

Module n°2

Création de base de données

1Z0-031

Auteur : Mathieu Musard
Création de base de données
Nombre de pages : 44

Table des matières

1. PROCÉDURE DE CRÉATION D'UNE BASE DE DONNÉES.....	4
1.1. PRÉ REQUIS NÉCESSAIRES À LA CRÉATION.....	4
1.1.1. <i>Gérer et sauvegarder les fichiers d'une base de données</i>	4
1.1.2. <i>Créer une base de données</i>	5
1.2. ENVIRONNEMENT DU SYSTÈME D'EXPLOITATION	6
1.2.1. <i>Création de fichiers de mots de passe</i>	6
1.3. FICHER DE PARAMÈTRES ET INSTANCE.....	7
1.3.1. <i>Création des fichiers de paramètres</i>	7
1.3.2. <i>Démarrage en mode NOMOUNT de l'instance</i>	7
1.4. CRÉER UNE BASE DE DONNÉES	8
1.4.1. <i>Assistant de configuration de base de données</i>	8
1.4.2. <i>Les options de la commande CREATE DATABASE</i>	8
1.4.3. <i>Créer une base de données en mode commande</i>	10
1.4.4. <i>Les conditions d'erreur lors de la création d'une base de données</i>	10
1.4.5. <i>Le contenu d'une base de données après sa création</i>	10
1.4.6. <i>Créer une base de données en utilisant OMF</i>	11
2. UTILISATION ET CONTENU DU DICTIONNAIRE DE DONNÉES.....	12
2.1. STRUCTURE DU DICTIONNAIRE DE DONNÉES	12
2.1.1. <i>Présentation du dictionnaire de données</i>	12
2.1.2. <i>Création et catégories des vues du dictionnaire de données</i>	13
2.1.3. <i>Informations contenues dans le dictionnaire de données</i>	14
2.2. SCRIPTS DU DICTIONNAIRE DE DONNÉES.....	15
2.2.1. <i>Création de fonctions PL/SQL</i>	15
2.2.2. <i>Scripts administratifs</i>	15
3. GESTION DES FICHIERS DE CONTRÔLE ET DE REDO LOG.....	17
3.1. FICHER DE CONTRÔLE.....	17
3.1.1. <i>Contenu des fichiers de contrôle</i>	17
3.1.2. <i>Multiplexage des fichiers de contrôle</i>	18
3.1.3. <i>Gérer les fichiers de contrôle avec OMF</i>	19
3.1.4. <i>Récupérer les informations des fichiers de contrôle</i>	19
3.2. LES FICHIERS DE REDO LOG.....	19
3.2.1. <i>Utiliser les fichiers de redo log</i>	19
3.2.2. <i>Structure des fichiers de redo log</i>	20
3.2.3. <i>Paramètres des fichiers de redo log</i>	20
3.3. GÉRER LES GROUPES DE REDO.....	20
3.3.1. <i>Fonctionnement des fichiers de redo log</i>	20
3.3.2. <i>Forcer une basculement de fichiers log et un point de synchronisation</i>	23
3.3.3. <i>Ajout de groupe de fichiers redo log online</i>	23
3.3.4. <i>Ajout des membres redo log online</i>	24
3.3.5. <i>Relocalisation des fichiers redo log online</i>	24
3.3.6. <i>Suppression de groupes de fichiers redo log online</i>	25
3.3.7. <i>Suppression des membres redo log online</i>	25
3.3.8. <i>Réinitialisation des fichiers redo log online</i>	26
3.4. PLANIFICATION DES FICHIERS REDO LOG ONLINE.....	26
3.4.1. <i>Nombre de fichiers redo log online</i>	26
3.4.2. <i>Emplacement des fichiers redo log online</i>	27
3.4.3. <i>Dimensionnement des fichiers redo log online</i>	27
3.4.4. <i>Gérer les redo log online with OMF</i>	28
3.4.5. <i>Obtenir des informations sur les groupes</i>	28
3.4.6. <i>Archivage de fichiers de redo log</i>	29
4. GÉRER LES TABLESPACES ET LES FICHIERS DE DONNÉES	30
4.1. ARCHITECTURE D'UNE BASE DE DONNÉES.....	30
4.1.1. <i>Vue d'ensemble de la structure d'une base de données</i>	30
4.1.2. <i>Relations</i>	30

4.1.3. Caractéristiques d'un tablespace.....	31
4.1.4. Utilisations d'un tablespace.....	31
4.1.5. Caractéristiques d'un fichier de données.....	32
4.2. CRÉER DES TABLESPACES	32
4.2.1. Types de tablespaces.....	32
4.2.2. Création d'un tablespace.....	33
4.2.3. Gestion de l'espace dans les tablespaces.....	34
4.2.4. Locally Managed tablespaces.....	34
4.2.5. Dictionary Managed Tablespaces	35
4.2.6. Paramètres de stockage.....	35
4.2.7. Les Undo Tablespaces	36
4.3. TABLESPACE TEMPORAIRE.....	37
4.3.1. Tablespace temporaire.....	37
4.3.2. Tablespace temporaire par défaut	37
4.3.3. Restrictions sur les Tablespace temporaire par défaut	37
4.4. MANIPULATION DE TABLESPACE.....	37
4.4.1. Offline Status	37
4.4.2. Tablespace en lecture seul.....	38
4.4.3. Tablespaces en lecture seul sur des supports en lecture seul.....	39
4.4.4. Supprimer un Tablespace.....	39
4.4.5. Conséquences de la suppression de tablespaces.....	40
4.5. FICHIERS DE MANIPULATION DES DONNÉES.....	40
4.5.1. Mettre en place l'extension automatique des fichiers de données	40
4.5.2. Changer la taille des fichiers de données manuellement	40
4.5.3. Ajouter un fichier de données à un Tablespace.....	41
4.5.4. Déplacer un fichier de données : ALTER TABLESPACE	41
4.6. AUTRES OPTIONS SUR LES TABLESPACES.....	43
4.6.1. Configurer OMF pour la création de Tablespace.....	43
4.6.2. Créer des Tablespaces avec OMF.....	43
4.6.3. Récupérer des informations sur les Tablespaces	43

1. Procédure de création d'une base de données

La création de la base de données est la première étape dans l'organisation et la gestion d'un système de base de données.

La création d'une base de données permet à un ensemble de fichiers du système d'exploitation de fonctionner en tant que base de données Oracle.

Indépendamment du nombre de fichiers de données et du nombre d'instances qui accèdent à la base de données, une base de données ne peut être créée qu'une seule fois.

Une base de données initiale peut être créée lors de l'installation d'Oracle. Cette base de données peut être utilisée comme base de données initiale, ou une nouvelle base de données peut être créée.

Pour créer une nouvelle base de données, il faut tenir compte de la taille des fichiers de données, des blocs, des tables et des index. L'emplacement des fichiers de contrôle doit être aussi planifié.

Lors de la création d'une nouvelle base de données, il faut également tenir compte du jeu de caractères que la base de données utilisera. Une fois la base de données créée, le jeu de caractères spécifié ne peut plus être changé, sauf si la base de données est recréée.

1.1. Pré requis nécessaires à la création

Une base de données est créée et gérée par un DBA (Database Administrator). Si le DBA connaît les conditions préalables à la création d'une base de données, il pourra créer efficacement une base de données.

- Pour créer une base de données, un DBA doit posséder un compte privilégié authentifié par le système d'exploitation ou par un fichier de mots de passe. Un compte privilégié d'un utilisateur définit les actions que cet utilisateur peut effectuer sur la base de données ou les objets de la base de données.
- Il doit y avoir suffisamment de mémoire pour démarrer une instance. Ce qui signifie qu'il doit y avoir suffisamment de mémoire pour que la zone mémoire global du système (System Global Area SGA), les fichiers exécutables et les autres processus en arrière plan s'exécutent normalement. La taille minimale de la SGA des petites bases de données est de 6,5 MB.
- L'espace disque disponible sur la machine doit être suffisant pour recevoir la base de données planifiée. Le disque doit comporter suffisamment d'espace de stockage pour stocker les fichiers de contrôle, les fichiers de données et les fichiers redo log online de la base de données.

1.1.1. Gérer et sauvegarder les fichiers d'une base de données

Une base de données peut présenter une taille relativement importante et comporter un grand nombre d'utilisateurs. Aussi, un DBA doit planifier correctement la structure physique d'une base de données de telle façon que l'échec d'un disque ne provoque pas l'arrêt de l'instance de la base de données.

- Créer au moins deux fichiers de contrôle et les stocker sur des disques différents. Si un fichier de contrôle ou un disque sur lequel est stocké un fichier de contrôle est corrompu, la base de données pourra toujours accéder à l'autre fichier de contrôle.
- Les fichiers redo log online doivent être organisés en groupes multiplexés. Un groupe de fichier redo log est constitué des copies identiques des fichiers redo log. Le multiplexage des

groupes de fichiers redo log online permet au processus Log Writer (LGWR) de continuer d'écrire des entrées log des membres disponibles dans un groupe si un membre de ce groupe est indisponible ou corrompu. Le processus LOGWR est un processus d'arrière plan qui écrit les entrées du cache redo log (redo log buffer) sur le disque.

- Les membres d'un groupe de redo log doivent également être stockés sur des disques différents. Ainsi si un disque est corrompu, le LGWR et l'instance de la base de données n'échoueront pas.
- Les fichiers de données contenant des objets de la base de données avec des durées de vie différentes, telles que les données d'une application et les données temporaires, doivent être séparés afin de minimiser la fragmentation.
- Les fichiers de données dont les données participent à la contention sur le disque doivent être séparés sur différents disques.
- Les fichiers de données qui contiennent des données avec des caractéristiques d'administration, différentes doivent être séparés. Par exemple, les objets avec des besoins d'entrées/sorties concurrents, tels que les tables et les index, doivent être séparés. Cette séparation garantit au DBA un bon équilibre des charges d'entrée/sortie.

1.1.2. Créer une base de données

Il est possible de créer une base de données à la main avec un script SQL qui utilise la commande CREATE DATABASE ou tout simplement avec Oracle Database Configuration Assistant.

Attention : Pendant l'installation d'Oracle Server, le DBCA (interface graphique utilisateur qui interagit avec Oracle Universal Installer) est lancé par **OUI (Oracle Universal Installer)** et permet de créer une base de données par défaut. Il est possible pour l'utilisateur de ne pas lancer cette option.

Si la structure de la base de données initiale ne remplit pas les besoins, une nouvelle base de données peut être créée.

Voici les étapes dans la création manuelle d'une base de données :

- Décider d'un nom unique pour l'instance et pour la base de données. Le nom de l'instance sert à identifier les instances associées à la base de données, et le nom de la base de données sert à identifier une base de données spécifique.
- Indiquer le jeu de caractères que la base de données doit utiliser. Ce choix est très important car une fois configuré, le jeu de caractères ne peut être changé pour cette base de données.
- Définir les variables du système d'exploitation. Les variables du système d'exploitation sont définies dans la base de registre. Leur définition détermine l'instance par défaut, le jeu de caractères de la base de données et les autres paramètres nécessaires au bon fonctionnement de la base de données.
- Créer un fichier de paramètres. Ce fichier est nécessaire pour créer et démarrer une instance. Pour créer un fichier de paramètres, il faut spécifier la taille des composants de la zone mémoire globale du système (SGA), le nom de la base de données, les fichiers de contrôle de la base de données et la taille des blocs de données de la base de données.
- Créer les services Oracle. Quand une base de données est créée, les services d'Oracle, tel que le service OraHomeTNSListener81 en environnement Windows NT, sont également créés. Les services d'Oracle préparent une base de données en vue de son utilisation. Le service OraHomeTNSListener81 identifie et accepte les demandes de connexion provenant des applications cliente.

- Créer un fichier de mots de passe. La création d'une base de données implique la création d'un fichier de mots de passe. Le fichier de mots de passe contient les mots de passe pour authentifier des utilisateurs privilégiés autorisés à effectuer des opérations telles que le démarrage et l'arrêt de la base de données.
- Démarrer une instance sans monter la base de données. Lorsqu'une instance est démarrée, la SGA est allouée à l'instance indiquée et les processus d'arrière plan sont démarrés en vue de la création de la base de données.
- Créer la base de données à l'aide de la commande SQL CREATE DATABASE et en définissant les paramètres facultatifs de la commande. Ces paramètres servent à spécifier le nombre maximal de fichier, leurs tailles et le mode de fonctionnement de ces fichiers.
- Exécuter les scripts qui génèrent les vues du dictionnaire de données. Un dictionnaire de données est un ensemble de tables qui fournissent des informations sur la base de données à laquelle elles sont associées.

1.2. Environnement du système d'exploitation

Oracle utilise des paramètres de configuration pour localiser des fichiers et spécifier les informations d'exécution de tous les produits Oracle. Ces informations de configuration sont stockées dans la base de registre du système d'exploitation.

Pour qu'une instance Oracle fonctionne proprement dans un environnement Windows NT, il est essentiel de stocker les informations de configuration adéquates dans la base de registre.

La base de registre Windows NT est une base de données hiérarchique et centralisée qui stocke les informations de configuration du matériel et logiciel d'une machine locale.

Les paramètres de la plupart des produits Oracle sont stockés dans la sous-clé

HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE de la base de registre Windows NT. Les cinq variables de la base de registre sont :

- Pour qu'une instance Oracle fonctionne proprement, il faut préciser dans la base de registres le chemin d'accès du répertoire racine où Oracle a été installé. Ce chemin est indiqué dans la variable ORACLE_HOME.
- Le SID (identificateur du système Oracle) de l'instance, à laquelle un utilisateur se connecte si la chaîne de connexion n'est pas spécifiée, est stocké dans la variable de la base de registre ORACLE_SID. Le nom de l'instance doit être unique pour chaque instance qui s'exécute sur une même machine. La valeur de la variable ORACLE_SID doit comporter un maximum de quatre caractères sous Windows NT et huit caractères sous UNIX. Par exemple, si le nom par défaut de l'instance est ORCL, la variable ORACLE_SID prendra la valeur ORCL.
- Si un jeu de caractères, autre que le jeu de caractères par défaut US7ASCII, est utilisé pour la création d'une base de données, le nom d'un répertoire contenant les informations de conversion pour le jeu de caractère choisi est stocké dans la variable ORA_NLS33. Si la variable ORA_NLS33 n'est pas définie et qu'une base de données est démarrée avec un jeu de caractères autres que celui par défaut, le jeu ne sera pas reconnu par la base de données.
- Pour qu'une instance d'Oracle fonctionne proprement, la variable PATH doit stocker le chemin d'un fichier spécifique.
- La dernière variable est LD_LIBRARY_PATH, elle contient les chemins pour les dossiers du système d'exploitation et des fichiers de la librairie Oracle. Par exemple : ORACLE_HOME/lib.

1.2.1. Création de fichiers de mots de passe

Les opérations telles que l'arrêt et le démarrage de la base de données doivent être effectuées par des utilisateurs autorisés à administrer une base de données. L'exécution de telles opérations par des utilisateurs de base de données non autorisés peut entraîner la perte de données. Par exemple, si un utilisateur arrête une base de données en mode Abort, les données non validées risquent d'être perdues.

Le problème des utilisateurs non autorisés à effectuer des opérations critiques spécifiques peut se résoudre en utilisant un fichier de mots de passe permettant d'authentifier les utilisateurs qui administrent la base de données.

L'exécutable ORAPWD.exe permet d'exécuter l'utilitaire ORAPWD.

1.3.Fichier de paramètres et instance

1.3.1.Création des fichiers de paramètres

Lors de l'installation d'Oracle, une base de données par défaut est créée avec un fichier de paramètre. Un fichier de paramètre est un fichier texte contenant une liste de paramètres d'initialisation pour une instance et pour une base de données particulière.

Il faut donc créer un nouveau fichier init<SID>.ora avec la commande :

```
$cp init.ora $ORACLE_HOME/dbs/initdb01.ora
```

Puis il est nécessaire d'éditer ce fichier pour modifier les paramètres de la base de données (DB_NAME, DB_BLOCK_SIZE ...).

Nous l'avons vu dans le module 1, Oracle 9i intègre un nouveau fichier de paramètre le SPFILE. La commande pour créer un SPFILE à partir du init<SID>.ora est :

```
CREATE SPFILE FROM PFILE;
```

Il est nécessaire d'avoir les privilèges SYSOPER ou SYSDBA pour créer un SPFILE à partir d'un PFILE.

1.3.2.Démarrage en mode NOMOUNT de l'instance

Avant la création de la base de données, il faut démarrer une instance en mode NOMOUNT. En mode NOMOUNT, la SGA est allouée et les processus en arrière plan sont démarrés pour la création de la base de données.

Exemple :

Le DBA veut démarrer l'instance ORCL en mode NOMOUNT en utilisant le fichier de paramètre. Sous une fenêtre SQL*Plus, le DBA va se connecter en tant que « internal » à la base de données.

```
SQL> STARTUP NOMOUNT PFILE=" C:\Oracle\Admin\SID\PFile\initorcl.ora"
ORACLE instance started.
```

Total System Global Area	38322124	bytes
Fixed Size	65484	bytes
Variable Size	21405696	bytes
Database Buffers	16777216	bytes
Redo Buffers	73728	bytes

L'instance ORCL a été démarrée en mode NOMOUNT. Le DBA peut ainsi préparer la création d'une base de données grâce à l'allocation de la SGA et au démarrage des processus en arrière plan de la nouvelle base de données.

1.4. Créer une base de données

1.4.1. Assistant de configuration de base de données

Il existe deux méthodes pour créer une base de données. Soit en utilisant l'assistant de configuration de base de données, soit en utilisant le commande CREATE DATABASE. L'assistant de configuration de base de données est un utilitaire qui se lance sous linux avec la commande *dbca* et qui permet :

- Créer une base de données : Option qui permet de créer une nouvelle base de données ou un template, qui permet de choisir avec fichiers de données ou sans fichiers de données.
- Configurer les options de base de données : Cette option permet d'ajouter des options à la base de données qui n'ont pas été configurées initialement. Par exemple : JVM, interMedia, Oracle OLAP Services...
- Supprimer une base de données : Option qui permet de supprimer une base de données.
- Gérer les templates : Il existe trois moyens de créer un template. Soit à partir d'un template déjà existant, soit à partir d'une base de données, soit à partir d'un template prédéfini. Il existe deux types de templates de base de données. Ceux qui contiennent seulement la structure physique de la base de données (options, tablespaces, datafiles...), ou les templates contenant la structure ainsi que les données contenues dans la base.

Lors de la création d'une base de données avec l'assistant de configuration, il vous sera demandé de choisir parmi l'un des quatre types de base de données suivant :

- **Installation typique ou personnalisée** : L'installation typique est pour une base de données avec un faible trafic. Le mode personnalisé permet de choisir parmi l'un des environnements suivant :
- **OLTP (OnLine Transaction Processing)** : Cet environnement est utilisé pour les grandes bases de données devant gérer des milliers de transactions (lecture, écriture, suppression) par jour. Les performances pour ce type d'environnement sont définies en terme de disponibilité des données.
- **Data Warehousing** : Cet environnement est utilisé avec des base de données qui sont souvent interrogées mais peut modifiées. Les performances de ce type d'environnement sont définies en terme de temps de réponse.
- **Mixed** : Environnement supportant à la fois OLTP et Data Warehousing. Mode par défaut pour l'installation d'une base de données.

1.4.2. Les options de la commande CREATE DATABASE

Le DBA peut créer une base de données en utilisant la commande CREATE DATABASE. La connaissance des fonctions des options de la commande CREATE DATABASE permettra au DBA d'identifier celles qu'il doit spécifier pour créer une base de données adaptée.

```
CREATE DATABASE <database>
```

```
[MAXLOGFILES integer]
```


Spécifie le nombre maximum de groupes de fichier log qui peuvent être créés pour la base de données.
La valeur de cette option affecte la taille du fichier de contrôle.

[**MAXLOGMEMBERS** *integer*]

Spécifie le nombre maximum de fichiers redo log membres d'un groupe de fichiers redo log. La valeur de cette option affecte la taille du fichier de contrôle.

[**MAXLOGHISTORY** *integer*]

Spécifie le nombre maximum de redo log archivés qui peuvent être utilisés pour la récupération automatique physique du serveur Oracle Parallel. La valeur de cette option affecte la taille du fichier de contrôle.

[**MAXDATAFILES** *integer*]

Spécifie le nombre maximum de fichiers de données qui peuvent être créés pour la base de données.
La valeur de cette option affecte la taille du fichier de contrôle.

[**MAXINSTANCES** *integer*]

Spécifie le nombre maximum d'instances pouvant monter et ouvrir la base de données simultanément.
La valeur de cette option affecte la taille du fichier de contrôle.

[**ARCHIVELOG** | **NOARCHIVELOG**]

Spécifie que les redo log doivent être archivés avant d'être réutilisés. L'option NOARCHIVELOG spécifie que les fichiers redo log peuvent être réutilisés sans archiver leur contenu. La valeur de cette option affecte la taille du fichier de contrôle.

[**CHARACTER SET** *charset*]

Spécifie le jeu de caractères utilisé par la base de données pour stocker les données. Si cette option n'est pas précisée, le jeu de caractère par défaut US7ASCII sera utilisé.

[**DATAFILE** *filespec* [, *filespec*]...]

Spécifie le fichier de données à créer pour le tablespace SYSTEM. Cette option est reliée à la clause `autoextend_clause` par la relation :

filespec ::= '*filename*' [**SIZE** *n* [**K**|**M**]] [**REUSE**]

[*autoextend_clause*]

autoextend_clause ::= **AUTOEXTEND** {**OFF**|**ON** [**NEXT** *integer* [**K**|**M**]]
[**MAXSIZE** *integer* [**K**|**M**]] }

active ou désactive l'extension automatique d'un fichier de données.

[**CONTROLFILE REUSE**]

Spécifie que les fichiers de contrôle identifiés dans le fichier de paramètre peuvent être écrasés si ils existent déjà.

[**NATIONAL CHARACTER SET** *charset*]

Spécifie le jeu de caractères national utilisé pour stocker les données dans des colonnes de type NCHAR, NCLOB ou NVARCHAR2.

[**LOGFILE** [**GROUP** *integer*] *filespec* [, [**GROUP** *integer*] *filespec*]]

Spécifie les fichiers log utilisés pour la base de données et les groupes auxquels ces fichiers appartiennent.

[**Defaut Temporary Tablespace**] Création d'un tablespace par défaut, chaque utilisateur ne spécifiant pas un tablespace par défaut sera automatiquement assigné au tablespace par défaut.

[**SET TIME_ZONE**] Fuseau horaire à utiliser pour la base de données.

1.4.3. Créer une base de données en mode commande

La principale responsabilité d'un DBA est de créer et de gérer une ou plusieurs bases de données. Dans ces bases de données peut être stocké un très grand nombre de données qui peuvent être récupérées et analysées. Le script utilisé pour créer une base de données doit être écrit dans un fichier texte. Le script est ensuite exécuté en utilisant SQL*Plus (par exemple).

Exemple de script permettant de créer une base de données :

```
@crdbdb01.sql
SQL> create database db01
 2  logfile
 3  GROUP 1 ('/u01/oradata/db01/log_01_db01.rdo') SIZE 15M,
 4  GROUP 2 ('/u01/oradata/db01/log_02_db01.rdo') SIZE 15M,
 5  GROUP 3 ('/u01/oradata/db01/log_03_db01.rdo') SIZE 15M
 6  datafile '/u01/oradata/db01/system_01_db01.dbf' SIZE 100M
 7  undo tablespace UNDO
 8  datafile '/u01/oradata/db01/undo_01_db01.dbf' SIZE 40M
 9  default temporary tablespace TEMP
10  tempfile '/u01/oradata/db01/temp_01_db01.dbf' SIZE 20M
11  extent management local uniform size 128k
12  character set AL32UTF8
13  national character set AL16UTF16
14  set time_zone = 'America/New_York'
15  ;|
```

1.4.4. Les conditions d'erreur lors de la création d'une base de données

Il existe plusieurs situations lors de la création d'une base de données avec la commande CREATE DATABASE qui peuvent provoquer un échec. La connaissance de ces situations permet au DBA d'identifier les erreurs faites lors de la création d'une base de données.

- Des erreurs de syntaxe dans le script SQL
- Les fichiers que la commande CREATE DATABASE doit créer existent déjà. Cette erreur apparaît lorsque l'option REUSE n'a pas été spécifiée ou lorsque la taille des fichiers ne correspond pas.
- Une erreur se produit au niveau du système, telles que des permissions de fichier et de répertoire non valides.
- Il n'y a pas assez d'espace disque pour créer les fichiers spécifiés dans la commande CREATE DATABASE.

Dans tous les cas d'échec, il faut arrêter l'instance. Il faut ensuite modifier la commande, redémarrer l'instance en mode NOMOUNT et créer à nouveau la base de données.

1.4.5. Le contenu d'une base de données après sa création

Oracle crée automatiquement plusieurs fichiers, segments et tables dans une base de données. La connaissance des éléments qui constituent une base de données permet au DBA d'identifier et de gérer les objets des bases de données.

Lors de la création de la base de données, le tablespace SYSTEM est créé.

Le tablespace SYSTEM est un ensemble de fichiers de données spécifiés dans la commande CREATE DATABASE. Il contient le dictionnaire de données.

Les tables du dictionnaire de données sont créées durant la création de la base de données. Les tables du dictionnaire de données contiennent les informations sur la structure de la base de données, sur les utilisateurs de la base de données et sur l'allocation de l'espace. Les vues du dictionnaire de données ne sont pas créées automatiquement lors de la création de la base de données.

Les vues du dictionnaire de données sont des vues sur les tables du dictionnaire de données.

Les fichiers de contrôle et les fichiers redo log sont créés à la création de la base de données. Les fichiers de contrôle contiennent le nom de la base de données et le chemin d'accès de tous les fichiers de données et fichiers redo log. Lors de la création de la base de données, les fichiers redo log sont vides ; ils enregistrent ensuite tous les modifications apportées aux données de la base de données.

Les deux utilisateurs par défaut SYS et SYSTEM sont créés à la création de la base de données. SYS et SYSTEM ont respectivement les mots de passe par défaut « change_on_install » et « manager ». Les utilisateurs SYS et SYSTEM possèdent tous les privilèges système sur la base de données.

Enfin, lorsque la base de données est créée, le segment de rollback SYSTEM est créé dans le tablespace SYSTEM. Le segment de rollback SYSTEM est utilisé pour enregistrer les modifications apportées dans le tablespace SYSTEM.

1.4.6. Créer une base de données en utilisant OMF

Pour créer une base de données de type OMF, il est nécessaire d'initialiser les paramètres dynamiques DB_CREATE_FILE_DEST et DB_CREATE_ONLINE_DEST_n du fichier de paramètre. Une fois l'initialisation faite, la syntaxe pour créer la base de données est très simple :

```
STARTUP NOMOUNT
CREATE DATABASE
DEFAULT TEMPORARY TABLESPACE TEMP;
```

2.Utilisation et contenu du dictionnaire de données

2.1. Structure du dictionnaire de données

2.1.1.Présentation du dictionnaire de données

Les utilisateurs des bases de données, les développeurs d'applications, les administrateurs de bases de données et le serveur Oracle utilisent le dictionnaire de données comme une source centrale d'information associée à une base de données.

Le dictionnaire de données est un ensemble de tables et de vues qui est utilisé comme une référence fournissant de l'information à jour concernant la base de données.

Le dictionnaire de données est automatiquement mis à jour par Oracle lorsque des commandes de langage de définition des données ou des commandes de manipulation des données, débouchant sur l'extension d'une table, sont exécutées.

Le dictionnaire de données stocke les informations sur :

- La structure logique de la base de données

Par exemple, les informations sur les tablespaces, les blocs de données, les extents et les segments d'une base de données.

- La structure physique de la base de données

Notamment les informations sur les fichiers de contrôle, les fichiers redo log et les fichiers de données.

- Les noms et les définitions des objets

Tels que les tables, les vues, les index, les clusters, les séquences, les triggers (ou déclencheurs), les fonctions, les procédures et les packages mais aussi les informations sur les espaces alloués et l'espace actuellement utilisé par les objets.

- Les contraintes d'intégrité définies pour les tables d'une base de données.

- Les noms des utilisateurs valides de la base de données et les privilèges et les rôles attribués à chaque utilisateur de la base de données.

Les privilèges définissent les opérations qu'un utilisateur peut effectuer dans la base de données. Un rôle est constitué d'un ensemble de privilèges.

- L'audit sur une base de données.

Par exemple, le dictionnaire de données fournit les informations sur les utilisateurs qui ont accédé à des objets ou les ont mis à jour. L'audit correspond à l'enregistrement d'opérations spécifiques en fonction d'un utilisateur, d'un objet ou d'un privilège.

Le dictionnaire de données possède deux composants :

- les tables de base
- les vues du dictionnaire de données

Les tables de base sont les tables réelles d'Oracle qui stockent les informations sur une base de données. Ces tables sont créées avec le script sql.bsq. Ce script est stocké dans le répertoire orant\rdms\admin. Les informations stockées dans les tables de base sont lues et écrites par le serveur Oracle. Ces informations sont rarement accédées directement par les utilisateurs car ces informations sont normalisées et stockées sous une forme encodée.

Les utilisateurs ou administrateurs de bases de données ne doivent pas utiliser de commandes DML, telles que INSERT, UPDATE et DELETE, pour mettre à jour les tables de base directement, à l'exception de la table de trace d'audit lorsque la fonctionnalité d'audit est utilisée.

Les vues du dictionnaire de données sont des vues sur les tables de base. Elles sont créées par le script catalog.sql. Les vues du dictionnaire de données simplifient et résument les informations contenues dans les tables de base. Les vues du dictionnaire stockent également ces informations sous une forme que les utilisateurs de la base de données peuvent lire facilement. Ces vues permettent au DBA de gérer et tuner la base de données.

2.1.2. Création et catégories des vues du dictionnaire de données

La création des vues du dictionnaire de données se fait à partir du script **catalog.sql**, stocké dans ORACLE_HOME\rdbms\admin.

Une fois la base de données ouverte il suffit seulement d'exécuter ce script. Il faut tenir compte que l'exécution de ce script est relativement longue.

Il existe trois principales catégories du dictionnaire de données :

- USER_<vues> :

Vues pouvant être accédées par les utilisateurs de la base de données. USER_<vues> se réfère au schéma d'un utilisateur de la base de données.

Un schéma est un domaine logique appartenant à un utilisateur où les objets créés par cet utilisateur peuvent être stockés.

USER_<vues> affiche les informations sur les objets appartenant à un utilisateur spécifique. Par exemple, la vue USER_TABLES contient les informations sur les tables appartenant à un utilisateur. USER_<vues> affiche les informations sur les privilèges et les rôles attribués par un utilisateur sur des objets créés par cet utilisateur. Les informations fournies par USER_<vues> font partie d'un sous-ensemble des informations fournies par ALL_<vues>.

- ALL_<vues>:

Vues affichant les informations sur des objets auxquels l'utilisateur peut accéder à travers l'obtention publique ou explicite de rôles et de privilèges. Le contenu de ALL_<vues> est un sous-ensemble de DBA_<vues>.

- DBA_<vues>:

Vues fournissant des informations sur tous les objets de la base de données. Ces vues sont généralement interrogées par les DBA ou les utilisateurs qui possèdent le privilège système SELECT ANY TABLE. Ce privilège permet aux utilisateurs d'interroger toutes les tables de la base de données.



2.1.3. Informations contenues dans le dictionnaire de données

Le dictionnaire de données possède des vues variées qui fournissent différents types d'informations.

Vues	Description
dictionary dict_columns	Vues générales
dba_tables dba_objects dba_lobs dba_tab_columns dba_constraints	Informations sur les objets, tels que les tables, les contraintes, les gros objets et les colonnes
dba_users dba_sys_privs dba_roles	Informations sur les privilèges et les rôles des utilisateurs
dba_extents dba_free_space dba_segments	Allocation d'espace pour les objets de la base de données
dba_rollback_segs dba_data_files dba_tablespace	Structures générales de la base de données
dba_audit_trail dba_audit_objects dba_audit_obj_opts	Informations d'audit

La vue `DICTIONARY` du dictionnaire de données fournit une vue d'ensemble de toutes les données du dictionnaire de données. L'interrogation de la vue `DICTIONARY` évite d'avoir à se souvenir de toutes les vues et des informations fournis par chacune de ces vues.

La vue `DICTIONARY` possède deux colonnes : `TABLE_NAME` et `COMMENT`. Ces colonnes indiquent la vue qui devra ensuite être interrogée pour extraire des informations.

Par exemple, pour récupérer les informations concernant les tables accessibles à un utilisateur, il faut interroger la vue `ALL_CATALOG`.

Exemple :

L'objectif est de récupérer les noms des segments de rollback de la base de données en interrogeant la vue DICTONARY.

```
SQL> SELECT *
  2 FROM dictionary
  3 WHERE table_name LIKE UPPER('%ROLLBACK%');

TABLE_NAME                                COMMENTS
-----
DBA_ROLLBACK_SEGS                        Description of rollback segments

1 row selected.
```

Cette requête affiche le nom des vues qui contiennent des informations sur les segments de rollback.

Pour afficher les noms des segments de rollback :

```
SQL> SELECT segment_name
  2 FROM dba_rollback_segs;

SEGMENT_NAME
-----
SYSTEM
RB_TEMP
RB1
RB2
RB3
...
```

Pour afficher la liste des colonnes d'une vue, il faut utiliser le mot clé :

```
DESCRIBE V$INSTANCE
```

2.2. Scripts du dictionnaire de données

2.2.1. Création de fonctions PL/SQL

Pour qu'un serveur Oracle fonctionne efficacement et de façon optimale, la base de données doit supporter les fonctionnalités PL/SQL.

Le script catproc.sql est utilisé pour ajouter des fonctionnalités PL/SQL à une base de données. Ce script lance tous les scripts utilisés ou nécessaires avec PL/SQL. Il crée également des packages PL/SQL qui étendent les fonctionnalités du SGBDR et des vues supplémentaires de gestion des files d'attente de messages et de restauration des tablespaces. Catproc.sql est stocké sous le répertoire ora92\rdbms\admin. Pour l'exécuter, il suffit d'ouvrir une session SQL*Plus et de lancer l'exécution du script avec la commande : @<chemin complet du script>.

2.2.2. Scripts administratifs

Pour gérer une base, il peut être nécessaire au DBA de créer des structures supplémentaires, telles que des tables, des vues et des packages.

Les scripts administratifs sont séparés en quatre catégories de fichiers se trouvant dans le répertoire :

- Les scripts utl*.sql :

Ils créent des vues et des tables additionnelles pour les utilitaires de la base de données. Par exemple, le script utlsamp.sql crée et remplit les tables d'exemple EMP, DEPT, SALGRADE et BONUS de l'utilisateur SCOTT.

- Les scripts cat*.sql :

Ils créent des vues du dictionnaire de données et des tables de base du dictionnaire de données. Par exemple, catalog.sql crée des vues du dictionnaire pour une base de données, catman.sql crée des tables de base pour la récupération et des vues pour l'utilitaire Recovery Manager.

- Les scripts dbms*.sql :

Ils créent des spécifications de package de base de données. Par exemple, le script dbmspool.sql crée la spécification d'un package qui permet d'afficher la taille des objets présent dans la shared pool.

- Les scripts prvt*.plb :

Ils créent le corps de package.

3. Gestion des fichiers de contrôle et de redo log

3.1. Fichier de contrôle

Le fichier de contrôle file est un petit fichier binaire, nécessaire pour démarrer et maintenir une base de données. Le fichier de contrôle doit être disponible chaque fois que la base de données est montée ou ouverte.

Lorsqu'une instance monte la base de données, elle lit le fichier de contrôle pour localiser les fichiers de données et les fichiers redo log online.

Une des fonctions du fichier de contrôle est de fournir les informations les plus récentes sur la base de données à l'instance Oracle mise à jour en permanence pendant l'utilisation de la base de données.

Le fichier de contrôle fournit également des informations sur la cohérence de la base de données. Ces informations sont utilisées lors de la restauration de la base de données.

Si le fichier de contrôle utilisé par la base de données devient indisponible, la base de données ne peut pas fonctionner proprement. Le fichier de contrôle peut devenir indisponible pour un grand nombre de raisons telles que la défaillance du disque ou la corruption du fichier.

Certains mots-clés utilisés pendant la création d'une base de données, affectent la taille du fichier de contrôle. Ceci est surtout visible quand les paramètres ont de très grandes valeurs. Les paramètres suivants affectent la taille du fichier de contrôle :

- MAXLOGFILES
- MAXLOGMEMBERS
- MAXLOGHISTORY
- MAXDATAFILES
- MAXINSTANCES

3.1.1. Contenu des fichiers de contrôle

Le fichier de contrôle stocke des informations permettant à la base de données de fonctionner correctement.

Le fichier de contrôle stocke :

- Le nom et l'identifiant de la base de données
- Les noms et les emplacements des fichiers de données et des fichiers redo log en ligne. Ces informations permettent au serveur Oracle de localiser les fichiers de la base de données sur le disque.
- Le nom des tablespaces
- La date et l'heure de création de la base de données.
- Le numéro de séquence de log actuel. Cette information permet de synchroniser tous les fichiers appartenant à la base de données.
- Les informations sur les points de synchronisation. Cette information est utilisée par le serveur Oracle lors de la restauration de l'instance.
- L'historique des fichiers log
- Les informations de sauvegarde de l'utilitaire Recovery Manager.
- Le début et à la fin des segments de undo.

Le fichier de contrôle se divise en deux parties :

- Réutilisable, qui stocke les informations du Recovery Manager, comme les fichiers de sauvegarde de redo log et les fichiers de sauvegarde de données. Ces fichiers ne sont réutilisables que par le Recovery Manager.
- Non réutilisable.

3.1.2. Multiplexage des fichiers de contrôle

Avec le SPFILE :

Dans le but de prévenir une erreur dans un fichier de contrôle, il est fortement recommandé de multiplexer les fichiers de contrôles et de les stocker séparément sur des disques différents. Si un fichier de contrôle est perdu, une copie du fichier de contrôle peut être utilisé pour redémarrer l'instance. On peut multiplexer jusqu'à 8 fichiers de contrôles. Ci-après les étapes de multiplexages pour un fichier de contrôle :

- Modifier le SPFILE :

```
ALTER SYSTEM SET control files =  
'$HOME/ORADATA/u01/ctrl01.ctl',  
'$HOME/ORADATA/u02/ctrl02.ctl' SCOPE = SPFILE ;
```

- Eteindre la base de données.
- Créer les fichiers de contrôles supplémentaires :

```
cp $HOME/ORADATA/u01/ctrl01.ctl  
$HOME/ORADATA/u02/ctrl02.ctl
```

- Redémarrer la base de données.

Avec le init.ora :

Les étapes sont les suivantes :

- Eteindre la base de données.
- Copier le fichier de contrôle existant sur un autre disque et changer son nom :

```
cp control01.ctl ../DISK3/control02.ctl
```

- Ajouter le nouveau fichier de contrôle à init.ora :

```
CONTROL_FILES = (/DISK1/control01.ctl,  
/DISK3/control02.ctl)
```

- Redémarrer la base de données.

Il est possible de créer une sauvegarde d'un fichier de contrôle, mais il n'est pas possible de récupérer un fichier de contrôle à partir du fichier de backup sans les fichiers de données appropriés.

```
ALTER DATABASE BACKUP CONTROLFILE TO 'FILENAME'
```

3.1.3. Gérer les fichiers de contrôle avec OMF

Les fichiers de contrôles sont reconnus comme OMF pendant la création de la base de données si le paramètre `CONTROL_FILES` n'est pas spécifié dans le fichier d'initialisation. Si on utilise un fichier `init.ora`, le paramètre `CONTROL_FILES` doit être défini au nom généré par OMF, qui peut être trouvé en interrogeant la vue `V$CONTROLFILE` ou le fichier `alert<SID>.log`. Si l'on utilise un `SPFILE`, le paramètre `CONTROL_FILES` est automatiquement défini et généré à la création de la base de données. Le fichier de contrôle généré s'appelle `ora_cmr7t30p.ctl` et contenu dans le fichier `alert<SID>.log`.

Il est toujours possible de créer un nouveau fichier de contrôle en utilisant la commande `CREATE CONTROLFILE`. Les noms des OMF doivent être utilisés dans les clauses `DATAFILE` et `LOGFILE`.

3.1.4. Récupérer les informations des fichiers de contrôle

Pour récupérer l'emplacement et les noms des fichiers de contrôles, il est nécessaire d'interroger la vue dynamique `V$CONTROLFILE`.

```
SELECT name FROM V$CONTROLFILE ;

NAME
-----
D:\ORACLE\ORADATA\IASDB\CONTROL01.CTL
D:\ORACLE\ORADATA\IASDB\CONTROL02.CTL
D:\ORACLE\ORADATA\IASDE\CONTROL03.CTL

2 rows selected.
```

D'autres vues peuvent être utilisées comme `V$PARAMETER` qui contient toutes les informations sur tous les paramètres, `V$CONTROLFILE_RECORD_SECTION` qui contient les informations sur les sections d'enregistrement des fichiers de contrôles, et la commande `SHOW PARAMETERS CONTROL_FILES` qui liste le nom, l'état et l'emplacement des fichiers de contrôles.

3.2. Les fichiers de redo log

3.2.1. Utiliser les fichiers de redo log

Comme il a été vu dans le module précédent, les fichiers de redo log permettent d'enregistrer tous les changements effectués par des transactions sur les données et fournissent un mécanisme de récupération des données en cas de dysfonctionnement du système ou d'un disque. Les fichiers redo log stockent toutes les modifications apportées aux données du buffer cache. L'organisation des fichiers de redo log s'effectue en groupe (au minimum 2) contenant un ou plusieurs fichiers de redo log. Chaque fichier de redo log appartenant à un groupe est appelé un membre.

Tous les membres redo log online présentent tous des numéros de séquence log identiques. Le numéro de séquence log est affecté chaque fois que le serveur Oracle commence à écrire dans un groupe de fichiers redo log online. Ces numéros sont utilisés pour identifier de façon unique chaque membre redo log online.

Le numéro de séquence log courant est stocké dans le fichier de contrôle ainsi que dans l'entête de tous les fichiers de données. Ce numéro est utilisé lors de la récupération de la base de données. Les membres redo log online ont tous la même taille.

Un groupe de redo log en ligne sert également à restaurer des données validées qui n'ont pas été écrites dans les fichiers de données. Une telle situation peut se produire après l'échec d'une instance.

3.2.2. Structure des fichiers de redo log

La charge de maintenir et gérer plusieurs fichiers de redo log appartient au DBA, pour éviter qu'un seul dysfonctionnement d'un fichier fasse perdre des informations à la base de données.

Les mêmes informations sur les modifications apportées aux données sont enregistrées dans tous fichiers redo log online d'un groupe de redo log online. Donc tous les fichiers redo log d'un groupe redo log en ligne possèdent exactement les mêmes informations. Ce travail est effectué pas le processus d'arrière plan LGWR.

3.2.3. Paramètres des fichiers de redo log

Les performances d'une base de données sont affectées par le nombre et la taille des fichiers et des groupes redo log online.

Le numéro initial et la taille des fichiers redo log en ligne et des groupes sont déterminés pas des paramètres d'initialisation :

- MAXLOGFILES spécifie le nombre maximal de groupes de fichiers redo log online. La limite de MAXLOGFILES est de 255.
- MAXLOGMEMBERS détermine le nombre maximal de membres redo log online pour chaque groupe de redo log online. Le nombre total de membres redo log online est un multiple du nombre de fichiers redo log.
- LOG_FILES spécifie le nombre maximal actuel de groupes de fichiers redo log online pour la durée de vie de l'instance courante. Par défaut, la valeur de LOG_FILES est plus petite ou égale au nombre total de membres redo log en ligne (MAXLOGMEMBERS * MAXLOGFILES).

Les fichiers redo log en ligne sont créés durant la création de la base de données.

Le serveur Oracle enregistre séquentiellement tous les changements fait sur la base de données dans le buffer de redo log. Ensuite, les données contenues dans le redo buffer sont écrites grâce au process LGWR dans le fichier de redo log online. LGWR déclenche le processus d'écriture lors d'un des évènements suivants :

- Une transaction est comité.
- Quand le buffer de redo log est plein au tiers.
- Quand le buffer de redo log contient plus de 1Mo d'enregistrements changés.
- Avant que le process DBWn écrivent les blocs modifiés du database buffer cache dans les fichiers de données.

3.3. Gérer les groupes de redo

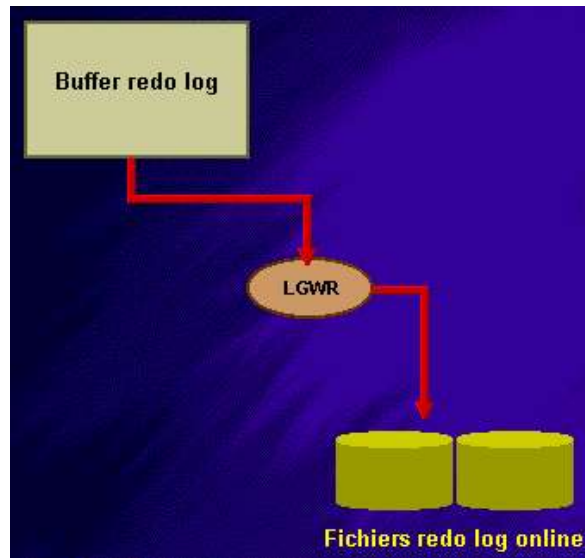
3.3.1. Fonctionnement des fichiers de redo log

Basculement de fichiers log (Log Switches) :

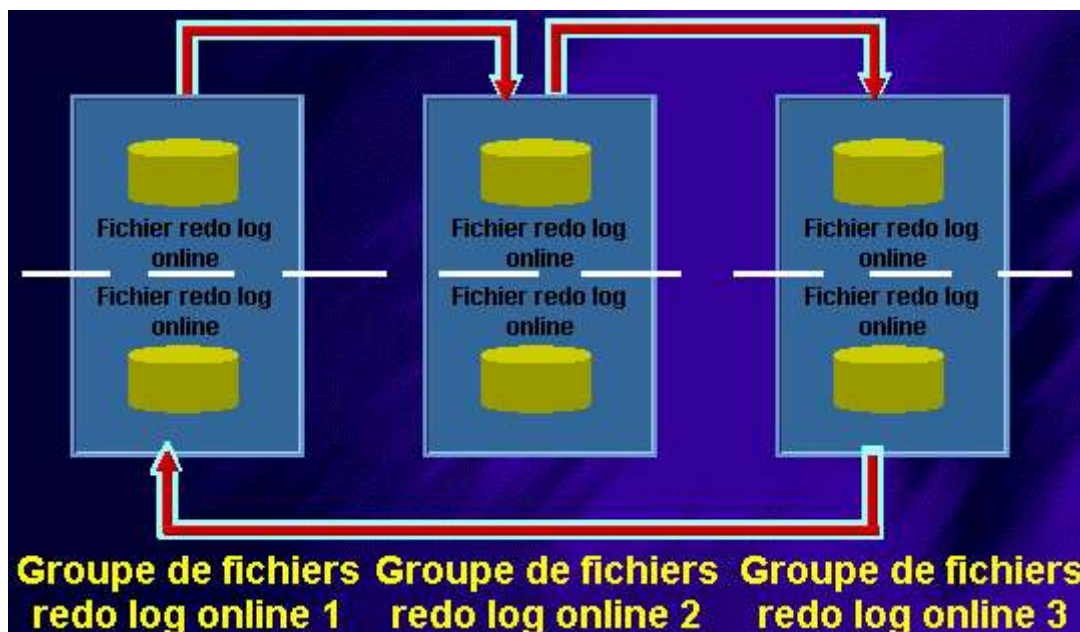
Toutes les modifications apportées à la base de données sont stockées séquentiellement dans le buffer redo log avant le remplissage des groupes de fichiers redo log online.

Le serveur Oracle écrit ensuite le contenu du redo log buffer dans les groupes de redo log online. Le redo log buffer est utilisé de façon circulaire.

Dans certains cas, les entrées du redo log buffer sont écrites dans un des groupes de redo log online par le processus d'arrière plan LGWR. Ce groupe est appelé le groupe courant de redo log online.



Le LGWR écrit séquentiellement dans les fichiers redo log online. Il commence à écrire dans le groupe suivant une fois que le groupe courant est plein. Lorsque le dernier fichier redo log online disponible est plein, le LGWR revient au premier groupe et écrase son contenu avec les nouvelles entrées.



L'événement durant lequel LGWR arrête d'écrire dans un groupe et commence à écrire dans un autre, est appelé basculement de fichier log ("log switch"). L'événement basculement de fichier log amorce un événement point de synchronisation("checkpoint"), qui entraîne l'écriture des buffers modifiés ("dirty blocks") du buffer cache dans les fichiers de données.

Un basculement de fichier log apparaît lorsque :

- le groupe courant de fichiers redo log online est plein

- l'administrateur force le basculement de fichier log.

Ces situations provoquent les événements qui permettent au serveur Oracle de conserver les données les plus récentes dans les fichiers de données.

Chaque fois qu'un basculement de fichier log se produit et que le LGWR commence l'écriture dans un nouveau groupe de fichiers redo log online, le serveur Oracle assigne un numéro appelé le numéro de séquence log. Ce numéro identifie l'ensemble des entrées redo log.

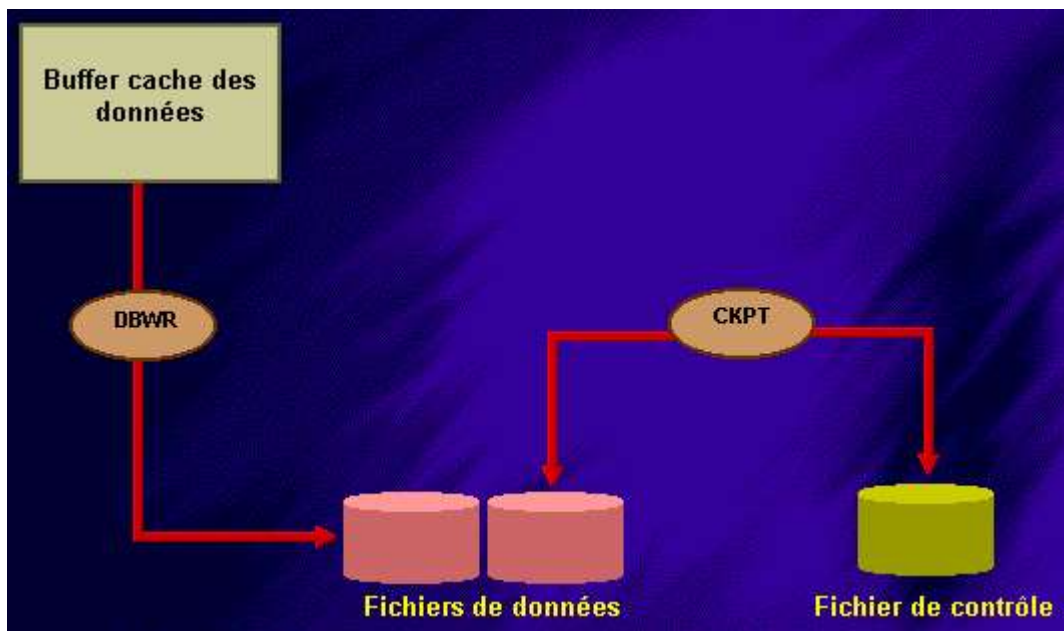
Points de synchronisation (Checkpoints) :

L'événement d'arrière plan du serveur Oracle qui met à jour les fichiers de données est le point de synchronisation.

Les points de synchronisation peuvent concerner tous les fichiers de données de la base de données ou seulement certains fichiers de données spécifiques.

Lors d'un point de synchronisation, tous les buffers modifiés ("dirty blocks") sont écrits dans les fichiers de données par le processus DBWR. Le nombre de buffers qui sont écrit par le process DBWR est déterminé par le paramètre FAST_START_MTTR_TARGET.

Si le point de synchronisation s'est achevé correctement, le processus d'arrière-plan CKPT met à jour les entêtes de tous les fichiers de données et fichiers de contrôle.



Un point de synchronisation se produit dans plusieurs situations :

- durant un basculement de fichier log.
- Lorsqu'une instance s'arrête en mode normal, transactional ou immediate
- Selon la configuration des paramètres d'initialisation LOG_CHECKPOINT_INTERVAL et LOG_CHECKPOINT_TIMEOUT
- lorsqu'il est imposé par l'administrateur
- lorsqu'un tablespace devient offline ou en lecture seule

Ces situations entraînent l'événement checkpoint. Ainsi les fichiers de données dans la base de données sont mis à jour à intervalles réguliers.

Les informations à propos de chaque checkpoint sont enregistrées dans le fichier alert<SID>.log si le paramètre LOG_CHECKPOINTS_TO_ALERT est initialisé à TRUE. Sa valeur par défaut est à FALSE.

3.3.2. Forcer un basculement de fichiers log et un point de synchronisation

Log switch :

Comme vu dans le point précédent, le basculement de fichiers log et les checkpoints sont fait automatiquement. Pourtant en cas de nécessité le DBA peut décider de forcer l'exécution d'un log switch ou d'un checkpoint.

Un log switch peut être forcé avec la commande SQL suivante :

```
ALTER SYSTEM SWITCH LOGFILE ;
```

Outre la commande SQL, Oracle offre une interface graphique appelée Oracle Backup Manager permettant d'imposer un basculement de fichier log.

Un basculement de fichier log peut être forcé même si aucune sauvegarde n'est prévue dans l'immédiat.

Point de synchronisation :

L'administrateur procède souvent à des sauvegardes permanentes (dite "à chaud") de tablespaces. Pour s'assurer que les fichiers de données contiennent les données les plus récentes, l'administrateur doit forcer un point de synchronisation avant de procéder à une sauvegarde.

La syntaxe suivante permet de forcer un point de synchronisation:

```
ALTER SYSTEM CHECKPOINT;
```

Outre la commande SQL, Oracle offre une interface graphique, Oracle Backup Manager qui permet de forcer un point de synchronisation.

Il est possible de forcer un checkpoint en utilisant le paramètre FAST_START_MTTR_TARGET. Par exemple FAST_START_MTTR_TARGET = 600 signifie que la restauration d'une instance après un crash ne doit pas prendre plus de 600 secondes, et la base de données doit ajuster les autres paramètres pour arriver à ce résultat.

3.3.3. Ajout de groupe de fichiers redo log online

Si la base de données est en mode ARCHIVELOG et que le fichier redo log online est gros, l'archivage peut prendre beaucoup de temps. Donc, un fichier redo log online peut ne pas être disponible pour les écritures effectuées par le buffer redo log. Dans ce cas, le processus LGWR est contraint d'attendre. Si le processus LGWR doit attendre, le DBWR attend et le système ralentit. Pour éviter les attentes et les ralentissements, l'ajout de groupes redo log en ligne peut être nécessaire.

Il existe deux méthodes pour ajouter un groupe redo log en ligne : avec une commande SQL ou avec l'outil Oracle Storage Manager.

Des groupes de fichiers redo log en ligne peuvent être créés à l'aide de la commande ALTER DATABASE :

```
ALTER DATABASE [database]
```

```
ADD LOGFILE [GROUP integer] 'filename' [SIZE n[K|M]] [REUSE]
[, [GROUP integer] 'filename' [SIZE n[K|M]] [REUSE] ]...;
```

La valeur du paramètre GROUP peut être choisie pour chaque groupe de fichier redo log. Si ce paramètre est omis, Oracle lui génère automatiquement une valeur.

Exemple :

```
SQL> ALTER DATABASE
2 ADD LOGFILE GROUP 3
3 ('c:\orant\database\logorc13.ora') SIZE 1000K;

Database altered.
```

Cette commande crée un nouveau groupe de fichiers redo log portant le numéro 3. Ce groupe est constitué d'un membre redo log nommé logorc13.ora d'une taille de 1000 Ko.

3.3.4. Ajout des membres redo log online

Des membres redo log en ligne peuvent être ajoutés à un groupe afin d'éviter des défaillances isolées. En effet, l'ajout de membres redo log permet de placer les fichiers redo log en miroir.

Des membres redo log peuvent être ajoutés grâce à la commande SQL suivante :

```
ALTER DATABASE [database]
ADD LOGFILE MEMBER 'filename' [REUSE]
TO GROUP n;
```

Le chemin complet des membres doit être spécifié. Si le chemin complet n'est pas spécifié, les fichiers seront créés dans le répertoire par défaut du serveur de base de données.

Si le fichier existe déjà, il doit avoir la même taille que le nouveau fichier redo log online et l'option de la commande REUSE doit être utilisée. Le groupe cible peut être identifié en spécifiant le numéro du groupe soit un ou plusieurs membres du groupe.

Exemple :

```
SQL> ALTER DATABASE
2 ADD LOGFILE MEMBER 'e:\orant\database\log7borc1.ora'
3 TO GROUP 7;

Database altered.
```

Cette commande ajoute un nouveau membre redo log nommé log7borc1.ora au groupe redo log 7. Ce nouveau membre sera stocké dans le répertoire e:\orant\database.

3.3.5. Relocalisation des fichiers redo log online

En raison de contraintes système, telles que l'insuffisance d'espace disque, les fichiers redo log online peuvent avoir besoin d'être relocalisés.

Étapes à suivre pour relocaliser un fichier redo log online :

- Arrêter la base de données,
- Copier le fichier redo log online vers la nouvelle destination,
- Monter la base de données,

- Changer le nom du fichier redo log online à l'aide d'une commande SQL :

```
ALTER DATABASE [database]  
RENAME FILE 'old filename' TO 'new filename';
```

Avant de changer le nom du fichier, il faut s'assurer que le nouveau fichier existe bien à l'emplacement prévu. Le serveur Oracle change seulement le pointeur dans le fichier de contrôle, mais il ne renomme pas physiquement ni ne crée de fichier.

- Ouvrir la base de données.

3.3.6. Suppression de groupes de fichiers redo log online

Pour améliorer les performances de la base de données, il peut s'avérer nécessaire d'augmenter ou de diminuer la taille des groupes de fichiers redo log online. Pour changer la taille d'un groupe de fichiers redo log online, il faut créer un nouveau groupe de fichiers redo log online et ensuite supprimer le vieux groupe de fichiers redo log online.

Pour supprimer un groupe de fichiers redo log online, il faut utiliser la commande SQL suivante :

```
ALTER DATABASE [database]  
DROP LOGFILE GROUP n;
```

Cependant un certain nombre de restrictions sont à prendre en compte lors de la suppression de groupes redo log online :

- L'instance doit avoir au moins deux groupes de fichiers redo log online. Un groupe de fichiers redo log online ne pourra pas être supprimé si il n'existe que deux groupes. Pour supprimer un groupe, il doit en rester au moins trois.
- Un groupe redo log online actif ne peut pas être supprimé.
- Si la base de données tourne en mode ARCHIVELOG, un groupe de fichiers redo log online non archivé ne peut pas être supprimé.

Quand un groupe redo log est supprimé, les fichiers du système d'exploitation ne sont pas supprimés automatiquement. Il est donc nécessaire de supprimer manuellement les fichiers redo log online afin de garder un espace disque propre.

3.3.7. Suppression des membres redo log online

Pour obtenir les meilleures performances d'une base de données Oracle, il est nécessaire de vérifier régulièrement le statut des fichiers redo log online. Parfois, un fichier redo log online peut devenir invalide à cause d'événements tels qu'une défaillance du support. Ce qui rend les fichiers inaccessibles. Dans ce cas, il faut supprimer ces fichiers redo log online.

La commande SQL qui permet de supprimer un membre redo log en ligne est la suivante :

```
ALTER DATABASE [database]  
DROP LOGFILE MEMBER 'filename';
```

Exemple :

Le fichier redo log online nommé log5borcl.ora dans le groupe 5 est devenu invalide. Ce fichier est stocké dans le répertoire e:\orant\oradata\orcl.

```
SQL> ALTER DATABASE  
2 DROP LOGFILE MEMBER 'e:\orant\oradata\orcl\log5borcl.ora';
```

```
Database altered.
```

Cette commande supprime le fichier redo log online invalide log5borcl.ora.

Cependant un certain nombre de restrictions sont à prendre en compte lors de la suppression de membres redo log en ligne :

- Le dernier membre redo log online valide d'un groupe ne peut pas être supprimé.
- Un basculement de fichier log doit être effectué avant de supprimer un membre redo log online actif.
- Si la base de données tourne en mode ARCHIVELOG, alors un membre redo log online appartenant à un groupe non archivé ne peut pas être supprimé

3.3.8. Réinitialisation des fichiers redo log online

Lors de l'utilisation de la base de données, il est possible que tous les membres d'un groupe de fichiers redo log online soient corrompus. Pour faire face à ce problème, il faut réinitialiser les fichiers redo log online.

Les commandes SQL pour réinitialiser les fichiers redo log en ligne sont les suivantes :

```
ALTER DATABASE [database]  
CLEAR [UNARCHIVED] LOGFILE GROUP n;
```

Ou

```
ALTER DATABASE [database]  
CLEAR [UNARCHIVED] LOGFILE 'filename';
```

L'utilisation de ces commandes revient à ajouter et supprimer un fichier redo log online. Cependant ces commandes peuvent être effectuées même si il y seulement deux groupes de fichiers redo log online avec un membre chacun. Ces commandes peuvent également être effectuées même si un groupe de fichiers redo log online réinitialisé est disponible mais pas archivé.

Exemple :

Le fichier redo log online redo04.log dans le groupe 1 est corrompu.

```
SQL> ALTER DATABASE  
2 CLEAR LOGFILE 'e:\orant\oradata\orcl\redo04.log';  
  
Database altered.
```

Le fichier redo log online redo04.log a été réinitialisé. Ce fichier est désormais prêt et disponible pour utilisation.

Un fichier redo log online peut être réinitialisé qu'il ait été archivé ou non. Si le fichier à réinitialiser n'a pas été archivé, il faut inclure le mot clé UNARCHIVED dans la commande SQL. Cependant, l'utilisation du mot clé UNARCHIVED aboutit à des sauvegardes inutilisables si le fichier redo log en ligne est nécessaire pour la récupération.

3.4. Planification des fichiers redo log online

3.4.1. Nombre de fichiers redo log online

Pour minimiser la perte de données causée par une défaillance, le serveur Oracle fournit des groupes de fichiers redo log online.

Cependant le nombre et la taille des groupes de fichiers redo log online qui peuvent être créés dépendent d'un certain nombre de facteurs :

- le nombre de transactions effectuées sur la base de données.

Quand le nombre de transaction est limité, une instance de base de données ne peut avoir besoin que de deux groupes. Quand le nombre de transactions est très grand, une instance de base de données peut avoir besoin de groupes additionnels pour garantir la disponibilité des groupes de fichiers redo log online au processus LGWR.

- La taille du buffer redo log.

Si des messages dans le fichier trace LGWR ou dans le fichier ALERT indiquent que le processus LGWR attend fréquemment pour écrire dans un groupe de fichiers redo log online, il faut ajouter des groupes de fichiers redo log online. Le processus LGWR peut attendre à cause d'un point de synchronisation pas terminé ou d'un groupe de fichiers redo log online pas encore archivé.

- La stabilité des gros fichiers du système d'exploitation.

Le système d'exploitation peut devenir instable en traitant de gros fichiers, il est donc préférable de maintenir des fichiers de petite taille, donc d'augmenter le nombre de fichiers redo log online.

Bien que le nombre de membres dans des groupes multiplexés puisse être différent, il est très conseillé de mettre en place une configuration symétrique.

Une configuration asymétrique peut être utilisée seulement temporairement dans une situation inutilisable telle qu'une défaillance du support.

3.4.2.Emplacement des fichiers redo log online

La disponibilité des fichiers redo log online détermine la performance d'une base de données Oracle. Pour améliorer la performance de la base de données, les membres des groupes de fichiers redo log online, les fichiers log archivés et les fichiers de données doivent être stockés sur des disques séparés. L'avantage de stocker les fichiers redo log online sur des disques séparés est que l'instance ne s'arrêtera pas même si un des membres n'est pas disponible. Un autre avantage de stocker les fichiers redo log online et les fichiers log archivés sur des disques différents est que la contention entre les processus ARCH et LGWR est réduite.

De plus, les fichiers de données placés sur différents disques permettent également de réduire la contention entre les processus LGWR et DBWR.

Le stockage des fichiers redo log et des fichiers de données sur différents disques permet de réduire de façon substantielle le risque de perdre à la fois les fichiers de données et les fichiers redo log online lors d'une défaillance du support.

3.4.3.Dimensionnement des fichiers redo log online

Les performances d'une base de données Oracle dépendent de la disponibilité des fichiers redo log online. Donc, la taille des fichiers redo log online est une décision importante.

La taille minimale d'un fichier redo log en ligne est de 50 Ko, et la taille maximale est spécifique au système d'exploitation.

Cependant, il existe des situations qui influencent la taille des fichiers redo log online :

- Le nombre de basculements de fichiers log et de points de synchronisation.

Si les fichiers redo log online sont de petite taille, un grand nombre de basculements de fichiers log aura lieu ce qui réduit l'efficacité de la base de données.

- Le nombre et la quantité d'entrées redo.

Si le nombre d'entrées redo est élevé, les fichiers redo log online seront remplis rapidement ce qui débouche sur la génération d'un grand nombre de basculement de fichiers log.

- L'espace disponible sur le support de stockage.

Si le support de stockage ne présente pas la capacité suffisante pour stocker le fichier redo log online en une seule entité, l'administrateur de la base de données est contraint d'utiliser un fichier redo log online plus petit ou plusieurs fichiers. Ces fichiers sont stockés sur plusieurs disques.

Les membres des groupes différents peuvent présenter des tailles différentes. Cependant, il n'est pas souhaitable d'avoir des groupes de tailles différentes.

Les groupes de tailles différentes ne sont nécessaires que provisoirement lors de la modification de la taille des membres des groupes de fichiers redo log online.

3.4.4. Gérer les redo log online with OMF

Pour créer un nouveau groupe de fichiers de redo log sans spécifications, le DBA utilise la commande :

```
ALTER DATABASE ADD LOGFILE ;
```

Cette commande ajoute un fichier de log avec un membre dans DB_CREATE_ONLINE_LOG_DEST_1 et un membre dans DB_CREATE_ONLINE_LOG_DEST_2. Les noms des fichiers uniques des fichiers de redo log sont générés automatiquement.

Il est bien évidemment possible d'utiliser la clause GROUP pour supprimer un fichier de log. Dans l'exemple précédent, le fichier du système d'exploitation associé à chaque fichier de log membre OMF est automatiquement supprimé.

Les fichiers archivés de redo log ne peuvent être des fichiers OMF. Un emplacement système pour les fichiers de log archivés peut être spécifié avec le paramètre d'initialisation LOG_ARCHIVE_DEST_n

3.4.5. Obtenir des informations sur les groupes

Il est possible d'obtenir des informations sur les groupes et les membres en interrogeant le dictionnaire de données.

La requête suivante retourne des informations sur le fichier de redo log online du fichier de contrôle.

```
SELECT group#, sequence#, bytes, members, status  
FROM V$LOG ;
```

GROUP#	SEQUENCE#	BYTES	MEMBERS	STATUS
1	1985	1048576	1	INACTIVE
2	1986	1048576	1	CURRENT
3	1984	1048576	1	INACTIVE

INACTIVE signifie que le groupe de fichier de redo n'est plus nécessaire pour la restauration d'instance.

CURRENT, indique le groupe courant de redo log. C'est ce groupe là qui est actif.

UNUSED, signifie que le groupe de redo vient juste d'être ajouté et qu'il n'a jamais été écrit.

Pour connaître tous les noms des membres d'un groupe, il faut interroger la vue V\$LOGFILE.

```
SELECT member FROM V$LOGFILE ;
```

```
MEMBER  
-----  
D:\ORACLE\ORADATA\IASDB\REDO03.LOG  
D:\ORACLE\ORADATA\IASDB\REDO02.LOG  
D:\ORACLE\ORADATA\IASDB\REDO01.LOG
```

La valeur du statut de la colonne peut être INVALID (fichier inaccessible), STALE (le contenu du fichier est incomplet), DELETED (le fichier n'est plus utilisé).

3.4.6. Archivage de fichiers de redo log

Une des décisions importantes qu'un DBA doit prendre est configurer la base de données dans le mode ARCHIVELOG ou NOARCHIVELOG.

NOARCHIVELOG :

En mode, NOARCHIVELOG, les fichiers de redo log online sont réécrits à chaque fois qu'un fichier de redo log online est rempli et qu'un log switch est lancé. LGWR ne réécrit pas sur un groupe de fichier de redo log tant que le checkpoint pour ce groupe n'est pas terminé.

ARCHIVELOG :

Si la base de données est configurée dans le mode ARCHIVELOG, alors les groupes de redo remplis doivent être archivés. Comme toutes les modifications faites sur la base de données sont enregistrées dans les fichiers de redo log online, le DBA peut utiliser la sauvegarde présente sur le disque dur et les fichiers de redo log archivés pour restaurer la base de données sans perdre aucune donnée comité.

On peut archiver les fichiers de redo log :

- Manuellement.
- Automatiquement : Méthode recommandée.

Le paramètre d'initialisation LOG_ARCHIVE_START, lorsqu'il est à TRUE indique que l'archivage se fait automatiquement. A FALSE, indique que le DBA le fait manuellement. En mode automatique, l'archivage se fait grâce au process ARCn, et manuellement avec des requêtes SQL.

4. Gérer les Tablespaces et les fichiers de données

4.1. Architecture d'une base de données

4.1.1. Vue d'ensemble de la structure d'une base de données

Les performances d'une base de données dépendent en grande partie de la gestion efficace des ressources du système, telles que l'espace du disque dur. Afin d'obtenir les performances optimales pour la base de données, Oracle facilite un contrôle poussé de la gestion de l'espace disque en divisant la base de données en structures logiques et physiques.

Les structures logiques et physiques qui constituent la base de données Oracle sont elles-mêmes constituées d'autres composants.

La structure physique est constituée de fichiers de contrôle, de fichiers redo log online et de fichiers de données.

Les composants de la structure logique d'une base de données sont les tablespaces, les segments, les extents et les blocs de données.

Lorsque les composants de la structure logique sont créés par Oracle, l'espace est alloué dans la base de données en fonction de paramètres prédéfinis. L'administrateur de la base de données peut remplacer ou configurer ces paramètres afin d'optimiser l'utilisation de l'espace.

4.1.2. Relations

La séparation des structures logique et physique d'une base de données Oracle facilite le contrôle poussé de la gestion de l'espace disque. L'administrateur peut configurer les paramètres d'allocation d'espace aux composants physiques et logiques de la base de données.

Pour utiliser efficacement l'espace du disque dur, il est important de connaître les relations entre les composants physiques et logiques de la base de données. Il est important également de savoir comment l'espace est alloué dans la base de données.

La première entité est la base de données Oracle. Elle est divisée en structures logique et physique.

La base de données est divisée en zones d'espace logiques plus petites, appelées tablespaces.

Chaque tablespace d'une base de données Oracle est constitué d'un ou plusieurs fichiers appelés fichiers de données. Il s'agit de structures physiques conformes au système d'exploitation sur lequel le serveur Oracle fonctionne.

Un tablespace est constitué de segments. Un segment est l'espace alloué pour un type spécifique de structure de stockage logique dans un tablespace. Les segments d'index, segments temporaires, rollback segments et segments de données représentent quelques exemples de segments.

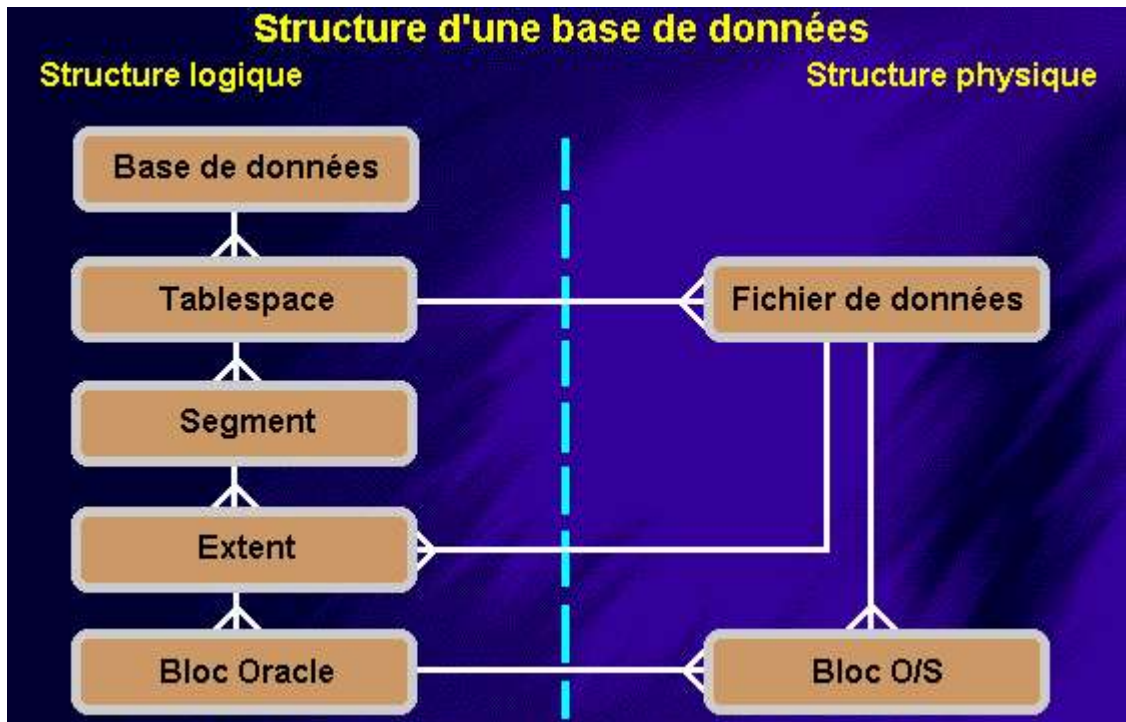
Un segment, tel qu'un segment de données, peut être réparti sur plusieurs fichiers appartenant au même tablespace.

Le niveau suivant de la structure logique d'une base de données est l'extent. Un extent est un ensemble de blocs contigus. Chaque segment est constitué d'un ou plusieurs extents. Le DBA peut ajouter des extents à un segment manuellement.

Un extent ne peut pas être réparti sur plusieurs fichiers de données.

Les blocs de données constituent le dernier niveau de granularité. Les données d'une base de données Oracle sont stockées dans les blocs de données.

Un bloc de données correspond à un ou plusieurs blocs de fichiers physiques alloués à partir de fichier de données existant.



4.1.3. Caractéristiques d'un tablespace

La structure de stockage logique la plus importante d'une base de données est le tablespace.

Caractéristiques d'un tablespace :

- Un tablespace ne peut appartenir qu'à une seule base de données
- Un tablespace est constitué d'un ou plusieurs fichiers du système d'exploitation.
- Les tablespaces peuvent être mis offline lorsque la base de données fonctionne. Cette fonctionnalité permet à l'administrateur de contrôler les données sans arrêter la base de données.
- Tous les tablespaces, à l'exception du tablespace SYSTEM ou d'un tablespace avec un undo segment actif, peuvent être mis offline. Dans ce cas, la base de données continue de fonctionner. L'administrateur peut ainsi assurer la gestion des données sans que la base de données ne soit indisponible.
- L'état d'un tablespace peut être basculé entre lecture écriture et lecture seule. Cette fonctionnalité permet d'empêcher les utilisateurs de modifier des données stockées et inversement.

4.1.4. Utilisations d'un tablespace

Le tablespace présente plusieurs utilisations possibles permettant la gestion efficace de l'espace disque.

Utilisations d'un tablespace :

- Aider les serveurs Oracle pour le contrôle de l'allocation de l'espace et l'attribution de quotas d'espace aux utilisateurs.
- Contrôler la disponibilité des données. Ce contrôle est effectué par la mise online ou offline des tablespaces individuels.

- Améliorer les performances E/S et réduire les contentions E/S grâce à la répartition du stockage des données sur plusieurs périphériques.
- Utiliser lors des opérations de sauvegardes et de restaurations partielles.
- Utiliser pour stocker de grandes quantités de données statiques sur des périphériques en lecture seule.

4.1.5. Caractéristiques d'un fichier de données

Un tablespace est la structure de stockage logique la plus importante d'une base de données. Les données du tablespace sont stockées dans les structures de stockage physiques appelées fichiers de données.

Chaque tablespace d'une base de données Oracle est constitué d'un ou plusieurs fichiers de données. Les fichiers de données sont les structures physiques conformes au système d'exploitation sur lequel le serveur Oracle est installé.

Caractéristiques d'un fichier de données :

- Un fichier de données ne peut appartenir qu'à un seul tablespace. Ceci permet à un administrateur de base de données de procéder à une sauvegarde du fichier de données sans affecter la disponibilité de tout autre tablespace.
- Un fichier de données est créé par le serveur Oracle. Cependant, la quantité disque à occuper par le fichier de données est spécifiée par l'administrateur de la base de données. Le serveur Oracle crée le fichier de données avec la quantité spécifiée d'espace disque plus un léger dépassement.
- Un fichier de données peut être modifié par l'administrateur de la base de données après sa création. L'administrateur peut également spécifier que la taille d'un fichier de données doit augmenter de façon dynamique, tout comme le nombre d'objets du tablespace. Cette caractéristique permet à l'administrateur de la base de données de contourner la limitation du paramètre MAXDATAFILES et permet à la base de données d'être constituée de moins de fichiers de données pour chaque tablespace. Elle évite également que les utilisateurs et les applications ne subissent des erreurs dues au manque d'espace libre dans un tablespace.

4.2. Créer des Tablespaces

4.2.1. Types de tablespaces

Lorsque l'administrateur crée une base de données, un tablespace appelé SYSTEM est créé par défaut. Tous les autres tablespaces, appelés tablespaces non-SYSTEM, sont créés par l'administrateur de la base de données.

L'administrateur crée un tablespace non-SYSTEM pour gérer facilement la base de données et permettre aux utilisateurs de l'utiliser de façon efficace.

Le tablespace SYSTEM est indispensable pour le bon fonctionnement de toute base de données. Le tablespace non-SYSTEM n'est pas obligatoire.

Le tablespace SYSTEM contient les informations du dictionnaire de données, les définitions des procédures stockées, des packages et des triggers base de données, les undo segments SYSTEM. D'autre part, les tablespaces non-SYSTEM peuvent contenir les segments de données, les segments d'index, les segments temporaires et les undo segments. Ils séparent aussi les données dynamiques et statiques

Le tablespace SYSTEM contient le rollback segment SYSTEM, tandis qu'un tablespace non-SYSTEM peut contenir tout autre rollback segment.

Il n'est pas souhaitable de stocker les données utilisateur dans le tablespace SYSTEM. Si le tablespace SYSTEM contient des données utilisateur, vous ne pouvez pas procéder à des sauvegardes offline des données sans arrêter la base de données.

4.2.2.Création d'un tablespace

L'administrateur est responsable de la gestion des données des bases de données. Un moyen efficace de gestion des données consiste à créer des tablespaces pour différents groupes d'utilisateurs, en créant, par exemple, un tablespace marketing pour le personnel du service marketing.

Syntaxe de création d'un tablespace :

```
CREATE TABLESPACE tablespace
[DATAFILE clause]
[MINIMUM EXTENT integer[K|M]]
[BLOCKSIZE integer [K]]
[LOGGING|NOLOGGING]
[DEFAULT storage_clause ]
[ONLINE|OFFLINE]
[PERMANENT|TEMPORARY]
```

- DATAFILE spécifie le ou les fichiers de données liés au tablespace. M spécifie la taille en Mo ou Kbits.
- LOGGING option par défaut qui spécifie que toutes les tables, indexes et partitions dans le tablespace génèrent des infos de redo log.
- DEFAULT définit les paramètres de stockage par défaut pour tous les objets créés dans le tablespace.
- OFFLINE rend le tablespace indisponible immédiatement après la création.
- PERMANENT spécifie que le tablespace peut être utilisé pour contenir des objets permanents.
- TEMPORARY cette clause est utilisée pour les objets temporaires.

```
storage_clause := (
                [INITIAL   integer [K|M]]
                [NEXT      integer [K|M]]
                [MINEXTENTS integer ]
                [MAXEXTENTS {integer | UNLIMITED}]
                [PCTINCREASE integer ])
```

Exemple :

Créez un tablespace nommé ACCOUNTS.
Le nom du fichier de données est accts01.dbf et sa taille est de 10 Mo.
Nom du tablespace : ACCOUNTS
Nom du fichier de données : accts01.dbf
Taille du fichier de données : 10 Mo
Emplacement : c:\oracle\oradata\database

Se connecter à la base de données avec les privilèges adéquates :

```
SQL> CREATE TABLESPACE accounts
      2  DATAFILE 'c:\oracle\oradata\database\accts01.dbf' SIZE 10M;
Tablespace created.
```

Cette commande est relativement complexe, Oracle offre une interface graphique, appelée Oracle Storage Manager, pour créer un tablespace.

Lors de la création de tablespaces, le nombre maximal de tablespaces pouvant être affectés à chaque base de données est de 64 000 et le nombre maximal de fichiers de données pouvant être affectés à chaque tablespace est de 1 023.

4.2.3. Gestion de l'espace dans les tablespaces

Les tablespaces peuvent être gérés avec les tables du dictionnaire de données ou des tables binaires d'allocations (bitmaps). L'une de ces méthodes doit être choisie lorsque l'on crée un tablespace et ne pourra être changée par la suite.

Locally Managed Tablespaces

Un tablespace qui gère ses propres extents garde un bitmap dans chaque fichier de données pour garder une trace des blocs libres ou utilisés dans ces fichiers de données. Chaque bitmap correspond à un bloc ou un groupe de blocs. Lorsqu'un extent est alloué ou prêt à être réutilisé, le serveur Oracle change la valeur du bitmap pour mettre à jour le nouveau statut des blocs.

Dictionary-Managed Tablespaces

Pour un tablespace qui utilise le dictionnaire de données pour gérer les extents, le serveur Oracle met à jour les tables appropriées dans le dictionnaire de données dès qu'un extent est alloué ou désalloué.

4.2.4. Locally Managed tablespaces

L'option LOCAL de la clause EXTENT MANAGEMENT spécifie qu'un tablespace est géré localement. Cette option est par défaut.

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

- DICTIONARY signifie que le tablespace est géré grâce aux tables du dictionnaire.
- LOCAL signifie que le tablespace est géré localement avec un bitmap. Si c'est cette clause est utilisée, on ne peut utiliser DEFAULT storage_clause, MINIMUM EXTENT, ou TEMPORARY.
- AUTOALLOCATE est l'option par défaut. Les utilisateurs ne peuvent spécifier une taille d'extent.
- UNIFORM signifie que le tablespace est géré avec des extents de tailles uniformes de SIZE bytes.

```
CREATE TABLESPACE userdata
DATAFILE 'u01/oradata/userdata01.dbf' SIZE 500M
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 256k;
```

Les avantages des tablespaces gérés localement sont les suivants :

- Pas de contention en mise à jour au niveau du dictionnaire.

- Par conséquence pas d'utilisation de Rollback segment pour ces transactions.
- Pas de soucis de gestion de l'espace (calcul d'un espace de stockage adéquat).
- "coalesce" automatique (fusion des espaces libres contigus pour optimiser l'espace libre).

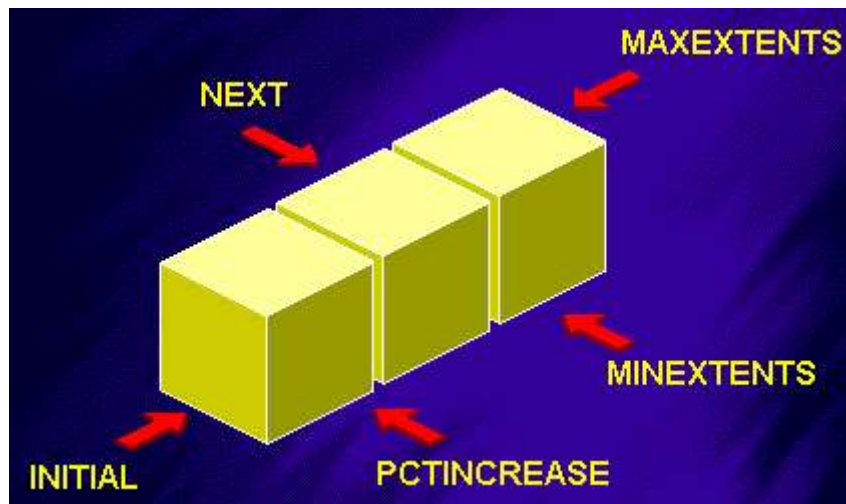
4.2.5.Dictionary Managed Tablespaces

Ce genre de tablespace a une gestion du stockage customisé, ceci est beaucoup plus flexible que les tablespaces gérés localement mais bien moins efficace.

```
CREATE TABLESPACE userdata
  DATAFILE '/u01/oradata/userdata01.dbf' SIZE 500M
  EXTENT MANAGEMENT DICTIONARY
  DEFAULT STORAGE (initial 1M NEXT 1M) ;
```

4.2.6.Paramètres de stockage

L'efficacité d'un tablespace dépend de la quantité d'espace utilisée par un segment. Ainsi, il est important de connaître les paramètres qui influent sur la quantité d'espace utilisée par un segment. Cinq paramètres déterminent et contrôlent la quantité d'espace utilisée par un segment. Il s'agit des paramètres INITIAL, NEXT, MAXEXTENTS, MINEXTENTS et PCTINCREASE.



La taille du premier extent de tout segment est définie par le paramètre INITIAL. La taille minimale du premier extent est de deux blocs ou de $2 * DB_BLOCK_SIZE$. La taille par défaut est de cinq blocs ou de $5 * DB_BLOCK_SIZE$.

Le paramètre NEXT fait référence à la taille du deuxième extent. La taille minimale de l'extent suivant est d'un bloc et la taille par défaut est de cinq blocs ou de $5 * DB_BLOCK_SIZE$.

La valeur du paramètre MINEXTENTS détermine le nombre d'extent alloués lors de la création du segment. Par défaut, la valeur minimale est de un.

Le PCTINCREASE est le pourcentage d'augmentation de la taille de NEXT.

Par exemple, NEXT est défini sur 200 Ko et PCTINCREASE est défini sur 50%. Dans une telle situation, le deuxième extent présente une taille de 200 Ko, le troisième extent une taille de 300 Ko et le quatrième extent une taille de 450 Ko.

La valeur minimale du paramètre PCTINCREASE est de zéro et sa valeur par défaut est de 50. La valeur calculée est arrondie à la valeur immédiatement supérieure de $5 * DB_BLOCK_SIZE$.

Le paramètre MAXEXTENTS détermine le nombre maximum d'extents qu'un segment peut contenir. La valeur minimale est de un. La valeur maximale par défaut dépend de la taille des blocs de données. La taille maximale peut également être spécifiée par le mot clé UNLIMITED, équivalent à une valeur de 2 147 483 645.

Il est important de garder à l'esprit que les valeurs de ces paramètres doivent représenter un objet typique qui sera créé dans le tablespace. Cependant, ces paramètres de stockage peuvent être rejetés de la création de segments individuels.

Il est possible d'utiliser la commande ALTER TABLESPACE pour modifier les paramètres par défaut. Par exemple :

```
ALTER TABLESPACE userdata
  MINIMUM EXTENT 2M;
```

Ou :

```
ALTER TABLESPACE userdata
  DEFAULT STORAGE (
    INITIAL      2M
    NEXT        2M
    MAXEXTENTS  999);
```

4.2.7. Les Undo Tablespaces

Les tablespaces de undo sont une nouveauté de Oracle 9i. Ils peuvent remplacer les rollback segments qui doivent être gérés de manière manuelle et servent au stockage de segments d'images avant modification des données pour des annulations éventuelles. Ils sont utilisés uniquement pour stocker les segments de undo. La gestion des segments de undo se fait grâce aux paramètres du fichier d'initialisation :

- UNDO_MANAGEMENT : MANUAL (rollback segments) ou AUTO (géré par l'instance donc undo tablespaces).
- UNDO_TABLESPACE : Nom du tablespace de undo.
- UNDO_SUPPRESS_ERRORS : Si positionné à TRUE, cela supprime les messages d'erreurs affichés, lorsqu'on est en mode AUTO et qu'une opération manuelle est exécutée (telle que ALTER ROLLBACK SEGMENT).
- UNDO_RETENTION : Durée pendant laquelle la donnée modifiée est gardée.

Pour créer un tablespace de undo :

```
CREATE UNDO TABLESPACE undo1
  DATAFILE 'u01/oradata/undo101.dbf' SIZE 40M;
```

Les informations d'utilisation des segments de UNDO sont stockées dans la vue V\$UNDOSTAT.

4.3. Tablespace temporaire

4.3.1. Tablespace temporaire

Ces tablespaces sont apparus avec la 9i et remplacent les segments temporaires qui étaient placés dans les tablespaces standard. Il est nécessaire de créer un tablespace temporaire par défaut autre que SYSTEM, où seront stockées uniquement les données temporaires (création d'index, jointures, tris, etc...). Un segment temporaire permet d'avoir un gain de performance lorsque, par exemple, plusieurs tris occupent trop de place pour la mémoire et doivent être stockés sur le disque dur temporairement. Ainsi la taille du segment de tris va augmenter en allouant des extents jusqu'à ce que la taille du segment soit suffisante pour stocker toutes les données que calculent les opérations de tris de l'instance.

La commande pour créer un tablespace temporaire est la suivante :

```
CREATE TEMPORARY TABLESPACE temp
TEMPFILE 'u01/oradata/temp01.dbf' SIZE 500m
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 10M;
```

Les tablespaces temporaires gérés localement ont des fichiers de données temporaires (tempfiles) qui sont identiques aux fichiers de données ordinaires. Mais certaines différences existent. Tout d'abord, les tempfiles sont toujours en mode NOLOGGING, ils ne peuvent être en lecture seule ou renommés. On ne peut créer de tempfile avec la commande ALTER DATABASE.

4.3.2. Tablespace temporaire par défaut

Lorsque une base de données est créée sans tablespace temporaire par défaut, le tablespace qui est assigné aux utilisateurs créés sans la clause TEMPORARY TABLESPACE est le tablespace SYSTEM.

Pour éviter que le tablespace système soit utilisé comme tablespace temporaire, il est nécessaire de créer un tablespace temporaire et de modifier la base de données avec la commande :

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

Après cela, tous les utilisateurs qui n'étaient pas assignés à un tablespace temporaire seront affectés directement à temp.

4.3.3. Restrictions sur les Tablespace temporaire par défaut

Les tablespaces temporaires ne peuvent être mis offline et ne peuvent être supprimés tant qu'un nouveau tablespace par défaut soit créé. D'autre part, un tablespace temporaire ne peut devenir un tablespace permanent.

4.4. Manipulation de Tablespace

4.4.1. Offline Status

L'administrateur doit assurer la gestion de bases de données. Dans le cadre des tâches de maintenance, telles que la relocalisation d'un fichier de données, l'administrateur peut être amené à rendre indisponible un tablespace tout en autorisant l'accès normal au reste de la base de données.

Pour cela, il peut mettre offline des tablespaces individuels.

La commande SQL permettant de mettre offline un tablespace est la suivante :

```
ALTER TABLESPACE tablespace  
{ ONLINE | OFFLINE [ NORMAL | TEMPORARY | IMMEDIATE ] } ;
```

Mettez le tablespace ACCOUNTS offline.

```
SQL> ALTER TABLESPACE accounts OFFLINE NORMAL;  
Tablespace altered
```

Oracle offre également une interface graphique, appelée Oracle Storage Manager, pour mettre un tablespace offline.

Le tablespace SYSTEM, et les tablespaces avec des segments de undo ou temporaires actifs ne peuvent être mis offline.

4.4.2. Tablespace en lecture seul

Pour répondre aux exigences en matière d'organisation, il se peut que l'administrateur dispose de tablespaces destinés à contenir des données de nature statique. Afin d'éviter d'avoir à procéder à des opérations de sauvegarde et de restauration de telles données statiques, l'administrateur peut mettre ces tablespaces en lecture seule. Ceci permet également de garantir qu'aucune opération d'écriture ne sera effectuée sur les données.

Les fichiers de données peuvent être stockés sur des supports en lecture seule, tels que des CD-ROM ou des disques WORM, car le serveur Oracle ne met jamais à jour les fichiers d'un tablespace en lecture seule.

La commande SQL permettant de mettre un tablespace en mode lecture seule est la suivante :

```
ALTER TABLESPACE tablespace READ { ONLY | WRITE } ;
```

Exemple :

Mettez le tablespace ACCOUNTS en lecture seule.

```
SQL> ALTER TABLESPACE accounts READ ONLY ;  
Tablespace altered.
```

Outre la commande SQL, Oracle offre une interface graphique nommée Oracle Storage Manager permettant de mettre un tablespace en mode lecture seule.

Il est possible de supprimer des objets comme des index ou des tables dans un tablespace en lecture seul. En effet, la commande DROP met à jour seulement le dictionnaire de données mais pas les fichiers physiques qui constituent le tablespace.

Cependant, plusieurs conditions doivent être remplies pour pouvoir mettre un tablespace en mode lecture seule :

- le tablespace doit être online,
- aucune transaction active ne doit être en cours d'exécution sur ce tablespace
- le tablespace ne doit contenir aucun rollback segment actif et ne doit pas être actuellement impliqué dans une sauvegarde online.

Le moyen recommandé de se conformer à ces restrictions consiste à démarrer l'instance en mode restreint.

En outre, la mise en lecture seule des tablespaces provoque un point de synchronisation sur les fichiers de données.

4.4.3. Tablespaces en lecture seul sur des supports en lecture seul

L'administrateur de bases de données souhaite diminuer le temps passé sur les sauvegardes de données. Un moyen de diminuer ce temps consiste à stocker les données statiques dans des tablespaces en lecture seule, puis à les stocker sur des supports en lecture seule.

Le stockage des données en lecture seule se déroule en trois étapes :

- mettre le tablespace en lecture seule,
- copier tous les fichiers de données du tablespace sur des supports en lecture seule,
- renommer les fichiers de données afin qu'ils pointent vers le nouvel emplacement.

4.4.4. Supprimer un Tablespace

En raison d'une modification des besoins des utilisateurs, il se peut que certains tablespaces ne soient plus nécessaires. La présence de ces tablespaces constitue un gaspillage d'espace disque. Pour résoudre ce problème, Oracle permet de supprimer de tels tablespaces.

La commande SQL permettant de supprimer de tels tablespaces est la suivante :

```
DROP TABLESPACE tablespace  
[INCLUDING CONTENTS [CASCADE CONSTRAINTS] ];
```

CASCADE CONSTRAINTS supprime les contraintes d'intégrité référentielle des tables se trouvant en dehors du tablespace. Il s'agit des contraintes qui font référence aux clés primaires et uniques des tables du tablespace supprimé.

Exemple :

Supprimez le tablespace ACCOUNTS

```
SQL> DROP TABLESPACE accounts  
2 INCLUDING CONTENTS CASCADE CONSTRAINTS ;  
Tablespace dropped.
```

Outre la commande SQL, Oracle offre une interface graphique, Oracle Storage Manager, permettant de supprimer un tablespace.

Il est important de suivre un certain nombre de règles lors de la suppression d'un tablespace.

- Il faut s'assurer que les transactions n'accèdent à aucun des segments du tablespace. Le meilleur moyen de garantir cela consiste à mettre d'abord le tablespace offline.
- Si le tablespace est supprimé avec la commande SQL, il est important de se souvenir qu'un tablespace contenant encore des données ne peut pas être supprimé sans l'option INCLUDING CONTENTS.

Lorsque le tablespace contient de nombreux objets, l'utilisation de l'option INCLUDING CONTENTS peut générer un grand nombre de rollback. Une alternative consiste à utiliser un script pour supprimer un par un les objets du tablespace, puis à supprimer le tablespace.

4.4.5. Conséquences de la suppression de tablespaces

Pour économiser de l'espace disque, l'administrateur supprime les tablespaces qui ne sont plus nécessaires.

Voici les conséquences de la suppression d'un tablespace :

- Les données du tablespace supprimé ne sont plus disponibles, elles ne peuvent donc plus être interrogées.
- Lors de la suppression d'un tablespace, seuls les pointeurs de fichiers dans le fichier de contrôle de la base de données associée sont supprimés. Pour récupérer l'espace disque utilisé par le tablespace, il faut supprimer explicitement les fichiers de données au niveau du système d'exploitation.
- Si un tablespace en lecture seule est supprimé, les segments qu'il contient sont également supprimés. Dans ce cas, seuls le dictionnaire de données et les fichiers de données doivent être mis à jour.

4.5. Fichiers de manipulation des données

4.5.1. Mettre en place l'extension automatique des fichiers de données

La clause AUTOEXTEND permet d'autoriser ou non l'extension automatique des fichiers de données. Les commandes suivantes peuvent être utilisées pour spécifier l'extension automatique.

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

Exemple :

```
ALTER DATABASE DATAFILE
'u01/oradata/userdata02.dbf' SIZE 200m
AUTOEXTEND ON NEXT 10M MAXSIZE 500M ;
```

4.5.2. Changer la taille des fichiers de données manuellement

L'administrateur doit estimer les besoins d'espace d'une base de données. Cependant, lorsque l'administrateur utilise réellement la base de données, il se peut qu'il se rende compte que son estimation des besoins d'espace est erronée. Pour faire face à ce problème, Oracle permet de redimensionner les fichiers de données.

La commande SQL permettant de redimensionner les fichiers de données d'un tablespace est la suivante :

```
ALTER DATABASE [database]
DATAFILE 'filename' [, 'filename']
RESIZE integer [ K | M ];
```

Exemple :

Affectez au fichier de données accts01.dbf du répertoire c:\ora92\database la nouvelle taille de 3 Mo à l'aide SQL*Plus :

```
SQL> ALTER DATABASE
  2 DATAFILE 'c:\ora92\database\accts01.dbf'
  3 RESIZE 3M ;
statement processed
```

4.5.3. Ajouter un fichier de données à un Tablespace

Lorsque les tables sont remplies avec des objets pendant une période donnée, l'espace disponible dans les fichiers de données s'épuise. Pour faire face à cette situation, Oracle vous permet d'ajouter des fichiers de données aux tablespaces. Cette opération alloue davantage d'espace disque aux objets de la base de données.

La commande SQL permettant d'ajouter un fichier de données à un tablespace est affichée à l'écran.

```
ALTER TABLESPACE tablespace
ADD DATAFILE 'filespec' [auto_extend_clause];
```

Exemple :

Ajoutez un fichier de données nommé accts02.dbf de 1 000 Ko situé dans le répertoire c:\ora92\database au tablespace ACCOUNTS.

```
SQL> ALTER TABLESPACE accounts
  2 ADD DATAFILE 'c:\ora92\database\accts02.dbf' SIZE 1000K;
tablespace altered.
```

Outre la commande SQL, Oracle offre une interface graphique appelée Oracle Storage Manager afin de permettre d'ajouter un fichier de données à un tablespace.

4.5.4. Déplacer un fichier de données : ALTER TABLESPACE

Fichier de données non-SYSTEM :

Avec l'augmentation du nombre d'enregistrements de la base de données, la taille du fichier de données augmente également. Par conséquent, il peut arriver que le support de stockage du fichier de données ne dispose plus de suffisamment d'espace. Pour faire face à ce problème, Oracle permet de déplacer les fichiers de données vers un nouvel emplacement.

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

Exemple :

Déplacez le fichier de données accts01.dbf du tablespace ACCOUNTS du répertoire c:\ora92\database.

→ Mettre offline le tablespace ACCOUNTS en mode normal

```
SQL> ALTER TABLESPACE accounts OFFLINE NORMAL;
Tablespace altered
```

→ Déplacez le fichier de données avec les commandes du système d'exploitation

```
C:\> copy c:\ora92\database\accts01.dbf
      d:\ ora92\database\accts01.dbf
```

```
1 file(s) copied
```

→ Modifier le chemin d'accès du fichier de données

```
SQL> ALTER TABLESPACE accounts
2  RENAME DATAFILE 'c:\ora92\database\accts01.dbf'
3  TO 'd:\ ora92\database\accts01.dbf' ;
Tablespace altered.
```

→ Mettez online le tablespace ACCOUNTS

```
SQL> ALTER TABLESPACE accounts ONLINE ;
Tablespace altered
```

→ Supprimez le fichier d'origine dans le répertoire c:\ora92\database.

Oracle Storage Manager peut déplacer les fichiers de données appartenant aux seuls tablespaces non-SYSTEM et ne contenant aucun rollback segment ou segment temporaire actif.

Fichier de données SYSTEM :

Le tablespace SYSTEM est nécessaire pour que la base de données fonctionne correctement. Cependant, dans certaines situations, telles que le remplacement du disque dur sur lequel les fichiers de données du tablespace SYSTEM sont actuellement stockés, l'administrateur doit déplacer les fichiers de données du tablespace SYSTEM vers un nouvel emplacement.

Oracle permet de déplacer un fichier de données du tablespace SYSTEM vers un nouvel emplacement. Voici les étapes à suivre :

- Arrêter la base de données
- Utiliser une commande du système d'exploitation pour déplacer les fichiers de données vers le nouvel emplacement.
- Monter la base de données
- Déplacer le fichier de données du tablespace SYSTEM
- Ouvrir la base de données

Il faut toujours fournir les noms complets des fichiers accompagnés de leur chemin d'accès pour identifier les anciens et les nouveaux fichiers de données.

Exemple :

Déplacez le fichier de données SYS1ORCL.ORA du répertoire c:\ora92\database appartenant au tablespace SYSTEM vers le répertoire d:\orclbackup.

→ Montez la base de données

→ Copiez le fichier dans le répertoire d:\orclbackup avec les commandes du système d'exploitation

→ Déplacer les fichiers de données dans la base de données :

```
SQL> ALTER DATABASE RENAME FILE 'c:\ora92\database\sys1orcl.ora'
2  TO 'd:\orclbackup\sys1orcl.ora';
```

→ Ouvrir la base de données

4.6. Autres options sur les Tablespaces

4.6.1. Configurer OMF pour la création de Tablespace

Lorsque l'on configure OMF pour créer un tablespace, un paramètre d'initialisation est spécifié : `DB_CREATE_FILE_DEST`. Ce paramètre stocke le chemin par défaut des fichiers de données. Il existe deux moyens de le configurer, soit directement dans le fichier d'initialisation soit avec la commande suivante :

```
ALTER SYSTEM SET
  db_create_file_dest = 'u01/oradata/db01';
```

4.6.2. Créer des Tablespaces avec OMF

Lorsque l'on crée un tablespace avec OMF, la clause `DATAFILE` n'est pas obligatoire. Par défaut les fichiers ont une taille de 100M, paramétrés à autoextend et de taille maximale illimitée.

```
CREATE TABLESPACE apps2_data DATAFILE SIZE 20M;
```

4.6.3. Récupérer des informations sur les Tablespaces

L'organisation des données dans le tablespaces présente plusieurs avantages. Avantages liés à l'utilisation de plusieurs tablespaces :

- L'utilisation de plusieurs tablespaces garantit que les données utilisateur et les données du dictionnaire de données peuvent être conservées séparément. Ceci permet une souplesse des opérations sur la base de données.
- Les données des différentes applications peuvent être stockées dans des tablespaces distincts. Ceci permet de garantir une bonne disponibilité des données.
- La réduction des contentions constitue un autre avantage de l'utilisation de plusieurs tablespaces. Les contentions E/S sont réduites grâce au stockage des fichiers de données de différents tablespaces sur des disques durs distincts.
- L'utilisation de plusieurs tablespaces constitue une protection contre la perte définitive de données. Cette protection est assurée par la séparation des rollback segments et des segments de données.
- L'utilisation de plusieurs tablespaces offre un degré élevé de flexibilité dans l'administration des données, car les tablespaces individuels peuvent être mis offline. Ceci permet également de garantir une bonne disponibilité des données.
- L'utilisation de plusieurs tablespaces permet également la sauvegarde des tablespaces individuels.
- L'utilisation de plusieurs tablespaces permet de réserver des tablespaces pour un type particulier d'activité sur la base de données, tel que des activités de mise à jour, des opérations en lecture seule et le stockage temporaire de segments. Ceci permet d'améliorer l'efficacité de la base de données.

L'administrateur a besoin d'informations sur les tablespaces afin d'assurer une gestion efficace de la base de données.

Oracle offre des vues du dictionnaire de données et des vues dynamiques sur les performances, qui vous permettent de rassembler des informations sur les tablespaces. Les noms des vues dynamiques sur les performances et des vues du dictionnaire de données sont les suivantes :

- DBA_TABLESPACES

Cette vue contient des informations sur les noms, les paramètres de stockage par défaut, les types et la disponibilité de tous les tablespaces.

Exemple :

Pour assurer une gestion efficace de la base de données, l'administrateur souhaite obtenir l'emplacement et les paramètres de stockage par défaut de tous les tablespaces.

```
SQL> SELECT * FROM dba_tablespaces ;
```

- DBA_DATA_FILES

Cette vue contient des informations sur les fichiers de données appartenant à un tablespace spécifique ainsi que sur leurs paramètres AUTOEXTEND

Exemple :

L'administrateur souhaite afficher les tablespaces auxquels les fichiers de données appartiennent, ainsi que la valeur de l'option AUTOEXTEND.

```
SQL> SELECT * FROM dba_data_files;
```

- V\$DATAFILE

Cette vue dynamique contient des informations sur le nom du fichier de données, la taille, la disponibilité et les noms des tablespaces auxquels les fichiers de données appartiennent.

- V\$TEMPFILE

Cette vue contient des informations sur les tablespaces temporaires.