Le MCD est utilisé pour définir les contraintes qui seront utilisées pour garantir la validité des données stockées dans les tables de la base.

Oracle Database 10g supporte les types de contraintes suivantes :

* **NOT NULL :** Interdit les valeurs nulles (entrées vides) dans la colonne de la table.
* **UNIQUE :** Interdit la duplication d’une valeur dans une ou plusieurs colonnes.
* **PRIMARY KEY :** Interdit la duplication et la valeur nulle dans une ou plusieurs colonnes.
* **FOREIGN KEY :** Exige que chaque valeur dans une colonne ou un jeu de colonnes corresponde avec une valeur des tables relationnelles UNIQUE ou PRIMARY KEY. Les contraintes d’intégrité FOREIGN KEY définissent également les actions d’intégrité référentielle qui indique ce qu’Oracle devrait faire sur des données dépendantes si les données sur lesquelles elles font références sont modifiées.
* **CHECK :** Interdit les valeurs qui ne satisfont pas l’expression logique de la contrainte.

Structured Query Language (SQL) est l’ensemble des requêtes permettant à tous les utilisateurs et applications d’accéder aux données de la base de données.

Un administrateur de base de données Oracle (DBA) est responsable de l’installation du logiciel Oracle et de la création de la base de données.

Approche prioritaire pour concevoir, mettre en application et maintenir une base de données Oracle :

1. Evaluer le matériel du serveur de base de données

2. Installer le logiciel Oracle

3. Concevoir la base de données

4. Créer et ouvrir la base de données

5. Sauvegarder la base de données

6. Inscrire les utilisateurs système

7. Mettre en application la conception de la base de données

8. Restaurer en cas d’erreur de la base de données

9. Surveiller les performances de la base de données

Les fichiers qui constituent une base de données Oracle sont organisés de cette façon :

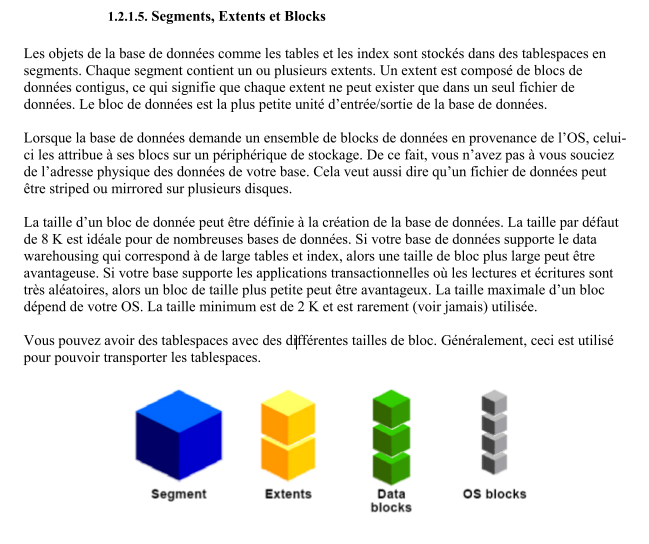
* **Control files :** Contiennent les données à propos de la base de données elle-même, appelés les méta-données. Ces fichiers sont critiques pour la base de données. Sans eux, vous ne pourrez pas ouvrir les fichiers de données et ainsi accéder aux données de la base de données.
* **Data files :** Contiennent les données de la base de données.
* **Online redo log files :** Permettent la récupération de l’instance de la base de données. Si la base de données crash et ne perd aucun fichier de données, l’instance peut récupérer la base de données grâce aux informations contenues dans ces fichiers.

Il existe d’autres fichiers qui ne font pas partie intégrante de la base de données mais ils sont importants pour le fonctionnement de la base :

* **Parameter file :** Utilisé pour définir comment l’instance sera configurée au démarrage.
* **Password file :** Permet aux utilisateurs de se connecter à distance à la base et d’effectuer des tâches administratives.
* **Archive log files :** Contiennent un historique des fichiers de redo générés par l’instance. Ces fichiers sont utilisés pour la récupération de la base de données ; en utilisant ces fichiers et une sauvegarde de la base de données, vous pouvez récupérer des fichiers perdus.

Une base de données est divisée en unités de stockage logique appelées tablespaces, qui peuvent être utilisés pour regrouper des structures logiques ensemble. Chaque base de données est divisée de façon logique en un ou plusieurs tablespaces. Un ou plusieurs fichiers de données sont explicitement créés pour chaque tablespace afin de stocker les données physiques de toutes les structures logiques dans un tablespace.

Remarque : Vous pouvez également créer des tablespaces de type bigfile, qui sont constitué d’un seul, mais gros (plus de 4 Go) fichier de données. Le type par défaut est smallfile, les tablespaces de ce type peuvent contenir plusieurs fichiers de données mais pas aussi gros.



Un serveur de base de données Oracle est composé d’une base de données Oracle et d’une instance Oracle. Une instance Oracle est composée de plusieurs structures mémoire connu sous le nom de System Global Area (SGA) et des processus d’arrière-plan.

L’instance n’existe pas avant son démarrage. Lorsque l’instance est démarrée, le fichier d’initialisation est lu et l’instance est configurée selon les paramètres du fichier d’initialisation.

Une fois l’instance démarrée et la base de données ouverte, les utilisateurs peuvent accéder à la base de données.

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

* System Global Area (SGA) : Partagée par tous les processus serveurs et d’arrière plan.
* Program Global Area (PGA) : Privée pour chaque processus serveur et d’arrière plan ; il y a une PGA pour chaque processus.

La SGA est une zone mémoire partagée qui contient des données et des informations de contrôle de l’instance.

La SGA est constituée des éléments suivants :

* Database buffer cache : Garde en cache les blocs de données récupérés de la base de données.
* Redo log buffer : Garde en cache les informations de redo (utilisés pour récupérer l’instance) jusqu’à ce qu’elles soient écrites dans le fichier de redo log stocké sur le disque.
* Shared pool : Garde en cache différentes informations pouvant être partagées par plusieurs utilisateurs.
* Large pool : Zone mémoire optionnelle utilisée pour les grandes demandes d’entrées/sorties.
* Java pool : Utilisé pour toutes les sessions spécifiques à du code Java et aux données de la Machine Virtuel Java (JVM).
* Streams pool : Utilisé par Oracle Streams.

Lorsque vous démarrez l’instance en utilisant Enterprise Manager or SQL\*Plus, la mémoire est allouée pour la SGA. Avec une configuration dynamique de la SGA, la taille du Database buffer cache, du Shared pool, du Large pool, du Java pool et du Streams pool change sans avoir besoin d’éteindre l’instance.

La PGA est une zone mémoire qui contient les données et contrôle les informations de chaque processus serveur. Un processus serveur est un processus qui reçoit les requêtes du client. Chaque processus serveur a sa propre zone PGA privée qui est créée lorsque le processus serveur est lancé.

L’accès à cette zone est exclusivement réservé à ce processus serveur, et il n’est lu et écrit que par le code Oracle agissant en son nom. La quantité de mémoire PGA utilisé et son contenu dépendent de la configuration de l’instance, en mode partagé ou non.

Généralement, la PGA contient ceci :

* Private SQL area : Contient les données comme les informations de correspondances et le temps d’exécution des structures mémoires. Chaque session faisant des requêtes SQL a une zone privée SQL.
* Session memory : Mémoire allouée aux variables de sessions et aux autres informations relatives à la session.

Les processus d’arrière-plan les plus communs sont :

* System monitor (SMON) : Exécute la récupération de l’instance après le crash de celle-ci.
* Process monitor (PMON) : Exécute le nettoyage des processus lorsqu’un processus utilisateur échoue.
* Database Writer (DBWn) : Ecrit les blocs modifiés du database buffer cache vers les fichiers stockés sur le disque.
* Checkpoint (CKPT) : Envoi un signal au processus DBWn lors d’un checkpoint et met à jour tous les fichiers de données et les fichiers de contrôle de la base de données afin d’indiquer le checkpoint le plus récent.
* Log writer (LGWR) : Ecrit les entrées de redo log sur le disque.
* Archiver (ARCn) : Copie les fichiers de redo log dans un espace d’archivage quand les fichiers de log sont remplis ou lors de l’exécution d’un log switch.

Le dictionnaire de données est un ensemble de tables et de vues qui est utilisé en lecture seule comme référence pour une base de données particulière. Le dictionnaire de données stocke des informations sures :

* La structure physique et logique de la base de données.
* Les utilisateurs valides de la base de données.
* Les informations sur les contraintes d’intégrité.
* L’espace alloué pour un objet d’un schéma et l’espace utilisé.

Le dictionnaire de données est créé lors de la création de la base de données et est automatiquement mis à jour lorsque la structure de la base de données est mise à jour. Enterprise Manager trouve les informations à propos des objets de la base dans le dictionnaire de données. Vous pouvez également interroger les tables du dictionnaire de données pour obtenir des informations. Enterprise Manager le fait pour vous et présente les informations dans un format simple d’utilisation. La vue DICTIONNARY contient les descriptions des tables et des vues du dictionnaire de données.

Ces tables et ces vues ont généralement un de ces préfixes :

* USER : Information sur les objets posséder par l’utilisateur courant.
* ALL : Information sur les objets accessibles par l’utilisateur courant.
* DBA : Information sur tous les objets de la base de données.

Oracle Database 10g est livré avec Oracle Enterprise Manager’s Database Control. Database Control est une console de contrôle par le web que l’administrateur de la base de données peut utiliser pour :

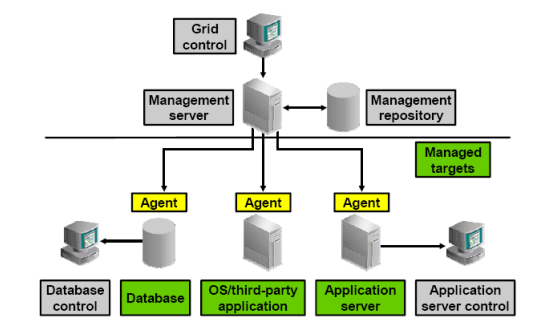
* Surveiller les performances.
* Gérer les alertes.
* Contrôler la maintenance des agents et des conseillers.
* Administrer les objets des utilisateurs et de la base de données.
* Sauvegarder et récupérer la base de données.
* Gérer le stockage des données.
* Et bien plus.

Chaque base de données Oracle 10g que vous créez aura son propre Database Control.

Les capacités de Database Control peuvent être étendus et intégrés avec le reste de votre système en utilisant Oracle Enterprise Manager’s Grid Control. L’architecture du Framework Grid Control permet un haut niveau de flexibilité et de fonctionnalité. Vous pouvez facilement personnaliser Enterprise Manager pour suivre la gestion et l’administration des besoins de votre environnement.

La configuration typique d’Enterprise Manager Framework contient les zones fonctionnelles suivantes :

* Les différentes cibles gérées
* La gestion des services
* Une console d’administration globale via le web
* Les outils Database control et Application server control



**Optimal Flexible Architecture**

OFA est une méthode permettant de configurer la base de données Oracle ainsi que d’autres bases de données. OFA tire profit des possibilités de l’OS et des disques pour faciliter la configuration. Ceci permet une flexibilité maximum pour la croissance et les performances des bases de données. Les méthodes décrites ici sont les fonctionnalités de base d’OFA.

OFA est utilisé pour :

* Organiser de grandes quantités de logiciel et de données sur le disque, pour éviter une baisse des performances.
* Faciliter les tâches administratives de routines comme la maintenance des logiciels et la sauvegarde des données, qui sont souvent vulnérable à la corruption de données.
* Faciliter le changement entre plusieurs bases de données Oracle.
* Contrôler et administrer la croissance des bases de données.
* Aider à éliminer la fragmentation de l’espace libre du dictionnaire de données, isoler les autres fragmentations et minimiser la corruption des ressources.

Le noyau d’OFA est un schéma nommé, qui donne une norme standard à appliquer aux points de montage (qui correspondent souvent aux disques physiques), aux répertoires, et aux fichiers eux-mêmes.

Voici une liste de sous répertoires pour les fichiers d’administration :

* adhoc : les scripts Ad hoc SQL pour une base de données particulière
* arch : les fichiers de redo log archivés
* adump : les fichiers d’audit (Indiquez dans le fichier de paramètre le chemin vers le dossier adump dans la variable AUDIT\_FILE\_DEST. Vider périodiquement ce sous répertoire.)
* Bdump : Les fichiers de trace des processus d’arrière-plan
* Cdump : Core dump files
* Create : Programmes utilisés pour créer la base de données
* Exp : Les fichiers d’exportation de la base de données
* Logbook : Les fichiers enregistrant le statu et l’historique de la base de données
* Pfile : Les fichiers de paramètre de l’instance
* udump : Les fichiers utilisateurs de trace SQL

Syntaxe des fichiers nommés : La convention de nommage suivante permet d’identifier facilement les fichiers :

* Control files : /pm/q/d/controln.ctl
* Redo log files : /pm/q/d/redon.log
* Data files : /pm/q/d/tn.dbf

Les variables utilisées dans ces fichiers sont :

* Pm : Un point de montage nommé comme décrit auparavant
* q : Chaîne de caractère permettant de distinguer les données Oracle des autres fichiers (généralement appelé ORACLE ou oradata)
* d : La valeur du paramètre d’initialisation DB\_NAME (le nom de la base de données)
* t : Le nom d’un tablespace Oracle
* n : Un nombre sur deux chiffres
* ORACLE\_BASE : Indique le chemin du répertoire Oracle pour la structure OFA. Son utilisation est optionnelle, mais elle peut faciliter les installations et mises à jour futures. Il s’agit du chemin du répertoire, par exemple : /u01/app/oracle
* ORACLE\_HOME : Indique le répertoire contenant les logiciels Oracle. Il s’agit du chemin du répertoire, par exemple : $ORACLE\_BASE/product/10.1.0
* ORACLE\_SID : Le nom initial de l’instance (par défaut ORCL). Il s’agit d’une chaîne de caractères et de nombres qui doit commencer par une lettre. Oracle recommande un maximum de 8 caractères.
* NLS\_LANG : Indique le paramètre NLS initial pour une session. Il est de la forme language\_territory.character. Par exemple : AMERICAN\_DENMARK.WE8MSWIN1252

**Oracle Database Enterprise Edition** offre aux leaders du marché une fiabilité et une évolutivité sur les configurations des systèmes simple ou en cluster. Il fournit les dispositifs les plus complets pour les transactions en ligne.

**Oracle Standard Edition** est pour les petites et moyennes entreprises, ou les applications départementales qui requièrent toute la puissance, la fiabilité et la sécurité d’Oracle sans les options de la version Enterprise.

Les options suivantes ne sont pas disponibles avec la version Standard d’Oracle :

* Oracle Data Guard: Ensemble de services qui permettent de créer, gérer et surveiller une ou plusieurs bases de données.
* Oracle Advanced Security, Oracle Label Security, Enterprise User Security, Virtual Private Database, N-tier authentication, et Fine Grained Auditing: Ces options améliorent la sécurité d’une base de données Oracle 10g.
* Oracle Partitioning, Oracle OLAP, Export Transportable Tablespace, et autres utilitaires qui supportent l’Enterprise Data Warehouses.

L’installation Custom permet de sélectionner les composants que l’on veut installer.

L’Application Serveur iSQL\*Plus doit être lancée avant de pouvoir lancer une session iSQL\*Plus.

L’utilisation d’une ligne de commande et d’un service Windows sont nécessaire pour démarrer et arrêter iSQL\*Plus sous Windows. L’Application Serveur iSQL\*Plus est démarrée par défaut lors de l’installation de la base de données Oracle.

Pour démarrer et arrêter l’Application Serveur iSQL\*Plus sous Unix (Linux) :

* Démarrer une session en terminal
* Entrer cette commande pour démarrer : isqlplus start
* Entrer cette commande pour arrêter : isqlplus stop

Remarque : Vous devrez naviguer dans le répertoire $ORACLE\_HOME/bin si ce répertoire n’est pas présent dans votre PATH.

Il y a trois composants majeurs dans la gestion du Framework Oracle Database 10g :

* La base de données et l’instance.
* Un listener permettant les connexions à la base de données.
* Une interface de gestion. Il peut s’agir d’un agent de gestion connectant le serveur à Oracle Enterprise Manager Grid Control, ou à Oracle Enterprise Manager Database Control.

Chacun de ces composants doit être démarré avant de pouvoir utiliser ses services et doit être arrêté proprement lorsque le serveur hébergeant la base de données Oracle 10g est arrêté.

Le premier composant devant être démarré est l’interface de gestion. Ensuite l’interface de gestion pourra être utilisée pour démarrer les autres composants.

Pour les bases de données non connectées au Framework Grid Control, Oracle fournit une console de gestion individuelle appelée Database Control. Chaque base de données gérée avec Database Control possède une installation séparée de Database Control, et chaque Database Control ne peut gérer qu’une seule base de données. Database Control requiert qu’un processus dbconsole soit démarré avant l’utilisation.

Pour démarrer le processus dbconsole :

***emctl start dbconsole***

Pour arrêter le processus dbconsole :

***emctl stop dbconsole***

Pour voir le status du processus dbconsole :

***emctl status dbconsole***

Remarque : Vous devrez vous placer dans le répertoire $ORACLE\_HOME/bin si ce répertoire n’est pas présent dans votre PATH.

Database Control utilise un processus agent du côté serveur. Ce processus agent démarre et s’arrête automatiquement lorsque le processus dbconsole est démarré ou arrêté.

Une instance est démarrée en mode NOMOUNT seulement durant la création de la base de données ou la recréation des fichiers contrôles.

Démarrer une instance comprend les points suivants :

* Lecture du fichier d’initialisation situé dans $ORACLE\_HOME/dbs dans l’ordre suivant :
* En premier le spfileSID.ora
* S’il n’est pas trouvé, spfile.ora
* S’il n’est pas trouvé, intiSID.ora

Spécifier le paramètre PFILE avec la commande STARTUP remplace celui par défaut.

* Allocation de la SGA
* Démarrage des processus d’arrière-plan
* Ouverture du fichier alertSID.log et des fichiers de trace.

Montée une base de données comprend les étapes suivantes :

* Associer la base de données avec une instance démarrée précédemment.
* Situer et ouvrir les fichiers de contrôles spécifiés dans le fichier de paramètres.
* Lire les fichiers de contrôles pour obtenir les noms et statuts des fichiers de données et des fichiers de redo. Cependant, aucune vérification n’est faite concernant l’existence des fichiers de données et des fichiers de redo logs à ce moment.

**SYSOPER** : Il s’agit d’un rôle d’administration spécial de la base de données qui permet à l’administrateur de la base de données d’effectuer les actions STARTUP, SHUTDOWN, ALTER DATABASE OPEN/MOUNT, ALTER DATABASE BACKUP, ARCHIVE LOG, et RECOVER, et inclus le privilège RESTRICTED SESSION. Lorsque vous vous connectez en tant que SYSOPER, vous êtes dans le schéma public.

**SYSDBA :** Il s’agit d’un rôle d’administration spécial de la base de données qui contient tous les privilèges système des privilèges système ADMIN OPTION et SYSOPER. SYSDBA peut également utiliser la clause CREATE DATABASE et la récupération partielle. Lorsque vous vous connectez en tant que SYSDBA, vous vous trouvez dans le schéma de l’utilisateur SYS.

Chaque base de données possède un alert\_sid.log. Ce fichier est sur le serveur avec la base de données et est stocké dans le répertoire spécifié par le paramètre d’initialisation background\_dump\_dest. Le fichier d’alerte d’une base de données est un fichier log chronologique de messages et d’erreurs, incluant :

* Toutes les erreurs internes (ORA-600), les erreurs dues aux blocs corrompus (ORA-1578), et les erreurs de deadlock (ORA-60) qui surviennent.
* Les opérations administratives, comme les instructions SQL CREATE, ALTER, DROP DATABASE, TABLESPACE, ROLLBACK SEGMENT et celles d’Enterprise Manager ou de SQL\*Plus STARTUP, SHUTDOWN, ARCHIVE LOG et RECOVER.
* Plusieurs messages et erreurs à propos des fonctions du serveur partagé et des dispatchers
* Les erreurs durant le rafraîchissement automatique d’une vue matérialisée.

Les bases de données, les tablespaces et les fichiers de données sont étroitement liés, cependant il y a une différence importante :

* Une base de données Oracle est composée d'une ou plusieurs unités logiques de stockage appelées tablespaces. Ils stockent collectivement toutes les données de la base.
* Chaque tablespace d’une base de données comporte un ou plusieurs fichiers appelés fichier de données. Lesquels sont des structures physiques conformes au système d’exploitation sur lequel le serveur Oracle fonctionne.
* Les données de la base sont stockées dans les fichiers de données, qui constituent chaque tablespace de la base. Par exemple : la plus simple base de données Oracle aurait un tablespace et un fichier de données. Une autre base de données peut avoir trois tablespaces, comprenant chacun deux fichiers de données chacun (donc un total de six fichiers de données). Une seule base de données peut avoir au maximum 65 535 fichiers de données.

Les tablespaces assignent de l’espace aux extents. Les tablespaces peuvent être crées en utilisant une des deux méthodes suivantes pour gérer l’espace libre et utilisé.

* **Gestion locale des tablespaces :**

Les extents sont gérés dans le tablespace via des bitmaps. Chaque bit, dans un bitmap, correspond à un bloc ou à un groupe de bloc. Lorsqu’un extent est alloué ou libre pour être réutilisé, le serveur Oracle change la valeur du bitmap pour mettre à jour le nouveau statut des blocs.

* **Gestion des tablespace par le dictionnaire des données :**

Les extents sont gérés par le dictionnaire des données. Le serveur Oracle met à jour les tables appropriées dans le dictionnaire des données dès qu’un extent est alloué ou désalloué. Pour une compatibilité avec les versions précédentes, vous devrez utiliser la gestion locale pour tous les tablespaces.

Les extents d’un tablespace géré localement peut-être alloués de deux manières :

* Automatique : appelé aussi auto allocation, spécifie que la taille des extents dans le tablespace est gérée par le système. Vous ne pouvez pas spécifier la taille des extents et vous ne pouvez pas spécifier le paramètre automatique pour un tablespace temporaire.
* Uniforme : Les extends d’un tablespaces géré de cette manière ont une taille fixe que vous spécifiée. La taille par défaut est de 1 Mo. Tout les extents du tablespace temporaire ont une taille uniforme, cette option est facultative pour un tablespace temporaire. Vous ne pouvez pas spécifier cette option pour un tablespace undo.

Les tablespaces suivant sont créé quand vous pré configurez la base de données :

* SYSTEM : le tablespace SYSTEM est utilisé par le serveur Oracle pour gérer la base de données. Il contient le dictionnaire des données et des informations administratives sur la base. Son contenu se trouve dans le schéma de SYS et ne peut être vu que celui-ci ou tout autre administrateur ayant les privilèges requis.
* SYSAUX : c’est un tablespace auxiliaire au tablespace SYSTEM. Quelques composants et produits qui utilisaient le tablespace SYSTEM, dans les versions précédentes d’Oracle, utilise maintenant le tablespace SYSAUX. Chaque base de donnée Oracle 10g doit posséder un tablespace SYSAUX.
* TEMP : Ce tablespace est utilisé pour stocker des tables temporaires et des indexes. Il est utilisé par exemple pour des opérations de tris. Chaque base de données devrait avoir un tablespace temporaire qui est assigné aux utilisateurs comme leur tablespace temporaire. Dans la pré-configuration de la base, le tablespace TEMP est considéré comme le tablespace temporaire par défaut. Dans ce cas, si aucun tablespace temporaire n’est spécifié à un utilisateur lors de sa création, Oracle lui assignera automatiquement celui-ci.
* UNDOTBS1 : c’est le tablespace undo utilisé par le serveur Oracle pour stocker des entée de undo. Chaque base de données doit avoir un tablespace UNDO c’est pour cela qu’il est créé lors de la création de la base de données.
* USERS : Ce tablespace est utilisé pour stocker les objets et les données des utilisateurs. C’est le tablespace par défaut, pour une base de données pré configurée, pour tous les objets créés par un utilisateur non system. Pour les utilisateurs SYS et SYSTEM, le tablespace par défaut est le tablespace SYSTEM.
* EXEMPLE : Ce tablespace contient des exemples de schéma qui peuvent être installés lors de la création de la base de données.

Grâce au menu Action, vous pouvez effectuer plusieurs taches sur vos tablespaces. Sélectionnez un

tablespace et une action à réaliser :

* Add Datafile : Ajoute un fichier de donnée au tablespace, permettant ainsi d’augmenter sa taille.
* Create Like : Crée un nouveaux tablespace en prenant celui sélectionné comme model.
* Generate DDL : Affiche l’instruction SQL de la création du tablespace. Vous pouvez copier/coller l’ordre SQL dans un fichier texte pour l’utiliser comme script.
* Make Locally Managed : Si l’espace du tablespace est géré actuellement par le dictionnaire des données, la gestion se fera localement.
* Make Readonly : Arrête toute écriture sur le tablespace. Vous pouvez terminer les transactions en cours mais aucune autre nouvelle ne sera permise comme toutes autres taches d’écriture.
* Make Writable : Les ordres DML seront désormais possibles sur le tablespace.
* Place Online : Si le tablespace est actuellement offline, celui-ci est mis online.
* Reorganize : Cela démarre l’outil de réorganisation, que vous pouvez utiliser pour déplacer des objets au sein du tablespace pour récupérer de l’espace qui n’aurait pas pu être utilisé. Cette tache ne peut être réalisée quand il y a beaucoup d’activité sur les objets du tablesapce.
* Show Dependencies : affiche les objets qui dépendent de ce tablespace ou ceux dont le tablespace dépend.
* Run Segment Advisor : L’outil Segment Advisor vous aide à déterminer si un objet a de l’espace disponible pour s’agrandir. Au niveau du tablespace, les conseils donnés s’appliquent à tous ses segments.
* Take Offline : Si un tablespace est actuellement online, cette action permet de le rendre indisponible. Le tablespace n’est pas supprimé, juste indisponible.

Dans certains cas, chaque utilisateur a son propre compte de base de données. Dans d'autres, plusieurs utilisateurs partagent un compte commun pour se connecter à la base de données. Indépendamment de la manière dont vous organisez votre base de données, chaque compte utilisateur aura :

* Un nom unique : le nom ne peut pas excéder 30 caractères, ne peut pas contenir de caractère spécial et doit commencer par une lettre.
* Une méthode d’authentification : la méthode d’authentification la plus utilisée est le mot de passe, cependant Oracle 10g supporte plusieurs autres méthodes comme la biométrie ou l’utilisation de certificats.
* Un tablespace par défaut : il s’agit d’un endroit où seront stockés les objets de l’utilisateur si il ne précise pas explicitement où il veut stocker ses objets. Cependant, le fait qu’un utilisateur ait un tablespace par défaut pour stocker ses objets n’implique pas forcément qu’il ait les privilèges suffisants ou un quota d’espace sur celui-ci pour créer des objets. Ces deux conditions sont accordées séparément.
* Un tablespace temporaire : il s’agit d’un endroit où un utilisateur peut créer des objets temporaires comme les tris et les tables temporaires.
* Un profil utilisateur : Il s’agit d’un ensemble de restrictions sur les ressources et les mots de passe assignées à un utilisateur.

Les profiles ne peuvent pas imposer de limitation au niveau des ressources pour un utilisateur tant que le paramètre d’initialisation RESSOURCES\_LIMIT n’est pas à TRUE. Ce paramètre est par défaut initialisé à FALSE, donc les limitations de ressources sont ignorées. Les profils permettent à l'administrateur de contrôler les ressources systèmes suivantes :

* CPU : les ressources CPU peuvent être limitées « per-session » (par session) ou « per-call » (par appel). Une limitation de CPU/session de 1000 signifie que si n’importe quelle session configurée avec ce profil consomme plus de 10 secondes de temps processeur (ce paramètre est exprimé en centième de seconde), alors un message d’erreur est retourné et la session fermée :
* ORA-02392: exceeded session limit on CPU usage, you are being logged off.
* Une limitation par appel effectue le même contrôle, mais s’applique pas à l’ensemble de la session, il empêche que n’importe quelle commande de consommer trop de temps CPU. Ce paramètre est limité et si l'utilisateur dépasse cette valeur, la commande échoue et l'utilisateur reçoit le message d'erreur suivant : ORA-02393: exceeded call limit on CPU usage
* Réseaux/Mémoire: chaque session de base de données consomme des ressources mémoires systèmes et des ressources réseaux (si celle-ci n’est pas locale) :
* Connect Time : Temps en minutes pendant lequel les utilisateurs peuvent rester connectés avant d'être automatiquement déconnectés.
* Idle Time : Temps en minute avant qu’une session utilisateur soit automatiquement fermée par le serveur s’il n’y a pas d’échanges de données. Le temps d’inactivité est calculé seulement au niveau du processus serveur. Il ne prend pas en compte l’activité d’une application. La limite IDLE\_TIME n’est tient pas compte du temps d’exécution de longues requêtes ou autres opérations.
* Concurrent session : Nombre de sessions pouvant être ouvertes simultanément en utilisant le même compte utilisateur.
* Private SGA : Limite l’espace utilisé dans la SGA pour trier, ou encore réorganiser des bitmaps. Cette restriction est effective seulement si la session utilise un serveur partagé.
* Disk I/O : Limite le volume de données qu’un utilisateur peut lire soit pour la session, soit pour un appel (« per session » et « per call »). Les paramètres Reads/Session et Reads/Call mettent en place des restrictions sur le nombre total d’accès, que ce soit pour la mémoire ou pour le disque. Cela permet de s’assurer qu’aucune requête n’utilise de manière intensive les entrées/sorties au niveau de la mémoire et des disques. Les profils peuvent aussi être configurés avec les limites « composées ». Les limites composées se basent sur la combinaison de CPU/session, reads/session, connect time, et de la SGA privée.

4.2.2. Authentification des utilisateurs

L’authentification consiste à vérifier l’identité de quelqu’un (un utilisateur, un dispositif ou une autre entité) qui veut utiliser des données, des ressources, ou des applications. Le fait de valider l’identité établit une relation de confiance pour d’autres interactions. L’authentification permet aussi l’imputabilité en rendant possible la liaison entre les accès et les actions spécifique liés à certaines identités. Après l'authentification, les processus d'autorisation vont autoriser ou limiter les niveaux d’accès et action à cette entité. Quand vous créer un utilisateur, vous devez choisir le mode d’authentification à utiliser, qui pourra être modifié par la suite.

**Mot de passe :** C’est l’authentification par la base de données. Chaque utilisateur est associé à un mot de passe qui doit être fourni quand l'utilisateur veut d'établir une connexion. Lors de la configuration du mot de passe, vous pouvez spécifier que celui-ci expire à la première connexion, ce qui oblige l’utilisateur à le changer. Si vous projetez d’activer l'option EXPIRE, il faut s'assurer que les utilisateurs ont la possibilité de le changer. Certaines applications n'ont pas cette fonctionnalité.

**Externe :** C’est l'authentification par le système d’exploitation, l’utilisateur se connecte à Oracle sans spécifier son login et son mot de passe. Avec cette authentification, votre base de données dépend du système d'exploitation ou d’une authentification par le réseau pour limiter l'accès à la base de données. Un mot de passe de base de données n’est pas utilisé pour ce type d’authentification. Si votre système d'exploitation ou les services réseaux le permettent, vous pouvez donc authentifier des utilisateurs. Si vous procédez ainsi, renseignez le paramètre OS\_AUTHENT\_PREFIX et utilisez ce préfixe dans les noms d'utilisateurs d'Oracle. Le paramètre OS\_AUTHENT\_PREFIX définit un préfixe qu'Oracle ajoute au début du nom du compte du système d'exploitation de chaque utilisateur. La valeur par défaut de ce paramètre est OPS$.

Par exemple, le paramètre OS\_AUTHENT\_PREFIX peut être initialisé comme suit :

OS\_AUTHENT\_PREFIX=OPS$

Si un utilisateur du système d’exploitation porte le nom « tsmith » et se veut se connecter à la base de données en utilisant l’authentification par le système d’exploitation, Oracle vérifie qu’il existe un utilisateur correspondant dans la base de données du non de : OPS$tsmith. Si c’est le cas, l’utilisateur est autorisé à se connecter. Toutes références à un utilisateur authentifié par le système d’exploitation doivent être préfixé, comme OPS$tsmith.

**Globale :** C’est l'authentification forte par l’intermédiaire des options d’Oracle Advansed Security. L'authentification globale permet à des utilisateurs d'être identifiés en utilisant la biométrie, les certificats x509, ou encore Oracle Internet Directory.

Le tablespace par défaut est le tablespace dans lequel les objets seront créés si aucun tablespace n’est spécifié pour la création de l’objet. Si vous ne choisissez pas de tablespace par défaut, alors le tablespace permanent par défaut défini au niveau de la base de données est utilisé.

Objets:

* Tables
* Triggers
* Indexe
* Vues
* Séquences
* Programmes stockés
* Synonymes
* Types de données définies par les utilisateurs
* Liens de base de données

Un quota est une l’allocation d’espace dans un tablespace donné. Par défaut, un utilisateur n'a aucun quota sur aucun tablespace. Vous avez trois options pour fournir un quota d'utilisateur sur un

**tablespace :**

* Illimité : cela signifie que l’utilisateur peut utiliser toute la place disponible sur le tablespace.
* Une valeur : c’est un nombre en Ko ou Mo. C’est la valeur du quota que l’utilisateur pourra utiliser sur le tablespace. Cette valeur peut être plus grande ou plus petite que l’espace actuellement disponible.
* Le privilège système UNLIMITED TABLESPACE : Ce privilège système outrepasse tous les différents quotas sur les tablespaces et donne à l'utilisateur un quota illimité sur tous les tablespaces, y compris SYSTEM et SYSAUX. Ce privilège doit être donné avec précaution.

D’une manière générale, vous ne devez pas donner de quota aux utilisateurs sur le tablespace SYSTEM ou SYSAUX. Normalement, seul les utilisateurs SYS et SYSTEM sont capables de créer des objets dans les tablespaces SYTEM et SYSAUX. Les utilisateurs n’ont pas besoin de quota sur le tablespace temporaire et sur les tablespaces undo.

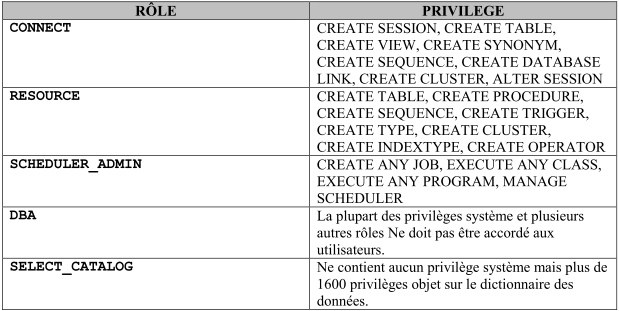
Pour la plupart des systèmes, il faut passer beaucoup de temps pour accorder les privilèges nécessaires à chaque utilisateur individuellement, augmentant de ce fait le risque d’erreur. Oracle fournit une gestion des privilèges contrôlée et facile grâce aux rôles. Les rôles sont des groupes de privilèges qui sont accordés aux utilisateurs ou à d'autres rôles. Ils permettent de faciliter l’administration des privilèges dans la base de données et par conséquent d’améliorer la sécurité.

Caractéristiques des rôles :

* Les privilèges sont accordés et révoqués des rôles comme si le rôle était un utilisateur.
* Les rôles peuvent être accordés ou révoqués à un utilisateur ou à un autre rôle comme s’ils étaient des privilèges système.
* Un rôle peut contenir des privilèges systèmes et objets.
* Un rôle peut être activé ou désactivé pour chaque utilisateur à qui le rôle a été attribué.
* Un rôle peut nécessiter un mot de passe pour être activé.
* Les rôles n’appartiennent à personne et ne sont dans aucun schéma.

Dans l’exemple ci-dessous, le rôle HR\_CLERK contient les privilèges SELECT et UPDATE sur la table employees. Le rôle HR\_MGR contient les privilèges DELETE et INSERT sur la table employees et le rôle HR\_CLERK. Le manager qui a le rôle HR\_MGR peut donc faire des SELECT, DELETE, INSERT et UPDATE sur la table employees.

* Une gestion plus facile des privilèges : Utiliser les rôles pour simplifier la gestion des privilèges. Plutôt que d'accorder le même ensemble de privilèges à plusieurs utilisateurs, vous pouvez accorder les privilèges à un rôle, et puis accordez ce rôle à chaque utilisateur
* Une gestion des privilèges dynamique : Si un privilège associé à un rôle est modifié, tous les utilisateurs qui ont ce rôle auront le privilège modifié automatiquement et immédiatement.
* Disponibilité sélective des privilèges : Des rôles peuvent être activés et désactivés pour ajouter ou enlever des privilèges temporairement. L’activation d’un rôle peut être utilisée pour vérifier que ce rôle a bien été attribué à un utilisateur.
* Attribué par le système d'exploitation : Des commandes du système d’exploitation ou des utilitaires peuvent être utilisés pour assigner un rôle à un utilisateur.



Les rôles sont normalement activés par défaut, ce qui signifie que si un rôle est accordé à un utilisateur, l’utilisateur peut exécuter les privilèges donnés à ce rôle. Il est possible de :

* Désactiver le rôle par défaut : quand un rôle est accordé à un utilisateur, décochez la case DEFAULT. Maintenant l'utilisateur doit explicitement activer le rôle avant que les privilèges du rôle puissent être exercés.
* Avoir un rôle qui requière une authentification supplémentaire. L'authentification par défaut pour un rôle est configurée à NONE mais il est possible que le rôle requière une authentification supplémentaire avant d’être utilisé.
* Créer des rôles d'application sécurisés qui peuvent être activés seulement en exécutant un bloc PL/SQL. Le bloc PL/SQL peut vérifier certaines choses telles que l'adresse réseau de l'utilisateur, quel programme l’utilisateur a lancé ou toute autre chose qui est nécessaire pour sécuriser correctement un groupe de permissions.

Pour activer un rôle qui n’est pas activé par défaut :

SET ROLE vacationdba;

Un rôle peut aussi être sécurisé par programmation:

CREATE ROLE secure\_application\_role

IDENTIFIED USING <security\_procedure\_name>;

Un rôle est un ensemble de privilèges qui peut être accordé à un utilisateur ou à un autre rôle. Vous pouvez utiliser les rôles pour gérer les privilèges de la base de données. Vous pouvez ajouter un privilège à un rôle et accorder ce rôle à un utilisateur. Un utilisateur peut alors activer le rôle, et bénéficier des privilèges de ce rôle Un rôle contient tous les privilèges qui lui ont été accordé et tous les privilèges des autres rôles qui lui ont été attribués.

Par défaut, Entreprise Manager accorde automatiquement le rôle CONNECT à un nouvel utilisateur. Ceci permet aux utilisateurs de se connecter à la base de données et de créer des objets dans leur propre schéma.

Un privilège est un droit qui permet d'exécuter un ordre SQL particulier ou d'accéder à l'objet d'un autre utilisateur. Oracle permet le contrôle pointu de ce que les utilisateurs peuvent ou ne peuvent pas faire dans la base de données. Les privilèges sont divisés en deux catégories :

* Privilèges systèmes : Chaque privilège système permet à un utilisateur d'exécuter une opération particulière sur la base de données, par exemple, le privilège CREATE TABLESPACE pour créer un tablespace. Les privilèges systèmes peuvent être accordés à un administrateur. Il y a plus de 100 privilèges systèmes.
* Privilèges objets : ce sont des privilèges qui permettent à l’utilisateur d’effectuer une action particulière sur un objet, comme une table, une vue, une séquence, une procédure, une fonction ou un package. Sans permission particulière, les utilisateurs ne peuvent avoir accès qu’à leurs objets. Les privilèges objets peuvent être accordés à un utilisateur, par un administrateur, ou pas quelqu’un à qui l’on a explicitement donné la permission d’accorder des droits sur cet objet.

Pour donner des privilèges systèmes, cliquer sur le lien Systems Privileges, sélectionnez le privilège approprié à partir de la liste des privilèges disponibles et déplacez le dans la liste des privilèges choisis en cliquant sur la flèche.

Accorder un privilège avec la clause ANY signifie que le privilège est accordé sur d'autres schémas.

Par exemple, le privilège CREATE TABLE ne vous permet de créer des tables que dans votre schéma.

Le privilège SELECT ANY TABLE vous donne le droit d’exécuter des ordres SELECT sur les tables des autres schémas. La check box Admin Option permet à cet utilisateur d'administrer le privilège et d'accorder le privilège système à d'autres utilisateurs.

Prenez bien en compte les conditions de sécurité avant d'accorder des privilèges systèmes. Certains privilèges systèmes ne sont généralement accordés qu’aux administrateurs :

* RESTRITED SESSION : ce privilège donne droit aux utilisateurs de se loguer même quand la base de données est ouverte en mode restreint.
* SYSDBA et SYSOPER : ces privilèges permettent aux utilisateurs d’éteindre, de démarrer, de restaurer et opérer d’autres taches administratives sur la base de données.
* DROP ANY objet : ce privilège permet d’effacer des objets même si ils ne nous appartiennent pas.
* CREATE, MANAGE, DROP, ALTER TABLESPACE : un non administrateur ne devrait pas avoir de contrôle sur les tablespaces.
* CREATE ANY DIRECTORY : Oracle permet à des développeurs d'appeler du code externe (exemple : une librairie C) depuis du PL/SQL. C’est une mesure de sécurité, les répertoires du système d’exploitation où le code est stocké doivent être reliés à un répertoire virtuel d’Oracle. Avec ce privilège, un utilisateur peut potentiellement appeler du code non sécurisé.
* EXEMPT ACCESS POLICY : Ce privilège permet à un utilisateur de dévier des fonctions de sécurité placées sur des tables ou des vues.
* GRANT ANY OBJECT PRIVILEGE : Ce privilège permet à des utilisateurs d'accorder des privilèges objets sur des objets qu'ils ne possèdent pas.
* ALTER DATABASE et ALTER SYSTEM : un non administrateur ne doit pas avoir le moyen d’effectuer des changements sur la base ou sur l’instance.

La base de données utilise des namespaces comme référence aux objets et schémas. Quand vous vous référez à un objet, Oracle va localiser l'objet dans le namespace approprié. Après avoir localisé l'objet, Oracle effectue l'opération de la requête sur l’objet. Si l’objet n’est pas trouvé dans le bon namespace, Oracle vous retournera une erreur.

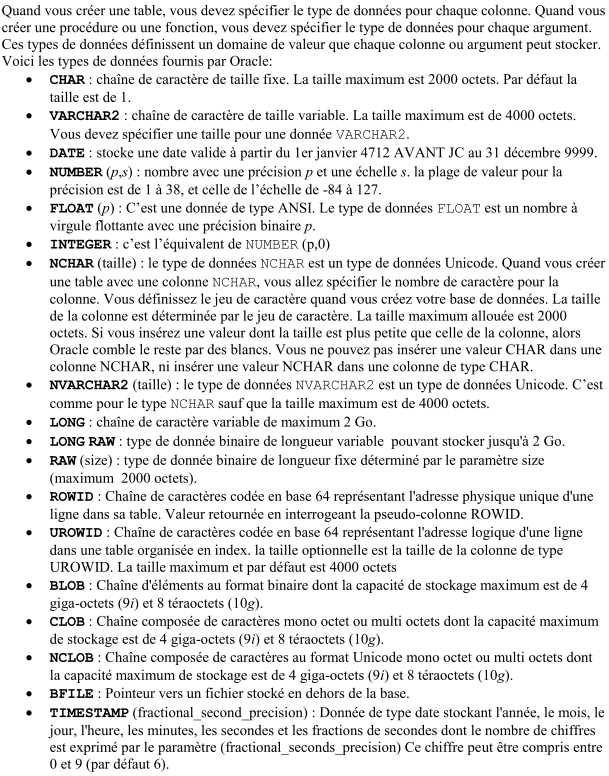
Les tables et les vues dans un même schéma ne peuvent pas porter le même nom car ces objets sont dans le même namespace. Cependant, les tables et les indexes sont dans différents namespace donc une table et un indexe dans un même schéma peuvent avoir le même nom.

Les objets suivants sont dans le même namespace:

* Tables
* Vues
* Séquences
* Synonymes
* Procédures stand-alone
* Fonctions stand-alone stockées
* Package
* Vues matérialisées
* Types définis par l’utilisateur

Les objets suivants ont leur propre namespace:

* Indexe
* Contraintes
* Clusters
* Triggers de la base de données
* Lien de base de données
* Dimensions



Vous pouvez imposer des contraintes pour l’insertion de données dans vos colonnes en configurant des contraintes d’intégrité :

* **NOT NULL :** Par défaut, toutes les colonnes peuvent avoir des valeurs NULL. La valeur NULL signifie une absence de valeur. La contrainte NOT NULL contraint la colonne de la table à ne pas contenir de valeur NULL. Par exemple, vous pouvez définir une contrainte NOT NULL pour vous assurer que pour chaque ligne de la table employees, la colonne last\_name contient une valeur.
* **UNIQUE key :** La contrainte d’intégrité UNIQUE requière que chaque valeur de la colonne soit unique. Il ne peut pas y avoir des lignes avec les mêmes valeurs. Par exemple, une contrainte UNIQUE est définie sur la colonne depatment\_name de la table department, interdisant les doublons pour cette colonne.
* **PRIMARY KEY :** chaque table de la base peut avoir au plus une clé primaire. Les valeurs formées par d'une ou plusieurs colonnes sujet à cette contrainte constituent un identifiant unique de la ligne.
* Une clé primaire garantie les deux conditions suivantes :
* Il ne peut pas y avoir deux lignes avec les mêmes valeurs dans la colonne spécifiée.
* Il ne peut pas y avoir de valeur NULL dans cette colonne.
* Contrainte d’intégrité référentielle :Différentes tables dans une base de données relationnelle peuvent être reliées par des colonnes communes et des règles existent pour garantir que ce lien entre ces colonnes doit être maintenu. Les règles d'intégrités référentielles garantissent que ces rapports sont préservés. Une contrainte d'intégrité référentielle exige que pour chaque ligne d'une table, la valeur des clés étrangères corresponde à une valeur de la clé primaire. Une clé étrangère est définie sur la colonne department\_id de la table employee. Cette contrainte garantie que chaque valeur de cette colonne correspond à une valeur de la clé primaire de la table departments (la colonne department\_id). Un autre type de contrainte référentielle d'intégrité s'appelle une contrainte d'intégrité autoréférentielle. Ce type de clé étrangère fait référence à une clé primaire présente dans la même table.
* **CHECK :** Une contrainte d'intégrité CHECK sur une colonne ou un ensemble de colonnes exige que la condition soit vérifiée pour chaque ligne de la table. Si un ordre DML ne remplit pas cette condition, alors celui-ci est annulé.

Vous pouvez choisir une table et puis effectuer plusieurs actions sur cette table :

* **CREATE LIKE :** Permet de créer une table avec la même structure que la table sélectionnée. Vous devez changer le nom de la table et le nom des contraintes. Vous pouvez aussi ajouter ou supprimer des colonnes ou faire des changements sur la structure de la table. Quand la table sera créée, elle sera vide.
* **CREATE INDEX :** Vous pouvez utiliser cette commande pour créer un indexe sur votre table. Des index doivent être créés uniquement sur les colonnes qui sont utilisées avec la condition WHERE d’un ordre SELECT et d’une instruction DML.
* **CREATE SYNONYM :** Un synonyme est un nom qui est utilisé à la place du nom complet de la table. Le synonyme peut être privé ou public.
* **CREATE TRIGGER :** Un trigger est un bloc PL/SQL qui est exécuté lorsqu’un événement particulier est effectué sur une table. Par exemple vous voulez garder une copie des données dans une table si on efface des données de la table employees. Vous allez donc créer un trigger « before delete » qui va insérer les lignes dans une autre table avant d’être effacé sur la table employees.
* **Flashback Table :** Flashback Table permet de récupérer une table à un moment précis... Il fournit une méthode de récupération pour une table qui a été accidentellement modifiée ou supprimée par un utilisateur dans le passé. Vous pouvez restaurer les données de la table ainsi que ses attributs, comme les indexes, triggers et autres. Cette opération est possible tant que la base de données est online en annulant les changements qui ont été apportés à la table. Vous pouvez revenir à un état de la table et à son contenu à un certain temps ou à un numéro de SCN (System Change Number) spécifié. Utilisez Flashback Table avec Flashback Query et Row Versions pour trouver un temps à partir duquel la table doit être restaurée.
* **Flashback by row version :** Il vous permet d’interroger les métadonnées et un historique des données pour une période donnée

**Les indexes** sont des structures optionnelles associées à une table. Ils sont créés afin d’améliorer les performances pendant la récupération des données. Un indexe Oracle fournit un chemin d'accès direct aux données de table. Des indexes peuvent être créés sur une ou plusieurs colonnes d'une table. Une fois l’index créé, il est automatiquement géré et utilisé par le serveur Oracle. Les mises à jour des données d'une table, comme l’ajout de nouvelles lignes, la mise à jour de lignes, ou la suppression de lignes, sont automatiquement propagées à tous les index appropriés avec une transparence complète pour l'utilisateur qui fait la modification.

Vous pouvez cliquer sur le lien Indexe de la page Administration pour voir la page des indexes. Vous pouvez voir les attributs des indexes où utiliser le menu Action pour voir les dépendances de ces indexes.

Les indexes peuvent être créé explicitement ou implicitement.

**Les vues** sont des représentations logiques des données d’une ou plusieurs tables ou d’une ou plusieurs vues. Les vues ne contiennent pas de données, mais à la place elles dérivent leurs données des tables sur lesquelles elles sont basées. Ces tables sont appelées tables de base d’une vue.

Une séquence est un objet de la base qui permet de générer des entiers uniques. Vous allez généralement utiliser des séquences pour générer les valeurs de vos clés primaires :

* Nom : Utilisez les conventions de nommage pour nommer une séquence.
* Schéma : C’est le propriétaire de la séquence.
* Type : Une séquence peut être en ordre croissant ou décroissant.
* Valeur maximum : Spécifie la valeur maximum que la séquence peut générer. Elle doit être supérieure à la valeur minimale ou initiale. Utilisez la clause Unlimited pour la valeur maximum de 10 27 pour une séquence croissante ou -1 pour une décroissante. La valeur par défaut est unlimited.
* Valeur minimum : Spécifie la valeur minimum que la séquence peut générer. Elle doit être inférieure ou égale à la valeur initiale et inférieure à la valeur maximale. Par défaut, il n’y a pas de limite. Utilisez le paramètre Unlimited pour une valeur minimum de1 pour une séquence croissante ou -10 26 pour une décroissante
* Intervalle : Spécifie l’intervalle entre deux nombres de séquences. Sa valeur peut être un entier positif ou négatif, mais ne peut pas être 0. Par défaut la valeur est 1 et doit être codée sur maximum 28 chiffres.
* Initial : Spécifie la première valeur que la séquence va générer. Ce chiffre doit être plus grand que la valeur minimum pour une séquence croissante et inversement pour une séquence décroissante
* Cycle : Un fois que la séquence a atteint sa valeur maximum, elle générera la valeur minimum. Pour une séquence décroissante, une fois le minimum atteint, celle-ci recommencera par la valeur maximum. Si vous ne choisissez pas cette option, une erreur sera retournée si vous demandé une nouvelle valeur à un une séquence épuisée.
* Cache : Spécifie combien de valeurs de la séquence sont mises en mémoire par Oracle pour avoir un accès plus rapide aux valeurs. Cette valeur peut être codée sur 28 chiffres ou moins. La valeur minimum pour ce paramètre est 2.

Par défaut tous les ordres DML ne sont pas validés de manière définitive (commit). Plusieurs outils (dont iSQL\*Plus) dispose d’une option qui permet de configurer le commit pour chaque ordre ou sur un groupe d’ordre.

Avant qu’un COMMIT ou qu’un ROLLBACK ne soit effectué, les changements sont dans un état intermédiaire (en suspens). Seul l’utilisateur qui a effectué les modifications est autorisé à voir les données changées. Les autres utilisateurs peuvent afficher ces données, mais verront uniquement les données telles qu’elles étaient avant la modification. Les autres utilisateurs ne peuvent pas non plus effectuer d’ordre DML sur les données modifiées non validées.

Par défaut lorsqu’un utilisateur modifie une ligne, si un second veut modifier cette même ligne il devra attendre que le premier effectue un commit ou un rollback pour sa transaction. Ceci est géré automatiquement par les mécanismes de verrouillage d’Oracle. La base de données ne peut fonctionner sans verrous parce que le verrouillage est effectué sur la ligne elle-même.

Par exemple :

COMMIT;

ROLLBACK;

Les personnes qui ne sont pas administrateurs n’ont pas besoin d’accéder au dictionnaire de données, mais elles peuvent y accéder si vous leur donnez les privilèges système du type \* ANY TABLE tels que SELECT ANY TABLE ou UPDATE ANY TABLE. Le dictionnaire de données contient des informations qu’un utilisateur mal intentionné pourrait utiliser pour pénétrer le système ou le corrompre. Pour protéger ce dictionnaire de données de ces privilèges système \* ANY TABLE, il suffit de mettre la valeur du paramètre d’initialisation O7\_DICTIONARY\_ACCESSIBILITY à FALSE. Et s’il y a des personnes non administrateur qui ont besoin d’accéder au dictionnaire de données, vous pouvez les y autoriser en :

* Utilisant la commande GRANT pour un objet bien défini du dictionnaire de données
* Donnant le privilège système SELECT ANY DICTIONARY pour donner un accès à l’ensemble du dictionnaire de données.

**6.1.2.2. Restreindre aux utilisateurs l’accès aux répertoires du système d’exploitation**

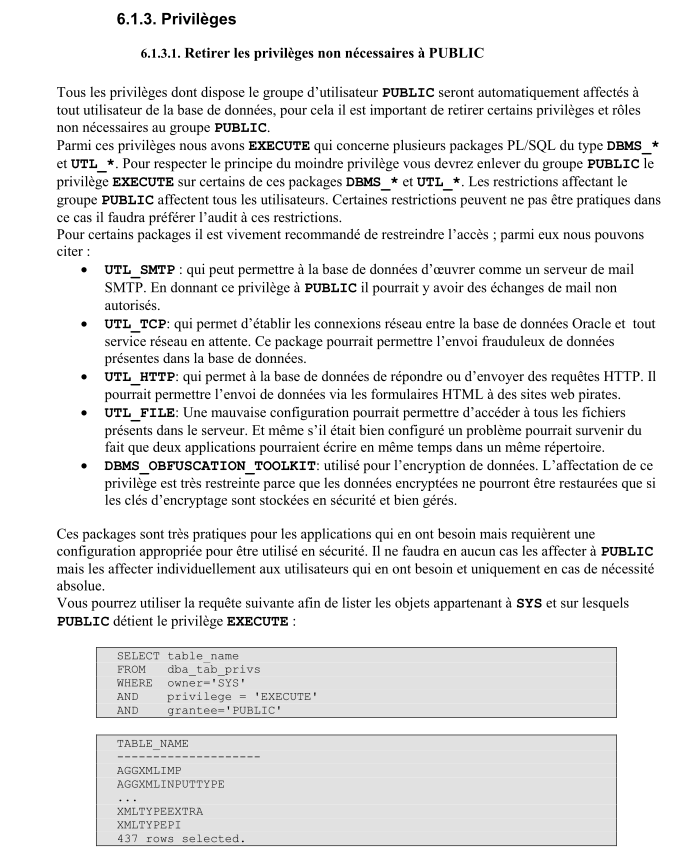
Le paramètre d’initialisation UTL\_FILE\_DIR permet de désigner sur quels répertoires les programmes PL/SQL auront le droit d’écrire ou de lire. Par défaut aucun répertoire n’est accessible, les privilèges du système d’exploitation prévalent. Les répertoires non accessibles par la personne ayant démarré la base de données ne le seront que s’il les configure grâce à UTL\_FILE\_DIR.

Pour spécifier plusieurs répertoires il suffit de les mettre entre apostrophes puis de les séparer par des virgules.

Ce paramètre n’est pas dynamique donc l’instance doit être redémarrée pour que les changements prennent effet. Aussi lorsque que vous spécifiez les chemins d’accès vers les répertoires vous ne pouvez pas utiliser les variables d’environnement.

L’instance ne vérifie pas si le répertoire spécifié existe ou pas donc on peut renseigner le paramètre UTL\_FILE\_DIR avant de créer les répertoires. Les informations contenues dans les dossiers autorisés sont accessibles par tous les utilisateurs PL/SQL donc le contenu de ces dossiers ne sera pas confidentiel pour aucun d’entre eux.

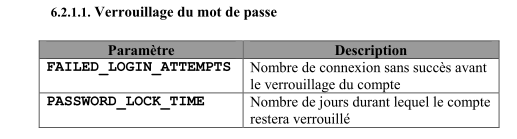
Remarque: Il ne faut jamais utiliser: UTL\_FILE\_DIR = \*, parce que cela permettrait aux utilisateurs d’accéder à tous les répertoires accessibles par l’instance Oracle, les répertoires des fichiers de données et de redo log inclus.



Il ne faut jamais donner aux utilisateurs plus de privilèges que ceux nécessaire. Pour implémenter le principe du moindre privilège, il faut procéder à des restrictions sur les privilèges suivants :

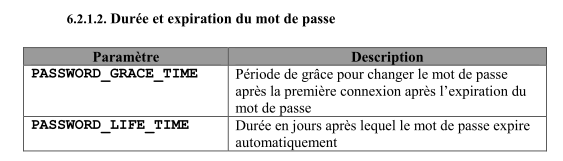
* Privilèges système et objet.
* Privilèges de connexion de l’utilisateur SYS comme SYSDBA et SYSOPER.
* Autres privilèges pour les DBA comme DROP ANY TABLE.

La gestion des mots de passe sous Oracle se fait grâce aux profils. Quand les utilisateurs sont créés ils ont **le profil DEFAULT** à moins qu’un autre profil leur soit attribué.



Le serveur Oracle verrouille automatiquement le compte après que l’utilisateur ait effectué un certain nombre de tentatives de connexion sans succès. Ce nombre de connexion sans succès est limité par le paramètre FAILED\_LOGIN\_ATTEMPTS. Ce compte sera automatiquement déverrouillé après la période de temps déterminée par le paramètre PASSWORD\_LOCK\_TIME ou l’administrateur devra utiliser la commande ALTER USER.

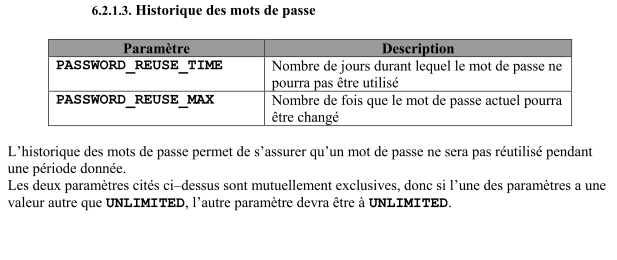
La commande ALTER USER peut-être utilisée pour verrouiller un compte ; Entreprise Manager peut aussi être utilisé. Par ailleurs lorsque le compte est verrouillé manuellement, il ne sera pas automatiquement déverrouillé après la période PASSWORD\_LOCK\_TIME, donc le DBA devra le faire manuellement.



L’administrateur de la base de données peut spécifier une période de grâce à l’aide du paramètre PASSWORD\_GRACE\_TIME, qui commence après la première connexion après l’expiration du mot de passe. Durant cette période un message d’alerte affichera à chaque connexion de l’utilisateur. Si l’utilisateur ne change pas de mot de passe, le compte sera verrouillé.

Remarque : Dans le cas d’un compte d’application (pas accessible via SQL\*PLUS), il faudra vérifier que l’application est capable de récupérer les messages d’erreurs avant d’activer l’expiration des mots de passe.

La plupart des DBA assigne des profils séparés pour les comptes d’utilisateurs et les outils de monitoring. Un compte d’utilisateur peut être forcé à expirer en faisant expirer le mot de passe.



Avant d’assigner un nouveau mot de passe à un utilisateur, une fonction PL/SQL peut être invoquée afin de vérifier la validité du mot de passe. La fonction de mot de passe doit être créée dans le schéma SYS.

Le serveur Oracle fournit une fonction de vérification par défaut que l’on peut visualiser en éditant le script $ORACLE\_HOME/rdbms/admin/utlpwdmg.sql. Le script utlpwdmg permet de créer la fonction VERIFY\_FUNCTION mais aussi de changer les paramètres du profil DEFAULT grâce à la commande ALTER PROFIL.

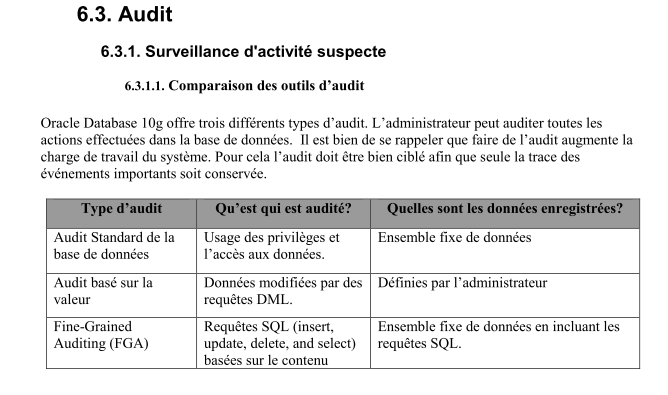
L’administrateur de la base de données peut écrire une fonction PL/SQL qui répond aux besoins de sécurité.

Si la fonction de mot de passe soulève une exception ou devient invalide, un message d’erreur est retourné et les commandes ALTER USER ou CREATE USER sont arrêtées.

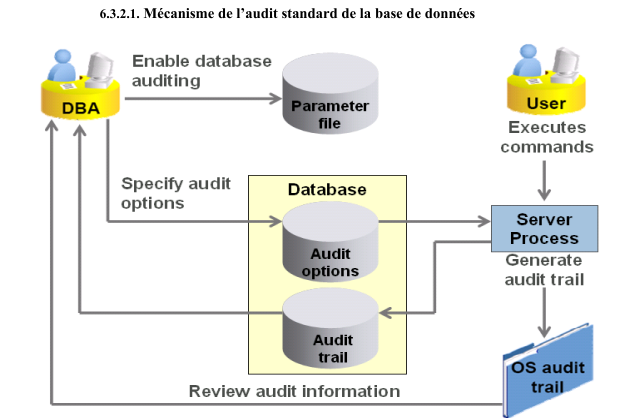
Oracle Database 10g compte plusieurs comptes créés par défaut. Ces comptes sont destinés à stocker des données mais aussi à posséder les objets PL/SQL et des programmes Java. Quand nous utilisons DBCA (Database Creation Assistant) pour créer la base de données; il verrouille et fait expirer tous les comptes par défaut sauf :

* SYS
* SYSTEM
* DBSNMP
* SYSMAN

Il faut savoir que vous pouvez créer une base de données grâce à un script. Les bases de données créées grâce à des scripts n’ont pas les comptes par défaut bloqués. Aussi plusieurs applications créent des comptes qui nécessitent d’être verrouillés.



L’audit standard de la base de données permet de capturer plusieurs informations telles qu’un événement a eu lieu, les informations sur l’auteur de cet événement et les informations sur la machine utilisée.

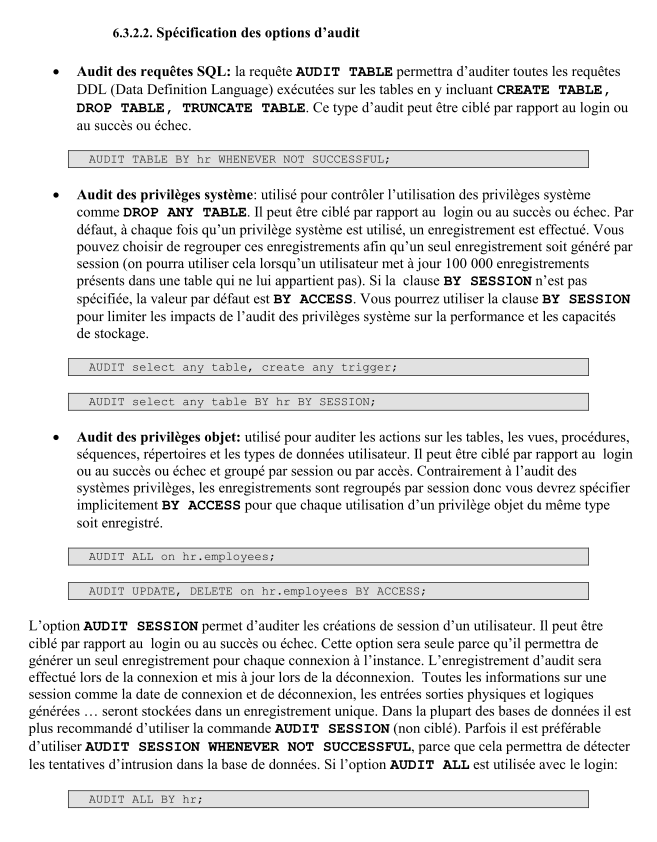


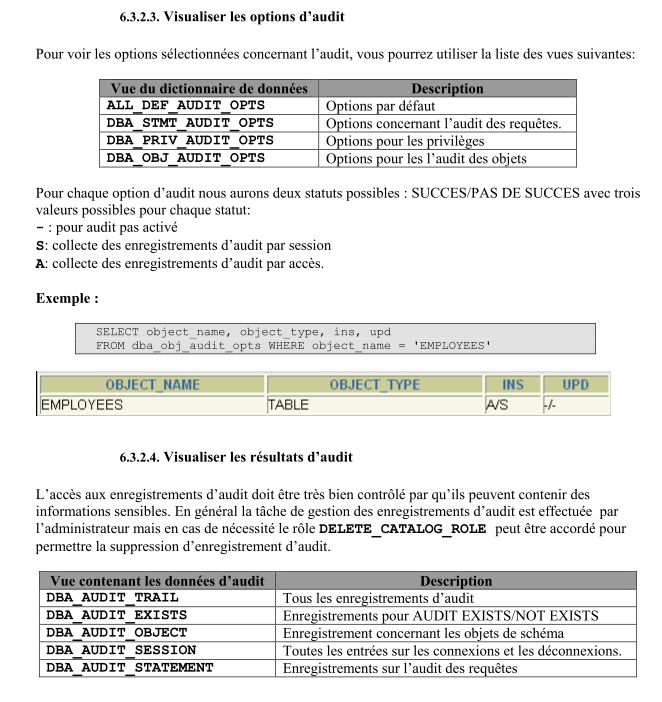
Après avoir activé l’audit de la base de données (grâce au paramètre AUDIT\_TRAIL) et spécifié les options d’audit (grâce à des requêtes SQL que nous verrons par la suite), la base de données pourra commencer à collecter les informations d’audit.

Si le paramètre AUDIT\_TRAIL est réglé à OS, alors les enregistrements d’audit seront stockés au niveau du système d’audit du système d’exploitation. Dans un environnement Windows ce sera un log d’événement. Dans le cas un environnement UNIX ou Linux, les enregistrements d’audit seront stockés dans un fichier. L’emplacement est déterminé grâce au paramètre AUDIT\_FILE\_DEST.

Si le paramètre AUDIT\_TRAIL est réglé à DB, les enregistrements d’audit seront dans une table qui appartiendra au schéma SYS.

L’audit est une tâche administrative très importante. En fonction des événements qui seront ciblés, la taille des données collectées peut croître de manière rapide. S’il n’est pas bien ciblé, l’audit peut consommer une si grande quantité d’espace que cela risque d’altérer les performances du système.





6.3.4. Fine-Grained Auditing (FGA)

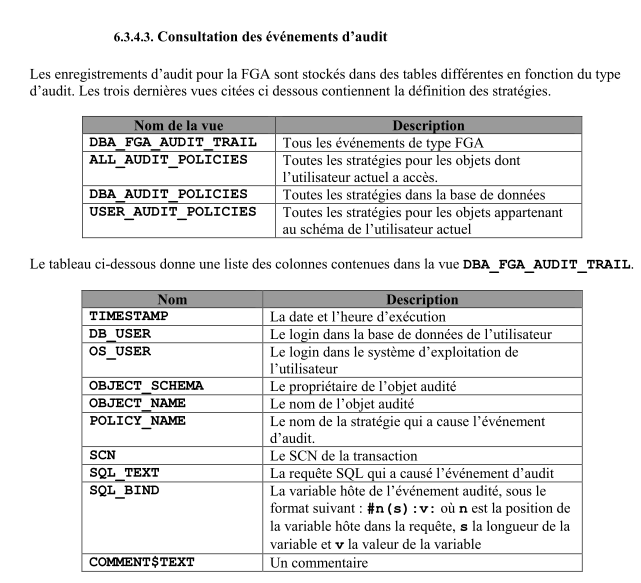
Les enregistrements de l’audit standard de la base de données permettent de savoir qu’une opération a eu lieu mais ne donnent pas la requête qui a été utilisée. L’audit Fine-Grained étend cette capacité en conservant les requêtes SQL permettant de manipuler des données. La FGA permet de cibler plus en détail l’audit que les deux autres types d’audit.

Contrairement à l’audit basé sur la valeur, la FGA n’utilise pas de trigger et a le même impact sur la performance que l’audit standard.

L’administrateur va utiliser le package PL/SQL DBMS\_FGA pour créer une stratégie d’audit sur une table ou une vue. Si une des lignes provenant du résultat d’une requête répond à la condition d’audit, alors un enregistrement d’audit sera créé et stocké. Optionnellement nous pourrons faire appel à une procédure qui sera exécutée lorsque la condition d’audit a été satisfaite. Le FGA peut avoir comme cible une requête SELECT qui retourne des milliers de lignes mais ne va générer qu’un seul enregistrement d’audit.

Le package DBMS\_FGA est l’outil d’administration pour les fonctions utilisées pour l’audit Fine-Grained. Le privilège EXECUTE sur le package DBMS\_FGA est nécessaire pour administrer les stratégies de ce type d’audit, par contre ce privilège devra être attribué uniquement qu’aux administrateurs. En désactivant la stratégie d’audit, on arrête aussi de générer des événements d’audit.

Vous pourrez redémarrer la génération d’événements d’audit en réactivant la stratégie.



6.3.4.5. Audit des utilisateurs de type SYSDBA et SYSOPER

Les utilisateurs de type SYSDBA et SYSOPER ont les privilèges de démarrer et d’arrêter la base de données. Les enregistrements d’audit pour ces utilisateurs doivent être stockés en dehors de la base de données car ces utilisateurs peuvent effectuer des modifications même si la base de données est éteinte. Oracle capture par défaut les événements de connexion pour les utilisateurs de type SYSDBA, les autres événements ne seront audités que lorsque cela sera explicitement spécifié.

Vous pourrez autoriser l’audit des autres événements pour les utilisateurs de type SYSDBA et SYSOPER en mettant le paramètre d’initialisation audit\_sys\_operations à TRUE en sachant que la valeur par défaut est FALSE.

Si les opérations de l’utilisateur SYS sont auditées, le paramètre AUDIT\_FILE\_DEST indiquera l’endroit où les enregistrements d’audit seront stockés. Dans un environnement de type Windows les enregistrements d’audit seront stockés dans les logs d’événements Windows. Dans les plateformes de type UNIX ou Linux les enregistrements d’audit seront stockés dans le répertoire :

$ORACLE\_HOME/rdbms/audit.

Au sein de la base de données, Oracle Net se matérialise par un processus actif appelé le listener.

Oracle Net Listener est responsable de la coordination des connexions entre la base de données et les applications externes. Sans listener aucune connexion depuis l’extérieur vers la base de données ne serait possible.

Oracle Net Listener sera donc la passerelle pour l’instance Oracle pour toutes les connexions à distance. Un Listener peut être utilisé par plusieurs instances de base de données et gérer des milliers de connexions de clients.

Le Listener est contrôlé grâce à Database Control ou à l’outil en ligne de commande lsnrctl.

Database Control permet de configurer les paramètres généraux du listener en utilisation comme la protection des mots de passe et la localisation des fichiers de logs.

Les administrateurs expérimentés pourront aussi configurer Oracle Net Services en éditant manuellement le fichier de configuration grâce à un éditeur de texte comme vi ou gedit sous Linux ou Notepad sous Windows.

La commande lsnrctl services donne le statut de tous les services Oracle Net que le processus doit gérer, qu’ils soient enregistrés statiquement et dynamiquement. Dans le résultat nous allons avoir les informations suivantes :

* Nom du service
* Le statut du service
* Le nombre de connexions établies ou refusées concernant les connexions dédiées ou partagées (les connexions partagées seront traitées dans le chapitre suivant).

La commande lsnrctl status retourne des informations détaillées sur le processus mais aussi un résumé des informations sur les services que ce processus d’écoute gère.

7.1.4. Enregistrement d’une base de données

Pour qu’un listener transfère les connexions de clients à une instance, il doit connaître le nom et l’emplacement du répertoire ORACLE\_HOME de l’instance. Le listener peut trouver les informations de deux manières :

* L’abonnement dynamique : les instances Oracle8i, Oracle9i et Oracle Database 10g vont s’abonner automatiquement au listener par défaut à leur démarrage. Le listener par défaut n’a besoin d’aucune configuration pour fonctionner.
* L’abonnement statique : Les anciennes versions d’Oracle Database ne s’enregistre pas automatiquement auprès du listener et donc nécessitent un fichier de configuration pour le listener qui va contenir la liste des services de toutes les bases de données qu’il doit servir. Vous pourrez utiliser ce type d’abonnement avec les nouvelles versions d’Oracle dans les cas suivants :
* Le port d’écoute du processus d’écoute n’est pas le port par défaut 1521.
* Votre application a besoin obligatoirement d’un abonnement en statique. Pour configurer un abonnement statique, sélectionnez Static Database Registration dans le panneau d’édition du listener et appuyez sur Add Database button. Entrez le nom du service (qui est équivalent au nom global de la base de données <DB\_NAME>.<DB\_DOMAIN>), le chemin d’accès du répertoire ORACLE\_HOME, et le SID (qui est équivalent au nom de l’instance). Cliquez sur Ok et redémarrer le listener pour que les changements prennent effet.

Entreprise Manager Database Control permet de configurer Oracle Net du côté du serveur, mais ne permet pas de le faire au niveau du client ou du serveur d’application. Oracle Net Manager est un outil graphique qui permet de configurer les connexions Oracle Net au niveau du client. Ainsi vous pourrez gérer :

* Le nom des services Oracle Net.
* Les profils Oracle Net
* La sécurité
* Le troubleshooting (logging et la conservation des traces)
* Les méthodes de résolution de noms.

Oracle Net Manager peut aussi être utilisé pour créer et configurer les processus d’écoute Oracle Net mais aussi de configurer la résolution de nom du service LDAP (Lightweight Directory Access Protocol). Pour démarrer Oracle Net Manager il suffit d’exécuter la commande run netmgr.

Lorsque vous allez ouvrir Oracle Net Manager, vous aller voir un arbre de navigation à votre gauche et la fenêtre des détails à votre droite. Vous pouvez consulter les options de chacune en dépliant le contenu grâce aux symboles + et -.

Les méthodes de résolution disponibles par défaut sont la résolution locale (TNSNAMES), Easy connecta (EZCONNECT) et la résolution locale (prédécesseur moins performant de Easy Connect).

tnsping est l’équivalent de l’utilitaire ping de TCP/IP. Il offre un test rapide qui permet de vérifier qu’une connexion réseau vers une destination donnée fonctionne correctement.

Cet utilitaire permet de vérifier qu’un nom d’hôte, un port, et un protocole donné permettent d’accéder à un processus d’écoute. Actuellement le tnsping ne vérifie pas le service dont le nom est spécifié est bien abonné au listener. Un autre avantage du tnsping c’est le fait de donner l’emplacement des fichiers de configuration ce qui peut être intéressant pour un système avec plusieurs répertoires ORACLE\_HOME.

Le contenu de la SGA (System Global Area) et de la PGA (Program Global Area) diffère en fonction du type d’architecture serveur utilisé (dédié ou partagé). Ce contenu concerne :

* Le texte et les formulaires analysés pour toutes les requêtes SQL stockées dans la SGA.
* L’état des curseurs qui va contenir les résultats en temps réel des requêtes
* Les données des sessions utilisateurs comme les informations sur la sécurité et l’usage des ressources.
* La pile qui contient les variables locales nécessaires pour les traitements.

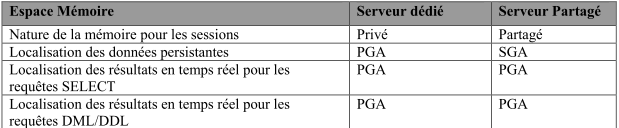
Les changements dans la SGA et la PGA seront transparents pour l’utilisateur, mais si vous avez beaucoup d’utilisateurs il faut penser à augmenter la valeur du paramètre LARGE\_POOL\_SIZE.

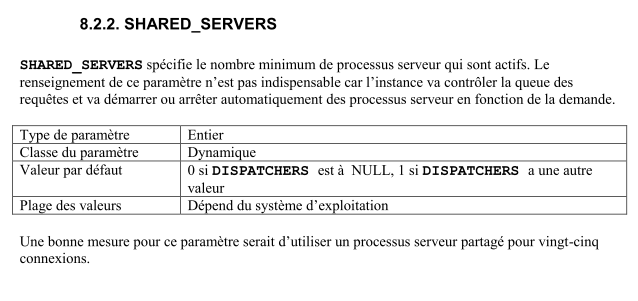
Chaque processus serveur partagé aura besoin d’avoir accès aux données de toutes les sessions afin de pouvoir traiter leurs requêtes. Vous pouvez contrôler la quantité de mémoire allouée à chaque session grâce au paramètre d’initialisation PRIVATE\_SGA.

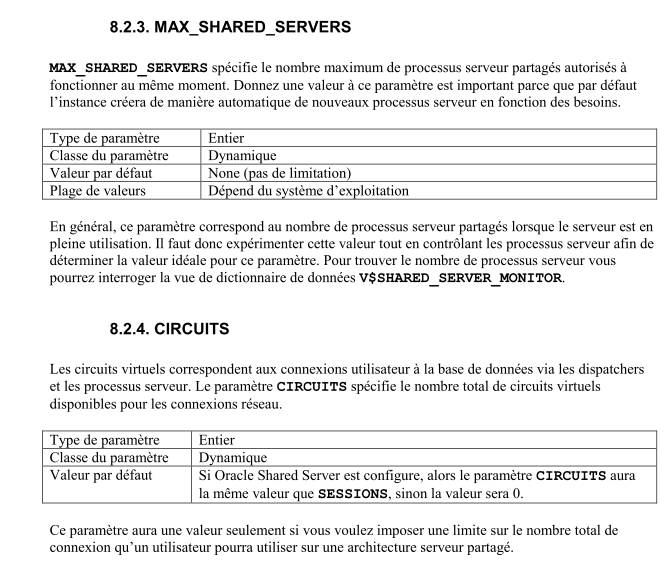
Dans le cas d’un serveur dédié les données de la session de l’utilisateur étaient stockées dans le PGA.

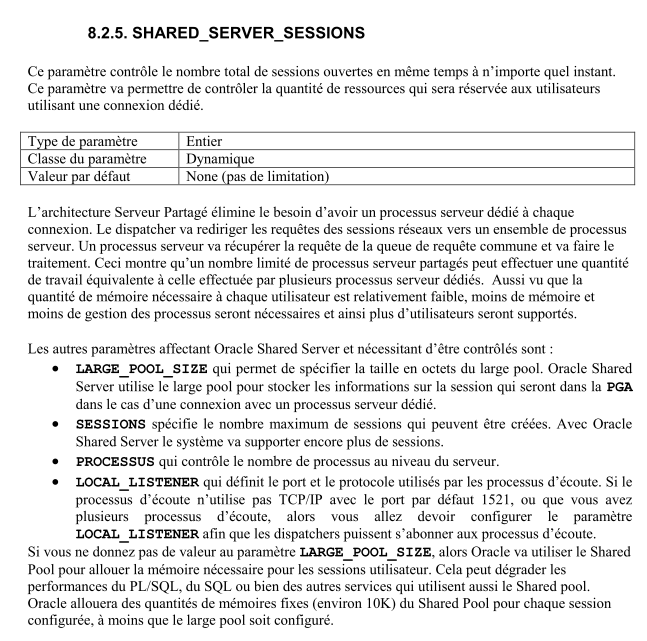
Dans ce cas la SGA contient le shared pool et les autres structures mémorielles, le PGA va contenir la pile mémoire, les données de la session de l’utilisateur et l’état des curseurs.

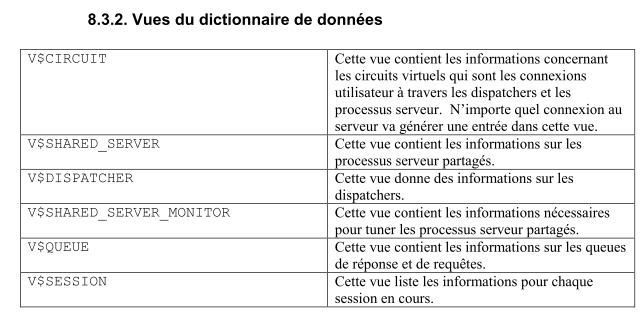
Dans le cas d’un serveur partagé, les données de la session de l’utilisateur seront stockées dans la SGA. Dans ce cas la SGA contient le shared pool, les autres structures mémorielles, les données de la session et l’état des curseurs. Le PGA va contenir seulement la pile mémoire.











9.1.4. Métriques du dictionnaire des données

Un des objectifs les plus importants des métriques du dictionnaire des données est de déterminer le statut des objets. Les indexes et les procédures stockées doivent être VALID pour être utilisés. Les indexes peuvent devenir UNUSABLE à cause des opérations de maintenance sur les tables. Les indexes stockent des pointeurs vers l’adresse physique d’une ligne spécifique. Si le DBA déplace les lignes pour des opérations de maintenance comme par exemple avec la commande MOVE, les pointeurs ne sont alors plus valides et l’indexe devra être reconstruit pour être à nouveau utilisé. Le code PL/SQL des objets peut avoir un statut INVALID pour différentes raisons. Si le code contient des erreurs, l’objet PL/SQL ne sera pas compilé correctement et sera marqué INVALID. Une procédure valide peut devenir invalide si un objet référencé devient invalide. Supposons qu’une procédure référence la colonne salary de la table employees. Si la définition de la colonne change, ou si la colonne est supprimée, rien ne dit que la procédure qui fonctionnait avec l’ancienne définition de la colonne fonctionnera avec la nouvelle. Les indexes UNUSABLE demande l’intervention du DBA pour les rétablir dans un état valide. Les objets PL/SQL INVALID sont normalement recompilés automatiquement la première fois qu’ils sont appelés, mais parfois demande l’intervention du DBA lorsque cette recompilation échoue.

Si vous trouvez des objets dont le statut est INVALID, la première question à vous poser est « l’objet a-t-il déjà été VALID ? » Souvent, les développeurs négligent de retravailler du code qui ne fonctionne pas. Si l’objet PL/SQL est INVALID à cause d’un code erreur, alors il y a peu de chose à faire pour résoudre le problème.

Si la procédure était valide à un moment dans le passé et est devenue récemment invalide, vous avez alors deux possibilités pour résoudre le problème:

* Ne rien faire. La plupart des objets PL/SQL sont automatiquement recompilés si nécessaire quand ils sont appelés. Les utilisateurs auront un léger délai d’attente pendant la recompilation des objets (dans la plupart des cas ce temps d’attente est transparent pour les utilisateurs).
* Recompiler manuellement les objets INVALID. Les objets PL/SQL invalides peuvent être recompilés manuellement en utilisant Enterprise Manager ou avec les commandes SQL.

Les indexes UNUSABLE sont marqués valides une fois reconstruit (recalcule des pointeurs). La reconstruction d’un indexe inutilisable recréé l’index à un nouvel emplacement et supprime l’index inutilisable.

La page Performance dans Enterprise Manager est le meilleur moyen de superviser le serveur si vous remarquez une dégradation des performances.

La page Performance se divise en 3 principales parties:

* Host: Métriques du serveur montrant le nombre de processus en attente et la taille de la mémoire paginée
* Active Sessions: En attentes et en cours de traitement: Une vue d’ensemble des performances de l’instance montrant des métriques de groupes de différentes catégories. Si une catégorie consomme une partie significative du temps d’attente pour l’instance, alors c’est à partir de cette catégorie que vous devriez commencer à résoudre les problèmes.
* Instance Throughput: Informations concernant les sessions en cours, la génération des redo, et l’activité concernant la lecture des fichiers de données.

Oracle Database 10g remonte des alertes à l’administrateur via Entreprise Manager Database Console.

Entreprise Manager peut être configuré optionnellement pour envoyer une notification par e-mail à l’administrateur sur un problème, de la même sorte qu’il est notifié dans la console.

Vous pouvez aussi sélectionner les seuils d’alertes (thresholds) sur plusieurs métriques de votre système. Oracle Database 10g peut vous prévenir si l’activité de la base dévie de la normale atteignant ainsi ces seuils d’alertes. Plus tôt les notifications aux problèmes potentiels sont remontées, plus vous pouvez y répondre rapidement, et les résoudre avant que ceux-ci n’affectent les utilisateurs.

Quelques métriques principales qui peuvent fournir des notifications plutôt :

- Le temps d’écriture moyen (centième de secondes)

- L’évolution de la taille du Dump Area (en pourcentage)

- Le temps de réponse (par transaction)

- Le temps de réponse des requêtes SQL (en pourcentage)

- L’espace utilisé des tablespaces (en pourcentage)

- Le temps d’attente (en pourcentage)

9.2.2. SQL Tuning et autres outils

Automatic Database Diagnostic Monitor (ADDM) : un outil coté serveur qui analyse les performances de la base de données toutes les 60 minutes. Le but d’ADDM est de détecter les goulots d’étranglements possibles dans le système et propose des solutions avant que les performances du serveur ne se dégradent.

SQL Tuning Advisor : Ce conseillé analyse les requêtes SQL individuellement et propose des recommandations pour améliorer les performances. Ces actions peuvent être une réécriture de la requête, un changement de la configuration de l’instance, ou l’ajout d’indexes. SQL Tuning Advisor n’est pas appelé directement. Cependant, il est appelé par d’autres outils tels que Top SQL ou Top Sessions, pour que l’optimisation sur les requêtes SQL ait un meilleur impact.

SQL Access Advisor : Cet outil analyse toutes les requêtes SQL exécutées pendant une période donnée et recommande la création d’indexes additionnels ou de vues matérialisées qui augmenteraient les performances.

Memory Advisor : Memory Advisor est un ensemble de fonctions conseillères qui aide à déterminer les meilleurs paramètres pour le shared pool, database buffer cache, et la PGA. En plus de ces diverses fonctions, cette page fournit un point de contrôle pour le large pool et le Java pool.

Mean-Time-To-Recover (MTTR) Advisor : MTTR advisor vous guide pour définir la durée nécessaire à la base de données pour effectuer sa récupération après un crash de l’instance. La notion MTTR et l’utilisation de ce conseillé sera abordé plus tard.

Segment Advisor : Il définit les tables et indexes qui consomme plus d’espace que nécessaire. Cet outils vérifie l’espace alloué inutilement au niveau d’un schéma ou d’un tablespace et fournit un script pour réduire celui-ci si possible.

Undo Management Advisor : Undo Management Advisor vous aide à déterminer la taille du tablespace undo qui est requise pour supporter la période de rétention définie.

Le rôle du DBA est de s'assurer que la base de données est ouverte et disponible pour tout utilisateur qui en a besoin :

* Anticiper et prévenir les dysfonctionnements les plus courants.
* Travailler pour augmenter Mean-Time-Between-Failures MTBF (le temps moyen entre les échecs), en s'assurant que le matériel est aussi fiable que possible, que les données critiques sont protégés par de la redondance, et que la maintenance du système d'exploitation est faite aux moments opportuns.

Oracle fournit des options de configuration avancée pour augmenter le MTBF incluant :

* Real Application Clusters
* Streams
* Diminuer Mean-Time-To-Recover MTTR (le temps moyen de restauration), grâce à des procédures de restauration anticipées et en configurant les sauvegardes pour qu'elles soient disponibles au moment voulu.
* Minimiser la perte de données. Les DBA peuvent configurer leurs bases de données de manière optimale afin d'eviter toute perte de données validée (COMMIT);

Les outils fournis dans cet optique sont :

* L'archivage des redo log
* Les bases de données standby et Oracle Data Guard (vu dans un prochain module)

Les dysfonctionnements peuvent être répartis en plusieurs catégories :

* Echec d'un requête: Une opération sur les données (select, insert, update, delete) échoue.
* Echec d'un user process: une session échoue.
* Echec réseau: la connectivité à la base de données est perdue.
* Erreur humaine: un utilisateur réalise une opération, mais celle-ci était incorrecte (suppression d'une table, entrée de données incorrectes).
* Echec de l'instance: l'instance de la base s'arrête de façon inattendue.
* Echec d'un media: un ou plusieurs fichiers de donnée sont perdus (supprimés, crash d'un disque).

