

Administration des Bases de données Oracle



FACULTE DES SCIENCES BEN M'SICK
UNIVERSITÉ HASSAN II DE CASABLANCA

Depuis 2014

—

Licence SMI

—

ELFILALI Sanaa

UNIVERSITE HASSAN II DE CASABLANCA FACULTE DES SCIENCES BEN M'SIK	 <p>U H I I F S B M</p> <p>FACULTE DES SCIENCES BEN M'SIK UNIVERSITE HASSAN II DE CASABLANCA</p>	SMI / S6
--	--	-----------------

DBA

Oracle 12c

Département math-info

PR. SANAA EL FILALI

Smi / s6

Année universitaire : 2022/2023

CHAPITRE 1

Composants de l'architecture Oracle

Introduction :

Ce chapitre présente l'architecture du serveur Oracle via l'étude des structures physiques, des structures mémoire, des processus et des structures logiques qui entrent en jeu dans l'établissement d'une connexion à une base de données, la création d'une session et l'exécution de commandes SQL.

Présentation des principaux composants

L'architecture Oracle comporte plusieurs composants principaux qui seront présentés plus loin dans ce chapitre.

- **Serveur Oracle :** Un serveur Oracle comporte plusieurs fichiers, processus et structures mémoire, mais ces éléments ne sont pas tous utilisés dans le traitement des instructions SQL. Certains de ces éléments améliorent les performances de la base de données, permettent de récupérer la base en cas d'incident logiciel ou matériel ou exécutent d'autres tâches nécessaires à la gestion de la base. Le serveur Oracle est constitué d'une instance Oracle et d'une base Oracle.
- **Instance Oracle :** Une instance Oracle est une combinaison des processus d'arrière-plan et des structures mémoire. Pour accéder aux données de la base, il est nécessaire de démarrer l'instance. A chaque démarrage d'instance, une mémoire SGA (System Global Area) est allouée et des processus d'arrière-plan Oracle sont lancés. Ces processus exécutent des fonctions pour le compte du processus appelant. Ils regroupent des fonctions qui, sinon, seraient gérées par plusieurs programmes Oracle exécutés par chaque utilisateur. Les processus d'arrière-plan effectuent des opérations d'entrée/sortie et surveillent d'autres processus Oracle afin de permettre un plus grand parallélisme et d'améliorer les performances et la fiabilité.
- **Base de données Oracle :** La base Oracle est constituée de fichiers du système d'exploitation, appelés fichiers de base de données. Ces fichiers constituent l'espace de stockage physique des données de la base. Ils permettent de maintenir la cohérence des données et peuvent être récupérés en cas d'échec de l'instance.
- **Autres fichiers importants :** Les fichiers qui ne sont pas des fichiers de base de données permettent de configurer l'instance, d'authentifier les utilisateurs ayant des privilèges et de récupérer la base en cas de défaillance du disque.
- **Processus utilisateur et serveur :** Les processus utilisateur et serveur sont les principaux processus intervenant dans l'exécution d'une instruction SQL. Néanmoins, d'autres processus peuvent être utilisés par le serveur pour le traitement des instructions SQL.
- **Autres processus :** Les options telles que Advanced Queuing, Real Application Clusters, Shared Server, Advanced Replication, etc. utilisent d'autres processus. Ces processus sont présentés dans les cours qui traitent de ces options.

Architecture du Serveur Oracle :

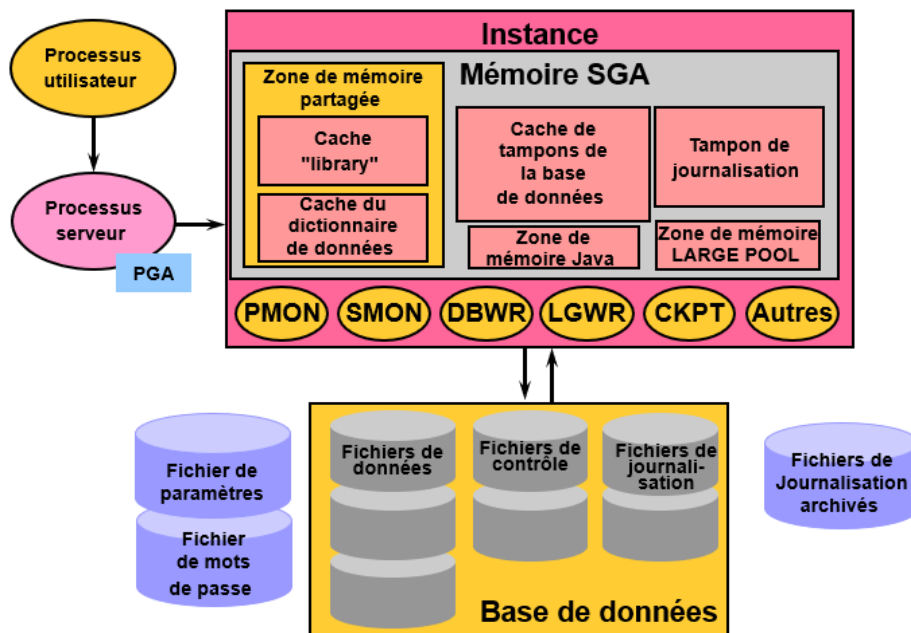


Figure 1 Architecture Oracle

Serveur Oracle

Un serveur Oracle est un système de gestion de base de données qui offre une méthode de gestion des informations ouverte, complète et intégrée, et aussi constitué d'une instance et d'une base de données Oracle.

Le serveur de bases de données est primordial pour la gestion des informations. En général, il doit gérer de façon fiable dans un environnement multiutilisateur une quantité importante de données pour que de nombreux utilisateurs puissent y accéder simultanément, et ce sans affecter les performances. Ce type de serveur doit également empêcher tout accès non autorisé et proposer des solutions efficaces de récupération des données après incident.

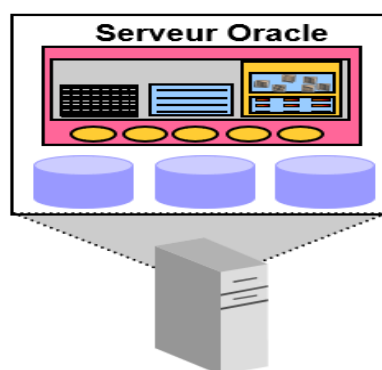


Figure 2 Serveur Oracle

Instance Oracle

Une instance Oracle permet d'accéder à une base de données Oracle, et n'ouvre qu'une seule base de données, et aussi constituée de structures de processus d'arrière-plan et de structures mémoire.

Une instance Oracle est constituée de la mémoire SGA et des processus d'arrière-plan utilisés pour gérer une base de données. Une instance est identifiée à l'aide de méthodes propres à chaque système d'exploitation. Elle ne peut ouvrir et utiliser qu'une seule base à la fois.

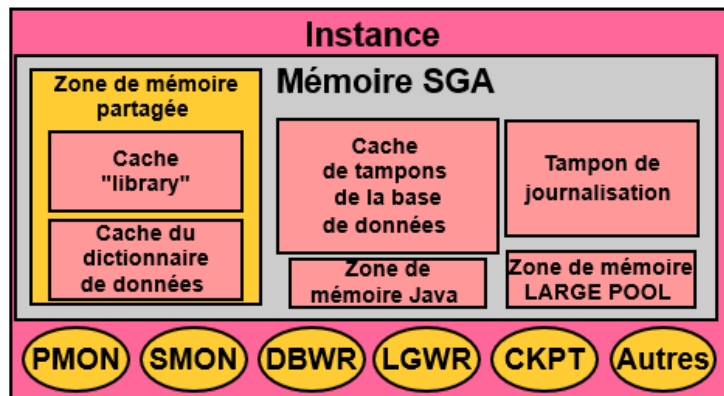


Figure 3 Instance Oracle

Etablir une connexion et créer une session

Pour soumettre des instructions SQL à une base de données Oracle, l'utilisateur doit se connecter à une instance.

- L'utilisateur démarre un outil, par exemple SQL*Plus, ou exécute une application développée à l'aide d'un outil tel qu'Oracle Forms. Cette application ou cet outil est exécuté en tant que processus utilisateur.
- Dans la configuration la plus simple, lorsqu'un utilisateur se connecte au serveur, un processus appelé processus serveur est créé sur l'ordinateur qui exécute le serveur Oracle. Ce processus communique avec l'instance Oracle pour le compte du processus utilisateur exécuté sur le client et il exécute des instructions SQL pour le compte de l'utilisateur.

Connexion

Une connexion est un chemin de communication entre un processus utilisateur et un serveur Oracle. L'utilisateur d'une base de données peut se connecter à un serveur Oracle de trois manières :

- Il peut se connecter au système d'exploitation qui exécute l'instance Oracle et démarrer une application ou un outil qui accède à la base de données sur ce système. La communication est établie à l'aide des mécanismes IPC (interprocess communication) du système d'exploitation hôte.
- Il peut démarrer une application ou un outil sur un ordinateur local et se connecter, via un réseau, à l'ordinateur qui exécute l'instance Oracle. Dans cette configuration, appelée client-serveur, un logiciel réseau permet à l'utilisateur et au serveur Oracle de communiquer.
- Dans une connexion à trois niveaux (three-tier), l'ordinateur de l'utilisateur communique, sur le réseau, avec une application ou un serveur réseau. Ce

dernier est lui-même connecté via un réseau à la machine qui exécute l'instance Oracle. Par exemple, l'utilisateur exécute un navigateur sur un ordinateur en réseau pour utiliser une application résidant sur un serveur NT. Ce serveur extrait les données d'une base Oracle s'exécutant sur un hôte UNIX.

Sessions

Une session est une connexion spécifique d'un utilisateur à un serveur Oracle. La session démarre lorsque l'utilisateur est authentifié par le serveur Oracle et se termine lorsque l'utilisateur se déconnecte ou en cas de déconnexion anormale. Un utilisateur de la base de données peut ouvrir plusieurs sessions en parallèle s'il se connecte simultanément à partir de plusieurs outils, applications ou terminaux. Le serveur Oracle doit être disponible pour que l'on puisse ouvrir une session, sauf dans les cas où certains outils spécialisés d'administration de base de données sont utilisés.

Remarque : Le type de connexion décrit ici, impliquant une correspondance un à un entre l'utilisateur et le processus serveur, est appelé connexion serveur dédié. Dans le cadre d'une configuration serveur partagé, plusieurs processus utilisateur peuvent partager des processus serveur.

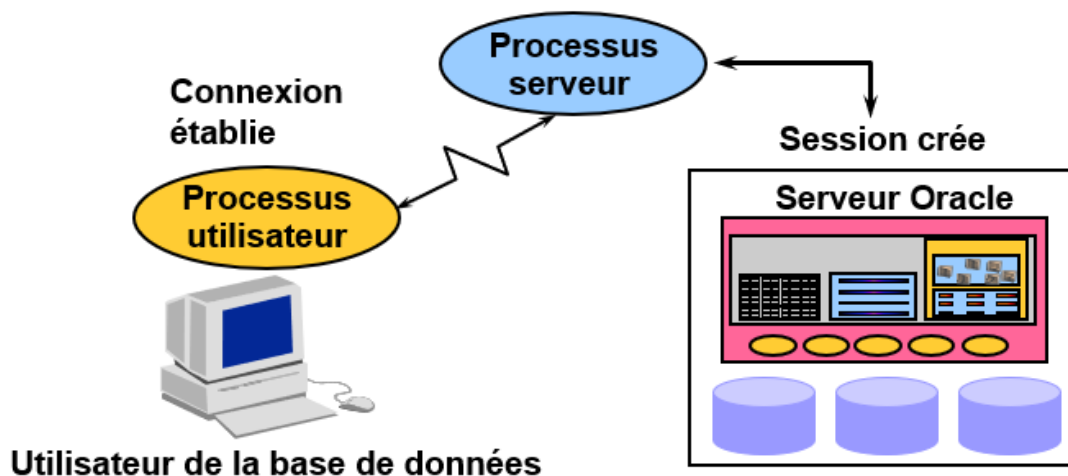


Figure 4 Etablir une connexion avec la base de données

Base de données Oracle

En règle générale, une base de données a pour fonction de stocker des informations associées et de permettre leur extraction. Une base Oracle possède une structure logique et une structure physique. La structure physique correspond à l'ensemble de fichiers du système d'exploitation constituant la base de données. Une base Oracle est composée de trois types de fichier :

- les fichiers de données, qui contiennent les données de la base,
- les fichiers de journalisation (fichiers redo log), qui contiennent un enregistrement des modifications apportées à la base afin de permettre la récupération des données en cas de panne,
- les fichiers de contrôle, qui contiennent les informations nécessaires au maintien et à la vérification de l'intégrité de la base de données.

Autres fichiers importants

Le serveur Oracle utilise également d'autres fichiers qui ne font pas partie de la base de données :

- Le fichier de paramètres définit les caractéristiques d'une instance Oracle. Ce fichier contient, par exemple, des paramètres qui définissent la taille de certaines structures de la mémoire SGA.
- Le fichier de mots de passe authentifie les utilisateurs autorisés à démarrer et à arrêter une instance Oracle.
- Les fichiers de journalisation archivés sont des copies hors ligne (offline) des fichiers de journalisation, qui peuvent être nécessaires à la récupération des données suite à une défaillance physique.

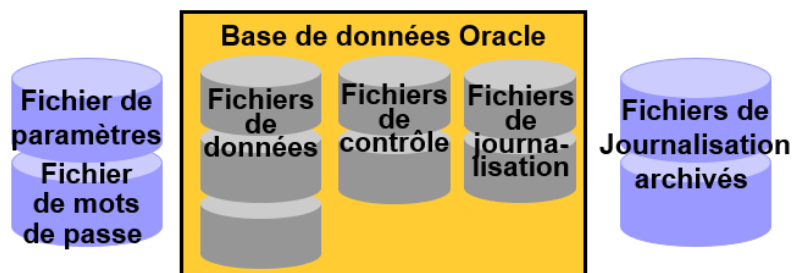


Figure 5 Base de données Oracle

Structure physique

La structure physique d'une base de données Oracle comprend trois types de fichier : fichiers de contrôle, fichiers de données et fichiers de journalisation.

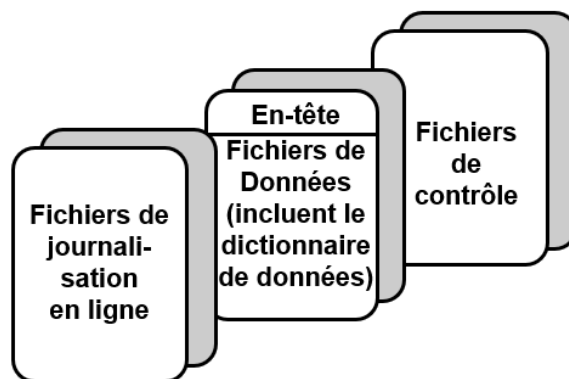


Figure 6 Structure Physique de la base de données

Structure mémoire

La structure mémoire d'Oracle est constituée des deux zones de mémoire ,la mémoire SGA, qui est allouée au démarrage de l'instance et qui est une composante fondamentale d'une instance Oracle et la mémoire PGA, qui est allouée au démarrage du processus serveur .

Mémoire SGA

La mémoire SGA est également appelée Shared Global Area. Elle stocke les informations de la base qui sont partagées par les processus de base de données. Elle contient les données et les informations de contrôle utilisées par le serveur Oracle. Enfin, elle est allouée dans la mémoire virtuelle de l'ordinateur sur lequel est installé le serveur Oracle.

L'instruction suivante permet d'afficher les allocations de la mémoire SGA :

```
SQL> SHOW SGA;
```

Mémoire SGA dynamique

A partir d'Oracle9i, une mémoire SGA dynamique met en oeuvre une infrastructure permettant de modifier sa configuration sans qu'il soit nécessaire d'arrêter l'instance. Il est donc possible de modifier la taille du cache de tampons (buffer cache) de la base de données et de la zone de mémoire partagée sans arrêter l'instance. En théorie, ces zones peuvent être sous-configurées à l'origine. Ensuite, elles augmentent ou diminuent en fonction de leur charge globale respective, jusqu'à un maximum de SGA_MAX_SIZE.

Cache "library"

La taille du cache "library" dépend du dimensionnement défini pour la zone de mémoire partagée. La mémoire est allouée lors de l'analyse d'une instruction ou de l'appel d'un programme. Si la taille de la zone de mémoire partagée est trop restreinte, les instructions sont sans cesse rechargées dans le cache "library" au détriment des performances. Le cache "library" est géré par un algorithme LRU (Least Recently Used). A mesure que le cache se remplit, les chemins d'exécution et les arborescences d'analyse (parse) les plus anciens sont supprimés du cache "library" afin de libérer de l'espace pour les nouvelles entrées. Si les instructions SQL ou PL/SQL ne sont pas réutilisées, elles sont finalement retirées de la mémoire (sur la base de la liste LRU).

Le cache "library" est composé de deux structures :

- Zone SQL partagée : La zone SQL partagée stocke et partage le plan d'exécution et l'arborescence de l'analyse des instructions SQL exécutées dans la base de données. Lorsqu'une instruction SQL identique est exécutée pour la deuxième fois, elle peut utiliser les informations d'analyse disponibles dans la zone SQL partagée pour accélérer son exécution. Afin que les instructions SQL utilisent une zone de mémoire SQL partagée lorsque cela s'avère possible, il est nécessaire que le texte, le schéma et les variables attachées (bind variables) soient en tous points identiques.
- Zone PL/SQL partagée : La zone PL/SQL partagée stocke et partage les dernières instructions PL/SQL exécutées. Les procédures et les programmes analysés et compilés (fonctions, déclencheurs (triggers) et packages) sont stockés dans cette zone.

Cache du dictionnaire de données

Le cache du dictionnaire de données est également appelé cache de lignes (row cache). La mise en mémoire cache des informations du dictionnaire dans le cache de tampons de la base de données et dans la zone de mémoire partagée améliore les performances. Les informations sur la base de données (données du compte utilisateur, noms des fichiers de données, noms des segments, emplacements d'extents, descriptions des tables et privilèges utilisateur) sont stockées dans les tables du dictionnaire de données. Lorsque le serveur a besoin de ces informations, les tables du dictionnaire sont lues, puis les données renvoyées sont placées dans le cache de ce dictionnaire.

Dimensionner le dictionnaire de données

La taille totale du dictionnaire de données dépend de celle de la zone partagée. Elle est gérée en interne par la base de données. Si le cache du dictionnaire est trop restreint, la base de données doit interroger de façon répétée les tables de ce dictionnaire pour obtenir les informations dont le serveur a besoin. Ces interrogations sont des appels récursifs. Elles sont plus lentes que les interrogations effectuées directement dans le cache du dictionnaire de données, parce qu'elles n'utilisent pas le langage SQL.

Cache de tampons de la base de données

Lors du traitement d'une interrogation, le processus serveur Oracle recherche les blocs dont il a besoin dans le cache de tampons de la base. Si un bloc nécessaire ne s'y trouve pas, le processus lit le bloc dans le fichier de données, puis place une copie de ce bloc dans le cache de tampons. Etant donné que ce bloc pourra par la suite être trouvé dans la mémoire, les demandes suivantes ne nécessitent pas de lecture physique. Le serveur Oracle utilise un algorithme LRU pour retirer de la mémoire les tampons (buffers) qui n'ont pas été utilisés récemment, afin de libérer de l'espace pour de nouveaux blocs dans le cache de tampons.

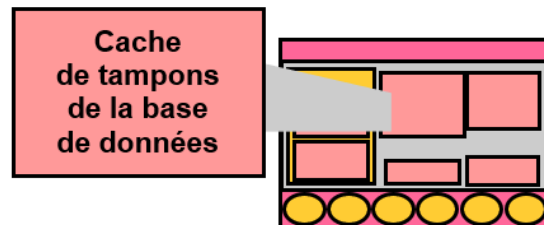


Figure 7: Cache de tampons de la base de données

Tampon de journalisation

Le tampon de journalisation est un tampon circulaire qui contient les modifications apportées aux blocs de fichiers de données. Ces informations sont stockées dans des entrées de journalisation. Ces entrées contiennent les informations nécessaires à la recreation des données avant toute modification via les opérations INSERT, UPDATE, DELETE, CREATE, ALTER ou DROP.

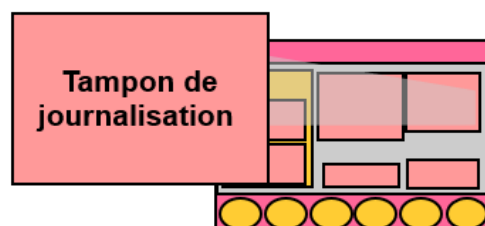


Figure 8 : Tampon de journalisation

Zone de mémoire LARGE POOL

En allouant au serveur partagé, à Oracle XA et aux mémoires tampon "Parallel Query" de la mémoire par session à partir de la zone LARGE POOL, Oracle peut principalement utiliser la zone de mémoire partagée pour la mise en mémoire cache des instructions SQL partagées. La charge des zones de la mémoire partagée est ainsi réduite. Cette dernière n'a pas besoin de céder de l'espace nécessaire à la mise en mémoire cache des arborescences d'analyse du code SQL au profit des informations de session du serveur partagé, des E/S et des processus de sauvegarde et de restauration. Le gain de performances provient de la réduction de la surcharge due à l'augmentation et à la réduction du cache SQL partagé.

Zone de mémoire Java

La configuration de la zone de mémoire Java est facultative, mais nécessaire si vous installez et utilisez Java. La taille de la zone est définie en octets par le paramètre JAVA_POOL_SIZE.

Mémoire PGA

La mémoire PGA (Program Global Area ou Process Global Area) est une région de la mémoire qui contient les données et les informations de contrôle d'un seul processus serveur ou d'un seul processus d'arrière-plan. Elle est allouée lors de la création d'un processus et libérée à la fin de celui-ci. Contrairement à la mémoire SGA, qui est partagée par plusieurs processus, la mémoire PGA n'est utilisée que par un seul processus.

Contenu de la mémoire PGA

Le contenu de la mémoire PGA varie selon que l'instance est exécutée dans une configuration serveur dédié ou serveur partagé. Elle inclut généralement les composants suivants :

- Zone privée de partage des ordres SQL : Contient des données telles que des informations attachées et des structures mémoire d'exécution. Chaque session qui émet une instruction SQL possède une zone privée de partage des ordres SQL. La zone de chaque utilisateur qui soumet la même instruction SQL utilise une seule zone SQL partagée. La zone privée de partage des ordres SQL d'un curseur est divisée en deux sous-zones :
 - La zone persistante, qui contient des informations attachées. Elle n'est libérée que lorsque le curseur est fermé.
 - La zone d'exécution, qui est créée au tout début d'une demande d'exécution. En ce qui concerne les commandes INSERT, UPDATE et DELETE, cette zone n'est libérée que lorsque l'instruction a été exécutée. En ce qui concerne les interrogations, elle n'est libérée que lorsque toutes les lignes sont extraites ou lorsque l'interrogation est annulée.
- Zone privée de partage des ordres SQL : l'emplacement de la cette zone dépend du type de connexion établi pour la session. Dans un environnement serveur dédié, les zones privées de partage des ordres SQL se trouvent dans la mémoire PGA de leur processus serveur. Dans un environnement serveur partagé, elles sont situées dans la mémoire SGA.

Différences d'allocation de mémoire entre le serveur dédié et le serveur partagé :

Le contenu de la mémoire PGA varie selon que l'instance est exécutée dans une configuration serveur dédié ou serveur partagé.

Structure de processus :

Oracle utilise différents types de processus :

- le processus utilisateur, qui est démarré au moment où un utilisateur de la base de données tente de se connecter au serveur Oracle,
- le processus serveur, qui établit la connexion à l'instance Oracle et démarre lorsqu'un utilisateur ouvre une session,
- les processus d'arrière-plan, lancés au démarrage d'une instance Oracle.

Processus utilisateur

Un utilisateur de base de données qui doit demander des informations contenues dans la base doit d'abord établir une connexion avec le serveur Oracle. La demande de connexion s'effectue via un outil d'interface de base de données (SQL*Plus, par exemple) et le lancement du processus utilisateur. Ce dernier n'entre pas directement en interaction avec le serveur Oracle. Il génère des appels via l'interface UPI (User Program Interface) qui crée une session et démarre un processus serveur

Processus serveur

Lorsque l'utilisateur a établi une connexion, un processus serveur démarre pour traiter les demandes du processus utilisateur. Un processus serveur peut être dédié ou partagé. Dans un environnement serveur dédié, le processus serveur traite la demande d'un seul processus utilisateur. Le processus serveur prend fin lorsque le processus utilisateur se déconnecte. Dans un environnement serveur partagé, le processus serveur traite les demandes de plusieurs processus utilisateur. Le processus serveur communique avec le serveur Oracle à l'aide de l'interface OPI (Oracle Program Interface).

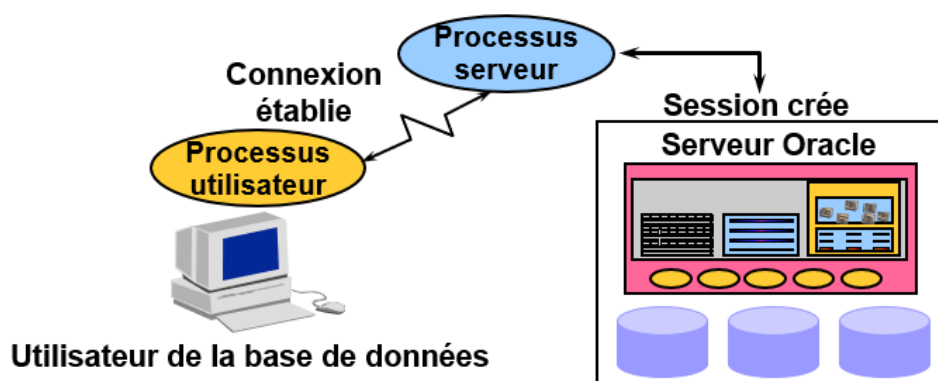


Figure 9 : processus Oracle

Processus d'arrière-plan

L'architecture Oracle possède cinq processus d'arrière-plan obligatoires qui seront présentés plus loin dans ce chapitre. Oracle possède, en outre, un certain nombre de processus d'arrière-plan facultatifs qui sont démarrés via l'exécution de l'option correspondante. Ces processus ne sont pas décrits dans ce chapitre, à l'exception du processus d'arrière-plan ARCn.

Processus database writer (DBWn)

Le processus serveur enregistre les modifications des blocs d'annulation et des blocs de données dans le cache de tampons de la base de données. Le processus DBWn écrit les tampons "dirty" du cache de tampons de la base de données dans les fichiers de données. Il garantit qu'un nombre suffisant de mémoires tampon libres (tampons qui peuvent être écrasés lorsque les processus serveur doivent lire des blocs dans les fichiers de données) est disponible dans le cache de tampons de la base de données. Les performances de la base sont améliorées puisque les processus serveur n'effectuent les modifications que dans le cache de tampons de la base de données.

Le processus DBWn diffère l'écriture dans les fichiers de données jusqu'à ce que l'un des événements suivants se produise :

- Point de reprise (checkpoint) normal ou incrémentiel.
- Le nombre de tampons "dirty" a atteint une valeur seuil.
- Un processus balaie un certain nombre de blocs à la recherche de mémoires tampon libres et n'en trouve pas.
- Le temps imparti est dépassé.
- Une demande de ping est émise dans l'environnement Real Application Clusters (RAC).
- Un tablespace normal ou temporaire est mis hors ligne.
- Un tablespace est mis en lecture seule.
- Une table est supprimée ou vidée.

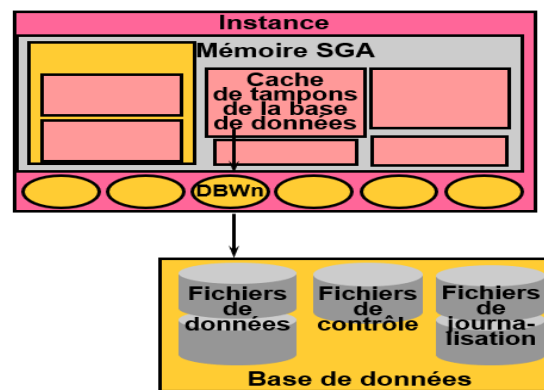


Figure 10 : Processus DBWR

Processus LGWR (Log Writer)

Le processus LGWR effectue des opérations d'écriture séquentielles à partir du tampon de journalisation vers le fichier de journalisation dans les cas suivants :

- Lorsqu'une transaction est validée (commit).
- Lorsqu'un tiers du tampon de journalisation est occupé.
- Lorsque le tampon de journalisation contient plus d'un mégaoctet de modifications enregistrées.
- Avant que le processus DBWn n'écrive les blocs modifiés du cache de tampons de la base de données vers les fichiers de données.
- Toutes les trois secondes.

Etant donné que la journalisation est nécessaire à la récupération des données, le processus LGWR ne confirme la validation qu'une fois les données journalisées enregistrées sur le disque.

Le processus LGWR peut également appeler le processus DBWn pour écrire dans les fichiers de données.

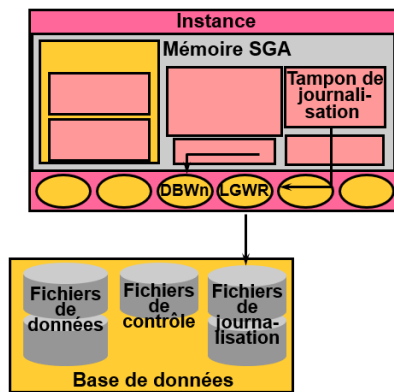


Figure 11 :Processus LGWR

Processus SMON (System Monitor)

En cas d'échec de l'instance Oracle, les informations présentes dans la mémoire SGA qui n'ont pas été écrites sur disque sont perdues. La défaillance du système d'exploitation, par exemple, provoque l'échec de l'instance. Lorsqu'une instance a échoué, le processus d'arrière-plan SMON la récupère automatiquement lors de la réouverture de la base de données. La récupération d'une instance implique les étapes suivantes :

1. Réimplémenter les modifications pour récupérer les données non enregistrées dans les fichiers de données, mais enregistrées dans le fichier de journalisation en ligne (online). Ces données n'ont pas été enregistrées sur disque du fait de l'effacement de la mémoire SGA suite à l'échec de l'instance. Lors de cette opération de réimplémentation, le processus SMON lit les fichiers de journalisation et applique aux blocs de données les modifications qui y sont enregistrées. Etant donné que toutes les transactions validées ont été enregistrées dans les fichiers de journalisation, ce processus permet de récupérer ces transactions dans leur intégralité.
2. Ouvrir la base de données pour permettre aux utilisateurs de s'y connecter. Les données qui ne sont pas verrouillées par des transactions non récupérées sont immédiatement disponibles.
3. Annuler les transactions non validées. Ces transactions sont annulées (rollback) par le processus SMON ou par les processus serveur lorsqu'ils accèdent aux données verrouillées.

Processus PMON (Process Monitor)

Suite à l'échec de processus, le processus d'arrière-plan PMON exécute des opérations de nettoyage :

- Il annule la transaction en cours de l'utilisateur
- Il libère tous les verrous posés sur des tables ou des lignes
- Il libère d'autres ressources réservées par l'utilisateur
- Il redémarre les répartiteurs interrompus

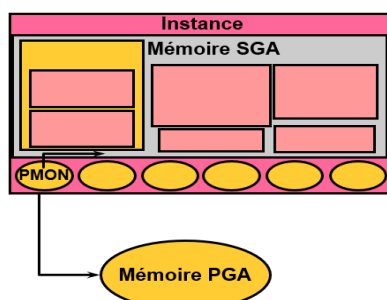


Figure 12 : Processus PMON

Processus CKPT (Checkpoint)

Le processus de point de reprise CKPT enregistre des données dans le fichier de contrôle toutes les trois secondes pour identifier l'emplacement de départ de la récupération dans le fichier de journalisation. Cette opération est appelée point de reprise. Elle vise à garantir que toutes les mémoires tampon du cache de tampons de la base de données qui ont été modifiées avant un point dans le temps ont été écrites dans les fichiers de données. Ce point dans le temps (position du point de reprise) détermine le début de la récupération de la base de données en cas d'échec d'une instance. Dans ce cas, le processus DBWn a déjà écrit tout le contenu des mémoires tampon du cache de la base modifiées avant ce point dans le temps. Dans les versions antérieures à Oracle9i, cette opération s'effectuait en fin de journalisation. En cas de changement de fichier de journalisation, le processus CKPT écrit également ces informations de point de reprise dans les en-têtes des fichiers de données.

Les points de reprise sont mis en oeuvre pour les raisons suivantes :

- Pour garantir que les blocs de données modifiés qui se trouvent en mémoire sont régulièrement écrits sur disque afin d'éviter une perte des données en cas de panne du système ou de la base de données.
- Pour réduire le temps de récupération d'une instance. Seules les entrées de journalisation postérieures au dernier point de reprise doivent être traitées pour que la récupération ait lieu.
- Pour garantir que toutes les données validées ont été écrites dans les fichiers de données lors de l'arrêt.

Processus d'archivage

Tous les autres processus d'arrière-plan sont facultatifs, selon la configuration de la base de données. Toutefois, l'un d'entre eux, ARCn, joue un rôle essentiel dans la récupération d'une base suite à une défaillance du disque. Le serveur Oracle passe d'un fichier de journalisation en ligne au suivant à mesure que ceux-ci se remplissent. Le passage d'un fichier de journalisation à un autre est un changement de fichier de journalisation. Le processus ARCn lance la sauvegarde ou l'archivage du groupe de fichiers de journalisation remplis à chaque changement de fichier de journalisation. Il archive automatiquement le fichier de journalisation en ligne pour qu'il puisse être réutilisé. Ainsi, toutes les modifications apportées à la base sont conservées. Ceci permet au DBA de récupérer la base de données jusqu'au point de panne, même si un disque est endommagé.

Archiver les fichiers de journalisation

L'une des principales décisions de l'administrateur consiste à déterminer s'il doit configurer la base de données pour un fonctionnement en mode ARCHIVELOG ou en mode NOARCHIVELOG.

Mode NOARCHIVELOG : Dans ce mode, les fichiers de journalisation en ligne sont écrasés à chaque changement de fichier de journalisation. Le processus LGWR n'écrase pas un groupe de fichiers de journalisation tant que le point de reprise du groupe n'est pas terminé. Ceci garantit la récupération des données validées en cas d'échec d'une instance. Dans ce cas, seules les données de la mémoire SGA sont perdues (aucune donnée des disques n'est perdue). Une défaillance du système d'exploitation, par exemple, provoque la défaillance de l'instance.

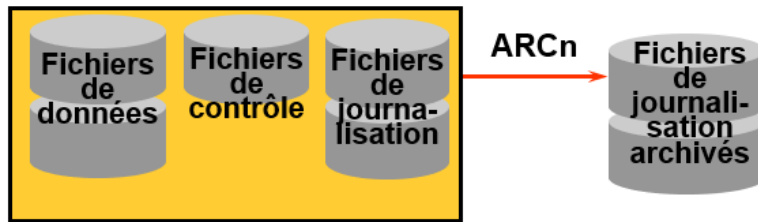


Figure 13 : Archivage

Mode ARCHIVELOG : Si la base de données fonctionne en mode ARCHIVELOG, les groupes inactifs de fichiers de journalisation en ligne remplis doivent être archivés pour pouvoir être réutilisés. Etant donné que les modifications de la base sont enregistrées dans les fichiers de journalisation en ligne, l'administrateur peut utiliser la sauvegarde physique des fichiers de données et les fichiers de journalisation en ligne archivés pour récupérer la base sans perdre les données validées en cas de panne en un point unique, notamment de perte d'un disque. En règle générale, une base de données de production est configurée pour fonctionner en mode ARCHIVELOG.

Structure logique

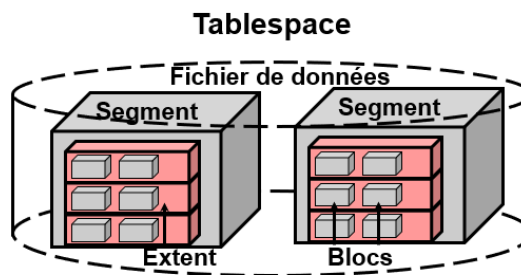


Figure 14 : structure logique

La hiérarchie de la structure logique se présente comme suit :

- Une base de données Oracle contient au moins un tablespace.
- Un tablespace comporte un ou plusieurs segments.
- Un segment est composé d'extents.
- Un extent est composé de blocs logiques.
- Un bloc représente la plus petite unité sur laquelle portent les opérations de lecture et d'écriture.

L'architecture d'une base de données Oracle contient les structures logiques et physiques qui constituent la base de données.

- La structure physique comprend les fichiers de contrôle, les fichiers de journalisation en ligne (online redo log) et les fichiers de données qui constituent la base de données.
- La structure logique est composée de tablespaces, de segments, d'extents (ensemble de blocs contigus) et de blocs de données.

Le serveur Oracle permet un contrôle précis de l'utilisation de l'espace disque à l'aide de tablespaces et de structures de stockage logiques constituées de segments, d'extents et de blocs de données.

Tablespaces

Les données d'une base de données Oracle sont stockées dans des tablespaces.

- Une base de données Oracle peut être divisée en plus petites zones logiques d'espace appelées tablespaces.
- Un tablespace ne peut appartenir qu'à une seule base de données à la fois.
- Chaque tablespace est constitué d'un ou de plusieurs fichiers du système d'exploitation, appelés fichiers de données.
- Un tablespace peut contenir un ou plusieurs segments.
- Les tablespaces peuvent être mis en ligne lorsque la base de données est active.
- A l'exception du tablespace SYSTEM ou des tablespaces contenant un segment d'annulation actif, les tablespaces peuvent être mis hors ligne sans interruption de la base de données.
- Les tablespaces peuvent être accessibles en lecture et écriture ou en lecture seule.

Fichiers de données (structure non logique)

- Chaque tablespace d'une base de données Oracle contient un ou plusieurs fichiers appelés fichiers de données. Ces fichiers sont des structures physiques conformes au système d'exploitation sur lequel s'exécute le serveur Oracle.
- Un fichier de données ne peut appartenir qu'à un seul tablespace.
- Un serveur Oracle crée un fichier de données dans un tablespace en allouant l'espace disque défini et un petit espace supplémentaire.
- L'administrateur de base de données peut modifier la taille des fichiers de données après leur création ou indiquer que leur taille peut augmenter de façon dynamique à mesure que celle des objets du tablespace s'accroît.

Segments

- Un segment est un espace alloué à une structure de stockage logique spécifique d'un tablespace.
- Un tablespace peut contenir un ou plusieurs segments.
- Un segment ne peut pas être réparti sur plusieurs tablespaces, mais peut s'étendre à plusieurs fichiers de données d'un même tablespace.
- Chaque segment est constitué d'un ou de plusieurs extents.

Extents

Les extents permettent d'allouer de l'espace à un segment.

- Un segment peut être constitué d'un ou de plusieurs extents.
 - Lorsque vous créez un segment, celui-ci contient au moins un extent.
 - A mesure que la taille du segment augmente, des extents lui sont ajoutés.
 - L'administrateur de base de données peut ajouter manuellement des extents à un segment.
- Un extent est constitué de blocs Oracle contigus.
- Un extent ne peut pas s'étendre sur plusieurs fichiers de données. Il doit donc appartenir à un seul fichier de ce type.

Blocs de données

Le serveur Oracle gère l'espace de stockage des fichiers de données à l'aide d'unités appelées blocs de données ou blocs Oracle.

- Lorsque le degré de détail maximum est atteint, les données d'une base Oracle sont stockées dans des blocs de données.
- Les blocs de données Oracle représentent la plus petite unité de stockage que le serveur Oracle peut allouer, écrire ou lire.
- Un bloc de données correspond à un ou plusieurs blocs du système d'exploitation alloués à partir d'un fichier de données existant.
- Le paramètre d'initialisation DB_BLOCK_SIZE permet de définir la taille de bloc standard lors de la création d'une base de données Oracle.
- La taille des blocs de données doit correspondre à un multiple de la taille des blocs du système d'exploitation afin d'éviter les opérations d'entrée/sortie inutiles.
- La taille maximale d'un bloc de données dépend du système d'exploitation utilisé.

CHAPITRE 2

Gérer une instance Oracle

Fichiers de paramètres d'initialisation

Pour démarrer une instance, le serveur Oracle doit lire le fichier de paramètres d'initialisation.

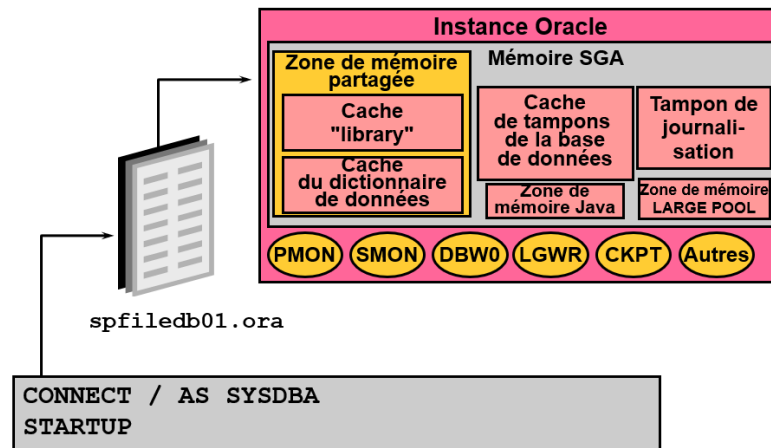


Figure 15 : fichier de paramètres

Fichiers de paramètres d'initialisation

Pour démarrer une instance, le serveur Oracle lit le fichier de paramètres d'initialisation. Ce dernier peut être de deux types :

- Fichier de paramètres statique, PFILE, généralement nommé `initSID.ora`.
- Fichier de paramètres persistant, SPFILE, généralement nommé `spfileSID.ora`.

Contenu des fichiers de paramètres d'initialisation :

- Liste de paramètres d'instance
- Nom de la base de données à laquelle l'instance est associée
- Affectations destinées aux structures mémoire de la mémoire SGA (System Global Area)
- Utilisation des fichiers de journalisation en ligne (online) remplis
- Noms et emplacements des fichiers de contrôle
- Informations relatives aux segments d'annulation

Une instance peut présenter plusieurs fichiers de paramètres d'initialisation afin d'optimiser les performances dans certaines situations.

PFILE

PFILE est un fichier texte que vous pouvez mettre à jour à l'aide d'un éditeur standard du système d'exploitation. Ce fichier est en lecture seule pendant le démarrage de l'instance. S'il est modifié, l'instance doit être interrompue et redémarrée pour que les nouvelles valeurs des paramètres soient effectives.

Par défaut, le fichier PFILE se trouve dans le répertoire `$ORACLE_HOME/dbs` sous le nom `initSID.ora`.

Créer un fichier PFILE

Au cours de l'installation, Oracle Universal Installer crée un exemple de fichier `init.ora`, que vous pouvez utiliser pour créer un fichier `initSID.ora` propre à l'instance. Les paramètres du fichier `initSID.ora` peuvent être modifiés à l'aide d'un éditeur de texte.

Fichier SPFILE

Oracle9i propose un nouveau fichier binaire nommé SPFILE. Ce fichier ne doit pas être modifié manuellement et doit toujours résider côté serveur. Une fois créé, le fichier est mis à jour par le serveur Oracle. S'il est modifié manuellement, il devient inutile. Il permet d'apporter à la base de données des modifications qui seront conservées après l'arrêt et le redémarrage. En outre, il peut régler lui-même les valeurs des paramètres, qui sont enregistrées dans le fichier. Il peut bénéficier de la prise en charge de RMAN pour effectuer des sauvegardes du fichier de paramètres d'initialisation, parce qu'il réside côté serveur. Par défaut, il se trouve dans le répertoire \$ORACLE_HOME/dba et son nom se présente sous le format spfileSID.ora.

Créer un fichier SPFILE

Vous pouvez créer un fichier SPFILE à partir d'un fichier PFILE à l'aide de la commande CREATE SPFILE, exécutable avant ou après le démarrage de l'instance si vous disposez du privilège SYSDBA.

```
SQL> CREATE SPFILE [= 'SPFILE-NAME']
```

```
2 FROM PFILE [= 'PFILE-NAME']
```

Où :

- SPFILE-NAME correspond au nom du fichier SPFILE à créer,
- PFILE-NAME correspond au nom du fichier PFILE utilisé pour créer le SPFILE.
Le fichier PFILE doit être disponible côté serveur.

Si vous ne précisez pas SPFILE-NAME et PFILE-NAME dans la syntaxe, Oracle utilise le fichier PFILE par défaut pour générer un SPFILE dont le nom sera défini par le système.

```
SQL> CREATE SPFILE FROM PFILE;
```

Exporter un fichier SPFILE

Le contenu d'un fichier SPFILE peut être exporté vers un PFILE.

```
SQL> CREATE PFILE FROM SPFILE;
```

Le PFILE est créé côté serveur sous forme de fichier texte. Cette commande peut être exécutée avant ou après le démarrage de l'instance. Vous pouvez ainsi afficher le fichier SPFILE et apporter des modifications en :

- exportant le SPFILE vers un PFILE,
- éditant le fichier PFILE,
- recréant le fichier SPFILE à partir du PFILE édité.

L'export d'un SPFILE vers un PFILE permet également de créer une sauvegarde du fichier de paramètres persistant.

Les commentaires qui figurent sur les mêmes lignes que les paramètres dans le fichier PFILE sont mis à jour dans le fichier SPFILE. Tous les autres commentaires sont ignorés.

Le SPFILE est un fichier binaire, bien que son texte soit lisible sous Unix. Sa modification manuelle le rend inutilisable. Si vous devez consulter le contenu d'un fichier SPFILE ou y apporter des modifications, exportez-le vers un fichier PFILE.

Fonctionnement de la commande STARTUP

Ordre des priorités :

- Lorsque la commande STARTUP est utilisée, le fichier spfileSID.ora côté serveur démarre l'instance.
- Si aucun fichier spfileSID.ora n'est détecté, le fichier SPFILE par défaut côté serveur est utilisé pour démarrer l'instance.
- Si aucun fichier SPFILE par défaut n'est détecté, le fichier initSID.ora côté serveur est utilisé pour démarrer l'instance.

Si vous indiquez un fichier PFILE, le fichier SPFILE par défaut n'est pas utilisé pour démarrer l'instance.

Un fichier PFILE peut éventuellement contenir une définition précisant qu'un SPFILE doit être utilisé. Il n'existe pas d'autre moyen de démarrer l'instance à l'aide d'un SPFILE qui ne se trouve pas à l'emplacement par défaut. Dans ce cas, pour démarrer la base de données, vous devez placer SPFILE=<full path and filename> dans le fichier PFILE.

Démarrer une base de données

Lorsque vous lancez la base de données, vous sélectionnez son statut de démarrage. Les scénarios suivants décrivent les différentes étapes de démarrage d'une instance.

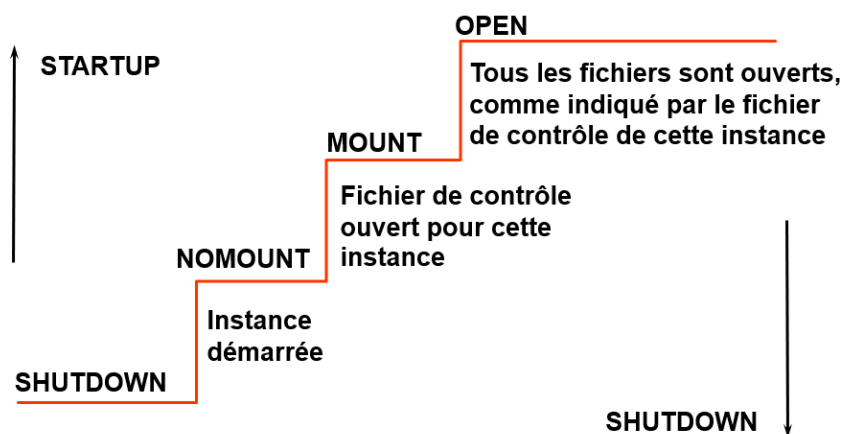


Figure 16 : Démarrer une base de données

Démarrer l'instance (NOMOUNT)

Le démarrage d'une instance en mode NOMOUNT ne s'effectue qu'à la création d'une base de données ou à la recreation de fichiers de contrôle.

Le démarrage d'une instance comprend les tâches suivantes :

- La lecture du fichier d'initialisation dans le répertoire \$ORACLE_HOME/dbs dans l'ordre suivant :
 - fichier spfileSID.ora,
 - s'il n'est pas détecté, fichier spfile.ora,
 - s'il n'est pas détecté, fichier initSID.ora.

La déclaration du paramètre PFILE avec STARTUP remplace les valeurs

par défaut.

- L'affectation de la mémoire SGA.
- Le démarrage des processus d'arrière-plan.
- L'ouverture du fichier alertSID.log et des fichiers trace.

Nommez la base de données à l'aide du paramètre DB_NAME dans le fichier de paramètres d'initialisation ou dans la commande STARTUP.

Monter la base de données (MOUNT) :

Pour effectuer des opérations de maintenance, vous démarrez une instance et montez une base de données sans l'ouvrir.

C'est le cas, par exemple, pour les tâches suivantes :

- renommer des fichiers de données,
- activer ou désactiver des options d'archivage de fichiers de journalisation,
- effectuer une récupération complète de la base de données.

Le montage d'une base de données comprend les tâches suivantes :

- Association d'une base de données à une instance démarrée
- Localisation et ouverture des fichiers de contrôle indiqués dans le fichier de paramètres
- Lecture des fichiers de contrôle pour extraire le nom et le statut des fichiers de données et des fichiers de journalisation. Toutefois, l'existence des fichiers de données et des fichiers de journalisation en ligne n'est pas vérifiée à ce stade.

Ouvrir la base de données (OPEN) :

En mode de fonctionnement normal, vous démarrez une instance avant de monter et d'ouvrir la base de données. Ainsi, les utilisateurs autorisés peuvent se connecter à la base et effectuer des opérations standard sur les données.

L'ouverture de la base de données comprend les tâches suivantes :

- Ouverture des fichiers de données en ligne
- Ouverture des fichiers de journalisation en ligne

Si un fichier de données ou de journalisation en ligne est absent lorsque vous tentez d'ouvrir la base de données, le serveur Oracle renvoie une erreur.

Au cours de cette dernière étape, le serveur Oracle vérifie que tous les fichiers de données et de journalisation en ligne peuvent être ouverts et contrôle la cohérence de la base de données. Si nécessaire, le processus d'arrière-plan SMON (System Monitor) déclenche la récupération de l'instance.

Commande ALTER DATABASE

Pour faire passer la base de données du statut NOMOUNT à MOUNT ou du statut MOUNT à OPEN, exécutez la commande ALTER DATABASE :

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

Vous pouvez ouvrir la base de données en mode lecture seule pour éviter que des transactions utilisateur ne modifient des données.

Pour démarrer une instance, exécutez la commande suivante :

```
ALTER DATABASE OPEN [READ WRITE] READ ONLY]
```

Où :

- READ WRITE : ouvre la base de données en lecture-écriture pour permettre aux utilisateurs de générer des fichiers de journalisation
- READ ONLY : limite l'accès aux transactions en lecture seule et ne permet pas de générer des informations de journalisation

Ouvrir une base de données en mode d'accès restreint

Une session en mode restreint s'avère utile, par exemple, lors de la mise à jour des structures ou de l'import/export d'une base de données. Vous pouvez démarrer la base de données en mode restreint pour qu'elle ne soit accessible qu'aux utilisateurs disposant du privilège RESTRICTED SESSION.

Vous pouvez également mettre la base de données en mode d'accès restreint à l'aide de la commande SQL ALTER SYSTEM :

```
ALTER SYSTEM [ {ENABLE|DISABLE} RESTRICTED SESSION ]
```

Où :

- ENABLE RESTRICTED SESSION accepte uniquement les connexions ultérieures des utilisateurs disposant du privilège RESTRICTED SESSION,
- DISABLE RESTRICTED SESSION désactive le privilège RESTRICTED SESSION pour que les utilisateurs qui ne le possèdent pas puissent se connecter.

Mettre fin à des sessions

Après avoir placé une instance en mode d'accès restreint, procédez comme suit pour fermer toutes les sessions utilisateur actives avant d'effectuer des tâches administratives :

```
ALTER SYSTEM KILL SESSION 'integer1,integer2'
```

Où :

- integer1 correspond à la valeur de la colonne SID de la vue V\$SESSION
- integer2 correspond à la valeur de la colonne SERIAL# de la vue V\$SESSION

Ouvrir une base de données en mode lecture seule

Vous pouvez ouvrir une base de données en lecture seule si elle n'est pas déjà ouverte en mode lecture-écriture. Cette fonctionnalité est particulièrement utile lorsqu'une base de données de secours doit décharger la base de données de production du traitement des interrogations.

Pour qu'une interrogation devant utiliser un tablespace temporaire (par exemple, pour les tris sur disque) aboutisse, l'utilisateur en cours doit disposer d'un tablespace géré localement et affecté en tant que tablespace temporaire par défaut. L'utilisateur SYS doit disposer d'un tablespace géré localement.

Remarque : Les tablespaces gérés localement sont traités dans un autre chapitre.

Le mode lecture seule ne limite ni la fonction de récupération de la base de données, ni les opérations de modification de son statut sans génération de données de journalisation. Ainsi, en mode lecture seule :

- les fichiers de données peuvent être mis hors ligne (offline) et en ligne (online),

- les fichiers de données et les tablespaces hors ligne peuvent être récupérés.

Les opérations d'écriture sur disque portant sur d'autres fichiers, tels que les fichiers de contrôle, les traces d'audit du système d'exploitation, les fichiers trace et les fichiers d'alertes, peuvent se poursuivre en mode lecture seule.

Arrêter la base de données

Arrêtez la base de données pour effectuer la sauvegarde hors ligne de toutes les structures physiques via le système d'exploitation et pour que les modifications appliquées aux paramètres d'initialisation statiques soient effectives au redémarrage.

Pour arrêter une instance, connectez-vous en tant que SYSOPER ou SYSDBA et utilisez la commande suivante :

SHUTDOWN [NORMAL | TRANSACTIONAL | IMMEDIATE | ABORT]

Mode d'arrêt	A	I	T	N
Permet de nouvelles connexions	Non	Non	Non	Non
Attend la fin des sessions en cours	Non	Non	Non	Oui
Attend la fin des transactions en cours	Non	Non	Oui	Oui
Applique un point de reprise et ferme les fichiers	Non	Oui	Oui	Oui

Figure 17 : Mode d'arrêt d'une base de données

Options d'arrêt

Effectuez un arrêt en mode Normal :

Le mode Normal est le mode d'arrêt par défaut. Il s'effectue dans les conditions suivantes :

- Aucune nouvelle connexion ne peut être établie.
- Le serveur Oracle attend la déconnexion préalable de tous les utilisateurs.
- Les tampons de journalisation et de la base de données sont écrits sur disque.
- Les processus d'arrière-plan prennent fin et la zone SGA est supprimée de la mémoire.
- Oracle ferme et démonte la base de données avant d'arrêter l'instance.
- La récupération de l'instance n'est pas nécessaire lors du redémarrage.

Arrêt en mode Transactional

L'arrêt en mode Transactional évite aux clients de perdre leurs travaux en cours. Il s'effectue dans les conditions suivantes :

- Aucun client ne peut lancer de nouvelle transaction pour l'instance indiquée.
- Le client est déconnecté lorsqu'il termine la transaction en cours.
- La fin de toutes les transactions entraîne l'arrêt immédiat de la base de données.
- La récupération de l'instance n'est pas nécessaire lors du redémarrage.

Arrêt en mode Immediate

L'arrêt en mode Immediate s'effectue dans les conditions suivantes :

- Les instructions SQL en cours de traitement par Oracle ne sont pas terminées.
- Le serveur Oracle n'attend pas la déconnexion des utilisateurs de la base de données.
- Oracle annule les transactions actives et déconnecte tous les utilisateurs.
- Oracle ferme et démonte la base de données avant d'arrêter l'instance.
- La récupération de l'instance n'est pas nécessaire lors du redémarrage.

Arrêt en mode Abort

Si les arrêts en modes Normal et Immediate échouent, vous pouvez abandonner l'instance de base de données en cours. Cette opération s'effectue dans les conditions suivantes :

- Les instructions SQL en cours de traitement par le serveur Oracle sont immédiatement interrompues.
- Oracle n'attend pas la déconnexion des utilisateurs de la base de données.
- Les tampons de journalisation et de la base de données ne sont pas écrits sur disque.
- Les transactions non validées ne sont pas annulées.
- L'instance est interrompue sans fermeture des fichiers.
- La base de données n'est pas fermée, ni démontée.
- Une récupération est nécessaire au redémarrage ; elle s'effectue automatiquement.

Remarque : Il n'est pas conseillé de sauvegarder une base de données incohérente.

Fichier d'alertes

Chaque instance Oracle possède un fichier d'alertes. Au besoin, il est créé au démarrage de l'instance. Vous êtes responsable de la gestion de ce fichier dont la taille augmente lors de l'utilisation de la base de données. Consultez le fichier d'alertes dès que vous devez diagnostiquer des opérations quotidiennes ou des erreurs. Ce fichier contient également des pointeurs sur des fichiers trace qui délivrent des informations plus détaillées.

Le fichier d'alertes conserve un enregistrement des informations suivantes :

- Date/heure d'arrêt ou de démarrage de la base de données
- Liste de tous les paramètres d'initialisation qui ne sont pas des paramètres par défaut
- Démarrage des processus d'arrière-plan
- Thread utilisé par l'instance
- Numéro de séquence du journal utilisé par le processus LGWR (Log Writer)
- Informations relatives à un changement de fichier de journalisation
- Création de tablespaces et de segments d'annulation

L'emplacement du fichier alert_SID.log est défini par le paramètre d'initialisation BACKGROUND_DUMP_DEST.

Fichiers trace de processus en arrière-plan

Les fichiers trace de processus en arrière-plan permettent de consigner des erreurs détectées par un processus d'arrière-plan, tel que SMON, PMON ou DBWn. Ils ne sont créés que lorsqu'une erreur exige une écriture dans les fichiers trace. Utilisez-les pour diagnostiquer et résoudre des problèmes. A sa création, un fichier trace de processus en arrière-plan contient des informations d'en-tête précisant le numéro de version du serveur de données et du système d'exploitation. Leur emplacement est défini par le paramètre d'initialisation BACKGROUND_DUMP_DEST.

CHAPITRE 3

Utiliser Les Vues Du Dictionnaire De Données et Les Vues Dynamiques Des Performances

Objets de base de données intégrés

Outre la création des fichiers de base de données, plusieurs autres structures sont créées.

- Le dictionnaire de données décrit les objets de la base de données.
- Les tables dynamiques des performances contiennent des informations permettant à l'administrateur de base de données (DBA) de surveiller et de régler la base de données et l'instance correspondante.
- Les packages PL/SQL sont des unités de programme ajoutant une fonctionnalité à la base de données. Ces packages sont créés lors de l'exécution du script catproc.sql après la commande CREATE DATABASE. Les packages PL/SQL ne seront pas présentés dans ce cours.

Les déclencheurs d'événements de base de données sont des procédures qui s'exécutent implicitement soit lors de la modification d'une table ou d'une vue, soit à la suite d'une action de l'utilisateur ou du système de base de données

Dictionnaire de données

L'un des composants essentiels d'une base de données Oracle est son dictionnaire de données, composé d'un jeu de tables et de vues accessibles en lecture seule, fournissant des informations sur la base de données associée.

Le dictionnaire de données est mis à jour par le serveur Oracle chaque fois qu'une commande LDD est exécutée. En outre, certaines commandes LMD, telles que celle qui provoque l'extension d'une table, peuvent mettre à jour le dictionnaire de données.

Le dictionnaire de données est vital pour la base de données Oracle, et constitue aussi une source importante d'informations pour l'ensemble des utilisateurs, de l'utilisateur final au développeur d'applications, en passant par l'administrateur de base de données.

Vous pouvez y accéder à l'aide d'instructions SQL. Le dictionnaire de données étant accessible en lecture seule, vous ne pouvez exécuter que des interrogations sur les tables et les vues qu'il contient.

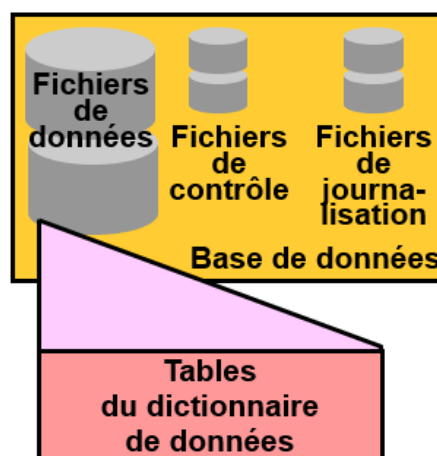


Figure 18 : Dictionnaire de base de données

Tables de base et vues du dictionnaire de données

Le dictionnaire de données contient la description des objets se trouvant dans la base de données. Il comprend deux types d'objet :

Tables de base

Les tables de base sont des tables sous-jacentes qui contiennent des informations sur la base de données. Elles constituent les premiers objets créés dans toute base de données Oracle. Elles sont automatiquement générées lorsque le serveur Oracle exécute le script sql.bsq à la création de la base. Seul le serveur Oracle doit écrire dans ces tables. Les utilisateurs y accèdent rarement directement, car la plupart des données sont stockées dans un format difficilement compréhensible. N'utilisez jamais de commandes LMD pour mettre à jour directement les tables de base, excepté pour la table AUD\$. La table IND\$, par exemple, est une table de base. Elle contient des informations sur les index de la base de données.

Vues du dictionnaire de données

Les vues du dictionnaire de données synthétisent les informations des tables de base et les affichent dans un format qui facilite leur exploitation. Par exemple, les vues du dictionnaire de données utilisent des noms d'objets plutôt que des numéros d'objet. Ces vues sont créées à l'aide du script catalog.sql qui est exécuté après la commande CREATE DATABASE.

Créer des vues du dictionnaire de données

Les tables de base du dictionnaire de données sont automatiquement générées lors de la création d'une base de données. Lorsque la base de données est créée à l'aide d'Oracle Universal Installer, les scripts utilisés pour créer les vues du dictionnaire de données et les vues dynamiques des performances, ainsi que les scripts correspondant aux options du serveur Oracle choisies s'exécutent automatiquement.

Vous devrez exécuter manuellement ces scripts lorsque vous créerez manuellement une nouvelle base de données. En outre, vous devrez peut-être les réexécuter lors d'une mise à niveau vers une nouvelle version du serveur Oracle. Exécutez ces scripts en tant qu'utilisateur SYS avec le privilège SYSDBA.

Contenu du dictionnaire de données

Un dictionnaire de données contient les éléments suivants :

- la définition de tous les objets de schéma de la base de données (tables, vues, index, clusters, synonymes, séquences, procédures, fonctions, packages, déclencheurs (triggers), etc.),
- la quantité d'espace allouée aux objets de schéma et celle qu'ils utilisent,
- les valeurs par défaut des colonnes,
- les informations sur les contraintes d'intégrité,
- le nom des utilisateurs Oracle,
- les privilèges et rôles attribués à chaque utilisateur

les informations d'audit (par exemple, sur l'identité des utilisateurs qui ont consulté ou mis à jour des objets de schéma).

Utilisation du dictionnaire de données

Comment Oracle Server utilise-t-il le dictionnaire de données ?

Les données des tables de base contenues dans le dictionnaire de données sont nécessaires au fonctionnement du serveur Oracle. Par conséquent, seul le serveur doit écrire ou modifier des informations dans ce dictionnaire. Lorsque la base de données est en service, le serveur Oracle lit le dictionnaire de données pour s'assurer que les objets de schéma existent et que les utilisateurs disposent des privilèges d'accès appropriés sur ces derniers. Il met également le dictionnaire à jour en permanence pour y répercuter les modifications apportées aux structures de la base.

Comment les utilisateurs et les administrateurs peuvent-ils exploiter le dictionnaire de données ?

Les vues du dictionnaire de données servent de référence à tous les utilisateurs de la base. Certaines sont accessibles à tous les utilisateurs Oracle, tandis que d'autres sont réservées aux administrateurs.

Catégories de vues du dictionnaire de données

Vues portant le préfixe DBA

Les vues portant le préfixe DBA fournissent une présentation globale de toute la base de données. Elles sont conçues pour être interrogées uniquement par les administrateurs de base de données. Tout utilisateur doté du privilège système SELECT ANY TABLE peut interroger les vues DBA du dictionnaire de données.

Pour exécuter une interrogation sur tous les objets de la base de données, l'administrateur peut lancer l'instruction suivante :

```
SELECT owner, object_name, object_type  
FROM dba_objects;
```

Vues portant le préfixe ALL

Les vues portant le préfixe ALL offrent une perspective générale de la base de données, du point de vue utilisateur. Elles renvoient des informations relatives aux objets de schéma accessibles à tout utilisateur doté de privilèges et de rôles accordés publiquement ou explicitement, ainsi que des informations sur les objets de schéma qui lui appartiennent.

Par exemple, l'interrogation suivante renvoie des informations sur tous les objets auxquels l'utilisateur a accès :

```
SELECT owner, object_name, object_type  
FROM all_objects;
```

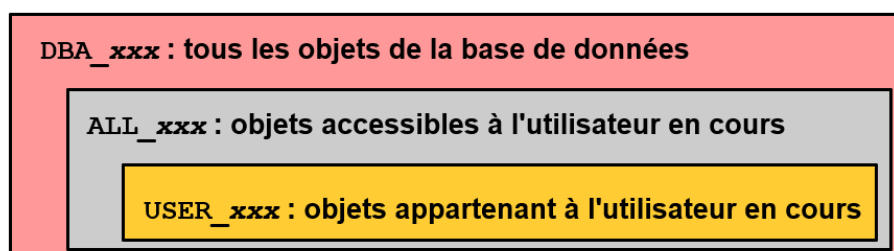


Figure 19 : Catégories de vues du dictionnaire de données

Vues portant le préfixe USER

Les vues qui portent le préfixe USER sont celles qui présentent le plus d'intérêt pour les utilisateurs de la base de données.

Ces vues :

- font référence à l'environnement privé de l'utilisateur dans la base de données,
- concernent généralement des objets appartenant à l'utilisateur en cours,
- comportent des colonnes identiques à celles des autres vues, à ceci près que la colonne OWNER désigne implicitement l'utilisateur en cours,
- renvoient un sous-ensemble des informations contenues dans les vues de catégorie ALL,
- peuvent être associées à des synonymes publics abrégés, pour plus de commodité.

Par exemple, l'interrogation suivante renvoie tous les objets contenus dans le schéma utilisateur :

```
SELECT owner, object_name, object_type  
FROM users_objects;
```

Vues du dictionnaire de données

Les vues du dictionnaire de données sont des vues statiques qui répondent aux questions du type :

- L'objet a-t-il été créé ?
- D'où provient-il ?
- Qui est son propriétaire ?
- De quels privilèges les utilisateurs disposent-ils ?
- Quelles sont les restrictions relatives à l'objet ?

Tables dynamiques des performances

Le serveur Oracle enregistre l'activité en cours de la base de données dans un ensemble de tables virtuelles nommées vues dynamiques des performances. Ces tables ne sont présentes en mémoire que lorsque la base est active, et ce pour présenter les conditions en temps réel de son fonctionnement. Elles pointent sur des sources réelles d'informations figurant dans la mémoire et le fichier de contrôle.

Il ne s'agit pas réellement de tables et elles ne sont pas accessibles à tous les utilisateurs. Toutefois, l'administrateur de base de données peut lancer des interrogations, accorder le privilège SELECT et créer des vues à partir de celles-ci. Les vues dynamiques des performances sont parfois appelées vues fixes, car elles ne peuvent pas être modifiées ni supprimées par l'administrateur .

Les tables dynamiques des performances appartiennent à l'utilisateur SYS et portent des noms qui commencent par le préfixe V_\$. Des vues sont créées à partir de ces tables et sont associées à des synonymes publics. Les synonymes commencent par le préfixe V\$. La vue V\$DATAFILE contient, par exemple, des informations sur les fichiers de données de la base et la vue V\$FIXED_TABLE contient des informations sur toutes les tables et vues dynamiques des performances figurant dans la base de données.

Les tables dynamiques des performances permettent de répondre à des questions du type :

- L'objet est-il en ligne et disponible ?
- L'objet est-il ouvert ?
- Quels verrous externes (locks) sont appliqués ?
- La session est-elle active ?

Exemples de vues dynamiques des performances :

Exemples :

- V\$CONTROLFILE répertorie les noms des fichiers de contrôle.
- V\$DATABASE contient les informations du fichier de contrôle sur la base de données.
- V\$DATAFILE contient les informations du fichier de contrôle sur le fichier de données.
- V\$INSTANCE affiche l'état de l'instance en cours.
- V\$PARAMETER répertorie les paramètres et les valeurs actuellement utilisés par la session.
- V\$SESSION répertorie les informations sur la session en cours.
- V\$SGA contient des informations de synthèse sur la mémoire SGA (System Global Area).
- V\$SPPARAMETER affiche le contenu du fichier SPFILE.
- V\$TABLESPACE affiche les informations du fichier de contrôle sur le tablespace.
- V\$THREAD contient des informations du fichier de contrôle sur les threads.
- V\$VERSION indique les numéros de version des composants de la bibliothèque principale (core library) du serveur Oracle.

CHAPITRE 4

Gérer les Fichiers De Contrôles

Fichier de contrôle

Le fichier de contrôle est un petit fichier binaire nécessaire au démarrage et au bon fonctionnement de la base de données. Chaque fichier de contrôle est associé à une seule base de données Oracle. Avant l'ouverture d'une base, le fichier de contrôle est lu afin de déterminer si cette base est opérationnelle.

Ce fichier est constamment mis à jour par le serveur Oracle pendant l'utilisation de la base de données. Il doit donc être accessible en écriture à chaque ouverture de la base. Seul le serveur Oracle peut modifier le contenu du fichier de contrôle ; l'administrateur de base de données et l'utilisateur final ne peuvent pas modifier ce fichier.

Si ce fichier est inaccessible, la base ne fonctionne pas correctement. Si toutes les copies des fichiers de contrôle d'une base de données sont perdues, cette dernière doit être récupérée pour être ouverte.

Dimensionner le fichier de contrôle

Les mots-clés définis à la création de la base de données ont un impact sur la taille du fichier de contrôle. Ceci est particulièrement visible lorsque les paramètres ont des valeurs élevées. La taille du fichier dépend des mots-clés suivants dans la commande CREATE DATABASE ou CREATE CONTROLFILE :

- MAXLOGFILES
- MAXLOGMEMBERS
- MAXLOGHISTORY
- MAXDATAFILES
- MAXINSTANCES

Contenu du fichier de contrôle

Le fichier de contrôle contient les informations suivantes :

- Le nom de la base de données correspond au nom indiqué par le paramètre d'initialisation DB_NAME ou à celui utilisé dans l'instruction CREATE DATABASE.
- L'identificateur de la base de données est enregistré à la création de cette dernière.
- L'horodatage de création de la base de données est également enregistré à la création de cette dernière.
- Le nom et l'emplacement des fichiers de données et des fichiers de journalisation en ligne (online redo logs) associés sont mis à jour lors de l'ajout, du changement de nom ou de la suppression de l'un de ces fichiers dans la base de données.
- Les informations relatives aux tablespaces sont mises à jour lors de l'ajout ou de la suppression de tablespaces.
- L'historique de journalisation est enregistré lors des changements de fichier de journalisation.
- L'emplacement et le statut des journaux archivés sont enregistrés lors de l'archivage.
- L'emplacement et le statut des sauvegardes sont enregistrés par l'utilitaire Recovery Manager.
- Le numéro de séquence du journal en cours est enregistré lors des changements de fichier de journalisation.

Les informations relatives aux points de reprise (checkpoints) sont enregistrées lorsque les points de reprise ont lieu.

Le fichier de contrôle comprend deux types de section :

- les sections réutilisables,

- les sections non réutilisables.

Les sections réutilisables contiennent les informations de Recovery Manager, telles que les noms des fichiers de données de sauvegarde et ceux des fichiers de journalisation de sauvegarde. Ces sections sont utilisées de manière circulaire et ne peuvent être réutilisées que par Recovery Manager.

Multiplexer le fichier de contrôle

Pour vous préserver d'une panne en un point unique affectant le fichier de contrôle, il est vivement recommandé de multiplexer ce fichier et de stocker chaque copie sur un disque physique distinct. Ainsi, si vous perdez un fichier de contrôle, vous pouvez utiliser une copie multiplexée de ce fichier pour redémarrer l'instance sans procéder à une récupération de la base de données.

Vous pouvez multiplexer un fichier de contrôle jusqu'à huit fois :

- en créant plusieurs fichiers de contrôle en même temps que la base de données et en indiquant leur nom et chemin d'accès complet dans le fichier de paramètres d'initialisation :
CONTROL_FILES=\$HOME/ORADATA/u01/ctrl01.ctl, \$HOME/ORADATA/u02/ctrl02.ctl
- en ajoutant un fichier de contrôle une fois la base de données créée.

Sauvegarder les fichiers de contrôle

Etant donné que le fichier de contrôle enregistre la structure physique de la base de données, vous devez immédiatement effectuer une sauvegarde de ce fichier lorsque vous modifiez la structure physique de la base de données. Pour plus

Multiplexer le fichier de contrôle lorsqu'un fichier SPFILE est utilisé

1. **Modifiez le SPFILE** : à l'aide de la commande ALTER SYSTEM SET, modifiez le fichier SPFILE afin d'y inclure la liste des fichiers de contrôle à utiliser (fichier de contrôle principal et copies multiplexées).
2. **Arrêtez la base de données** : arrêtez la base afin de créer les fichiers de contrôle supplémentaires sur le système d'exploitation.
3. **Créez les fichiers de contrôle supplémentaires** : à l'aide de la commande de copie du système d'exploitation, créez autant de fichiers de contrôle supplémentaires que nécessaire et vérifiez que ces fichiers sont créés dans les répertoires appropriés.
4. **Démarrez la base de données** : au démarrage de la base, le fichier SPFILE est lu et le serveur Oracle met à jour tous les fichiers de contrôle répertoriés dans le paramètre CONTROL_FILES.

Multiplexer le fichier de contrôle lorsqu'un fichier PFILE est utilisé

1. **Arrêtez la base de données** : arrêtez la base afin de créer les fichiers de contrôle supplémentaires sur le système d'exploitation.
2. **Créez les fichiers de contrôle supplémentaires** : à l'aide de la commande de copie du système d'exploitation, créez autant de fichiers de contrôle supplémentaires que nécessaire et vérifiez que ces fichiers sont créés dans les répertoires appropriés.
3. **Ajoutez les noms des fichiers de contrôle au fichier PFILE** : modifiez le fichier PFILE afin d'y inclure la liste des fichiers de contrôle.
4. **Démarrez la base de données** : au démarrage de la base, le fichier PFILE est lu et le serveur Oracle met à jour tous les fichiers de contrôle répertoriés dans le paramètre CONTROL_FILES.

CHAPITRE 5

Gérer les fichiers de Journalisation

Utiliser des fichiers de journalisation

Les fichiers de journalisation (fichiers redo log) permettent de journaliser les transactions en cas de panne de la base de données. Chaque transaction est écrite de manière synchrone dans le tampon de journalisation (redo log buffer), puis transférée dans les fichiers de journalisation pour fournir un mécanisme de récupération en cas de défaillance physique (à l'exception, par exemple, du chargement de données par chemin direct dans des objets lorsque la clause NOLOGGING est activée). Il peut s'agir de transactions qui n'ont pas encore été validées, d'informations sur des segments d'annulation et d'instructions de gestion des schémas et des objets. Les fichiers de journalisation sont utilisés, par exemple, en cas d'échec d'instance pour récupérer les données validées qui n'ont pas été écrites dans les fichiers de données. Ils ne servent qu'à la récupération de données.

Structure des fichiers de journalisation

L'administrateur de base de données peut configurer la base Oracle pour gérer des copies de fichiers de journalisation en ligne afin d'éviter de perdre des données en cas d'incident.

Groupes de fichiers de journalisation en ligne :

- Un ensemble de copies identiques de fichiers de journalisation en ligne est nommé groupe de fichiers de journalisation en ligne.
- Le processus d'arrière-plan LGWR écrit simultanément les mêmes informations dans tous les fichiers de journalisation en ligne d'un groupe.
- Le serveur Oracle nécessite au moins deux groupes de fichiers de journalisation en ligne pour garantir un fonctionnement correct de la base de données.

Membres des fichiers de journalisation en ligne

- Chaque fichier de journalisation en ligne d'un groupe est nommé membre.
- Les membres d'un groupe portent tous le même numéro de séquence de journal et ont tous la même taille. Ce type de numéro, permettant d'identifier de manière unique chaque fichier de journalisation, est attribué lorsque le serveur Oracle écrit dans un groupe de fichiers de journalisation. Le numéro en cours est stocké dans le fichier de contrôle et dans l'en-tête de tous les fichiers de données.

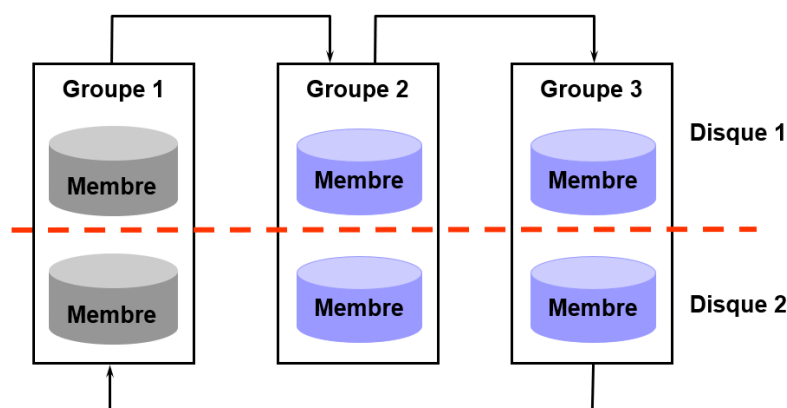


Figure 20 : structure des fichiers journaux

Créer des fichiers de journalisation initiaux

Les groupes de fichiers de journalisation en ligne (online) et les membres initiaux sont créés en même temps que la base de données.

Les paramètres suivants limitent le nombre de fichiers de journalisation en ligne :

- Le paramètre MAXLOGFILES de la commande CREATE DATABASE définit le nombre maximum absolu de groupes de fichiers de journalisation en ligne.
- La valeur maximale et la valeur par défaut du paramètre MAXLOGFILES dépendent du système d'exploitation.
- Le paramètre MAXLOGMEMBERS de la commande CREATE DATABASE détermine le nombre maximum de membres par groupe. La valeur maximale et la valeur par défaut de ce paramètre dépendent du système d'exploitation.

Mode de fonctionnement des fichiers de journalisation

Le serveur Oracle enregistre de manière séquentielle toutes les modifications apportées à la base de données dans le tampon de journalisation. Les entrées de journalisation sont écrites par le processus LGWR dans l'un des groupes de fichiers de journalisation, appelé groupe de fichiers de journalisation en cours, dans les cas suivants :

- lorsqu'une transaction est validée,
- lorsqu'un tiers du tampon de journalisation est occupé,
- lorsque le tampon de journalisation contient plus d'un mégaoctet d'enregistrements modifiés,
- avant que le processus DBWn n'écrive les blocs modifiés du cache de tampons (buffer cache) de la base de données dans les fichiers de données.

Les fichiers de journalisation sont utilisés de façon cyclique. Chaque groupe est identifié par un numéro de séquence remplacé à chaque nouvelle utilisation du journal.

Changements de fichier de journalisation

Le processus LGWR écrit de façon séquentielle dans les fichiers de journalisation en ligne. Lorsque le groupe de fichiers de journalisation en ligne en cours est complet, le processus LGWR passe au groupe suivant. On parle alors de changement de fichier de journalisation.

Lorsque le dernier fichier de journalisation en ligne disponible est complet, le processus LGWR revient au premier groupe et reprend l'écriture.

Ajouter des groupes de fichiers de journalisation en ligne

Dans certains cas, vous pouvez être amené à créer des groupes de fichiers de journalisation supplémentaires. Par exemple, vous pouvez ajouter des groupes pour résoudre les problèmes de disponibilité. Utilisez la commande SQL suivante pour créer un groupe de fichiers de journalisation en ligne :

```
ALTER DATABASE [database]
    ADD LOGFILE [GROUP integer] filespec
    [, [GROUP integer] filespec]...
```

Définissez le nom et l'emplacement des membres en suivant les spécifications de fichier. Vous pouvez choisir la valeur du paramètre GROUP pour chaque groupe de fichiers de journalisation. Si vous omettez ce paramètre, le serveur Oracle génère automatiquement la valeur.

Ajouter des groupes de fichiers de journalisation en ligne

Pour ajouter des membres à des groupes de fichiers de journalisation existants, utilisez la commande ALTER DATABASE LOGFILE MEMBER suivante :

```
ALTER DATABASE [database]
  ADD LOGFILE MEMBER
  [ 'filename' [REUSE]
  [, 'filename' [REUSE]]...
  TO {GROUP integer
      [('filename'[, 'filename']...)]
  }
  ]...
```

Utilisez le nom complet des membres pour ne pas créer les fichiers dans un répertoire par défaut du serveur de bases de données.

Si le fichier existe déjà, il doit être de même taille et vous devez indiquer l'option REUSE. Vous pouvez identifier le groupe cible en indiquant un ou plusieurs membres ou le numéro du groupe.

Supprimer des groupes de fichiers de journalisation en ligne

Pour augmenter ou réduire la taille des groupes de fichiers de journalisation en ligne, ajoutez-en de nouveaux (ayant la nouvelle taille), puis supprimez les anciens.

Utilisez la commande ALTER DATABASE DROP LOGFILE suivante pour supprimer l'ensemble d'un groupe de fichiers de journalisation en ligne :

```
ALTER DATABASE [database]
  DROP LOGFILE {GROUP integer|('filename'[, 'filename']...)}
               [, {GROUP integer|('filename'[, 'filename']...)}]...
```

Restrictions :

- Une instance nécessite au moins deux groupes de fichiers de journalisation en ligne.
- Vous ne pouvez pas supprimer un groupe actif ou en cours.
- Lorsque vous supprimez un groupe de fichiers de journalisation en ligne, les fichiers du système d'exploitation sont conservés.

Supprimer des membres de fichiers de journalisation

Vous pouvez être amené à supprimer un membre d'un fichier de journalisation en ligne parce qu'il n'est plus valide. Utilisez la commande ALTER DATABASE DROP LOGFILE MEMBER suivante pour supprimer un ou plusieurs membres d'un fichier de journalisation en ligne :

```
ALTER DATABASE [database]
  DROP LOGFILE MEMBER 'filename'[, 'filename']...
```

Restrictions :

- Vous ne pouvez pas supprimer le dernier membre valide du groupe.
- S'il s'agit du groupe en cours, vous devez imposer un changement de fichier de journalisation pour supprimer le membre.
- Si la base de données fonctionne en mode ARCHIVELOG et que le groupe de fichiers de journalisation du membre n'est pas archivé, vous ne pouvez pas supprimer le membre.

Transférer ou renommer des fichiers de journalisation en ligne

Vous pouvez changer l'emplacement des fichiers de journalisation en ligne en renommant les fichiers. Avant de renommer les fichiers, assurez-vous que les nouveaux fichiers existent. Le serveur Oracle modifie uniquement les pointeurs dans les fichiers de contrôle, il ne renomme pas physiquement les fichiers de système d'exploitation et ne crée pas de fichiers.

La commande ALTER DATABASE RENAME FILE suivante permet de renommer un fichier de journalisation en ligne (online) :

```
SQL> ALTER DATABASE [database]
2      RENAME FILE 'filename' [, 'filename']...
3      TO 'filename']...
```

Configuration des fichiers de journalisation en ligne

Pour déterminer le nombre de fichiers de journalisation en ligne nécessaires à une instance de base de données, vous devez tester différentes configurations.

Dans certains cas, une instance de base de données n'a besoin que de deux groupes. Dans d'autres cas, elle peut nécessiter des groupes supplémentaires pour garantir leur disponibilité pour le processus LGWR. Par exemple, si des messages du fichier trace LGWR ou du fichier d'alertes indiquent que le processus LGWR doit souvent attendre la fin d'un point de reprise ou l'archivage d'un groupe pour accéder à un groupe, vous devez ajouter des groupes.

Bien que les groupes multiplexés du serveur Oracle puissent contenir un nombre de membres différent, tentez de créer une configuration symétrique. Une configuration asymétrique ne doit être que temporaire et résulter d'une situation anormale, telle qu'une défaillance du disque.

Définir l'emplacement des fichiers de journalisation en ligne

Lorsque vous créez des fichiers de journalisation en ligne multiplexés, placez les membres d'un groupe sur différents disques. Ainsi, l'instance ne s'arrête pas lorsqu'un membre n'est pas disponible, alors que d'autres membres le sont.

Placez les fichiers de journalisation archivés et les fichiers de journalisation en ligne sur des disques différents afin de réduire la contention entre les processus d'arrière-plan ARCn et LGWR.

Placez les fichiers de données et les fichiers de journalisation en ligne sur des disques différents afin de réduire la contention entre les processus LGWR et DBWn et limiter les risques de perte de ces fichiers en cas de défaillance physique.

La taille minimale d'un fichier de journalisation en ligne est de 50 Ko, et sa taille maximale est fonction du système d'exploitation. Les membres des groupes peuvent avoir des tailles différentes. Toutefois, la différence de taille ne procure aucun avantage.

Des groupes de tailles différentes ne doivent exister que temporairement lorsque vous souhaitez changer la taille des membres des groupes de fichiers de journalisation en ligne. Dans ce cas, vous devez créer de nouveaux groupes de taille différente et supprimer les anciens groupes.

La configuration des fichiers de journalisation en ligne peut être affectée par :

- le nombre de changements de fichier de journalisation et de points de reprise,
- le nombre d'entrées de journalisation et leur volume,
- l'espace de stockage disponible sur le support (une bande, par exemple, si l'archivage est activé).

Supprimer un groupe

Dans l'exemple ci-dessus, le groupe de fichiers de journalisation 3 et les fichiers du système d'exploitation associés à chaque membre du fichier journal OMF du groupe 3 sont supprimés.

Fichiers de journalisation archivés

L'une des principales décisions d'un administrateur de base de données consiste à déterminer si la base de données doit être configurée pour fonctionner en mode ARCHIVELOG ou NOARCHIVELOG.

Mode NOARCHIVELOG

En mode NOARCHIVELOG, lorsque les fichiers de journalisation en ligne sont complets, ils sont remplacés et un changement de fichier de journalisation se produit. Le processus LGWR ne remplace pas un groupe de fichiers de journalisation tant que le processus de point de reprise de ce groupe n'est pas terminé.

Mode ARCHIVELOG

Si la base de données est configurée pour fonctionner en mode ARCHIVELOG, les groupes inactifs de fichiers de journalisation en ligne complets doivent être archivés. Dans la mesure où toutes les modifications apportées à la base de données sont enregistrées dans les fichiers de journalisation en ligne, l'administrateur de base de données peut utiliser la sauvegarde physique et les fichiers de journalisation en ligne archivés pour récupérer la base de données sans perdre les données validées. Vous pouvez archiver les fichiers de journalisation en ligne de deux manières :

- manuellement,
- automatiquement (méthode recommandée).

Le paramètre d'initialisation LOG_ARCHIVE_START indique si l'archivage doit être automatique ou manuel lors du démarrage de l'instance.

- TRUE indique que l'archivage est automatique. ARCn lance l'archivage du groupe de fichiers de journalisation complet à chaque changement de fichier.
- La valeur par défaut FALSE indique que l'administrateur de base de données archive manuellement les fichiers de journalisation complets. Il doit exécuter manuellement une commande à chaque fois qu'il souhaite archiver un fichier de journalisation en ligne. Vous pouvez archiver manuellement tous les fichiers de ce type ou uniquement certains d'entre eux.

CHAPITRE 6

Gérer les Fichiers de Données et Les Tables Spaces

Tablespaces et fichiers de données

Les bases de données, les tablespaces et les fichiers de données sont très proches, mais ils présentent d'importantes différences.

- Une base de données Oracle est composée d'une ou de plusieurs unités de stockage logiques appelées tablespaces, qui de manière collective stockent toutes les données de la base de données.
- Chaque tablespace d'une base de données Oracle contient un ou plusieurs fichiers appelés fichiers de données. Ces fichiers sont des structures physiques conformes au système d'exploitation sur lequel s'exécute le serveur Oracle.
- Les données d'une base sont stockées de manière collective dans les fichiers de données constituant chaque tablespace de la base de données. Par exemple, la base de données Oracle la plus simple comportera un tablespace et un fichier de données. Une autre base de données pourra comporter trois tablespaces, chacun étant composé de deux fichiers de données (soit un total de six fichiers de données).

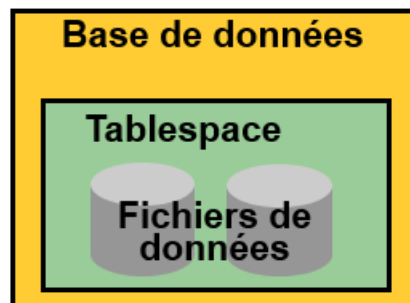


Figure 21 : structure de fichier de données

Types de tablespace

L'administrateur de base de données crée des tablespaces pour améliorer le contrôle et faciliter la gestion de la base. Le serveur Oracle accepte deux types de tablespace : le tablespace SYSTEM et tous les autres tablespaces.

Le tablespace SYSTEM :

- est créé en même temps que la base de données,
- doit exister dans toutes les bases de données,
- contient le dictionnaire de données, y compris les programmes stockés,
- contient le segment d'annulation SYSTEM,
- ne doit pas contenir de données utilisateur, bien que cela soit possible.

Les tablespaces non SYSTEM :

- facilitent l'administration de la base de données,
- séparent les segments d'annulation, les segments temporaires, les segments de données d'application et les segments d'index d'application,
- séparent les données en fonction des besoins de sauvegarde,
- séparent les données dynamiques des données statiques,
- gèrent la quantité d'espace allouée aux objets utilisateur.

Gestion de l'espace dans les tablespaces

Les tablespaces affectent de l'espace dans les extents. Lors de leur création, vous pouvez choisir l'une ou l'autre des méthodes de gestion de l'espace libre et utilisées suivantes :

Tablespaces gérés localement : les extents sont gérés dans le tablespace via des bitmaps. Chaque bit du bitmap correspond à un bloc ou à un groupe de blocs. Lorsqu'un extent est alloué ou libéré pour être réutilisé, le serveur Oracle modifie les valeurs bitmap pour indiquer le nouveau statut des blocs. Cette méthode est utilisée par défaut dans Oracle9i.

Tablespaces gérés au moyen du dictionnaire : Les extents sont gérés à l'aide du dictionnaire de données. Le serveur Oracle met à jour les tables appropriées dans le dictionnaire de données chaque fois qu'un extent est alloué ou libéré.

Tablespaces gérés localement

L'option LOCAL (option par défaut) de la clause EXTENT MANAGEMENT permet d'indiquer qu'un tablespace doit être géré localement.

extent_management_clause:

```
[ EXTENT MANAGEMENT [ DICTIONARY | LOCAL  
[ AUTOALLOCATE | UNIFORM [SIZE integer[K|M]] ] ] ]
```

Où :

DICTIONARY indique que le tablespace est géré à l'aide des tables du dictionnaire.

LOCAL indique que le tablespace est géré localement à l'aide d'un bitmap. Si vous définissez LOCAL, vous ne pouvez pas définir DEFAULT storage_clause, MINIMUM EXTENT ou TEMPORARY.

AUTOALLOCATE (option par défaut) indique que le tablespace est géré par le système (les utilisateurs ne peuvent pas définir de taille d'extent).

UNIFORM indique que le tablespace est géré à l'aide d'extents d'une taille uniforme de SIZE octets. Utilisez K ou M pour définir la taille d'extent en kilo-octets ou en mégaoctets. La taille par défaut est de 1 Mo.

Tablespaces gérés localement (suite)

La clause EXTENT MANAGEMENT peut être utilisée dans diverses commandes CREATE :

- Pour un tablespace permanent non SYSTEM, vous pouvez définir EXTENT MANAGEMENT LOCAL dans la commande CREATE TABLESPACE.
- S'il s'agit d'un tablespace TEMPORARY, vous pouvez définir EXTENT MANAGEMENT LOCAL dans la commande CREATE TEMPORARY TABLESPACE.

Avantages des tablespaces gérés localement :

Les tablespaces gérés localement présentent les avantages suivants par rapport aux tablespaces gérés au moyen du dictionnaire :

- La gestion locale permet d'éviter les opérations récursives de gestion d'espace qui peuvent se produire avec les tablespaces gérés au moyen du dictionnaire, lorsque l'utilisation ou la libération d'espace dans un extent provoque l'exécution d'une autre opération qui utilise ou libère l'espace dans un segment d'annulation ou une table du dictionnaire de données.
- Dans la mesure où les tablespaces gérés localement n'enregistrent pas l'espace libre dans des tables du dictionnaire de données, ils réduisent la contention au niveau des tables.

- La gestion locale des extents identifie automatiquement les espaces libres adjacents, ce qui évite d'avoir à fusionner les extents libres.
- Le système peut déterminer automatiquement la taille des extents gérés localement.
- La modification des bitmaps d'extent ne génère pas d'informations d'annulation, car elle ne met à jour aucune table dans le dictionnaire de données (sauf dans certains cas précis, tels que les informations sur les quotas de tablespace).

Tablespaces gérés au moyen du dictionnaire

Il est possible de personnaliser le stockage des segments des tablespaces gérés au moyen du dictionnaire. Cette méthode est plus souple que celle applicable aux tablespaces gérés localement, mais elle s'avère beaucoup moins efficace.

Tablespace d'annulation

Un tablespace d'annulation est utilisé dans la gestion automatique des annulations (AUM). Pour plus d'informations, voir le chapitre "Gérer les données d'annulation".

```
CREATE UNDO TABLESPACE tablespace
[DATAFILE clause]
```

Tablespaces TEMPORARY

Vous pouvez gérer plus efficacement l'espace pour les opérations de tri en définissant des tablespaces TEMPORARY réservés exclusivement aux segments de tri. Aucun objet de schéma permanent ne peut résider dans un tablespace TEMPORARY.

Un segment de tri ou un segment temporaire est utilisé lorsqu'un segment est partagé par plusieurs opérations de tri. Les tablespaces TEMPORARY améliorent les performances lorsque plusieurs tris ne peuvent tenir dans la mémoire. Le segment de tri d'un tablespace TEMPORARY donné est créé lors de la première opération de tri dans l'instance. La taille du segment de tri augmente par allocation d'extents jusqu'à ce qu'elle soit égale ou supérieure au nombre total des demandes de stockage de tous les tris en cours exécutés dans l'instance.

Les tablespaces temporaires gérés localement contiennent des fichiers de données temporaires (tempfiles) semblables aux fichiers de données standard, à ceci près que :

- les fichiers temporaires sont toujours en mode NOLOGGING,

- les fichiers temporaires ne peuvent pas être en lecture seule,
- vous ne pouvez pas renommer un fichier temporaire,
- vous ne pouvez pas créer un fichier temporaire à l'aide de la commande ALTER DATABASE,
- des fichiers temporaires sont requis pour les bases de données en lecture seule,
- la restauration physique ne permet pas de restaurer les fichiers temporaires,

Pour optimiser les performances d'une opération de tri dans un tablespace temporaire, affectez au paramètre UNIFORM SIZE une valeur multiple de celle du paramètre SORT_AREA_SIZE.

Tablespace TEMPORARY par défaut

Lorsque vous ne définissez pas de tablespace TEMPORARY par défaut à la création de la base de données, le tablespace SYSTEM est affecté à tous les utilisateurs créés sans la clause TEMPORARY TABLESPACE. Par ailleurs, un avertissement indiquant que le tablespace SYSTEM est le tablespace temporaire par défaut est placé dans le fichier alert_*sid*.log. Si vous créez un tablespace TEMPORARY

par défaut en même temps que la base de données, le tablespace SYSTEM ne sera pas utilisé à des fins de stockage temporaire.

Pour définir un tablespace TEMPORARY par défaut une fois la base de données créée, vous devez créer un tablespace TEMPORARY, puis modifier la base de données.

```
SQL> ALTER DATABASE DEFAULT TEMPORARY TABLESPACE temp;
```

Les utilisateurs qui n'ont pas été explicitement affectés à un tablespace TEMPORARY seront alors affectés au tablespace TEMPORARY par défaut.

Vous pouvez modifier la base de données temporaire par défaut à tout moment à l'aide de la commande ALTER DATABASE DEFAULT TEMPORARY TABLESPACE. En cas de modification du tablespace TEMPORARY par défaut, tous les utilisateurs affectés à ce tablespace sont réaffectés au nouveau tablespace par défaut.

Créer un tablespace TEMPORARY par défaut

Pendant la création de la base de données :

Lorsque vous ne définissez pas de tablespace TEMPORARY par défaut à la création de la base de données, le tablespace SYSTEM est affecté à tous les utilisateurs créés sans la clause TEMPORARY TABLESPACE. Par ailleurs, un avertissement est placé dans le fichier alert_sid.log, indiquant que le tablespace SYSTEM est le tablespace TEMPORARY par défaut.

Si vous créez un tablespace TEMPORARY par défaut en même temps que la base de données, le tablespace SYSTEM ne sera pas utilisé à des fins de stockage temporaire. Lorsqu'un tablespace TEMPORARY par défaut est créé à l'aide de la commande CREATE DATABASE, il est géré localement.

Une fois la base de données créée :

Un tablespace TEMPORARY par défaut peut être créé et défini à l'aide de :

- la commande CREATE TABLESPACE pour créer un tablespace TEMPORARY,
- la commande ALTER DATABASE comme indiqué ci-dessus.

Les utilisateurs qui n'ont pas été explicitement affectés à un tablespace temporaire seront alors affectés au tablespace temporaire par défaut.

Vous pouvez modifier le tablespace TEMPORARY par défaut à tout moment à l'aide de la commande ALTER DATABASE DEFAULT TEMPORARY TABLESPACE. En cas de modification du tablespace TEMPORARY par défaut, tous les utilisateurs affectés à ce tablespace sont réaffectés au nouveau tablespace par défaut.

Restrictions relatives au tablespace TEMPORARY par défaut

Supprimer un tablespace TEMPORARY par défaut

Avant de supprimer le tablespace TEMPORARY par défaut, vous devez en créer un nouveau à l'aide de la commande ALTER DATABASE. L'ancien tablespace n'est supprimé que lorsque le nouveau est disponible. Les utilisateurs affectés à l'ancien tablespace sont automatiquement réaffectés au nouveau.

Modifier le type de tablespace TEMPORARY par défaut

Dans la mesure où le tablespace TEMPORARY par défaut doit être le tablespace SYSTEM ou un tablespace TEMPORARY, vous ne pouvez pas le transformer en tablespace permanent.

Mettre le tablespace TEMPORARY par défaut hors ligne (offline)

La mise hors ligne des tablespaces vise à rendre cette partie de la base de données indisponible pour d'autres utilisateurs (sauvegarde hors ligne, maintenance ou modification d'une application qui utilise

le tablespace, par exemple). Etant donné que ces situations ne s'appliquent pas aux tablespaces TEMPORARY, vous ne pouvez pas mettre hors ligne un tablespace TEMPORARY par défaut.

Tablespaces accessibles en lecture seule

La commande ALTER TABLESPACE [tablespace] READ ONLY place le tablespace en mode lecture seule de transition. Ce mode n'autorise pas les opérations d'écriture dans le tablespace, excepté pour l'annulation de transactions existantes qui ont préalablement modifié des blocs du tablespace. Une fois toutes les transactions existantes validées ou annulées, la commande en lecture seule s'exécute et place le tablespace en lecture seule.

Vous pouvez supprimer des éléments, tels que des tables et des index, d'un tablespace en lecture seule, car ces commandes n'affectent que le dictionnaire de données. Cette opération est possible car la commande DROP met à jour le dictionnaire de données et non les fichiers physiques qui constituent le tablespace. Pour les tablespaces gérés localement, le segment supprimé est converti en segment temporaire pour éviter la mise à jour du bitmap. Pour rendre accessible en écriture un tablespace qui est en lecture seule, tous ses fichiers de données doivent être en ligne (online). La mise en lecture seule d'un tablespace crée un point de reprise dans les fichiers de données du tablespace.

La mise en lecture seule d'un tablespace permet d'empêcher les opérations d'écriture dans les fichiers de données de ce tablespace. Les fichiers de données peuvent donc résider sur des supports en lecture seule, tels que des CD-ROM ou des unités non réinscriptibles. Grâce aux tablespaces en lecture seule, il n'est pas nécessaire de sauvegarder de grandes sections de données statiques d'une base de données.

Pour créer un tablespace accessible en lecture seule sur un lecteur non réinscriptible :

1. ALTER TABLESPACE...READ ONLY
2. Utilisez la commande appropriée du système d'exploitation pour transférer les fichiers de données d'un tablespace vers un lecteur accessible en lecture seule.
3. ALTER TABLESPACE...RENAME DATAFILE

Mettre un tablespace hors ligne

Un tablespace est généralement en ligne, ce qui permet aux utilisateurs de la base d'accéder aux données qu'il contient. Toutefois, l'administrateur de base de données peut le mettre hors ligne pour :

- rendre une partie de la base de données indisponible tout en permettant l'accès normal au reste de la base,
- sauvegarder un tablespace hors ligne (bien qu'il soit possible de sauvegarder un tablespace pendant qu'il est en ligne et en cours d'utilisation),
- restaurer un tablespace ou un fichier de données lorsque la base de données est ouverte,
- déplacer un fichier de données lorsque la base est ouverte.

Statut hors ligne d'un tablespace

Lorsque vous mettez un tablespace hors ligne, le serveur Oracle ne permet pas aux instructions SQL qui suivent de faire référence aux objets contenus dans le tablespace. Dans ce cas, les utilisateurs qui tentent d'accéder à ces objets reçoivent un message d'erreur.

La mise hors ligne ou en ligne des tablespaces est enregistrée dans le dictionnaire de données et le fichier de contrôle. Si un tablespace est hors ligne lorsque vous arrêtez une base de données, il reste hors ligne et n'est pas vérifié au remontage et à la réouverture de la base de données.

Lorsque certaines erreurs se produisent (par exemple, si le processus DBWn tente plusieurs fois en vain d'écrire dans un fichier de données du tablespace), l'instance Oracle met automatiquement hors ligne un tablespace en ligne.

Mettre un tablespace hors ligne

Lorsque la base de données est ouverte, l'administrateur de base de données peut mettre hors ligne n'importe quel tablespace, à l'exception du tablespace SYSTEM ou des tablespaces contenant des segments d'annulation ou temporaires actifs. Lorsque vous mettez un tablespace hors ligne, le serveur Oracle met tous les fichiers de données associés hors ligne.

ALTER TABLESPACE tablespace

{ONLINE | OFFLINE [NORMAL] | TEMPORARY | IMMEDIATE | FOR RECOVER]}

Où :

NORMAL supprime de la mémoire SGA tous les blocs de tous les fichiers de données du tablespace (action par défaut). Il n'est pas nécessaire d'effectuer une restauration physique du tablespace avant de le remettre en ligne. Dans la mesure du possible, utilisez la clause NORMAL.

TEMPORARY exécute un point de reprise pour les fichiers de données en ligne du tablespace uniquement, même si certains fichiers ne peuvent pas être écrits. Les fichiers hors ligne peuvent nécessiter une restauration physique.

IMMEDIATE ne garantit pas la disponibilité des fichiers du tablespace et n'exécute pas de point de reprise. Vous devez effectuer une restauration physique du tablespace avant de le remettre en ligne.

FOR RECOVER met les tablespaces hors ligne en vue d'une récupération jusqu'à un point dans le temps.

Redimensionner un tablespace

Vous pouvez augmenter la taille d'un tablespace de deux manières :

- en modifiant automatiquement ou manuellement la taille d'un fichier de données,
- en ajoutant un fichier de données au tablespace.

Indiquer AUTOEXTEND pour un nouveau fichier de données

La clause AUTOEXTEND permet d'activer ou de désactiver l'extension automatique des fichiers de données. La taille des fichiers est incrémentée selon vos indications mais est limitée par une valeur maximale.

Avantages de l'utilisation de la clause AUTOEXTEND :

- Le besoin d'intervention immédiate est réduit lorsque l'espace disponible est insuffisant dans le tablespace.
- Les applications ne s'arrêtent plus à cause des pannes d'allocation des extents.

Lors de la création d'un fichier de données, utilisez les commandes SQL suivantes pour activer l'augmentation automatique de la taille du fichier de données :

- CREATE DATABASE
- CREATE TABLESPACE ... DATAFILE
- ALTER TABLESPACE ... ADD DATAFILE

Ajouter des fichiers de données à un tablespace

Utilisez la commande ALTER TABLESPACE ADD DATAFILE pour ajouter des fichiers de données à un tablespace afin d'augmenter l'espace disque total qui lui est alloué.

Méthodes de déplacement des fichiers de données

Selon le type de tablespace, l'administrateur de base de données peut déplacer des fichiers de données en utilisant l'une des méthodes suivantes :

Commande ALTER TABLESPACE

La commande ALTER TABLESPACE présentée ci-dessous ne s'applique qu'aux fichiers de données d'un tablespace non SYSTEM qui ne contiennent pas de segments d'annulation ou temporaires actifs :

```
ALTER TABLESPACE tablespace  
    RENAME DATAFILE 'filename'[, 'filename']...  
    TO 'filename'[, 'filename']...
```

Les noms des fichiers source doivent correspondre à ceux stockés dans le fichier de contrôle.

Commande ALTER DATABASE

La commande ALTER DATABASE permet de déplacer tous les types de fichier de données.

```
ALTER DATABASE [database]  
    RENAME FILE 'filename'[, 'filename']...  
    TO 'filename'[, 'filename']...
```

Dans la mesure où le tablespace SYSTEM ne peut pas être mis hors ligne, vous devez utiliser cette méthode pour déplacer ses fichiers de données.

Pour renommer les fichiers des tablespaces qui ne peuvent pas être mis hors ligne, procédez comme suit :

1. Arrêtez la base de données.
2. Utilisez la commande de système d'exploitation appropriée pour déplacer les fichiers.
3. Montez la base de données.
4. Exécutez la commande ALTER DATABASE RENAME FILE.
5. Ouvrez la base de données.

Supprimer des tablespaces

Vous pouvez utiliser la commande SQL DROP TABLESPACE pour supprimer un tablespace de la base de données lorsque le tablespace et son contenu ne sont plus utiles :

```
DROP TABLESPACE tablespace  
[INCLUDING CONTENTS [AND DATAFILES] [CASCADE CONSTRAINTS]]
```

Où :

tablespace correspond au nom du tablespace à supprimer.

INCLUDING CONTENTS supprime tous les segments du tablespace.

AND DATAFILES supprime les fichiers du système d'exploitation associés.

CASCADE CONSTRAINTS supprime les contraintes d'intégrité référentielles des tables situées en dehors du tablespace qui font référence aux clés primaires et uniques des tables du tablespace supprimé.

Instructions :

- Un tablespace qui contient toujours des données ne peut pas être supprimé sans l'option INCLUDING CONTENTS. Cette option peut générer un grand nombre d'annulations lorsque le tablespace contient de nombreux objets.
- Une fois qu'un tablespace a été supprimé, ses données ne figurent plus dans la base de données.
- Lorsque vous supprimez un tablespace, seuls les pointeurs de fichier du fichier de contrôle de la base de données associée sont supprimés. Les fichiers du système d'exploitation sont conservés et doivent être supprimés explicitement à l'aide de la commande appropriée du système d'exploitation si la clause AND DATAFILES n'est pas utilisée ou si les fichiers de données ne sont pas des fichiers OMF.
- Vous pouvez supprimer un tablespace en lecture seule et les segments qu'il contient.
- Il est recommandé de mettre le tablespace hors ligne avant de le supprimer, afin d'empêcher d'éventuelles transactions d'accéder à ses segments.

CHAPITRE 7

Gérer les Utilisateurs

Utilisateurs et sécurité

Domaine de sécurité

L'administrateur de base de données définit le nom des utilisateurs autorisés à accéder à la base de données. Un domaine de sécurité définit les paramètres qui s'appliquent à l'utilisateur.

Mécanisme d'authentification

Trois méthodes permettent d'authentifier un utilisateur qui souhaite accéder à la base de données :

- Par le dictionnaire de données
- Par le système d'exploitation
- Par le réseau ;

La méthode d'authentification est précisée lorsque vous définissez l'utilisateur dans la base de données. Elle peut ensuite être modifiée. Le présent chapitre porte uniquement sur l'authentification par la base de données et par le système d'exploitation.

Quotas de tablespace

Les quotas de tablespace contrôlent la quantité d'espace physique de stockage allouée à un utilisateur dans les tablespaces de la base de données.

Tablespace par défaut

Le tablespace par défaut définit l'emplacement de stockage des segments créés par un utilisateur si celui-ci n'indique pas explicitement de tablespace lors de la création des segments.

Tablespace temporaire

Le tablespace temporaire définit l'emplacement d'allocation des extents (ensembles de blocs contigus) par le serveur Oracle lorsque l'utilisateur exécute une opération dans laquelle les données triées sont écrites sur disque.

Verrouillage de compte

Vous pouvez verrouiller les comptes pour empêcher les utilisateurs de se connecter à la base de données. Le verrouillage peut être effectué automatiquement, mais l'administrateur de base de données peut également verrouiller ou déverrouiller les comptes manuellement.

Limites relatives aux ressources

Vous pouvez définir des limites pour l'utilisation de ressources telles que le temps CPU, les entrées/sorties (E/S) logiques et le nombre de sessions ouvertes par l'utilisateur.

Privilèges directs

Les privilèges directs permettent de contrôler les actions qu'exécute l'utilisateur dans la base de données.

Privilèges de rôle

Vous pouvez accorder indirectement des privilèges à un utilisateur à l'aide de rôles.

Schéma de base de données

Un schéma est un ensemble nommé d'objets, tels que des tables, des vues, des clusters, des procédures et des packages, associés à un utilisateur. Lorsque vous créez un utilisateur de base de données, un schéma correspondant portant le nom de l'utilisateur est créé. Un utilisateur ne peut être associé qu'à un schéma du même nom ; par conséquent, *nom utilisateur* et *schéma* sont souvent utilisés indifféremment.

La diapositive montre les objets dont les utilisateurs peuvent être propriétaires dans une base de données Oracle.

Créer un utilisateur : authentification par la base de données

Syntaxe :

Utilisez la commande suivante pour créer un utilisateur :

```
CREATE USER user
IDENTIFIED {BY password | EXTERNALLY}
[ DEFAULT TABLESPACE tablespace ]
[ TEMPORARY TABLESPACE tablespace ]
[ QUOTA {integer [K | M] | UNLIMITED} ON tablespace
[ QUOTA {integer [K | M] | UNLIMITED} ON tablespace      ]... ]
[ PASSWORD EXPIRE ]
[ ACCOUNT { LOCK | UNLOCK } ]
[ PROFILE { profile | DEFAULT } ]
```

Où :

BY password indique que l'utilisateur est authentifié par la base de données et qu'il doit fournir un mot de passe pour se connecter.

EXTERNALLY indique que l'utilisateur est authentifié par le système d'exploitation.

GLOBALLY AS indique que l'utilisateur est authentifié de façon globale.

DEFAULT TABLESPACE ou TEMPORARY TABLESPACE désigne le tablespace par défaut ou le tablespace temporaire de l'utilisateur.

QUOTA définit l'espace maximum alloué aux objets détenus par l'utilisateur dans le tablespace (le quota peut être défini par un entier représentant des octets ou des kilo-octets et des mégaoctets. Le mot-clé UNLIMITED permet d'indiquer que les objets détenus par l'utilisateur peuvent utiliser l'ensemble de l'espace disponible du tablespace. Par défaut, aucun quota de tablespace n'est affecté aux utilisateurs).

PASSWORD EXPIRE force l'utilisateur à réinitialiser le mot de passe lorsqu'il se connecte à la base de données à l'aide de SQL*Plus (cette option n'est valide que si l'utilisateur est authentifié par la base de données).

ACCOUNT LOCK/UNLOCK permet de verrouiller ou de déverrouiller explicitement le compte de l'utilisateur (la valeur par défaut est UNLOCK).

PROFILE permet de contrôler l'utilisation des ressources et de définir le mécanisme de contrôle par mot de passe à appliquer à l'utilisateur.

Créer un utilisateur : authentification par le système d'exploitation

Authentification par le système d'exploitation

Utilisez la clause IDENTIFIED EXTERNALLY de la commande CREATE USER pour indiquer que l'utilisateur doit être authentifié par le système d'exploitation. Cette option est généralement utile lorsque l'utilisateur se connecte directement à la machine sur laquelle s'exécute le serveur Oracle.

Modifier les quotas de tablespace d'un utilisateur

Utilisez la commande suivante pour modifier les quotas d'espace de tablespace ou redéfinir les tablespaces :

```
ALTER USER user  
[ DEFAULT TABLESPACE tablespace]  
[ TEMPORARY TABLESPACE tablespace]  
[ QUOTA {integer [K | M] | UNLIMITED } ON tablespace  
[ QUOTA {integer [K | M] | UNLIMITED } ON tablespace ]      ...]
```

Si vous définissez un quota de 0, les objets de l'utilisateur sont conservés dans le tablespace révoqué, mais aucun nouvel espace ne peut leur être alloué. Par exemple, si le tablespace USERS contient une table de 10 Mo et qu'un quota de 0 est affecté à ce tablespace, plus aucun extent ne peut être alloué à la table.

Les options non modifiées ne sont pas affectées.

Remarque : Le privilège UNLIMITED TABLESPACE est prioritaire sur les paramètres des quotas.

Supprimer un utilisateur

```
DROP USER user [CASCADE]
```

Règles :

- L'option CASCADE supprime tous les objets du schéma avant de supprimer l'utilisateur. Elle doit être définie si le schéma contient des objets.
- Vous ne pouvez pas supprimer un utilisateur connecté au serveur Oracle.

Obtenir des informations sur les utilisateurs

L'interrogation ci-après permet de rechercher le tablespace par défaut (default_tablespace) de tous les utilisateurs.

```
SQL> SELECT username, default_tablespace  
2 FROM dba_users;
```

CHAPITRE 8

Gérer les Privilèges

Privilèges

Un privilège est un droit d'exécution d'un type donné d'instruction SQL ou un droit d'accès à l'objet d'un autre utilisateur. Il autorise son détenteur à :

- se connecter à une base de données,
- créer une table,
- sélectionner des lignes dans la table d'un autre utilisateur,
- exécuter la procédure stockée d'un autre utilisateur.

Privilèges système

Chaque privilège système permet à un utilisateur d'effectuer une opération spécifique ou une catégorie d'opérations sur la base de données. Par exemple, le privilège lié à la création de tablespaces est un privilège système.

Privilèges objet

Chaque privilège objet autorise un utilisateur à exécuter une action spécifique sur un objet tel qu'une table, une vue, une séquence, une procédure, une fonction ou un package.

Dans sa tâche de contrôle des privilèges, un administrateur de base de données est chargé :

- d'autoriser un utilisateur à effectuer un type d'opération,
- d'accorder et de révoquer le droit d'effectuer certaines opérations système,
- d'accorder des privilèges directement à des utilisateurs ou à des rôles,
- d'accorder des privilèges à tous les utilisateurs (PUBLIC).

Privilèges système

Ces privilèges peuvent être classés comme suit :

- Privilèges autorisant l'exécution d'opérations sur l'ensemble du système, tels que CREATE SESSION, CREATE TABLESPACE
- Privilèges autorisant la gestion d'objets dans un schéma propre à l'utilisateur, tels que CREATE TABLE
- Privilèges autorisant la gestion d'objets de n'importe quel schéma, tels que CREATE ANY TABLE

Les privilèges peuvent être gérés à l'aide des commandes LDD GRANT et REVOKE, qui permettent d'accorder et de révoquer des privilèges système à un utilisateur ou à un rôle (pour plus d'informations sur les rôles, voir le chapitre "Gérer les rôles").

Privilèges système : exemples

- Il n'existe pas de privilège CREATE INDEX.
- Le privilège CREATE TABLE permet d'exécuter les commandes CREATE INDEX et ANALYZE. L'utilisateur doit disposer d'un quota de tablespace ou du privilège UNLIMITED TABLESPACE.
- Les privilèges tels que CREATE TABLE, CREATE PROCEDURE et CREATE CLUSTER impliquent la suppression des objets concernés.
- Le privilège UNLIMITED TABLESPACE ne peut pas être accordé à un rôle.
- Le privilège DROP ANY TABLE est requis pour vider une table appartenant à un autre schéma.

Accorder des privilèges système

Utilisez l'instruction SQL GRANT pour accorder des privilèges système aux utilisateurs.

Le bénéficiaire peut ensuite accorder un privilège système à d'autres utilisateurs avec l'option ADMIN, mais cette opération doit s'effectuer avec précaution. En effet, ces privilèges sont généralement réservés aux administrateurs de la sécurité et sont rarement accordés à d'autres utilisateurs.

```
GRANT {system_privilege|role}
      [, {system_privilege|role} ]...
      TO {user|role|PUBLIC}
      [, {user|role|PUBLIC} ]...
      [WITH ADMIN OPTION]
```

Où :

system_privilege : désigne le privilège système à accorder.

role : désigne le nom du rôle à accorder.

PUBLIC : accorde le privilège système à tous les utilisateurs.

WITH ADMIN OPTION : autorise le bénéficiaire à accorder son privilège ou son rôle à d'autres utilisateurs ou rôles.

Privilèges SYSDBA et SYSOPER

Seul l'administrateur de base de données peut se connecter à une base de données avec des privilèges administrateur. Lorsqu'un utilisateur se connecte à une base de données en tant que SYSDBA, il dispose de tous les privilèges lui permettant d'exécuter n'importe quelle opération sur la base ou sur les objets de celle-ci.

Restrictions relatives aux privilèges système

Le mécanisme de protection du dictionnaire de données d'Oracle9i empêche les utilisateurs non autorisés d'accéder aux objets du dictionnaire.

L'accès aux objets du dictionnaire est limité aux rôles SYSDBA et SYSOPER. En effet, les privilèges système qui permettent d'accéder aux objets d'autres schémas n'autorisent pas l'accès aux objets du dictionnaire. Par exemple, le privilège SELECT ANY TABLE vous permet d'accéder à des vues et des tables d'autres schémas, mais ne vous autorise pas à sélectionner des objets du dictionnaire (tables de base, vues, packages et synonymes).

Si le paramètre possède la valeur TRUE, l'accès aux objets du schéma SYS est autorisé (fonction Oracle7). Si sa valeur est FALSE, les privilèges SYSTEM qui permettent d'accéder aux objets d'autres schémas n'autorisent pas l'accès aux objets du schéma du dictionnaire de données.

Par exemple, si O7_DICTIONARY_ACCESSIBILITY=FALSE, l'instruction SELECT ANY TABLE autorise l'accès aux vues et tables de tous les schémas, à l'exception de SYS (les dictionnaires sont inaccessibles, par exemple). Le privilège système EXECUTE ANY PROCEDURE permet d'accéder aux procédures de n'importe quel autre schéma, à l'exception de celles du schéma SYS.

Révoquer des privilèges système

Vous pouvez révoquer des privilèges système à l'aide de l'instruction SQL REVOKE. Tout utilisateur disposant d'un privilège système avec l'option ADMIN OPTION peut révoquer le privilège de n'importe quel autre utilisateur de la base de données, même si ce n'est pas lui qui l'a accordé.

```
REVOKE {system_privilege|role}
[, {system_privilege|role} ]...
FROM {user|role|PUBLIC}
[, {user|role|PUBLIC} ]...
```

Remarque :

- La commande REVOKE permet uniquement la révocation des privilèges système accordés directement à l'aide de la commande GRANT.
- La révocation de privilèges système peut avoir un impact sur certains objets dépendants. Par exemple, si vous accordez le privilège SELECT ANY TABLE à un utilisateur qui a créé des procédures ou des vues utilisant une table d'un autre schéma, la révocation de ce privilège invalide ces procédures ou vues.

Révoquer des privilèges système (suite)

La révocation d'un privilège système n'entraîne aucun effet de cascade, que le privilège ait été accordé avec l'option ADMIN OPTION ou non.

Suivez les étapes de l'exemple suivant :

Scénario

1. L'administrateur de base de données accorde le privilège système CREATE TABLE à Jeff avec l'option ADMIN OPTION.
2. Jeff crée une table.
3. Jeff accorde le privilège système CREATE TABLE à Emi.
4. Emi crée une table.
5. L'administrateur de base de données révoque le privilège système CREATE TABLE précédemment accordé à Jeff.

Résultat :

La table de Jeff n'est pas supprimée, mais il ne peut plus créer de table.

La table d'Emi n'est pas supprimée et cette dernière dispose toujours du privilège système CREATE TABLE.

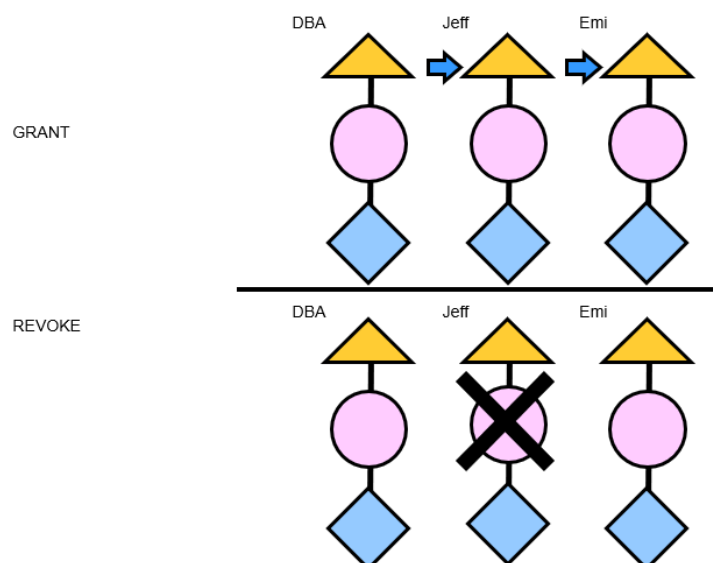


Figure 22 : Gestion des privilèges systèmes Oracle

Privilèges objet

Un privilège objet est un privilège ou droit autorisant la réalisation d'une action donnée sur une table, une vue, une séquence, une procédure, une fonction ou un package spécifique. Chaque objet présente un ensemble propre de privilèges pouvant être accordés. Le tableau de la diapositive répertorie les privilèges correspondant à différents objets. Ainsi, les seuls privilèges qui s'appliquent à une séquence sont SELECT et ALTER. Vous pouvez restreindre les privilèges UPDATE, REFERENCES et INSERT en précisant un sous-ensemble de colonnes pouvant être mises à jour. Vous pouvez limiter un droit de type SELECT en créant une vue présentant un sous-ensemble de colonnes et en accordant le privilège SELECT sur la vue. Un privilège accordé sur un synonyme octroie directement un droit sur la table de base référencée par ce synonyme.

Remarque : La diapositive ne fournit pas la liste exhaustive des privilèges objet.

Accorder des privilèges objet

```
GRANT { object_privilege [(column_list)]
      [, object_privilege [(column_list)] ]...
      | ALL [PRIVILEGES]}
ON    [schema.]object
TO    {user|role|PUBLIC}[, {user|role|PUBLIC} ]...
      [WITH GRANT OPTION]
```

Où :

object_privilege désigne le privilège objet à accorder.

column_list désigne une colonne de table ou de vue (cette valeur ne peut être définie que lors de l'octroi du privilège INSERT, REFERENCES ou UPDATE).

ALL accorde tous les privilèges objet qui ont été accordés avec l'option WITH GRANT OPTION.

ON object identifie l'objet sur lequel les privilèges doivent être accordés.

WITH GRANT OPTION autorise le bénéficiaire à accorder ses privilèges objet à d'autres utilisateurs ou rôles.

Utilisez l'instruction GRANT pour accorder des privilèges objet.

- Pour cela, l'objet doit se trouver dans votre schéma ou vous devez avoir reçu le privilège avec l'option GRANT OPTION.
- Par défaut, si vous disposez d'un objet, tous les privilèges associés vous sont automatiquement accordés.
- Par souci de sécurité, soyez prudent lorsque vous accordez à d'autres utilisateurs des privilèges sur vos objets.

Révoquer des privilèges objet

L'instruction REVOKE permet de révoquer des privilèges objet. L'utilisateur qui révoque un privilège objet doit être celui qui l'a accordé.

Utilisez la commande suivante pour révoquer des privilèges objet :

```
REVOKE { object_privilege
      [, object_privilege ]...
      | ALL [PRIVILEGES] }
ON    [schema.]object
FROM {user|role|PUBLIC}
      [, {user|role|PUBLIC} ]...
      [CASCADE CONSTRAINTS]
```

Où :

object_privilege désigne le privilège objet à révoquer.

ALL révoque tous les privilèges objet accordés à l'utilisateur.

ON désigne l'objet sur lequel les privilèges objet doivent être révoqués.

FROM identifie les utilisateurs ou les rôles dont les privilèges objet sont révoqués.

CASCADE CONSTRAINTS supprime toutes les contraintes d'intégrité référentielle définies à l'aide du privilège REFERENCES ou ALL.

Restriction :

Les utilisateurs qui accordent des privilèges objet peuvent les révoquer uniquement aux utilisateurs auxquels ils les ont accordés.

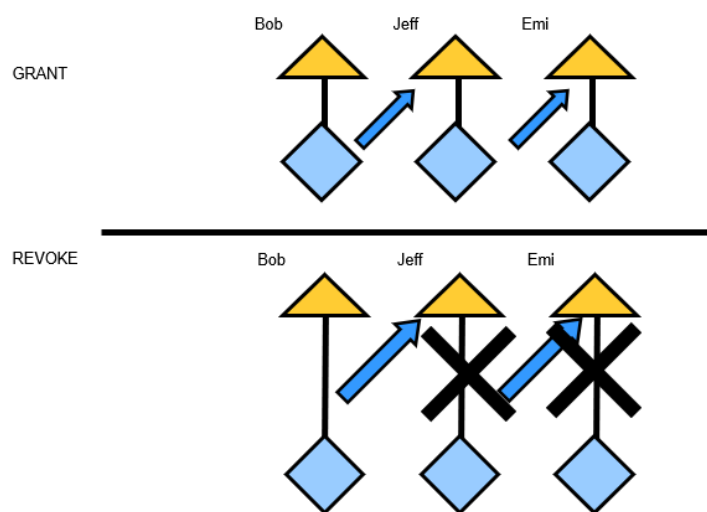


Figure 23 : Gestion des privilèges Objets Oracle

Vous pouvez observer des effets de cascade lorsque vous révoquez un privilège système lié à une opération LMD. Par exemple, si le privilège SELECT ANY TABLE est accordé à un utilisateur qui a créé des procédures utilisant la table, toutes les procédures contenues dans le schéma de l'utilisateur doivent être recompilées pour être réutilisées.

La révocation de privilèges objet accordés avec l'option WITH GRANT OPTION provoque également un effet de cascade.

Suivez les étapes de l'exemple suivant :

Scénario

- Jeff reçoit le privilège objet SELECT sur la table EMPLOYEES avec l'option GRANT OPTION.
- Jeff accorde ce privilège sur la table EMPLOYEES à Emi.
- Par la suite, le privilège SELECT de Jeff est révoqué en entraînant la révocation du privilège d'Emi.

Obtenir des informations sur les privilèges

- DBA_SYS_PRIVS affiche la liste des privilèges système accordés aux utilisateurs et aux rôles.
- SESSION_PRIVS affiche la liste des privilèges auxquels l'utilisateur a accès.
- DBA_TAB_PRIVS affiche la liste de tous les octrois de privilèges sur les objets de la base de données.
- DBA_COL_PRIVS décrit tous les octrois de privilèges sur les colonnes de la base de données.

CHAPITRE 9

Gestion de Rôles

Définition d'un rôle

Oracle permet de gérer et de contrôler aisément les privilèges à l'aide de rôles. Les rôles sont des groupes nommés de privilèges associés qui sont accordés à des utilisateurs ou à d'autres rôles. Ils facilitent l'administration des privilèges dans une base de données.

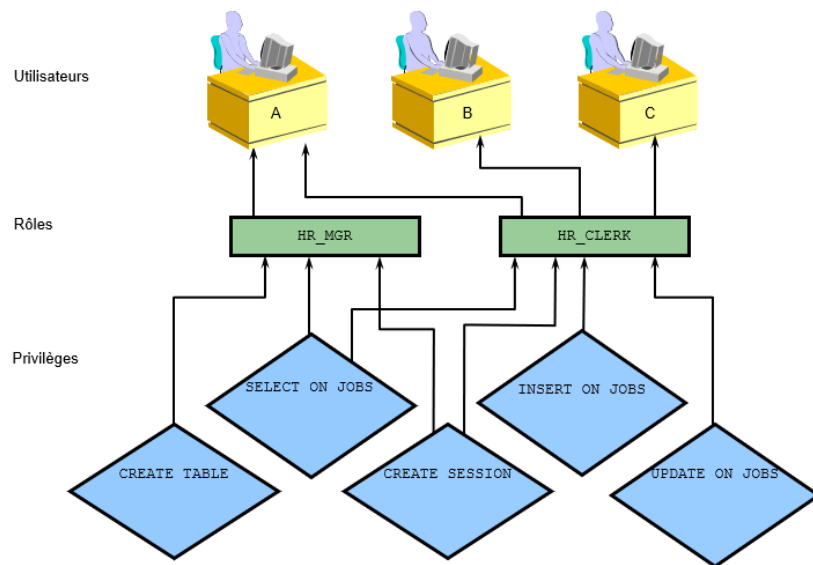


Figure 24 : Les rôles

Les rôles :

- sont accordés et révoqués à l'aide des commandes qui permettent d'accorder et de révoquer des privilèges système.
- peuvent être accordés à tout utilisateur ou rôle. En revanche, un rôle ne peut pas être accordé à lui-même ou de façon circulaire.
- peuvent être constitués de privilèges système et de privilèges objet.
- peuvent être activés ou désactivés pour chaque utilisateur auquel ils ont été accordés.
- peuvent nécessiter un mot de passe pour être activés.
- doivent posséder un nom unique différent des noms utilisateur et des noms de rôle existants.
- n'ont pas de propriétaire et ne se trouvent dans aucun schéma.
- sont décrits dans le dictionnaire de données.

Avantages des rôles

Gestion simplifiée des privilèges

Utilisez des rôles pour simplifier la gestion des privilèges. Au lieu d'accorder les mêmes privilèges à plusieurs utilisateurs, vous pouvez accorder les privilèges à un rôle et associer ce rôle à chaque utilisateur.

Gestion dynamique des privilèges

En cas de modification des privilèges associés à un rôle, tous les utilisateurs auxquels ce rôle a été accordé bénéficient automatiquement et immédiatement des nouveaux privilèges.

Disponibilité sélective des privilèges

Vous pouvez activer et désactiver les rôles pour activer et désactiver temporairement les privilèges. L'activation d'un rôle permet également de vérifier qu'un utilisateur dispose de ce rôle.

Octroi possible via le système d'exploitation

Vous pouvez utiliser des commandes ou des utilitaires du système d'exploitation pour accorder des rôles aux utilisateurs dans la base de données.

Créer des rôles

Si vous disposez du privilège système CREATE ROLE, vous pouvez créer des rôles à l'aide de l'instruction CREATE ROLE. Lorsque vous créez un rôle défini comme NOT IDENTIFIED, IDENTIFIED EXTERNALLY ou BY password, Oracle l'accorde avec l'option ADMIN.

Utilisez la commande suivante pour créer un rôle :

```
CREATE ROLE role [NOT IDENTIFIED | IDENTIFIED  
                {BY password | EXTERNALLY | GLOBALLY | USING package}]
```

Où :

role : correspond au nom du rôle.

NOT IDENTIFIED : indique qu'aucune vérification n'est nécessaire lorsque le rôle est activé.

IDENTIFIED : indique qu'une vérification est nécessaire lorsque le rôle est activé.

BY password : fournit le mot de passe que l'utilisateur doit indiquer pour activer le rôle.

USING package : crée un rôle d'application, qui ne peut être activé que par des applications utilisant un package autorisé.

EXTERNALLY : indique que l'utilisateur doit avoir reçu une autorisation d'un service externe (tel que le système d'exploitation ou un service tiers) pour activer le rôle.

GLOBALLY : indique que l'utilisateur doit être autorisé par le service de répertoire d'entreprise à utiliser le rôle pour l'activer à l'aide de l'instruction SET ROLE ou à la connexion.

Rôles prédéfinis

Les rôles présentés dans le tableau sont automatiquement définis pour les bases de données Oracle à l'exécution des scripts de création de base de données. Les rôles CONNECT, RESOURCE et DBA sont fournis pour assurer une compatibilité descendante avec les versions antérieures du serveur Oracle. Les rôles EXP_FULL_DATABASE et IMP_FULL_DATABASE sont fournis pour faciliter l'utilisation des utilitaires Import et Export.

Les rôles DELETE_CATALOG_ROLE, EXECUTE_CATALOG_ROLE et SELECT_CATALOG_ROLE permettent d'accéder aux vues et aux packages du dictionnaire de données. Ils sont accordés aux utilisateurs qui ne disposent pas du rôle DBA, mais qui doivent accéder aux vues et aux tables du dictionnaire de données.

Autres rôles spéciaux

Le serveur Oracle crée également d'autres rôles qui vous permettent d'administrer la base de données. Sur la plupart des systèmes d'exploitation, il s'agit des rôles OSOPER et OSDBA. Leur nom peut varier selon le système d'exploitation utilisé.

D'autres rôles sont définis dans les scripts SQL fournis avec la base de données. Par exemple, AQ_ADMINISTRATOR_ROLE permet d'administrer Advanced Queuing.

Modifier des rôles

Vous ne pouvez modifier un rôle que pour changer sa méthode d'authentification. Pour cela, vous devez disposer du rôle approprié avec l'option ADMIN ou du privilège système ALTER ANY ROLE.

Utilisez la commande suivante pour modifier un rôle :

```
ALTER ROLE role {NOT IDENTIFIED | IDENTIFIED  
{BY password | USING package| EXTERNALLY | GLOBALLY }};
```

Où :

role : correspond au nom du rôle.

NOT IDENTIFIED : indique qu'aucune vérification n'est nécessaire lorsque le rôle est activé.

IDENTIFIED : indique qu'une vérification est nécessaire lorsque le rôle est activé.

BY password : fournit le mot de passe permettant d'activer le rôle.

EXTERNALLY : indique que l'utilisateur doit avoir reçu une autorisation d'un service externe (tel que le système d'exploitation ou un service tiers) pour activer le rôle.

GLOBALLY : indique que l'utilisateur doit être autorisé par le service de répertoire d'entreprise à utiliser le rôle pour l'activer à l'aide de l'instruction SET ROLE ou à la connexion.

Accorder des rôles

Pour accorder un rôle à un utilisateur, utilisez la même commande que pour accorder un privilège système :

```
GRANT role [, role ]...  
TO {user|role|PUBLIC}  
[, {user|role|PUBLIC} ]...  
[WITH ADMIN OPTION]
```

Où :

role : correspond à un ensemble de rôles à accorder.

PUBLIC : accorde le rôle à tous les utilisateurs.

WITH ADMIN OPTION : permet au bénéficiaire d'accorder le rôle à d'autres utilisateurs ou rôles. Si vous accordez un rôle avec cette option, le bénéficiaire peut l'accorder à d'autres utilisateurs ou le révoquer, le modifier ou le supprimer.

L'utilisateur qui crée un rôle le reçoit de façon implicite avec l'option ADMIN OPTION. Un utilisateur qui n'a pas reçu de rôle avec cette option doit disposer du privilège système GRANT ANY ROLE pour accorder des rôles aux autres utilisateurs ou révoquer des rôles accordés.

Remarque : Le paramètre d'initialisation MAX_ENABLED_ROLES définit le nombre maximum de rôles de base de données que les utilisateurs peuvent activer.

Rôles par défaut

Un utilisateur peut disposer d'un grand nombre de rôles. Un rôle par défaut est un sous-ensemble de ces rôles activé automatiquement lorsque l'utilisateur se connecte. Par défaut, tous les rôles d'un utilisateur sont activés à la connexion sans indication de mot de passe. La commande ALTER USER permet de limiter les rôles par défaut d'un utilisateur.

La clause DEFAULT ROLE s'applique uniquement aux rôles qui ont été accordés directement à l'utilisateur via une instruction GRANT. Elle ne permet pas d'activer :

- les rôles qui n'ont pas été accordés à l'utilisateur,
- les rôles qui ont été accordés par l'intermédiaire d'autres rôles,
- les rôles gérés par un service externe (le système d'exploitation, par exemple).

Utilisez la commande suivante pour accorder des rôles par défaut à un utilisateur :

```
ALTER USER user DEFAULT ROLE  
{role [,role]... | ALL [EXCEPT role [,role]... ] | NONE}
```

Où :

user : correspond au nom de l'utilisateur qui reçoit les rôles.

role : correspond au rôle par défaut de l'utilisateur.

ALL : transforme tous les rôles accordés à l'utilisateur en rôles par défaut, à l'exception de ceux figurant dans la clause EXCEPT (valeur par défaut).

EXCEPT : indique que les rôles qui suivent ne doivent pas être inclus dans les rôles par défaut.

NONE : ne convertit aucun des rôles accordés à l'utilisateur en rôle par défaut (lors de la connexion, l'utilisateur ne dispose que des privilèges qui lui ont été accordés directement).

Etant donné que les rôles doivent être accordés pour pouvoir être définis comme rôles par défaut, vous ne pouvez pas définir de rôles par défaut à l'aide de la commande CREATE USER.

Rôles d'application

La clause de package USING de l'instruction CREATE ROLE crée un rôle d'application. Seules les applications utilisant un package PL/SQL autorisé peuvent activer un rôle d'application. Les développeurs d'applications n'ont pas besoin d'intégrer des mots de passe aux applications pour sécuriser un rôle. Ils peuvent en effet créer un rôle d'application et désigner le package PL/SQL autorisé à l'activer.

```
SQL> CREATE ROLE admin_role IDENTIFIED USING hr.employees;
```

Dans l'exemple, admin_role est un rôle d'application qui ne peut être activé que par les modules définis dans le package PL/SQL hr.employees.

Activer et désactiver les rôles

Activez ou désactivez les rôles pour activer ou désactiver temporairement les privilèges associés. Un rôle ne peut être activé que s'il a été accordé à l'utilisateur.

Lorsqu'un rôle est activé, l'utilisateur peut utiliser les privilèges associés. Lorsque le rôle est désactivé, l'utilisateur ne peut pas utiliser les privilèges associés s'ils ne lui ont pas été accordés directement ou s'ils n'ont pas été accordés à un autre rôle activé pour cet utilisateur. Les rôles ne sont activés que pour une session. A la session suivante, les rôles actifs de l'utilisateur sont remplacés par les rôles par défaut.

Définir les rôles à activer

La commande SET ROLE et la procédure DBMS_SESSION.SET_ROLE permettent d'activer tous les rôles inclus dans la commande et de désactiver tous les autres. Vous pouvez activer les rôles à partir de n'importe quel outil ou programme autorisant les commandes PL/SQL à l'exception des procédures stockées.

Utilisez la commande ALTER USER...DEFAULT ROLE pour indiquer les rôles qui seront activés à la connexion de l'utilisateur. Tous les autres rôles seront alors désactivés.

Un mot de passe peut être nécessaire pour activer un rôle. Celui-ci doit être inclus dans la commande SET ROLE. Les rôles par défaut accordés à un utilisateur ne nécessitent pas de mot de passe ; ils sont activés à la connexion, comme les rôles sans mot de passe.

Restrictions

Vous ne pouvez pas activer un rôle à partir d'une procédure stockée, car cette action peut modifier le domaine de sécurité (ensemble de privilèges) qui a permis d'appeler la procédure. Ainsi, dans les instructions PL/SQL, vous pouvez activer et désactiver des rôles dans des blocs anonymes et des procédures d'application (par exemple, des procédures Oracle Forms), mais pas dans des procédures stockées.

Activer et désactiver les rôles

La commande SET ROLE désactive tous les autres rôles accordés à l'utilisateur.

```
SET ROLE {role [ IDENTIFIED BY password ]  
        [, role [ IDENTIFIED BY password ] ]...  
        | ALL [ EXCEPT role [, role ] ...]  
        | NONE }
```

Où :

role : correspond au nom du rôle.

IDENTIFIED BY password : fournit le mot de passe permettant d'activer le rôle.

ALL : active tous les rôles accordés à l'utilisateur en cours, à l'exception de ceux figurant dans la clause EXCEPT (vous ne pouvez pas utiliser cette option pour activer des rôles nécessitant des mots de passe).

EXCEPT role : n'active pas les rôles indiqués.

NONE : désactive tous les rôles de la session en cours (seuls les privilèges accordés directement à l'utilisateur sont actifs).

L'option ALL sans la clause EXCEPT ne fonctionne que lorsque les rôles activés ne nécessitent pas de mot de passe.

Révoquer des rôles accordés à des utilisateurs

L'instruction SQL REVOKE permet de révoquer un rôle accordé à un utilisateur. Tout utilisateur possédant un rôle avec l'option ADMIN peut retirer ce rôle à un autre utilisateur ou rôle de la base de données. Par ailleurs, les utilisateurs disposant du privilège GRANT ANY ROLE peuvent révoquer n'importe quel rôle.

```
REVOKE role [, role ]  
        FROM {user|role|PUBLIC}  
        [, {user|role|PUBLIC} ]
```

où :

role : correspond au rôle à révoquer ou au rôle à partir duquel des rôles doivent être révoqués.

user : définit l'utilisateur dont les privilèges système ou les rôles doivent être révoqués.

PUBLIC : révoque le privilège ou le rôle accordé à tous les utilisateurs.

Supprimer des rôles

Pour supprimer un rôle de la base de données, utilisez la commande suivante :

```
DROP ROLE role
```

Lorsque vous supprimez un rôle, le serveur Oracle le révoque pour tous les utilisateurs et tous les rôles auxquels il était accordé et le supprime de la base de données.

Vous pouvez supprimer le rôle s'il vous a été accordé avec l'option ADMIN OPTION ou si vous disposez du privilège système DROP ANY ROLE.

Instructions relatives à la création de rôles

Etant donné qu'un rôle contient les privilèges nécessaires à l'exécution d'une tâche, le nom du rôle correspond généralement à une tâche d'application ou à un intitulé de poste. L'exemple de la diapositive utilise des tâches d'application et des intitulés de poste comme noms de rôle. Pour créer et accorder des rôles à des utilisateurs, procédez comme suit :

1. Créez un rôle pour chaque tâche d'application. Le nom du rôle d'application correspond à une tâche de l'application, telle que PAYROLL.
2. Accordez au rôle d'application les privilèges nécessaires à l'exécution de la tâche.
3. Créez un rôle pour chaque type d'utilisateur. Le nom du rôle utilisateur correspond à un intitulé de poste, tel que PAY_CLERK.
4. Accordez des rôles d'application aux rôles de l'utilisateur.
5. Accordez les rôles de l'utilisateur à d'autres utilisateurs.

Si une modification de l'application implique l'octroi de nouveaux privilèges pour exécuter la tâche Payroll, il suffit à l'administrateur de base de données d'accorder les nouveaux privilèges au rôle d'application PAYROLL. Tous les utilisateurs qui exécutent cette tâche reçoivent les nouveaux privilèges.

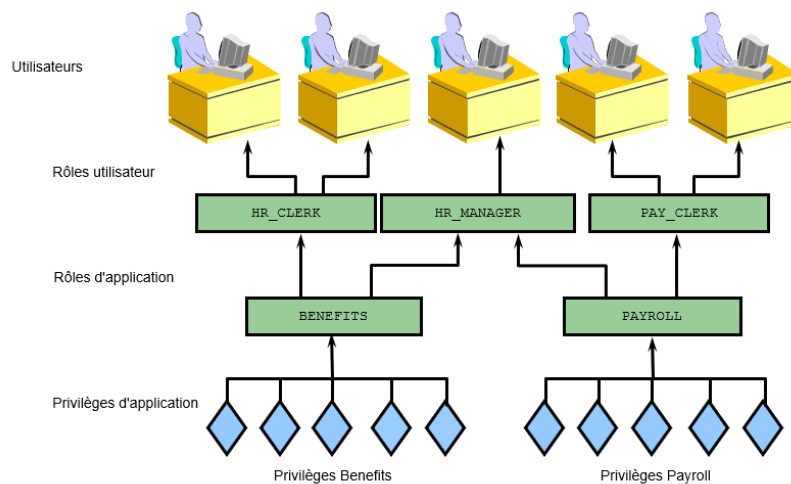


Figure 25 : Création des rôles

Règles d'utilisation des mots de passe et des rôles par défaut

Les mots de passe permettent de renforcer la sécurité lorsqu'un rôle est activé. Par exemple, l'application peut demander à l'utilisateur d'entrer un mot de passe pour activer le rôle PAY_CLERK, car ce rôle peut servir à émettre des chèques.

Les mots de passe ne permettent d'activer un rôle que par l'intermédiaire d'une application. Cette méthode est décrite dans l'exemple de la diapositive.

- L'administrateur de base de données a accordé les rôles PAY_CLERK et PAY_CLERK_RO à l'utilisateur.
- Le rôle PAY_CLERK a reçu tous les privilèges nécessaires à l'exécution de la fonction d'établissement des bulletins de salaire.
- Le rôle PAY_CLERK_RO (RO signifiant Read Only, lecture seule) dispose uniquement de privilèges SELECT sur les tables nécessaires à l'exécution de la fonction d'établissement des bulletins de salaire.
- L'utilisateur peut se connecter à SQL*Plus pour effectuer des interrogations, mais il ne peut pas modifier les données, car il ne connaît pas le mot de passe du rôle PAY_CLERK, qui n'est pas un rôle par défaut.
- Lorsque l'utilisateur se connecte à l'application d'établissement des bulletins de salaire, il active le rôle PAY_CLERK en fournissant le mot de passe approprié. Le mot de passe étant codé dans le programme, aucun message ne demande à l'utilisateur de l'entrer.

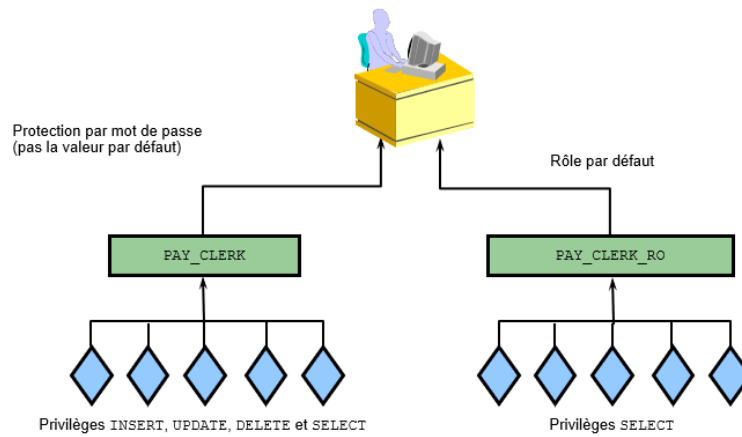


Figure 26 : utilisation des rôles par défaut

Rechercher les informations sur les rôles

La plupart des vues du dictionnaire de données qui contiennent des informations sur les privilèges des utilisateurs indiquent également si le rôle requiert un mot de passe.

```
SQL> SELECT role, password_required
```

```
2 FROM dba_roles;
```