

# Oracle 12c

## Administration

**Razvan Bizoï**

**Équivalent  
à 8 jours  
de formation  
en salle !**

**30 heures  
de travaux  
pratiques**

# Avant-propos



Oracle est le système de base de données le plus utilisé au monde. Il fonctionne de façon relativement identique sur tout type d'ordinateur. C'est pourquoi les connaissances acquises sur une plate-forme sont utilisables sur une autre, et les utilisateurs et développeurs Oracle expérimentés constituent une ressource très demandée.

## Un support de formation

Le guide de formation complet se compose de deux ouvrages, *Administration* et *Sauvegarde et restauration*, consacrés à l'administration Oracle 12c. Il vous permettra d'acquérir des connaissances solides sur les tâches fondamentales liées à l'administration des bases de données : concevoir, créer et gérer une base de données Oracle 12c

Ce premier livre, qui comprend 19 modules, est idéal comme support de formation avec un animateur, car il permet à l'élève de suivre sans avoir à prendre beaucoup de notes. Par ailleurs, le formateur peut acquérir auprès de l'éditeur Tsoft ([www.tsoft.fr](http://www.tsoft.fr)) un ensemble de diapositives, qui rythmeront la progression pédagogique, afin d'appuyer ses explications. Très complet, cet ouvrage peut aussi servir de manuel d'autoformation, car il va beaucoup plus loin qu'un simple support de cours.

Cette formation est prévue pour durer huit jours avec un animateur, à condition de posséder au préalable une connaissance suffisante du modèle relationnel et de maîtriser les langages de programmation SQL et PL/SQL.

Ce guide vise surtout à être plus clair et plus agréable à lire que les documentations techniques, exhaustives et nécessaires mais aussi ingrates, dans lesquelles vous pourrez toujours vous plonger ultérieurement. Par ailleurs, l'auteur a souhaité éviter l'écueil de ne fournir qu'une collection supplémentaire de « trucs et astuces ».

Chaque exposé théorique, qui permet de préciser les concepts et les mécanismes d'administration de la base de données, est accompagné d'une suite de travaux pratiques, afin que les lecteurs puissent comprendre les modalités d'application de chaque partie théorique et connaître les pièges à éviter.

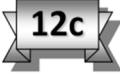
Tous les exemples présentés ici illustrent des cas réels exécutés par l'auteur et permettent de suivre la mise en œuvre de la démarche théorique. Les travaux pratiques détaillent les différences entre les tâches d'administration d'une base de données Oracle 12c dans les systèmes d'exploitation Windows et Linux.

Une fois cette lecture terminée, complétez-la par celle de l'ouvrage *Oracle 12c – Sauvegarde et restauration* (du même auteur et chez le même éditeur). Ces deux livres peuvent vous préparer aux examens de certification Oracle :

- 1Z0-062 Oracle Database 12c: Installation and Administration
- 1Z0-063 Oracle Database 12c: Advanced Administration

Si vous le désirez, vous pouvez dialoguer avec l'auteur en lui écrivant à l'adresse suivante : [razvan@bizoi.fr](mailto:razvan@bizoi.fr), ou directement depuis son site web : [www.bizoi.fr](http://www.bizoi.fr).

## Conventions utilisées dans l'ouvrage

|                                                                                     |                                                                                                                                                                      |
|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>MAJUSCULES</b>                                                                   | Les ordres SQL ou tout identifiant ou mot-clé. Utilisé pour les mots-clés, les noms des tables, les noms des champs, les noms des blocs, etc.                        |
| [ ]                                                                                 | L'information qui se trouve entre les crochets est facultative.                                                                                                      |
| [, ...]                                                                             | L'argument précédent peut être répété plusieurs fois.                                                                                                                |
| { }                                                                                 | Liste de choix exclusive.                                                                                                                                            |
|                                                                                     | Séparateur dans une liste de choix.                                                                                                                                  |
| ...                                                                                 | La suite est non significative pour le sujet traité.                                                                                                                 |
|    | La définition est valable à partir de la version Oracle 11g release 1.                                                                                               |
|   | La définition est valable à partir de la version Oracle 11g release 2.                                                                                               |
|  | La définition est valable à partir de la version Oracle 12c.                                                                                                         |
|  | La définition est uniquement valable pour l'environnement de travail UNIX/Linux.                                                                                     |
|  | La définition est uniquement valable pour l'environnement de travail Windows.                                                                                        |
|  | Ce sigle introduit un exemple de code avec la description complète telle qu'elle est présente à l'écran dans l'outil de commande.                                    |
|  | Une note qui présente des informations intéressantes en rapport avec le sujet traité.                                                                                |
|  | Un encadré <i>Attention</i> met en évidence les problèmes potentiels et vous aide à les éviter. Il peut être également une mise en garde ou une définition critique. |
|  | Un encadré <i>Conseil</i> indique, une démarche impérative à suivre pour pouvoir résoudre le problème.                                                               |

# ***Table des matières***

|                                               |                |
|-----------------------------------------------|----------------|
| <b>Module 1 L'ARCHITECTURE D'ORACLE .....</b> | <b>1-1</b>     |
| La base de données .....                      | 1-2            |
| Le stockage des données .....                 | 1-4            |
| L'instance .....                              | 1-6            |
| La zone « Shared Pool » .....                 | 1-7            |
| La zone « Buffer Cache » .....                | 1-8            |
| L'exécution d'une interrogation .....         | 1-9            |
| La zone mémoire du programme .....            | 1-11           |
| Les autres composants .....                   | 1-13           |
| Les transactions .....                        | 1-14           |
| Le segment UNDO .....                         | 1-21           |
| Les journaux .....                            | 1-22           |
| L'exécution d'un ordre LMD .....              | 1-23           |
| L'écriture des données .....                  | 1-27           |
| L'écriture des journaux .....                 | 1-28           |
| Les points de contrôle .....                  | 1-29           |
| L'archivage .....                             | 1-30           |
| Les autres processus .....                    | 1-30           |
| <br><b>Module 2 L'INSTALLATION .....</b>      | <br><b>2-1</b> |
| La préparation de l'installation .....        | 2-2            |
| Liste de prérequis .....                      | 2-2            |
| Un utilisateur pour l'installation .....      | 2-4            |

|                                             |      |
|---------------------------------------------|------|
| L'architecture OFA .....                    | 2-6  |
| Le paramétrage du système .....             | 2-10 |
| L'installation d'Oracle 12c.....            | 2-16 |
| Les tâches post-installation .....          | 2-22 |
| L'infrastructure de stockage.....           | 2-25 |
| Le privilège SYSASM .....                   | 2-26 |
| L'installation d'ASMLib.....                | 2-26 |
| La configuration des disques.....           | 2-28 |
| L'installation du Grid Infrastructure ..... | 2-31 |
| L'installation automatique.....             | 2-35 |

### **Module 3 LA CRÉATION D'UNE BASE DE DONNÉES .....3-1**

|                                   |      |
|-----------------------------------|------|
| La base de données.....           | 3-2  |
| La création manuelle .....        | 3-2  |
| La configuration du système ..... | 3-3  |
| Le fichier paramètres.....        | 3-6  |
| La création de la base .....      | 3-7  |
| La création du dictionnaire.....  | 3-10 |
| L'assistant DBCA .....            | 3-12 |

### **Module 4 LA GESTION D'UNE INSTANCE .....4-1**

|                                       |      |
|---------------------------------------|------|
| Les utilisateurs SYS et SYSTEM.....   | 4-2  |
| Les méthodes d'authentification ..... | 4-3  |
| L'authentification .....              | 4-3  |
| Le fichier de mot de passe.....       | 4-6  |
| Le fichier paramètre .....            | 4-9  |
| Le fichier SPFILE .....               | 4-17 |
| La création du fichier SPFILE .....   | 4-18 |
| La modification des paramètres .....  | 4-20 |
| Le démarrage et l'arrêt .....         | 4-22 |
| La commande STARTUP .....             | 4-24 |
| La commande ALTER DATABASE.....       | 4-27 |
| L'arrêt du serveur .....              | 4-29 |
| Les vues dynamiques .....             | 4-32 |
| Les fichiers de trace.....            | 4-37 |
| L'architecture de diagnostic .....    | 4-39 |

### **Module 5 L'INFRASTRUCTURE DE STOCKAGE .....5-1**

|                             |      |
|-----------------------------|------|
| L'outil Oracle Restart..... | 5-2  |
| L'instance ASM .....        | 5-13 |

|                                                 |            |
|-------------------------------------------------|------------|
| Le groupe de disques .....                      | 5-22       |
| Le groupe d'échec .....                         | 5-23       |
| La résolution de panne.....                     | 5-23       |
| Les fichiers logiques.....                      | 5-24       |
| La création d'un groupe.....                    | 5-25       |
| La modification d'un groupe .....               | 5-29       |
| Le rééquilibrage dynamique .....                | 5-31       |
| La resynchronisation rapide.....                | 5-34       |
| L'utilitaire ASMCA.....                         | 5-35       |
| L'utilitaire ASMCMD .....                       | 5-35       |
| <b>Module 6 L'ARCHITECTURE MUTUALISÉE.....</b>  | <b>6-1</b> |
| L'architecture Oracle 11g.....                  | 6-2        |
| L'architecture mutualisée .....                 | 6-3        |
| La création d'un conteneur CDB.....             | 6-4        |
| La connexion aux CDB et PDB.....                | 6-7        |
| L'arrêt et le démarrage .....                   | 6-8        |
| La création d'une PDB .....                     | 6-9        |
| Le clonage d'une base insérée .....             | 6-11       |
| L'effacement d'une base insérée .....           | 6-14       |
| La gestion d'une base insérée.....              | 6-14       |
| L'extraction et l'insertion .....               | 6-16       |
| SQL Developer .....                             | 6-19       |
| <b>Module 7 L'ARCHITECTURE ORACLE NET .....</b> | <b>7-1</b> |
| Le modèle de réseau Oracle.....                 | 7-2        |
| Le descripteur de connexion.....                | 7-3        |
| Les services de base de données.....            | 7-5        |
| Les méthodes de résolution de noms.....         | 7-9        |
| La résolution de noms simplifiés.....           | 7-10       |
| Les alias de services .....                     | 7-11       |
| Le processus de connexion.....                  | 7-13       |
| Le processus LISTENER .....                     | 7-17       |
| L'enregistrement automatique.....               | 7-24       |
| LOCAL_LISTENER.....                             | 7-25       |
| L'enregistrement statique .....                 | 7-26       |
| Les multiples processus LISTENER.....           | 7-28       |
| L'outil TNSPING .....                           | 7-31       |
| Liens de base de données .....                  | 7-32       |

## **Module 8 LE FICHIER DE CONTRÔLE .....8-1**

|                                            |      |
|--------------------------------------------|------|
| La gestion automatique des fichiers .....  | 8-2  |
| L'emplacement des fichiers .....           | 8-2  |
| Le nom des fichiers .....                  | 8-4  |
| Le contenu du fichier de contrôle.....     | 8-7  |
| La taille du fichier de contrôle .....     | 8-8  |
| L'information du fichier de contrôle ..... | 8-11 |
| Le multiplexage.....                       | 8-12 |

## **Module 9 LES FICHIERS JOURNAUX .....9-1**

|                                        |      |
|----------------------------------------|------|
| Les fichiers journaux.....             | 9-2  |
| Les groupes de fichiers journaux ..... | 9-3  |
| Le traitement des données .....        | 9-4  |
| Le dimensionnement des fichiers.....   | 9-6  |
| La vue V\$LOG.....                     | 9-7  |
| La création d'un groupe .....          | 9-9  |
| La suppression d'un groupe .....       | 9-11 |
| La suppression d'un membre .....       | 9-14 |
| Le changement des groupes .....        | 9-15 |
| Le mode NOARCHIVELOG .....             | 9-17 |
| L'archivage.....                       | 9-18 |
| Le mode ARCHIVELOG.....                | 9-21 |
| La gestion des archives .....          | 9-23 |

## **Module 10 LES TABLESPACES .....10-1**

|                                         |       |
|-----------------------------------------|-------|
| Le tablespace .....                     | 10-2  |
| Les types de tablespaces.....           | 10-3  |
| La création d'un tablespace.....        | 10-4  |
| Le tablespace par défaut.....           | 10-7  |
| Le tablespace BIGFILE.....              | 10-8  |
| La taille du bloc.....                  | 10-10 |
| Le cryptage transparent .....           | 10-12 |
| Le tablespace temporaire.....           | 10-15 |
| Le groupe tablespaces temporaires ..... | 10-17 |
| Le tablespace undo .....                | 10-18 |

## **Module 11 LA GESTION DES TABLESPACES.....11-1**

|                                            |      |
|--------------------------------------------|------|
| Les informations sur les tablespaces ..... | 11-2 |
| Les informations sur les fichiers .....    | 11-4 |

|                                            |             |
|--------------------------------------------|-------------|
| L'agrandissement d'un tablespace.....      | 11-6        |
| L'extension d'un tablespace .....          | 11-7        |
| L'extension d'un fichier .....             | 11-7        |
| Le compactage d'un tablespace .....        | 11-8        |
| Le tablespace OFFLINE .....                | 11-10       |
| Le fichier OFFLINE .....                   | 11-12       |
| La création d'un fichier de données .....  | 11-13       |
| Les informations sur les tablespaces ..... | 11-14       |
| Le changement de nom.....                  | 11-15       |
| Le déplacement d'un tablespace.....        | 11-15       |
| Le déplacement des fichiers .....          | 11-17       |
| Le déplacement en ligne .....              | 11-19       |
| La suppression d'un tablespace .....       | 11-21       |
| <b>Module 12 LES SEGMENTS UNDO .....</b>   | <b>12-1</b> |
| Le segment UNDO .....                      | 12-2        |
| L'utilisation des segments UNDO.....       | 12-2        |
| La lecture cohérente.....                  | 12-3        |
| La taille et la rétention.....             | 12-4        |
| La gestion du tablespace UNDO .....        | 12-6        |
| La conservation des blocs.....             | 12-6        |
| La suppression d'un tablespace UNDO.....   | 12-8        |
| Flashback.....                             | 12-9        |
| Fonctions de conversion .....              | 12-10       |
| Interrogation FLASHBACK .....              | 12-11       |
| Interrogation des versions.....            | 12-13       |
| DBMS_FLASHBACK .....                       | 12-15       |
| L'interrogation des transactions .....     | 12-17       |
| L'annulation des transactions .....        | 12-22       |
| Le FLASHBACK ARCHIVE .....                 | 12-25       |
| Le FLASHBACK table .....                   | 12-28       |
| Le FLASHBACK DROP .....                    | 12-31       |
| <b>Module 13 LES TABLES .....</b>          | <b>13-1</b> |
| Les types de données .....                 | 13-2        |
| Les grands objets .....                    | 13-10       |
| La création d'une table .....              | 13-10       |
| Stockage des données LOB .....             | 13-15       |
| La table temporaire.....                   | 13-17       |
| La création depuis une requête .....       | 13-18       |



|                                     |       |
|-------------------------------------|-------|
| Définition de contraintes .....     | 13-20 |
| NOT NULL .....                      | 13-22 |
| CHECK .....                         | 13-23 |
| PRIMARY KEY .....                   | 13-24 |
| UNIQUE.....                         | 13-26 |
| REFERENCES.....                     | 13-27 |
| Ajouter une nouvelle colonne .....  | 13-32 |
| Modification d'une colonne .....    | 13-32 |
| Supprimer une colonne.....          | 13-33 |
| Renommer une table.....             | 13-36 |
| Modification d'une contrainte ..... | 13-37 |
| Table en lecture seule .....        | 13-40 |
| Suppression d'une table .....       | 13-40 |

## **Module 14 LA GESTION DU STOCKAGE.....14-1**

|                                            |       |
|--------------------------------------------|-------|
| La structure du stockage.....              | 14-2  |
| Les types de segments .....                | 14-3  |
| Les paramètres de stockage.....            | 14-5  |
| Les informations sur le stockage.....      | 14-6  |
| La gestion locale.....                     | 14-7  |
| Les extents de taille uniforme .....       | 14-9  |
| L'allocation automatique.....              | 14-10 |
| Les options obsolètes .....                | 14-11 |
| L'allocation et libération d'extents ..... | 14-12 |
| Le bloc de données .....                   | 14-13 |
| La gestion automatique de l'espace .....   | 14-14 |
| La gestion automatique des blocs.....      | 14-15 |
| Le package DBMS_SPACE.....                 | 14-15 |

## **Module 15 L'OPTIMISATION DU STOCKAGE.....15-1**

|                                     |       |
|-------------------------------------|-------|
| La migration et le chaînage .....   | 15-2  |
| L'élimination de migrations .....   | 15-5  |
| Le HWM d'une table.....             | 15-6  |
| L'évolution du HWM.....             | 15-7  |
| La réorganisation d'une table ..... | 15-11 |
| Le déplacement d'une table.....     | 15-12 |
| Le compactage de l'espace.....      | 15-14 |
| La redéfinition d'une table .....   | 15-14 |
| La compression .....                | 15-19 |
| La compression 11g .....            | 15-21 |

|                                           |             |
|-------------------------------------------|-------------|
| La compression 12c .....                  | 15-24       |
| L'analyse de l'activité .....             | 15-27       |
| La compression automatique .....          | 15-28       |
| Le déplacement automatique .....          | 15-30       |
| <b>Module 16 LES INDEX.....</b>           | <b>16-1</b> |
| Les types d'index.....                    | 16-2        |
| Index B-tree .....                        | 16-7        |
| Avantages et inconvénients .....          | 16-9        |
| Les cas d'utilisation .....               | 16-11       |
| Index bitmap .....                        | 16-14       |
| Index bitmap de jointure.....             | 16-18       |
| Table organisée en index .....            | 16-22       |
| La gestion d'index .....                  | 16-24       |
| <b>Module 17 LE PARTITIONNEMENT .....</b> | <b>17-1</b> |
| Le partitionnement.....                   | 17-2        |
| Partitionnement par hachage .....         | 17-2        |
| Partitionnement par plages .....          | 17-5        |
| Partitionnement par intervalle .....      | 17-8        |
| Partitionnement par liste.....            | 17-10       |
| Partitionnement par référence.....        | 17-13       |
| Ajout et suppression .....                | 17-17       |
| Fusion .....                              | 17-18       |
| Répartition .....                         | 17-20       |
| Gestion du partitionnement .....          | 17-21       |
| Partitionnement composite .....           | 17-22       |
| Les index locaux.....                     | 17-30       |
| Les index globaux.....                    | 17-32       |
| <b>Module 18 LES UTILISATEURS.....</b>    | <b>18-1</b> |
| La gestion des mots de passe.....         | 18-2        |
| Les paramètres de mots de passe.....      | 18-2        |
| Création d'un profil .....                | 18-4        |
| La création d'un utilisateur.....         | 18-5        |
| L'authentification par os.....            | 18-8        |
| La modification d'un utilisateur .....    | 18-9        |
| La suppression d'un utilisateur.....      | 18-11       |
| Informations sur les utilisateurs.....    | 18-12       |
| Restaurer un utilisateur.....             | 18-13       |

|                                 |       |
|---------------------------------|-------|
| L'architecture mutualisée ..... | 18-15 |
|---------------------------------|-------|

## **Module 19 LES PRIVILÈGES .....19-1**

|                                          |       |
|------------------------------------------|-------|
| Les privilèges .....                     | 19-2  |
| Privilèges de niveau système.....        | 19-2  |
| SYSDBA et SYSOPER privilèges .....       | 19-3  |
| Les privilèges objet .....               | 19-4  |
| Octroyer des privilèges système.....     | 19-5  |
| Octroyer des privilèges objet.....       | 19-8  |
| Les privilèges communs.....              | 19-12 |
| Révoquer des privilèges objet .....      | 19-13 |
| Les informations sur les privilèges..... | 19-14 |
| Création d'un rôle.....                  | 19-16 |
| Gestion d'un rôle .....                  | 19-18 |
| Les rôles par défaut .....               | 19-18 |
| Activation d'un rôle .....               | 19-20 |
| Les rôles standards .....                | 19-20 |
| Les informations sur les rôles.....      | 19-21 |

## **INDEX .....I-1**

# 3

- *Instance et base de données*
- *CREATE DATABASE*
- *Les scripts de création*
- *DBCA (DataBase Configuration Assistant)*

## La création d'une base de données



### Objectifs

À la fin de ce module, vous serez à même d'effectuer les tâches suivantes :

- Décrire les étapes de création de la base de données.
- Préparer le système d'exploitation pour l'installation de la base de données.
- Créer une base de données manuellement.
- Décrire les étapes de création de la base de données.
- Utiliser DBCA (DataBase Configuration Assistant)
- Choisir les paramètres et les propriétés de la base de données.
- Créer un modèle de création de base de données.
- Créer les scripts de création d'une base de données.



### Contenu

|                             |     |                             |      |
|-----------------------------|-----|-----------------------------|------|
| La base de données          | 3-2 | La création de la base      | 3-7  |
| La création manuelle        | 3-2 | La création du dictionnaire | 3-10 |
| La configuration du système | 3-3 | L'assistant DBCA            | 3-12 |
| Le fichier paramètres       | 3-6 |                             |      |

## La base de données

---

Comme nous avons pu le voir précédemment, la base de données est l'ensemble des trois types de fichiers obligatoires : les fichiers de contrôles, les fichiers de données et les fichiers des journaux. Les fichiers de la base de données sont des fichiers binaires ne pouvant être lus ou écrits directement.

Le fichier des paramètres contient les paramètres de démarrage de la base et d'autres valeurs qui déterminent l'environnement dans lequel la base s'exécute. Lorsqu'elle est démarrée, le fichier des paramètres est lu et plusieurs structures mémoire sont allouées en fonction de son contenu.

Le fichier de mots de passe est utilisé pour établir l'authenticité des utilisateurs privilégiés de la base de données.

Vous pouvez utiliser deux modalités de création de la base de données :

- Manuellement à l'aide de l'instruction SQL « CREATE DATABASE ».
- Avec l'assistant de configuration de base des données (DBCA DataBase Configuration Assistant).

## La création manuelle

---

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

La création d'une base de données est une tâche consistant à préparer plusieurs fichiers du système d'exploitation, qu'il n'est nécessaire d'effectuer qu'une fois, quel que soit le nombre de fichiers de données de la base. Il s'agit d'une tâche très importante, l'administrateur de la base de données devant déterminer des paramètres de la base, tels que le nom de la base ou la taille du bloc, qui ne peuvent plus être modifiés après la création.

Les étapes de création d'une base de données sont :

- Choisissez le nom de l'instance.
- Choisissez le nom de la base des données.
- Préparez le système d'exploitation.
- Créez le fichier des paramètres « **PFILE** ».
- Créez le fichier des paramètres à serveur « **SPFILE** ».
- Démarrez votre instance.
- Créez la base de données.
- Exécutez les scripts de création du dictionnaire de données.
- Exécutez les scripts de création de votre schéma relationnel.
- Sauvegardez votre base de données.

### Le nom de la base de données

Le nom de la base de données, stocké dans le paramètre « **DB\_NAME** » ne doit pas dépasser 8 caractères. Le nom de la base de données est également utilisé dans la commande SQL « **CREATE DATABASE** » ; ainsi il est stocké directement dans le fichier des contrôles. Il n'est plus possible de modifier le nom de la base de données après sa création.

## Le nom du domaine

Le nom du domaine « **DB\_DOMAIN** » spécifie le domaine de réseau où la base de données est créée. Si la base de données que vous vous apprêtez à créer devait faire partie d'un système de bases de données distribuées, accordez une attention particulière à ce paramètre d'initialisation de la base de données avant la création. Il permet de définir avec un nom unique de la base de données dans votre domaine qui est :

« **DB\_NAME** » . « **DB\_DOMAIN** »

## Le nom unique de la base de données

Dans le cas où plusieurs bases de données ont le même nom à l'intérieur de votre domaine, vous devez préciser le paramètre « **DB\_UNIQUE\_NAME** » qui représente le nom unique de la base de données dans votre domaine. Le nom unique de la base de données peut atteindre jusqu'à 30 caractères.

## Le nom de l'instance

Le nom de l'instance est stocké dans le paramètre « **INSTANCE\_NAME** », un paramètre facultatif depuis la version Oracle 10g.

Il est préférable de choisir un nom d'instance et un nom de base de données identiques.

### **Attention**

Pour vous connecter à la base de données à l'aide de SQL\*Plus, vous avez deux solutions :

- La connexion à l'aide du réseau Oracle Net.
- La deuxième solution est possible uniquement si vous vous connectez directement à partir du serveur. Il s'agit d'une connexion à l'instance par défaut du serveur. L'instance par défaut du serveur est définie par la variable d'environnement « **ORACLE\_SID** ».

Par conséquent, il est impératif de définir cette variable avant de se connecter à SQL\*Plus pour pouvoir créer la base de données.

Dans l'environnement Windows, la variable est configurée au sein de la base de registres dans « **HKEY\_LOCAL\_MACHINE\SOFTWARE\ORACLE\KEY\_\*** ». Vous pouvez la modifier directement ou utiliser une variable d'environnement dans votre environnement de commandes.

# La configuration du système

---

## **L'arborescence de répertoires**

Rappelez-vous, pour chaque base de données vous devez créer une arborescence de répertoires pour les fichiers d'administration et une deuxième arborescence de répertoires pour l'ensemble des fichiers de la base de données.

Le répertoire pour stocker les fichiers d'administration se trouve dans le répertoire suivant :

\$ORACLE\_BASE/admin/DB\_NAME ou %ORACLE\_BASE%\admin\DB\_NAME

## **La méthode d'authentification**

Vous devez choisir la méthode que vous souhaitez utiliser pour l'authentification de l'utilisateur administrateur de votre base de données. Il doit disposer de la totalité des privilèges relatifs au système d'exploitation ou utiliser l'authentification par fichier mot de passe. Pour plus d'informations voir le Module « La gestion d'une instance ».



## La création du service

Dans l'environnement Windows, l'instance ne peut fonctionner que s'il existe un environnement pour faire tourner les processus d'arrière-plan et partager les zones mémoire.

L'environnement nécessaire est un service Windows. Oracle fournit un utilitaire pour la gestion des différents services Windows.

L'utilitaire a la syntaxe suivante :

```
ORADIM -{NEW | DELETE | EDIT } -SID SID
        [-SYSPWD mot-de-passe] [-MAXUSERS nombre]
        [-PFILE fichier | -SPFILE]
        [-STARTMODE auto|manual] [-SHUTMODE normal|immediate|abort]
        [-SRVCSTART system | demand] [-TIMEOUT secs]
```

|                  |                                                                                                                                                                                                                                                          |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>NEW</b>       | L'argument permet la création d'un service Windows.                                                                                                                                                                                                      |
| <b>DELETE</b>    | L'argument permet l'effacement d'un service Windows.                                                                                                                                                                                                     |
| <b>EDIT</b>      | L'argument permet la modification des paramètres d'un service Windows, à savoir passer du mode de démarrage du service automatique en mode de démarrage manuel et l'inverse.                                                                             |
| <b>SID</b>       | L'argument permet de préciser le nom de l'instance.                                                                                                                                                                                                      |
| <b>SYSPWD</b>    | L'argument permet d'initialiser un mot de passe qui sera stocké dans le fichier de mots de passe. En effet « <b>ORADIM</b> » vous permet des créer également un fichier de mots de passe positionné dans le répertoire par défaut de la base de données. |
| <b>MAXUSERS</b>  | Le nombre maximum d'utilisateurs gérés par le fichier de mots de passe.                                                                                                                                                                                  |
| <b>STARTMODE</b> | L'argument détermine si au démarrage du service Windows l'instance est démarrée automatiquement ou manuellement.                                                                                                                                         |
| <b>SHUTMODE</b>  | L'argument détermine le mode d'arrêt de l'instance à l'arrêt du service Windows.                                                                                                                                                                         |
| <b>SRVCSTART</b> | L'argument vous permet de définir si le service Windows démarre automatiquement ou manuellement après un redémarrage du serveur.                                                                                                                         |
| <b>PFILE</b>     | Le nom du fichier des paramètres par défaut utilisés pour démarrer l'instance.                                                                                                                                                                           |
| <b>SPFILE</b>    | Le nom du fichier des paramètres serveur par défaut utilisés pour démarrer l'instance.                                                                                                                                                                   |
| <b>TIMEOUT</b>   | Le temps d'attente maximum, en secondes, pour un service Windows avant l'arrêt final.                                                                                                                                                                    |

Nous allons considérer à présent un exemple mettant en œuvre la création d'un service Windows à l'aide de l'utilitaire « **ORADIM** ». Pour pouvoir utiliser l'environnement de l'instance, il faut d'abord initialiser une variable d'environnement déjà rencontrée auparavant qui s'appelle « **ORACLE\_SID** ».



```
C:\>set ORACLE_SID=ambre
C:\>sqlplus /nolog
SQL*Plus: Release 12.1.0.1.0 Production on Jeu. Févr. 27 19:02:28 2014
Copyright (c) 1982, 2013, Oracle. All rights reserved.

SQL> connect / as sysdba
ERROR:
```

```
ORA-12560: TNS : erreur d'adaptateur de protocole
```

```
SQL> exit
```

```
C:\>oradim -new -sid AMBRE -startmode manual -spfile -syspwd Razvanpwd3
```

```
Saisissez le mot de passe de l'utilisateur du service Oracle : XXXX
```

```
Instance créée.
```

```
C:\>sqlplus / as sysdba
```

```
SQL*Plus: Release 12.1.0.1.0 Production on Jeu. Févr. 27 19:05:05 2014
```

```
Copyright (c) 1982, 2013, Oracle. All rights reserved.
```

```
Connecté à une instance inactive.
```

```
SQL> $dir O:\app\oracle\product\12.1.0\db_home\database\PWDAMBRE.ORA /B
```

```
PWDAMBRE.ORA
```

Après l'initialisation de la variable d'environnement, on se connecte à SQL\*Plus sans définir les informations de connexion à la base de données. La connexion à l'instance par défaut « **ambre** » définie par la variable d'environnement « **ORACLE\_SID** » ne peut pas être effectuée car le service Windows n'a pas encore été défini. Vous pouvez remarquer que, dès lors que service Windows pour l'instance appelée « **ambre** » est créé, il est possible de se connecter normalement.

**12c**

L'installation d'Oracle 12c pour Windows, pour des raisons de sécurité, permet de créer un compte utilisateur Windows standard pour gérer le répertoire d'origine de cette installation. Ce compte est utilisé également pour exécuter les services Windows pour le répertoire d'origine Oracle Home.

On crée le service Windows pour l'instance appelée « **ambre** » et en même temps un fichier des mots de passe « **PWDAMBRE.ORA** » avec le mot de passe « **Razvanpwd3** » pour les administrateurs de la base des données. La dernière commande de l'exemple vous permet de visualiser l'emplacement du fichier de mots de passe.

## Attention

Il faut faire la différence entre le service Windows et l'instance de la base de données. Vous pouvez avoir le cas où le service Windows est démarré mais où l'instance n'est pas démarrée pour autant. Dans ce cas vous pouvez démarrer manuellement votre instance.

Toutefois, si le service Windows n'est pas démarré, vous ne pouvez pas accéder à l'instance.



| Nom                           | État                 | Type de démarra... | Ouvrir une session en tant que |
|-------------------------------|----------------------|--------------------|--------------------------------|
| OracleASMServices+ASM         | En cours d'exécution | Manuel             | Système local                  |
| OracleJobSchedulerAMBRE       |                      | Désactivé          | .\oracle                       |
| OracleOHSservice              | En cours d'exécution | Automatique        | Système local                  |
| OracleOraDB12Home1TNSListener | En cours d'exécution | Automatique        | .\oracle                       |
| OracleRemExecServiceV2        | En cours d'exécution | Désactivé          | Système local                  |
| OracleServiceAMBRE            | En cours d'exécution | Manuel             | .\oracle                       |
| OracleVssWriterAMBRE          | En cours d'exécution | Automatique        | .\oracle                       |

Le service ainsi créé n'est pas démarré automatiquement au démarrage de la machine. Pour que le service démarre automatiquement, il faut exécuter la commande suivante :

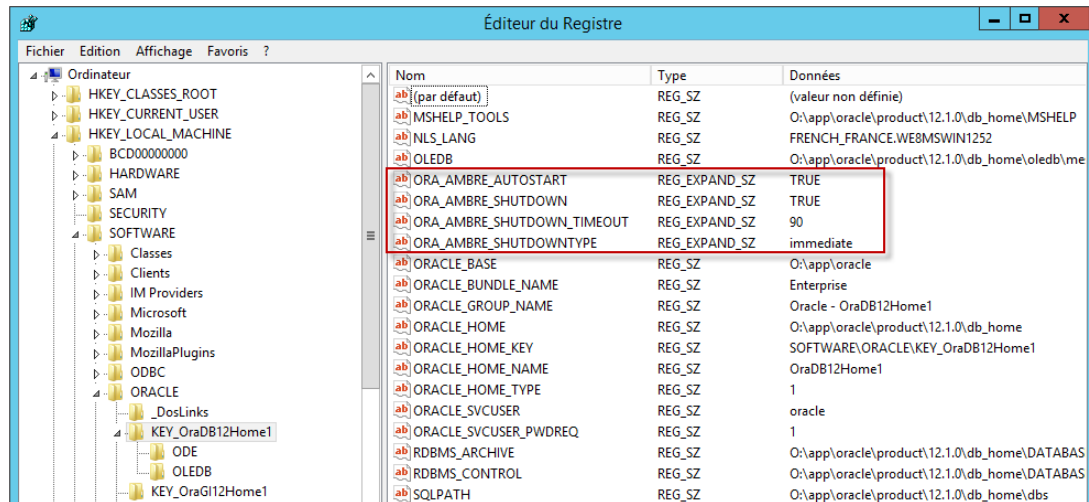
```
C:\>oradim -edit -sid AMBRE -startmode auto -srvstart system
```

Chaque fois que vous créez un service, quatre clés sont créées qui déterminent le mode de fonctionnement de ce service :

- ORA\_\*\_AUTOSTART** L'argument détermine si au démarrage du service Windows l'instance est démarrée automatiquement.
- ORA\_\*\_SHUTDOWN** L'argument détermine si à l'arrêt du service Windows l'instance est arrêtée automatiquement.
- ORA\_\*\_SHUTDOWN\_TIMEOUT** L'argument définit le temps d'attente maximum en secondes avant que le service ne s'arrête.



**ORA\_\*\_SHUTDOWNTYPE** L'argument définit le type d'arrêt de l'instance.



## Le fichier paramètres

Nous allons créer le fichier paramètres utilisant les différents paramètres nécessaires pour le démarrage et le bon fonctionnement de votre base de données. Pour plus d'informations voir le Module « La gestion d'une instance ».

### Le fichier paramètres SPFILE

Vous devez vous connecter à l'instance et, à l'aide de la commande SQL, « **CREATE SPFILE** », créer le fichier des paramètres serveur.



```
C:\>set ORACLE_SID=ambre
C:\>type O:\app\oracle\admin\ambre\scripts\init.ora
compatible=12.1.0.0.0
db_block_size=8192
db_create_file_dest=D:\donnees\oradata
db_create_online_log_dest_1=R:\recuperations\oradata
db_recovery_file_dest=O:\app\oracle\fast_recovery_area
db_recovery_file_dest_size=48G
open_cursors=300
memory_target=3240m
db_domain=olimp.fr
db_name="ambre"
nls_language="FRENCH"
nls_territory="FRANCE"
processes=300
undo_tablespace=UNDOTBS1
dispatchers="(PROTOCOL=TCP) (SERVICE=ambreXDB) "
audit_file_dest="O:\app\oracle\admin\ambre\adump"
audit_trail=db
remote_login_passwordfile=EXCLUSIVE
C:\>sqlplus / as sysdba
Connecté à une instance inactive.
SQL> create spfile from pfile=
  2 'O:\app\oracle\admin\ambre\scripts\init.ora';

Fichier créé.
SQL> startup nomount
```

Instance ORACLE lancée.

```
Total System Global Area 3390558208 bytes
Fixed Size                2408312 bytes
Variable Size             1962934408 bytes
Database Buffers         1409286144 bytes
Redo Buffers              15929344 bytes
```

## La création de la base

Les fichiers de la base de données sont des fichiers binaires qui ne peuvent pas être lus par d'autres programmes que ceux des processus d'arrière-plan de l'instance.

La création de ces fichiers respecte la même règle, à savoir que l'on a besoin de l'instance pour pouvoir travailler avec les fichiers de données.

La création des fichiers de la base de données est effectuée à partir du **SQL\*Plus** utilisant la commande suivante :

```
CREATE DATABASE nom { ARCHIVELOG | NOARCHIVELOG }
    MAXLOGFILES val_int    MAXLOGMEMBERS val_int
    MAXLOGHISTORY val_int  MAXDATAFILES valr_int MAXINSTANCES val_int
    CONTROLFILE REUSE LOGFILE GROUP nom_groupe [fichier] [,...]
DATAFILE [fichier
    [AUTOEXTEND {
        OFF | ON [NEXT val_int [{K|M|G}]]
        [MAXSIZE{UNLIMITED | val_int [{K|M|G}]] } ] ] [,...]
SYSAUX DATAFILE [fichier ...]
UNDO TABLESPACE nom_tablespace [DATAFILE fichier ...]
DEFAULT TEMPORARY TABLESPACE nom_tablespace [TEMPFILE fichier ...]
DEFAULT TABLESPACE nom_tablespace [DATAFILE fichier ...]
CHARACTER SET val_charset
NATIONAL CHARACTER SET val_charset
SET TIME_ZONE = time_zone
SET DEFAULT [BIGFILE | SMALLFILE] TABLESPACE;
```

**11g**

### **MAXLOGFILES**

Spécifie le nombre maximal de groupes de fichiers redo log en ligne qui peuvent être créés dans la base. Oracle utilise cette valeur pour déterminer la quantité d'espace à allouer au nom des fichiers redo log dans le fichier de contrôle.

### **MAXLOGMEMBERS**

Spécifie le nombre maximal de membres, ou de copies identiques, pour un groupe de fichiers redo log. Oracle utilise cette valeur pour déterminer la quantité d'espace à allouer au nom des fichiers redo log dans le fichier de contrôle.

### **MAXLOGHISTORY**

Spécifie le nombre maximal de fichiers redo log archivés pour la récupération de média automatique avec l'option Real Application Clusters.

|                           |                                                                                                                                                                                                      |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>MAXDATAFILES</b>       | Spécifie le dimensionnement initial de la section des fichiers de données dans le fichier de contrôle au moment de l'exécution.                                                                      |
| <b>MAXINSTANCES</b>       | Spécifie le nombre maximal d'instances qui peuvent simultanément monter et ouvrir la base de données.                                                                                                |
| <b>CONTROLFILE REUSE</b>  | L'argument vous permet de réutiliser les fichiers de contrôle existants identifiés par le paramètre « <b>CONTROL FILES</b> ».                                                                        |
| <b>LOGFILE</b>            | Spécifie un ou plusieurs fichiers à utiliser en tant que fichiers redo log.                                                                                                                          |
| <b>GROUP nom_groupe</b>   | Identifie de façon unique un groupe de fichiers redo log. Une base de données doit disposer d'au moins deux groupes de fichiers redo log.                                                            |
| <b>ARCHIVELOG</b>         | Spécifie que le contenu d'un groupe de fichiers redo log doit être archivé avant d'être réutilisé.                                                                                                   |
| <b>NOARCHIVELOG</b>       | Spécifie que le contenu d'un groupe de fichiers redo log ne doit pas être archivé avant d'être réutilisé.                                                                                            |
| <b>DATAFILE</b>           | Spécifie un ou plusieurs fichiers à utiliser en tant que fichier de données. Tous ces fichiers font partie du tablespace « <b>SYSTEM</b> ».                                                          |
| <b>SYSAUX</b>             | Spécifie un ou plusieurs fichiers à utiliser en tant que fichier de données. Tous ces fichiers font partie du tablespace « <b>SYSAUX</b> ».                                                          |
| <b>AUTOEXTEND</b>         | L'argument active ou désactive l'extension automatique d'un nouveau fichier de données ou d'un fichier temporaire. Si vous omettez cette clause, ces fichiers ne seront pas automatiquement étendus. |
| <b>NEXT val_int</b>       | Spécifie la taille, en octets, de l'incrément d'espace disque suivant à allouer automatiquement.                                                                                                     |
| <b>K</b>                  | Valeurs spécifiées pour préciser la taille en kilo-octets.                                                                                                                                           |
| <b>M</b>                  | Valeurs spécifiées pour préciser la taille en mégaoctets.                                                                                                                                            |
| <b>G</b>                  | Valeurs spécifiées pour préciser la taille en gigaoctets.                                                                                                                                            |
| <b>MAXSIZE</b>            | L'argument spécifie l'espace disque maximal autorisé pour l'extension automatique du fichier.                                                                                                        |
| <b>UNLIMITED</b>          | Définit une allocation d'espace illimitée pour le fichier.                                                                                                                                           |
| <b>DEFAULT TEMPORARY</b>  | L'argument définit le tablespace temporaire par défaut pour la base de données. Si vous omettez cette clause, le tablespace « <b>SYSTEM</b> » servira de tablespace temporaire par défaut.           |
| <b>DEFAULT TABLESPACE</b> | L'argument définit le tablespace par défaut pour la base de données.                                                                                                                                 |
| <b>MANAGEMENT LOCAL</b>   | Indique qu'une partie du tablespace est réservée pour un bitmap de gestion des extents.                                                                                                              |
| <b>UNIFORM</b>            | Tous les extents d'un tablespace temporaire sont de la même taille.                                                                                                                                  |
| <b>UNDO TABLESPACE</b>    | L'argument vous permet de définir le tablespace qui servira à stocker les données undo.                                                                                                              |

|                      |                                                                                                                                                                                              |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>CHARACTER SET</b> | Spécifie le jeu de caractères utilisé par la base de données pour stocker les données. Les jeux supportés et la valeur par défaut de ce paramètre dépendent de votre système d'exploitation. |
| <b>NATIONAL</b>      | Spécifie le jeu de caractères national utilisé pour stocker les données dans des colonnes définies spécifiquement avec les types « <b>NCHAR</b> », « <b>NCLOB</b> » ou « <b>NVARCHAR2</b> ». |
| <b>SET TIME_ZONE</b> | L'argument permet de définir la zone horaire de la base de données.                                                                                                                          |
| <b>BIGFILE</b>       | Indique que par défaut tous les tablespaces sont de type « <b>BIGFILE</b> ». Le tablespace est créé avec un seul fichier pouvant contenir jusqu'à 2 <sup>32</sup> blocs.                     |
| <b>SMALLFILE</b>     | Indique que par défaut tous les tablespaces sont de type « <b>SMALLFILE</b> ». Le tablespace peut avoir plusieurs fichiers, chacun pouvant contenir jusqu'à 2 <sup>22</sup> blocs.           |

L'exemple suivant récapitule l'ensemble des commandes **SQL** et **SQL\*Plus** jusqu'à la création de la base de données.



```
SQL> CREATE DATABASE "ambre"
  2  MAXINSTANCES 8 MAXLOGHISTORY 1 MAXLOGFILES 16
  3  MAXLOGMEMBERS 3 MAXDATAFILES 1024
  4  DATAFILE SIZE 700M AUTOEXTEND ON NEXT 10240K
  5      MAXSIZE UNLIMITED EXTENT MANAGEMENT LOCAL
  6  SYSAUX DATAFILE SIZE 550M AUTOEXTEND ON NEXT 10240K
  7      MAXSIZE UNLIMITED
  8  DEFAULT TEMPORARY TABLESPACE TEMP TEMPFILE SIZE 20M
  9      AUTOEXTEND ON NEXT 640K MAXSIZE UNLIMITED
 10  UNDO TABLESPACE "UNDOTBS1" DATAFILE SIZE 200M
 11      AUTOEXTEND ON NEXT 5120K MAXSIZE UNLIMITED
 12  DEFAULT TABLESPACE "USERS" LOGGING DATAFILE SIZE 5M
 13      AUTOEXTEND ON NEXT 5M MAXSIZE UNLIMITED
 14  CHARACTER SET WE8MSWIN1252 NATIONAL CHARACTER SET AL16UTF16
 15  LOGFILE GROUP 1 SIZE 1G, GROUP 2 SIZE 1G, GROUP 3 SIZE 1G,
 16      GROUP 4 SIZE 1G, GROUP 5 SIZE 1G, GROUP 6 SIZE 1G
 17  USER SYS IDENTIFIED BY "&&sysPassword"
 18  USER SYSTEM IDENTIFIED BY "&&systemPassword";
Entrez une valeur pour syspassword : Razvanpwd3
Entrez une valeur pour systempassword : Razvanpwd3

Base de données créée.
SQL> select instance_name, name, open_mode from v$database, v$instance;

INSTANCE_NAME      NAME              OPEN_MODE
-----
ambre              AMBRE             READ WRITE
```



L'ordre SQL de création de la base de données comporte également, comme vous pouvez le voir, les ordres des modifications des mots de passe pour les utilisateurs « **SYS** » et « **SYSTEM** ». Vous pouvez remarquer que l'ordre de création de la base de données a créé la base, mais également ouvert cette base de données.

Nous avons à présent une base de données, mais elle est une coquille vide. Pour pouvoir utiliser cette base de données, il faut, maintenant, passer à l'étape suivante : la création du dictionnaire de données.

## La création du dictionnaire

Le dictionnaire de données est un élément important d'une base de données Oracle. Le dictionnaire permet de répertorier tous les objets créés dans la base et leur définition.

Le dictionnaire de données est un ensemble de tables et de vues mis à jour par Oracle. Ce dictionnaire contient toutes les informations sur les données de la base, sur les utilisateurs et l'espace physique.

Après la création de la base de données, il faut exécuter les scripts suivants dans le compte SYS.

| <i>Nom du script</i> | <i>Description</i>                                                                                                                                            |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| catalog.sql          | Création des vues sur les tables de base du dictionnaire<br>Création des vues de performances dynamiques<br>Création des synonymes                            |
| catproc.sql          | Ce script permet d'implémenter les fonctionnalités du langage PL/SQL. Plusieurs packages sont créés pour étendre les fonctionnalités du noyau du SGBD Oracle. |
| catblock.sql         | Crée les vues qui affichent dynamiquement les verrous                                                                                                         |
| cathss.sql           | Installe les packages pour l'administration des services hétérogènes.                                                                                         |
| catio.sql            | Crée les vues du dictionnaire de données pour pouvoir tracer les entrées et les sorties table par table.                                                      |
| cathss.sql           | Installe les packages pour l'administration des services hétérogènes.                                                                                         |
| catqueue.sql         | Crée les vues du dictionnaire de données pour Advanced Queuing.                                                                                               |
| catrep.sql           | Crée les outils nécessaires pour la réplication de la base de données.                                                                                        |
| catrman.sql          | Crée les vues du dictionnaire de données nécessaires pour l'utilitaire Recovery Manager (RMAN).                                                               |
| catocbk.sql          | Crée les outils nécessaires pour Oracle Cryptographic Toolkit.                                                                                                |

Les deux premiers scripts sont les scripts de création du catalogue obligatoire, mais, comme vous pouvez le remarquer, les autres scripts sont également très utiles pour le fonctionnement de la base des données. Dans le tableau précédent, seule a été présentée une sélection des scripts nécessaires pour la création de la base de données.

Tous les scripts de création du catalogue ainsi que les scripts utilisés pour l'administration étendue se retrouvent dans le répertoire :

\$ORACLE\_HOME/rdbms ou %ORACLE\_HOME%\rdbms

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 ce répertoire.

Dans l'exemple suivant on commence par la connexion à la base des données, connexion avec les privilèges étendus « **SYSDBA** ».



```
SQL> spool O:\app\oracle\admin\ambre\scripts\CreateDBCatalog.log append
SQL> @?\rdbms\admin\catalog.sql;
...
SQL> @?\rdbms\admin\catblock.sql;
...
SQL> @?\rdbms\admin\catproc.sql;
...
SQL> @?\rdbms\admin\catoctk.sql;
...
SQL> @?\rdbms\admin\owminst.plb;
...
```

| <i>Convention</i> | <i>Description</i>                                                                               |
|-------------------|--------------------------------------------------------------------------------------------------|
| cat*.sql          | Ils créent des vues du dictionnaire de données et des tables de base du dictionnaire de données. |
| utl*.sql          | Ils créent des vues et des tables additionnelles pour les utilitaires de la base de données.     |
| dbms*.sql         | Ils créent des spécifications de package de base de données.                                     |
| prvt*.plb         | Ils créent le corps de package.                                                                  |

Les scripts de création du dictionnaire de données sont relativement longs ; il faut par conséquent prendre la précaution que toutes les informations affichées puissent être lues par la suite. C'est pourquoi on ouvre un fichier spool dans le répertoire correspondant à l'architecture OFA. Sans le fichier spool il n'est pas possible de vérifier l'exécution de ces scripts.

Les scripts peuvent retourner des messages d'erreur due aux effacements des objets qui n'existent pas. À la fin du script, il ne faut pas oublier de fermer le fichier spool.



```
SQL> select comp_id, status from dba_registry;

COMP_ID                                STATUS
-----                                -
OWM                                     VALID
XDB                                     VALID
CATALOG                                INVALID
CATPROC                                INVALID

SQL> @?\rdbms\admin\utlrp.sql;
...
SQL> select comp_id, status from dba_registry;

COMP_ID                                STATUS
-----                                -
OWM                                     VALID
XDB                                     VALID
CATALOG                                VALID
CATPROC                                VALID
```

### 12c

L'exécution du script « **utlrp.sql** » permet de recompiler tous les codes PL/SQL ou Java qui n'ont pas été compilés. La vue « **DBA\_REGISTRY** » permet de contrôler la nécessité d'exécuter le script « **utlrp.sql** ».

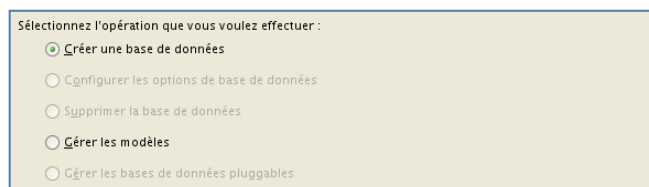
## Les scripts de votre schéma relationnel

La base de données qui a été créée ne contient que le dictionnaire de données. Pour pouvoir travailler avec vos propres applications, il faut créer le schéma relationnel nécessaire pour ces applications.

## L'assistant DBCA

Une fois le logiciel Oracle installé, vous pouvez utiliser DBCA (**DataBase Configuration Assistant**) pour créer et configurer vos bases de données.

L'assistant vous guide dans le processus de création d'une nouvelle base de données, de modification de la configuration d'une base de données existante ou de la suppression d'une base de données. La plupart des tâches de création de base de données que vous exécuteriez normalement manuellement sont exécutées automatiquement lorsque vous avez sélectionné vos options de base de données avec **DBCA (DataBase Configuration Assistant)**.



**12c**

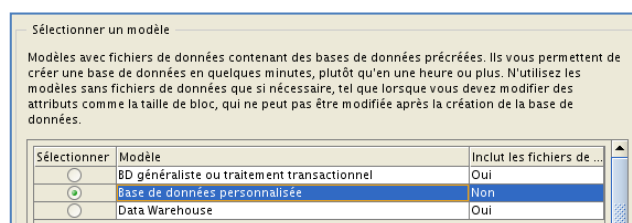
Une base de données consolidée est un type d'architecture mutualisée qui permet d'héberger en même temps plusieurs bases de données différentes. Vous pouvez créer, déconnecter ou supprimer une base de données pluggable. Pour plus d'informations, voir le Module « L'architecture mutualisée ».

L'assistant est identique dans l'environnement Windows et l'environnement Unix/Linux.

## Les modèles

L'option « Créer une base de données » vous guide dans les étapes de création d'une nouvelle base de données ou d'un nouveau modèle.

Lorsque vous sélectionnez un modèle, vous pouvez choisir soit un modèle de base de données personnalisé, soit un modèle avec des fichiers de données. Si vous sélectionnez un modèle de base de données personnalisé (sans fichiers de données), vous pouvez enregistrer les informations de création en tant que script et les utiliser ultérieurement pour créer des bases de données similaires. Il est conseillé de sélectionner un modèle de base de données personnalisé, car ainsi vous avez la possibilité de paramétrer l'ensemble des informations de la base de données.



Dans l'image suivante vous pouvez observer l'ensemble des modèles de votre serveur. Les modèles sont des fichiers XML avec l'extension « **DBT** », database template, stockés dans le répertoire :

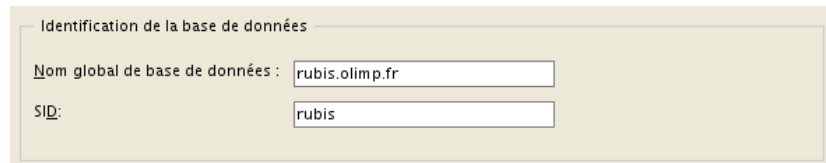
**\$ORACLE\_HOME/assistants/dbca/templates**

Vous pouvez copier le fichier sur un autre serveur pour déployer le modèle, afin de pouvoir créer une base de données avec les mêmes caractéristiques.

Le modèle de base de données personnalisé est le seul modèle qui n'inclut pas les fichiers de données, ce qui vous permet de paramétrer l'ensemble des informations concernant les fichiers de données.

## Le nom de la base

Le pas suivant consiste à choisir le nom de la base de données et le nom de l'instance. L'assistant modifie automatiquement le nom de l'instance pour qu'il soit identique au nom que vous choisissez pour votre base de données. Ce n'est pas pour autant que vous ne pouvez pas modifier le nom de l'instance pour lui attribuer le nom que vous choisissez.



Les paramètres suivants sont renseignés par les informations saisies dans l'exemple précédent :

- le nom de l'instance « **INSTANCE\_NAME** » = rubis
- le nom de la base de données « **DB\_NAME** » = rubis
- le nom unique de la base de données « **DB\_UNIQUE\_NAME** » = « **DB\_NAME** »
- le nom du domaine « **DB\_DOMAIN** » = olimp.fr

Le nom global de la base de données est constitué de la concaténation du nom de la base de données avec le nom du domaine et il est utilisé pour la configuration d'accès à la base.



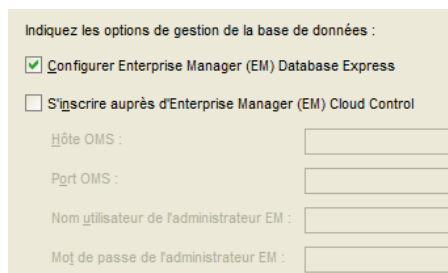
```
SYS@rubis>SELECT NAME, VALUE FROM V$PARAMETER
2  WHERE NAME IN ('db_name','instance_name',
3    'db_unique_name','db_domain')
4  UNION ALL
5  SELECT PROPERTY_NAME, PROPERTY_VALUE
6  FROM DATABASE_PROPERTIES WHERE PROPERTY_NAME = 'GLOBAL_DB_NAME';
```

| NAME           | VALUE          |
|----------------|----------------|
| db_domain      | olimp.fr       |
| instance_name  | rubis          |
| db_name        | rubis          |
| db_unique_name | rubis          |
| GLOBAL_DB_NAME | RUBIS.OLIMP.FR |

## Les options de gestion

DBCA configure la base de données afin qu'elle puisse être gérée par Oracle Database Express ou Oracle Enterprise Manager Cloud Control.

Vous pouvez utiliser OEM Cloud Control pour la gestion des instances de base de données elles-mêmes, ainsi que des outils de gestion centralisée pour la gestion de l'ensemble de l'environnement Oracle : bases de données multiples, hôtes, serveurs d'applications et autres composants du réseau.





## Les informations d'identification

L'étape suivante consiste à définir les mots de passe pour les comptes d'administration de base de données les plus importants. Il est préférable de définir des modes passe différentes pour chaque utilisateur.

Pour des raisons de sécurité, vous devez indiquer des mots de passe pour les comptes utilisateur suivants dans la nouvelle base de données.

☒ Utiliser des mots de passe d'administration différents

| Nom utilisateur | Mot de passe | Confirmer le mot de passe |
|-----------------|--------------|---------------------------|
| SYS             | .....        | .....                     |
| SYSTEM          | .....        | .....                     |

☐ Utiliser le même mot de passe d'administration pour tous les comptes

Mot de passe :

Confirmez le mot de passe :

## La configuration du réseau

Cette étape permet de créer un processus d'écoute pour cette base de données ou d'en utiliser un dans la liste des processus disponibles.

Sélection de processus d'écoute

Les processus d'écoute du répertoire d'origine Oracle Home Grid Infrastructure et du répertoire d'origine Oracle Home Database sont listés ci-dessous. Pour créer un processus d'écoute dans le répertoire d'origine Oracle Home Database, indiquez le nom et le port du processus d'écoute.

Sélectionner des processus d'écoute

| Séle...                             | Nom      | Port | Répertoire d'origine Oracle Home       | Statut |
|-------------------------------------|----------|------|----------------------------------------|--------|
| <input checked="" type="checkbox"/> | LISTENER | 1521 | /u01/app/oracle/product/12.1.0/db_home | Haut   |
| <input type="checkbox"/>            |          |      | /u01/app/oracle/product/12.1.0/db_home |        |

## Les options de stockage

L'étape suivante consiste à choisir le système de stockage que vous souhaitez utiliser pour vos fichiers de base de données. Il s'agit de trois types de stockage.

### Systeme de fichiers

Cette option gère vos fichiers de base de données à instance unique dans un répertoire de votre système de fichiers actuel. Par défaut, DBCA sauvegarde les fichiers de base de données en utilisant l'architecture OFA, dans laquelle les fichiers de base de données et les fichiers administratifs, fichiers d'initialisation compris, respectent la convention standard de choix de nom et d'emplacement.

Vous pouvez modifier les valeurs par défaut plus tard, lors de la configuration de la base de données, via la page Stockage de base de données de DBCA.

### Automatic Storage Management (ASM)

ASM est un gestionnaire de volume propriétaire qui simplifie l'administration des fichiers de base de données. Au lieu de gérer des fichiers de base de données, vous ne gérez qu'un petit nombre de groupes de disques.

Un groupe de disques est un ensemble d'unités de disques qui sont gérés par ASM comme une seule unité logique. Si vous définissez un groupe de disques particulier comme groupe de disques par défaut pour une base de données, Oracle alloue automatiquement de la mémoire pour les fichiers associés à l'objet de base de données, et crée ou supprime automatiquement ces fichiers. Lorsque vous administrez la base de données, il vous suffit de faire référence aux objets de la base en indiquant leur nom, et non le nom du fichier.

## Les emplacements des fichiers

Vous avez la possibilité de vérifier les variables d'environnement et de créer vos propres variables, indiquez leur nom et leur valeur. Les variables vous permettent de centraliser des informations de données et de créer des modèles de base de données avec des paramètres généraux.

| Variable            | Valeur                                 |
|---------------------|----------------------------------------|
| ORACLE_BASE         | /u01/app/oracle                        |
| ORACLE_HOME         | /u01/app/oracle/product/12.1.0/db_home |
| DB_NAME             | rubis                                  |
| DB_UNIQUE_NAME      | rubis                                  |
| SID                 | rubis                                  |
| PDB_NAME            |                                        |
| RUBIS_DONNEES       | +GD_DONNEES                            |
| RUBIS_RECUPERATIONS | +GD_RECUPERATIONS                      |
| RUBIS_ARCHIVES      | +GD_ARCHIVES                           |

L'étape suivante consiste à choisir la gestion des fichiers de données de la base. Vous avez trois possibilités :

## L'emplacement des fichiers du modèle

Cette option vous permet d'utiliser l'emplacement défini par le modèle de base de données que vous avez sélectionné pour cette base. Vous avez la possibilité de revoir et de modifier les noms et les emplacements de fichiers de base de données plus tard dans l'assistant.

## Un emplacement commun pour tous les fichiers

Sélectionnez cette option pour spécifier une nouvelle zone commune pour tous vos fichiers de base de données. Vous avez la possibilité de revoir et de modifier les noms et les emplacements de fichiers de base de données plus tard dans l'assistant.

## Utiliser les fichiers gérés par Oracle

Oracle Managed Files (OMF) vous permet d'éviter, en tant qu'administrateur de base de données, de gérer directement les fichiers de système d'exploitation qui composent une base de données Oracle. Vous spécifiez les opérations en termes d'objets de base de données, et non de noms de fichiers. Oracle utilise en interne des interfaces de système de fichiers standards pour créer et supprimer des fichiers en fonction des besoins des tablespaces, des fichiers de journalisation et des fichiers de contrôle.

Le répertoire saisi dans *Emplacement des fichiers de base de données* est stocké dans le paramètre « **DB\_CREATE\_FILE\_DEST** » qui est utilisé pour stocker tous les fichiers de la base de données. Chaque fois que vous voulez créer un fichier de la base de données Oracle crée automatiquement dans ce répertoire un sous-répertoire avec le nom du paramètre « **DB\_UNIQUE\_NAME** » s'il n'existe pas déjà. Ensuite pour stocker le fichier il crée un autre sous-répertoire suivant le type du fichier de

la base de données que vous voulez créer « **CONTROLFILE** » pour les fichiers de contrôles, « **DATAFILE** » pour les fichiers de données ou « **ONLINELOG** » pour les fichiers de journaux.

Vous pouvez également multiplexer les fichiers de journalisation et les fichiers de contrôle. Lorsque vous sélectionnez cette option, vous pouvez identifier les emplacements où peuvent être stockés les doubles des fichiers. Le multiplexage offre une plus grande tolérance de panne pour le fichier de journalisation et le fichier de contrôle si l'une des destinations échoue. Pour plus d'informations voir les modules correspondants.



### Attention

Si vous initialisez uniquement le paramètre « **DB\_CREATE\_FILE\_DEST** » Oracle multiplexe automatiquement les fichiers de contrôles et les fichiers de journaux en créant une copie dans le répertoire de défaut dans la zone de récupération rapide si elle a été définie. Le fonctionnement de la zone de récupération rapide est décrit dans l'étape suivante.

Pour ne pas avoir un multiplexage automatique et pouvoir définir les répertoires dans lesquels sont créés les doubles des fichiers, vous devez les renseigner. Ainsi les fichiers de contrôles et les fichiers de journaux sont positionnés dans les emplacements saisis.

Les paramètres suivants sont renseignés par les informations saisies dans cette étape :

- « **DB\_CREATE\_FILE\_DEST** » = {RUBIS\_DONNEES}
- « **DB\_CREATE\_ONLINE\_LOG\_DEST\_1** » = {RUBIS\_RECUPERATIONS}
- « **DB\_CREATE\_ONLINE\_LOG\_DEST\_2** » = {RUBIS\_DONNEES}



```
SYS@rubis>show parameter db_create
```

| NAME                        | TYPE   | VALUE             |
|-----------------------------|--------|-------------------|
| db_create_file_dest         | string | +GD_DONNEES       |
| db_create_online_log_dest_1 | string | +GD_RECUPERATIONS |
| db_create_online_log_dest_2 | string | +GD_DONNEES       |
| db_create_online_log_dest_3 | string |                   |
| db_create_online_log_dest_4 | string |                   |
| db_create_online_log_dest_5 | string |                   |

```
SYS@rubis>select name from v$controlfile union all
2 select name from v$datafile union all
3 select member from v$logfile;
```

NAME

```
+GD_RECUPERATIONS/RUBIS/CONTROLFILE/current.261.840669157
+GD_DONNEES/RUBIS/CONTROLFILE/current.270.840669157
+GD_DONNEES/RUBIS/DATAFILE/system.256.840664649
+GD_DONNEES/RUBIS/DATAFILE/sysaux.257.840664683
+GD_DONNEES/RUBIS/DATAFILE/undotbs1.258.840664697
+GD_DONNEES/RUBIS/DATAFILE/users.260.840664725
+GD_RECUPERATIONS/RUBIS/ONLINELOG/group_1.258.840668719
+GD_DONNEES/RUBIS/ONLINELOG/group_1.262.840668721
```

```
+GD_RECUPERATIONS/RUBIS/ONLINELOG/group_2.262.840668823
+GD_DONNEES/RUBIS/ONLINELOG/group_2.263.840668825
+GD_RECUPERATIONS/RUBIS/ONLINELOG/group_3.257.840668731
+GD_DONNEES/RUBIS/ONLINELOG/group_3.264.840668733
...
```

Le fichier de contrôle ainsi que les fichiers de journaux ont été créés dans l'arborescence correspondante aux deux répertoires précisés dans les paramètres « **DB\_CREATE\_ONLINE\_LOG\_DEST\_n** ». Les fichiers de données sont eux créés dans l'arborescence correspondante aux deux répertoires précisés dans le paramètre « **DB\_CREATE\_FILE\_DEST** ».

Contrairement aux autres options de la page, celle-ci ne permet pas de modifier ultérieurement les noms des fichiers de base de données dans la page Stockage, disponible après dans l'assistant Configuration de base de données.

### La configuration de la récupération

Vous pouvez configurer la base de données de manière à pouvoir récupérer vos données en cas de défaillance du système.

La zone de récupération rapide permet de récupérer des données qui, autrement, auraient été perdues en cas de défaillance du système. Elle est également utilisée par Enterprise Manager si vous avez sélectionné la gestion locale et les sauvegardes quotidiennes dans la page Options de gestion de **DBCA** (DataBase Configuration Assistant), qui s'est affichée précédemment.

La zone de récupération rapide est identifiée par le paramètre « **DB\_RECOVERY\_FILE\_DEST** » mais également par le paramètre « **DB\_RECOVERY\_FILE\_DEST\_SIZE** » qui détermine la taille maximale des de stockage dans ce répertoire.

### Attention



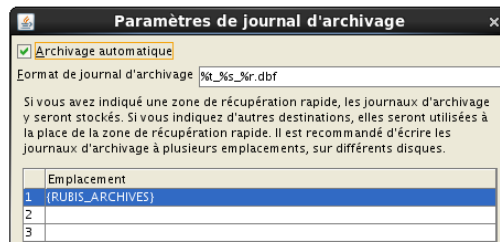
La zone de récupération rapide contient le multiplexage automatique des fichiers de contrôle et des journaux si vous n'avez pas initialisé un des paramètres « **DB\_CREATE\_ONLINE\_LOG\_DEST\_n** ».

Elle est également l'emplacement de stockage par défaut des fichiers des journaux archivés de la base de données.

Les sauvegardes quotidiennes automatiques de la base de données, si vous avez sélectionné dans la page Options de gestion de **DBCA**, sont également stockées.

Les journaux flashback si l'option « **FLASHBACK** » a été configuré. Mais le répertoire ne peut stocker que maximum la taille précise dans le paramètre « **DB\_RECOVERY\_FILE\_DEST\_SIZE** ». Une fois que la base de données a rempli cette taille aucune information ne peut plus être écrite dans ce répertoire malgré l'espace libre qui peut se trouver sur disque. Ces informations stockées dans ce répertoire sont vitales pour votre base de données ; il faut faire en sorte que l'espace ne manque pas.

Vous devez activer l'archivage pour que la base de données puisse être récupérée en cas de défaillance de disque. Pour plus d'informations sur l'archivage et notamment le format de fichiers archives voir le Module « Les fichiers journaux ».



Vous pouvez activer l'archivage, la base de données archive ses fichiers de journalisation. Les fichiers de journalisation archivés peuvent être utilisés pour récupérer une base de données, et mettre à jour une base de données en attente.

## Le contenu de la base

L'assistant de configuration de base de données vous permet de sélectionner les fonctions de base de données que vous voulez utiliser pour votre base Oracle.

### Oracle JVM

Le produit Oracle JVM (Java Virtual Machine) offre les fonctionnalités clés vous permettant de développer des applications Java avec la base de données Oracle.

### Oracle Text

Le produit Oracle Text permet de gérer du contenu multimédia, à la fois pour les applications Internet et pour les applications classiques qui doivent accéder à des informations audio, vidéo, à des images, du texte ou des données d'emplacement.

### Oracle Multimedia

Le produit Oracle Multimedia est une fonction qui permet à Oracle de stocker, de gérer et d'extraire des informations d'emplacements géographiques, des images, des sons, des vidéos ou d'autres données de supports hétérogènes de manière intégrée avec d'autres informations d'entreprise.

### Oracle OLAP

Les produits OLAP (Online Analytical Processing) offrent des services qui prennent en charge les calculs statistiques, mathématiques et financiers complexes dans un modèle de données multidimensionnel.

### Oracle Spatial

Le produit Oracle Spatial est essentiellement utilisé pour les systèmes GIS (Geographical Information Systems). Les applications GIS peuvent comporter des opérations de saisie de données, de modification de données, de création de cartes, de traitement d'images, de classement d'images, de conversion de données, de migration de données, d'interrogation de données, de création de rapports et d'analyse de données.

### Oracle Label Security

Le produit Oracle Label Security repose sur les concepts de labels utilisés par les organisations officielles et de défense pour protéger les informations sensibles et permettre la séparation des données.

### Oracle Application Express

Il s'agit d'un environnement de développement déclaratif hébergé permettant de développer et de déployer des applications web centrées sur une base de données. Le moteur Oracle Application Express affiche les applications en temps réel à partir des données stockées dans les tables de base de données.

## Oracle Database Vault

Le produit vous aide à résoudre les problèmes liés à la sécurité, tels que la protection contre les menaces internes, le respect des exigences relatives à la conformité aux réglementations en vigueur et l'imposition de la séparation des tâches. Il fournit plusieurs fonctionnalités flexibles permettant d'appliquer des contrôles d'accès fins à vos données sensibles.

## Les paramètres mémoire

L'assistant de configuration de base de données vous permet de définir les paramètres d'initialisation qui contrôlent la façon dont la base de données gère l'utilisation de sa mémoire.



La gestion automatique de la mémoire consiste à définir la taille globale de la mémoire (SGA et PGA) ensuite l'instance gère automatiquement la taille allouée pour l'ensemble des PGA et la taille allouée pour toutes les zones mémoire du SGA. En fonction de cette valeur, l'instance affecte automatiquement aux structures de mémoire de la base de données la quantité de mémoire la plus efficace suivant l'utilisation et le chargement de la base.

Pour que l'instance Oracle puisse gérer automatiquement sa mémoire, il suffit de définir le paramètre « **MEMORY\_TARGET** ».

Le paramètre « **MEMORY\_TARGET** » est dynamique ; vous pouvez le modifier à tout moment, sans avoir à redémarrer la base de données. Pour plus de détails voir le Module « La gestion de la mémoire ».

Vous pouvez sélectionner la gestion de la mémoire « Personnalisée » pour allouer des quantités spécifiques de mémoire pour l'ensemble des PGA et pour toutes les zones mémoire dynamiques du SGA. En fonction de la valeur allouée pour SGA, l'instance affecte automatiquement aux structures de mémoire dynamiques la quantité de mémoire la plus efficace suivant l'utilisation et le chargement de la base.

Les paramètres pour la gestion automatique personnalisée de mémoire sont : « **SGA\_TARGET** » pour l'ensemble des zones mémoire dynamiques du SGA et pour l'ensemble des zones PGA des utilisateurs « **PGAAggregateTarget** ». Pour plus de détails sur la gestion voir le Module « La gestion de la mémoire ».

Pour personnaliser la répartition de la mémoire SGA entre les sous-structures de mémoire SGA, sélectionnez Manuelle et entrez des valeurs spécifiques pour chaque sous-composant SGA.

## Attention

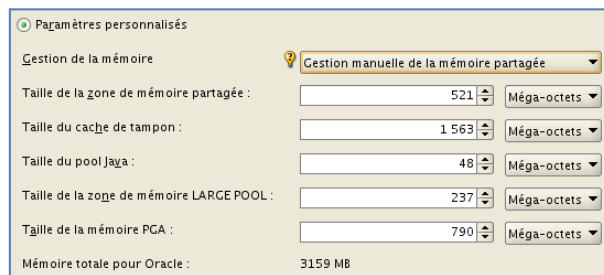
La somme de l'ensemble des zones mémoire est mentionnée dans la description « Mémoire totale pour Oracle ». Il faut se rappeler le fonctionnement des différentes zones mémoire, elles sont utilisées comme des tampons mémoire pour accélérer la lecture et écriture des informations dans les fichiers de données.





Toutes les données de la base avant d'être récupérées par un utilisateur sont chargées en mémoire. En effet si les zones mémoire correspondantes ne se retrouvent pas dans la mémoire vive de la machine, mais en swap, le serveur va effectuer une lecture et une écriture sur disque à la place d'une lecture et d'une écriture dans la mémoire vive.

Avant de créer la base de données il faut s'assurer qu'on dispose d'un espace mémoire suffisant pour stocker nos zones mémoire.



### Pool partagé

Vous devez entrer la taille du paramètre « **SHARED\_POOL\_SIZE** », en octets, ou acceptez la valeur par défaut. Les valeurs élevées améliorent les performances des systèmes multi-utilisateurs. Les valeurs faibles utilisent moins de mémoire.

### Cache de tampons

Vous devez entrer la taille du paramètre « **DB\_CACHE\_SIZE** », en octets. Tous les processus utilisateur connectés simultanément à l'instance partagent l'accès au cache de tampons de la base de données.

### Pool Java

Vous devez entrer la taille du paramètre « **JAVA\_POOL\_SIZE** ». La zone mémoire est utilisée dans la mémoire du serveur pour tout le code Java propre aux sessions et toutes les données de la JVM.

### Large Pool

Vous devez entrer la taille du paramètre « **LARGE\_POOL\_SIZE** », en octets. La zone mémoire est utilisée par le serveur partagé pour la mémoire de session du serveur partagé, par l'exécution parallèle pour les tampons de messages et par la sauvegarde pour les tampons d'entrées-sorties disque.

### Attention

Si « **LARGE\_POOL\_SIZE** » n'est pas affectée et si le pool doit être utilisé pour une exécution en parallèle, Oracle calculera automatiquement une valeur. Le calcul par défaut peut donner une taille trop importante à affecter ou causer des problèmes de performances.



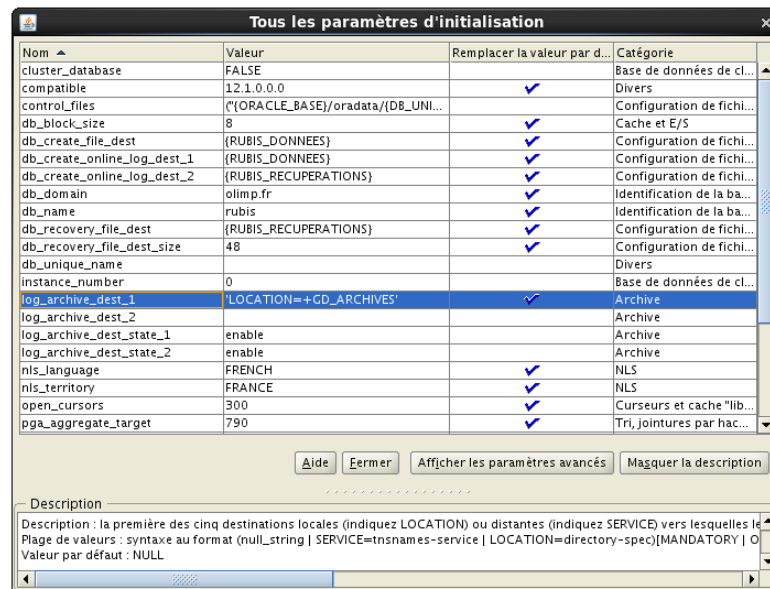
## Les paramètres

La boîte de dialogue « Tous les paramètres d'initialisation » vous permet d'afficher et de modifier tous les paramètres de configuration de la base de données.

Vous pouvez choisir les paramètres d'initialisation de base. Ceux sont les paramètres d'initialisation les plus courants qui sont le plus souvent modifiés ou ajustés par les administrateurs de base de données.

Vous pouvez également afficher une fenêtre qui décrit brièvement le paramètre sélectionné.

Tous les paramètres d'initialisation peuvent être également retrouvés à partir de la vue dynamique « **V\$PRAMETER** ». Pour plus d'informations sur la vue voir le Module « La gestion d'une instance ».



## Le dimensionnement de la base

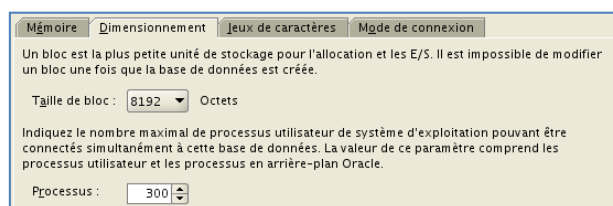
### La taille de bloc

Le paramètre définissant la taille du bloc Oracle est le paramètre « **DB\_BLOCK\_SIZE** ». Une fois la base de données créée, la valeur du paramètre « **DB\_BLOCK\_SIZE** », ne peut plus être modifiée.

Les valeurs du bloc standard ou non standard doivent être choisies parmi la liste suivante : 2Ko, 4Ko, 8Ko, 16Ko et 32Ko. La taille de block de 32Ko est accessible uniquement sur les systèmes d'exploitation 64 bits.

### Les Processus

Indique le nombre maximum de processus utilisateur du système d'exploitation pouvant se connecter simultanément à Oracle. La valeur doit être suffisante pour tous les processus d'arrière-plan, les processus des travaux et les processus d'exécution parallèle. Le paramètre définissant le nombre des processus utilisateur est « **PROCESSUS** ». Pour voir tous les processus qui s'exécutent sur un serveur vous pouvez interroger la vue « **V\$PROCESS** ».



## Les jeux de caractères

Les jeux de caractères sont les encodages utilisés pour l'affichage des caractères. Le choix d'un jeu de caractères détermine les langues pouvant être représentées dans la base.

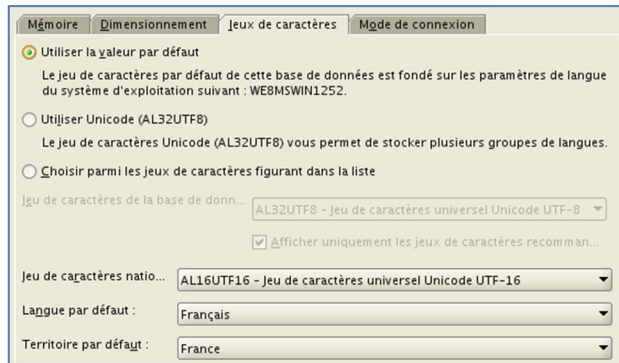
### Le jeu de caractères de la base de données

La base de données Oracle utilise le jeu de caractères de base de données pour les éléments suivants :

- Les données stockées avec des types alphanumériques « CHAR », « VARCHAR2 », « CLOB » et « LONG ».
- Les identificateurs tels que les noms de table, noms de colonne et variables PL/SQL.
- L'entrée et le stockage du code source de programme SQL et PL/SQL.



Les jeux supportés et la valeur par défaut de ce paramètre dépendent de votre système d'exploitation.



### ***Le jeu de caractères national***

Un jeu de caractères national est un autre jeu de caractères vous permettant de stocker des caractères Unicode dans une base dont le jeu de caractères de base de données n'est pas Unicode. Le choix d'un jeu de caractères national peut également être motivé par les raisons suivantes :

- Les propriétés d'un autre encodage de caractères peuvent mieux convenir à des opérations lourdes de traitement de caractères.
- La programmation dans le jeu de caractères national est plus facile.

Le jeu de caractères national utilisé pour stocker les données dans des colonnes définies spécifiquement avec les types « **NCHAR** », « **NCLOB** » ou « **NVARCHAR2** ».

### ***Langue par défaut***

La langue par défaut choisie détermine la façon dont la base de données prend en charge les informations locales, par exemple :

- Les noms de jour et de mois avec leurs abréviations.
- Ordre de tri par défaut des données alphanumériques lorsque la clause « **ORDER BY** » est indiquée.
- Sens d'écriture (vers la gauche ou vers la droite).

Le paramètre qui détermine la langue par défaut est « **NLS\_LANGUAGE** » pour prendre en compte la configuration française la valeur est « **FRENCH** ».

### ***Le territoire par défaut***

Les diverses conventions d'affichage pour la numérotation des jours et des semaines, le format de date par défaut, le séparateur décimal et le séparateur de groupes par défaut, le symbole ISO par défaut et les symboles monétaires locaux par défaut.

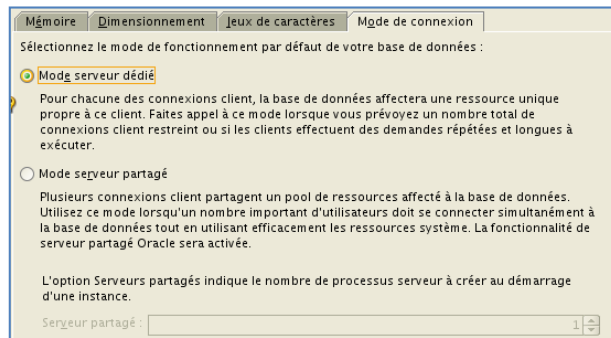
Le paramètre qui détermine le territoire par défaut est « **NLS\_TERRITORY** » pour prendre en compte la configuration française la valeur est « **FRANCE** ».

## **Le mode de connexion**

L'assistant de configuration de base de données vous permet de définir le mode de connexion de la base de données.

### ***Mode serveur dédié***

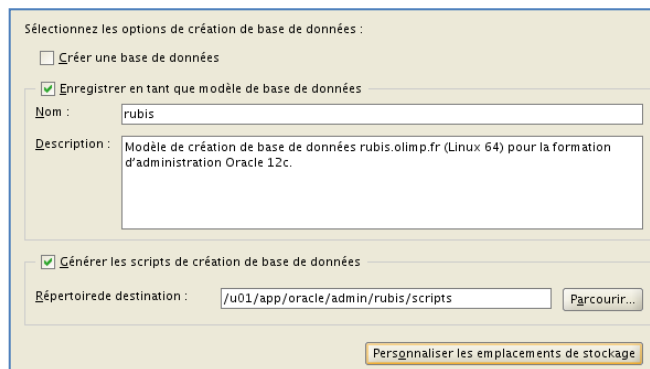
Une base de données Oracle en mode serveur dédié demande un processus serveur dédié pour chaque processus utilisateur. Oracle Net renvoie au client l'adresse d'un processus serveur existant. Le client renvoie alors sa demande de connexion à l'adresse de serveur fournie.



### Mode serveur partagé

Une base de données en mode serveur partagé est configurée pour qu'un grand nombre de processus utilisateur partage un petit nombre de processus serveur, de façon à démultiplier le nombre d'utilisateurs pris en charge.

En configuration Serveur dédié, de nombreux processus utilisateur se connectent à un répartiteur. Le répartiteur transmet plusieurs demandes de session réseau entrantes à une file d'attente commune. Un processus de serveur partagé inactif dans un pool partagé de processus serveur sélectionne une demande dans la file d'attente. Un petit pool de processus serveur peut donc desservir un grand nombre de clients.



## Les options de création

L'assistant de configuration de base de données est maintenant fini, vous pouvez à présent choisir ce que vous voulez faire de l'ensemble des options sélectionnées.

### Créer une base de données

Vous devez sélectionner cette option pour créer la base de données, à partir de l'assistant.

### Enregistrer en tant que modèle de base de données

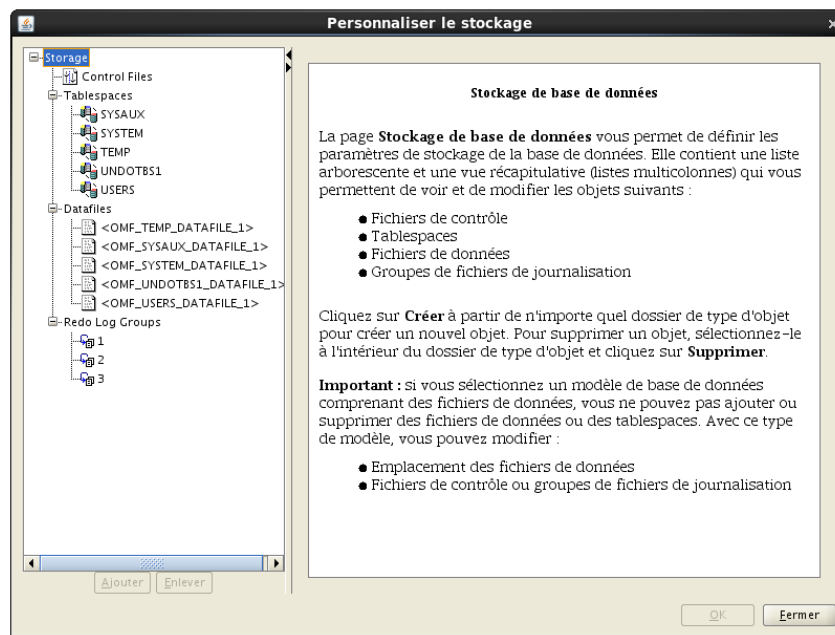
Vous sélectionnez cette option pour enregistrer les paramètres de création de base de données en tant que modèle. Ce modèle est automatiquement ajouté à la liste des modèles de base de données disponibles.

### Générer les scripts de création de base de données

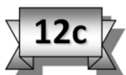
Le choix de cette option vous permet de créer les scripts SQL ainsi que les scripts des commandes pour créer cette base de données avec les options que vous avez choisies.

### La gestion du stockage

L'assistant de configuration de base de données vous permet de spécifier les paramètres de stockage de la base de données.



La fin de l'assistant est matérialisé par une fenêtre récapitulative comprenant les informations sur les paramètres d'initialisation, les fichiers de données, les fichiers de contrôle et les groupes de fichiers de journalisation. Il s'agit des paramètres qui seront utilisés lorsque **DBCA (DataBase Configuration Assistant)** crée la nouvelle base de données au moyen de ce modèle.



Dans le cas où vous avez besoin de corriger des paramètres ou la configuration de la base de données, vous pouvez choisir dans l'arborescence à gauche de la fenêtre, l'étape à laquelle vous voulez revenir.

- *Container Database*
- *Pluggable database*
- *CDB\_*
- *V\$PDBS*
- *SQL Developer*

# 6

## ***L'architecture mutualisée***



### **Objectifs**

À la fin de ce module, vous serez à même d'effectuer les tâches suivantes :

- Décrire les concepts de l'architecture mutualisée.
- Énumérer les composants constitutifs d'une architecture mutualisée.
- Créer une base de données conteneur et des bases de données insérées.
- Arrêter et démarrer l'ensemble de l'architecture.
- Créer une base de données insérée à partir du modèle.
- Cloner une base de données insérée dans le même conteneur ou dans un autre conteneur mutualisé.
- Se connecter aux bases des données insérées en vue de leur maintenance.
- Paramétrer et configurer séparément les bases de données insérées.

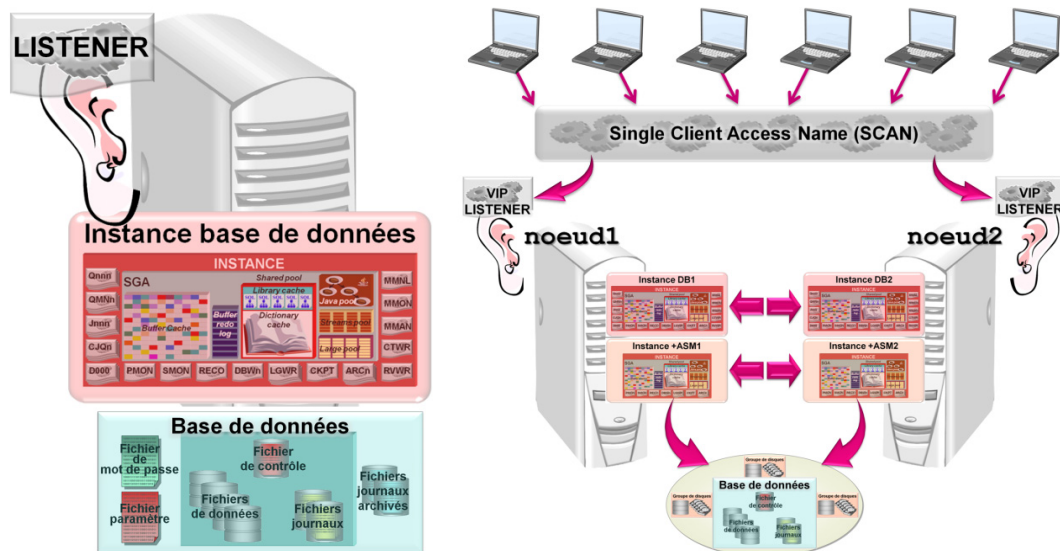


### **Contenu**

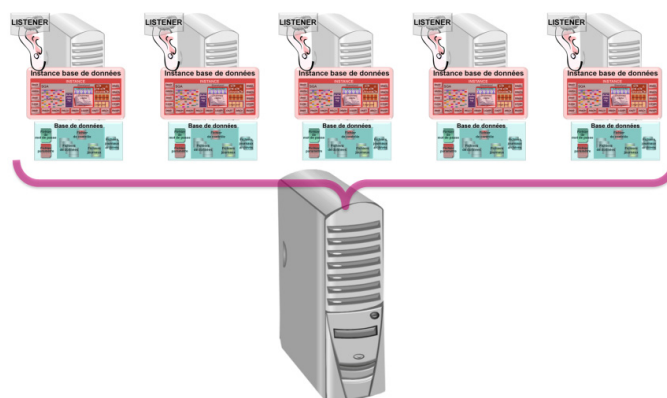
|                                |     |                                 |      |
|--------------------------------|-----|---------------------------------|------|
| L'architecture Oracle 11g      | 6-2 | Le clonage d'une base insérée   | 6-11 |
| L'architecture mutualisée      | 6-3 | L'effacement d'une base insérée | 6-14 |
| La création d'un conteneur CDB | 6-4 | La gestion d'une base insérée   | 6-14 |
| La connexion aux CDB et PDB    | 6-7 | L'extraction et l'insertion     | 6-16 |
| L'arrêt et le démarrage        | 6-8 | SQL Developer                   | 6-19 |
| La création d'une PDB          | 6-9 |                                 |      |

# L'architecture Oracle 11g

Avant la version Oracle 12c, il existait deux types d'architectures possibles pour une base de données. La première architecture est traitée dans le Module « L'architecture d'Oracle » ; elle décrit qu'une instance ouvre une seule base de données et qu'une base de données est accessible par une seule instance à la fois.



La deuxième architecture possible est Oracle Real Application Cluster, qui consiste à mutualiser les ressources de plusieurs serveurs pour travailler avec une base de données. Ainsi vous avez plusieurs instances déployées sur plusieurs serveurs mais une base de données unique est mise à jour à travers ces instances. Dans un cas comme dans l'autre, pour des raisons applicatives et/ou de sécurité, on est obligé de multiplier le nombre de bases de données. La première raison est la sécurité, car dans le cas où plusieurs clients ou plusieurs services d'une même société sont hébergés sur la même base de données, les applications peuvent provoquer des conflits car elles ont créé des schémas identiques, des synonymes publics, des liens de bases de données publiques, etc.



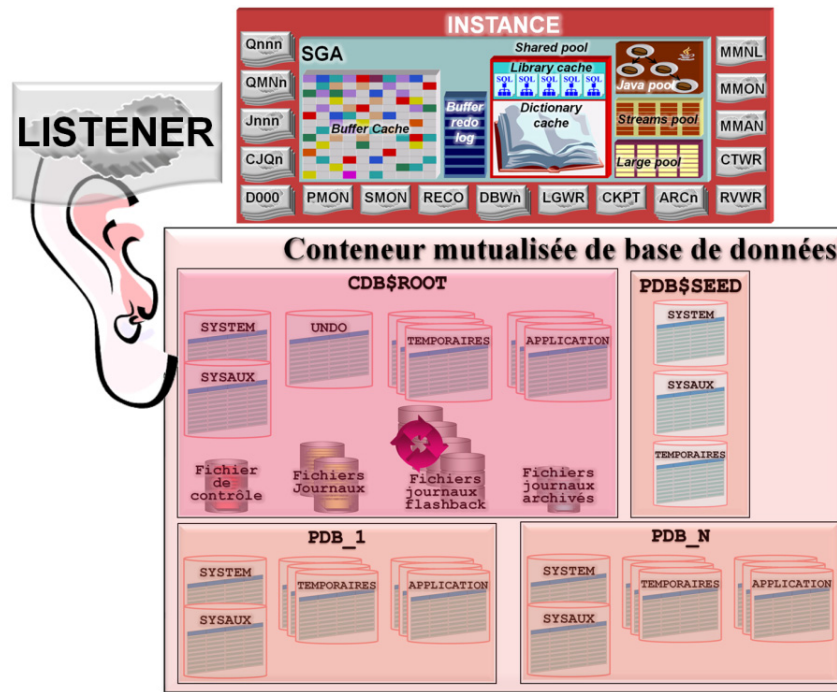
Les multiplications de bases de données posent plusieurs problèmes d'administration, de coût et de disponibilité. Voici les plus grands inconvénients de ce type d'architecture :

- Le gaspillage des ressources, de mémoire, des processeurs, etc.
- L'administration de plusieurs bases de données, l'arrêt et le redémarrage des serveurs, la sauvegarde séparée de chaque serveur, les mises à jour et l'audit de chaque base de données effectués séparément.
- La gestion des bases de données de secours doit être mise en œuvre pour chaque base à la fois et pour les architectures de type clusters RAC.

# L'architecture mutualisée

12c

À partir de la version Oracle 12c, vous avez la possibilité de mettre en œuvre une architecture mutualisée pour consolider plusieurs bases de données en une seule, accessible par une instance unique.



## Container Database CDB

La pièce maîtresse de cette architecture est un nouveau type de base de données, appelé conteneur (Container Database ou **CDB**). La base de données conteneur mutualisé permet de loger jusqu'à 252 données insérées (Pluggable Database **PDB**). Une base de données standard ne peut pas être convertie en base de données de type conteneur mutualisé.

Chaque base de données conteneur mutualisé CDB contient les éléments suivants :

- La base de données maître connue sous le nom de conteneur racine « **CDB\$ROOT** ». Cette base contient le dictionnaire de données maître qui englobe certaines informations sur les bases de données insérées PDB.
- Le modèle de type base de données insérée PDB, appelé « **PDB\$SEED** », utilisé pour la création d'autres bases de données insérées PDB.
- L'ensemble des bases jusqu'à 252 bases de données insérées PDB. Chaque base de données insérée PDB est autonome et fonctionne comme une base de données standard.

12c

À partir de la version Oracle 12c, pour toutes les vues du dictionnaire de données en plus des préfixes « **DBA** », « **ALL** » et « **USERS** » il existe un nouveau préfixe : « **CDB** ». Ces vues permettent d'interroger les informations de la base de données conteneur racine « **CDB\$ROOT** », de la base de données modèle insérée « **PDB\$SEED** », ainsi que de toutes les autres bases de données insérées.

Beaucoup d'autres vues du dictionnaire de données contiennent une nouvelle colonne « **CONN\_ID** », qui représente l'identifiant unique de la base de données du conteneur. Ainsi chaque base de données de l'architecture a un identifiant unique qui a la valeur : 1 pour la base de données conteneur racine, 2 pour la base de données modèle « **PDB\$SEED** », et ensuite dans l'ordre de la séquence toutes les bases de données insérées. Une autre colonne « **ISPDB\_MODIFIABLE** » est introduite dans la vue dynamique « **V\$PARAMETER** », elle permet de savoir si un paramètre est modifiable uniquement pour une base de données insérée PDB.





```
SYS@gemmes>SELECT PDB_ID, CON_ID, DBID, PDB_NAME, STATUS, USERNAME
2 FROM CDB_PDBS JOIN CDB_USERS USING (CON_ID) WHERE USERNAME = 'SYSTEM';
```

| PDB_ID | CON_ID | DBID       | PDB_NAME  | STATUS | USERNAME |
|--------|--------|------------|-----------|--------|----------|
| 2      | 1      | 4081142505 | PDB\$SEED | NORMAL | SYSTEM   |
| 3      | 1      | 2890003885 | SAPHIR    | NORMAL | SYSTEM   |
| 4      | 1      | 2134453907 | TOPAZE    | NORMAL | SYSTEM   |
| 5      | 1      | 1011669824 | AMBRE     | NORMAL | SYSTEM   |
| 6      | 1      | 2134436100 | JASPE     | NORMAL | SYSTEM   |

### Pluggable Database PDB

La base de données insérée PDB est autonome et fonctionne du point de vue d'un utilisateur comme une base de données normale ; un ensemble de fichiers de données correspondant à ses propres tablespaces, une collection de schémas, de tables et d'autres métadonnées. Lorsqu'un utilisateur est connecté à une base de données insérée PDB, il n'a pas de visibilité sur le conteneur racine CDB ou d'autres bases de données insérées PDB. À la création, chaque base de données insérée PDB possède son propre tablespace « **SYSTEM** », « **SYSAUX** » et « **TEMP** ». Ainsi chacune possède son propre dictionnaire ; dans ce cas pas de conflits entre les schémas, liens de base de données et synonymes publics. Posséder son propre dictionnaire, c'est intéressant également pour des raisons de sécurité, ainsi chaque base de données insérée PDB peut gérer directement la sienne. Les bases de données insérées PDB peuvent avoir des jeux de caractères et des zones horaires distinctes et elles ne peuvent être accessibles qu'à travers une connexion réseau.

L'arrêt et le démarrage de ces bases peuvent être opérés indépendamment l'un de l'autre mais bien sûr l'instance ne peut pas être arrêtée car elle est utilisée par les autres bases de données de l'architecture. Mettre hors ligne une base de données insérée sans incidence sur les autres bases permet une meilleure maintenance.

Toutes les informations concernant l'instance (unique) se retrouvent gérées directement par le conteneur racine, malheureusement il n'y a qu'un seul fichier d'alerte pour toute l'architecture. De même il existe un seul tablespace « **UNDO** » et les fichiers de contrôle et les journaux sont uniques pour l'ensemble de l'architecture.

Dans cette architecture il est très facile de créer une nouvelle base de données insérée par clonage du modèle « **PDB\$SEED** », ou d'une autre base de données insérée déjà existante. Il est également possible de retirer une base de données insérée PDB d'un conteneur mutualisé CDB et de l'insérer dans un autre.

L'architecture mutualisée permet une gestion centralisée de l'ensemble des bases de données du conteneur, tout en respectant la séparation de chacune. Ainsi du point de vue de l'administrateur, il n'y a qu'une seule base de données à sauvegarder, l'ensemble des bases de données insérées sont automatiquement sauvegardées avec le conteneur. Cette vision de base de données unique est très intéressante pour la mise en œuvre de l'architecture base de données de secours ou celle de type clusters RAC.

## La création d'un conteneur CDB

L'architecture mutualisée nécessite la création dans une base de données d'un conteneur racine « **CDB\$ROOT** » ainsi qu'un modèle de type base de données insérée « **PDB\$SEED** », utilisé pour la création d'autres bases de données insérées PDB.

Pour créer une base de données, vous devez fournir un ensemble d'informations afin de renseigner les caractéristiques de l'architecture utilisant la syntaxe suivante :

```
CREATE DATABASE nom ...
ENABLE PLUGGABLE DATABASE
```

```
[ SEED
    FILE_NAME_CONVERT = { ( 'modèle01', 'modèle02'[,...] ) | NONE }
[ SYSTEM DATAFILE [fichier ...]]
[ SYSAUX DATAFILE [fichier ...]]
[ BIGFILE|SMALLFILE ] USER_DATA TABLESPACE nom DATAFILE[fichier ...]

USER_DATA          Le nom et les caractéristiques du tablespace permanent par
                    défaut des bases de données insérées.

modèle01          Le modèle représente une partie du nom des fichiers de données
                    qui est remplacée par le deuxième modèle.

FILE_NAME_CONVERT La conversion des noms des fichiers de données entre la base de
                    données conteneur mutualisé CDB et les bases de données
                    insérées PDB. L'argument est utilisé uniquement dans le cas où
                    vous n'utilisez pas Oracle Managed Files.
```

Dans un premier temps, voici un exemple de création d'une base de données conteneur mutualisé CDB, en laissant Oracle gérer les noms des fichiers (Oracle Managed Files).



```
SYS@gemmes> select i.instance_name instance, i.status, p.name, value
  2  from v$parameter p, v$instance i where p.name in
  3  ('db_create_file_dest','db_create_online_log_dest_1');

INSTANCE STATUS   NAME                                VALUE
-----
Gemmes   STARTED   db_create_file_dest                /u02/donnees/oradata
Gemmes   STARTED   db_create_online_log_dest_1      /u03/recuperations/oradata

SYS@gemmes> CREATE DATABASE "gemmes"
  2      MAXDATAFILES 1024
  3  DATAFILE SIZE 700M AUTOEXTEND ON NEXT 10M
  4      MAXSIZE UNLIMITED EXTENT MANAGEMENT LOCAL
  5  SYSAUX DATAFILE SIZE 550M AUTOEXTEND ON NEXT 10M
  6      MAXSIZE UNLIMITED
  7  DEFAULT TEMPORARY TABLESPACE TEMP TEMPFILE SIZE 20M
  8      AUTOEXTEND ON NEXT 640K MAXSIZE UNLIMITED
  9  UNDO TABLESPACE UNDOTBS1 DATAFILE SIZE 200M
 10      AUTOEXTEND ON NEXT 5M MAXSIZE UNLIMITED
 11  DEFAULT TABLESPACE USERS LOGGING DATAFILE SIZE 5M
 12      AUTOEXTEND ON NEXT 5M MAXSIZE UNLIMITED
 13  CHARACTER SET WE8MSWIN1252 NATIONAL CHARACTER SET AL16UTF16
 14  LOGFILE GROUP 1 SIZE 50M,GROUP 2 SIZE 50M,GROUP 3 SIZE 50M,
 15      GROUP 4 SIZE 50M,GROUP 5 SIZE 50M,GROUP 6 SIZE 50M
 16  USER SYS IDENTIFIED BY "&&sysPassword"
 17  USER SYSTEM IDENTIFIED BY "&&systemPassword"
 18  ENABLE PLUGGABLE DATABASE SEED
 19      SYSTEM DATAFILES SIZE 125M
 20      AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED
 21      SYSAUX DATAFILES SIZE 100M
 22      USER_DATA TABLESPACE PBDUSERS
 23      DATAFILE SIZE 5M AUTOEXTEND ON MAXSIZE UNLIMITED ;

Entrez une valeur pour syspassword : Razvanpwd3
Entrez une valeur pour systempassword : Razvanpwd3

Base de données créée.

SYS@gemmes>select f fichier from (
  2  select con_id, name f from v$controlfile union all
  3  select con_id, name from v$datafile      union all
  4  select con_id, name from v$tempfile     union all
```



```

5  select con_id, member from v$logfile ) order by con_id,fichier ;

ID FICHIER
-----
0 /u03/recuperations/oradata/GEMMES/controlfile/o1_mf_9kcs414y_.ctl
0 /u03/recuperations/oradata/GEMMES/onlinelog/o1_mf_1_9kcs41pm_.log
0 /u03/recuperations/oradata/GEMMES/onlinelog/o1_mf_2_9kcs41qp_.log
0 /u03/recuperations/oradata/GEMMES/onlinelog/o1_mf_3_9kcs41rs_.log
0 /u03/recuperations/oradata/GEMMES/onlinelog/o1_mf_4_9kcs41sw_.log
0 /u03/recuperations/oradata/GEMMES/onlinelog/o1_mf_5_9kcs41tx_.log
0 /u03/recuperations/oradata/GEMMES/onlinelog/o1_mf_6_9kcs41wl_.log
1 /u02/donnees/oradata/GEMMES/datafile/o1_mf_sysaux_9kcs5bql_.dbf
1 /u02/donnees/oradata/GEMMES/datafile/o1_mf_system_9kcs4fd5_.dbf
1 /u02/donnees/oradata/GEMMES/datafile/o1_mf_temp_9kcs63dk_.tmp
1 /u02/donnees/oradata/GEMMES/datafile/o1_mf_undotbs1_9kcs5xoc_.dbf
1 /u02/donnees/oradata/GEMMES/datafile/o1_mf_users_9kcs63k3_.dbf
2 /u02/donnees/oradata/GEMMES/F...3/datafile/o1_mf_pbdusers_9kcs646h_.dbf
2 /u02/donnees/oradata/GEMMES/F...3/datafile/o1_mf_sysaux_9kcs5t3g_.dbf
2 /u02/donnees/oradata/GEMMES/F...3/datafile/o1_mf_system_9kcs55dt_.dbf
2 /u02/donnees/oradata/GEMMES/F...3/datafile/o1_mf_temp_9kcs63fj_.tmp
2 /u02/donnees/oradata/GEMMES/F...3/datafile/o1_mf_users_9kcs63sy_.dbf

SYS@gemmes>!ls /u02/donnees/oradata/GEMMES/
datafile  F3C9CEAE4D1BB53FE0437009A8C00B33

```

Tous les fichiers du modèle de base de données insérée sont stockés dans un sous-répertoire créé automatiquement avec la création des fichiers. Pour plus de détails, voir le Module Le fichier de contrôle.



```

SYS@gemmes>!mkdir -p /u02/donnees/oradata/gemmes/pdb
SYS@gemmes>!mkdir -p /u03/recuperations/oradata/gemmes/pdb
SYS@gemmes> CREATE DATABASE "gemmes" MAXDATAFILES 1024
2  DATAFILE '/u02/donnees/oradata/gemmes/system.dbf'
3      SIZE 700M AUTOEXTEND ON NEXT 10M
4      MAXSIZE UNLIMITED EXTENT MANAGEMENT LOCAL
5  SYSAUX DATAFILE '/u02/donnees/oradata/gemmes/sysaux.dbf'
6      SIZE 550M AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED
...
17 LOGFILE
18 GROUP 1 '/u03/recuperations/oradata/gemmes/journal01.log' SIZE 50M,
...
23 ENABLE PLUGGABLE DATABASE SEED
24 FILE_NAME_CONVERT = ('/gemmes/', '/gemmes/pdb/');          <-----

SYS@gemmes> select con_id id,name fichier from (
2  select con_id, name from v$datafile union all
3  select con_id, name from v$tempfile union all
4  select con_id, member from v$logfile ) order by con_id,fichier ;

ID FICHIER
-----
0 /u03/recuperations/oradata/gemmes/journal01.log
0 /u03/recuperations/oradata/gemmes/journal02.log
0 /u03/recuperations/oradata/gemmes/journal03.log
1 /u02/donnees/oradata/gemmes/sysaux.dbf
1 /u02/donnees/oradata/gemmes/system.dbf
1 /u02/donnees/oradata/gemmes/temp.tmp
1 /u02/donnees/oradata/gemmes/undotbs1.dbf
1 /u02/donnees/oradata/gemmes/users.dbf
2 /u02/donnees/oradata/gemmes/pdb/sysaux.dbf
2 /u02/donnees/oradata/gemmes/pdb/system.dbf
2 /u02/donnees/oradata/gemmes/pdb/temp.tmp
2 /u02/donnees/oradata/gemmes/pdb/users.dbf

```

L'argument « **FILE\_NAME\_CONVERT** » permet de modifier le nom d'une partie de l'arborescence, il est également possible d'utiliser une liste de modèles de remplacement.

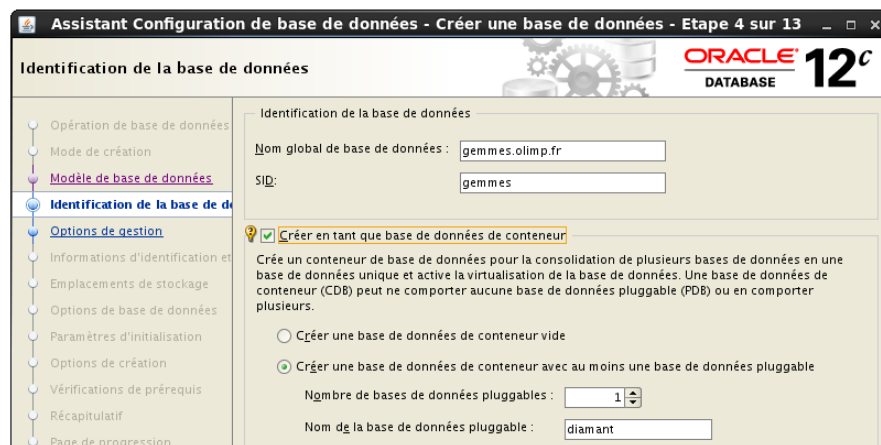
Une fois la base de données créée, il faut exécuter les scripts de création de l'interface publique du dictionnaire de données. Pour les bases de données de conteneur mutualisé Oracle, préconise l'utilisation du script perl « **catcon.pl** ». Il ne faut pas oublier d'initialiser d'abord la variable d'environnement « **PERL5LIB** » avec le répertoire suivant :

```
$ORACLE_HOME/perl/lib/5.14.1
```



```
SQL> spool /u01/app/oracle/admin/gemmes/scripts/CreateDBCatalog.log append
SQL> host perl /u01/app/oracle/product/12.1.0/db_home/rdbms/admin/catcon.pl -n 1 -1
/u01/app/oracle/admin/gemmes/scripts -b catalog
/u01/app/oracle/product/12.1.0/db_home/rdbms/admin/catalog.sql;
...
SQL> host perl /u01/app/oracle/product/12.1.0/db_home/rdbms/admin/catcon.pl -n 1 -1
/u01/app/oracle/admin/gemmes/scripts -b catblock
/u01/app/oracle/product/12.1.0/db_home/rdbms/admin/catblock.sql;
...
SQL> host perl /u01/app/oracle/product/12.1.0/db_home/rdbms/admin/catcon.pl -n 1 -1
/u01/app/oracle/admin/gemmes/scripts -b catproc
/u01/app/oracle/product/12.1.0/db_home/rdbms/admin/catproc.sql;
...
SQL> host perl /u01/app/oracle/product/12.1.0/db_home/rdbms/admin/catcon.pl -n 1 -1
/u01/app/oracle/admin/gemmes/scripts -b catoctk
/u01/app/oracle/product/12.1.0/db_home/rdbms/admin/catoctk.sql;
...
SQL> host perl /u01/app/oracle/product/12.1.0/db_home/rdbms/admin/catcon.pl -n 1 -1
/u01/app/oracle/admin/gemmes/scripts -b owminst
/u01/app/oracle/product/12.1.0/db_home/rdbms/admin/owminst.plb;
...
```

En utilisant « **DBCA** » vous pouvez choisir de créer une base de données conteneur mutualisé CDB et en même temps de créer une ou plusieurs bases de données insérées.



## La connexion aux CDB et PDB

Lorsque vous êtes connecté à une base de données conteneur ou à une base de données insérée vous pouvez changer de base, pourvu que le niveau des privilèges de votre session le permette, en utilisant la syntaxe suivante :

```
ALTER SESSION SET CONTAINER = nom ;
```

Vous bénéficiez également de deux pseudo-colonnes « **CON\_NAME** » et « **CON\_ID** » qui permettent de retrouver facilement le nom de la base de données insérée et son identifiant.



```
[oracle@jupiter ~]$ echo $ORACLE_SID
gemmes
[oracle@jupiter ~]$ sqlplus / as sysdba
SYS@gemmes> show con_name

CON_NAME
-----
CDB$ROOT
SYS@gemmes> alter session set container = pdb$seed;

Session modifiée.

SYS@gemmes> show con_name

CON_NAME
-----
PDB$SEED
```

La vue du dictionnaire de données « **CDB\_PDBS** » ou vue dynamique « **V\$PDBS** » vous permettent de retrouver l'ensemble des bases de données insérées dans le conteneur mutualisé. La vue dynamique permet l'interrogation même si la base de données insérée n'est pas ouverte.



```
[oracle@jupiter ~]$ sqlplus / as sysdba
SYS@gemmes> select con_id,pdb_id,pdb_name,dbid,creation_scn from cdb_pdb;
```

| CON_ID | PDB_ID | PDB_NAME  | DBID       | CREATION_SCN |
|--------|--------|-----------|------------|--------------|
| 1      | 2      | PDB\$SEED | 4084020650 | 235          |

```
SYS@gemmes>select con_id,name,open_mode,restricted, total_size from v$pdb;
```

| CON_ID | NAME      | OPEN_MODE | RES | TOTAL_SIZE |
|--------|-----------|-----------|-----|------------|
| 2      | PDB\$SEED | READ ONLY | NO  | 393216000  |

## L'arrêt et le démarrage

Pour modifier l'état d'une base de données insérée du mode « **MOUNT** » à « **OPEN** » ou l'inverse vous devez utiliser la syntaxe suivante :

```
ALTER PLUGGABLE DATABASE
{ nom { OPEN READ { ONLY | WRITE } [RESTRICTED] [FORCE]
      | CLOSE [ IMMEDIATE ] } | ALL [ EXCEPT nom [, ...]] } ;
```

Comme vous pouvez le voir, cette syntaxe permet de modifier une base mais également l'ensemble des bases insérées. Il est possible d'utiliser cette syntaxe pour modifier les caractéristiques de la base comme la taille maximale, le tablespace par défaut, etc.

### Attention

Attention après le démarrage de la base de données conteneur l'ensemble des bases de données insérées n'est pas automatiquement démarré.

La meilleure solution est de créer un trigger qui est lancé automatiquement après chaque démarrage de la base de données conteneur.





```

SYS@pierres>startup force
...
SYS@pierres>select con_id, name, open_mode, restricted from v$pdb;

  CON_ID NAME                                OPEN_MODE  RES
-----
      2 PDB$SEED                            READ ONLY  NO
      3 JASPE                                MOUNTED    <-----

SYS@pierres>CREATE OR REPLACE TRIGGER STARTUP_PDB
2      AFTER STARTUP ON DATABASE DECLARE
3      BEGIN EXECUTE IMMEDIATE 'ALTER PLUGGABLE DATABASE ALL OPEN'; END;
4      /

Déclencheur créé.

SYS@pierres>startup force
...
SYS@pierres> select con_id, name, open_mode, restricted from v$pdb;

  CON_ID NAME                                OPEN_MODE  RES
-----
      2 PDB$SEED                            READ ONLY  NO
      3 JASPE                                READ WRITE NO
                                           <-----

```

## La création d'une PDB

Une fois l'architecture mise en place, il est possible de créer une base de données insérée à partir du modèle de base de données insérée « **PDB\$SEED** » ou en clonant n'importe quelle base de données insérée. Vous pouvez également déplacer une base de données insérée d'un conteneur mutualisé vers un autre. Il est également possible d'insérer une base de données standard dans un conteneur ou d'extraire une base de données insérée pour la transformer en base de données standard. Ces deux dernières possibilités seront traitées dans la partie sauvegarde et restauration.

La syntaxe de création d'une base de données insérée à partir du modèle de base de données insérée « **PDB\$SEED** » est :

```

CREATE PLUGGABLE DATABASE nom
ADMIN USER utilisateur IDENTIFIED BY mot_de_passe
DEFAULT TABLESPACE nom_tablespace [fichier ...]
FILE_NAME_CONVERT = { ( 'modèle01', 'modèle02' [,...] ) | NONE }
STORAGE ( MAXSIZE {taille|UNLIMITED} MAX_SHARED_TEMP_SIZE ... )
PATH_PREFIX = 'répertoire' ;

```

**PATH\_PREFIX** L'option permet de limiter le stockage, le répertoire spécifié ou ses sous-répertoires. Attention vous ne pouvez plus modifier le réglage après avoir créé votre base de données insérée.

**MAXSIZE** Limite la taille maximale qui peut être utilisée par tous les tablespaces de cette base. Par défaut, il n'y a pas de limite.

**MAX\_SHARED\_TEMP\_SIZE** Limite la taille maximale des espaces temporaires communs qui peut être utilisée par cette base. Par défaut il n'y a pas de limite.



```
C:\>set ORACLE_SID
ORACLE_SID=pierres

C:\>sqlplus / as sysdba

SYS@pierres>select con_id, name, open_mode, guid from v$pdb;

  CON_ID NAME                OPEN_MODE  GUID
-----
      2 PDB$SEED              READ ONLY  9B59917F82DF4EF59C2B89D1DBDD8805

SYS@pierres>select NAME from v$pdb where con_id = 2 union all
  2  select name from v$datafile where con_id = 2 union all
  3  select name from v$tempfile where con_id = 2 ;

NAME
-----
PDB$SEED
D:\DONNEES\ORADATA\PIERRES\9B5...805\DATAFILE\O1_MF_SYSTEM_9KC03PTQ_.DBF
D:\DONNEES\ORADATA\PIERRES\9B5...805\DATAFILE\O1_MF_SYSAUX_9KC040NJ_.DBF
D:\DONNEES\ORADATA\PIERRES\9B5...805\DATAFILE\O1_MF_TEMP_9KC04536_.TMP

SYS@pierres>$dir D:\DONNEES\ORADATA\PIERRES\9B5*805 /B
9B59917F82DF4EF59C2B89D1DBDD8805

SYS@pierres>create pluggable database jaspe
  2  admin user jaspe_admin identified by oracle
  3  file_name_convert = (
  4    '9B5...805\DATAFILE\O1_MF_SYSTEM_9KC03PTQ_.DBF',
  5    '\JASPE\system_jaspe.dbf',
  6    '9B5...805\DATAFILE\O1_MF_SYSAUX_9KC040NJ_.DBF',
  7    '\JASPE\sysaux_jaspe.dbf',
  8    '9B5...805\DATAFILE\O1_MF_TEMP_9KC04536_.TMP',
  9    '\JASPE\temp_jaspe.dbf' );

Base de données pluggable créée.

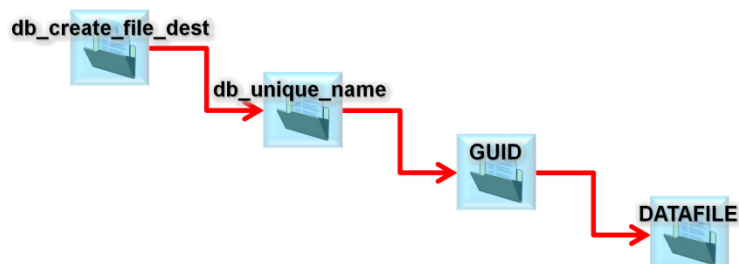
SYS@pierres>select pdb_id, pdb_name, dbid, creation_scn from cdb_pdb;

  PDB_ID PDB_NAME          DBID  CREATION_SCN
-----
      2 PDB$SEED          4083899391      227
      3 JASPE            2135255245     2247648

SYS@pierres>select con_id, name, open_mode, guid from v$pdb;

  CON_ID NAME                OPEN_MODE  GUID
-----
      2 PDB$SEED              READ ONLY  9B59917F82DF4EF59C2B89D1DBDD8805
      3 JASPE                  MOUNTED    2EBA1C5AA90341449960CA3F1C847FA6
```

Le répertoire de stockage du modèle « **PDB\$SEED** » dans le cas où la base de données conteneur mutualisé CDB est créé avec la gestion automatique des noms des fichiers OMF est la suivante :



**GUID**

L'identifiant unique de la base de donnée insérée (**G**lobal **U**nique **I**dentifier). Vous pouvez trouver cet identifiant en interrogeant la vue « **V\$PDBS** ».

## Le clonage d'une base insérée

La syntaxe de création d'une base de données insérée à partir d'une autre base de données insérée est :

```
CREATE PLUGGABLE DATABASE nom FROM source
DEFAULT TABLESPACE nom_tablespace [fichier ...]
FILE_NAME_CONVERT = { ( 'modèle01', 'modèle02' [,...] ) | NONE }
STORAGE ( MAXSIZE {taille|UNLIMITED} MAX_SHARED_TEMP_SIZE ... )
PATH_PREFIX = 'répertoire' ;
```

La connexion aux bases de données insérées est traitée dans le Module L'architecture Oracle Net. Pour l'instant nous allons utiliser la commande « **ALTER SESSION SET CONTAINER** » pour travailler avec les bases de données insérées.



```
C:\>sqlplus / as sysdba
Connecté.

SYS@pierres>select con_id, name, open_mode, restricted from v$pdb;

   CON_ID NAME                                OPEN_MODE  RES
-----
      3 JASPE                                MOUNTED

SYS@pierres>alter database open;

Base de données modifiée.

SYS@pierres>create tablespace dtb_tran datafile
  2  'D:\DONNEES\ORADATA\PIERRES\JASPE\dtb_tran.dbf' size 10m;

Tablespace créé.
```

### Attention



Un clonage des bases de données insérées n'est possible que si la base de données source est en lecture seule. Le modèle de base de données insérée « **PDB\$SEED** », est toujours en lecture seule.

Ainsi il faut d'abord mettre la base de données insérée source en lecture seule et ensuite cloner la base de données.



```
SYS@pierres>connect / as sysdba
Connecté.

SYS@pierres>alter pluggable database jaspe close immediate;

Base de données pluggable modifiée.

SYS@pierres>alter pluggable database jaspe open read only;

Base de données pluggable modifiée.

SYS@pierres>create pluggable database onyx from jaspe
  2  file_name_convert = ('\JASPE\','\ONYX\') storage ( maxsize 10g);
```

Base de données pluggable créée.

```
SYS@pierres>alter pluggable database onyx open;
```

Base de données pluggable modifiée.

```
SYS@pierres>select tb.con_id, db.pdb_name, tb.tablespace_name
  2   from cdb_tablespaces tb, cdb_pdbas db
  3   where tb.con_id = db.pdb_id(+) order by 1;
```

| CON_ID | PDB_NAME  | TABLESPACE_NAME |
|--------|-----------|-----------------|
| 1      |           | USERS           |
| 1      |           | TEMP            |
| 1      |           | SYSTEM          |
| 1      |           | SYSAUX          |
| 1      |           | UNDOTBS1        |
| 2      | PDB\$SEED | SYSTEM          |
| 2      | PDB\$SEED | TEMP            |
| 2      | PDB\$SEED | SYSAUX          |
| 3      | JASPE     | TEMP            |
| 3      | JASPE     | SYSAUX          |
| 3      | JASPE     | SYSTEM          |
| 3      | JASPE     | DTB_TRAN        |
| 4      | ONYX      | SYSTEM          |
| 4      | ONYX      | SYSAUX          |
| 4      | ONYX      | TEMP            |
| 4      | ONYX      | DTB_TRAN        |

```
SYS@pierres>select NAME from v$pdbas where con_id = 4 union all
  2   select name from v$datafile where con_id = 4 union all
  3   select name from v$tempfile where con_id = 4;
```

| NAME                                             |
|--------------------------------------------------|
| ONYX                                             |
| D:\DONNEES\ORADATA\PIERRES\ONYX\SYSTEM_JASPE.DBF |
| D:\DONNEES\ORADATA\PIERRES\ONYX\SYSAUX_JASPE.DBF |
| D:\DONNEES\ORADATA\PIERRES\ONYX\DTB_TRAN.DBF     |
| D:\DONNEES\ORADATA\PIERRES\ONYX\TEMP_JASPE.DBF   |

Vous pouvez laisser Oracle gérer le nom des fichiers de la base de données insérée, ce qui facilite beaucoup la maintenance de cette base par la suite.



```
SYS@pierres>connect / as sysdba
```

Connecté.

```
SYS@pierres>   alter pluggable database hematite open;
```

Base de données pluggable modifiée.

```
SYS@pierres>select name from v$pdbas      where con_id = 5 union all
  2   select name from v$datafile where con_id = 5 union all
  3   select name from v$tempfile where con_id = 5 ;
```

| NAME                                                                        |
|-----------------------------------------------------------------------------|
| HEMATITE                                                                    |
| D:\DONNEES\ORADATA\PIERRES\105...BFF\DATAFILE\O1_MF_SYSTEM_9KFX07DD_.DBF    |
| D:\DONNEES\ORADATA\PIERRES\105...BFF \DATAFILE\O1_MF_SYSAUX_9KFX07CW_.DBF   |
| D:\DONNEES\ORADATA\PIERRES\105...BFF \DATAFILE\O1_MF_DTB_TRAN_9KFX07DD_.DBF |
| D:\DONNEES\ORADATA\PIERRES\105...BFF \DATAFILE\O1_MF_TEMP_9KFX0SHB_.DBF     |



```
SYS@pierres>$dir D:\DONNEES\ORADATA\PIERRES\105*BFF /B
1057D324F3004D8B984882E1FD037BFF
```

Vous pouvez utiliser la clause « **FILE\_NAME\_CONVERT** » pour créer une base de données insérée dans les groupes de disques de votre instance ASM.

```
SYS@pierres> connect / as sysdba
Connecté.
SYS@pierres> alter pluggable database hematite close immediate;

Base de données pluggable modifiée.

SYS@pierres> alter pluggable database hematite open read only;

Base de données pluggable modifiée.

SYS@pierres> create pluggable database emeraude from hematite
  2 FILE_NAME_CONVERT = ('D:\DONNEES\ORADATA\PIERRES\105...BFF\DATAFILE\' ,
  3   '+GD_DONNEES/PIERRES/') storage ( maxsize 10g);

Base de données pluggable créée.

SYS@pierres>alter pluggable database emeraude open;

Base de données pluggable modifiée.

SYS@pierres>select name from (
  2 select con_id id, name,0 t from v$pdb$ union all
  3 select con_id id, name,2 t from v$datafile union all
  4 select con_id id, name,3 t from v$tempfile) order by id, t;

NAME
-----
D:\DONNEES\ORADATA\PIERRES\DATAFILE\O1_MF_USERS_9KC04PQP_.DBF
...
PDB$SEED
D:\DONNEES\ORADATA\PIERRES\9B5...805\DATAFILE\O1_MF_SYSTEM_9KC03PTQ_.DBF
D:\DONNEES\ORADATA\PIERRES\9B5...805\DATAFILE\O1_MF_SYSAUX_9KC040NJ_.DBF
D:\DONNEES\ORADATA\PIERRES\9B5...805\DATAFILE\O1_MF_TEMP_9KC04536_.TMP
JASPE
D:\DONNEES\ORADATA\PIERRES\JASPE\SYSTEM_JASPE.DBF
D:\DONNEES\ORADATA\PIERRES\JASPE\SYSAUX_JASPE.DBF
D:\DONNEES\ORADATA\PIERRES\JASPE\DTB_TRAN.DBF
D:\DONNEES\ORADATA\PIERRES\JASPE\TEMP_JASPE.DBF
ONYX
D:\DONNEES\ORADATA\PIERRES\ONYX\DTB_TRAN.DBF
D:\DONNEES\ORADATA\PIERRES\ONYX\SYSAUX_JASPE.DBF
D:\DONNEES\ORADATA\PIERRES\ONYX\SYSTEM_JASPE.DBF
D:\DONNEES\ORADATA\PIERRES\ONYX\TEMP_JASPE.DBF
HEMATITE
D:\DONNEES\ORADATA\PIERRES\105...BFF\DATAFILE\O1_MF_SYSAUX_9KFX07CW_.DBF
D:\DONNEES\ORADATA\PIERRES\105...BFF \DATAFILE\O1_MF_DTB_TRAN_9KFX07DD_.DBF
D:\DONNEES\ORADATA\PIERRES\105...BFF \DATAFILE\O1_MF_SYSTEM_9KFX07DD_.DBF
D:\DONNEES\ORADATA\PIERRES\105...BFF \DATAFILE\O1_MF_TEMP_9KFX0SHB_.DBF
EMERAUDE
+GD_DONNEES/PIERRES/o1_mf_sysaux_9kfx07cw_.dbf
+GD_DONNEES/PIERRES/o1_mf_system_9kfx07dd_.dbf
+GD_DONNEES/PIERRES/o1_mf_dtb_tran_9kfx07dd_.dbf
+GD_DONNEES/PIERRES/o1_mf_temp_9kfx0shb_.dbf
```



## L'effacement d'une base insérée

Vous pouvez effacer une base de données insérée en conservant ou en effaçant les fichiers de données ou bien à l'aide de la syntaxe suivante :

```
DROP PLUGGABLE DATABASE nom [ { KEEP | INCLUDING } DATAFILES ] ;
```



```
SYS@pierres>connect / as sysdba
Connecté.
SYS@pierres>select name, open_mode, restricted from v$pdb;
```

| NAME      | OPEN_MODE  | RES |
|-----------|------------|-----|
| PDB\$SEED | READ ONLY  | NO  |
| JASPE     | READ WRITE | NO  |
| ONYX      | READ WRITE | NO  |
| HEMATITE  | READ ONLY  | NO  |
| EMERAUDE  | READ WRITE | NO  |

```

SYS@pierres>alter pluggable database hematite close immediate;

Base de données pluggable modifiée.

SYS@pierres>drop pluggable database hematite including datafiles;

Base de données pluggable supprimée.

SYS@pierres>select name, open_mode, restricted from v$pdb;
```

| NAME      | OPEN_MODE  | RES |
|-----------|------------|-----|
| PDB\$SEED | READ ONLY  | NO  |
| JASPE     | READ WRITE | NO  |
| ONYX      | READ WRITE | NO  |
| EMERAUDE  | READ WRITE | NO  |

Il est évident que vous ne pouvez pas effacer une base de données insérée si elle n'est pas arrêtée.

## La gestion d'une base insérée

Il est possible de modifier un certain nombre de paramètres uniquement pour une base de données insérée. Ainsi vous pouvez interroger la vue « **V\$PARAMETER** » pour visualiser quels sont les paramètres qui peuvent être modifiés pour chaque base de données insérée.



```
SYS@pierres>connect / as sysdba
Connecté.
SYS@pierres>select name, isinstance_modifiable i, value from v$parameter
  2  where ispdb_modifiable='TRUE' order by name;
```

| NAME                        | I    | VALUE                    |
|-----------------------------|------|--------------------------|
| ...                         |      |                          |
| cursor_sharing              | TRUE | EXACT                    |
| db_create_file_dest         | TRUE | D:\donnees\oradata       |
| db_create_online_log_dest_1 | TRUE | R:\recuperations\oradata |
| ...                         |      |                          |

```

SYS@pierres>alter session set container = onyx;

Session modifiée.

SYS@pierres>show con_name

CON_NAME
-----
ONYX
SYS@pierres>show parameter db_cre

NAME                                TYPE          VALUE
-----
db_create_file_dest                 string        D:\donnees\oradata
db_create_online_log_dest_1         string        R:\recuperations\oradata
...
SYS@pierres>alter system set db_create_file_dest='+GD_DONNEES';

Système modifié.

SYS@pierres>create tablespace donnees_onyx datafile size 10M;

Tablespace créé.

SYS@pierres>select file_name from dba_data_files
  2  where tablespace_name ='DONNEES_ONYX';

FILE_NAME
-----
+GD_DONNEES/PIERRES/FE2...4F9/DATAFILE/donnees_onyx.294.841422653

SYS@pierres>connect / as sysdba
Connecté.
SYS@pierres>show parameter db_cre

NAME                                TYPE          VALUE
-----
db_create_file_dest                 string        D:\donnees\oradata
db_create_online_log_dest_1         string        R:\recuperations\oradata
...

```

Il est également possible de changer de nom d'une base de données insérée à l'aide de la syntaxe suivante :

```

ALTER PLUGGABLE DATABASE nom
    RENAME GLOBAL_NAME TO nouveau_nom ;

```



```

SYS@pierres>connect / as sysdba
Connecté.
SYS@pierres>alter session set container = onyx;

Session modifiée.

SYS@pierres>show con_name

CON_NAME
-----
ONYX
SYS@pierres>shutdown immediate
Base de données pluggable fermée.
SYS@pierres>startup restrict

```

```

Base de données pluggable ouverte.
SYS@pierres>alter pluggable database onyx rename global_name to ambre;

Base de données pluggable modifiée.

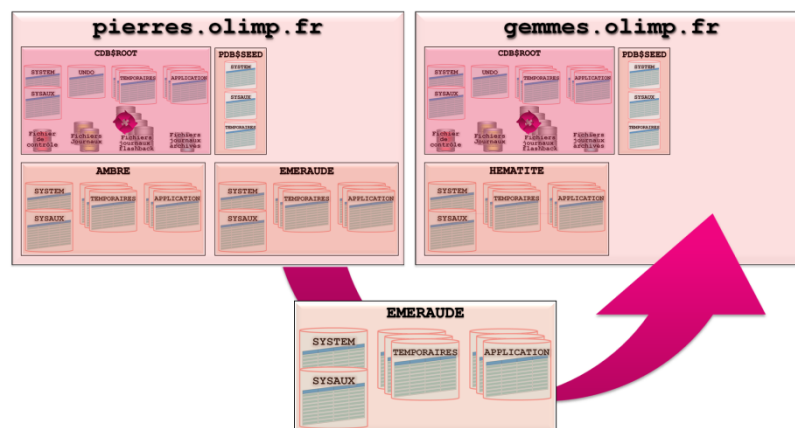
SYS@pierres>select name, open_mode, restricted from v$pdb;

NAME                                OPEN_MODE  RES
-----
AMBRE                                READ WRITE YES

```

## L'extraction et l'insertion

Vous pouvez extraire une base de données insérée d'un conteneur mutualisé et l'insérer dans un autre conteneur. Dans ce module nous allons uniquement étudier l'extraction et l'insertion dans deux conteneurs situés sur la même machine, les autres cas sont décrits dans la partie sauvegarde et restauration.



### L'extraction d'une base de données insérée

L'extraction consiste à créer un fichier de type « **XML** » qui contient les informations concernant la base de données insérée. La syntaxe d'extraction du fichier est :

```
ALTER PLUGGABLE DATABASE nom UNPLUG INTO 'fichier' ;
```

Le nom du fichier de type « **XML** » créé par l'extraction doit être spécifié de manière absolue à partir du disque racine.

#### Attention

Attention une fois que la base de données insérée est extraite du conteneur mutualisé, il n'est plus possible de l'utiliser. Elle doit être impérativement effacée, car elle ne peut plus être accessible avant une éventuelle réinsertion dans le même conteneur.

Pour l'extraction, la base de données insérée doit être fermée au préalable car l'opération ne peut pas s'exécuter à partir d'une base de données ouverte.

```

C:\>set ORACLE_SID=pierres
C:\>sqlplus / as sysdba
SYS@pierres>select NAME from v$pdb where con_id = 5 union all
 2  select name from v$datafile where con_id = 5 union all
 3  select name from v$tempfile where con_id = 5 ;

NAME
-----

```

```

EMERAUDE
+GD_DONNEES/PIERRES/o1_mf_system_9kfx07dd_.dbf
+GD_DONNEES/PIERRES/o1_mf_sysaux_9kfx07cw_.dbf
+GD_DONNEES/PIERRES/o1_mf_dtb_tran_9kfx07dd_.dbf
+GD_DONNEES/PIERRES/o1_mf_temp_9kfx0shb_.dbf

SYS@pierres>alter pluggable database emeraude close immediate;

Base de données pluggable modifiée.

SYS@pierres>select name, open_mode, restricted from v$pdb;

NAME                                OPEN_MODE  RES
-----
PDB$SEED                            READ ONLY  NO
JASPE                               READ WRITE NO
AMBRE                               READ WRITE YES
EMERAUDE                            MOUNTED

SYS@pierres>alter pluggable database emeraude unplug into
  2  'O:\app\oracle\admin\pierres\admin_scripts\emeraude.xml';

Base de données pluggable modifiée.

SYS@pierres>drop pluggable database emeraude keep datafiles;

Base de données pluggable supprimée.

```

Dans notre exemple, la base de données est extraite du conteneur mutualisé mais ses fichiers sont restés au même emplacement sur le groupe de disque ASM « **+GD\_DONNEES** ». En pratique le fichier de type « **XML** » généré doit être contrôlé avant son utilisation, car il peut y avoir des incompatibilités de transport. Le contrôle du fichier est effectué en utilisant la fonction « **CHECK\_PLUG\_COMPATIBILITY** » du package « **DBMS\_PDB** ».



```

SYS@gemmes>desc DBMS_PDB
FUNCTION CHECK_PLUG_COMPATIBILITY RETURNS BOOLEAN
Nom d'argument                      Type                      E/S par défaut ?
-----
PDB_DESCR_FILE                      VARCHAR2                  IN
PDB_NAME                            VARCHAR2                  IN      DEFAULT
...
SYS@pierres>declare
  2  controle_fichier boolean;
  3  begin
  4  controle_fichier :=
  5  DBMS_PDB.CHECK_PLUG_COMPATIBILITY( pdb_descr_file=>
  6  'O:\app\oracle\admin\pierres\admin_scripts\emeraude.xml');
  7  if controle_fichier then
  8  dbms_output.put_line('Fichier valide vous pouvez continuer. ');
  9  else
  10  dbms_output.put_line('Fichier invalide arrêtez. ');
  11  end if;
  12  end;
  13  /
Fichier valide vous pouvez continuer.

```

## L'insertion à partir d'un fichier

Il faut bien préciser que le fichier de type « **XML** » permet d'avoir les informations concernant la construction de la base de données, l'emplacement des fichiers, la taille des blocs, la version base de données, etc. Les données de votre base se retrouvent bien dans les fichiers de données. Ainsi la

syntaxe d'insertion d'une base de données à partir d'un fichier de type « **XML** » et des fichiers de données est la suivante :

CREATE PLUGGABLE DATABASE nom USING 'fichier' NOCOPY ;



```
C:\>set ORACLE_SID=gemmes
C:\>sqlplus / as sysdba
SYS@gemmes>$dir O:\app\oracle\admin\pierres\admin_scripts\emeraude.xml /B
emeraude.xml
SYS@gemmes>select name, open_mode, restricted from v$pdb;
```

| NAME      | OPEN_MODE  | RES |
|-----------|------------|-----|
| PDB\$SEED | READ ONLY  | NO  |
| HEMATITE  | READ WRITE | NO  |

```
SYS@gemmes>create pluggable database emeraude using
2 'O:\app\oracle\admin\pierres\admin_scripts\emeraude.xml' nocopy;
```

Base de données pluggable créée.

```
SYS@gemmes>alter pluggable database emeraude open read write;
```

Base de données pluggable modifiée.

```
SYS@gemmes>select name, open_mode, restricted, guid from v$pdb;
```

| NAME      | OPEN_MODE  | RES | GUID                             |
|-----------|------------|-----|----------------------------------|
| PDB\$SEED | READ ONLY  | NO  | EBB99B7EA03A4EE48D3A91CA55F78217 |
| HEMATITE  | READ WRITE | NO  | 372C95D6EF7A4DA9A4DECC3C39002B2E |
| EMERAUDE  | READ WRITE | NO  | BF1A900C778844A99EF6970E98D85286 |

```
SYS@gemmes>select NAME from v$pdb where con_id = 4 union all
2 select name from v$datafile where con_id = 4 union all
3 select name from v$tempfile where con_id = 4 ;
```

| NAME                                             |
|--------------------------------------------------|
| EMERAUDE                                         |
| +GD_DONNEES/PIERRES/o1_mf_system_9kfx07dd_.dbf   |
| +GD_DONNEES/PIERRES/o1_mf_sysaux_9kfx07cw_.dbf   |
| +GD_DONNEES/PIERRES/o1_mf_dtb_tran_9kfx07dd_.dbf |
| +GD_DONNEES/PIERRES/o1_mf_temp_9kfx0shb_.dbf     |

```
SYS@gemmes>exit
```

```
C:\>set ORACLE_SID=+ASM
```

```
C:\>set ORACLE_HOME=O:\app\oracle\product\12.1.0\gr_home
```

```
C:\>asmcmd
```

```
ASMCMD> lsof -G GD_DONNEES
```

| DB_Name | Instance_Name | Path                                             |
|---------|---------------|--------------------------------------------------|
| ...     |               |                                                  |
| gemmes  | gemmes        | +GD_DONNEES/PIERRES/o1_mf_dtb_tran_9kfx07dd_.dbf |
| gemmes  | gemmes        | +GD_DONNEES/PIERRES/o1_mf_sysaux_9kfx07cw_.dbf   |
| gemmes  | gemmes        | +GD_DONNEES/PIERRES/o1_mf_system_9kfx07dd_.dbf   |
| gemmes  | gemmes        | +GD_DONNEES/PIERRES/o1_mf_temp_9kfx0shb_.dbf     |
| ...     |               |                                                  |

```
ASMCMD> lsct GD_DONNEES
```

| DB_Name | Status    | Software_Ver | Compatible_ver | Instance_Name | Disk_Group |
|---------|-----------|--------------|----------------|---------------|------------|
| +ASM    | CONNECTED | 12.1.0.1.0   | 12.1.0.1.0     | +asm          | GD_DONNEES |
| gemmes  | CONNECTED | 12.1.0.1.0   | 12.1.0.0.0     | gemmes        | GD_DONNEES |
| pierres | CONNECTED | 12.1.0.1.0   | 12.1.0.0.0     | pierres       | GD_DONNEES |

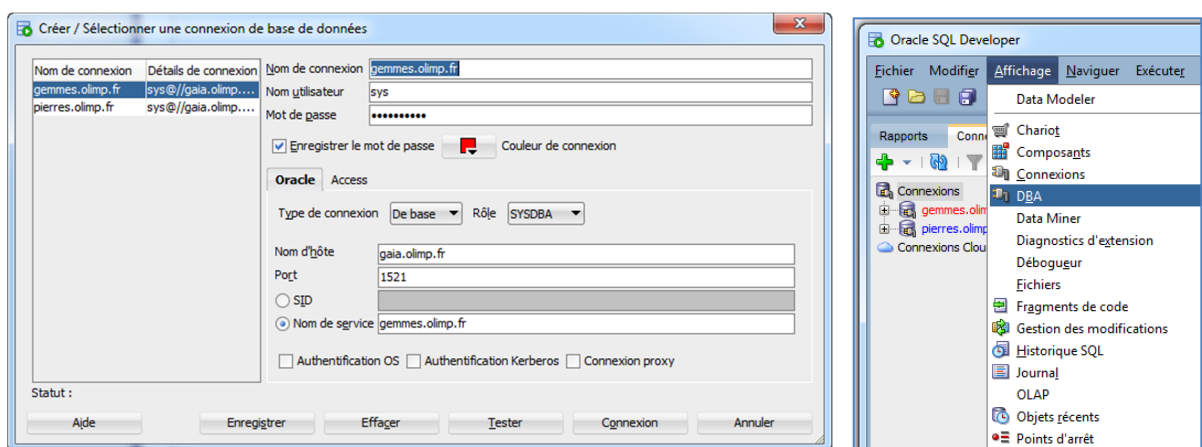
# SQL Developer

SQL Developer est l'outil de développement graphique fourni à partir de la version Oracle 11g. Il est installé directement avec le serveur de bases de données ou avec installation du produit client. Il est également téléchargeable gratuitement sur le site à l'adresse :

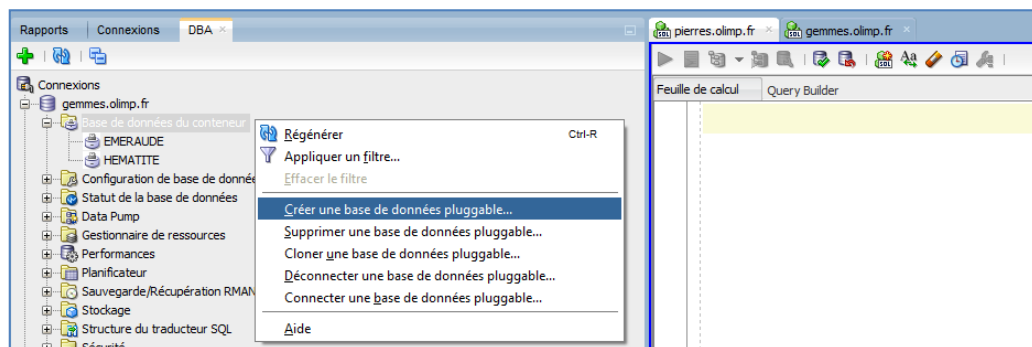
<http://www.oracle.com/technetwork/developer-tools/sql-developer/overview/index.html>

Oracle 12c installe la SQL Developer version 3.2 mais vous pouvez trouver à l'adresse mentionnée une nouvelle version 4.0. Ces deux versions proposent une extension pour l'administration de la base de données qui est un outil complémentaire très riche en fonctionnalités. L'outil ne nécessite aucune installation, il est fourni pour toutes les plates-formes Windows ou Unix/Linux.

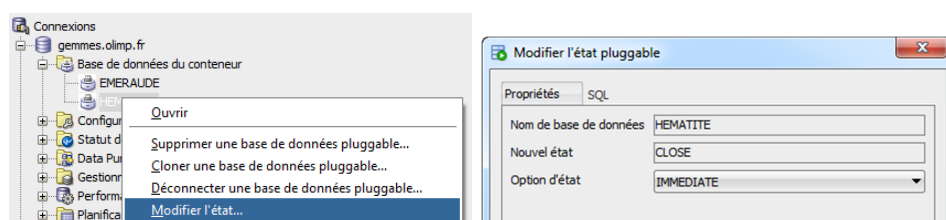
Une fois que les informations de connexion à la base de données sont fournies, vous pouvez accéder à l'outil d'administration de base de données par le menu Affichage/DBA.



Ainsi une fois aussi connectées les deux bases de données dans l'outil d'administration vous pouvez :

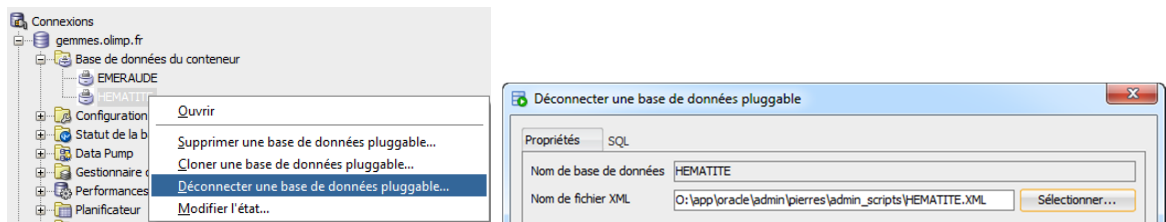


Ainsi pour extraire la base de données insérée « **hematite** » du conteneur mutualisé « **gemme** » et l'insérer dans le conteneur « **pierres** », on commence par arrêter cette base.

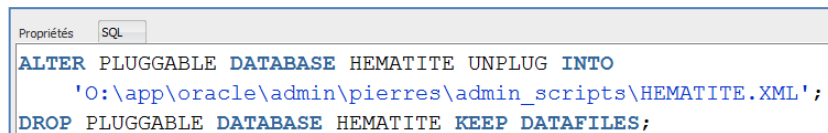


Ensuite on extrait la base de données du conteneur « **gemmes** », il faut faire attention car le fichier « **XML** » est stocké directement sur le serveur et non sur le poste client comme on pourrait le croire.

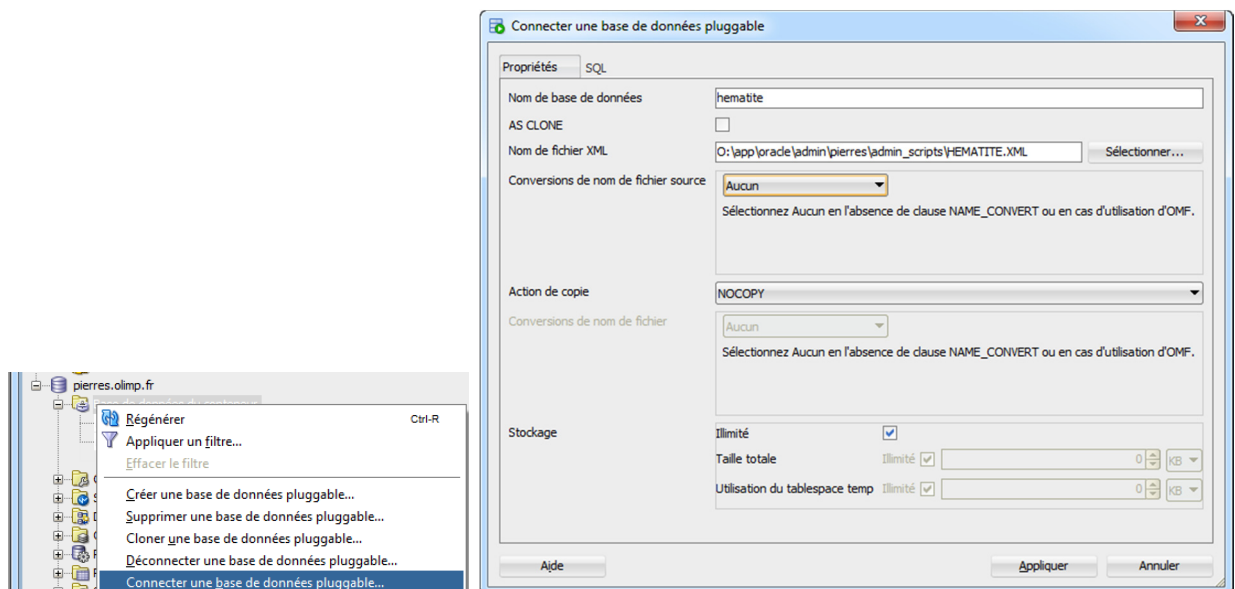
L'opération d'extraction de cette base de données insérée à l'aide de SQL Developer supprime automatiquement la base de données sans effacer les fichiers.



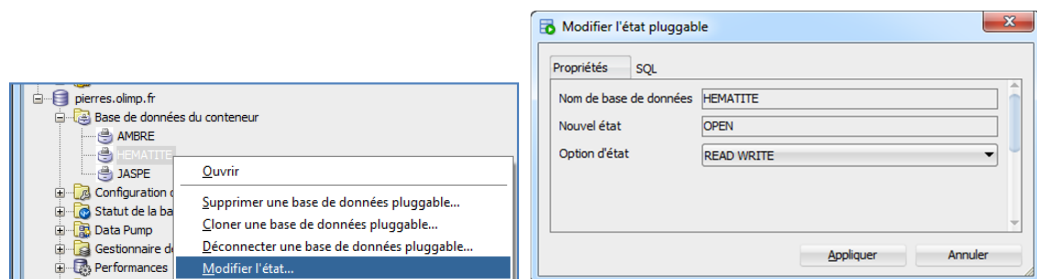
L'onglet SQL permet de visualiser les syntaxes des commandes SQL qui seront exécutées.



Comme les fichiers de la base de données n'ont pas été effacés, l'insertion dans le conteneur mutualisé « **pierres** » peut se poursuivre.



Il reste uniquement l'ouverture de la base de données insérée « **hematite** ».



- *CREATE TABLESPACE*
- *DB\_Nx\_CACHE\_SIZE*
- *TEMPORARY*
- *UNDO*

# 10

## ***Les tablespaces***



### ***Objectifs***

À la fin de ce module, vous serez à même d'effectuer les tâches suivantes :

- Créer un espace logique de stockage permanent.
- Définir le tablespace permanent par défaut pour une base de données.
- Créer un tablespace de type temporaire.
- Créer un groupe de tablespaces temporaires.
- Définir le tablespace temporaire par défaut pour la base de données.
- Créer un tablespace utilisant un stockage crypté.
- Créer un tablespace avec une taille de bloc autre que celle par défaut.



### ***Contenu***

|                             |      |                                   |       |
|-----------------------------|------|-----------------------------------|-------|
| Le tablespace               | 10-2 | La taille du bloc                 | 10-10 |
| Les types de tablespaces    | 10-3 | Le cryptage transparent           | 10-12 |
| La création d'un tablespace | 10-4 | Le tablespace temporaire          | 10-15 |
| Le tablespace par défaut    | 10-7 | Le groupe tablespaces temporaires | 10-17 |
| Le tablespace BIGFILE       | 10-8 | Le tablespace undo                | 10-18 |



# Le tablespace

---

La séparation des structures logique et physique d'une base de données 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.

Comme on l'a vu précédemment, la base de données est divisée en zones d'espace logique plus petites, appelées tablespaces.

## Note



Un tablespace est constitué d'un ou plusieurs fichiers de données (datafiles) ; par contre un fichier de données ne peut appartenir qu'à un seul tablespace.

Un fichier de données est créé automatiquement par le serveur Oracle chaque fois que vous créez le tablespace ou que vous avez ajouté un nouveau fichier au tablespace. La quantité de disque occupé par le fichier de données est spécifiée par l'administrateur de la base de données. Chaque fois que le fichier est créé, l'espace est automatiquement réservé sur disque.

Après la création d'un tablespace, vous pouvez ajouter d'autres fichiers de données. Un fichier de données peut être modifié par l'administrateur de la base de données après sa création.

Dès lors qu'un fichier de données est créé pour un tablespace, il est attaché à ce tablespace, et il va pouvoir être détaché uniquement par la destruction du tablespace.

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, undo 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. Un extent ne peut pas être stocké sur plusieurs fichiers de données.

## Note



Un segment, peut être stocké sur un ou plusieurs fichiers appartenant au même tablespace.

Par contre un extent ne peut pas être stocké sur plusieurs fichiers de données ; il doit être absolument contenu dans le même fichier 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 fichiers de données existants.

Dans ce module, nous allons voir la gestion des tablespaces et des fichiers de données ; la partie stockage est détaillée dans le Module « La gestion du stockage ».

## Attention



Un tablespace est un container qui n'a pas de concept de propriété d'objet. Il n'y a aucune relation d'appartenance entre un tablespace et un propriétaire de structure (ou un propriétaire de table). Les objets possédés par un utilisateur peuvent résider dans de multiples tablespaces ou dans un même tablespace.

La valeur du paramètre « **DB\_CREATE\_FILE\_DEST** » est le nom d'un répertoire existant indiquant à Oracle où créer les fichiers de données et les fichiers temporaires. Chaque fois que vous voulez créer un fichier de données, Oracle crée automatiquement dans ce répertoire un sous-répertoire avec

le nom du paramètre « **DB\_UNIQUE\_NAME** » s'il n'existe pas déjà. Ensuite, pour stocker le fichier, il crée un autre sous-répertoire « **DATAFILE** » pour les fichiers de données ou pour les fichiers temporaires.

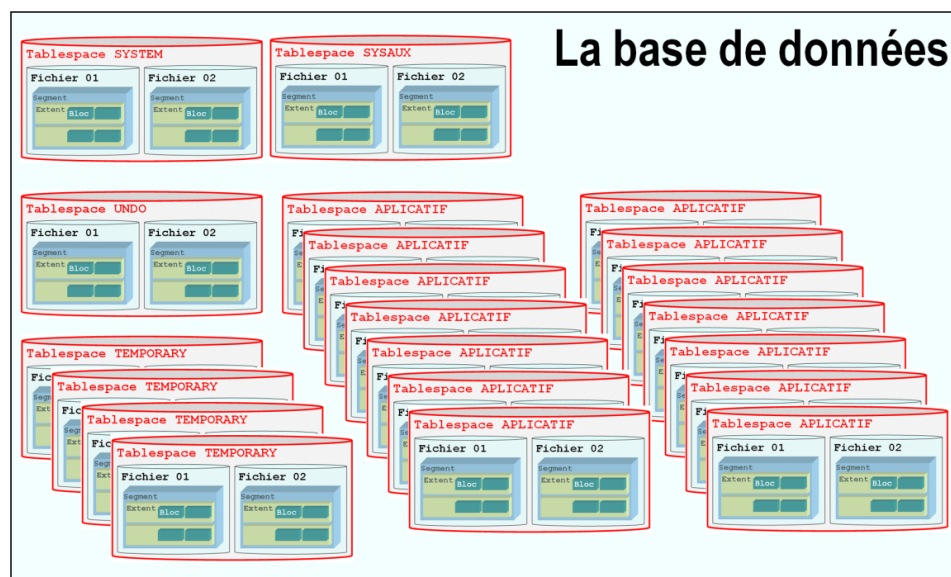
Pour les fichiers stockés dans des répertoires de votre système de fichiers, le format du nom du fichier est le suivant : « **o1\_mf\_%t\_%u\_.dbf** » ou « **o1\_mf\_%t\_%u\_.tmp** ».

Pour les fichiers stockés dans des groupes de disques, le format du nom du fichier est le suivant : « **nom.fichier.incarnation** ».

L'utilisation de plusieurs bases de données sur des environnements indépendants montre que la gestion **OMF** (Oracle Managed Files) utilise les mêmes syntaxes de création et de modification des tablespaces quel que soit l'environnement ou le type stockage de la base.

## Les types de tablespaces

Chaque base de données Oracle créée possède un tablespace « **SYSTEM** ». Il s'agit de l'emplacement où Oracle garde toute l'information du catalogue exigée pour le fonctionnement de la base de données. Un deuxième tablespace « **SYSAUX** » est créé automatiquement à la création de la base de données. Il contient les objets système complémentaires qui permettent ainsi de diminuer le temps d'attente pour les lectures des informations système. Les deux tablespaces « **SYSTEM** » et « **SYSAUX** » sont utilisés uniquement par Oracle ; il faut prendre soin de ne pas stocker des objets utilisateur dans ces deux tablespaces.



Lors d'importantes opérations de tri (telles que select distinct, union et create index), si la taille de la zone **PGA** du processus utilisateur ne suffit pas, Oracle va stocker dans les tablespaces de la base de données des informations concernant le tri des enregistrements avant de retourner l'information aux utilisateurs. En raison de leur nature dynamique, ces espaces de tri ne devraient pas être stockés avec d'autres types de segments. Lorsqu'un tablespace temporaire « **TEMPORARY** » est défini, un segment de tri est aussi créé. Celui-ci est capable de croître si nécessaire afin de pouvoir héberger toutes les opérations de tri de données, et existe jusqu'à ce que la base de données soit fermée puis redémarrée.

Lors de la création de la base de données, vous avez la possibilité de définir un tablespace avec la clause « **DEFAULT TEMPORARY TABLESPACE** » de la commande « **CREATE DATABASE** ». Certains utilisateurs d'une base de données Oracle peuvent avoir besoin de volumes de stockage temporaires beaucoup plus grands que ceux de tous les autres utilisateurs de l'application. Dans ce

cas, vous pouvez créer plusieurs tablespaces temporaires « **TEMPORARY** », pour distribuer les espaces de stockages des utilisateurs ayant des besoins semblables sur les mêmes tablespaces.

Il est également possible lorsque vous créez la base de données de spécifier le tablespace permanent par défaut avec la clause « **DEFAULT TABLESPACE** » de la commande « **CREATE DATABASE** ». Tous les objets qui ne comportent pas, dans la syntaxe de création, les mentions de stockage dans un tablespace, sont stockés automatiquement dans le tablespace permanent par défaut.

Toutes les données d'annulation sont stockées dans un tablespace spécial appelé « **UNDO** ». Lorsque vous créez un tablespace « **UNDO** », Oracle gère le stockage, la rétention et l'emploi de l'espace pour les données de rollback par l'intermédiaire de la fonction SMU (System-Managed Undo). Aucun objet permanent n'est placé dans le tablespace **UNDO**.

Rappelez-vous, la syntaxe SQL de création de la base de données comporte d'abord la création d'un tablespace « **SYSTEM** » et d'un tablespace « **SYSAUX** », ainsi que la création d'un tablespace « **TEMP** » et « **UNDO** ».



```
SQL> CREATE DATABASE "ambre"
 2  MAXINSTANCES 8 MAXLOGHISTORY 1 MAXLOGFILES 16
 3  MAXLOGMEMBERS 3 MAXDATAFILES 1024
 4  DATAFILE SIZE 700M AUTOEXTEND ON NEXT 10240K
 5      MAXSIZE UNLIMITED EXTENT MANAGEMENT LOCAL
 6  SYSAUX DATAFILE SIZE 550M AUTOEXTEND ON NEXT 10240K
 7      MAXSIZE UNLIMITED
 8  DEFAULT TEMPORARY TABLESPACE TEMP TEMPFILE SIZE 20M          <-----
 9      AUTOEXTEND ON NEXT 640K MAXSIZE UNLIMITED
10  UNDO TABLESPACE "UNDOTBS1" DATAFILE SIZE 200M
11      AUTOEXTEND ON NEXT 5120K MAXSIZE UNLIMITED
12  DEFAULT TABLESPACE "USERS" LOGGING DATAFILE SIZE 5M        <-----
13      AUTOEXTEND ON NEXT 5M MAXSIZE UNLIMITED
14  CHARACTER SET WE8MSWIN1252 NATIONAL CHARACTER SET AL16UTF16
15  LOGFILE GROUP 1 SIZE 1G, GROUP 2 SIZE 1G, GROUP 3 SIZE 1G,
16      GROUP 4 SIZE 1G, GROUP 5 SIZE 1G, GROUP 6 SIZE 1G
17  USER SYS IDENTIFIED BY "&&sysPassword"
18  USER SYSTEM IDENTIFIED BY "&&systemPassword";
```

Un tablespace applicatif typique contient tous les objets principaux associés à une application. L'important volume de lectures-écritures dont elles font l'objet, justifie l'isolation de ces tables dans leur propre tablespace, séparant ainsi leurs fichiers de données des autres fichiers de données dans la base. La répartition de ces fichiers sur des disques différents peut de plus améliorer les performances (grâce à une réduction de la contention lors des accès au disque) et simplifier leur gestion.

Les index ne devraient pas être stockés dans le même tablespace que les tables de données sur lesquelles ils ont été définis, car ces deux types de structures font l'objet de nombreuses opérations de lectures-écritures concurrentes lors des manipulations de données et des interrogations.

## La création d'un tablespace

À la création d'un tablespace, vous indiquez le ou les fichiers de données, leurs tailles, leur mode d'agrandissement. La création du tablespace permet également de configurer la taille du bloc, les informations de journalisation ou les options de compression, etc. ; ces options seront détaillées au fur et à mesure que nous avancerons dans ce Module.

L'instruction SQL qui permet de créer un tablespace permanent est :

```
CREATE {BIGFILE|SMALLFILE} TABLESPACE nom_tablespace
    [ DATAFILE ['nom_fichier'] [ SIZE integer {K|M|G|T} ]
    [ AUTOEXTEND {OFF | ON [ NEXT integer {K|M|G|T}]
```

```

[ MAXSIZE {UNLIMITED | integer {K|M|G|T}}] [, ...]
[BLOCKSIZE integer [ K ]]
[({LOGGING | NOLOGGING}) [FORCE LOGGING]
[({ONLINE | OFFLINE})] [FLASHBACK {ON | OFF}]
[ENCRYPTION [ USING 'algorithm']
[IDENTIFIED BY mot_de_passe] DEFAULT STORAGE ENCRYPT]
[ DEFAULT { COMPRESS { BASIC
| FOR { OLTP | { QUERY | ARCHIVE } [ LOW | HIGH ] }
| NOCOMPRESS}}] ;

```

|                    |                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AUTOEXTEND</b>  | L'argument active ou désactive l'extension automatique d'un nouveau fichier de données ou temporaire. Si vous omettez cette clause, ces fichiers ne seront pas automatiquement étendus.                                                                                                                                                                         |
| <b>BIGFILE</b>     | Indique que le tablespace est créé avec un seul fichier pouvant contenir jusqu'à $2^{32}$ blocks. Pour un tablespace d'un block de 32k, vous pouvez stocker jusqu'à 128Tb.                                                                                                                                                                                      |
| <b>SMALLFILE</b>   | Indique que le tablespace peut avoir un ou plusieurs fichiers de données. Aucun fichier ne peut contenir plus de $2^{22}$ blocks. Pour un tablespace d'un block de 32Kb, chaque fichier peut stocker jusqu'à 128Gb.                                                                                                                                             |
| <b>nom_fichier</b> | Le ou les fichiers de données qui constituent le tablespace.                                                                                                                                                                                                                                                                                                    |
| <b>integer</b>     | Spécifie une taille en octets si vous ne précisez pas de suffixe pour définir une valeur en K, M, G, T.                                                                                                                                                                                                                                                         |
| <b>K</b>           | Valeurs spécifiées pour préciser la taille en kilo-octets.                                                                                                                                                                                                                                                                                                      |
| <b>M</b>           | Valeurs spécifiées pour préciser la taille en méga-octets.                                                                                                                                                                                                                                                                                                      |
| <b>G</b>           | Valeurs spécifiées pour préciser la taille en giga-octets.                                                                                                                                                                                                                                                                                                      |
| <b>T</b>           | Valeurs spécifiées pour préciser la taille en téra-octets.                                                                                                                                                                                                                                                                                                      |
| <b>NEXT</b>        | Définit la taille en octets du prochain incrément d'espace disque qui doit être alloué automatiquement au fichier lorsque davantage d'espace de stockage est requis.                                                                                                                                                                                            |
| <b>MAXSIZE</b>     | Définit l'espace disque maximal autorisé pour l'extension automatique du fichier de données.                                                                                                                                                                                                                                                                    |
| <b>UNLIMITED</b>   | Définit une allocation d'espace illimitée pour le fichier.                                                                                                                                                                                                                                                                                                      |
| <b>BLOCKSIZE</b>   | Indique une taille de bloc non standard pour le tablespace.                                                                                                                                                                                                                                                                                                     |
| <b>LOGGING</b>     | Définit que la base de données effectue les journalisations pour toutes les opérations sur tous les index, tables et partitions contenus dans le tablespace. L'attribut de journalisation de niveau tablespace peut être modifié par les spécifications de journalisation au niveau table, index ou partition.                                                  |
| <b>NOLOGGING</b>   | Définit que la base de données n'effectue pas de journalisation pour les « <b>INSERT /*+APPEND*/</b> » ou les chargements des données à l'aide de SQL*Loader ainsi que les opérations DDL sur toutes les créations d'index, tables et partitions contenus dans le tablespace. Les modifications de tables avec les options « <b>MOVE</b> » ou « <b>SPLIT</b> ». |

|                      |                                                                                                                                                                                    |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>FORCE LOGGING</b> | Force le travail en mode « <b>LOGGING</b> » pour toutes les opérations de la base de données, même si l'opération concernée est effectuée dans le mode « <b>NOLOGGING</b> ».       |
| <b>ONLINE</b>        | Permet de créer un tablespace mis à la disposition des utilisateurs qui ont reçu le droit d'y accéder immédiatement après sa création. Il s'agit du choix par défaut               |
| <b>OFFLINE</b>       | Permet de créer un tablespace dans un état indisponible immédiatement après sa création.                                                                                           |
| <b>FLASHBACK</b>     | Indique que le tablespace peut être utilisé dans des opérations de récupération de type « <b>FLASHBACK</b> ».                                                                      |
| <b>ENCRYPTION</b>    | Indique que le tablespace stocke les informations cryptées dans ces fichiers. Cette option doit être accompagnée de la clause de stockage :<br>« <b>DEFAULT STORAGE ENCRYPT</b> ». |
| <b>COMPRESS</b>      | Indique que le stockage de segments dans le tablespace est compressé. Pour plus d'informations sur le stockage des segments, voir le module « La gestion du stockage ».            |



```
SYS@saphir>create tablespace tbsdata datafile
  2 size 10m autoextend on next 10m maxsize 1g,
  3 size 10m autoextend on next 10m maxsize 1g
  4 force logging online flashback on;
```

Tablespace créé.

```
SYS@saphir>select file_name
  2 from dba_data_files
  3 where tablespace_name = 'TBSDATA';
```

FILE\_NAME

```
-----
/u02/donnees/oradata/SAPHIR/datafile/o1_mf_tbsdata_63mdggwt_.dbf
/u02/donnees/oradata/SAPHIR/datafile/o1_mf_tbsdata_63mdgh16_.dbf
```

```
SYS@saphir>!ls -l /u02/donnees/oradata/SAPHIR/datafile
```

```
...
-rw-r----- 1 oracle oinstall 10493952 jui 11 13:58 o1_mf_tbsdata_63mdggwt_.dbf
-rw-r----- 1 oracle oinstall 10493952 jui 11 13:58 o1_mf_tbsdata_63mdgh16_.dbf
...
```

Dans l'exemple précédent, vous pouvez remarquer la création d'un tablespace avec deux fichiers de données, chacun d'une taille de 10M. L'espace de stockage défini pour chaque fichier de 10Mb est réservé sur disque.

### Attention



À la création du tablespace, il faut veiller à ce que les tailles des fichiers mentionnés ne dépassent pas l'espace libre qui se trouve sur disque.

En effet Oracle réserve automatiquement l'espace précisé pour chaque fichier de données. S'il n'y a pas assez d'espace sur disque, Oracle projette la création du tablespace.

Vous pouvez également définir si le tablespace créé est accessible tout de suite après la création, comme celui créé dans l'exemple précédent. Le tablespace créé « **OFFLINE** » n'est pas accessible immédiatement après sa création ; il faut modifier son état pour pouvoir travailler avec un tel type de tablespace.



```
SYS@saphir>create table table_01 tablespace tbsdata
  2 as select * from cat;
```

Table créée.

```

SYS@saphir>create tablespace tbsdata02 datafile
  2   size 10m autoextend on next 10m maxsize 1g
  3   force logging offline flashback on;

Tablespace créé.

SYS@saphir>create table table_02 tablespace tbsdata02 as
  2   select * from cat;
select * from cat
          *
ERREUR à la ligne 2 :
ORA-01542: tablespace 'TBSDATA02' hors ligne ; impossible de lui affecter de l'espace

SYS@saphir>alter tablespace tbsdata02 online;

Tablespace modifié.

SYS@saphir>create table table_02 tablespace tbsdata02 as
  2   select * from cat;

Table créée.

```

## Le tablespace par défaut

Tous les objets qui ne comportent pas, dans la syntaxe de création, les mentions de stockage dans un tablespace, sont stockés automatiquement dans le tablespace permanent par défaut.

Il faut prendre l'habitude de définir un tablespace permanent par défaut qui reçoive l'ensemble des objets créés sans mention de stockage.

Vous pouvez définir le tablespace permanent par défaut de la base de données à l'aide de la commande SQL suivante :

```
ALTER DATABASE DEFAULT TABLESPACE nom_tablespace ;
```



```

SYS@saphir>alter database default tablespace tbsdata;

Base de données modifiée.

SYS@saphir>select property_value from database_properties
  2   where property_name like 'DEFAULT_PERMANENT_TABLESPACE';

PROPERTY_VALUE
-----
TBSDATA

SYS@saphir>create table table_03 as select * from cat;

Table créée.

SYS@saphir>select table_name, tablespace_name
  2   from dba_tables
  3   where tablespace_name = 'TBSDATA';

TABLE_NAME                                TABLESPACE
-----
TABLE_01                                TBSDATA

```



La vue « **DATABASE\_PROPERTIES** » permet d'afficher toutes les informations sur les propriétés par défaut d'une base de données comme : le nom global de la base de données, le tablespace permanent, le tablespace temporaire, le type de tablespace, le fuseau horaire, la langue, le territoire, ...

Le tablespace permanent par défaut peut être défini lors de la création de la base de données, grâce à l'argument « **DEFAULT TABLESPACE** ».



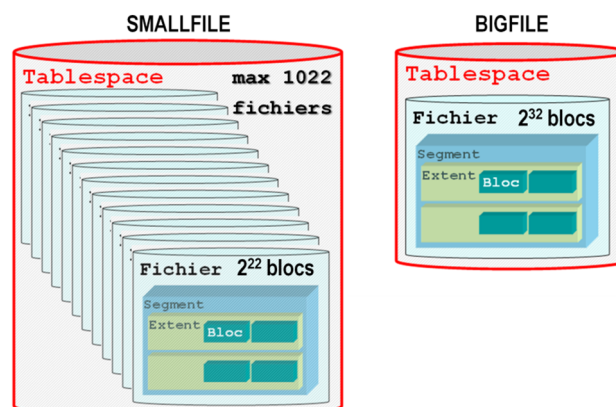
```
SQL> create database "tpdba"
...
7  datafile size 300m autoextend on
8      next 10240k
9      maxsize unlimited
10     extent management local
11  sysaux datafile size 120m autoextend on
12      next 10240k
13      maxsize unlimited
14  default temporary tablespace temp tempfile size 20m
15      autoextend on next 640k maxsize unlimited
16  undo tablespace "UNDOTBS1" datafile size 200m
17      autoextend on next 5120k maxsize unlimited
16  default tablespace app_users datafile size 200m
17      autoextend on next 5120k maxsize unlimited
...
```

Comme vous pouvez le remarquer dans l'exemple précédent, la création de la base de données permet de créer en plus des quatre tablespaces « **SYSTEM** », « **SYSAUX** », « **UNDOTBS1** » et « **TEMP** », un autre tablespace permanent « **APP\_USERS** » pour devenir le tablespace permanent par défaut de la base.

## Le tablespace BIGFILE

Le tablespace de type « **SMALLFILE** » peut avoir plusieurs fichiers de données, au maximum **1022** fichiers, qui peuvent chacun contenir jusqu'à **2<sup>22</sup>** blocs des données. Ainsi pour un tablespace construit avec un bloc de **32Kb**, vous pouvez avoir un espace de stockage par fichier de **128Gb**. La taille maximale de stockage pour un tablespace « **SMALLFILE** » est d'approximativement **128Tb** (il faut **1024** fichiers pour une taille exacte de **128Tb**).

L'inconvénient de ce type de tablespace, c'est que lors d'un point de contrôle (checkpoint), le processus « **DBWn** » écrit les blocs modifiés dans les fichiers de données correspondants et le processus « **CKPT** » met à jour l'en-tête de tous les fichiers de la base de données qui sont en mode lecture écriture. Ainsi plus il y a de fichiers plus les checkpoints durent longtemps même si ces fichiers n'ont pas de blocs de données modifiés.



Par ailleurs, quand vous utilisez un gestionnaire de volume logique comme Oracle ASM, les fichiers dans ces architectures sont uniquement des informations logiques.

Le tablespace type « **BIGFILE** » est un tablespace avec un seul fichier de données qui peut contenir jusqu'à  $2^{32}$  blocs de données. Ainsi un fichier de données pour un tablespace construit avec des blocs de **32Kb** peut stocker jusqu'à **128Tb**. Ainsi ce type de tablespace réduit le nombre de fichiers de données en occurrence avec le temps des checkpoints.

Lorsque vous créez votre base de données, vous pouvez spécifier le type de tablespace par défaut que vous souhaitez, à l'aide de la syntaxe :

```
... SET DEFAULT {BIGFILE | SMALLFILE} TABLESPACE ...
```

Sans précision explicite, le type du tablespace « SYSTEM » est considéré comme le type par défaut. Après la création de la base de données vous pouvez également changer le type par défaut à l'aide de la syntaxe :

```
ALTER DATABASE SET DEFAULT {BIGFILE | SMALLFILE} TABLESPACE ;
```

La vue « **DATABASE\_PROPERTIES** » vous permet d'afficher le type de tablespace par défaut de votre base de données.



```
SYS@rubis>alter database set default bigfile tablespace;
```

Base de données modifiée.

```
SYS@rubis>select property_name, property_value
2   from database_properties
3   where property_name = 'DEFAULT_TBS_TYPE';
```

| PROPERTY_NAME    | PROPERTY_VALUE |
|------------------|----------------|
| DEFAULT_TBS_TYPE | BIGFILE        |

```
SYS@rubis>create tablespace tbsbig datafile
2   size 10m autoextend on next 10m;
```

Tablespace créé.

```
SYS@rubis>create tablespace tbssmall datafile
2   size 10m autoextend on next 10m ,
3   size 10m autoextend on next 10m;
```

```
create tablespace tbssmall datafile
```

\*

ERREUR à la ligne 1 :

ORA-32774: plusieurs fichiers ont été indiqués pour le tablespace BIGFILE TBSSMALL

```
SYS@rubis>create smallfile tablespace tbssmall datafile
2   size 10m autoextend on next 10m ,
3   size 10m autoextend on next 10m;
```

Tablespace créé.

```
SYS@rubis>select tablespace_name, bigfile, file_name
2   from dba_data_files join
3        dba_tablespaces using ( tablespace_name)
4   where tablespace_name in ( 'TBSBIG', 'TBSSMALL');
```

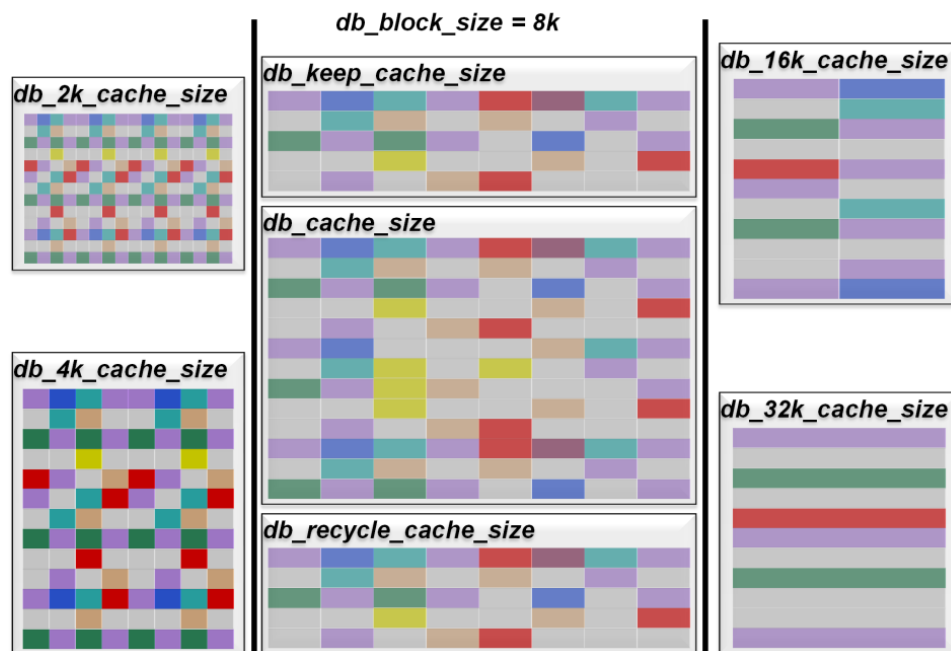
| TABLESPACE | BIG | FILE_NAME                                         |
|------------|-----|---------------------------------------------------|
| TBSBIG     | YES | +GD_DONNEES/rubis/datafile/tbsbig.330.724090169   |
| TBSSMALL   | NO  | +GD_DONNEES/rubis/datafile/tbssmall.329.724090417 |
| TBSSMALL   | NO  | +GD_DONNEES/rubis/datafile/tbssmall.328.724090419 |



Dans l'exemple précédent, vous pouvez voir la configuration de la base de données pour créer par défaut des tablespaces de type « **BIGFILE** ». Par la suite, la création du tablespace « **TBSBIG** » est effectuée avec succès mais la création d'un tablespace « **TBSSMALL** » avec plusieurs fichiers échoue. En précisant qu'il s'agit d'un tablespace de type « **SMALLFILE** » le tablespace est cette fois-ci créé.

## La taille du bloc

Pour lire ou écrire les données de la base de données, Oracle charge d'abord les blocs correspondants dans le buffer cache (cache de tampon). Ainsi il faut d'abord paramétrer le buffer cache pour pouvoir recevoir les blocs respectifs.



Le paramètre qui vous permet de réserver de l'espace pour les blocs différents des blocs par défaut est : **DB\_nK\_CACHE\_SIZE**.

Les valeurs autorisées pour le « **n** » sont 2, 4, 8, 16 et 32 (certains systèmes d'exploitation ne supportent pas cette valeur). La valeur du bloc par défaut est « **DB\_BLOCK\_SIZE** », paramètre défini à la création de la base de données.



```
SYS@rubis>show parameter db_block_size
```

| NAME          | TYPE    | VALUE |
|---------------|---------|-------|
| db_block_size | integer | 8192  |

```
SYS@rubis>show parameter k_cache_size
```

| NAME              | TYPE        | VALUE |
|-------------------|-------------|-------|
| db_16k_cache_size | big integer | 0     |
| db_2k_cache_size  | big integer | 0     |
| db_32k_cache_size | big integer | 0     |
| db_4k_cache_size  | big integer | 0     |
| db_8k_cache_size  | big integer | 0     |

```
SYS@rubis>alter system set db_32k_cache_size=16m;
```

Système modifié.

```
SYS@rubis>alter system set db_4k_cache_size=16m;
```

Système modifié.

```
SYS@rubis>show parameter k_cache_size
```

| NAME              | TYPE        | VALUE |
|-------------------|-------------|-------|
| db_16k_cache_size | big integer | 0     |
| db_2k_cache_size  | big integer | 0     |
| db_32k_cache_size | big integer | 16M   |
| db_4k_cache_size  | big integer | 16M   |
| db_8k_cache_size  | big integer | 0     |

L'exemple précédant montre le paramétrage de la mémoire pour pouvoir créer des tablespaces qui utilisent une taille de bloc de 4Kb et 16Kb en plus de la taille par défaut de 8Kb.



```
SYS@rubis>create tablespace data_4k datafile size 10m autoextend on
2 next 10m , blocksize 4k;
```

Tablespace créé.

```
SYS@rubis>create tablespace data_32k datafile size 10m autoextend on
2 next 10m blocksize 32k;
```

Tablespace créé.

```
SYS@rubis>create tablespace data_2k datafile size 10m autoextend on
2 next 10m blocksize 2k;
```

```
create tablespace data_2k datafile
*
```

ERREUR à la ligne 1 :

ORA-29339: la taille de bloc de tablespace 2048 ne correspond pas aux tailles de blocs configurés

```
SYS@rubis>select tablespace_name, block_size/1024 "bs K", file_name
2 from dba_data_files join dba_tablespaces using ( tablespace_name)
3 where tablespace_name like 'DATA%K';
```

TABLESPAC bs K FILE\_NAME

```
DATA_4K      4 +GD_DONNEES/rubis/datafile/data_4k.327.724092783
DATA_32K     32 +GD_DONNEES/rubis/datafile/data_32k.326.724092799
```

Les deux premières créations de tablespace sont acceptées par la base de données, et le paramétrage du buffer cache est déjà effectué. Par contre la dernière création est rejetée ; il faut d'abord paramétrer le buffer cache pour pouvoir recevoir les blocs de 2Kb.



```
SYS@rubis>create table tdata_32k tablespace data_32k as select * from cat;
```

Table créée.

```
SYS@rubis>alter system set db_32k_cache_size=0;
```

Système modifié.

```
SYS@rubis>select * from tdata_32k;
select * from tdata_32k
*
```

ERREUR à la ligne 1 :

ORA-00379: aucun tampon libre dans le pool de tampons DEFAULT pour la taille de bloc (32 ko)

```

SYS@rubis>alter system set db_32k_cache_size=16m;

Système modifié.

SYS@rubis>select count(*) from tdata_32k;

COUNT(*)
-----
4854

```



## Attention

Pour lire ou écrire les données de la base de données, Oracle charge d'abord les blocs correspondants dans le buffer cache (cache de tampon).

Si une des zones d'une taille de bloc non standard vient d'être mise à zéro, alors tous les objets des tablespaces qui ont cette taille de bloc ne peuvent plus être lus, comme c'est le cas dans l'exemple précédent.

Une fois que la zone mémoire a été initialisée, les données sont à nouveau accessibles.

## Le cryptage transparent

Pour pouvoir mettre en place le cryptage transparent, il faut d'abord configurer « Oracle Encryption Wallet » s'il n'a pas déjà été initialisé pour votre base de données.

Initialiser consiste à créer un fichier « **WALLET** » portefeuille dans le répertoire par défaut pour ce fichier de la base de données :

\$ORACLE\_BASE/admin/SID/wallet ou %ORACLE\_BASE%\admin\SID\wallet

Le répertoire doit exister, sinon la création par défaut du fichier est impossible.

Vous pouvez visualiser l'emplacement du fichier « **WALLET** » portefeuille et le statut en interrogeant la vue « **V\$ENCRYPTION\_WALLET** ».

La syntaxe de création du portefeuille est la suivante :

```
ALTER SYSTEM SET ENCRYPTION KEY IDENTIFIED BY "mot de passe";
```

**mot de passe**

Le mot de passe pour l'activation du cryptage une fois le fichier portefeuille créé.

Vous pouvez visualiser les informations concernant les algorithmes de cryptage en interrogeant la vue « **V\$RMAN\_ENCRYPTION\_ALGORITHMS** ».



```

SYS@topaze>select algorithm_name "Nom",
2 algorithm_description "Description",is_default "Déf"
3 from v$rman_encryption_algorithms;

```

| Nom    | Description     | Déf |
|--------|-----------------|-----|
| AES128 | AES 128-bit key | YES |
| AES192 | AES 192-bit key | NO  |
| AES256 | AES 256-bit key | NO  |

```

SYS@topaze>!mkdir /u01/app/oracle/admin/topaze/wallet
...
drwxr-xr-x. 2 oracle oinstall 4096 8 mars 18:51 wallet

```

<-----

```
drwxr-x---. 2 oracle oinstall 4096 7 mars 12:08 xdb_wallet

SYS@topaze>select wrl_type, wrl_parameter, status from v$encryption_wallet;

WRL_TYPE          WRL_PARAMETER          STATUS
-----
FILE              /u01/app/oracle/admin/topaze/wallet  NOT_AVAILABLE

SYS@topaze>alter system set encryption key identified by "Razvanpwd3";

Système modifié.

SYS@topaze>select wrl_type, wrl_parameter, status from v$encryption_wallet;

WRL_TYPE          WRL_PARAMETER          STATUS
-----
FILE              /u01/app/oracle/admin/topaze/wallet  OPEN

SYS@topaze>!ls -la /u01/app/oracle/admin/topaze/wallet
...
-rw-r--r--. 1 oracle oinstall 3112 8 mars 18:58 ewallet.p12
```

## Attention

Il est impératif, après chaque démarrage de la base, d'ouvrir le fichier portefeuille pour pouvoir bénéficier des fonctionnalités de cryptage.

La syntaxe d'ouverture ou de la fermeture du fichier portefeuille est la suivante :

```
ALTER SYSTEM SET WALLET {OPEN|CLOSE} IDENTIFIED BY "mot de passe";
```

```
SYS@topaze>startup force
Instance ORACLE lancée.
...
SYS@topaze>select wrl_type, wrl_parameter, status from v$encryption_wallet;

WRL_TYPE          WRL_PARAMETER          STATUS
-----
FILE              /u01/app/oracle/admin/topaze/wallet  CLOSED

SYS@topaze>alter system set wallet open identified by "Razvanpwd3";

Système modifié.

SYS@topaze>create tablespace tbs_ne datafile size 256k;

Tablespace créé.

SYS@topaze>create tablespace tbs_e datafile size 256k
2 encryption using 'AES256' default storage ( encrypt);

Tablespace créé.

SYS@topaze>create table nom01 ( nom varchar2(50)) tablespace tbs_ne;

Table créée.

SYS@topaze>create table nom02 ( nom varchar2(50)) tablespace tbs_e;

Table créée.
```

```

SYS@topaze>insert all into nom01 into nom02 select 'RazvanBIZOI' from dual;

2 ligne(s) créée(s).

SYS@topaze>commit;

Validation effectuée.

SYS@topaze>select * from nom01, nom02

NOM                                NOM
-----
RazvanBIZOI                      RazvanBIZOI

SYS@topaze>alter system set wallet close identified by "Razvanpwd3";

Système modifié.

SYS@topaze>select * from nom02;
select * from nom02
                *
ERREUR à la ligne 1 :
ORA-28365: le portefeuille n'est pas ouvert

SYS@topaze>select * from nom01;

NOM
-----
RazvanBIZOI

1 ligne sélectionnée.

SYS@topaze>select file_name from dba_data_files join
  2  dba_tablespaces using ( tablespace_name)
  3  where tablespace_name like 'TBS%E';

FILE_NAME
-----
/u01/app/oracle/oradata/TOPAZE/datafile/o1_mf_tbs_e_9kppzw1w_.dbf
/u01/app/oracle/oradata/TOPAZE/datafile/o1_mf_tbs_ne_9kppz1q1_.dbf

SYS@topaze>!mv /u01/app/oracle/admin/topaze/wallet/ewallet.p12
/u01/app/oracle/admin/topaze/wallet/ewallet.p12.sav

SYS@topaze>alter system set wallet open identified by "Razvanpwd3";
alter system set wallet open identified by "Razvanpwd3"
*
ERREUR à la ligne 1 :
ORA-28367: le portefeuille n'existe pas

SYS@topaze>alter system set encryption key identified by "Razvanpwd3";
alter system set encryption key identified by "Razvanpwd3"
*
ERREUR à la ligne 1 :
ORA-28362: clé maître introuvable

SYS@topaze>alter system set wallet close identified by "Razvanpwd3";

```

Système modifié.

```
SYS@topaze>mv /u01/app/oracle/admin/topaze/wallet/ewallet.p12.sav
/u01/app/oracle/admin/topaze/wallet/ewallet.p12
```

```
SYS@topaze>alter system set wallet open identified by "Razvanpwd3";
```

Système modifié.

```
SYS@topaze>!find /u01/app/oracle/oradata/TOPAZE/datafile/o1_mf_tbs_* | xargs grep
'Razvan' -sl
/u01/app/oracle/oradata/TOPAZE/datafile/o1_mf_tbs_ne_9kppz1q1_.dbf
```

```
SYS@topaze>!grep -lR "Razvan" /u01/app/oracle/oradata/TOPAZE/datafile/o1_mf_tbs_*
/u01/app/oracle/oradata/TOPAZE/datafile/o1_mf_tbs_ne_9kppz1q1_.dbf
```

L'utilisation du fichier portefeuille permet de créer des tablespaces cryptés. Ainsi le tablespace **TBS\_NE** est un tablespace classique qui contient une table avec un seul champ et un seul enregistrement avec la valeur 'RazvanBIZOI'. Le deuxième tablespace **TBS\_E** est un tablespace crypté contenant également une table avec un seul champ et un seul enregistrement. Évidemment si vous perdez le fichier portefeuille vous n'avez plus aucune possibilité de lire les données stockées dans les tablespaces.



```
SYS@topaze>startup force
[oracle@terra ~]$ find /u01/app/oracle/oradata/TOPAZE/datafile/o1_mf_tbs_* | xargs
grep 'Razvan' -sl
/u01/app/oracle/oradata/TOPAZE/datafile/o1_mf_tbs_ne_9kppz1q1_.dbf
[oracle@terra ~]$ grep -lR "Razvan"
/u01/app/oracle/oradata/TOPAZE/datafile/o1_mf_tbs_*
/u01/app/oracle/oradata/TOPAZE/datafile/o1_mf_tbs_ne_9kppz1q1_.dbf
```

Dans l'environnement Linux/Unix vous pouvez utiliser la commande « **find** » ou « **grep** » pour rechercher la chaîne 'Razvan' dans le fichier binaire. Dans le cas d'un tablespace crypté il n'est pas possible de trouver l'information.

## Le tablespace temporaire

Lors d'importantes opérations de tri (telles que select distinct, union et create index) Oracle a besoin de stocker dans la base de données, des informations concernant le tri des enregistrements, avant de retourner l'information aux utilisateurs. En raison de leur nature dynamique, ces espaces de tri ne devraient pas être stockés avec d'autres types de données.

Lorsqu'un tablespace temporaire « **TEMPORARY** » est défini, un segment de tri est aussi créé. Celui-ci est capable de croître si nécessaire afin de pouvoir héberger toutes les opérations de tri de données, et existe jusqu'à ce que la base de données soit fermée puis redémarrée.

### Attention

Vous ne pouvez pas créer de tablespaces temporaires avec une taille de bloc différente de la taille du bloc par défaut de la base de données. Les tablespaces temporaires ainsi que les tablespaces système sont toujours créés avec une taille de bloc égale au bloc par défaut de la base de données.



La syntaxe de création d'un tablespace temporaire est :

```
CREATE {BIGFILE|SMALLFILE} TEMPORARY TABLESPACE nom_tablespace
[ TEMPFILE ['nom_fichier'] [ SIZE integer {K|M|G|T} ]
[ AUTOEXTEND {OFF | ON [ NEXT integer {K|M|G|T}]}
[ MAXSIZE {UNLIMITED | integer {K|M|G|T}}] [, ...]]
[ {ONLINE | OFFLINE} ] ;
```

### TEMPORARY

Indique que le tablespace est de type temporaire.

### TEMPFILE

Indique qu'il s'agit d'un fichier temporaire et pas d'un fichier de données.

### REUSE

Définit la réutilisation du fichier temporaire s'il existe déjà.



```
SYS@jaspe>create bigfile temporary tablespace temp01 tempfile
2 size 10m autoextend on next 10m ;
```

Tablespace créé.

```
SYS@jaspe>create bigfile temporary tablespace temp02 tempfile
2 size 10m autoextend on next 10m ;
```

Tablespace créé.

```
SYS@jaspe>create bigfile temporary tablespace temp03 tempfile
2 size 10m autoextend on next 10m ;
```

Tablespace créé.

```
SYS@jaspe>select tablespace_name, file_name from dba_temp_files join
2 dba_tablespaces using ( tablespace_name)
3 where tablespace_name like 'TEMP0_';
```

TABLESPAC FILE\_NAME

```
-----
TEMP01      +GD_DONNEES/jaspe/tempfile/temp01.283.724094305
TEMP02      +GD_DONNEES/jaspe/tempfile/temp02.281.724094309
TEMP03      +GD_DONNEES/jaspe/tempfile/temp03.280.724094315
```

Dans l'exemple précédent, vous pouvez observer la création de trois tablespaces temporaires de type « **BIGFILE** ». Les fichiers temporaires ainsi créés peuvent s'agrandir automatiquement par des tranches de 10Mb.



```
SYS@ambre>create temporary tablespace temp04
2 tempfile 'W:\DONNEES\ORADATA\AMBRE\DATAFILE\TEMP04.TMP'
3 size 10m autoextend on next 10m;
```

Tablespace créé.

```
SYS@ambre>drop tablespace temp04;
```

Tablespace supprimé.

```
SYS@ambre>create temporary tablespace temp04
2 tempfile 'W:\DONNEES\ORADATA\AMBRE\DATAFILE\TEMP04.TMP'
3 size 10m autoextend on next 10m;
create temporary tablespace temp04
*
```

ERREUR à la ligne 1 :

ORA-01119: échec de création du fichier de base de données  
'W:\DONNEES\ORADATA\AMBRE\DATAFILE\TEMP04.TMP'

ORA-27038: le fichier créé existe déjà

OSD-04010: option <create> indiquée ; le fichier existe déjà

```
SYS@ambre>create temporary tablespace temp04
2 tempfile 'W:\DONNEES\ORADATA\AMBRE\DATAFILE\TEMP04.TMP'
3 size 20m reuse autoextend on next 10m;
```

Tablespace créé.

```

SYS@ambre>$dir w:\DONNEES\ORADATA\AMBRE\DATAFILE\TEMP04.TMP
Le volume dans le lecteur W s'appelle Nouveau nom
Le numéro de série du volume est C051-C97C

Répertoire de w:\donnees\oradata\AMBRE\DATAFILE

11/07/2010  18:03          20 979 712 TEMP04.TMP
...

```

L'exemple précédent commence par la création d'un tablespace « **TEMP04** ». Ensuite on détruit le tablespace sans effacer le fichier temporaire. Vous pouvez remarquer que la création d'un tablespace ne peut pas être effectuée si le fichier correspondant existe déjà, et cela est valable pour tous les tablespaces. Vous pouvez utiliser l'argument « **REUSE** » pour demander à Oracle de réutiliser le fichier existant sur disque. En effet Oracle réutilise le nom du fichier temporaire et il le dimensionne au volume demandé.

Cette démarche de rattachement d'un fichier temporaire est utilisée dans le cas de perte du fichier de contrôle et la création à l'aide du script de sauvegarde.

## Le groupe tablespaces temporaires

Certains utilisateurs d'une base de données Oracle peuvent avoir besoin de volumes de stockage temporaire beaucoup plus grands que ceux de tous les autres utilisateurs de l'application. Dans ce cas, vous pouvez créer plusieurs tablespaces temporaires « **TEMPORARY** » pour distribuer les espaces de stockages des utilisateurs ayant des besoins semblables sur les mêmes tablespaces.

**10g**

À partir de la version Oracle 10g, il est possible de créer des groupes de tablespaces temporaires pour équilibrer les charges et diminuer les contentions d'accès aux segments temporaires. Un groupe de tablespaces temporaires est géré par la base de données comme un tablespace temporaire ; vous pouvez ainsi définir que c'est le tablespace temporaire par défaut.

La syntaxe de création d'un groupe de tablespaces temporaires est en effet uniquement un assignement d'un tablespace temporaire existant au nom du groupe qui va être créé, comme suit :

```
ALTER TABLESPACE nom TABLESPACE GROUP nom_groupe;
```

```
SYS@jaspe>alter tablespace temp01 tablespace group ngrp_temp;
```

Tablespace modifié.

```
SYS@jaspe>alter tablespace temp02 tablespace group ngrp_temp;
```

Tablespace modifié.

```
SYS@jaspe>alter tablespace temp03 tablespace group ngrp_temp;
```

Tablespace modifié.

```
SYS@jaspe>select * from dba_tablespace_groups;
```

| GROUP_NAME | TABLESPAC |
|------------|-----------|
| NGRP_TEMP  | TEMP01    |
| NGRP_TEMP  | TEMP02    |
| NGRP_TEMP  | TEMP03    |

Les trois tablespaces temporaires ont été ajoutés au groupe « **NGRP\_TEMP** » ; à présent le nom groupe peut être utilisé à la place du nom d'un tablespace temporaire. Pour retirer un tablespace d'un groupe de tablespaces temporaires il suffit d'utiliser la syntaxe suivante :

```
ALTER TABLESPACE nom TABLESPACE GROUP ' ' ;
```





Vous pouvez maintenant définir que le groupe de tablespaces temporaires est le tablespace temporaire par défaut de la base.

```
SYS@jaspe>alter database default temporary tablespace ngrp_temp;
```

Base de données modifiée.

```
SYS@jaspe>select property_value from database_properties  
2 where property_name like 'DEFAULT_TEMP_TABLESPACE';
```

PROPERTY\_VALUE

-----

NGRP\_TEMP

Dans l'exemple précédent, vous pouvez remarquer la définition du tablespace temporaire par défaut avec l'utilisation d'un groupe de tablespaces « **NGRP\_TEMP** », et l'utilisation de la vue « **DATABASE\_PROPERTIES** » pour retrouver la valeur du tablespace temporaire par défaut de la base de données.

## Le tablespace undo

Toutes les données d'annulation sont stockées dans un tablespace spécial appelé « **UNDO** ». Lorsque vous créez un tablespace « **UNDO** », Oracle gère le stockage, la rétention et l'emploi de l'espace pour les données de rollback par l'intermédiaire de la fonction SMU (System-Managed Undo). Aucun objet permanent n'est placé dans le tablespace undo.

Pour pouvoir créer un tablespace undo à la création de la base de données, il faut prendre soin d'initialiser le paramètre

**UNDO\_MANAGEMENT=AUTO**

À partir de la version Oracle 9i, il est fortement conseillé d'utiliser la gestion automatique des segments d'annulation. À partir de la version Oracle 10g, il est impératif d'utiliser ce mode gestion, sans quoi une grande partie des fonctionnalités de la base de données sont inutilisables. Voir le Module « Les segments UNDO ».

Rappelez-vous, la syntaxe SQL de création de la base de données comporte d'abord la création d'un tablespace « **SYSTEM** » et d'un tablespace « **SYSAUX** », ainsi que la création d'un tablespace « **TEMP** » et « **UNDO** ».

### Attention



À la création de la base de données, si vous ne précisez pas de tablespace undo, Oracle crée un tablespace undo appelé « **SYS\_UNDOTS** ».

Le paramètre « **UNDO\_TABLESPACE** » doit avoir la valeur « **SYS\_UNDOTS** », sinon la base de données ne pourra pas être créée.

La syntaxe de création d'un tablespace UNDO est :

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

**UNDO**

Indique que le tablespace est de type undo.

Attention, les tablespaces pérenne et temporaire par défaut sont attribués à l'aide de la syntaxe « **ALTER DATABASE DEFAULT ...** » ; ce n'est pas le cas pour le tablespace de type « **UNDO** ». Le tablespace « **UNDO** » par défaut est celui qui est précisé dans le paramètre « **UNDO\_TABLESPACE** ».



```
SYS@rubis>show parameter undo_tablespace
```

| NAME            | TYPE   | VALUE    |
|-----------------|--------|----------|
| undo_tablespace | string | UNDOTBS1 |

```
SYS@rubis>create bigfile undo tablespace undo datafile
2 size 512m autoextend on next 512m;
```

Tablespace créé.

```
SYS@rubis>select contents, file_name
2 from dba_data_files join
3 dba_tablespaces using ( tablespace_name)
4 where tablespace_name like 'UNDO';
```

| CONTENTS | FILE_NAME                                     |
|----------|-----------------------------------------------|
| UNDO     | +GD_DONNEES/rubis/datafile/undo.325.724104703 |

```
SYS@rubis>alter system set undo_tablespace =undo;
```

Système modifié.

```
SYS@rubis>show parameter undo_tablespace
```

| NAME            | TYPE   | VALUE |
|-----------------|--------|-------|
| undo_tablespace | string | UNDO  |



## Attention

Comme pour les tablespaces système et pour les tablespaces temporaires, le tablespace **UNDO** doit être créé avec la taille de bloc par défaut.

La base de données Oracle utilise pour ses fichiers de gestion la taille de bloc par défaut définie lors de la création de cette base de données.