | ABORT]

Arrêt NORMAL (par défaut) :

- Le serveur attend la déconnexion de tous les utilisateurs avant de terminer l'arrêt (temps d'attente peut être important).
- Oracle ferme et démonte la BD avant d'arrêter l'instance.
- Pas de restauration de l'instance lors du démarrage suivant (car toutes les informations modifiées encore présentes dans la SGA sont écrites dans les data files et redo log files).

Arrêt TRANSACTIONNEL :

- Le client ne sera plus connecté dès la fin de la transaction en cours.
- Pas de restauration de l'instance lors du démarrage suivant.

Arrêt IMMEDIATE :

- Le serveur n'attend pas la déconnexion des utilisateurs avant de terminer l'arrêt.
- Les ordres SQL en cours ne seront pas traités.
- Oracle ferme et démonte la base avant d'arrêter l'instance.
- Pas de restauration de l'instance lors du démarrage suivant.

Arrêt ABORT :

- Le serveur oracle n'attend pas la déconnexion des utilisateurs avant de terminer l'arrêt.
- Les transactions non commitées ne seront pas effectuées (rollback).
- L'instance sera fermée sans la fermeture des fichiers.
- Restauration de l'instance lors du démarrage suivant.

Les structures de stockage Oracle

Structures de stockage logiques et physiques :

Tablespaces : Une BD est définie avec au moins un tablespace (SYSTEM tablespace) contenant le dictionnaire de données, n'appartenant qu'à une seule BD.

Un tablespace ne peut jamais être désactivé et peut être :

- Actif (online) : ses données sont accessibles aux utilisateurs.
- Désactivé (offline) : ses données ne sont plus accessibles aux utilisateurs.

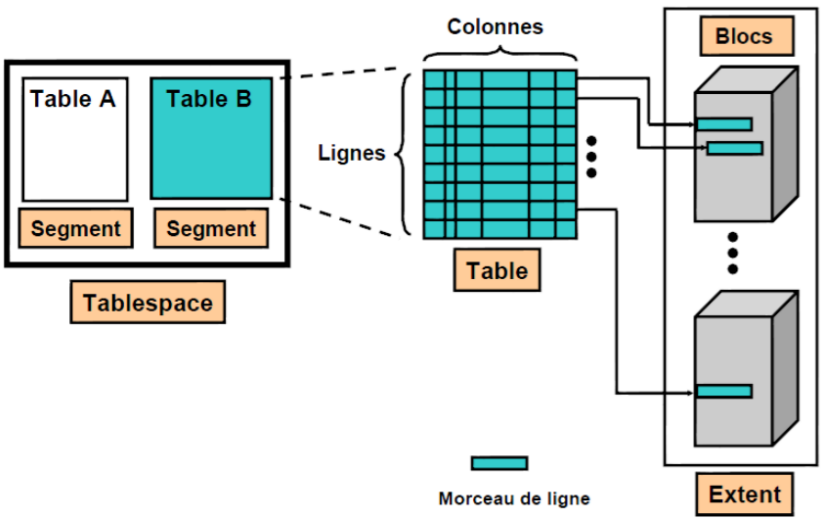
Définir et paramétrer différemment plusieurs tablespaces permet au DBA de :

- Organiser la base : assigner des quotas de ressources aux utilisateurs et contrôler la disponibilité des données en mettant hors service ou en lecture seule certains tablespaces.
- Améliorer la performance en répartissant les zones de stockage sur plusieurs disques.

Extension (ou extents) :

Extension (ou extents) : C'est une suite contiguë de blocs (au sens de l'emplacement sur le disque). Une extension est affectée à un type de données (ex : enregistrements d'une table) et le nombre de blocs dans celle-ci est fixé par le DBA.

Segments : C'est un ensemble d'extensions. Chaque segment est dédié au stockage d'un type particulier d'informations (tables, index, etc.)



Manipuler les tablespaces :

Gestion de l'espace dans les tablespaces :

Tablespace géré localement (recommandé) :

- Les extents libres sont gérés dans le tablespace.
- Un bitmap est utilisé pour enregistrer les extents libres.
- Chaque bit correspond à un bloc.
- La valeur du bit indique si le bloc est libre ou utilisé.

Tablespace géré au moyen du dictionnaire :

- Les extents libres sont gérés par le dictionnaire de données.
- Les tables appropriées sont mises à jour lorsque des extents sont alloués ou libérés.

Créer un TABLESPACE :

```
CREATE {BIGFILE|SMALLFILE} TABLESPACE nom_tablespace
[DATAFILE ['nom_fichier'] [SIZE integer {K|M|G|T}]
[AUTOEXTEND
      {OFF| ON [NEXT integer {K|M|G|T}]
      [MAXSIZE {UNLIMITED | integer {K|M|G|T}] } ] [,...]
]
[BLOCKSIZE integer [K]]
[{LOGGING|NOLOGGING}] [{ONLINE|OFFLINE}]
[Extent_mangement_clause]
[DEFAULT storage_clause] ;
```

Étendre l'espace de stockage d'un datafile avec une limite.

Journalisation ou pas des modifications.

Supprimer un TABLESPACE :

Supprimer logique : **drop tablespace nom_tablespace**

Supprimer logique et physique : **drop tablespace nom_tablespace and datafiles**

Supprimer les objets stockés dans le tablespace : **drop tablespace nom_tablespace including contents**

Suppression des contraintes d'intégrité : **drop tablespace nom_tablespace including contents cascade constraints**

Administrer la sécurité utilisateur

Compte utilisateur de la BD :

Compte utilisateur de la BD :

A chaque création d'un compte utilisateur de BD il faut définir un nom unique et une méthode d'authentification. Les éléments pouvant être attribués par défaut sont : un tablespace par défaut, quotas sur un ou plusieurs tablespaces, un tablespace temporaire, un profil utilisateur, un statut de compte (verrouillé ou expiré), etc. Il ne peut pas être modifié.

Compte prédéfinis SYS et SYSTEM :

Le compte SYS reçoit le rôle DBA. Il est requis pour les opérations de démarrage et d'arrêt, ainsi que pour certaines commandes de maintenance.

Authentification des utilisateurs :

Operating System :

Mode d'authentification qui nécessite le privilège SYSDBA ou SYSOPER (ce qui permet de copier les mots de passe des utilisateurs du dictionnaire de données dans un external password file qui peut être lu par l'instance (même si la BD n'est pas ouverte).

CONNECT / AS [SYSOPER | SYSDBA] ;

Password File Authentication :

Mode d'authentification qui nécessite le privilège SYSDBA ou SYSOPER.

CONNECT username / password AS [SYSOPER | SYSDBA] ;

Password (authentification par la BD Oracle) :

Associer à chaque utilisateur créer un mot de passe qu'il devra saisir lors de chaque connexion.

CREATE USER <name> IDENTIFIED BY ;

Externe :

Oracle délègue la tâche d'authentification à un service externe.

Si l'option de sécurité avancée est autorisée, le service externe peut être un serveur Kerberos, un serveur RADIUS, ou (dans l'environnement Windows), le service d'authentification natif de Windows. Sinon l'authentification se fait par le système d'exploitation.

Global :

Ce mode d'authentification permet d'identifier les utilisateurs via Oracle Internet Directory.

CREATE USER <name> IDENTIFIED GLOBALLY ;

Privilèges :

C'est le droit d'exécuter un type particulier d'instruction SQL ou d'accéder à l'objet d'un autre utilisateur (commande **GRANT** ou **REVOKE**). Il existe deux types de privilèges utilisateur :

Privilèges système :

Autorisent un utilisateur à effectuer certaines opérations de la BD (Ex : tablespace, sessions).

Privilèges objet :

Permettent à un utilisateur d'effectuer une action particulière sur un objet spécifique (Ex : table, vue).

Rôles :

Accorder des privilèges à un rôle, puis accorder ce rôle à chaque utilisateur. → Gestion simplifiée.

Rôles prédéfinis :

Rôle	Privilèges
CONNECT	CREATE SESSION, CREATE TABLE, CREATE VIEW, CREATE SYNONYM, CREATE SEQUENCE, CREATE DATABASE LINK, CREATE CLUSTER, ALTER SESSION
RESOURCE	CREATE TABLE, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TRIGGER, CREATE TYPE, CREATE CLUSTER, CREATE INDEXTYPE, CREATE OPERATOR
SCHEDULER_ADMIN	CREATE ANY JOB, CREATE JOB, EXECUTE ANY CLASS, EXECUTE ANY PROGRAM, MANAGE SCHEDULER
DBA	La plupart des privilèges système et plusieurs autres rôles. Ce rôle ne doit pas être accordé aux utilisateurs qui ne sont pas administrateurs.
SELECT_ CATALOG_ROLE	Pas de privilèges système, mais plus de 1600 privilèges objet sur le dictionnaire de données.

Créer un rôle :

Création d'un rôle SUPERDBA avec mot de passe et intégrant le rôle DBA et des privilèges système :

CREATE ROLE "SUPERDBA"IDENTIFIED BY "sup"; – le mdp sera demandé à l'activation du rôle.

GRANT ALTER ANY INDEXTYPE TO "SUPERDBA"

GRANT ALTER ANY PROCEDURE TO "SUPERDBA"

GRANT ALTER ANY TABLE TO "SUPERDBA"

GRANT "DBA"TO "SUPERDBA"WITH ADMIN OPTION ;

Affectation du rôle SUPERDBA à l'utilisateur STOCK :

GRANT "SUPERDBA" TO STOCK WITH ADMIN OPTION ;

Profils :

Un seul profil est affecté à un utilisateur à un instant donné.

Un profil gère les fonctionnalités de mots de passe (option toujours activée) et contrôle la consommation des ressources (option activée si la valeur de paramètres d'initialisation RESOURCE LIMIT est à TRUE. Sa valeur par défaut est FALSE).

FAILED_LOGIN_ATTEMPTS	Nombre d'échecs de connexion avant le verrouillage du compte	PASSWORD_LIFE_TIME	Durée de vie du mot de passe, en jours, avant expiration
PASSWORD_LOCK_TIME	Nombre de jours pendant lesquels le compte est verrouillé après le nombre déterminé d'échecs de connexion	PASSWORD_GRACE_TIME	Période de grâce, en jours, permettant le changement de mot de passe après la première connexion réussie suite à l'expiration du mot de passe
PASSWORD_REUSE_TIME	Nombre de jours pendant lesquels le mot de passe ne peut pas être réutilisé	PASSWORD_VERIFY_FUNCTION	Fonction PL/SQL qui effectue une vérification de complexité avant l'affectation d'un mot de passe
PASSWORD_REUSE_MAX	Nombre de fois le mot de passe actuel peut être utilisé		

Création d'un profil :

```
CREATE PROFILE <profile_name> LIMIT
[Limit_ressourse> value]
[Limit_passowrd> value] ;
value :={integer | UNLIMITED | DEFAULT} ;
value :={function | null | DEFAULT} pour le paramètre PASSWORD_VERIFY_FUNCTION
```

Sécurité et audit des bases de données oracle

Sécurité de la Base de Données :

La sécurité englobe l'authentification des utilisateurs, le contrôle d'accès et la surveillance des activités suspectes.

Principe du moindre privilège :

Il consiste à n'accorder à un utilisateur que les privilèges dont il a réellement besoin.

Protéger le dictionnaire de données :

Affecter la valeur **FALSE** au paramètre d'initialisation O7 DICTIONARY ACCESSIBILITY.

Empêcher les utilisateurs dotés du privilège système ANY TABLE d'accéder aux tables de base du dictionnaire. La valeur FALSE empêche également l'utilisateur SYS de se connecter sous un autre compte que SYSDBA.

Révoquer les privilèges non nécessaires du rôle PUBLIC :

Le privilège d'exécution sur les packages suivants doit toujours être révoqué de PUBLIC :

- **UTL_SMTP** qui permet l'envoi de messages électroniques arbitraires.
- **UTL_HTTP** qui permet au serveur de la BD de demander et d'extraire des données.
- **UTL_FILE** qui permet l'accès au niveau texte à n'importe quel fichier de l'OS.

Limiter les utilisateurs dotés de privilèges d'administrateur :

- Limiter les types de privilège « système et objet », « SYS : SYSDBA et SYSOPER » et « DBA » (tels que DROP ANY TABLE).
- Répertorier tous les utilisateurs avec le rôle DBA.
- Répertorier les utilisateurs auxquels le privilège SYSDBA ou SYSOPER a été accordé.

Désactiver l'authentification à distance par le système d'exploitation :

Pour désactiver l'authentification à distance, il faut vérifier que la valeur **FALSE** (par défaut) est affectée au paramètre d'initialisation d'instance **REMOTE OS AUTHENT**.

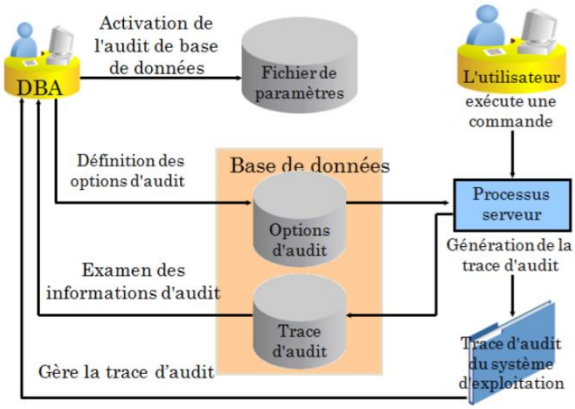
Audit de la Base de Données :

Les outils d'audit intégrés d'Oracle sont les suivants :

- Oracle Standard Auditing appelé Audit de la BD.
- Auditing par Trigger appelé aussi Audit basé sur les valeurs ou sur les données.
- Fine Grained Auditing n'est pas disponible sur toutes les versions (audit détaillé).

Audit standard de la BD :

L'audit se fait sur l'utilisation des privilèges, notamment l'accès aux objets. Il permet d'identifier les commandes exécutées sur une table, mais ne peut pas fournir des informations sur les modifications faites.



Les preuves d'audit sont un ensemble fixe de données.

L'audit est activé via le paramètre AUDIT TRAIL.

NONE : désactive les enregistrements.

OS : active l'audit, les enregistrements sont dans un fichier défini par **AUDIT_FILE_DEST**.

DB : active l'audit, les enregistrements sont dans la table BD défini par **SYS.aud$**.

Pour changer la valeur du paramètre AUDIT_TRAIL :

- Si c'est un pfile, alors il faut d'éditer le contenu du fichier.
- Si c'est un spfile, alors il faut exécuter **ALTER system set audit trail=DB scope=spfile ;**

L'audit via l'**OS** est préféré lorsqu'on veut auditer plusieurs BDs et dans une même destination.

L'audit via la BD permet de visualiser les résultats par des requêtes simples contre les vues du dictionnaire relatives à l'audit. Il faut aussi auditer les actions de la table SYS.AUD$ par **audit all on SYS.AUD$ by access**.

Niveaux d'audit :

Oracle permet un Auditing standard sur 4 niveaux :

- Auditing de commandes (LDD).
- Audit de privilèges (systèmes).
- Audit d'objets de schéma (privilèges objets).
- Audit de connexion à la BD.

Audit de privilèges ou de commandes

```
AUDIT {commande | privilège_système}
      [ , {commande | privilège_système} ] ...
      [BY user [ , user ] ... ]
      [BY {SESSION | ACCESS} ]
      [WHENEVER [NOT] SUCCESSFUL]
```

Audit d'objets

```
AUDIT commande [ , commande ] ...
      ON { [ schema. ] objet | DEFAULT }
      [BY {SESSION | ACCESS} ]
      [WHENEVER [NOT] SUCCESSFUL]
```

BY SESSION : n'insérer par session qu'un enregistrement par objet de BD.

BY ACCESS : insérer un enregistrement dans la trace à chaque action soumise.

WHENEVER : les audits ne doivent être exécutés que lorsque l'exécution de commandes SQL est terminée, réussie ou non.

Vues d'audit :

DBA_AUDIT_TRAIL : Toutes les entrées de la trace d'audit.

DBA_AUDIT_OBJECT : Enregistrements des objets de schémas.

DBA_AUDIT_SESSION : Toutes les entrées de connexion et de déconnexion.

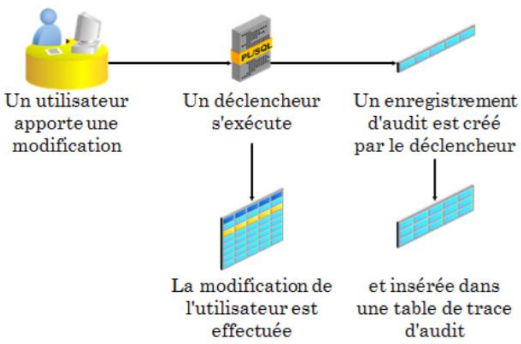
DBA_AUDIT_STATEMENT : Enregistrements d'audit des instructions.

Audit basé sur les données (avec les triggers) :

L'audit se fait sur les données modifiées par les instructions LMD. Les preuves d'audit sont définies par l'administrateur.

DML trigger : permet d'enregistrer les valeurs de toutes les modifications de la BD. (Tâche impossible pour l'audit intégré d'Oracle).

System trigger : auditing de toute action de création, de suppression ou de connexion à la BD. (Inutile puisqu'elles seront auditées en utilisant l'audit intégré d'Oracle).




```
CREATE [OR REPLACE] TRIGGER nom_trigger
{BEFORE|AFTER|INSTEAD OF}
{INSERT|UPDATE [ OF nom_colonne [, nom_colonneN]] |DELETE}
OR {INSERT|UPDATE [OF nom_colonne [, nom_colonneN ]] |DELETE}
REFERENCING {[OLD [AS] ancien] | [NEW [AS] nouveau]}
ON nom_table
[FOR EACH ROW]
[WHEN] (condition)
DECLARE
/* déclaration */
BEGIN
/* traitement */
[EXCEPTION]
END ;
```

Audit détaillé « Fine Grained Auditing » (FGA) :

L'audit se fait sur les instructions SQL. Les preuves d'audit sont un ensemble fixe de données incluant l'instruction SQL. Il permet de définir des conditions fines pour l'auditing :

- Surveille l'accès aux données en fonction du contenu.
- Peut être lié à une table ou à une vue.
- Peut exécuter une procédure PL/SQL.
- Est administré via le package DBMS FGA.

Une stratégie d'audit détaillé est constituée de 2 parties :

- Les critères d'audit : définit la condition à vérifier pour que l'instruction soit auditée.
- L'action d'audit : le nom d'une procédure à exécuter lorsque la condition est vérifiée.

Elle est créée via la procédure **ADD_POLICY** du package **DBMS_FGA**.

DBMS FGA inclut les sous-programmes suivants :

- **ADD_POLICY** : crée une stratégie d'audit à l'aide du prédicat fourni en tant que condition d'audit.
- **DROP_POLICY** : supprime une stratégie d'audit.
- **ENABLE_POLICY** : active une stratégie d'audit.
- **DISABLE_POLICY** : désactive une stratégie d'audit.

Vues d'audit :

DBA_FGA_AUDIT_TRAIL : Tous les événements d'audit détaillé.

ALL_AUDIT_POLICIES : Toutes les stratégies d'audit détaillé pour les objets auxquels l'utilisateur actuel peut accéder.

DBA_AUDIT_POLICIES : Toutes les stratégies d'audit détaillé dans la BD.

USER_AUDIT_POLICIES : Toutes les stratégies d'audit détaillé pour les objets du schéma de l'utilisateur actuel.

Règles d'audit détaillé :

- Pour auditer toutes les instructions, il faut utiliser une condition NULL.
- Une erreur est générée lors de l'ajout d'une stratégie existante.
- La table ou la vue auditée doit déjà exister lorsqu'il y a création de la stratégie.
- Si la colonne d'audit n'existe pas dans la table, aucune ligne n'est auditée.

Déplacement des données

Il existe plusieurs méthodes pour charger des données dans les tables d'une BD Oracle :

- Utilitaire d'export / import ORACLE DATA PUMP.
- SQL*Loader.

Oracle Data Pump :

C'est un utilitaire (côté serveur) de déplacement de données et de métadonnées de masse de manière très rapide entre des BD Oracle (deux utilitaires d'import et d'export des données).

Pour lancer Data Pump, il faut créer un Directory Oracle où il y aura stockage des exports.

Data Pump peut être appelé via :

- Enterprise Manager (EM) Database Control.
- Les binaires EXPDP et IMPDP situés dans le dossier bin d'ORACLE HOME.
- Le package SYS.DBMS DATAPUMP.

Il existe 4 modes d'export/import avec Data Pump :

- Export / Import COMPLET : opération demandée par le paramètre FULL.
- Export / Import de SCHEMA (metadata) : permet l'imp/exp d'un ou plusieurs schémas de la BD (mode par défaut).
- Export / Import de TABLE : mode commandé par le paramètre TABLES qui permet de sélectionner des tables à exporter à partir d'un schéma.
- Export / Import de TABLESPACE : permet d'exporter les tables d'au moins un espace disque logique (Seules les tables seront exportées, et non les espaces disque logiques).

3 types de fichiers sont générés avec Data Pump :

Fichiers SQL : contiennent les instructions LDD de la création des objets de la BD.

Fichiers DUMP : contiennent les données exportées.

Fichiers LOG : contiennent le journal d'historique de l'exécution du job (export ou import).

EXPDP/ IMPDP :

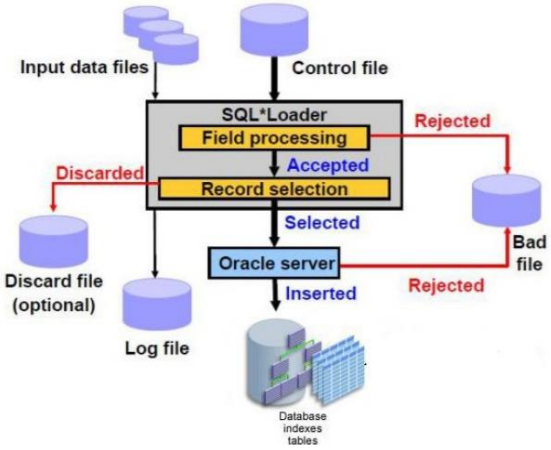
Privilèges nécessaires :

- EXP FULL DATABASE et IMP FULL DATABASE : permet d'exporter l'intégralité d'une BD, un Tablespace, un schéma autre que le sien ou une table située dans un autre schéma.
- Tous les utilisateurs qui possèdent le rôle DBA ont la possibilité d'effectuer un import/ export d'une BD.

SQL*Loader :

C'est un utilitaire qui charge les données de fichiers externes dans des tables d'une BD Oracle.

Il dispose d'un puissant moteur d'analyse (parse) des données, qui ne limite que très peu le format des données du fichier.



Log File :

C'est un fichier qui enregistre les activités SQL Loader durant un chargement de données :

- Les noms des fichiers CONTROL FILE, BAD FILE, DISCARD FILE, Input Data File.
- Les champs et types de données qui ont été chargées.
- Messages d'erreurs sur les enregistrements non chargés.
- Le nombre d'enregistrements lus dans le fichier de données ou rejetés en raison d'erreurs ou de critères de sélection.
- Le temps de chargement.

Bad File :

SQL Loader enregistre les erreurs (de lecture ou de chargement d'un enregistrement) dans un fichier BAD FILE : enregistrement non conforme au format dans le fichier de contrôle, violations de contraintes d'intégrité, tablespace plein, etc.

Discard File :

Les enregistrements qui ne répondent pas aux conditions de sélection spécifiées dans le Control File sont rejetés et écrits dans le fichier DISCARD FILE.

Control File :

Le fichier de contrôle indique à SQL*Loader :

- L'emplacement des fichiers Bad File, Discard File et Log File.
- Les détails de configuration : Gestion de la mémoire Rejet des enregistrements.
- L'emplacement, la structure, types, longueurs, précisions des données à charger.
- Les noms de tables à charger.
- La correspondance entre les champs des données et les colonnes des tables de la BD.
- Les critères de sélection des enregistrements à insérer dans les tables de la BD.
- Formatage des enregistrements de données : si le format est délimité, etc.

Exemple :

- Soit le fichier « Emp.dat » contenant :
  10001,"Scott Tiger", 1000, 40
  10002,"Frank Naude", 500, 20
- Pour charger le fichier emp.dat dans la table « emp » du schéma HR, le control file doit contenir les informations suivantes :

```
load data
infile 'c :\Emp.dat'
into table emp    – { INSERT | REPLACE | TRUNCATE | APPEND }
fields terminated by "," optionally enclosed by '"'
( empno, empname, sal, deptno )
```

Pour charger les données on doit exécuter la commande suivante :

```
SQL> host
C :\> Sqlldr user/password control=<control_file>.ctl
log=<log_file>.log bad=<bad_file>.bad discard=<discard_file>.dsc
```