

Module n°1

Gérer une instance

1Z0-031

Auteur : Mathieu Musard
Gérer une instance – d septembre yyyy
Nombre de pages : 38

Table des matières

1.COMPOSANTS DE L'ARCHITECTURE ORACLE.....	4
1.1. INTRODUCTION AU SERVEUR ORACLE.....	4
1.1.1.Tâches de l'administrateur de base de données.....	4
1.2. CONNEXION A UN SERVEUR ORACLE.....	4
1.2.1.Processus utilisateur.....	5
1.2.2.Processus serveur.....	5
1.3. INSTANCE ORACLE ET BASE DE DONNÉES.....	6
1.3.1.System Global Area (SGA).....	6
1.3.2.SGA dynamique.....	7
1.3.3.Processus d'arrière plan.....	7
1.3.4.Fichiers de la base de données.....	8
1.3.5.Fichiers relatant à la Base de données.....	8
1.4. TRAITEMENT DE REQUÊTES.....	9
1.4.1.Etapes lors du traitement d'une requête.....	9
1.4.2.Pool partagé (Shared Pool).....	9
1.4.3.Buffer de Cache de la base de données (Database Buffer Cache).....	10
1.4.4.Buffer de Redo Log.....	10
1.4.5.Program Global Area (PGA).....	10
1.5. TRAITEMENT DE REQUÊTES DML.....	11
1.5.1.Phases d'exécution d'un requête DML.....	11
1.5.2.Segments de Rollback.....	12
1.5.3.Database Writer (DBWn).....	12
1.5.4.Log Writer (LGWR).....	13
1.6. TRAITEMENT D'UN COMMIT.....	14
1.6.1.Numéro de changement du système (SCN).....	14
1.6.2.Traitement d'un COMMIT : Etapes.....	14
1.6.3.Traitement d'un COMMIT RAPIDE (Fast Commit).....	15
2.OUTILS D'ADMINISTRATION.....	16
2.1. UTILITAIRES ADMINISTRATIFS.....	16
2.1.1.Utilitaires d'administration de base de données.....	16
2.1.2.Optimal Flexible Architecture (OFA).....	17
2.2. ORACLE ENTREPRISE MANAGER (OEM).....	17
2.2.1.Architecture d'OEM : Vue d'ensemble.....	17
2.2.2.Composants de la console OEM.....	18
2.2.3.Services communs d'OEM.....	19
2.2.4.Configurer OEM.....	19
2.3. PACKAGE D'ADMINISTRATION.....	19
2.3.1.Pack de management DBA.....	19
2.3.2.Packs de management supplémentaires.....	20
3.GÉRER UNE INSTANCE ORACLE.....	22
3.1.AUTHENTIFICATION ET PRIVILÈGES.....	22
3.1.1.Configurer l'authentification par le système d'exploitation.....	22
3.1.2.Configurer l'authentification par fichier de mots de passe.....	23
3.2.FICHIERS DE PARAMÈTRES.....	23
3.2.1.Paramètres présents dans le fichier de paramètre.....	23
3.2.2.Le PFILE (initSID.ora).....	23
3.2.3.Le SPFILE (spfileSID.ora).....	24
3.2.4.Règles pour les valeurs de paramètre.....	25
3.2.5.Paramètres essentiels.....	25
3.2.6.Paramètres communément modifiés.....	26
3.3. DÉMARRAGE ET ARRÊT D'UNE INSTANCE.....	26
3.3.1.Etapes de démarrage d'une instance.....	26
3.3.2.Démarrer une instance.....	27
3.3.3.Changer la disponibilité de la Base de données.....	27
3.3.4.Etapes d'arrêt d'une instance.....	27

3.3.5. Modes d'arrêts.....	28
3.4. VUE DYNAMIQUE DES PERFORMANCES.....	28
3.4.1. Caractéristiques des vues dynamiques de performance.....	28
3.4.2. Descriptions des vues dynamiques de performance.....	28
3.5. PARAMÈTRES DYNAMIQUES.....	29
3.5.1. Afficher les valeurs des paramètres courants.....	29
3.5.2. Paramètres d'initialisation dynamique.....	29
3.6. GÉRER LES SESSIONS.....	30
3.6.1. Activer les modes Restreint et Lecture seule.....	30
3.6.2. Terminaison de Session.....	30
3.7. CONFIGURATION D'ORACLE MANAGED FILES (OMF).....	31
3.8. FICHIER DE LOG ET D'ALERTES.....	31
3.8.1. Fichiers de Log.....	31
3.8.2. Fichiers d'Alerte.....	32
4. NATIONAL LANGUAGE SUPPORT (NLS).....	33
4.1. JEU DE CARACTÈRES.....	33
4.1.1. Caractéristiques d'NLS.....	33
4.1.2. Plans d'encodage de caractères.....	33
4.1.3. Caractéristiques du jeu de caractères.....	34
4.2. COMPORTEMENT DÉPENDANT DU LANGAGE.....	34
4.2.1. Paramètres NLS côté serveur.....	34
4.2.2. Paramètres NLS dérivés.....	35
4.2.3. Contourner le comportement par défaut NLS.....	35
4.2.4. Comportement dépendant du langage pour une session.....	36
4.2.5. NLS et le tri.....	36
4.2.6. Paramètres NLS dans les fonctions SQL.....	37
4.3. INFORMATION DU DICTIONNAIRE DE DONNÉES.....	37
4.3.1. Obtenir des informations sur les jeux de caractères.....	37
4.3.2. Obtenir des informations sur les configurations NLS.....	38

1. Composants de l'architecture Oracle

1.1. Introduction au serveur Oracle

1.1.1. Tâches de l'administrateur de base de données

L'administrateur de base de données (appelé **DBA**) est responsable du contrôle et de la configuration du système et des utilisateurs.

De nombreuses tâches vont être effectuées par celui-ci :

- Installer le serveur Oracle et les outils applicatifs, créer les bases de données.
- Assurer la maintenance et la disponibilité de la base de données.
- Créer la structure logique de stockage, c'est à dire les tablespaces, les tables, les vues et les indexes.
- Manager la structure physique de stockage, comprenant les fichiers de données, les fichiers de contrôle et les fichiers de redo log.
- Allouer et prévoir l'espace disque système de stockage nécessaire aux spécifications de la base de données.
- Maintenir, contrôler et monitorer les accès des utilisateurs.
- Assurer la sécurité du système.
- Monitorer les performances de la base de données.
- Créer un plan de sauvegarde et de récupération.

1.2. Connexion a un serveur Oracle

Pour travailler et consulter les données, les utilisateurs doivent tout d'abord se connecter à un Serveur Oracle. Il existe trois types de connexions grâce auxquelles un utilisateur peut accéder à un Serveur Oracle :

Connexion locale : Selon cette méthode, un utilisateur est directement connecté sur la machine faisant office de Serveur Oracle.

Connexion Deux Tiers : Ce type de connexion est couramment nommé "Connexion Client Serveur", un utilisateur se connecte à partir d'une machine directement connectée à un Serveur Oracle. Par exemple, lorsque qu'un utilisateur lance un Oracle Forms (Front End pour consulter des base de données Oracle à travers des formulaires.) sur un poste client sous Windows pour accéder à un Serveur Oracle résident sur une machine Windows NT Serveur.

Connexion Multi Tiers : Dans une architecture multi tiers, la machine de l'utilisateur se connecte à un Serveur applicatif (Par exemple un Serveur Web) qui lui même va se connecter au serveur Oracle pour récupérer les données issues de la base de données.

Lorsque qu'un utilisateur est connecté à une machine sur laquelle réside un Serveur Oracle, deux process sont invoqués : Le **process utilisateur** et le **process serveur**.

Un process utilisateur est créé lorsque tout outil client allant communiquer avec le serveur Oracle est lancé (Un outil d'administration, une application, SQL*Plus, une application Oracle Forms).

Un process utilisateur est un mécanisme qui va exécuter le code d'une application ou d'un outil Oracle.

Un process serveur est créé lorsque un utilisateur va se loguer sur le serveur en spécifiant un nom d'utilisateur, un mot de passe et un nom de base de données. Le process Serveur est créé sur le serveur Oracle.

Le process serveur va permettre l'interaction entre le process utilisateur et le serveur Oracle en récupérant les requêtes et en les exécutant sur les serveurs Oracle.

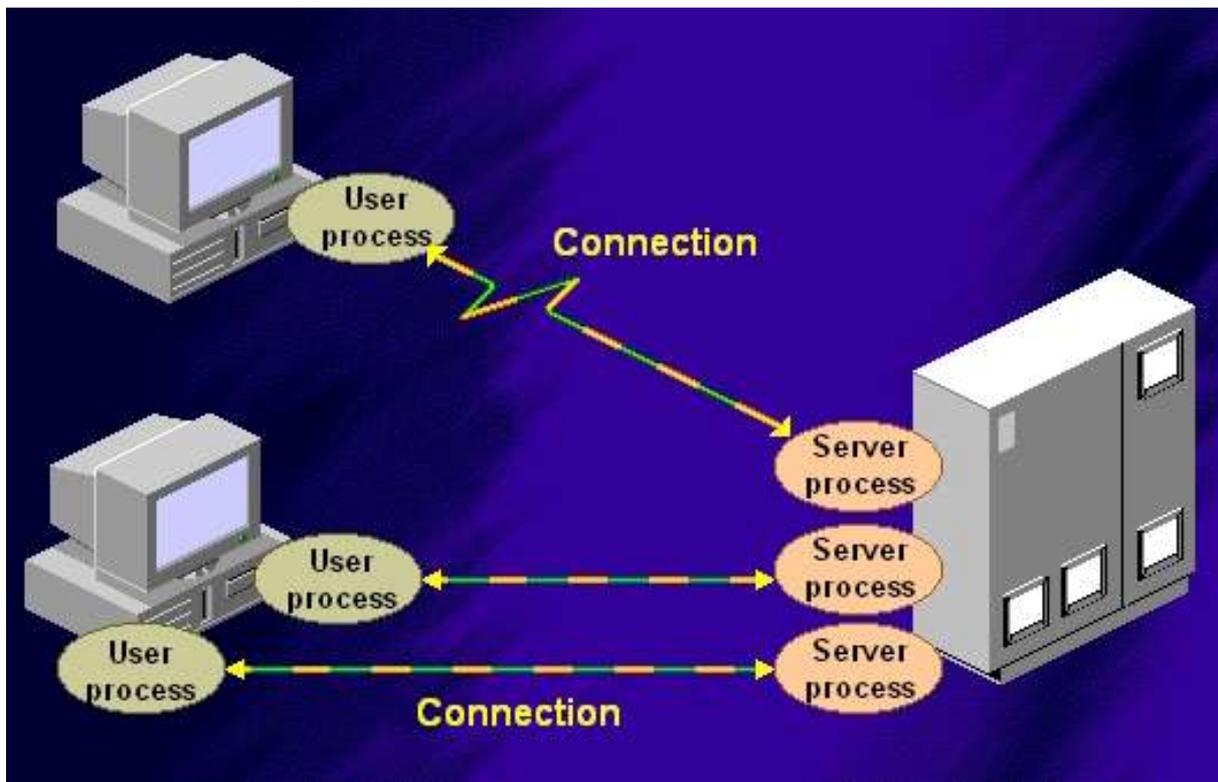
Le lien entre le process utilisateur et le process serveur est appelé une connexion.

- Si l'utilisateur se connecte localement sur le serveur, le chemin de communication est établi via un mécanisme de communication inter process.
- Si l'utilisateur se connecte via une machine cliente, un logiciel réseau est utilisé.

Une connexion spécifique entre un utilisateur et un serveur Oracle est appelé une **Session**.

La session démarre lorsque la connexion de l'utilisateur est validée par le serveur Oracle et se termine lorsqu'il se déconnecte ou lorsqu'une fin de connexion prématurée se produit.

De nombreuses sessions concurrentes d'un même utilisateur ou de plusieurs peuvent s'exécuter sur le serveur Oracle.



1.2.1.Processus utilisateur

Pour faire fonctionner un outil application tel que SQL*Plus ou des applicatifs Oracle Forms, Oracle crée un process utilisateur (Client). L'UPI (User Program Interface) est un mécanisme standardisé utilisé par un process utilisateur pour communiquer avec un process serveur. C'est l'UPI qui génère l'appel au serveur Oracle.

1.2.2.Processus serveur

Le process serveur a les caractéristiques suivantes :

Il traite les requêtes du process utilisateur et retourne le statut et le résultat de cette requête. Le DBA à la responsabilité de l'administration de ces process serveurs.

Chaque process serveur utilise une zone de mémoire appelée la PGA (Program Global Area). Cette zone est créée au démarrage du process serveur.

Le process serveur inclut l'OPI (Oracle Program Interface) qui est utilisé pour communiquer avec la base de données Oracle.

1.3. Instance Oracle et Base de données

1.3.1. System Global Area (SGA)

Un serveur Oracle est constitué d'une **instance** Oracle et d'une **base de données** Oracle. L'instance Oracle comprend une région de la mémoire appelée La **SGA** (System Global Area), ainsi que les processus d'arrière plan utilisé pour gérer la base de données. La SGA contient des données et des informations de contrôle sur le serveur Oracle.

La mémoire est allouée à la SGA lors du démarrage d'une instance et dés allouée lors de son arrêt. Cette zone mémoire est partagée et réside dans la mémoire vive qui n'est pas swappée ni paginée pour des raisons de performances.

La SGA se compose de plusieurs structures de groupe de mémoire, les trois principales étant :

Le pool partagé (Shared Pool) : Contient les requêtes SQL le plus récemment exécuté et l'information du dictionnaire de données la plus récemment utilisée.

Le buffer de cache de la base de données (Database Buffer Cache) : Cette structure stocke les données utilisatrices les plus récentes tirées des tables.

Le buffer de redo Log (Redo Log Buffer) : Ce buffer est utilisé pour enregistrer les modifications effectuées sur la base de données.

Une nouvelle unité d'allocation : Le granule.

C'est une unité d'allocation mémoire contiguë dépendant de la taille de la SGA. Si la `SGA_MAX_SIZE` est inférieur à 128 Mo, alors la taille du granule sera de 4 Mo. Si la taille est supérieure ou égal à 128 Mo alors la taille du granule sera de 16 Mo. On exprime donc la taille de sa SGA en granule qui en contient au minimum 3 (Fixed SGA (dont redo log buffer), buffer cache, shared pool).

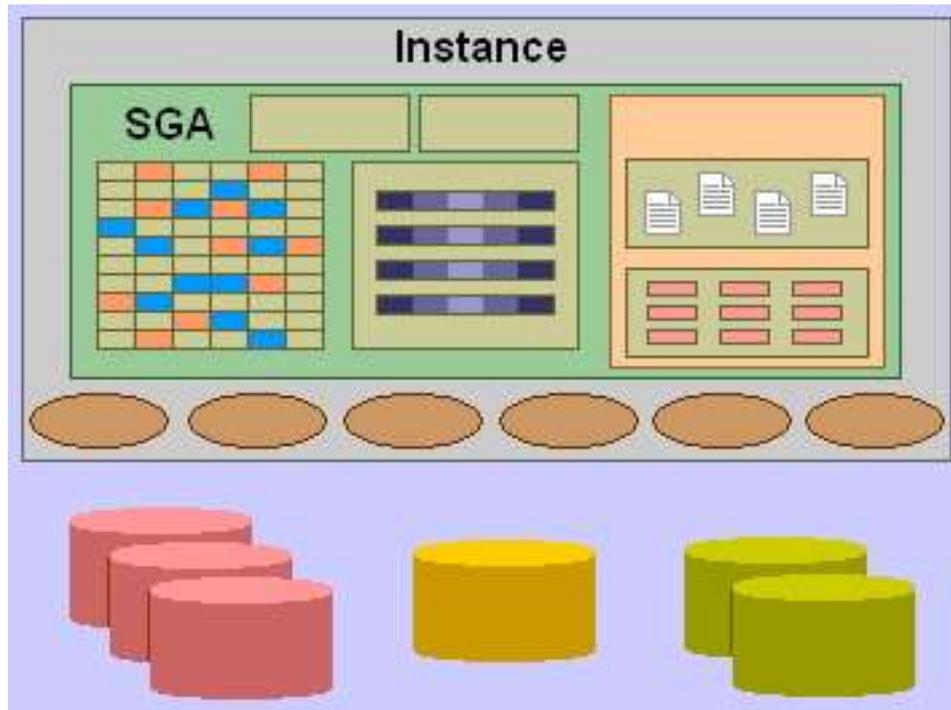
L'allocation des granules est montrée par la vue `V$BUFFER_POOL`.

On trouve aussi dans la SGA :

Le pool large (Large Pool) : Le large pool est une zone mémoire optionnel de la SGA configurée uniquement dans un environnement serveur partagé. Quand un utilisateur se connecte à un serveur partagé, Oracle a besoin d'allouer de l'espace supplémentaire dans les shared pool pour stocker les informations de connexions entre les différents process. Le pool large permet donc d'alléger certaines zones mémoires de la SGA. En effet, il n'a pas besoin de diminuer la mémoire allouée aux arbres de parcours au profit des informations de sessions, d'entrées/sortie, de sauvegarde et des processus de récupérations. Le gain de performance vient de la réduction de la charge occasionnée par l'augmentation ou la réduction de la mémoire cache partagée.

Recover Manager (RMAN), utilise le large pool pour restaurer les données en cas de défaillance.

Le pool Java (Java Pool) : Sa taille par défaut est de 24Mo et doit être utilisé en cas d'utilisation du langage Java.



1.3.2.SGA dynamique

Une SGA dynamique implémente une infrastructure qui permet de modifier la taille dynamiquement de la SGA sans éteindre l'instance. La taille de la SGA va augmentée par l'intermédiaire du data buffer cache, du shared pool, et du pool large sans jamais dépasser le SGA_MAX_SIZE. Les paramètres qui affectent la taille de la SGA sont :

DB_CACHE_SIZE : La taille de cache des blocs standard.

LOG_BUFFER : Le nombre de bit pour le redo log buffer cache.

SHARED_POOL_SIZE : La taille en bit du shared pool.

LARGE_POOL_SIZE : La taille du large pool, par défaut 0.

1.3.3.Processus d'arrière plan

Une instance Oracle comprend, en plus de la SGA, un ensemble de process d'arrière plan qui sont au niveau du système d'exploitation.

Oracle 9i comprend cinq process d'arrière plan obligatoire pour une instance :

Database Writer (DBWn) : Il écrit les données changées du buffer de cache de base de données vers les fichiers de données.

Log Writer (LGWR) : Il enregistre les changements qui sont instanciés dans le buffer de redo log vers les fichiers de redo log. (Ces changements sont référencés comme les données du redo log)

System Monitor (SMON) : Il exécute une récupération de l'instance lors de son démarrage, il nettoie les segments temporaires qui ne sont plus utilisés et annule les transaction qui n'avaient pas été validées lors d'un plantage du système. Il défragmente l'espace disponible dans les fichiers de données.

Process Monitor (PMON) : Il récupère les process lorsque qu'un process utilisateur échoue. Il est responsable du nettoyage du cache et de déverrouiller les ressources système bloquées par un process utilisateur.

Check point (CKPT) : il met à jour les statuts d'information sur la base de données tels que les entêtes des fichiers de base de données. Ceci est fait lors d'un événement de check point tel qu'une rotation des logs. Les changements dans le buffer de cache de la base de données sont alors écrits de manière permanente dans les fichiers de données.

En plus de ces process obligatoires, on trouve un ensemble de process optionnels :

Recoverer (RECO)

Archiver (ARCn)

Lock (LCKn)

Dispatcher (Dnnn)

Shared Server (Snnn)

Parallel Query Slaves (Pnnn)



1.3.4. Fichiers de la base de données

Une base de données Oracle a une structure logique et physique. Entendons par structure physique que les fichiers contenu dans la base de données ont une réelle existence (ex : fichiers de données). Pour sa part, la base de données Oracle est constituée de trois types de fichiers :

- **Les fichiers de données** contenant les données des blocs de données qui sont modifiées par les transactions courantes, le dictionnaire de données, les objets utilisateurs.
- **Les fichiers de redo log** contenant un enregistrement des changements qui ont été fait sur la base de données, pour avoir la possibilité de revenir en arrière en cas de dysfonctionnement. Une base Oracle contient au moins deux groupes de redo log chacun contenant au moins un fichier de redo log. Pour renforcer leur tolérance à une panne de disque dur, Oracle supporte les fichiers de redo log multiplexés. On peut ainsi maintenir une ou plusieurs copies de fichiers de redo log sur différents disques (On parle de fichiers mirrorés).
- **Les fichiers de contrôles** contenant les informations nécessaires pour maintenir et vérifier l'intégrité des données de la base. Une base de données Oracle nécessite au moins un fichier de contrôle, il est toutefois possible de les mirrorer sur plusieurs disques dur.

1.3.5. Fichiers relatant à la Base de données

Ce sont les fichiers qui ne font pas partie de la base de données mais qui sont utilisés par le serveur Oracle pour des opérations telles que le démarrage d'une instance ou l'authentification des utilisateurs. Il existe trois types de fichiers :

Les fichiers de paramètre (Parameter File) :

Il va définir les caractéristiques de l'instance. Lors du démarrage d'une instance, le serveur Oracle va lire les paramètres d'initialisation du fichier de paramètres.

On retrouve des informations telles que le nom de la base de données, la taille de la mémoire à allouer, le nom des fichiers de contrôle ainsi que les autres paramètres système.

Le fichier de mots de passe (Password File) :

Il est utilisé pour authentifier les utilisateurs privilégiés.

Les fichiers de redo log archivés (Archived Redo Log Files) :

Ils contiennent des copies off line des fichiers de redo log. Ils sont utilisés lors d'une récupération suite à une défaillance d'un média. Les fichiers de redo log sont sauvés lorsqu'ils sont pleins, cette fonctionnalité n'est utilisée que lorsque la base de données est en mode "ARCHIVELOG".

1.4. Traitement de requêtes

1.4.1. Etapes lors du traitement d'une requête

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

Ce traitement est découpé en trois étapes :

Parcours (Parse) :

Le process serveur reçoit la requête, analyse la syntaxe et vérifie les privilèges de l'utilisateur pour les objets référencés à accéder.

La section de mémoire du pool partagé de la SGA est utilisée pour compiler la requête et créer un arbre de parcours.

Le statut de réussite ou non de l'analyse est renvoyé au process utilisateur.

Exécution (Exécution) :

Préparation de la récupération des données.

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

Récupération des données (Fetch) :

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

1.4.2. Pool partagé (Shared Pool)

Le pool partagé est une zone mémoire faisant parti de la SGA et dont la taille doit être définie par le DBA. Le Shared Pool peut être retaillé dynamiquement avec la commande :

```
ALTER SYSTEM SET SHARED_POOL_SIZE = 128M;
```

Nous allons ici voir les différentes composantes de cette zone utilisée par le process serveur pour traiter les requêtes.

On trouve dans le pool partagé :

Le cache de librairie (Library Cache) :

Contient les informations sur les requêtes SQL et/ou PL/SQL récemment utilisées. Le plan d'exécution de ces requêtes est également stocké ainsi que l'arbre de parcours et le texte des requêtes. Les verrous et les handles sur le cache de librairie sont également consignés. Le cache de librairie est composé de deux structures : la zone partagée SQL et la zone partagée PL/SQL

Le cache du dictionnaire de données (Data Dictionary Cache) :

Contient des informations les plus récemment utilisées sur le dictionnaire de données, les objets de la base de données, sa structure et la définition des colonnes. On trouve également les utilisateurs valides, leur mot de passe et leurs privilèges. Les informations du dictionnaire de données sont mises en cache lorsque elles sont demandées par la base de données. Ceci permet d'augmenter la rapidité d'accès aux données.

1.4.3. Buffer de Cache de la base de données (Database Buffer Cache)

Lorsqu'une requête est exécutée, le process serveur consulte le database buffer cache (Compris dans la SGA) pour les blocs de données requis par la requête.

Le DBA doit configurer la taille de cette zone mémoire. Le buffer de cache est composé de trois caches indépendants :

- **DB_CACHE_SIZE** : Ne peut être mis à 0. Gère la taille du buffer par défaut
- **DB_KEEP_CACHE_SIZE** : Gère la taille du keep buffer cache, utilisé pour stocker les blocs dans la mémoire susceptible d'être réutilisés.
- **DB_RECYCLE_CACHE_SIZE** : Gère la taille du recycle buffer cache, employé pour supprimer les blocs de mémoire réutilisé qui ont subi peu de changement.
- **DB_CACHE_ADVICE (ON, OFF, READY)** : Permet d'activer les statistiques qui prédisent le comportement de la base lorsqu'on modifie la taille du cache. Les résultats sont consultables à l'aide de la vue `V$DB_CACHE_ADVICE`.

Le database buffer cache contient des blocs de données lus dans les fichiers de données. Cela permet au process serveur d'accéder plus rapidement aux données, cette zone comporte les caractéristiques suivantes :

Le database buffer cache contient les données les plus récemment et fréquemment utilisées.

Il contient des blocs de données modifiées et non modifiées datés selon un algorithme Oracle (LRU) pour mettre à jour le buffer.

La taille de chaque buffer du buffer cache est égale à la taille d'un bloc Oracle et spécifié par le paramètre `DB_BLOCK_SIZE`.

1.4.4. Buffer de Redo Log

Le buffer de redo log enregistre les changements faits lors d'une requête DML.

Ces caractéristiques sont les suivantes :

- Il fait parti de la SGA.
- Sa taille est définie par le paramètre `LOG_BUFFER` et doit être définie par le DBA.
- Il se compose d'entrées qui contiennent les blocs changés, la position du changement et la nouvelle valeur.
- Il n'identifie pas le bloc où les changements sont faits, c'est-à-dire qu'il traite de manière similaire les blocs de données, d'index ou de segments de ROLLBACK.
- Son utilisation est séquentielle et circulaire. Il est renouvelé lorsqu'il est plein et que les entrées ont été enregistrées dans les fichiers de redo log on-line.

1.4.5. Program Global Area (PGA)

Lorsque qu'un process serveur est créé, ou un process d'arrière plan, la base de données Oracle alloue une zone globale de programme (PGA Program Global Area ou bien Process Global Area) dans la mémoire. C'est un buffer mémoire qui contient les données et les informations de contrôle pour ce process serveur.

Elle contient pour un serveur dédié :

Une zone de tri (Sort area) : Utilisé pour stocker, si nécessaire, les résultats intermédiaires lors du tri.

Des informations de session : Contient les privilèges utilisateurs pour la session en cours.

Un état de curseur (Cursor State) : Etat de l'étape dans l'exécution des éventuels curseurs de la session.

Un curseur est un pointeur sur la mémoire associé à une requête SQL donnée. C'est une zone de travail utilisé pour stocker le résultat de la requête.

Un espace de pile (Stack Space) : Contient les variables de session et les arrays.

Dans la version 9i, de nouveaux paramètres dynamiques de gestion de la PGA sont apparus. Par exemple, PGA_AGGREGATE_TARGET définit la taille maximale que peut atteindre la PGA totale allouée à toute une instance. La taille total de la PGA est inférieur au PGA_AGGREGATE_TARGET en mode auto de WORKAREA_SIZE_POLICY.

1.5. Traitement de requêtes DML

1.5.1. Phases d'exécution d'un requête DML

Les requêtes de manipulations de données (DML), tel que les UPDATE, DELETE ou INSERT, sont utilisées pour manipuler les données. Comprendre leur exécution permet de créer le data buffer cache et les segments de ROLLBACK appropriés.

Le traitement d'une requête DML se compose de deux phases :

Parcours (Parse) : Contrôle des erreurs et placement dans une mémoire partagée SQL.

Exécution : Si la phase de parcours s'est effectuée avec succès, l'exécution se fait en plusieurs étapes.

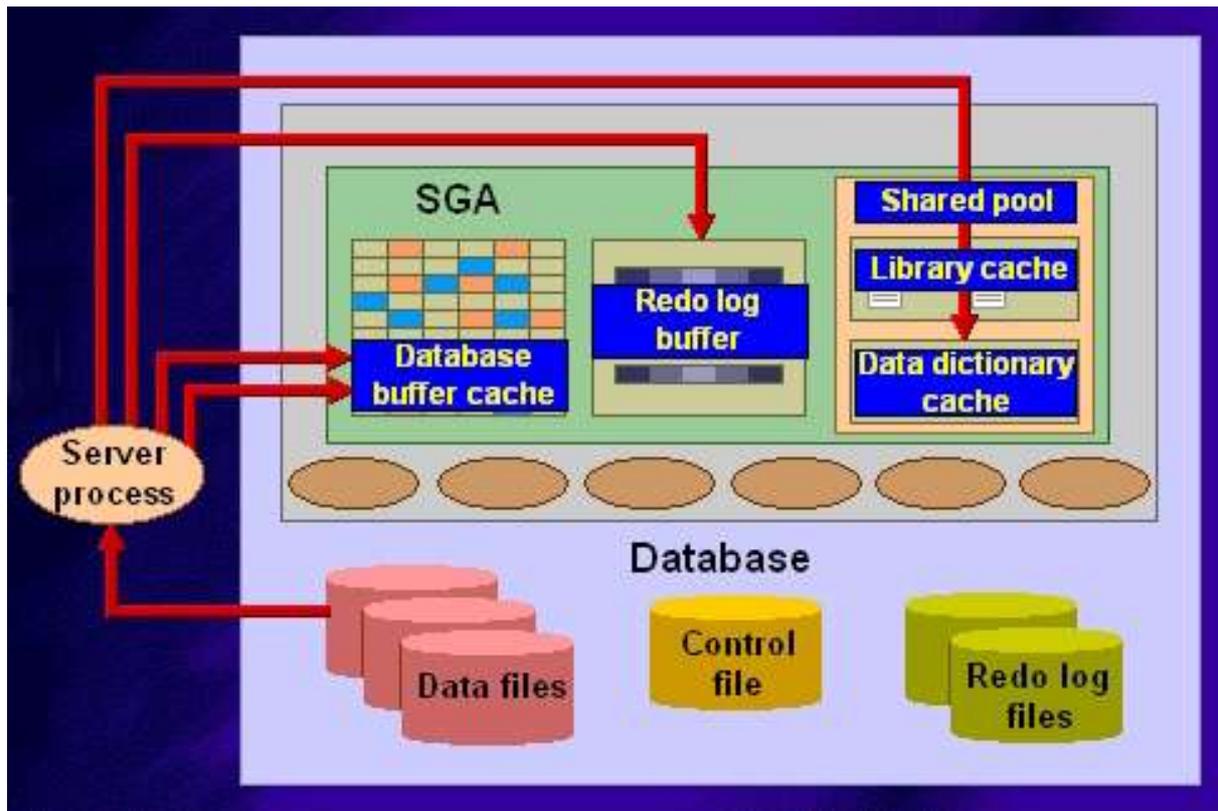
Lecture des blocs de données et de ROLLBACK dans le database buffer cache.

Mise en place de verrous sur les données qui vont être changées.

Ecriture de l'image-avant dans les segments de roll back et enregistrement des changements du buffer cache dans le redo log buffer pour protéger l'image-avant en cas de défaillance.

Finalement, les données sont enregistrées dans le buffer cache et les changements sont également enregistrés dans le redo log buffer protégeant ainsi l'image-après.

Les blocs de données altérées dans le buffer cache sont marqués comme " sales (Dirty) ", c'est-à-dire qu'ils ne correspondent plus aux blocs dans les fichiers de données.



1.5.2.Segments de Rollback

Les segments de Rollback sont utilisés pour sauver les données lors d'une requête DML afin de les restaurer en cas d'annulation de la transaction ainsi que pour empêcher à d'autres transactions de voir les données en cours de modification.

Ces segments ne sont pas visibles par un utilisateur.

Une base de données contient un ou plusieurs segments de Rollback.

Le serveur Oracle assigne un segment de Rollback à chaque transaction.

Les segments de Rollback peuvent permettre de récupérer des données non commitées lors d'une transaction.

Ils permettent également de récupérer un état consistant de la base de données en cas de défaillance système.

Les segments de Rollback sont contenus dans les fichiers de données et sont lus dans le database buffer cache si nécessaire.

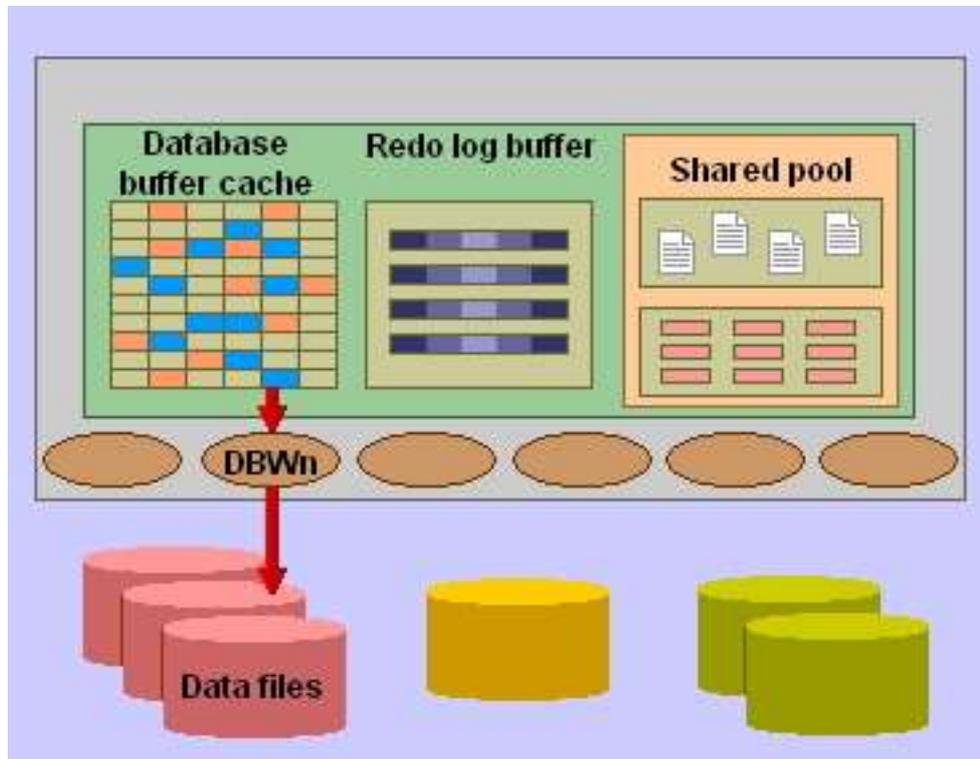
1.5.3.Database Writer (DBWn)

Ce process d'arrière plan écrit les données du database buffer cache vers les fichiers de données sur le disque.

Ces blocs de données sont écrits lorsque certains évènements se produisent tel que :

- L'écriture a lieu lorsque le nombre de buffers « sales » atteint un seuil, cela permet de libérer des buffers dans le database buffer cache.
- L'écriture se produit lorsque qu'un process serveur ne trouve plus de buffer libre après avoir recherché un certain nombre de buffers.
- L'écriture s'effectue lors d'un timeout, par exemple lorsque le DBWn reste inactif plus de trois secondes. Le DBWn recherche alors des buffers « sales » selon l'algorithme LRU ou bien écrit la totalité si la base de données est inactive.

- Le DBWn se déclenche lorsqu'un check point est atteint, cela se produit par exemple lors de l'arrêt de la base de données.
- Lorsqu'un tablespace temporaire ou normal est mis offline.
- Lorsqu'un tablespace est mis en lecture seule.
- Lorsqu'une table est tronquée ou supprimée.
- Lors d'un timeout.

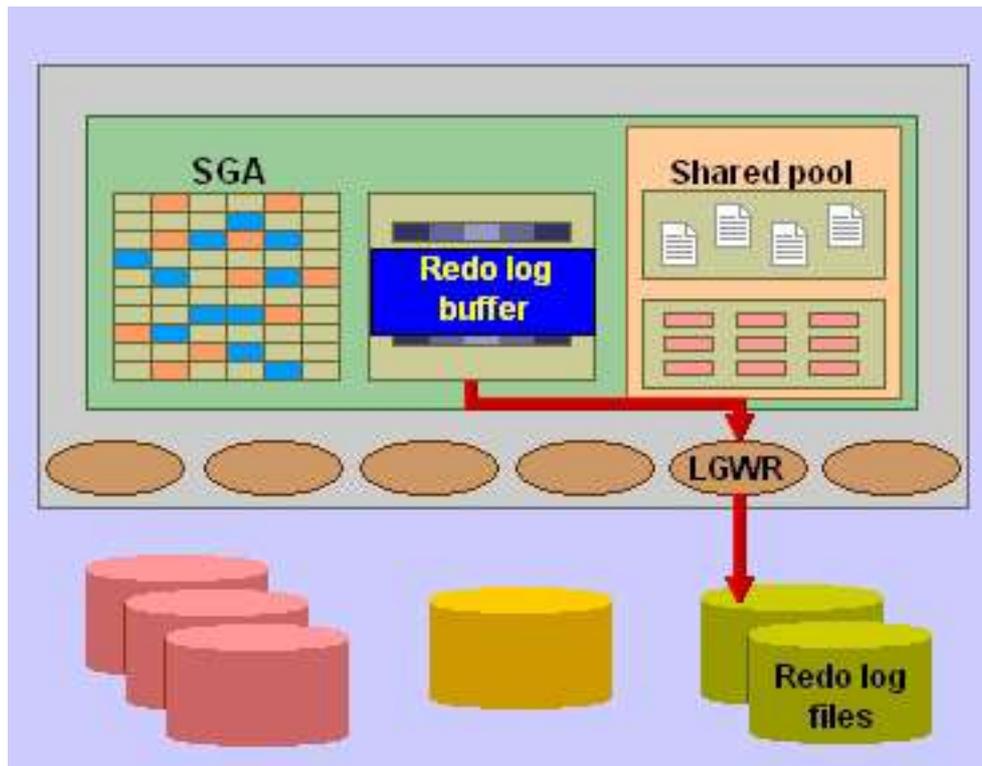


1.5.4. Log Writer (LGWR)

Le Log Writer écrit séquentiellement des entrées du redo log buffer vers les fichiers de redo log. Le process LGWR écrit lorsque l'un des quatre événements suivants se produit :

- Lorsque le buffer de redo log est plein à un tiers de sa capacité.
- Lorsqu'un timeout se produit, par exemple lors d'une inactivité de plus de trois secondes.
- Avant que le process DBWn écrive les données du database buffer cache vers les fichiers de données.
- Lorsqu'une transaction utilisateur est commitée.
- Toutes les 3secondes.

LGWR peut demander à DBWn d'écrire dans les fichiers de données.



1.6. Traitement d'un commit

1.6.1. Numéro de changement du système (SCN)

La base de données Oracle assigne un SCN (System Change Number) à chaque transaction qui est committée dans le but de l'identifier de manière unique.

Ce numéro unique est enregistré dans les segments de Rollback ainsi que dans les fichiers de redo log. Il est également utilisé comme une marque temporelle (Timestamp) pour permettre à la base de données de synchroniser les données. Les événements sont organisés dans un ordre temporel au sein d'une instance et entre instances.

Il permet au serveur Oracle de lire les données de manière consistante en faisant abstraction de la date système du système d'exploitation lors de la restauration de la base de données.

1.6.2. Traitement d'un COMMIT : Etapes

Une transaction doit être écrite de manière permanente après le traitement de celle-ci. Pour cela un COMMIT est réalisé. Nous allons voir les étapes qui constitue cette action de commit :

- Un process serveur enregistre une ligne COMMIT accompagnée de son SCN dans le buffer de redo log.
- Le process LGWR écrit les entrées du buffer de redo log dans les fichiers de redo log en commençant par la ligne de COMMIT (LGWR confirme la réussite du commit seulement après que le buffer de redo log soit écrit sur le disque.).
- Le process serveur enregistre les informations indiquant que la transaction est terminée. Les ressources verrouillées dans les tables et les enregistrements sont alors libérées.
- Le process serveur informe le process utilisateur de la fin du COMMIT.

Un commit réalisant ces étapes est appelé un commit rapide (Fast Commit), on remarquera que seul le buffer de redo log est écrit sur le disque. Les buffers sales sont écrits dans le fichier de données de manière indépendante par le process DBWn.

Une seule écriture sur le disque est donc nécessaire pour réaliser un COMMIT. Si plusieurs process utilisateurs réalisent un commit simultanément, le serveur Oracle rassemble les commits et n'écrit qu'une seule fois.

1.6.3. Traitement d'un COMMIT RAPIDE (Fast Commit)

Un serveur Oracle utilise un mécanisme de COMMIT rapide. Cela lui permet de restaurer les données en cas de défaillance du système d'exploitation.

Une écriture séquentielle est réalisée dans les fichiers de redo log ce qui est un gain de temps considérable comparé à une écriture des données dans différents blocs des fichiers de données.

Seules les informations nécessaires pour enregistrer les changements sont écrites dans les fichiers de redo log, ce qui est plus économique en temps car écrire les données dans les fichiers de données demande d'écrire les blocs entiers et non les seuls changements.

Plusieurs transactions peuvent être committées en une seule fois et à l'aide d'une seule écriture.

Le FAST COMMIT ne requiert qu'une seule écriture synchronisée par transaction, mis à part si le buffer de redo log est plein, la taille d'une transaction n'affecte donc pas la rapidité d'un commit.

2. Outils d'administration

2.1. Utilitaires administratifs

2.1.1. Utilitaires d'administration de base de données

Oracle Universal Installer :

L'installateur Universel est un installateur Java utilisé pour installer Oracle9i, les outils Oracle et les utilitaires.

Il peut installer des logiciels à partir d'un CDROM ou d'Internet, détecter les dépendances entre les différents composants. Il détecte le langage courant du système d'exploitation ou bien utilise le langage par défaut.

Il est possible de spécifier différents répertoires principaux (Le ORACLE_HOME) pour permettre d'installer plusieurs versions d'un même produit. L'installation peut être guidée par des assistants ou permet d'utiliser des packages standard nécessitant un minimum de réponses.

Il est également possible de le lancer en mode silencieux, dans ce cas, un fichier de réponse est utilisé pour remplacer les réponses aux différentes questions lors de l'installation. Le mode silencieux est utilisé lorsqu'il n'y a pas de terminal graphique ou lorsque l'on veut que l'utilisateur n'ait rien à faire pour l'installation.

La syntaxe de lancement de la commande varie en fonction du système d'exploitation :

```
Unix :  
$ ./runInstaller  
  
Unix (Installation silencieuse en utilisant les fichiers de  
réponses) :  
./runInstaller -responsefile myrespfile -silent
```

Un fichier de log est créé lors de l'installation et contient l'ensemble des étapes et les résultats.

Une liste des produits installés est maintenue en cas de désinstallation de l'un d'entre eux.

Les outils de post-installation sont également lancés si besoin est. Par exemple, après l'installation d'OEM (Oracle Enterprise Manager), l'assistant de configuration de base de données est lancé.

Oracle Database Configuration Assistant :

Cet assistant permet de créer une base de données, configurer les options de la base de données, supprimer une base de données, gérer les templates.

Outils pour la création des fichiers de mots de passe :

La commande orapwd permet de créer un fichier de mot de passe :

```
$orapwd file=$ORACLE_HOME/dbs/orapw15  
Password=admin entries=5
```

Ensuite, il suffit de mettre REMOTE_LOGIN_PASSWORDFILE à EXCLUSIVE dans le fichier de paramètre. Puis, ajouter les utilisateurs au fichier de mot de passe et de leur attribuer des privilèges.

SQL* Plus : SQL* Plus est un outil Oracle qui permet de manipuler et interagir avec la base de données. Une présentation plus complète de SQL* Plus est disponible dans le module SQLP.

Oracle Enterprise Manager : C'est un outil important permettant, via une interface graphique, d'administrer, de contrôler et de tuner une ou plusieurs bases de données localisées sur un ou plusieurs serveurs.

2.1.2. Optimal Flexible Architecture (OFA)

OFA est l'architecture standard qu'il est recommandé d'utiliser dans une base de données Oracle. Les trois règles importantes pour respecter cette architecture sont :

- Etablir une structure de répertoires, où n'importe quel fichier de base de données peut être enregistré sur n'importe quel espace disque.
- Séparer les objets avec un comportement différent sur différents tablespaces.
- Augmenter les performances de la base de données en séparant les composants de la base de données sur des disques différents.

OFA organise donc la base de données en fonction des types de fichiers et de leurs utilisations. Les fichiers binaires, de contrôles, de logs et administratifs peuvent être séparés sur des disques différents. L'administration en sera simplifiée et les performances grandement améliorées.

Oracle Software and File Locations

oracle_base	Software	oradata/	Files
/product		db01/	
/release_number		system01.dbf	
/bin		contro101.ctl	
/dba		redo0101.log	
/rdbms		...	
/sqlplus		db02/	
/admin		system01.dbf	
/inst_name		contro101.ctl	
/pfile		redo0101.log	
		...	

L'arborescence ci-dessus montre une structure de stockage selon le modèle OFA.

2.2. Oracle Enterprise Manager (OEM)

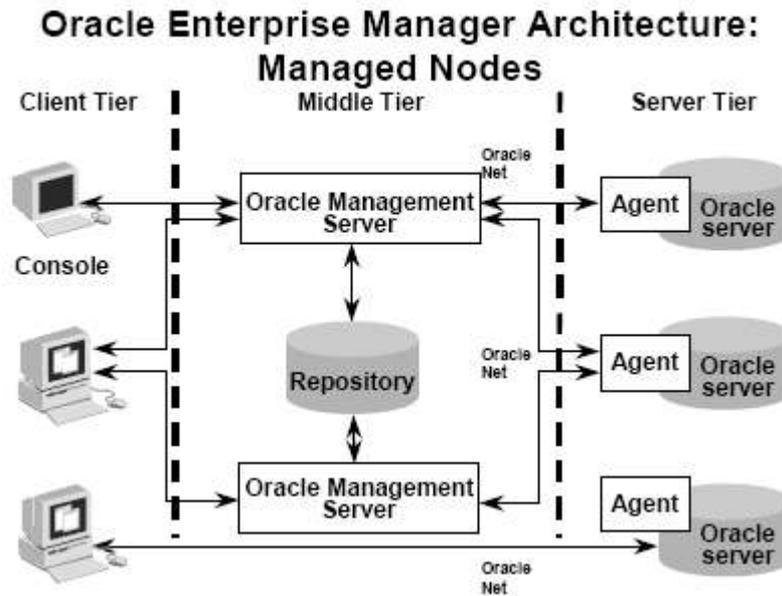
2.2.1. Architecture d'OEM : Vue d'ensemble

Oracle Enterprise Manager fournit un outil graphique de configuration système pour administrer une ou plusieurs bases de données.

OEM se présente sous la forme d'une architecture trois tiers, elle comprend :

- La console est un outil graphique, utilisés par les DBA pour administrer une ou plusieurs bases de données à partir d'une même console. La console OEM comporte des menus, des barres d'outils et elle peut intégrer des outils provenant d'autres fabricants. Elle permet d'accéder à un ensemble d'outils tel que la planification de tâches, la gestion des événements, la recherche de services sur le réseau et la sécurité.

- Les composants OMS forment le second tiers, les données de ces composants sont en effet stockées dans le repository OEM. Le repository OEM est un jeu de tables dans la base de données contenant les données système, les données applicatives et l'état des nœuds administrés sur l'environnement Oracle. La base de données stockant le repository OEM doit être accessible à la console OEM. Un repository est associé à des utilisateurs et non à une machine, ainsi, il devient indépendant de sa localité et peut être accédé de partout. Un repository OEM distinct peut être associé à chaque utilisateur. OMS peut donc effectuer toutes les tâches administratives, et ce de manière distribuée, de plus, plusieurs OMS peuvent être mis en place et agir en load balancing, ce qui augmente la tolérance de panne.



- Les Nœuds constituent la troisième couche de l'architecture qu'un utilisateur va chercher à administrer. Ces nœuds sont contrôlés par les « intelligent agents ». Un « intelligent agent » est un process qui s'exécute sur le nœud distant du réseau et qui permet de l'administrer à distance.

2.2.2. Composants de la console OEM

La console OEM est l'interface graphique utilisateur de l'application OEM, elle contient un ensemble de composants qui permettent à un ensemble de DBA d'administrer une ou plusieurs bases de données dans un environnement Oracle.

Elle se compose de menus permettant de lancer des applications d'administrations, des outils d'accès et de services et l'accès à l'ensemble des fonctionnalités de la console OEM.

Elle contient un ensemble d'icônes présentant sous forme de palettes les différents outils, et donnant un accès similaire aux éléments des menus de manière plus rapide.

Une fenêtre de navigation apporte une vue hiérarchique des différents nœuds du réseau, elle permet de visualiser l'ensemble des services Oracle à travers le réseau ainsi que les objets qu'ils contiennent. Les objets réseaux sont également représentés sur une carte graphique qui permet de les situer géographiquement.

La console OEM comprend également la gestion des tâches planifiées (Job) qui permettent de lancer à distance des tâches concernant les « listeners », les bases de données ou bien les hôtes. Un historique et un suivi de l'état des jobs sont également disponibles.

On peut également consulter les événements systèmes sur les différents nœuds et selon les critères que l'on souhaite. Le DBA peut alors être contacté par email ou autres. Certains événements peuvent faire l'objet de jobs correctifs qui seront lancés au déclenchement de celui-ci.

2.2.3. Services communs d'OEM

OEM comporte un ensemble de services permettant au DBA d'administrer les serveurs Oracle et leurs bases de données.

Les principaux services fournis par les OMS sont les suivants :

Le service de tâches panifiées (Job Service) :

Il gère les jobs programmés et permet donc d'automatiser des tâches répétitives.

Le service de gestion des événements (Event Service) :

Il permet de contrôler les événements systèmes survenant sur l'ensemble des serveurs Oracle du réseau, il permet de contrôler des limites applicatives et systèmes tel que l'espace occupé par les fichiers de données, la charge CPU ou bien des pertes de disponibilité d'un service.

Il peut, lors d'un événement donné, déclencher des jobs de récupération ou autres.

Le service de découverte (Discovery Service) :

C'est un assistant permettant de découvrir sur le réseau les différents services tel que les bases de données, les serveurs Web, les listeners, les machines, les « parallel Servers » et les serveurs de vidéo.

Le service de sécurité (Security Service) :

Il permet de contrôler l'accès aux services Oracle ainsi qu'aux nœuds du réseau. Un super administrateur peut également gérer les différents administrateurs et leurs privilèges.

Les authentifications sont encodées dans le repository OEM et les utilisateurs doivent spécifier leur identité pour accéder à chaque ressource à travers la console OEM.

2.2.4. Configurer OEM

Cette section présente la configuration d'OEM, lors de son installation, l'installateur universel lance l'assistant de configuration d'OEM, mais il est également possible de le lancer à nouveau lors de la création d'un repository supplémentaire.

Les étapes se découpent comme suit :

Création du repository OEM : L'assistant de configuration demande l'hôte, le port, et le SID du lieu où va être installé le repository.

Lancement du service OMS : La seconde étape consiste à lancer le service OMS, il démarre automatiquement au démarrage de Windows ou bien il peut être démarré manuellement.

Lancement de la console OEM : La console OEM peut être lancée via le menu « Démarrer ». Il est alors possible de se connecter au repository OEM et de découvrir automatiquement (Ceux où l'intelligent agent est lancé) ou manuellement les différents nœuds et services présents sur le réseau.

Configuration de la sécurité : La sécurité appropriée doit être mise en place pour contrôler l'accès aux différents nœuds et services du réseau. L'authentification pour un utilisateur doit être configurée pour chaque objet à accéder.

Travail sur une base de données : La dernière étape consiste à choisir une base de données via l'explorateur et s'authentifier pour pouvoir l'administrer.

2.3. Package d'administration

2.3.1. Pack de management DBA

Le pack de management DBA est un ensemble d'outils utilisés pour l'administration des bases de données. Il se compose de cinq applications et de deux jeux d'utilitaires et d'assistants :

- Instance Manager : Cet outil permet de monter, fermer et surveiller une base de donnée
- Security Manager : Il permet de contrôler l'accès aux services Oracle ainsi qu'aux nœuds du réseau. Un super administrateur peut également gérer les différents administrateurs et leurs privilèges.
- Storage Manager : Permet de surveiller les tablespaces, fichiers de données, segments de rollback.
- Schema Manager : Il permet de créer et maintenir des objets comme les tables, indexes et vues.
- SQL* Plus Worksheet : C'est une application en ligne de commandes permettant de lancer des requêtes SQL, du code PL/SQL ou encore des commandes SQL*Plus.

Les outils et les assistants de sauvegarde Oracle : Ils permettent de sauvegarder, restaurer ou bien recouvrer une base de données. On peut également gérer les fichiers de redo log grâce à ces outils. On lance ces outils à partir de la console ou du Gestionnaire de Schéma.

Les outils et assistants de gestion des données : Ils aident au chargement et l'organisation des données dans les bases de données. Ils permettent également le transfert des données à partir et vers une base de données Oracle. On lance ces outils à partir de la console ou du Gestionnaire de Stockage.

2.3.2.Packs de management supplémentaires

OEM peut intégrer dans sa console trois outils administratifs supplémentaires :

Le Pack de Diagnostic : Il fournit des utilitaires d'audit et de monitoring des bases de données, il devient possible de collecter et d'inspecter des performances spécifiques selon des statistiques aidant au paramétrage fin d'une base de données.

Le Gestionnaire de Performances Oracle permet de réaliser cela en temps réel et fournit des vues graphiques sur la performance de la base de données et du système.

Oracle TopSessions permet de lister l'ensemble des sessions connectées au serveur Oracle. On peut également voir les détails d'une session et y mettre fin.

Le planificateur de capacité permet de collecter des données et de stocker un historique des informations sur la base de données pour parer à l'augmentation de la capacité requise.

Le Visualisateur de données de trace Oracle permet de suivre et de programmer les informations de traçages de la base de données et de Net8.

Le Pack de Tuning : Il se compose d'un ensemble d'outils visant à améliorer les performances d'une base de données.

L'analyseur SQL permet d'analyser et d'éditer des requêtes SQL pour augmenter leurs performances.

Le Gestionnaire de Tablespace permet de monitorer et contrôler l'espace de stockage. Il fournit un ensemble d'informations sur les tablespaces et leurs segments. On peut défragmenter l'espace occupé et réorganiser les données.

Oracle Expert permet de tuner et d'optimiser le système de manière très fine.

Le Pack de Gestion du Changement : Il se compose d'utilitaires utilisés pour contrôler le changement de la définition des objets de la base de données.

L'application de Capture permet de capturer l'image d'un objet à un instant donné.

L'application de Différentiel permet de comparer les différences entre la définition d'objet pour deux base de données ou selon un modèle.

Les Objet de Changement rapide et de modification permettent de propager des changements aux objets d'une ou plusieurs bases de données.

L'application de Gestion des Plans permet de centraliser l'accès à tous ces outils.

The screenshot shows the Oracle TopSessions interface. The main window displays a table of sessions with columns: USERNAME, SID, OSUSER, SESSION..., COMMAND, STATUS, MACHINE, and PROGRAM. The sessions listed are:

USERNAME	SID	OSUSER	SESSION...	COMMAND	STATUS	MACHINE	PROGRAM
BACKGROUND...	5	SYSTEM	5545394	UNKNOWN	ACTIVE	WWEDF-175	ORACLE.EXE
SYSMAN	12	SYSTEM	132034	UNKNOWN	INACTIVE	WWED-DO...	ire -nojit -mx32m -D0mxThrP...
SYSMAN	17	SYSTEM	131806	UNKNOWN	INACTIVE	WWED-DO...	ire -nojit -mx32m -D0mxThrP...
SYSMAN	9	SYSTEM	20839	UNKNOWN	INACTIVE	WWED-DO...	ire -nojit -mx32m -D0mxThrP...
SYSMAN	13	SYSTEM	17071	UNKNOWN	INACTIVE	WWED-DO...	ire -nojit -mx32m -D0mxThrP...
SYSMAN	16	SYSTEM	11899	UNKNOWN	INACTIVE	WWED-DO...	ire -nojit -mx32m -D0mxThrP...
SYSMAN	18						x32m -D0mxThrP...
SYSMAN	15						x32m -D0mxThrP...
SYSMAN	26						x32m -D0mxThrP...
SYSMAN	25						x32m -D0mxThrP...

A pop-up window titled "SYSMAN - TopSessions Details" is open, showing the following details:

Name	Value
SADDR	00FF7000
SID	12
SERIAL#	625
AUDSID	952
PADDR	00FCC608
USER#	39
USERNAME	SYSMAN
COMMAND	UNKNOWN
OWNERID	2147483644
TADDR	
LOCKWAIT	

For Help, press F1. Press ESC to quit. Refreshed At: 14:06:56

3. Gérer une Instance Oracle

3.1. Authentification et privilèges

Les utilisateurs administrateurs de la base de données sont responsables de la gestion et de l'administration du serveur Oracle. Des privilèges particuliers sont requis pour permettre de faire ces tâches.

Deux comptes utilisateur DBA sont créés automatiquement et ont le rôle DBA :

Le compte SYS : Créé lors de l'installation d'Oracle avec tous les privilèges systèmes.

Mot de passe par défaut : `change_on_install`, à changer après l'installation.

Le schéma de SYS stocke toutes les tables et les vues du dictionnaire de base de données.

Aucun utilisateur ne peut modifier ou créer de table dans ce schéma.

Le compte SYSTEM : Créé lors de l'installation d'Oracle avec tous les privilèges systèmes.

Mot de passe par défaut : `manager` à changer après l'installation.

Le schéma de SYSTEM stocke les tables et les vues administratives ainsi que des informations administratives propres à certains produits Oracle supplémentaires.

3.1.1. Configurer l'authentification par le système d'exploitation

Des privilèges particuliers sont attribués aux utilisateurs DBA pour les tâches administratives. Pour s'assurer que seul des utilisateurs valides puissent se connecter à la base de données avec les privilèges adéquats, Oracle fournit deux méthodes d'authentification. Ces méthodes sont l'authentification par l'OS ou bien par un fichier de mots de passe (Méthode d'authentification propre à Oracle).

Lors d'une administration locale, les deux méthodes peuvent être utilisées au même titre.

Il en va de même lors d'une administration distante si vous avez une connexion sécurisée à la base de données.

Par contre, si la connexion n'est pas sécurisée, il est nécessaire d'utiliser l'authentification par fichier de mots de passe.

Dans le cas de l'authentification par le système d'exploitation, le nom d'utilisateur système du DB fait partie d'un groupe spécial ayant les privilèges d'administration. Si l'on place un compte utilisateur dans ce groupe, il hérite des droits DBA.

Configuration :

Sous Windows NT, il est possible de créer un groupe spécifique à chaque instance ou pour toutes les instances. Pour cela, il faut utiliser User Manager.

Groupes spécifiques à chaque instance :

ORA_<SID>_DBA et **ORA_<SID>_OPER** (Le <SID> étant le nom de l'instance).

Groupes non spécifiques à une instance :

ORA_DB et **ORA_OPER**.

Il est ensuite nécessaire d'éditer le fichier de paramètres pour préciser que l'authentification par fichier de mots de passe n'est pas nécessaire. Il suffit d'affecter la valeur NONE à la variable **REMOTE_LOGIN_PASSWORDFILE**.

Il est ensuite possible de se connecter avec les privilèges SYSDBA en tapant dans SQL*Plus Worksheet :

Connect / AS SYSDBA.

3.1.2. Configurer l'authentification par fichier de mots de passe

Lorsque la connexion à un serveur distant n'est pas sécurisée (Pas d'authentification par le système d'exploitation), la solution la plus sécurisée consiste à utiliser l'authentification Oracle (Ou authentification par fichier de mot de passe).

L'utilitaire ORAPWD.exe permet de créer un fichier de mot de passe.

C'est dans ce fichier que sont consignés les comptes et mots de passe des utilisateurs pour se connecter au schéma SYS.

Avec cette méthode d'authentification, l'utilisateur SYS peut utiliser la commande GRANT pour fournir des privilèges DBA à d'autres utilisateurs Oracle.

Le contenu du fichier de mot de passe est stocké dans un format crypté et non lisible par des utilisateurs Oracle d'une autre base de données. Ce fichier est un fichier caché se trouvant sous Windows NT dans le répertoire %ORACLE_HOME%\DATABASE. Sous UNIX, c'est dans le répertoire \$ORACLE_HOME/dbs que l'on peut le trouver.

La commande suivante doit être utilisée pour configurer ce mode d'authentification :

```
ORAPWD FILE=<fname> PASSWORD=<password> ENTRIES=<entries>
```

Ou :

<fname> est le nom et le chemin complet du fichier de mot de passe.

<password> est le mot de passe pour les comptes SYS et INTERNAL.

<entries> est le nombre maximum d'utilisateurs qui peuvent se connecter en tant que SYSDBA et SYSOPER.

Le paramètre REMOTE_LOGIN_PASSWORDFILE peut être positionné avec la valeur EXCLUSIVE ou SHARED.

En mode EXCLUSIVE, une seule instance peut utiliser le fichier de mot de passe et il peut contenir d'autres noms d'utilisateurs que SYS et SYSTEM.

En mode SHARED, plusieurs instances peuvent se partager le fichier de mots de passe. Par contre, seul l'utilisateur SYS est reconnu.

3.2. Fichiers de paramètres

3.2.1. Paramètres présents dans le fichier de paramètre

Une base de données Oracle devient accessible aux utilisateurs lorsque le DBA démarre l'instance et ouvre la base de données. Pour démarrer une instance, Oracle lit le fichier d'initialisation. Ces fichiers sont au nombre de deux (le dernier est une nouveauté de la 9i) : le PFILE et le SPFILE. Le PFILE fait communément référence au fichier init<SID>.ora et le SPFILE (fichier binaire) fait référence au fichier spfile<SID>.ora où SID est le nom de l'instance.

Le fichier de paramètre est un fichier texte contenant la liste des paramètres de démarrage de l'instance.

3.2.2. Le PFILE (initSID.ora)

Sous Windows NT, ce fichier se trouve par défaut dans le répertoire %ORACLE_HOME%\database et contient un paramètre IFILE qui fait référence à un fichier plus complexe situé dans le répertoire %ORACLE_HOME%\admin\<SID>\pfile.

Sous UNIX, on peut le trouver dans le répertoire \$ORACLE_HOME/dbs.

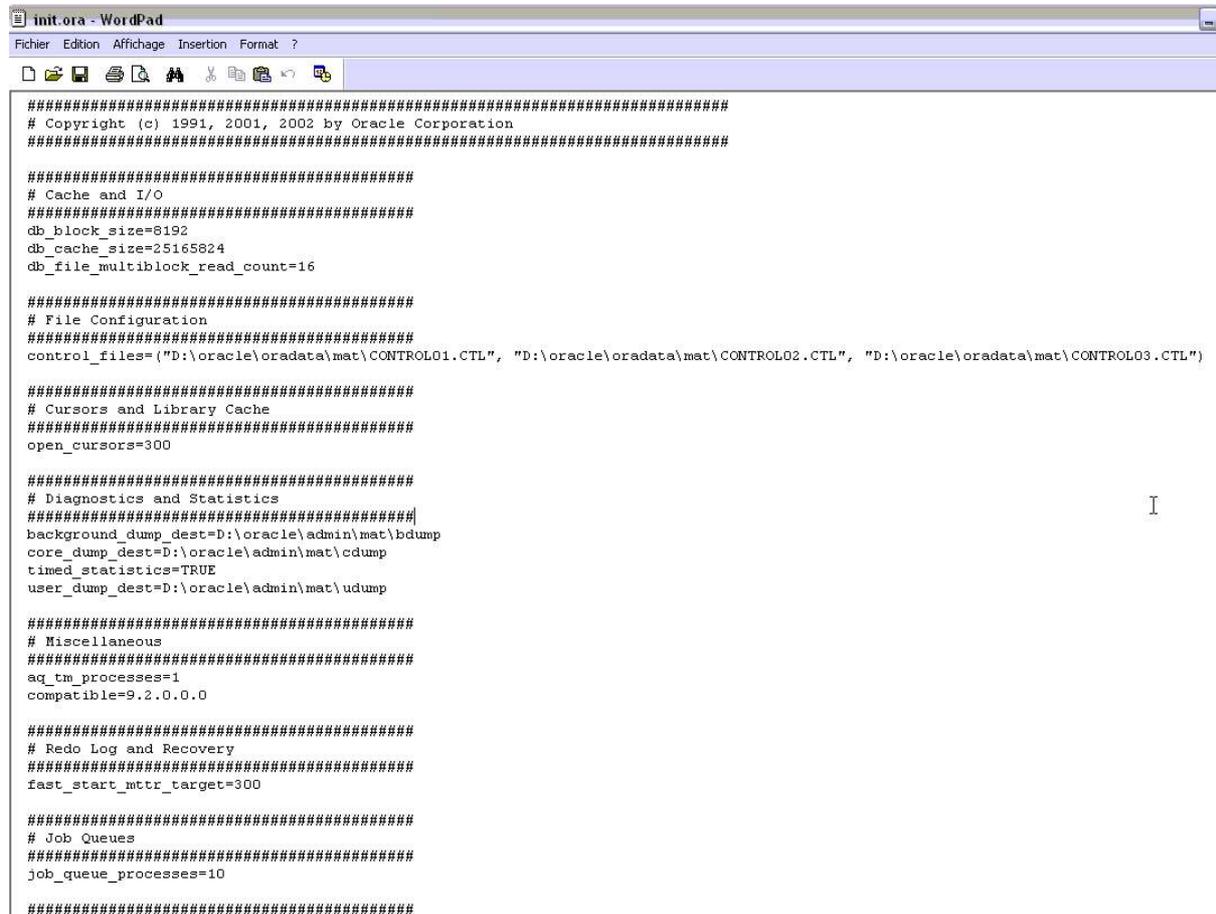
Ce fichier est lu lors du démarrage de l'instance, on peut l'éditer avec n'importe quel éditeur de texte et le redémarrage de l'instance est nécessaire pour que les changements prennent effet.

Il est préférable de faire une copie de ce fichier sur l'ordinateur où se trouve OEM s'il en existe un. Parmi les paramètres du fichier de paramètre, on trouve le nom de la base de données pour laquelle l'instance est démarrée.

Les paramètres contrôlent les performances de la base de données ainsi que la quantité de mémoire utilisée par les différentes composantes de la SGA.

On peut également définir certains attributs physiques de la base de données au moment de sa création telle que la taille des blocs de données.

On trouve également dans ce fichier le nom et le chemin des fichiers de contrôle, des fichiers de log archivés et des fichiers de trace de la base de données.



```
#####
# Copyright (c) 1991, 2001, 2002 by Oracle Corporation
#####

#####
# Cache and I/O
#####
db_block_size=8192
db_cache_size=25165824
db_file_multiblock_read_count=16

#####
# File Configuration
#####
control_files=('D:\oracle\oradata\mat\CONTROL01.CTL', 'D:\oracle\oradata\mat\CONTROL02.CTL', 'D:\oracle\oradata\mat\CONTROL03.CTL')

#####
# Cursors and Library Cache
#####
open_cursors=300

#####
# Diagnostics and Statistics
#####
background_dump_dest=D:\oracle\admin\mat\bdump
core_dump_dest=D:\oracle\admin\mat\cdump
timed_statistics=TRUE
user_dump_dest=D:\oracle\admin\mat\udump

#####
# Miscellaneous
#####
aq_tm_processes=1
compatible=9.2.0.0

#####
# Redo Log and Recovery
#####
fast_start_mttr_target=300

#####
# Job Queues
#####
job_queue_processes=10

#####
```

3.2.3. Le SPFILE (spfileSID.ora)

Le spfile est un fichier binaire de paramètres dynamiques qui est géré par le serveur Oracle. Il se situe dans le même répertoire que le initSID.ora et contient aussi les mêmes données. Si le fichier spfileSID.ora existe, il est prioritaire par rapport au PFile. En effet, au démarrage de l'instance le serveur essaiera de lire le spfile plutôt que le pfile, le pfile sera lu seulement en cas d'échec de la lecture du spfile.

Il est possible de créer le spfile à partir du pfile avec la commande suivante :

```
CREATE SPFILE = 's_params.ora'
FROM PFILE = '$ORACLE_HOME/work/t_init1.ora'
```

Comme pour le pfile il est possible de modifier les paramètres d'initialisation avec la commande :

```
ALTER SYSTEM SET paramètre = valeur
[COMMENT = 'moncommentaire']
[SCOPE = MEMORY | SPFILE | BOTH]
[DEFERRED]
```

Où MEMORY : en mémoire uniquement (par défaut comme la 8i).

SPFILE : uniquement dans le SPFILE.

BOTH : en mémoire et dans le SPFILE.

DEFERRED : pour les paramètres dynamiques, les active pour les sessions futures (pas celles en cours).

COMMENT : commentaire associer au paramètre de longueur maximale de 255 caractères.

Remarque : Le contenu du spfile est visible dans la vu dynamique **V\$SPPARAMETER** et non pas **V\$PARAMETER** qui est utilisé par le pfile.

3.2.4.Règles pour les valeurs de paramètre

Un ensemble de règles régissent les paramètres compris dans le fichier de paramètre :

- La plupart des paramètres sont optionnels.
- Leur ordre n'est pas important.
- Le format pour définir un paramètre est de la forme **nom_de_paramètre=valeur** et s'écrit sur une ligne.
- La plupart des paramètres ont une valeur par défaut.
- Si un paramètre comprend des espaces ou des tabulations, il doit être mis entre doubles quotes.
- Certain paramètres tel que CONTROL_FILE peuvent prendre plusieurs valeurs, elles doivent être mises entre parenthèses et séparées par des virgules.
- Des lignes de commentaires peuvent être incluses dans le fichier et doivent commencer par le symbole #.
- Le mot clé IFILE est utilisé pour inclure d'autres fichiers à un fichier de paramètre.
- Si le système d'exploitation est sensible à la casse, le fichier de paramètre le devient également.

3.2.5.Paramètres essentiels

La plupart des paramètres sont optionnels, cependant certains paramètres sont au contraire essentiels pour l'instance Oracle.

La liste suivante les présente :

DB_NAME : Identifiant de la base de données.

DB_CACHE_SIZE : Ne peut être mis à 0. Gère la taille du buffer par défaut.

DB_KEEP_CACHE_SIZE : Gère la taille du keep buffer cache, utilisé pour stocker les blocks dans la mémoire susceptible d'être réutilisés.

DB_RECYCLE_CACHE_SIZE : Gère la taille du recycle buffer cache, employé pour supprimer les blocs de mémoire réutilisé qui ont subit peu de changement.

DB_BLOCK_SIZE : Taille d'un bloc de données spécifié lors de la création de la base de données. Ne peut être changé après la création.

COMPATIBLE : Version du serveur avec laquelle l'instance pourra être compatible.

CONTROL_FILES : Spécifie le nom des fichiers de contrôle de la base de données.

SHARED_POOL_SIZE : Spécifie la taille du pool partagé en bytes.

BACKGROUND_DUMP_DEST : Spécifie le répertoire ou le process de fond de taches écrit les fichiers de trace et les fichiers de log d'alertes.

USER_DUMP_DEST : Spécifie le lieu où les fichiers de trace utilisateurs sont créés.

3.2.6. Paramètres communément modifiés

Les paramètres le plus souvent modifiés par le DBA pour une instance donnée sont les suivants :

IFILE : Permet d'inclure un autre fichier de paramètres dans le fichier courant. Peut être utilisé jusqu'à trois niveaux.

LOG_BUFFER : Spécifie le nombre de bytes qui sont alloués pour le buffer de redo log dans la SGA.

MAX_DUMP_FILE_SIZE : Spécifie la taille maximale en bloc des fichiers de trace.

PROCESSES : Définit le nombre maximum de process du système d'exploitation pouvant être simultanément connecté à l'instance.

SQL_TRACE : Permet d'activer ou non l'utilitaire de trace SQL pour toutes les sessions utilisateurs.

TIMED_STATISTICS : Permet d'activer ou non la datation des événements dans les fichiers de trace et les écrans de monitoring.

3.3. Démarrage et arrêt d'une instance

3.3.1. Etapes de démarrage d'une instance

Le démarrage d'une instance se fait en plusieurs étapes. Elles sont au nombre de trois et se présentent tel que suit : le démarrage de l'instance, le montage de la base de données et l'ouverture de la base de données.

Le démarrage de l'instance est la première étape, on parle de **mode NOMOUNT**. On démarre ce mode lors de la création de la base de données ou la régénération des fichiers de contrôle.

C'est à ce moment que le serveur va essayer de lire les fichiers de paramètres dans l'ordre suivant :

- le spfileSID.ora. Si il n'est pas trouvé alors,
- le spfile.ora (fichier de paramètres crée automatiquement avec une base de données). Sinon,
- le initSID.ora

La SGA est allouée et les process démarrés. Les fichiers de trace et d'alertes sont également ouverts.

La seconde étape consiste à monter la base de données. C'est le **mode MOUNT**. La base de données n'est pas encore ouverte et donc non accessible.

Ce mode est utilisé pour certaines taches de maintenance de la base de données.

A ce moment, la base de données est associée à une instance démarrée, le serveur Oracle localise les fichiers de contrôle et les ouvres. Ces fichiers sont alors lus pour obtenir le nom et le statut des fichiers de données et de redo log.

La troisième étape de démarrage est l'ouverture de la base de données. On l'appelle également le **mode OPEN** ou mode normal de la base de données. A ce moment, un utilisateur valide de la base de données peut se connecter et accéder aux données.

Lors de cette étape finale, le serveur Oracle vérifie que les fichiers de données et de redo log puissent être ouverts. Si l'ouverture de l'un d'entre eux échoue, une erreur est renvoyée et la base de données n'est pas ouverte.

La consistance de la base de données est également vérifiée pour s'assurer que les fichiers de données soient bien synchronisés.

Si nécessaire, le process d'arrière plan SMON lance la restauration d'instance. Les derniers checkpoints sont identifiés, les fichiers de Rollback et les données utilisateurs sont mises à jour, les transactions non committées font l'objet d'un rollback

3.3.2. Démarrer une instance

Nous allons voir comment démarrer de manière pratique une instance. Il est possible au DBA de démarrer une instance soit par la commande startup soit par le Manager d'Instance Oracle.

```
STARTUP [FORCE] [RESTRICT] [PFILE=Filename]
        [EXCLUSIVE | PARALLEL | SHARED]
[OPEN [RECOVER [database] | MOUNT | NOMOUNT]
```

Via le Manager d'Instance Oracle, il faut se connecter avec les privilèges DBA. Il devient alors possible de démarrer l'Instance avec les trois modes NOMOUNT, MOUNT ou OPEN.

Il est possible de préciser le fichier de paramètre ou encore d'ouvrir la base de données en mode restreint.

Sous Windows NT, une instance Oracle est assimilée à un service. Le service démarre automatiquement et permet de démarrer l'instance.

3.3.3. Changer la disponibilité de la Base de données

Un DBA peut démarrer une instance Oracle dans l'un des trois modes NOMOUNT, MOUNT et OPEN. Ce mode peut être changé de l'un à l'autre lorsque l'instance a été lancée (de NOMOUNT vers MOUNT puis VERS OPEN).

Il est possible de le changer soit grâce à la commande ALTER DATABASE ou grâce au Manager d'Instance Oracle.

En ligne de commande et en mode NOMOUNT, il est possible de taper ceci :

```
ALTER DATABASE {MOUNT | OPEN}
```

Il est possible d'agir sur le mode de la base de données à l'aide du Manager d'Instance Oracle. On peut par exemple changer l'état de la base de données de NOMOUNT à MOUNT. Cela se fait à l'aide du menu Base de données.

3.3.4. Etapes d'arrêt d'une instance

Il y a trois étapes dans le process d'arrêt d'une instance et de la base de données :

La base de données est d'abord fermée : Lors de cette étape, le serveur Oracle écrit tous les changements des buffers caches dans les fichiers de données.

Tous les buffers de redo log sont écrits dans les fichiers de redo log en ligne.

Ces mêmes fichiers sont ensuite fermés.

La base de données devient indisponible aux utilisateurs mais les fichiers de contrôle restent ouverts.

Elle est ensuite démontée : La base de données est ensuite démontée de son instance. Les fichiers de contrôle sont fermés à leur tour.

L'instance est finalement arrêtée : Les fichiers de traçage et d'alerte sont fermés. La SGA est désallouée de la mémoire et les process de tâche de fond sont terminés.

3.3.5. Modes d'arrêts

Un DBA peut fermer une base de données dans l'un des quatre différents mode disponibles. Ces modes sont : normal, transactionnel, immédiat et abandon.

Le **mode normal** est le mode par défaut, les connexions à la base de données deviennent impossibles, toutefois, le serveur Oracle attend que les utilisateurs connectés se déconnectent de la base de données. Une fois tous les utilisateurs déconnectés, le serveur Oracle démonte et ferme la base de données puis l'instance. Dans ce mode de fermeture, une restauration de l'instance ne sera pas nécessaire au redémarrage.

Dans le **mode transactionnel**, les clients ne peuvent perdre leur travail. Lors de l'utilisation de ce mode, aucun utilisateur ne peut lancer une nouvelle transaction. Les clients sont déconnectés lorsque leurs transactions en cours se terminent. La base de données est ensuite immédiatement fermée. La restauration d'instance ne sera pas nécessaire dans ce mode également.

Le **mode d'arrêt immédiat** entraîne l'arrêt immédiat des requêtes SQL en cours, les utilisateurs sont déconnectés sans attente. Le serveur Oracle rollback annule les transactions actives. Une restauration de l'instance ne sera pas nécessaire.

Le **mode abandon (ABORT)** est utilisé seulement lorsque les autres modes ne peuvent fonctionner. (Ce mode est exécuté lorsque la commande STARTUP FORCE est utilisée).

Les requêtes SQL sont terminées et les utilisateurs déconnectés, les transactions non commitées ne font pas l'objet d'un rollback. Le prochain démarrage de l'instance nécessitera une restauration de l'instance par SMON. En mode abandon, les fichiers ne sont pas fermés.

```
SHUTDOWN [NORMAL | TRANSACTIONAL | IMMEDIATE | ABORT]
```

3.4. Vue dynamique des performances

3.4.1. Caractéristiques des vues dynamiques de performance

Le serveur Oracle maintient un jeu de tables virtuelles qui enregistre l'activité courante de la base de données. Ces tables sont appelées les tables de performances dynamiques. Elles sont visibles pour les utilisateurs via les vues dynamiques de performances. Ces tables appartiennent au DBA et ne sont pas accessibles à la plupart des utilisateurs. Les vues sont également accessibles au DBA qui peut donner le droit d'accès à d'autres utilisateurs.

Ces vues sont identifiées avec le préfixe V_\$ mais le serveur Oracle créé un synonyme public avec le préfixe V\$.

Les tables sous-jacentes à ces vues sont constamment mises à jour lorsque la base de données est en cours d'utilisation. Les informations les plus à jours deviennent alors disponibles à travers ces vues. Ces vues sont utilisées pour fournir des données relatives aux performances telles que des informations sur les fichiers de données et les structures de la mémoire.

Lorsqu'une instance est démarrée en mode NOMOUNT, seul les vues lisant des données de la mémoire sont accessibles.

Lorsque la base de données est en mode MOUNT, les vues lisant les données des fichiers de contrôle sont alors accessibles. La vue V\$FIXED_TABLE montre les vues dynamiques de performance.

3.4.2. Descriptions des vues dynamiques de performance

Les principales vues accessibles en mode NOMOUNT lisent des données de la mémoire et sont les suivantes :

V\$PARAMETER: Liste les informations sur les paramètres d'initialisation avec leur nom, leur nombre, leur valeur et leur type.

V\$SPPARAMETER : Liste les informations sur les paramètres d'initialisation du serveur avec leur nom, leur nombre, leur valeur et leur type.

V\$SGA : Donne des informations récapitulatives sur la SGA.

V\$OPTION : Liste les options installées avec le serveur Oracle.

V\$PROCESS : Contient des informations sur les process actifs.

V\$SESSION : Donne les informations sur la session courante.

V\$VERSION : Liste le numéro de version et les composants du serveur Oracle.

V\$INSTANCE : Donne l'état de l'instance courante.

Les vues lisant les données des fichiers de contrôle sont accessibles seulement lorsque la base de données est montée. Les principales vues sont les suivantes :

V\$THREAD : Présente les informations sur les threads des fichiers de contrôle tel que les informations sur les groupes de redo log.

V\$CONTROLFILE : Fournit les noms des fichiers de contrôle.

V\$DATABASE : Contient des informations sur la base de données tel que le nom ou la date de création.

V\$DATAFILE : Donne les informations sur les fichiers de données tel que leur nom, leur statut et d'autres détails.

V\$DATAFILE_HEADER : Donne des informations sur les en-têtes des fichiers de contrôle.

V\$LOGFILE : Contient des informations sur les fichiers de redo log en ligne

3.5. Paramètres dynamiques

3.5.1. Afficher les valeurs des paramètres courants

Les paramètres d'initialisation contrôlent la configuration de l'instance et de la base de données, il est possible d'afficher ces paramètres grâce à la commande `SHOW PARAMETER` ou bien via la vue `V$PARAMETER`.

La commande `SHOW PARAMETER` affiche les informations sur les paramètres de la base de données lancée. Ils sont présentés par ordre alphabétique avec leur valeur.

La commande peut également afficher un paramètre donné en spécifiant son nom, par exemple :

```
SHOW PARAMETER control
```

Il est possible d'interroger la vue `V$PARAMETER` grâce à une requête SQL tel que :

```
SELECT name, value
FROM V$PARAMETER
WHERE name LIKE '%control%';
```

Il est par ailleurs possible de consulter la valeur des paramètres via le noeud des paramètres d'initialisation dans le Manager d'Instance Oracle.

Note : Il est nécessaire de redémarrer l'instance Oracle pour que les changements apportés aux paramètres du fichier `init.ora` soient effectifs.

3.5.2. Paramètres d'initialisation dynamique

Ces paramètres peuvent être modifiés par :

Les commandes SQL:

```
ALTER SESSION SET <param>=<value> ;
```

```
ALTER SYSTEM SET <param>=<value> [DEFERRED] ;
```

Soit l'on utilise la commande ALTER SESSION pour modifier les paramètres de la session courante, soit l'on utilise le ALTER SYSTEM pour toutes les sessions. La valeur devient effective immédiatement et jusqu'à l'arrêt de l'instance. L'option DEFERRED permet de modifier la valeur d'un paramètre uniquement pour les sessions à venir.

On peut vérifier la valeur d'un paramètre soit par la vue V\$PARAMETER (Pour les paramètres de session) ou V\$SYSTEM_PARAMETER (Pour les paramètres système). Les champs ISSUES_MODIFIABLE, ISSYS_MODIFIABLE, ISMODIFIED et ISADJUSTED sont les champs de ces vues concernant le statut de ces paramètres.

Le Manager d'Instance Oracle : En lançant la console, il faut cliquer sur le nœud des paramètres d'initialisation. Les paramètres sont affichés sur la droite et leurs valeur peuvent être éditées.

3.6. Gérer les sessions

3.6.1. Activer les modes Restreint et Lecture seule

Lors de la réalisation de tâches administratives, la base de données peut être placée en mode restreint. Cela peut être fait au démarrage et permet de laisser l'accès à la base de données seulement aux utilisateurs ayant le privilège "RESTRICTED SESSION".

La base de données peut être changée en mode restreint via la commande ALTER SYSTEM ou le Manager d'Instance Oracle.

La commande est la suivante :

```
{ENABLE | DISABLE} RESTRICTED SESSION
```

Les sessions en cours ne sont pas déconnectées, le mode restreint n'interviendra que pour les futures connexions.

Il est possible de voir les informations sur le mode restreint, en utilisant la vue V\$INSTANCE.

Dans le Manager d'Instance Oracle, il faut utiliser le menu "Session" et sélectionner l'option "Restrict". Il faudra choisir l'option "Allow All" pour revenir en mode normal.

Il est également possible d'ouvrir la base de données en mode "Lecture seule". Les utilisateurs peuvent alors accéder aux données, mais ne peuvent les modifier.

Pour cela, on peut utiliser la commande SQL suivante :

```
ALTER DATABASE OPEN READ ONLY
```

3.6.2. Terminaison de Session

Lors de certaines tâches administratives, le DBA doit parfois déconnecter les utilisateurs en cours. En effet, les modes restreints vus précédemment ne déconnectent pas les sessions en cours.

La commande SQL suivante déconnectera les utilisateurs ayant des sessions en cours :

```
ALTER SYSTEM KILL SESSION 'SID, SERIAL#' [POST_TRANSACTION]
```

Les paramètres sont le SID et le numéro de série ils permettent à eux deux d'identifier une session de manière unique. On peut les retrouver en consultant la vue V\$SESSION.

Il est également possible de mettre fin à une session via le Manager d'Instance Oracle. Le noeud concerné est le noeud "Sessions", il est possible de mettre fin à l'une d'elle en cliquant dessus pour la sélectionner, puis en sélectionnant dans le menu "Session", le sous-menu "Disconnect" et soit "Immediate", soit "Post Transaction".

Lorsqu'une session est terminée, le process de tâche de fond PMON rollback annule la transaction courante. Toutes les tables et les enregistrements verrouillés sont libérés. Le message ORA-00028 est

envoyé à l'utilisateur pour l'informer de la fin de la session. La colonne STATUS de la vue V\$SESSION est marquée comme "killed".

3.7. Configuration d'Oracle Managed Files (OMF)

OMF est géré par le SGBD et permet de simplifier l'administration en éliminant le besoin de gérer directement les fichiers dans la base de données, notamment ceux de contrôle, de données, et de redo logs.

OMF crée un nom unique pour chaque fichier et se charge de les supprimer automatiquement en cas de suppression logique. Ainsi, les répertoires physiques sont gérés par 2 paramètres dynamiques :

- DB_CREATE_FILE_DEST : utilisé pour les fichiers de données
- Db_CREATE_ONLINE_LOG_DEST_N (1<=n<=5) : utilisé pour les fichiers de contrôle et les redo logs.

La commande suivante peut être lancée pour définir le premier paramètre :

```
ALTER SYSTEM SET db_create_file_dest = '/oracle/dbs/files';
```

Si seul le premier paramètre est initialisé, alors tous les fichiers (contrôle, données et redo) seront localisés dans le même répertoire

Créer une base de données où les fichiers de données, de contrôles et les redo log sont créés dans des répertoires séparés :

La première étape consiste à créer l'arborescence pour les fichiers de paramètres :

```
DB_CREATE_FILE_DEST = '/u01/oradata'  
DB_CREATE_ONLINE_LOG_DEST_1 = '/u02/oradata/'  
DB_CREATE_ONLINE_LOG_DEST_2 = '/u03/oradata/'
```

Le paramètre DB_CREATE_FILE_DEST initialise le répertoire par défaut pour les fichiers de données.

Les paramètres DB_CREATE_ONLINE_LOG_DEST_1 et DB_CREATE_ONLINE_LOG_DEST_2 initialisent les répertoires pour les fichiers de redo log et les fichiers de contrôles.

La dernière étape consiste à créer la base de données en elle-même.

3.8. Fichier de Log et d'Alertes

3.8.1. Fichiers de Log

Les fichiers de Log et d'Alerte permettent au DBA de maintenir et d'augmenter les performances du serveur Oracle.

Les fichiers de trace sont utilisés pour consigner les erreurs internes. Elles contiennent également des informations pour tuner les applications et les instances. Ce sont les processus de fond qui inscrivent les erreurs détectées dans ces fichiers. Les processus serveur peuvent être également tracés si le traçage SQL est activé.

On peut activer le traçage SQL selon les deux méthodes suivantes :

Placer la valeur du paramètre d'initialisation SQL_TRACE à TRUE en utilisant la commande SQL :

```
ALTER SESSION SET sql_trace=TRUE;
```

Le paramètre d'initialisation BACKGROUND_DUMP_DEST spécifie le chemin des fichiers de trace générés par le processus d'arrière plan.

Le paramètre d'initialisation `USER_DUMP_DEST` spécifie le chemin des fichiers de trace générés par les process serveur.

La taille des fichiers de trace est spécifiée par le paramètre `MAX_DUMP_FILE_SIZE`.

3.8.2.Fichiers d'Alerte

Les fichiers d'alerte de la base de données sont un archivage chronologique des messages et des erreurs. Sous Windows NT, le fichier est nommé `<SID>ALRT.LOG` et `alert_<SID>.log` sous UNIX.

Ces fichiers sont créés lors du démarrage de la base de données s'ils n'existent pas.

Ces fichiers permettent de contrôler si une instance fonctionne normalement, les erreurs se produisant lors de son fonctionnement sont écrites dans ces fichiers.

Le chemin de ces fichiers est spécifié par le paramètre `BACKGROUND_DUMP_DEST`. Par défaut, le chemin sous Windows NT est `%ORACLE_HOME%\RDBMS\TRACE` et `$ORACLE_HOME/rdbms/log` sous UNIX.

Leur maintenance doit être faite à la main. Ainsi, leur sauvegarde et leur suppression sont à la charge du DBA. Ils peuvent être supprimés même lorsqu'une instance est lancée.

Les informations contenues dans ces fichiers contiennent toutes les informations sur les erreurs internes, les blocs corrompus, et les erreurs de « deadlock » (verrous morts). Les opérations Oracle affectant les structures et les paramètres, de la base de données, sont également consignées. On retrouve également les actions telles que le lancement ou l'arrêt d'une instance ou encore les paramètres d'initialisation dont la valeur n'est pas la valeur par défaut.

4.National Language SuppOrt (NLS)

4.1.Jeu de caractères

4.1.1.Caractéristiques d'NLS

Le support national de langage permet aux utilisateurs d'interagir avec une base de données dans leur langue d'origine et de lancer des applications dans des environnements comportant différents langages.

NLS permet aux utilisateurs d'interagir, de stocker, d'interroger et de manipuler des données dans leur langue d'origine. Il est inclus le support de langues tel que celles d'Europe de l'Ouest, de l'Europe de l'Est, de l'Europe Centrale ou de l'Asie. Les jeux de caractères sont également propres à différents pays ou régions.

Des jeux de caractères sur un bit, plusieurs bits variant ou plusieurs bits fixes sont supportés par NLS. Des séquences de tri correctes sont également disponibles pour les différentes langues.

La traduction des messages d'erreurs et des interfaces des utilitaires est également assurée dans plus de 40 langues.

Les formats de date et d'heure sont également permis en accord avec les standards ISO.

Les calendriers tel que le calendrier Grégorien, Japonais, Impérial, et Thai Budda sont également supportés.

Les valeurs numériques sont également représentées dans le format local approprié.

Les symboles monétaires, les crédits et les débits sont également supportés selon les normes ISO selon les pays.

Cependant, les mots clés sont uniquement spécifiés en anglais. De même, le jeu de caractères de la base de données ne peut être de longueur fixe sur plusieurs caractères.

Note : Les jeux de caractères de longueur variable stockent les données de manière plus efficace que les jeux de caractères de longueur fixe.

4.1.2.Plans d'encodage de caractères

L'affichage d'un caractère sur une station de travail dépend des conventions de représentation qui sont supportées par celle-ci. Le code numérique spécifié pour un caractère définit le plan d'encodage de caractères.

Plusieurs types d'encodage de caractères sont disponibles :

Bit simple : Chaque caractère occupe un bit (Utilisé pour l'Europe et les langues du moyen Orient). Ils sont codés sur 7 ou 8 octets.

Sur 7 octets, il est possible de coder jusqu'à 128 caractères, ce qui permet de coder l'alphabet.

Sur 8 octets, jusqu'à 256 caractères sont utilisés, cela permet de coder plusieurs langues telles que celles de l'Europe de l'Ouest.

Bit multiple à longueur variable : Un caractère est cette fois-ci codé sur un ou plusieurs bits tel que pour les langues asiatiques.

Longueur fixe à bits multiples : Les informations stockées sont similaires au précédent type de codage, il est seulement toujours utilisé le même nombre de bits pour l'encodage.

Unicode : C'est un codage qui représente la totalité des caractères informatiques possible, ainsi que les caractères de symboles techniques et de publication. 38 885 caractères sont représentés. Les deux principaux formats sont l'UCS2 (Deux bits fixes) et l'UTF8 (Longueur variable).

4.1.3. Caractéristiques du jeu de caractères

Lors de la création d'une base de données, vous devez sélectionner des jeux de caractères adaptés à votre base de données.

Il est tout d'abord préférable de sélectionner un jeu de caractères correspondant au plan d'encodage de caractères supportés par le système d'exploitation. Il peut toutefois être différent de celui-ci.

Une base de données Oracle utilise un jeu de caractères de la base de données et un jeu de caractères National. Ces deux jeux de caractères sont configurés lors de la création de la base de données et ne peuvent être changés ensuite. Ils ont les caractéristiques suivantes :

Jeu de caractères de la base de données	Jeu de caractères National
Défini à la création de la base de données.	Défini à la création de la base de données.
Données des colonnes stockées pour les types CHAR, VARCHAR2, CLOB et LONG.	Données des colonnes stockées pour les types NCHAR, NVARCHAR2 et NCLOB (Le type LONG n'est pas supporté).
Stocke les jeux de caractères encodés sur plusieurs bits en taille variable.	Stocke les jeux de caractères encodés sur plusieurs bits en taille variable. Stocke les jeux de caractères encodés sur plusieurs bits de taille fixe.

Les considérations suivantes sont à prendre en compte pour choisir les jeux de caractères :

Sélectionner un jeu de caractères proche pour celui de la base de données et le National.

Evaluer les performances pour les opérations sur les chaînes de caractères. Les recherches et comparaisons sur les chaînes de caractères sont plus rapides si l'encodage du jeu de caractères est fixe pour le jeu de caractères national.

Ne pas oublier que les jeux de caractères encodés sur une taille variable sont plus économes en place.

4.2. Comportement dépendant du langage

4.2.1. Paramètres NLS côté serveur

Les paramètres NLS permettent de spécifier la langue, les noms de mois et des jours, le symbole de la monnaie locale, et d'autres attributs.

Il est possible de configurer ces paramètres selon trois méthodes :

Via les paramètres d'initialisation : Il est possible de configurer via les paramètres d'initialisation un comportement par défaut dépendant du langage pour le serveur. Cette configuration n'aura pas d'impact sur les clients.

Via les variables d'environnement : En configurant un langage via les variables d'environnement, il est possible de configurer le comportement d'une session cliente et de prendre le pas sur la configuration du serveur.

Via la commande ALTER SESSION : Elle permet de configurer le langage soit pour la session, soit pour le serveur.

Les paramètres d'initialisation sont :

NLS_LANGUAGE : pour les messages d'erreur serveur.

Les messages sont stockés dans des fichiers binaires et sont nommés selon la convention <Nom de produit><ID de langage>.MSB (Par exemple ORAF.MSB pour un Oracle Français).

Ce paramètre agit sur :

La langue des mois et jours dans une date.

Les symboles pour les heures et les siècles.

Le tri alphabétique du jeu de caractères.

NLS_TERRITORY : Il définit les caractéristiques pour :

Les caractéristiques du formatage des dates et des nombres.

Les symboles monétaires et ISO.

Les caractères décimaux, les séparateurs de groupes, de listes, le jour de début de la semaine, le symbole des crédits et débits, et le calcul ISO des numéros de jour.

Ils peuvent être configurés via le fichier de paramètres d'initialisation ou bien par la commande ALTER SESSION.

4.2.2. Paramètres NLS dérivés

Les paramètres NLS dérivés permettent d'outrepasser la configuration par défaut des paramètres tels que NLS_LANGUAGE et NLS_TERRITORY.

Les paramètres dérivés de NLS_LANGUAGE sont :

NLS_DATE_LANGUAGE pour l'affichage des jours et des mois.

NLS_SORT pour le tri des caractères.

Les paramètres dérivés de NLS_TERRITORY sont :

NLS_CURRENCY pour le symbole monétaire local à renvoyer pour le masque L.

NLS_ISO_CURRENCY pour le symbole monétaire local ISO à renvoyer pour le masque C.

Par exemple, le symbole monétaire ISO pour le dollar américain est USD ou AUD pour le dollar australien.

NLS_DATE_FORMAT pour le format du masque de date utilisé avec les fonctions TO_CHAR et TO_DATE, il doit être mis entre double guillemets.

NLS_NUMERIC_CHARACTERS définit le caractère décimal (Qui sépare la partie entière de la partie décimale d'un nombre) et le séparateur de groupes (Qui sépare les groupes d'entier pour les milliers, les millions, etc. Ces deux caractères doivent être différents et les opérateurs arithmétiques ainsi que les nombres ne sont pas autorisés.

Ces paramètres sont modifiables via les paramètres d'initialisation, les variables d'environnement ou la commande ALTER SESSION et permettent de contourner les valeurs par défaut.

4.2.3. Contourner le comportement par défaut NLS

Le serveur Oracle supporte un environnement hétérogène dans le cas où les clients et le serveur n'utilisent pas les mêmes jeux de caractères.

Le jeu de caractères à utiliser par un client pour une session peut être configuré via le paramètre **NLS_LANG**.

La valeur de cette variable contourne les valeurs par défaut des paramètres d'initialisation.

Le paramètre NLS_LANG se compose de trois composants : Le langage, le territoire et le jeu de caractères. Sa syntaxe est la suivante :

```
NLS_LANG = <langue>_<Territoire>.<Jeu de caractères>
```

Ce paramètre fixe les conventions pour les messages serveur et l'affichage des jours et des mois.

Langue : Chaque nom de langue dans Oracle se nomenclature avec leur nom anglais, par exemple : American, French ou German. La valeur par défaut est l'anglais américain. Il contourne le paramètre NLS_LANGUAGE.

Territoire : Il spécifie des valeurs pour le format par défaut de date ou numérique et passe outre le paramètre NLS_TERRITORY. Les valeurs sont par exemple America, France ou Canada et la valeur par défaut est l'amérique.

Jeu de caractères : Il spécifie le jeu de caractères pour le client sur son poste de travail. C'est un acronyme de la forme US7ASCII ou WE8ISO8859P1.

Il est possible de modifier la valeur de cette variable d'environnement sous Windows NT via l'onglet des paramètres d'environnement dans les « Propriétés Systèmes ».

D'autres variables d'environnement peuvent être spécifiées pour une station cliente :

NLS_LANG
NLS_CREDIT
NLS_DEBIT
NLS_DISPLAY
NLS_LIST_SEPARATOR
NLS_MONETARY
NLS_NCHAR

4.2.4. Comportement dépendant du langage pour une session

Il est possible de contrôler la valeur par défaut du langage et du territoire en utilisant la commande ALTER SESSION.

Une commande ALTER SESSION implicite est réalisée lors du démarrage d'une session. Ces appels implicites positionnent les variables NLS_LANGUAGE et NLS_TERRITORY avec les valeurs des variables stockées dans NLS_LANG.

De plus, il est possible de changer ces variables en utilisant la commande ALTER SESSION comme suit :

```
ALTER SESSION SET NLS_LANGUAGE = <langue> NLS_TERRITORY =  
<territoire> ;
```

4.2.5. NLS et le tri

Pour réaliser un résultat trié selon la convention de langage supportée par la station de travail, le serveur Oracle propose un mécanisme de tri linguistique.

Le mécanisme de tri conventionnel est le tri binaire. Dans ce cas, le tri est réalisé selon la valeur binaire des caractères qui est défini par le plan d'encodage des caractères propre à la station de travail. Cependant, l'ordre alphabétique des caractères peut varier d'un langage à l'autre. L'ordre des caractères après un tri binaire peut donc parfois ne pas correspondre à l'ordre alphabétique de la langue. Pour parer à ce problème, Oracle dispose d'un mécanisme de tri linguistique.

Ce tri linguistique est lié au paramètre NLS_SORT qui peut passer outre la variable NLS_LANGUAGE.

Par exemple, l'allemand a son propre ordre de tri nommé « GERMAN ».

Le tri linguistique n'est pas supporté pour les caractères encodés sur des longueurs de bit variables.

La commande suivante permet de positionner la variable avec la valeur souhaitée :

```
ALTER SESSION SET nls_sort = <langue>
```

Le paramètre d'initialisation NLS_COMP permet de contrôler le comportement des opérateurs de comparaison face à l'ordonnement des langues.

La valeur par défaut est binaire, mais elle peut être passée à ANSI pour utiliser le tri linguistique.

4.2.6. Paramètres NLS dans les fonctions SQL

Il est parfois nécessaire de réaliser des requêtes SQL indépendantes du langage. Les fonctions SQL suivantes peuvent utiliser avec leurs paramètres NLS correspondants :

Fonctions SQL	Paramètre NLS
TO_DATE	NLS_DATE_LANGUAGE NLS_CALENDAR
TO_NUMBER	NLS_NUMERIC NLS_CURRENCY NLS_ISO_CURRENCY
TO_CHAR	NLS_DATE_LANGUAGE NLS_NUMERIC NLS_CURRENCY NLS_ISO_CURRENCY NLS_CALENDAR
NLS_UPPER, NLS_LOWER, NLS_INITCAP, NLSSORT	NLS_SORT

Exemple :

```
SELECT TO_CHAR(hiredate, 'dd.mm.yyyy', 'NLS_DATE_LANGUAGE=GERMAN')  
FROM emp;
```

```
SELECT ename, TO_CHAR('99G999D99', 'NLS_NUMERIC_CHARACTERS=,');
```

Note : Les paramètres suivants ne sont pas pris en compte dans les fonctions SQL : NLS_LANGUAGE, NLS_TERRITORY, NLS_DATE_FORMAT.

4.3. Information du dictionnaire de données

4.3.1. Obtenir des informations sur les jeux de caractères

Il est possible d'utiliser la vue NLS_DATABASE_PARAMETERS pour consulter les informations sur les jeux de caractères courants de la base de données.

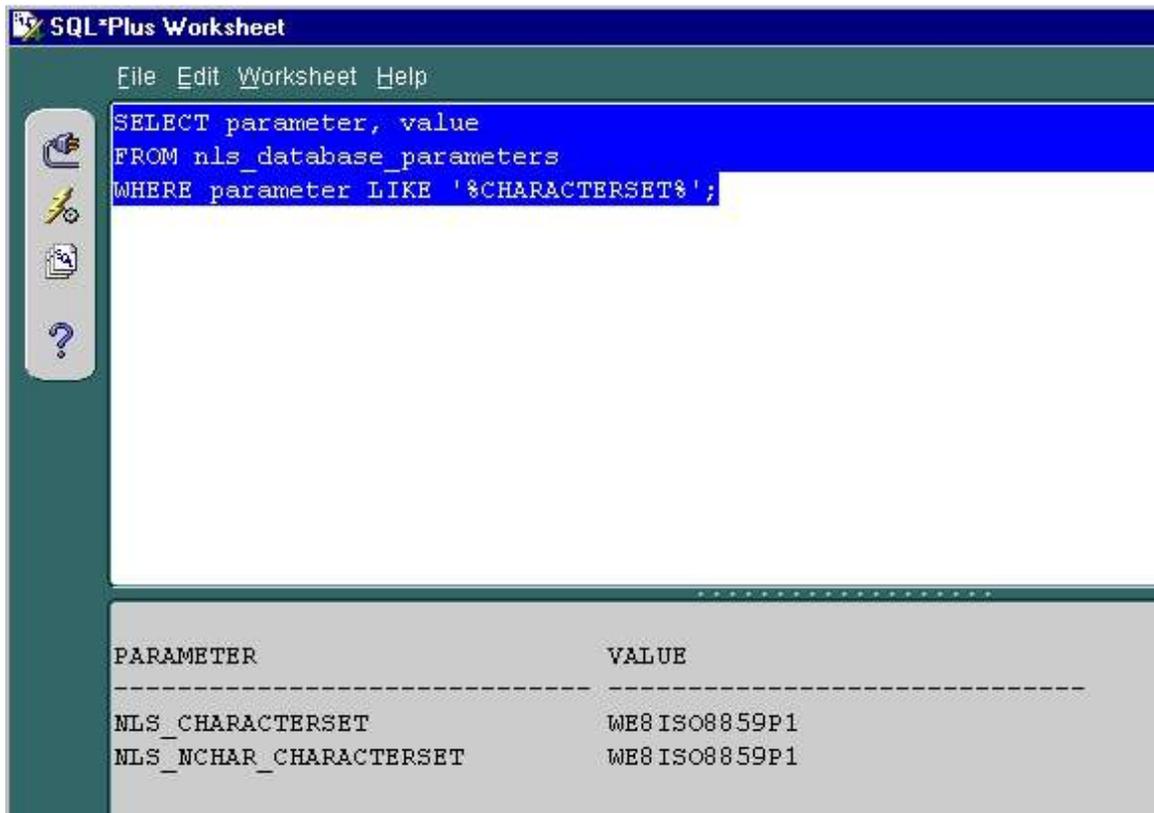


Figure 4-3-1

Les paramètres à considérer sont NLS_CHARACTERSET et NLS_NCHAR_CHARACTERSET.

La requête SQL suivante permettra d'afficher les informations sur ces variables :

```
SELECT parameter, value FROM nls_database_parameters
WHERE parameter LIKE '%CHARACTERSET%';
```

4.3.2. Obtenir des informations sur les configurations NLS

Les vues suivantes permettent de récolter des informations sur la configuration NLS courante :

NLS_INSTANCE_PARAMETERS : C'est une vue du dictionnaire affichant les paramètres NLS explicitement initialisés dans le fichier de paramètres.

NLS_SESSION_PARAMETERS : C'est une vue du dictionnaire affichant les paramètres de session NLS.

V\$NLS_VALID_VALUES : Cette vue permet de récupérer les paramètres NLS valides tel que LANGUAGE, SORT, TERRITORY et CHARACTERSET.

V\$NLS_PARAMETERS : Cette vue permet d'afficher les valeurs courantes des paramètres NLS.