Oracle

Exploitation

Auteur : Clotilde Garde
Société TELLORA
Version 2.2
Du 6 Avril 2017

Sommaire

1 Présentation 7
1.1 Les produit Database proposes par Oracle 7
1.2 Notion de Grid Computing 9
1.2.1 La gestion ASM (gestionnaire de fichiers : Automatic Storage Management) 10
1.2.2 Les composants développés par Oracle pour le Grid Computing 11
1.2.3 Outils de développement 11
1.3 Règles de nommage dans Oracle Database 11
2 La documentation 12
2.1 Le support oracle 12
3 Notion de schéma 14
4 Le dictionnaire de données 15
4.1 Les tables et vues statiques 15
4.2 Les tables et vues dynamiques de performance 16
5 Outils d'administration 17
5.1 L'outil SQL*Plus 17
5.1.1 Environnement de travail 18
5.1.2 Quelques commandes SQL*Plus 18
5.2 L'outil *i*SQL*Plus 19
5.3 Le Database Control et le Grid control 20
6 L'architecture OFA 23
7 Installation Oracle 25
7.1 Pré-requis matériel 25
7.2 Installation du client 27
8 Architecture Oracle 28
8.1 Connexion utilisateur 30
8.1.1 La PGA (Program Global Area) 30
8.1.2 La SGA: System Global Area 30
8.2 Le fichier de paramètres (init.ora ou SPFILE.ORA) 32
8.3 Les processus d'arrière plan 33
8.4 La base de données 34
8.4.1 Les fichiers de données 34
9 Utilisateurs et connexion à la base de données 36
9.1 Syntaxe pour la connexion classique 36
9.2 Syntaxe pour la connexion spéciale SYSDBA ou SYSOPER 36
9.3 Les connexions SYSDBA et SYSOPER 37
9.4 Le fichier de mots de passe 37
9.5 Les variables d'environnement 38
10 Démarrer & Arrêter une base de données 39
10.1 Démarrer la base de données 40
10.2 Modifier la disponibilité de la base de données 41
10.3 Arrêter la base de données 42
10.4 Ouvrir la base de données en mode RESTRICT 43

10.5 Mettre l'instance dans un état QUIESCE 44
10.6 Vues du dictionnaire de données 44
11 Gestion de l'instance et SPFILE 46
11.1 Créer le fichier du paramètre SPFILE 46
11.2 Exporter un fichier de paramètres serveur SPFILE 47
11.3 Modifier des paramètres de l'instance ou du SPFILE 48
11.4 Vues du dictionnaire de données 49
12 Jeux de caractères et paramètres NLS 50
12.1 Introduction 50
12.2 Migration de jeux de caractères 53
12.2.1 Migration du jeu de caractères par EXPORT/IMPORT 54
Vues du dictionnaire de données 55
13 Créer une base de données 56
13.1 Présentation du script de création de la base 57
13.2 Présentation de l'outil DBCA 60
13.3 Valeurs des paramètres 63
13.4 Vues du dictionnaire de données 66
13.5 EMCA : Création de l'OEM repository (Database Control) 66
14 Automatiser le démarrage de la base 69
14.1 Sous unix 69
14.2 Sous Windows 69
15 Accéder à une base distante 71
15.1 Configuration coté serveur 72
15.2 Configuration coté client 74
15.3 Changer de machine automatiquement 75
15.4 EZCONNECT 75
15.5 Bases distantes et database Links 76
16 Sécuriser la base de données 78
16.1 Le fichier de contrôle 78
16.2 Protection du fichier de contrôle 80
16.2.1 Multiplexer le fichier de contrôle 81
16.3 Vues du dictionnaire de données 82
16.4 Protection des fichiers de Redo Log 82
16.4.1 Dimensionner les fichiers de Redo Log 83
16.4.2 Multiplexer les fichiers de Redo Log 84
16.4.3 Ajouter un groupe de Redo Log 85
16.4.4 Déplacer les fichiers de Redo Log 85
16.4.5 Supprimer un groupe de fichiers redo log 86
16.4.6 Supprimer un membre d'un groupe de redo log 86
16.4.7 Forcer le basculement du groupe courant 87
16.4.8 Trouver des informations sur les fichiers Redo Log 88
17 Gestion du stockage 90
17.1 Notion de tablespace 90
17.2 Organisation du stockage dans un tablespace 92
17.3 Notion de BIGFILE ou de SMALLFILE 94
18 Tablespaces permanents 96
18.1 Créer un tablespace permanent 96

18.2 Modifier un tablespace permanent 97
18.2.1 Agrandir un tablespace 97
18.2.2 Passer un tablespace OFFLINE ou ONLINE : 98
18.2.3 Passer un tablespace READ ONLY ou READ WRITE : 98
18.2.4 Déplacer un fichier de données 99
18.2.5 Renommer un tablespace 100
18.2.6 Supprimer un tablespace 102
18.2.7 Créer un tablespace avec une taille de bloc non standard 103
18.2.8 Le cryptage des données d'un tablespace (nouveauté 11g) 103
18.2.9 Tablespace de travail par défaut 104
18.2.10 Vues du dictionnaire de données 105
19 Tablespaces SYSTEM & SYSAUX 106
19.1 Tablespace SYSTEM 106
19.2 Tablespace SYSAUX 106
19.2.1 Avantages du tablespace SYSAUX 107
19.2.2 Délocaliser les occupants du tablespace SYSAUX 107
20 Tablespace UNDO 109
20.1 Fonctionnement du tablespace UNDO 111
20.2 Positionner les paramètres de gestion automatique 112
20.3 Créer un tablespace UNDO 112
20.4 Changer de tablespace UNDO actif 113
20.4.1 Créer un tablespace UNDO après création de la base 113
20.4.2 Changer de tablespace UNDO pendant l'activité de la base 114
20.5 Administrer un tablespace UNDO 115
20.5.1 Dimensionner le tablespace UNDO 116
20.5.2 Supprimer un tablespace UNDO 117
20.6 Vues du dictionnaire de données 117
21 Tablespaces temporaires 120
21.1 Créer un tablespace temporaire 122
21.2 Groupes de tablespaces temporaires 122
21.3 Administrer les tablespaces temporaires 123
21.3.1 Agrandir un tablespace temporaire 124
21.3.2 Modifier la clause AUTOEXTEND : 124
21.3.3 Modifier la taille d'un fichier temporaire 125
21.3.4 Rétrécir un tablespace temporaire (Nouveautés 11g) 125
21.3.5 Supprimer un tablespace temporaire 126
21.4 Définir un tablespace temporaire par défaut 126
21.5 Vues du dictionnaire de données 127
22 Monitoring de l'utilisation d'un tablespace 128
22.1 Configuration des seuils de tablespace 129
23 Mémoire dynamique et performances 131
23.1 La notion de granule 131
23.2 Gestion automatique du partage de la mémoire en 10g 131
23.2.1 Principes de tuning de la SGA 132
23.2.2 SGA_TARGET et le Database Control (OEM) 133
23.2.3 Configuration manuelle de SGA_TARGET 134
23.2.4 Comportement des paramètres Auto-tuned 134
23.2.5 Comportement des paramètres Manuels 135
23.2.6 Redimensionner SGA_TARGET 136

23.2.7 Désactiver la gestion automatique de la mémoire en version 10g 136
23.3 Gestion automatique de la mémoire en 11g 136
23.3.1 Désactiver la gestion automatique de la mémoire en version 11g 137
23.3.2 La vue dynamique V$MEMORY_TARGET_ADVICE 137
23.3.3 Nouveau cache en version 11g : `result_cache` 137
23.4 L'optimiseur Oracle 137
23.4.1 Les optimiseurs RBO et CBO 138
23.4.2 Présentation du SQL Tuning Advisor 139
23.4.3 Impacte sur les Statistiques 140
23.5 L'optimiseur Oracle et la gestion des statistiques 141
23.5.1 Statistiques sur les tables 142
23.5.2 Interpréter les statistiques générées sur les tables 143
23.5.3 Statistiques sur les index 144
23.5.4 Problèmes détectés sur les index 145
23.6 Outil de collecte des statistiques 145
23.6.1 GATHER_STATS_JOB 146
23.6.2 Modifier l'exécution des statistiques 147
23.7 Automatic Database Diagnostic Monitor (ADDM) 147
23.7.1 Méthode d'analyse utilisée par ADDM 149
23.7.2 Résultats de l'analyse ADDM dans le grid Control 150
23.7.3 Recommandations d'ADDM 151
23.7.4 Nouvelles vues en version 11g pour ADDM 151
23.7.5 Exemple de génération de rapport ADDM 152
24 Présentation de l'utilitaire DATA Pump 157
24.1 Opérations d'IMPORT et d'EXPORT du DATA Pump 158
24.2 Avantages de l'export et de l'import `DATA Pump`: 159
24.3 Le mode intéractif du DATA Pump 160
24.3.1 Commandes du mode interactif 160
24.4 Méthode d'extraction des données avant et aprés *`data pump`* 161
24.4.1 Méthode direct path (chemin direct) du *`data pump`* 161
24.4.2 Données conduisant un accès utilisant des tables externes : 162
25 Export/Import Data Pump 163
25.1 Fichiers supportés par les outils *`DATA Pump`* 163
25.2 Paramètres le l'export et de l'import DATA Pump 164
25.2.1 Paramètres communs 164
25.2.2 Paramètres de l'EXPORT DATA Pump 166
25.2.3 Paramètres de l'IMPORT DATA Pump 166
25.3 Filtrer les données à exporter 166
25.4 Exemples d'export et d'import DATA Pump 168
25.4.1 Estimation de la taille de l'Export 168
25.4.2 Exports Parallélisés 168
25.4.3 Import Parallélisé 169
25.4.4 Export de schéma 169
25.4.5 Import de schéma 170
25.5 Remarques et modes opératoires 170
25.5.1 Export et jeux de caractères 170
25.5.2 Remarques sur les dépendances entre les objets 171
25.5.3 Export de niveau tablespace 171
25.6 Vues du dictionnaire de données de *`DATA Pump`* 172

26 SQL*Loader 174
26.1 Fichier de paramètres 175
26.2 Le fichier de contrôle 177
26.3 Exemples de chargements 178
26.3.1 Exemples de fichiers de contrôle : Longueur variable enregistrements 179
26.3.2 Exemples de fichiers de contrôle : Longueur fixe avec élimination d'enregistrements 180
26.3.3 Chargement dans deux tables 180
26.3.4 Chargement dans deux tables avec utilisation d'une colonne FILLER 181
26.4 Chargement de données LOB 182
26.5 Chargement de formats XML 182
27 Stratégie de Sauvegardes et Restaurations 185
27.1 Les modes NOARCHIVELOG et ARCHIVELOG 186
27.1.1 Le mode NOARCHIVELOG 187
27.1.2 Le mode ARCHIVELOG 187
27.1.3 Mettre la base en mode ARCHIVELOG 188
27.1.4 Les paramètres du processus *ARCH* 188
27.2 Passer la base en mode ARCHIVELOG 189
27.3 Administrer le processus *ARCH* 190
27.3.1 Forcer l'archivage de façon périodique 191
28 Sauvegardes 192
28.1 Sauvegarde base arrêtée 193
28.2 Sauvegarde base en ligne 194
28.2.1 Sauvegarde du fichier de contrôle 194
28.3 Sauvegarde partielle d'un tablespace ONLINE 195
28.4 Sauvegarde de tous les tablespaces de la base ONLINE 196
28.5 Vues du dictionnaire de données 197
28.6 Stratégie recommandée par Oracle 198
28.7 Recover Manager (RMAN) 198
28.8 Le Flash Back 200
29 Restaurations 201
29.1 La commande RECOVER 201
29.1.1 Exemples de restaurations 202

1 Présentation

Oracle Database représente la nouvelle génération de la gestion des informations en entreprise, qui permet de faire face aux exigences qu'imposent la croissance rapide des volumes de données, l'évolution constante de l'environnement et la nécessité de fournir une qualité de service maximale tout en réduisant et en contrôlant les coûts informatiques. Oracle Database offre une performance améliorée du stockage sur fichiers, des fonctionnalités renforcées pour la sécurité, d'importantes améliorations de performances pour Oracle XML DB, et des fonctions nouvelles pour l'OLAP et le datawarehouse.

Oracle Database reste centré sur le grid computing : il permet de constituer des matrices de serveurs et de systèmes de stockage économiques, capables de traiter les données de façon rapide, fiable et évolutive, en supportant les environnements les plus exigeants, qu'il s'agisse de datawarehouse, de transactionnel ou de gestion de contenus.

Oracle Database multiplie les outils de gestion et introduit de nouvelles fonctionnalités d'auto gestion et d'automatisation. *Automatic SQL*, *Partitioning Advisor* ou *Support Workbench* accompagnent les administrateurs pour améliorer les performances et les informer le plus rapidement possible des incidents.

1.1 Les produit Database proposes par Oracle

Les différents produits d'Oracle DATABASE sont proposés en trois gammes

- **Enterprise Edition** - La gamme pour les grosses applications critiques de l'entreprise, intégrant des options supplémentaires telles que le partitionnement des tables.
- **Standard Edition** - La gamme destinée à des serveurs possédant 4 processeurs et ne proposant que l'option RAC/ASM.
- **Standard Edition ONE** - la gamme destinée aux serveurs biprocesseurs, sans option.
- **Personal Edition** - La gamme pour l'utilisateur indépendant (développeur, consultant, …), elle utilise un noyau Enterprise Edition.

Quatre nouvelles options apparaissent dans Oracle Database 11g Enterprise Edition

- · Oracle Real Application Testing
- · Oracle Advanced Compression
- · Oracle Total Recall
- · Oracle Active Data Guard

Oracle Real Application Testing aide ses clients à réduire les délais, les risques et les coûts de test de ses modifications de leur environnement informatique, de façon contrôlée et économique. Outil de tests et de gestion des changements, cet outil est bienvenu là où les infrastructures et environnements sont plus que jamais multiples.

Oracle Advanced Compression intègre de nouveaux mécanismes de compression applicables à tous les types de données permettant d'atteindre des taux de compression de 2x ou 3x, et parfois plus. Associé à de nouveaux mécanismes de partitionnement, Oracle Advanced Compression permet de déployer dans la base de données des stratégies de gestion du cycle de vie des informations, sans avoir à modifier les applications, afin de réduire encore plus les besoins de stockage.

Oracle Total Recall permet de conserver et de retrouver les historiques des données modifiées, mais aussi d'en simplifier l'accès. Les administrateurs peuvent intervenir plus tôt dans les processus, ce qui apporte une nouvelle dimension de temps dans la gestion des données, comme le tracking (*suivi, en temps réel des flux d'informations*), les audits ou le respect des règles.

Oracle active DATA GUARD porte la protection des données jusqu'aux risques de défaillances des systèmes et de désastres. L'application permet simultanément d'écrire et récupérer les données d'une base de données, ce qui augmente les performances et apporte une solution économique de '*Disaster Recovery*'. **Oracle Active Data Guard** peut être employé pour améliorer la performance des bases de données de production en transférant vers une base de données physique secondaire des opérations requérant beaucoup de ressources, telles que certaines requêtes ou les sauvegardes. Cette solution améliore fortement le retour sur investissement pour une base de données physique de secours, car celle-ci peut être utilisée à la fois pour la protection en cas de panne générale et pour l'amélioration de la qualité de service de l'environnement de production.

1.2 Notion de Grid Computing

A partir de la version 10g, la base de données intègre la notion de *Grid Computing* (*réseau distribué d'ordinateurs hétérogènes en grille*).

Le but du Grid est de créer des pools de ressources :

- de stockage
- de serveurs

Le Grid Computing autorise un accès transparent et évolutif (en termes de capacité de traitement et de stockage), à un réseau distribué d'ordinateurs hétérogènes.

Oracle permet à ces machines d'intéropérer ; l'ensemble étant considéré comme une seule ressource unifiée.
- Chaque ressource est vue comme un service !

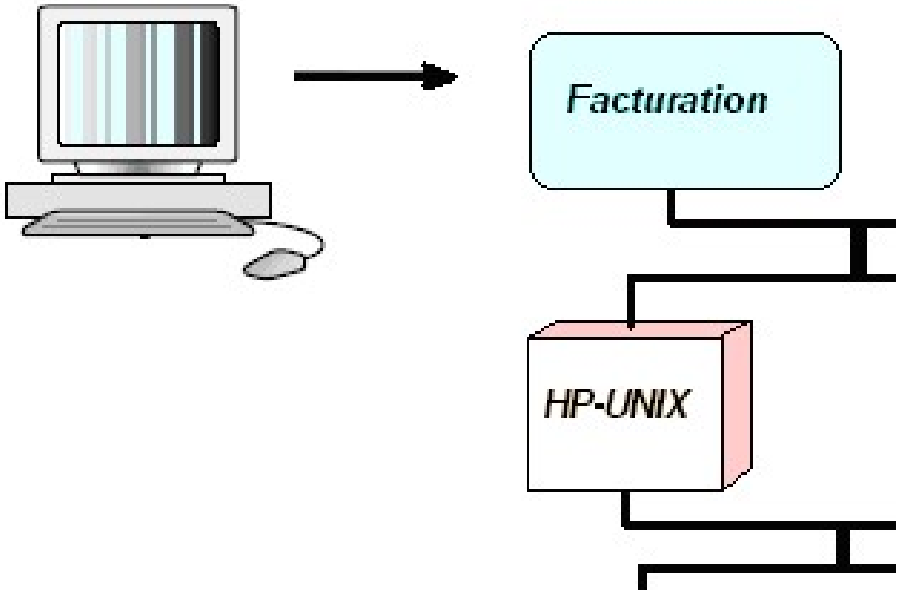
Il est possible de mettre en place des réseaux grille nationaux, voire mondiaux.

Ainsi chaque nouveau système peut être rapidement mis à disposition à partir du pool de composants

Exemple d'application en Grid Computing

Les deux applications présentées ci-dessous, Facturation et Comptabilité se partagent des ressources de deux serveurs.

- Chacune peut être hébergée sur n'importe lequel d'entre eux et les fichiers de base de données peuvent se trouver sur n'importe quel disque.



1.2.1 La gestion ASM (gestionnaire de fichiers : Automatic Storage Management)

La nouvelle fonctionnalité *Automatic Storage Management* (*ASM*) permet à la base de données de gérer directement les disques bruts, elle élimine le besoin pour un gestionnaire de fichiers de gérer à la fois des fichiers de données et des fichiers de journaux.

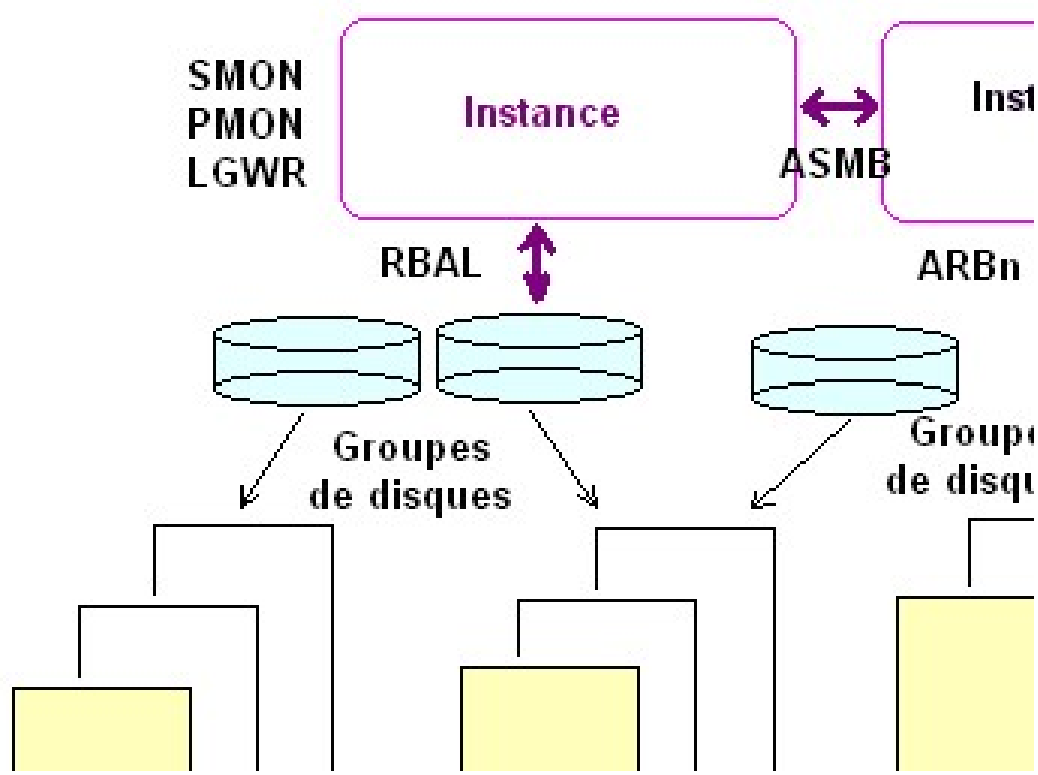
L'ASM répartit automatiquement toutes les données de bases de données entre tous les disques, délivrant le débit le plus élevé sans aucun coût de gestion.

Au fur et à mesure de l'ajout et de l'abandon de disques, l'ASM actualise automatiquement la répartition des données.

Pour utiliser ASM vous devez démarrer une instance appelée « *ASM instance* » qui doit être démarrée avant de démarrer l'instance de votre propre base de données.

Les instances ASM ne montent pas de base de données (ensemble de fichiers constituant la base) mais gère les *metadatas* requises pour rendre les fichiers ASM disponibles à n'importe quelle instance de base de données.

Les deux, instance ASM et instance « *ordinaire* » ont accès au contenu des fichiers. Communicant avec l'instance ASM seulement pour connaître le *layout* des fichiers utilisés.



1.2.2 Les composants développés par Oracle pour le Grid Computing

- **Real Application cluster (RAC)** : Supporte l'exécution d'Oracle sur un cluster d'ordinateurs qui utilisent un logiciel de cluster indépendant de la plate forme assurant la transparence de l'interconnexion.
- **Automatic Storage Management (ASM)** : Regroupe des disques de fabricants différents dans des groupes disponibles pour toute la grille. ASM simplifie l'administration car au lieu de devoir gérer de nombreux fichiers de bases de données, on ne gère que quelques groupes de disques.
- **Oracle Ressource Manager** : Permet de contrôler l'allocation des ressources des nœuds de la grille
- **Oracle Scheduler** : Contrôle la distribution des jobs aux nœuds de la grille qui disposent de ressources non utilisées.
- **Oracle Streams** : Transfère des données entre les nœuds de la grille tout en assurant la synchronisation des copies. Représente la meilleure méthode de réplication.

1.2.3 Outils de développement

Oracle offre l'accès à un choix d'outils et processus de développement, avec de nouvelles fonctionnalités comme ***Client Side Caching***, ***Binary XML***, un nouveau compilateur Java, l'intégration native avec *Microsoft Visual Studio 2005* pour les applications « .NET », ***Oracle Application Express*** pour les outils de migration, ou encore ***SQL Developer*** pour coder rapidement les routines SQL et PL/SQL.

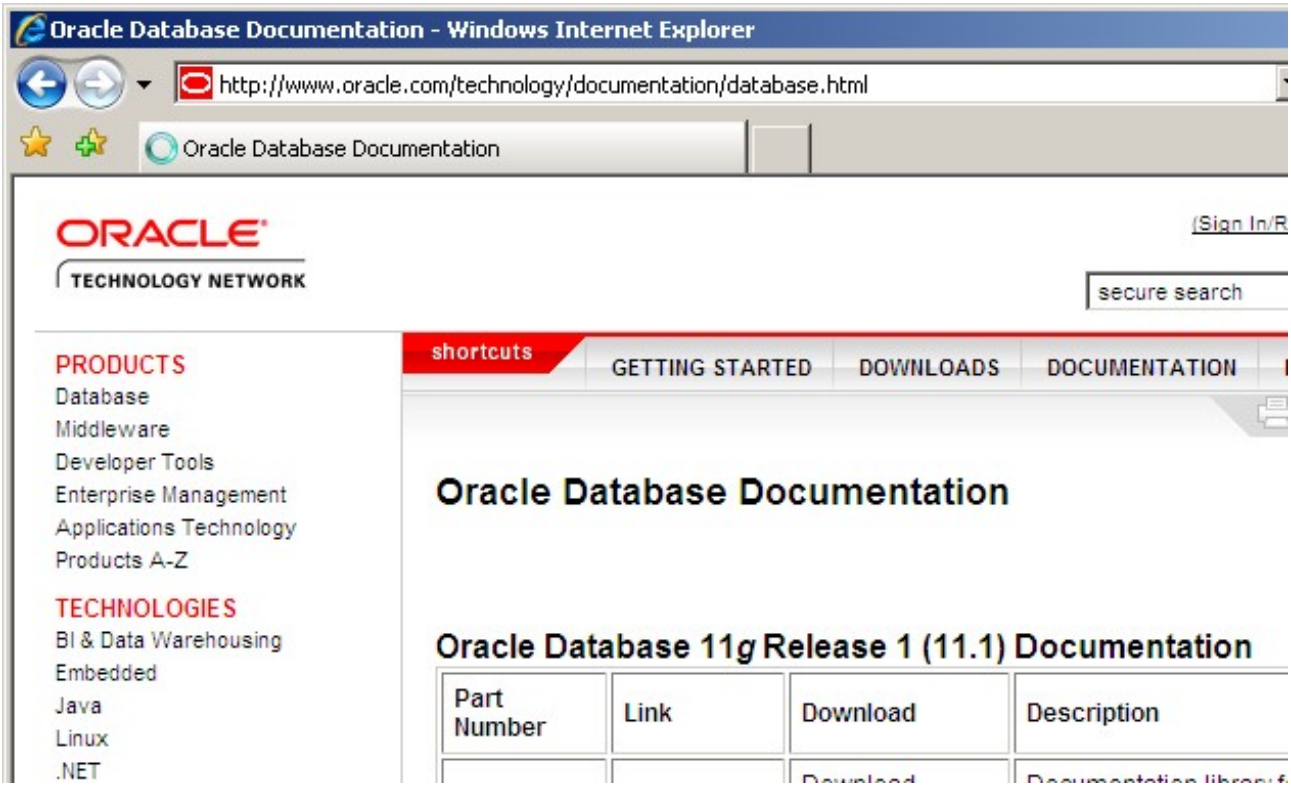
1.3 Règles de nommage dans Oracle Database

Un nom de structure Oracle doit respecter les règles suivantes :

- 30 caractères maximums
- Doit commencer par une lettre
- Peut contenir des lettres, des chiffres et certains caractères spéciaux (_$#)
- N'est pas sensible à la casse
- Ne doit pas être un mot réservé Oracle

2 La documentation

La documentation Oracle est consultable à partir du serveur : http://www.oracle.com

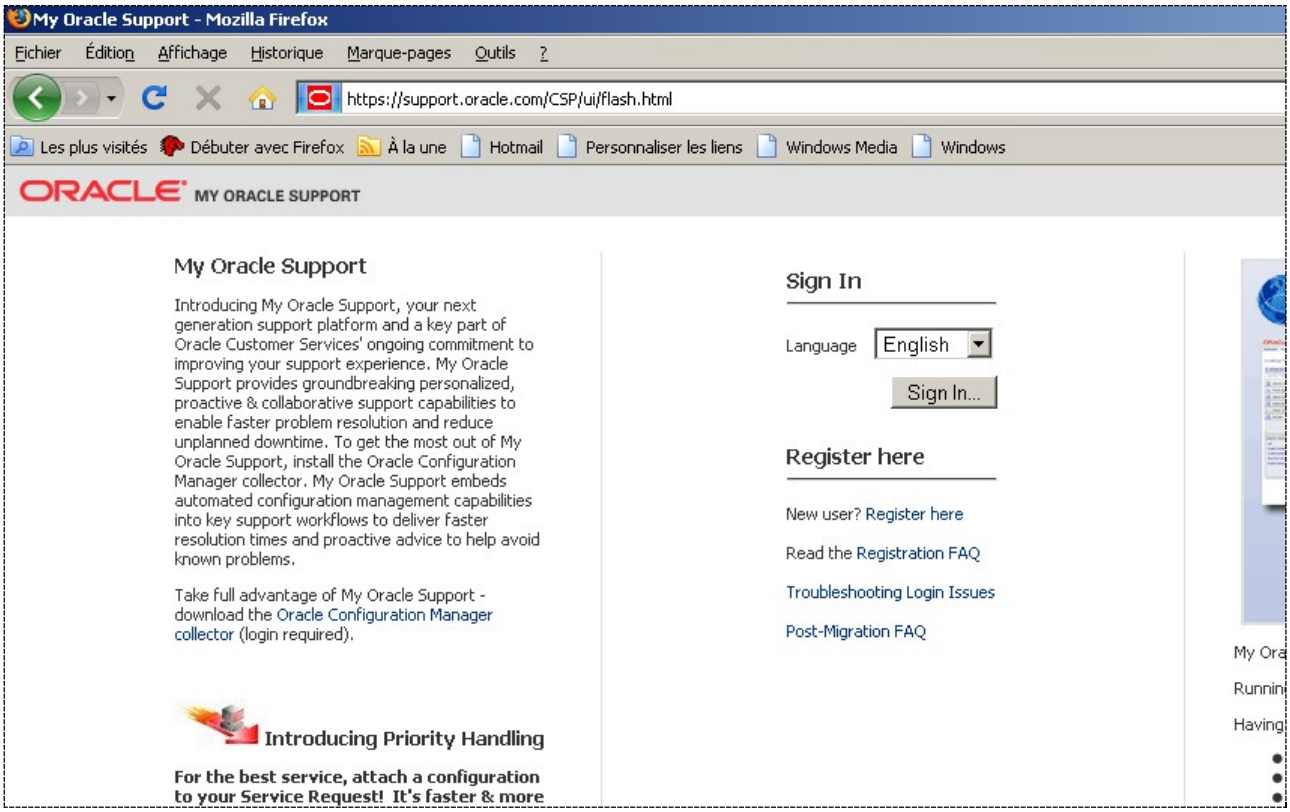


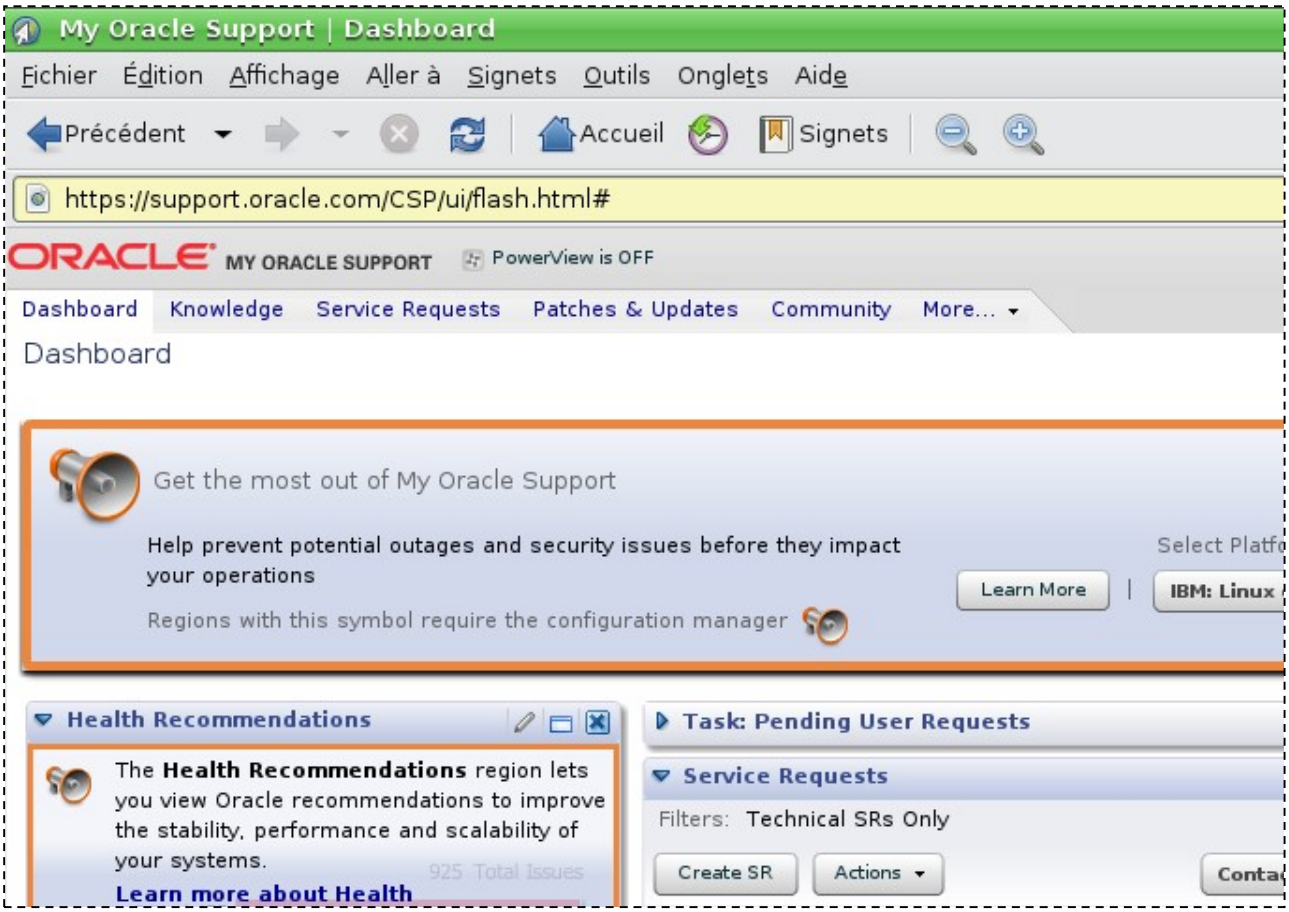
Elle est également consultable à partir du serveur : http://tahiti.oracle.com

2.1 Le support oracle

Le site Metalink est le site de hotline en ligne : : http://metalink.oracle.com remplacé par support.oracle.com

On y trouve des résolutions d'erreurs référencées, des patchs et des scripts d'administration.


My Oracle Support - Mozilla Firefox
Fichier Édition Affichage Historique Marque-pages Outils ?
https://support.oracle.com/CSP/ui/flash.html
Les plus visités Débuter avec Firefox À la une Hotmail Personnaliser les liens Windows Media Windows
ORACLE MY ORACLE SUPPORT
My Oracle Support
Introducing My Oracle Support, your next generation support platform and a key part of Oracle Customer Services' ongoing commitment to improving your support experience. My Oracle Support provides groundbreaking personalized, proactive & collaborative support capabilities to enable faster problem resolution and reduce unplanned downtime. To get the most out of My Oracle Support, install the Oracle Configuration Manager collector. My Oracle Support embeds automated configuration management capabilities into key support workflows to deliver faster resolution times and proactive advice to help avoid known problems.
Take full advantage of My Oracle Support - download the Oracle Configuration Manager collector (login required).
Introducing Priority Handling
For the best service, attach a configuration to your Service Request! It's faster & more
Sign In
Language English
Sign In...
Register here
New user? Register here
Read the Registration FAQ
Troubleshooting Login Issues
Post-Migration FAQ


My Oracle Support | Dashboard
Fichier Édition Affichage Aller à Signets Outils Onglets Aide
Précédent Accueil Signets
https://support.oracle.com/CSP/ui/flash.html#
ORACLE MY ORACLE SUPPORT PowerView is OFF
Dashboard Knowledge Service Requests Patches & Updates Community More...
Dashboard
Get the most out of My Oracle Support
Help prevent potential outages and security issues before they impact your operations
Regions with this symbol require the configuration manager
Learn More
IBM: Linux
Health Recommendations
The Health Recommendations region lets you view Oracle recommendations to improve the stability, performance and scalability of your systems.
925 Total Issues
Learn more about Health
Task: Pending User Requests
Service Requests
Filters: Technical SRs Only
Create SR
Actions

3 Notion de schéma

Le terme **SCHÉMA** désigne l'ensemble des objets qui appartiennent à un utilisateur, ces objets sont préfixés par le nom de l'utilisateur qui les a créés. Il s'agit d'une notion logique désignant la totalité des objets créés par un utilisateur.

C'est ainsi que la base Oracle peut faire la différence entre la table AVION appartenant à l'utilisateur BETTY (BETTY.AVION) et la table avion appartenant à l'utilisateur CHARLY (CHARLY.AVION).

Chacun des utilisateurs propriétaire des objets à tous les droits sur ces objets !

En général on indique sous le terme de schéma, l'ensemble des tables et des index d'une même application.

Les schémas d'exemple fournis par Oracle sont décrits dans la documentation *Oracle Database Sample Schémas*.

Ces schémas peuvent être installés lors de la création de la base de données (appelée par défaut ORCL) au moment de l'installation des binaires d'Oracle.

Principaux types d'objets de schéma :

- Tables et index
- Directory
- Vues, séquences et synonymes
- Programmes PL/SQL (*procédures, fonctions, packages, triggers*)

4 Le dictionnaire de données

C'est un ensemble de tables et de vues qui donne des informations sur le contenu d'une base de données.

Il contient :

- Les structures de stockage
- Les utilisateurs et leurs droits
- Les objets (tables, vues, index, procédures, fonctions, …)
- …

Il appartient à l'utilisateur `SYS` et est stocké dans le tablespace `SYSTEM`.
Sauf exception, toutes les informations sont stockées en `MAJUSCULE`.

Il est créé lors de la création de la base de données, et mis à jour par Oracle lorsque des ordres `DDL` (*Data Définition Langage*) sont exécutés, par exemple `CREATE`, `ALTER`, `DROP` …

Le dictionnaire de données chargé en mémoire est utilisé par Oracle pour traiter les commandes SQL.

4.1 Les tables et vues statiques

Les vues statiques sont basées sur de vraies tables stockées dans le tablespace SYSTEM, et sont accessibles uniquement quand la base est ouverte.

Les **vues statiques** sont caractérisées par leur préfixe :

- `USER_*` : Informations sur les objets qui appartiennent à l'utilisateur
- `ALL_*` : Information sur les objets auxquels l'utilisateur a accès (les siens et ceux sur lesquels il a reçu des droits)
- `DBA_*` : Information sur tous les objets de la base

Derrière le préfixe, le reste du nom de la vue est représentatif de l'information accessible, au pluriel.

4.2 Les tables et vues dynamiques de performance

Ces tables sont basées sur des informations en mémoire ou extraites du fichier de contrôle.

Elles donnent des informations sur le fonctionnement de la base de données, notamment sur les performances. Elles sont remises à zéro si on arrête la base de données.

Elles sont accessibles même lorsque la base n'est pas complètement ouverte (MOUNT)

Les **vues dynamiques** de performance sont :

Préfixées par « `V$` »

Derrière le préfixe, le reste du nom de la vue est représentatif de l'information accessible

```
V$INSTANCE
V$DATABASE
V$SGA
V$DATABASE
V$PARAMETER
```

Les vues `DICTIONARY` et `DICT_COLUMNS` donnent la description de toutes les tables et vues du dictionnaire (statiques et dynamiques).

- la liste complète des vues statiques est obtenue par la requête :

```
SELECT view_name   FROM  ALL_VIEWS
WHERE ALL_VIEWS  like  'DBA*_%'   escape  '*'
;
```

5 Outils d'administration

Trois outils sont présents pour administrer une base de données Oracle

- SQL*Plus (sqlplus), interface d'accès à la base de données en mode commande
- iSQL*Plus, peut être utilisé en application indépendante ou connecté à un référentiel Oracle *Management Server* (OMS)
- *Oracle Enterprise Manager* (OEM), appelé Grid Control ou Database Control.
  - *Database control* est créé à la création d'une base oracle et ne permet d'administrer graphiquement que cette base de données.
  - *Grid control* est un outil qui permet d'administrer une ferme de bases de données (oracle ou non oracle).

5.1 L'outil SQL*Plus

Outil ligne de commande nommé SQLPLUS.

```
SQLPLUS [ connexion ] [ @fichier_script [argument [,…]] ]
```

Il permet de saisir et d'exécuter des ordres SQL ou du code PL/SQL et dispose en plus d'un certain nombre de commandes.

```
•      sans connexion
C:\> SQLPLUS /NOLOG
•      avec connexion
C:\> SQLPLUS system/tahiti@tahiti
•      avec connexion et lancement d'un script sur la ligne de commande
C:\> SQLPLUS system/tahiti@tahiti @info.sql

•      sous dos ---------------
set ORACLE_SID=TAHITI
•      connection sans fichier de mots de passe
SQL> connect /as sysdba
ConnectÚ.

SQL> show user
USER est « SYS »


•      sous unix ---------------
Export ORACLE_SID=TAHITI
•      Connexion avec un fichier de mots de passe
SQL> connect sys/secret as sysdba
ConnectÚ.

SQL> show user
USER est "SYS"
SQL>
```

5.1.1 Environnement de travail

SQL*PLUS est avant tout un « interpréteur » de commandes SQL. Il est également fortement interfacé avec le système d'exploitation. Par exemple, sous UNIX, on pourra lancer des commandes UNIX sans quitter sa session SQL*PLUS.

Un SGBDR est une application qui fonctionne sur un système d'exploitation donné. Par conséquent, il faut se connecter au système avant d'ouvrir une session ORACLE. Cette connexion peut être implicite ou explicite.

Pour lancer SQL Plus sans se connecter à une base de données utilisez la commande :

```
C:\> sqlplus  /nolog
```

5.1.2 Quelques commandes SQL*Plus

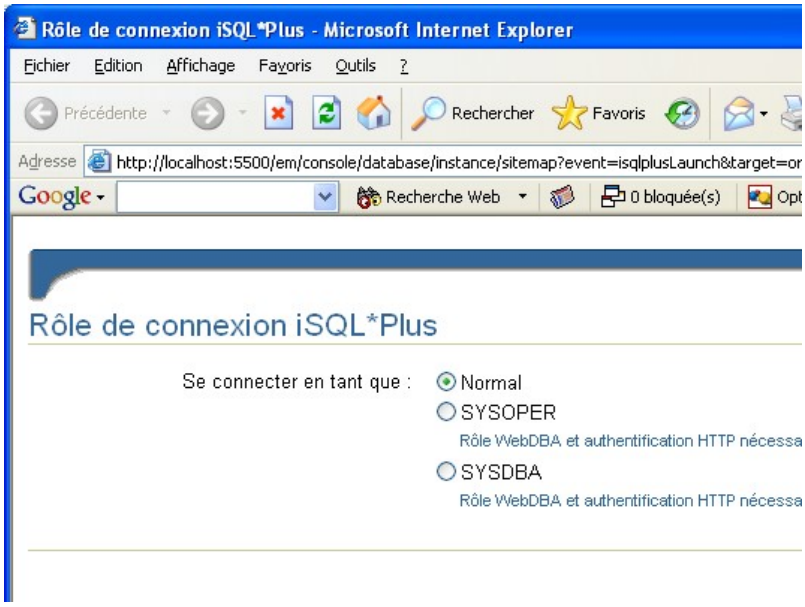
SQL*Plus est un outil composé de commandes de mise en forme et d'affichage :

- `COL ADRESSE FORMAT A20`, formater l'affichage d'une colonne `ADRESSE` sur 20 caractères
- `COL PRIXUNIT FORMAT 99.99`, formater l'affichage d'une colonne `PRIXUNIT`
- `CLEAR COL`, ré-initialiser la taille des colonnes par défaut
- `SET LINESIZE 100`, reformater la taille de la ligne à 100 caractères
- `SET PAUSE ON`, afficher un résultat page par page
- `SHOW USER`, visualiser le user sous lequel on est connecté
- `CONNECT` , se connecter à l'instance
- `User/MotPass@adresseServeur` , permet de changer de session utilisateur
- `CLEAR SCREEN`, ré-initialiser l'écran
- `SET SQLPROMPT TEST>` , afficher le prompt SQL en : `TEST>`
- `DESC Nom_Table`, afficher la structure d'une table ou d'une vue
- `SPOOL nomfichier.txt`, permet d'activer un fichier de format texte dans lequel on retrouvera les commandes et résultats affichés dans SQL Plus
- `SPOOL OFF`, permet de désactiver le spool ouvert précédemment
- `@ nom_ficher`, permet d'exécuter le contenu d'un fichier sql
- `/`, ré-active la dernière commande
- `SET ECHO ON/OFF`, affiche ou non le texte de la requête ou de la commande à exécuter
- `SAVE nom_ficher [append|create|replace]`, permet de sauvegarder le contenu du buffer courant dans un fichier « .sql »
- `TIMING ON|OFF`, provoque l'affichage d'informations sur le temps écoulé, le nombre d'E/S après chaque requête
- `TI ON|OFF`, provoque l'affichage de l'heure avec l'invite de commande
- `TERM [ON|OFF]`, supprime tout l'affichage sur le terminal lors de l'exécution d'un fichier

- `VER [ON|OFF]`, provoque l'affichage des lignes de commandes avant et après chaque substitution de paramètre
- `SQL }`, spécifie le caractère « } » comme étant le caractère de continuation d'une commande SQL*Plus
- `SUFFIX txt`, spécifie l'extension par défaut des fichiers de commande SQL*Plus

5.2 L'outil *i*SQL*Plus

Outil Internet d'accès à une base de données Oracle, permettant d'écrire des requêtes SQL (d'une façon plus ou moins graphique).



Par défaut, seule la connexion en tant qu'utilisateur « normal » (non `SYSDBA` ou `SYSOPER`) est autorisée.

Par contre, la connexion en tant qu'utilisateur `SYSDBA` ou `SYSOPER` est protégée par une authentification au niveau du serveur `HTTP`

Pour l'autoriser, il faut au choix :

- Ajouter des entrées (utilisateur / mot de passe) à l'aide de l'utilitaire *htpasswd* dans un fichier d'authentification du serveur `HTTP` (défini par défaut dans le fichier de configuration `isqlplus.conf` à : `ORACLE_HOME\sqlplus\admin\iplusdba.pw`
- Désactiver l'authentification du serveur `HTTP` pour ce type de connexion (directive <`Location /isqlplusdba`> dans le fichier de configuration `isqlplus.conf`)

Lors d'une connexion `SYSDBA` ou `SYSOPER`, l'URL est modifiée en :

⇨ `http://serveur[:port]/isqlplusdba`

5.3 Le Database Control et le Grid control

À partir de la version 10g la base de données Oracle s'est dirigée vers le WEB pour fournir une nouvelle version d'Entreprise Manager à la place de celui de la 9i basé sur java possédant une apparence Windows; ainsi que des variantes selon l'utilisation Dbcontrol pour une seule Base de données ou grid control pour centraliser la gestion de plusieurs bases cibles.
Le *Grid Control* est la console graphique qui permet d'administrer un ensemble de bases de données sur des serveurs distants, appelé « ferme de bases de données ».

Le *Database Contrôle* est en réalité un sous ensemble du Grid Control, correspondant à l'administration de la base de données choisie.

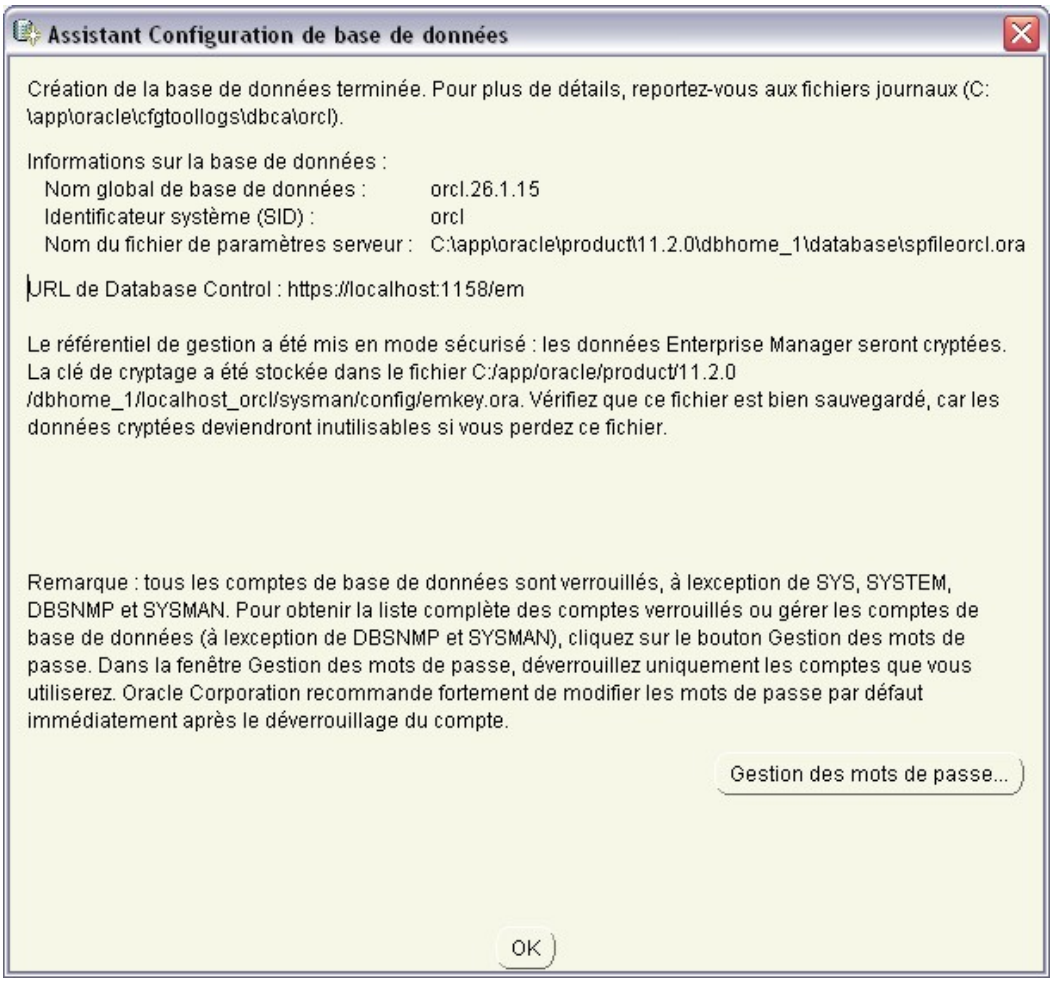
Contrairement au *Grid Control*, le *Database Control* est inclus dans l'installation standard.

Il contient un référentiel et est créé après la création de la base de données.

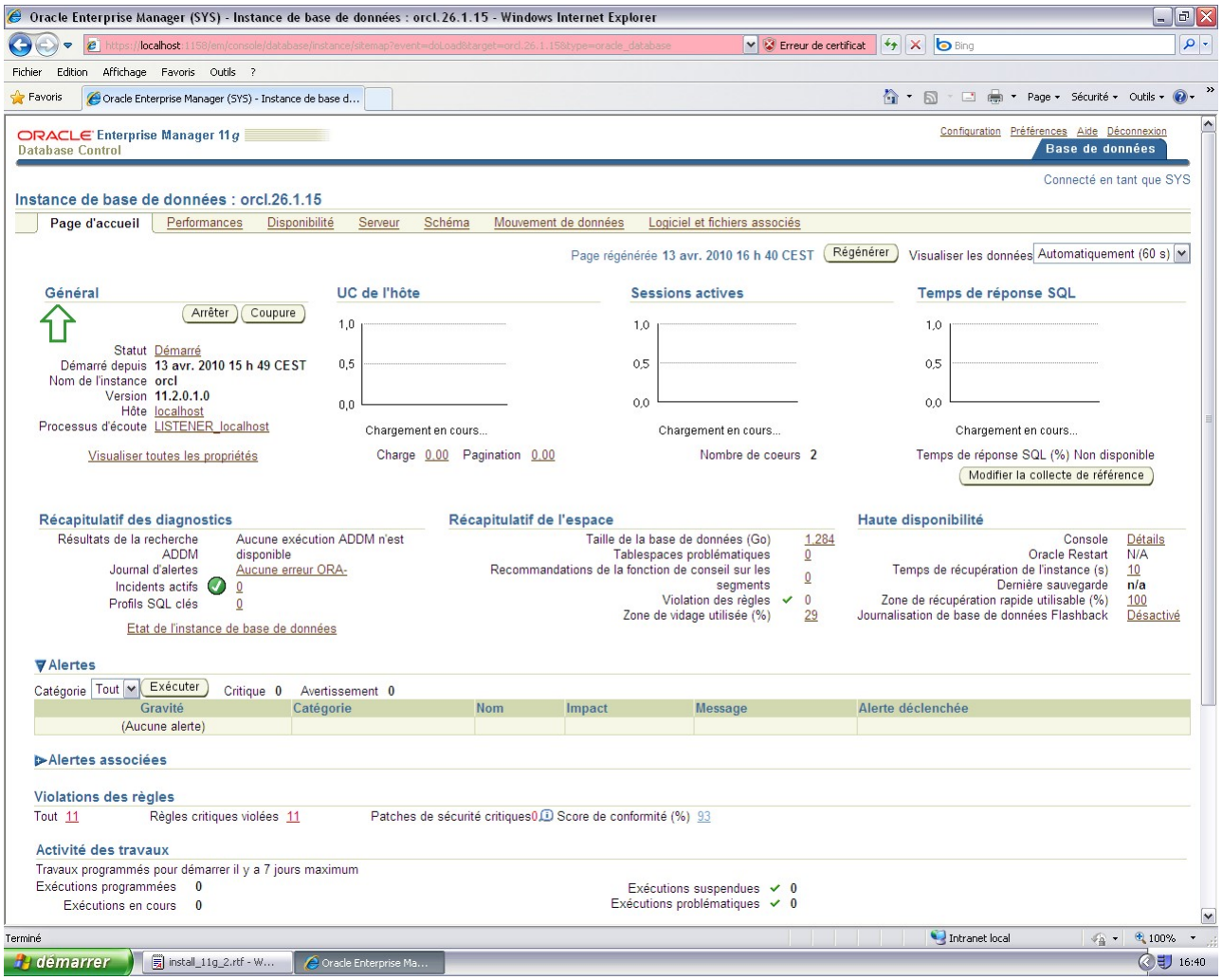
Cette console permet d'administrer directement la base de données :

- Arrêt/Démarrage, gestion du stockage, gestion des utilisateurs, gestion des schémas, …
- Remontée d'alerte, de planification de tâche, de sauvegarde/restauration, d'export/import, …

Après création d'une base de données Oracle, le *Database Control* peut être affiché sur demande dans le navigateur.



Présentation du database control



En cas de besoin, l'utilitaire *Entreprise Manager Configuration Assistant* (EMCA) peut être utilisé pour créer l'environnement du Database Control pour cette nouvelle base.

```
EMCA [ -r | -x <SID> ]
```

- Sans option l'utilitaire crée l'environnement complet du Database Control.
- –r le référentiel n'est pas créé
- -x <SID> supprime l'environnement du Database control

Si vous utilisez successivement [-x] puis [-r] vous pouvez recréer l'environnement tout en conservant le référentiel existant.

```
D:\cours_Admin10G>emca -x TAHITI
EMCA dÚmarrÚ Ó Sat Mar 19 12:57:58 CET 2005
La configuration d'Enterprise Manager a rÚussi.
EMCA arrÛtÚ Ó Sat Mar 19 12:57:58 CET 2005
```

Configuration du database contrôle en fin de création de la base de données

```
connect "SYS"/"&&sysPassword" as SYSDBA
startup ;
host C:\app\oracle\product\11.2.0\dbhome_1\bin\emca.bat -config dbcontrol db -silent -
DB_UNIQUE_NAME tahiti -PORT 1521 -EM_HOME C:\app\oracle\product\11.2.0\dbhome_1 -LISTENER
LISTENER -SERVICE_NAME tahiti -SID tahiti -ORACLE_HOME
C:\app\oracle\product\11.2.0\dbhome_1 -HOST localhost -LISTENER_OH
C:\app\oracle\product\11.2.0\dbhome_1 -LOG_FILE
C:\app\oracle\admin\tahiti\scripts\emConfig.log;
spool off
```

6 L'architecture OFA

OFA, *Oracle Flexible Architecture*, est un ensemble de recommandations sur l'arborescence et le nommage des fichiers du serveur contenant les produits et les bases de données en tenant compte de la possibilité d'avoir plusieurs bases de données et plusieurs versions d'Oracle par plate-forme

Un des avantages est de séparer les produits Oracle des fichiers des bases de données.

La norme de la version 11g est présentée page suivante.

Le répertoire `/app/oracle/oradata/orcl/` contient les fichiers de la base de données « orcl »

Le répertoire `/app/oracle/admin/orcl/` contient les répertoires destinés aux exports Data Pump ou non de la base de données ainsi qu'au fichier de paramètre utilisé lors de la création de la base de données « orcl ».

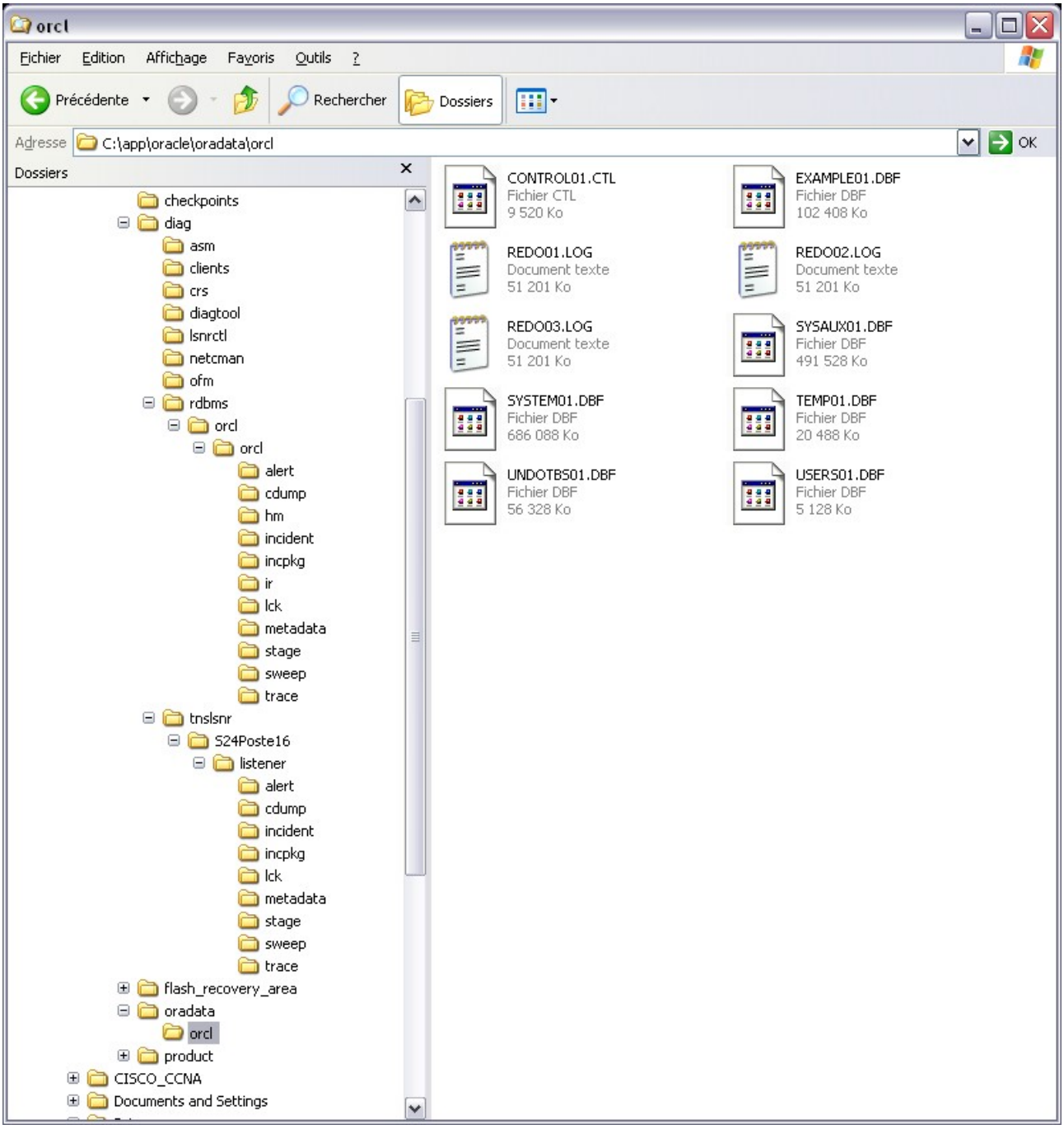
- `/app/oracle/admin/orcl/`
  - `Adump`
  - `Dpdump`
  - `pfile`

le répertoire `/app/oracle/diag/` contient les répertoires

- `/app/oracle/diag/rdbms/orcl/orcl/`
  - `Alert` dans lequel est stocké le fichier des alertes en format xml
  - `cdump`
  - `hm`
  - `incident`
  - `incpkg`
  - `ir`
  - `lck` contient un ensemble de fichiers vide représentant des locks
  - `metadata` contient un ensemble de fichiers binaires « `.ams` »
  - `stage`
  - `sweep`
  - `trace` contient un ensemble de fichiers de traces de l'instance
- `/app/oracle/flash_recovery_area/orcl/` contient les fichiers de controle multiplexes, et un repertoire ONLINELOG destine aux fichiers de flashback.

Le répertoire `/app/oracle/product/11.2.0/dbhome_1` contient les répertoires des binaires d'oracle. On y retrouve les répertoires

- BIN qui contient les binaires d'oracle et certains outils comme « sqlplus.exe ».
- Database qui contient sous Windows les fichiers de mot de passe et SPFILE ainisi qu'un sous répertoire d'archive de Redo Log lorsque l'archivage est activé
- Dbs qui contient sous unix, les fichiers de mot de passe et SPFILE ainsi qu'un sous répertoire d'archive de Redo Log lorsque l'archivage est activé
- NETWORK qui contient le listener et tnsnames.ora



7 Installation Oracle

L'installeur `OUI` (*Oracle Universal Installeur*) est un outil d'installation Oracle compatible `OFA (Oracle flexible Architecture)`.

7.1 Pré-requis matériel

Une installation standard peut être effectuée sur une machine avec 1 Go de RAM et 1 Go de swap (mémoire virtuelle) en supplément vous pouvez utiliser le produit avec des composants supérieurs.

Selon votre activité, quand vous installez Oracle, l'installation standard peut être effectuée en moins de 20 minutes.

Sous Unix, bien suivre les pré-requis demandés pour chaque version Unix.

Il faut toujours se référer à la documentation Oracle spécifique à la plate-forme.

- ⇨ Installation Guide & Release Notes

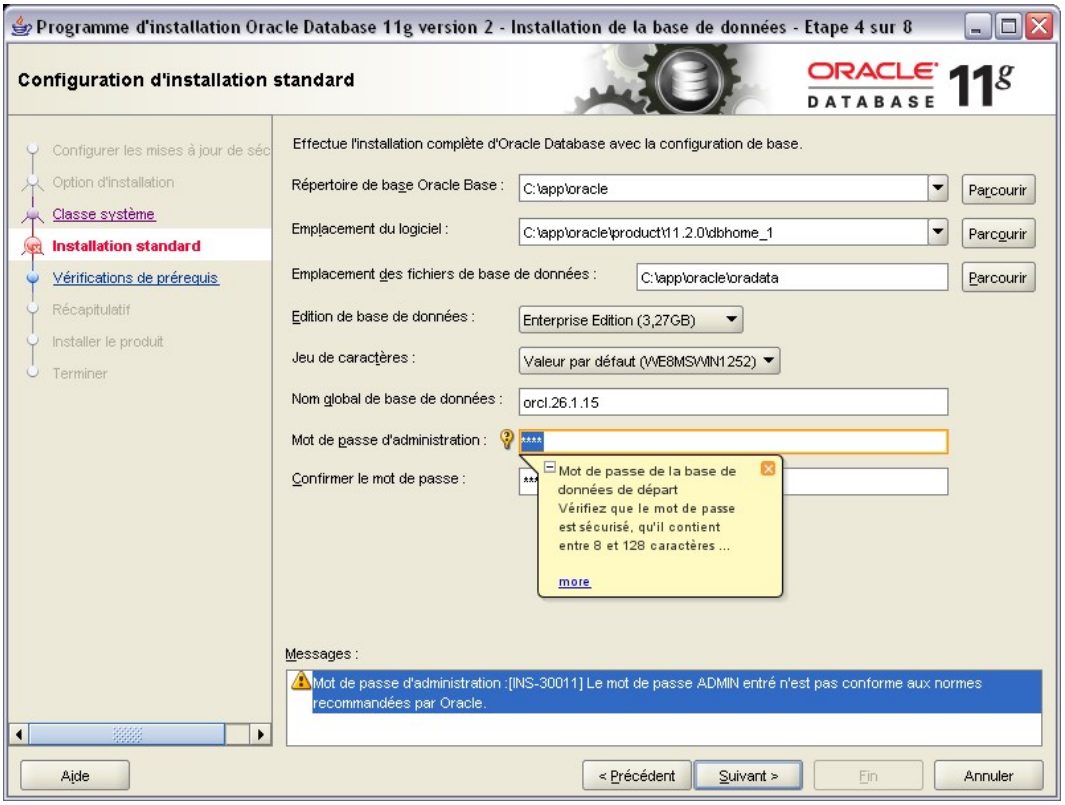
Un écran de synthèse est affiché, permettant de vérifier l'installation.

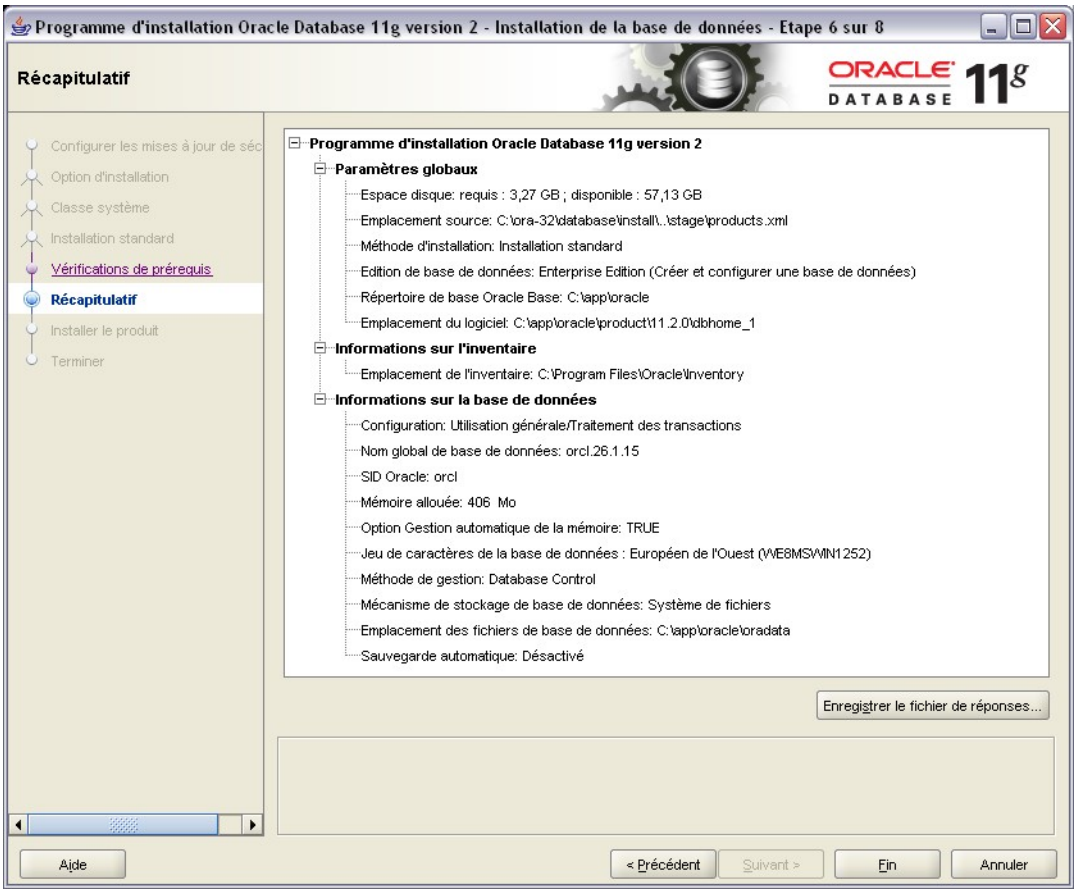
L'installation de Oracle Database automatise la plupart des vérifications de pré-requis pour l'installation.

Si vous choisissez de créer une base de données pendant l'installation d'Oracle, vous devrez répondre à quelques questions permettant de configurer votre base de données.

- Nom de la base de données, par défaut = ORCL
- Jeu de caractères à définir
- …

A partir de la version 11g, la casse utilisée pour les mots de passe est sensitive !


Programme d'installation Oracle Database 11g version 2 - Installation de la base de données - Etape 4 sur 8
Configuration d'installation standard
ORACLE DATABASE 11g
Configurer les mises à jour de séc
Option d'installation
Classe système
Installation standard
Vérifications de prérequis
Récapitulatif
Installer le produit
Terminer
Effectue l'installation complète d'Oracle Database avec la configuration de base.
Répertoire de base Oracle Base : C:\app\oracle
Parcourir
Emplacement du logiciel : C:\app\oracle\product\11.2.0\dbhome_1
Parcourir
Emplacement des fichiers de base de données : C:\app\oracle\oradata
Parcourir
Edition de base de données : Enterprise Edition (3,27GB)
Jeu de caractères : Valeur par défaut (WE8MSWIN1252)
Nom global de base de données : orcl.26.1.15
Mot de passe d'administration :
Confirmer le mot de passe :
Mot de passe de la base de données de départ
Vérifiez que le mot de passe est sécurisé, qu'il contient entre 8 et 128 caractères ...
more
Messages :
Mot de passe d'administration :[INS-30011] Le mot de passe ADMIN entré n'est pas conforme aux normes recommandées par Oracle.
Aide
< Précédent
Suivant >
Fin
Annuler


Programme d'installation Oracle Database 11g version 2 - Installation de la base de données - Etape 6 sur 8
Récapitulatif
ORACLE DATABASE 11g
Configurer les mises à jour de séc
Option d'installation
Classe système
Installation standard
Vérifications de prérequis
Récapitulatif
Installer le produit
Terminer
Programme d'installation Oracle Database 11g version 2
Paramètres globaux
Espace disque: requis : 3,27 GB ; disponible : 57,13 GB
Emplacement source: C:\ora-32\database\install\..\stage\products.xml
Méthode d'installation: Installation standard
Edition de base de données: Enterprise Edition (Créer et configurer une base de données)
Répertoire de base Oracle Base: C:\app\oracle
Emplacement du logiciel: C:\app\oracle\product\11.2.0\dbhome_1
Informations sur l'inventaire
Emplacement de l'inventaire: C:\Program Files\Oracle\Inventory
Informations sur la base de données
Configuration: Utilisation générale/Traitement des transactions
Nom global de base de données: orcl.26.1.15
SID Oracle: orcl
Mémoire allouée: 406 Mo
Option Gestion automatique de la mémoire: TRUE
Jeu de caractères de la base de données : Européen de l'Ouest (WE8MSWIN1252)
Méthode de gestion: Database Control
Mécanisme de stockage de base de données: Système de fichiers
Emplacement des fichiers de base de données: C:\app\oracle\oradata
Sauvegarde automatique: Désactivé
Enregistrer le fichier de réponses...
Aide
< Précédent
Suivant >
Fin
Annuler

7.2 Installation du client

Cette installation permet d'installer, au minimum, les fichiers nécessaires pour accéder à une base Oracle du réseau (*Couche Oracle Net*.

L'installation d'un client Oracle peut intégrer également :

- Des outils d'interrogation ou d'administration
- Des produits pour le développement

L'installation s'effectue avec OUI (*Oracle Universal Installer*) selon les principales étapes suivantes :

- Désignation de l'emplacement de l'installation (*Oracle Home*)
- Type d'installation
    - Administrateur, installation de tous les composants
    - Runtime, qui ne contient que Oracle Net, SQL*Plus et les drivers JDBC),
    - Instant Client, ou « client instantané, installation minimale qui ne propose que les « bibliothèques » nécessaires aux applications OCI (*Oracle Call Interface*)
    - Personnalisé permet de choisir les composants à installer.
- Affichage d'un écran de synthèse permettant de confirmer l'installation

Rappel

L'OCI (Oracle Call Interface) est une application de programmation d'interface (API) qui permet à un développeur d'applications d'utiliser une procédure naturelle, d'un langage de troisième génération ou d'appels de fonctions, pour avoir accès au serveur de base de données d'Oracle pour contrôler toutes les phases de l'exécution de l'expression de SQL. OCI fournit une bibliothèque standard de bases de données et des fonctions de recherche sous la forme de bibliothèques dynamiques en phase d'exécution, ORA, DLL qui peuvent être liées par l'application.

8 Architecture Oracle

L'architecture oracle est constituée d'une instance et d'une base de données appelée database.

Une instance est constituée :

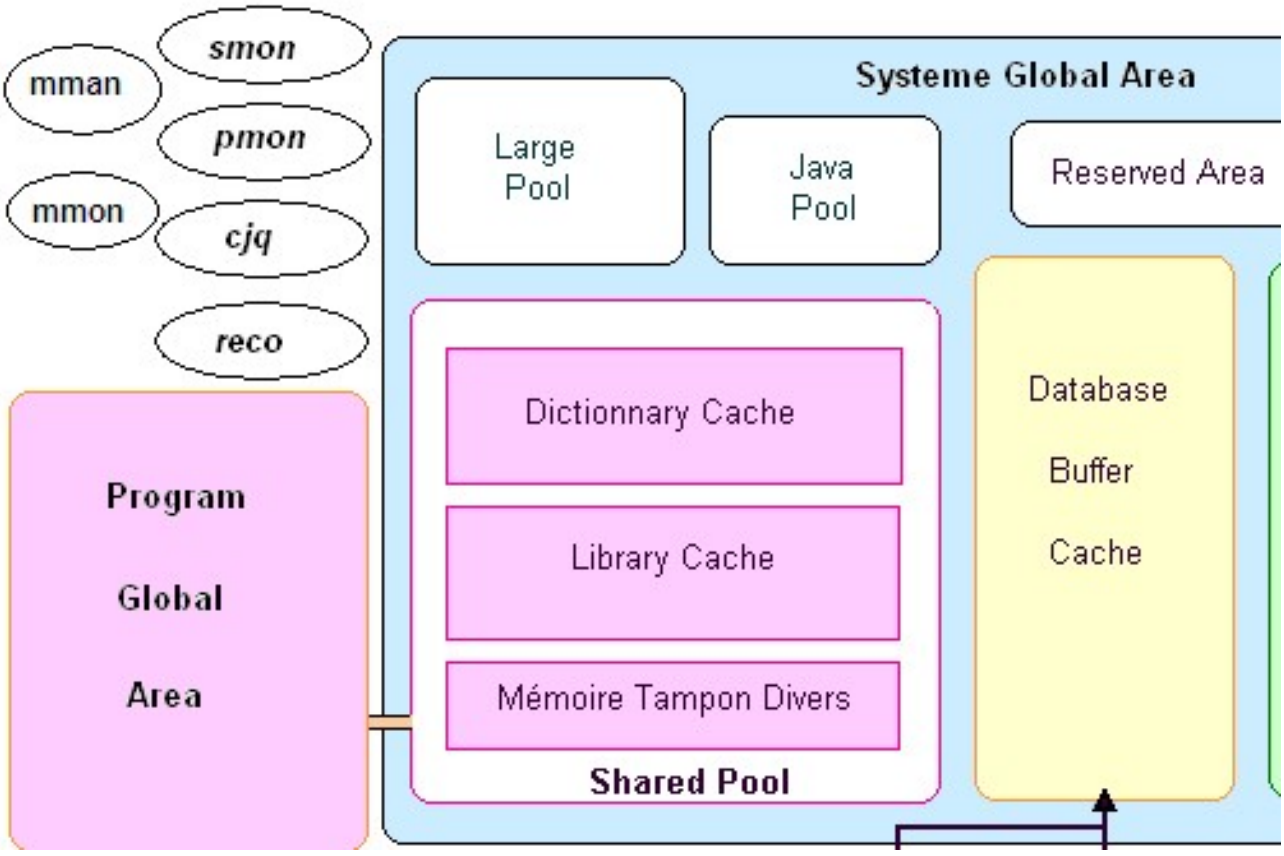
- D'une zone de mémoire partagée appelée System Global Area (SGA)
- D'un ensemble de processus d'arrière plan ayant chacun un rôle bien précis
- D'un ensemble de processus serveur chargés de traiter les requêtes des utilisateurs

La base de données est l'ensemble des fichiers qui permettent de gérer les données de la base.

Une base de données est constituée de :

- Un fichier de contrôle, contenant les informations sur tous les autres fichiers de la base (nom, emplacement, taille).
- Fichiers de Redo Log, contenant l'activité des sessions connectées à la base. Ce sont des journaux de transactions de la base. Ils sont organisés en groupe possédant le même nombre de membres.

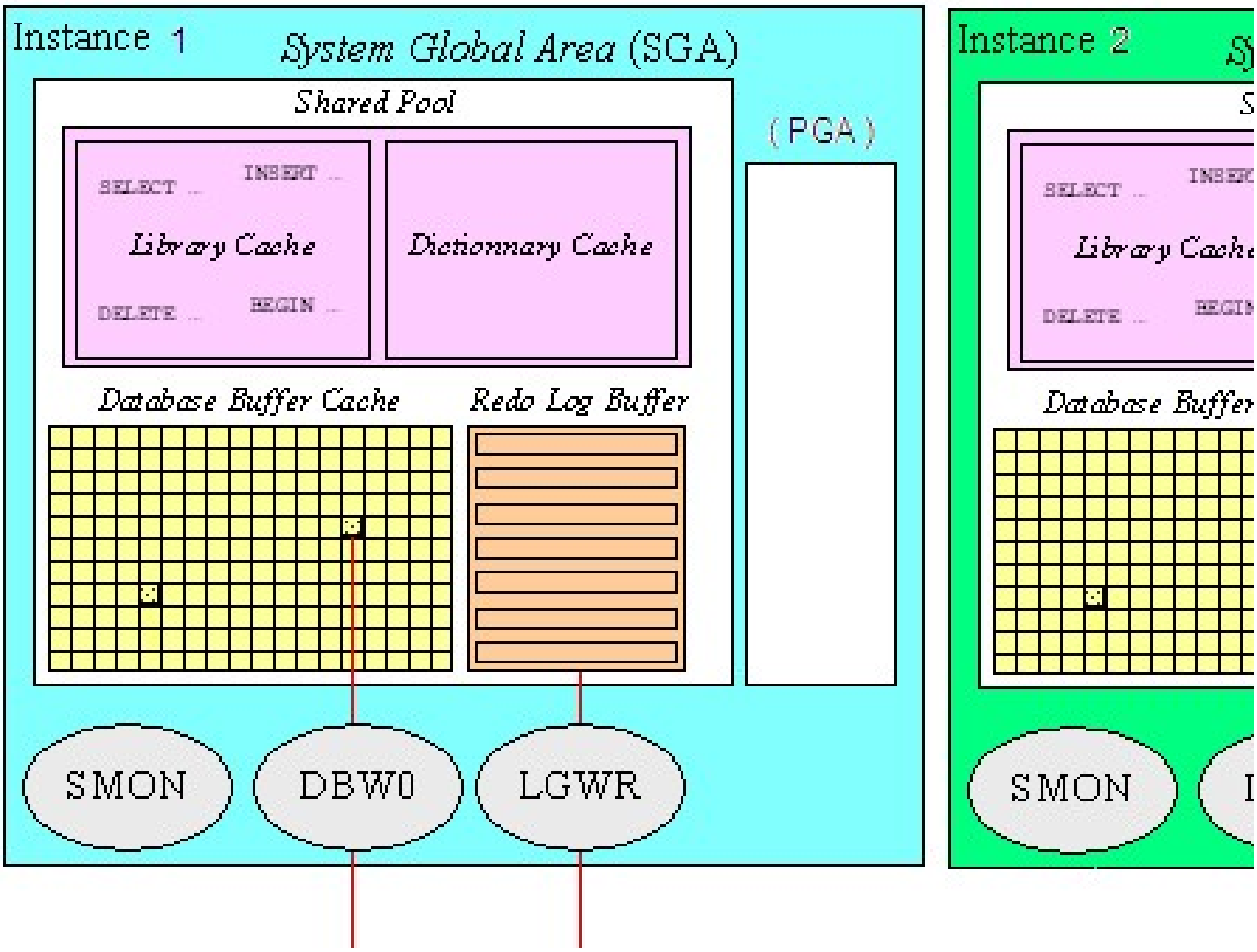
  Et éventuellement, de fichiers de Redo Log archivés contenant les archives d'anciens fichiers de Redo Log.
- D'un ou plusieurs fichiers de données qui contiennent les données des tables de la base.



Une instance est l'ensemble des processus d'arrière-plan (*background process*) et de zones mémoire qui sont allouées au démarrage de la base de données, pour permettre l'exploitation des données.

Une instance ne peut ouvrir qu'une seule base de données à la fois et dans la grande majorité des cas, une base de données est ouverte par une seule instance.

Néanmoins, moyennant la mise en œuvre de l'option RAC (Oracle Real Application Clusters), permettant d'utiliser Oracle sur des serveurs en cluster, une base de données peut être ouverte par plusieurs instances situées sur des nœuds distincts d'un cluster de serveurs ; cette option est intéressante pour la haute disponibilité mais elle est relativement complexe à mettre en œuvre.



En dehors des processus de l'instance, il existe des processus utilisateurs correspondant à l'application utilisée par l'utilisateur pour se connecter à la base de données (SQL*Plus, un progiciel, un logiciel spécifique, …).

Dans une architecture client/serveur, ces processus utilisateurs sont situés sur le poste de l'utilisateur et communiquent avec le serveur à travers le réseau grâce à la couche *Oracle Net*.

8.1 Connexion utilisateur

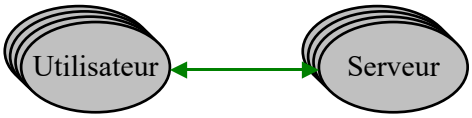
Lorsqu'un utilisateur se connecte à la base de données, il ouvre une session.

Les processus utilisateur sont alors pris en charge par les Processus serveur qui sont chargés de traiter les requêtes des utilisateurs, notamment de charger dans le Database Buffer Cache les données nécessaires.

Le processus serveur communique (localement ou à travers le réseau) avec un processus utilisateur correspondant à l'application de l'utilisateur.

Dans la configuration par défaut, Oracle lance un processus serveur dédié à chaque utilisateur (*dedicated server configuration*)

Mais Oracle peut être configuré en *multithreaded server (*MTS) de manière à avoir des processus serveur partagés par plusieurs processus utilisateur.



L'instance

L'instance est dimensionnée par un ensemble de paramètres stockés dans le fichier de paramètres système SPFILE<SID>.ora, celui-ci a été créé à la création de la base de données à partir d'un fichier de paramètres caractère : PFILE<SID>.ora.

- <SID> correspond au nom de l'instance

8.1.1 La PGA (Program Global Area)

Mémoire privée des différents processus distribuée au moment de la connexion d'un client.

Pour un processus serveur, la PGA contient :

- Une zone de tri (allouée dynamiquement lors d'un tri)
- Des informations sur la session
- Des informations sur le traitement des requêtes de la session
- Les variables de session

Dans une configuration multithreaded, une partie de la PGA est en fait stockée dans la SGA (dans la *Large Pool* ou éventuellement dans la *Shared Pool*).

A partir de la version 9i, la PGA devient dynamique et est configurée par le paramètre PGA_AGGREGATE_TARGET.

8.1.2 La SGA: System Global Area

Cette zone de mémoire partagée par les différents processus de l'instance est allouée au démarrage de l'instance et est libérée lors de l'arrêt de celle-ci.

Les principaux composants de la SGA sont :

- **SPA : Shared Pool Area** : zone de partage des requêtes et du dictionnaire Oracle.
  La *Shared Pool Area* est la partie de la SGA qui est utilisée par Oracle pour partager les requêtes *(Library Cache)* et le dictionnaire de données (*Dictionary Cache*) entre les différents processus.
  La Shared Pool est globalement dimensionnée par le paramètre `SHARED_POOL_SIZE` ; la répartition entre le Library Cache et le Dictionary Cache est assurée par Oracle.
  Dimensionnée par le paramètre `SHARED_POOL_SIZE` .
- **Database Buffer Cache** : Le Database Buffer Cache est un cache de données qui joue le même rôle que la Shared Pool mais pour les données de la base.
  Les données de la base ne sont accessibles, en lecture ou en mise à jour, qu'après avoir été chargées dans le Database Buffer Cache.
  Dans la pratique, le Database Buffer Cache ayant une taille finie, Oracle utilise un algorithme LRU (*Least Recently Used*) pour gérer le cache : en cas de manque de place, Oracle supprime du cache les données utilisées le moins récemment.
  Généralement, augmenter la taille du Database Buffer Cache améliore les performances.
  La taille du bloc (`DB_BLOCK_SIZE`) étant fixée à la création de la base, la taille du Database Buffer Cache est définie par la valeur du paramètre `DB_BLOCK_BUFFERS` qui fixe le nombre de buffers en mémoire, chaque buffer ayant une taille égale à `DB_BLOCK_SIZE`.
  Le paramètre `DB_BLOCK_BUFFERS` est typiquement compris entre un millier (pour une petite base de test) et plusieurs dizaines/centaines de milliers d'octets.
  Dimensionné par le paramètre `DB_CACHE_SIZE`.

- **Redo Log Buffer** : Le Redo Log Buffer stocke les informations sur les modifications apportées à la base, avant leur écriture dans un fichier de Redo Log.
  L'écriture dans le Redo Log Buffer est séquentielle (les modifications de plusieurs transactions se mélangent) et circulaire (quand le buffer est plein, il repart au début … après avoir été écrit sur disque dans les fichiers de Redo Log).
  Dimensionné par le paramètre `LOG_BUFFER`.
- **Large Pool** (à partir de la Version 8), Ajouté en version 8 à l'extérieur du pool partagé pour procurer de l'espace spécifique aux opérations parallèles, à l'usage de la configuration MTS et du module RMAN. En version 10g, la mise en œuvre de l'ASM utilise le Large Pool. Oracle conseille de le dimensionner à 100 Mo dans ce cas.
  Dimensionnée par le paramètre `LARGE_POOL_SIZE`.
- **Java Pool** (à partir de la Version 8), zone réservée aux programmes Java.
  Dimensionné par le paramètre `JAVA_POOL_SIZE`.
  L'installation des composants Java impose que cette zone soit configurée, les instructions Java s'y exécutent.

- **Streams Pool** (à partir de la Version 10), zone réservée notamment lors de la réplication de données entre bases de données distantes.
  Dimensionné par le paramètre `STREAMS_POOL_SIZE`.
- **Reserved Area** (à partir de la version 7.3), zone réservée destinée à l'enregistrement d'objets SQL de grande taille (y compris des packages, des procédures et des fonctions).
  Dimensionnée par le paramètre `SHARED_POOL_RESERVED_SIZE`.
- ***Nouveauté 11g* : result_cache**
  Ce cache est un nouveau composant de la SGA et est utilisé par Oracle pour initialiser le paramètre MEMORY_TARGET.
  Par défaut ce paramètre est positionné à une valeur égale à 128K.

Ces différentes zones mémoires sont configurées à l'aide du paramètre contenu dans le fichier de paramètres `SPFILE`.

En dehors de la `SGA`, chaque processus possède une zone de mémoire privée appelée `PGA` (*Program Global Area*).

La version 11g, offre la possibilité d'automatiser la gestion de l'instance grâce aux paramètres MEMORY_TARGET et MEMORY_MAX_SIZE.

La vue dynamique V$MEMORY_TARGET_ADVICE

Cette vue dynamique de performances, permet de suivre l'allocation dynamique et visualiser les différentes valeurs de l'allocation dynamique de la mémoire.

Cette vue contient les colonnes :

- Memory size : taille réelle de la mémoire totale allouée à l'instance
- Size_factor : coefficient de taille
- Estd_db_time : taille de l'instance utilisée en mémoire en moyenne par rapport aux facteurs size-factor et time_factor.
- Time_factor : coefficient de temps
- Version :

```
Select * from v$memory_target_advice ;
```

La vue V$MEMORY_DYNAMIC_COMPONENTS, permet de visualiser les différentes valeurs de chaque pool, entre autre la shared_pool, le database buffer cache, le large pool, etc …

```
Select component, current_size, min_size, max_size
from v$memory_dynamic_components ;
```

8.2 Le fichier de paramètres (init.ora ou SPFILE.ORA)

Au démarrage, l'instance lit un fichier de paramètres qui contient des paramètres d'initialisation.

Ce fichier est géré par le DBA.

Les paramètres d'initialisation permettent notamment à l'instance :

- ⇨ D'allouer la mémoire souhaitée aux différentes structures de la SGA
- ⇨ De trouver le nom et l'emplacement des fichiers de contrôle de la base

Il existe 2 types de paramètres, les paramètres dynamiques et les paramètres statiques.

Les paramètres dynamiques sont modifiables sans avoir besoin d'arrêter la base de données.

Les vues V$SYSTEM_PARAMETER et V$SYSTEM_PARAMETER2 (idem V$SYSTEM_PARAMETER avec une mise en forme des paramètres) permettent de connaitre la valeur des paramètres de l'instance en cours de fonctionnement.

Règles :

- Les paramètres sont spécifiés sous la forme nom_paramètre = valeur
- Tous les paramètres sont optionnels et ont une valeur par défaut
- Des commentaires peuvent être inclus et commencent par le caractère #
- La valeur peut être spécifiée entre des guillemets doubles si elle contient des caractères spéciaux (égal, espace, …)
- Les valeurs multiples sont spécifiées entre parenthèses, séparées par des virgules

8.3 Les processus d'arrière plan

Il est important de distinguer les processus d'arrière plan des autres processus.

Ils sont indépendants de la connexion des utilisateurs. Ils sont lancés au démarrage de l'instance et arrêtés lors de l'arrêt de l'instance.

Ils réalisent des opérations sur l'instance et sur la base de données, comme l'écriture des fichiers de données, la récupération de la base de données ou la résolution des erreurs.

Certains processus aident à augmenter les performances globales du système.

Principaux processus :

- **Database Writer** (DBWRn) : écrit sur disque les données modifiées dans le Database Buffer Cache. Les informations de la base de données manipulées par les sessions transitent par ce cache dédié à cet usage.
- **Log Writer** (LGWR) : écrit sur disque le contenu du Redo Log Buffer dans les fichiers Redo.
- **Checkpoint** (CKPT) : enregistre les checkpoints dans l'en-tête des fichiers de données. Lorsque qu'un Checkpoint a lieu, toutes les informations qui se trouvent en mémoire sont enregistrées sur disque à l'emplacement prévu. Cet évènement correspond à un « jalon » permettant la restauration des données jusqu'à ce point précis dans le temps. CKPT peut à son tour déclencher DBWR et LGWR.
- **Process Monitor** (PMON) : chargé du nettoyage lors du plantage d'un processus utilisateur. Il libère les ressources de sessions qui se sont mal terminées.
- **System Monitor** (SMON) : restauration de l'instance après un arrêt anormal. C'est le gardien de la cohésion des données. Une instance cohérente est établie chaque fois que la base est démarrée.
- **Job Queue Coordinator** (CJQ) : utilisé par le *Scheduler*, il génère les processus pour exécuter les jobs planifiés qui se trouvent dans la file d'attente interne d'Oracle.
  Les utilisateurs peuvent créer des jobs et les soumettre à ce coordinateur.
  `JOB_QUEUE_PROCESSES > 0` permet de définir le nombre de jobs soumis en simultané.
- **Memory Manager** (MMAN) : il agit comme un distributeur de mémoire et coordonne la taille allouée aux différents composants.
- **Memory Monitor** (MMON) : programme et déclenche ADDM (*L'Automatic Database Diagnostic Monitor*) qui effectue des analyses pour déterminer des problèmes potentiels.

Selon la configuration du serveur, d'autres processus d'arrière plan peuvent être présents :

- *Archiver* (`ARCn`) : en base « archivée » il archive des fichiers de *Redo Log* chaque fois qu'un fichier Redo est plein.
- *Recover* (`RECO`) : gère les bases de données distribuées.
- *Dispatcher* (`Dnnnn`) : présent en serveur partagé.

- *Global Cache service* (LMS) : présent en option RAC (*Real Application Cluster*).
- *Job Queue* (SNPn) : processus chargé de rafraîchir les *snapshots* ou d'exécuter périodiquement des tâches programmées avec le package DBMS_JOB.

8.4 La base de données

La base de données est l'ensemble des fichiers qui permettent de gérer les données stockées dans de la base de données.

Une base de données est constituée de :

- **Un fichier de contrôle**, contenant les informations sur tous les autres fichiers de la base (nom, emplacement, taille).
- **Fichiers de Redo Log**, contenant l'activité des sessions connectées à la base. Ce sont des journaux de transactions de la base. Ils sont organisés en groupe possédant le même nombre de membres.

  Et éventuellement, de fichiers de Redo Log archivés contenant les archives d'anciens fichiers de Redo Log.
- **D'un ou plusieurs fichiers de données** qui contiennent les données proprement dites, elle contient à la création de la base de données au minimum :
  - Tablespace SYSTEM, contenant le dictionnaire de données.
  - Tablespace SYSAUX, c'est le tablespace auxiliaire du tablespace SYSTEM contenant des fonctions Oracle ou des données utilisées par des outils tels que le référentiel d'OEM (Oracle Enterprise Manager), placées avant dans un tablespace OEM_REPOSITORY, situées aujourd'hui dans le tablespace SYSAUX.
  - Tablespace Temporaire TEMP, récupérant les segments temporaires utilisés par les requêtes SQL de la base de données.
  - Tablespace UNDO, récupérant la version précédente des données en cours de modification par les transactions se déroulant sur la base.
  - Tablespace USERS, tablespace de travail par défaut des utilisateurs.
- **Un fichier de paramètres binaire SPFILE<SID>.ORA**, contenant les paramètres de démarrage de l'instance et d'autres valeurs qui déterminent l'environnement dans lequel elle s'exécute.
  -- Créé à partir d'un fichier de paramètres caractère (INIT<SID>.ora)
- **Un fichier de mots de passe**, contenant le mot de passe du privilège SYSDBA.

8.4.1 Les fichiers de données

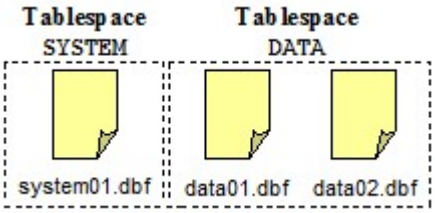
Ils contiennent les données proprement dites de la base (tables et index notamment).

Ils sont logiquement regroupés en tablespaces.

Un tablespace est une unité logique de stockage composée d'un ou plusieurs fichiers physiques.

La quasi totalité des opérations d'administration relatives au stockage s'effectue en travaillant sur le tablespace et non sur le fichier de données.

Dans la pratique, une base comportera donc d'autres fichiers de données appartenant à d'autre tablespaces.

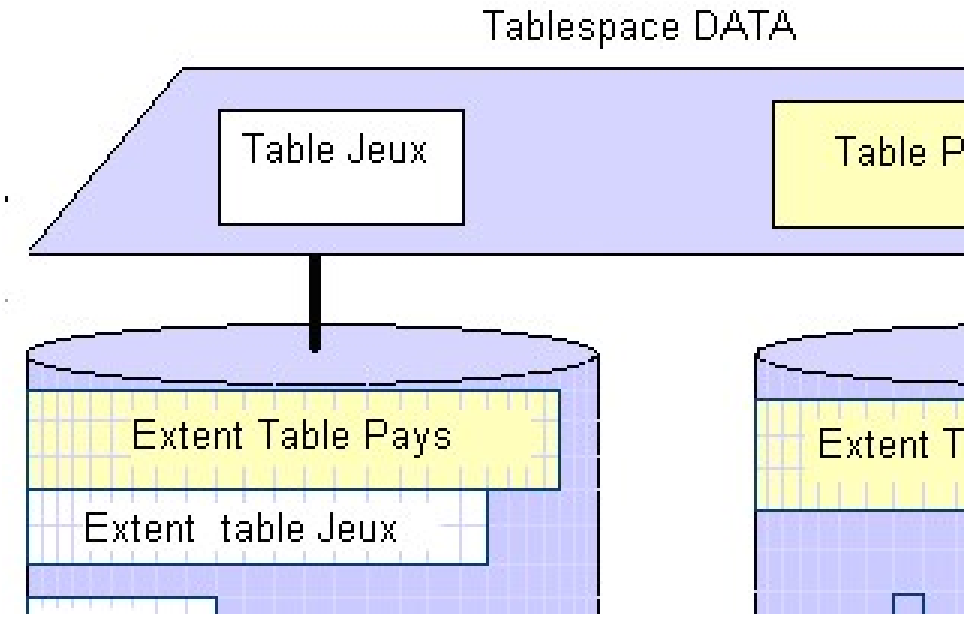


Les fichiers de données sont découpés en blocs d'une taille définie à la création de la base (2 ko, 4 ko, 8 ko, …). La taille du bloc oracle est définie par le paramètre DB_BLOC_SIZE.

L'espace occupé par un objet dans un tablespace est désigné par le terme générique de segment.

Un segment appartient à un tablespace et est constitué d'extents.

Un extent est un ensemble de blocs contigus dans un fichier de données.



Dans l'image présentée ci-dessus nous voyons que la table Pays objet logique stocké dans le tablespace DATA est constituée de 3 extents ;

- 2 extents sont stockés dans le fichier DATA01.DBF
- 1 extent est stocké dans le fichier DATA02.DBF.

9 Utilisateurs et connexion à la base de données

A la création d'une base de données un ensemble d'utilisateurs sont créés, dont SYSTEM et SYS.

SYSTEM est l'utilisateur que l'on préfèrera **pour créer les objets de schéma** tels que les users, les tables ou les index, … (SYSTEM est un utilisateur qui a des privilèges dba).

L'utilisateur **SYS** (super utilisateur) sera utilisé avec le privilège SYSDBA, pour effectuer des tâches d'administration « lourdes » telles que démarrage ou arrêt de base de données, modification de paramètres systèmes, restauration de base, bref tout ce qui concerne la structure même de la base de données ou de l'instance.

- Utiliser un autre compte SYSTEM pour l'administration courante (objets de schémas).
- Réserver le compte SYS pour les connexions AS SYSDBA
- Ne jamais créer d'objets dans le schéma SYS (autres que ceux du dictionnaire)

9.1 Syntaxe pour la connexion classique

La connexion d'un utilisateur quelconque à une base de données oracle se fait en suivant la syntaxe :

```
CONNECT utilisateur/mot_de_passe@service_OracleNet
```

```
SQLPLUS /nolog
SQL> Connect CHARLY/monpass@bora
ConnectÚ.
SQL> Connect SYSTEM/manager@bora
ConnectÚ.
```

9.2 Syntaxe pour la connexion spéciale SYSDBA ou SYSOPER

Avec une identification par le système d'exploitation

```
CONNECT / AS { SYSDBA | SYSOPER }
```

```
$ Export ORACLE_SID=TAHITI
$ sqlplus /nolog

SQL> Connect /as sysdba
ConnectÚ.
```

Avec une identification par un fichier de mot de passe

```
CONNECT utilisateur/mot_de_passe AS { SYSDBA | SYSOPER }
```

```
SQL> Connect SYS/secret as sysdba
ConnectÚ.
```

9.3 Les connexions SYSDBA et SYSOPER

SYSDBA : permet toutes les opérations « lourdes » d'administration (création, arrêt, démarrage, restauration, …).

SYSOPER : même droits que SYSDBA, à l'exception de la création de la base et des restaurations partielles.

Sur un serveur Unix ou Windows, on va vérifier que la variable d'environnement est bien positionnée avant de se connecter à la base de données.

S'assurer que l'instance souhaitée est bien désignée par la variable d'environnement ORACLE_SID, et se connecter en SYSDBA

```
Sous DOS
C:\>set oracle_sid=TAHITI
C:\>sqlplus /nolog
SQL > CONNECT /AS SYSDBA
```

```
Sous UNIX
Export ORACLE_SID=TAHITI
Echo ORACLE_SID
TAHITI

SQLPLUS /nolog
SQL> Connect /as sysdba
ConnectÚ.
```

9.4 Le fichier de mots de passe

Autrefois créé avec l'utilitaire `ORAPWD`, il est aujourd'hui créé automatiquement lors de la création de la base de données avec l'outil `dbca`.

Ce fichier protège le compte `SYS` associé au privilège `SYSDBA` permettant une administration lourde (création, démarrage, arrêt, restaurations).

```
orapwd file=<fichier> password=<mot de passe> [entries=<valeur>]
```

```
rem *** Création du fichier de mots de passe ***
C> REM orapwd
FILE=/app/oracle/product/11.2.0/dbhome_1/database/PWDtahiti.ORA
PASSWORD=secret ENTRIES=10
```

Mettre le paramètre REMOTE_LOGIN_PASSWORDFILE à EXCLUSIVE

Se connecter au système d'exploitation.

Lancer l'outil d'administration et se connecter, en tant que SYS à l'aide du mot de passe défini avec le privilège SYSDBA ou SYSOPER.

```
CONNECT sys/mot_de_passe AS { SYSDBA | SYSOPER }
```

```
SQLPLUS /nolog
SQL> Connect SYS/secret@bora as SYSDBA
ConnectÚ.
```

9.5 Les variables d'environnement

Ces variables doivent être positionnées avant le lancement de l'outil SQL*Plus en mode commande, sous le système d'exploitation.

```
Set ORACLE_SID=orcl
Sqlplus  /NOLOG
Connect  as  sysdba
```

Exemple sous unix

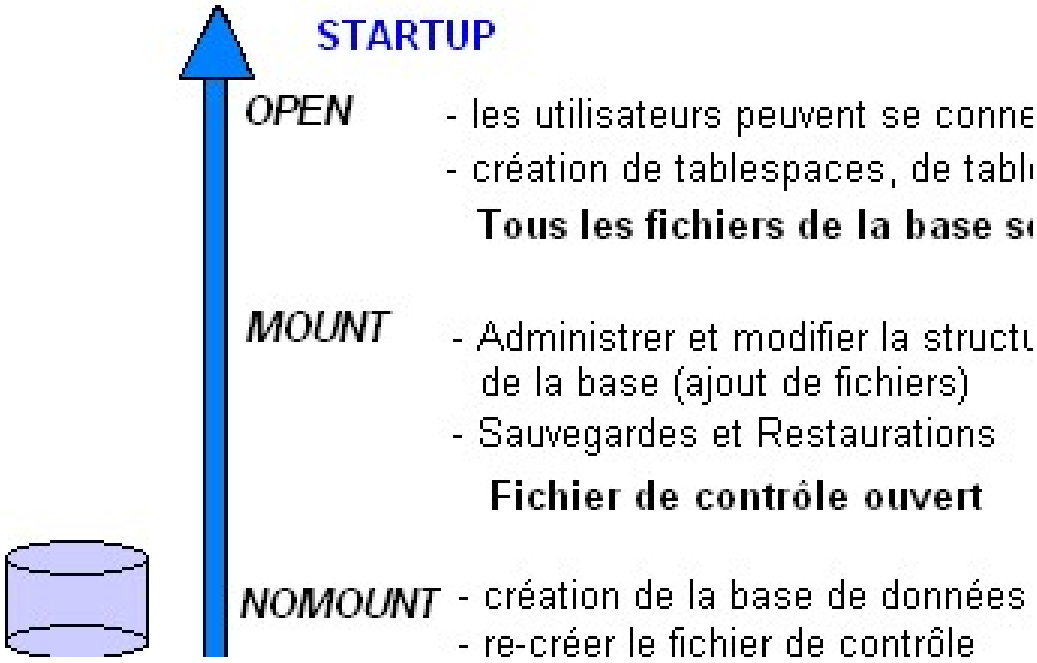
```
export ORACLE_SID=orcl
echo ORACLE_SID
orcl
Sqlplus  /NOLOG
Connect  as  sysdba
```

Les principales variables d'environnement sont

- **ORACLE_HOME = définie l'emplacement du noyau Oracle C:\oracle\product\11.2.0\dbhome_1**
- **ORACLE_BASE = définie l'emplacement des bases oracle C:\app\oracle**
- **ORACLE_SID = désigne le nom de l'instance sur laquelle on veut se positionner**
- **NLS_LANG = langage du système d'exploitation FRENCH_FRANCE.WE8MSWIN1252**

10 Démarrer & Arrêter une base de données

Une instance peut être démarrée avec 3 niveaux successifs de disponibilité de la base :



Pour rendre une base accessible à tous les utilisateurs, il faut démarrer une instance et ouvrir la base avec cette instance.

Il y a trois étapes dans le processus de démarrage :

- Démarrage de l'instance
- Montage de la base
- Ouverture de la base

Un fichier de paramètres `SPFILE` est lu lors du démarrage de l'instance. Il permet de configurer les paramètres de l'instance.

```
SQL> startup
Instance ORACLE lancÚe.

Total System Global Area  135338868 bytes
Fixed Size                   453492 bytes
Variable Size              117440512 bytes
Database Buffers            16777216 bytes
Redo Buffers                  667648 bytes
Base de donnÚes montÚe.
Base de donnÚes ouverte.
SQL>
```

De même, il y a trois étapes dans le processus d'arrêt :

- Fermeture de la base
- Démontage de la base
- Arrêt de l'instance

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
```

10.1 Démarrer la base de données

Dans SQL*Plus, la commande STARTUP permet de démarrer une instance et de lui associer une base de données avec le niveau de disponibilité souhaité.

```
STARTUP [NOMOUNT | MOUNT [nom_base] | OPEN [nom_base]]
[RESTRICT] [PFILE=nom_fichier]
;
```

- NOMOUNT | MOUNT | OPEN : niveau de disponibilité souhaité
- nom_base : nom de la base à monter ou à ouvrir
- RESTRICT : restreint l'accès à la base aux utilisateurs ayant le privilège RESTRICTED SESSION
- PFILE : nom du fichier de paramètres à utiliser

S'assurer que l'instance souhaitée est bien désignée par la variable d'environnement ORACLE_SID, et se connecter en SYSDBA.

Taper la commande `STARTUP` avec les options souhaitées, puis démarrer une instance sans associer de base (en vue d'en créer une nouvelle ou de recréer le fichier de contrôle) :

- Démarrer une instance à l'état `MOUNT` pour effectuer certaines tâches d'administration :

```
SQL> startup mount
ORACLE instance started.

Total System Global Area  159383552 bytes
Fixed Size                   788204 bytes
Variable Size             141293844 bytes
Database Buffers           16777216 bytes
Redo Buffers                 524288 bytes
Database mounted.
```

- Démarrer avec un fichier de paramètres caractère (PFILE)

```
SQL> startup pfile='D:\cours_admin10G\inittahiti02.ora';
ORACLE instance started.

Total System Global Area  159383552 bytes
Fixed Size                   788204 bytes
Variable Size             141293844 bytes
Database Buffers           16777216 bytes
Redo Buffers                 524288 bytes
Database mounted.
Database opened.
```

10.2 Modifier la disponibilité de la base de données

Si l'instance a été démarrée dans un niveau intermédiaire (NOMOUNT ou MOUNT), il est possible de la faire passer au niveau supérieur grâce à la commande SQL ALTER DATABASE :

- **NOMOUNT ⇨ MOUNT**

```
ALTER DATABASE MOUNT;
```

⇨ **MOUNT ⇨ OPEN**

```
ALTER DATABASE OPEN;
```

La commande SQL ALTER DATABASE ne permet pas de revenir à un niveau inférieur.
Pour cela, il faut arrêter la base et la redémarrer avec le niveau souhaité.

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> startup nomount
ORACLE instance started.

Total System Global Area  159383552 bytes
Fixed Size                   788204 bytes
Variable Size             141293844 bytes
Database Buffers           16777216 bytes
Redo Buffers                 524288 bytes
SQL>
SQL> alter database mount;
Base de donnÚes modifiÚe.
SQL> alter database open;
Base de donnÚes modifiÚe.
```

Pour forcer la base à redémarrer vous pouvez utiliser la commande :

```
STARTUP FORCE
```

```
SQL> startup force
ORACLE instance started.

Total System Global Area  159383552 bytes
Fixed Size                   788204 bytes
Variable Size             141293844 bytes
Database Buffers           16777216 bytes
Redo Buffers                 524288 bytes
Database mounted.
Database opened.
SQL>
```

10.3 Arrêter la base de données

Dans SQL*Plus, la commande SHUTDOWN permet d'arrêter l'instance et la base de données.

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

- NORMAL : Oracle attend que tous les utilisateurs soient déconnectés (pas de nouvelle connexion autorisée) puis ferme proprement la base.
- IMMEDIATE : Oracle déconnecte tous les utilisateurs (en effectuant un ROLLBACK des éventuelles transactions en cours) puis ferme proprement la base.
- TRANSACTIONNAL : Oracle attend que toutes les transactions en cours se terminent avant de déconnecter les utilisateurs (pas de nouvelle transaction autorisée) puis ferme et démonte proprement la base.
- ABORT : Oracle déconnecte tous les utilisateurs (sans effectuer de ROLLBACK des éventuelles transactions en cours) puis ferme brutalement la base ; une restauration de l'instance sera nécessaire lors du prochain démarrage.

Lancer l'outil d'administration et se connecter `AS SYSDBA`, en s'assurant que l'instance souhaitée est correctement désignée.

```
SQL> connect /@tahiti as sysdba
ConnectÚ.

SQL> select instance_name from v$instance;
INSTANCE_NAME
Tahiti
```

10.4 Ouvrir la base de données en mode RESTRICT

Pour ouvrir la base en mode restreint, il suffit d'ouvrir la base en précisant la clause : `ENABLE RESTRICTED SESSION`.

Lorsque vous avez placé l'instance en mode `RESTRICTED SESSION` vous pouvez effectuer des tâches d'administration en étant seul connecté.

Pour ouvrir la base en mode `RESTRICT` il faut avoir les privilèges system : `CREATE SESSION` et `RESTRICTED SESSION`

Pour ouvrir l'instance en mode `RESTRICT`, exécutez la commande :

```
STARTUP RESTRICT
```

```
SQL> startup restrict
ORACLE instance started.

Total System Global Area  159383552 bytes
Fixed Size                   788204 bytes
Variable Size             141293844 bytes
Database Buffers           16777216 bytes
Redo Buffers                 524288 bytes
Database mounted.
Database opened.

SQL> select instance_name,logins from v$instance;
INSTANCE_NAME    LOGINS
---------------- ----------
tahiti           RESTRICTED
```

Puis pour repasser l'instance en mode `NORMAL`, utilisez la commande :

```
ALTER SYSTEM DISABLE RESTRICTED SESSION ;
```

```
SQL> ALTER SYSTEM DISABLE RESTRICTED SESSION ;
System altered.
SQL> select instance_name,logins from v$instance;
INSTANCE_NAME    LOGINS
---------------- ----------
tahiti           ALLOWED
```

10.5 Mettre l'instance dans un état QUIESCE

Oracle 9i permet de mettre l'instance dans un état QUIESCE où seule l'activité de SYS et SYSTEM est autorisée pour réaliser des manipulations sur la base de données en évitant les accès concurrents.

Les autres utilisateurs ne peuvent pas travailler même s'ils possèdent un rôle DBA ou le privilège SYSDBA.

Oracle laisse les sessions actives (requêtes en cours) se terminer avant de passer l'instance dans l'état QUIESCE (ce qui peut être long).

Pendant ce temps, aucune session inactive ne peut devenir active (pas de nouvelle requête autorisée).

Pendant que l'instance est en état QUIESCE, les demandes de connexion ou les nouvelles requêtes sont mises en attente sans message (la session paraît bloquée).

La colonne ACTIVE_STATE de la vue V$INSTANCE donne l'état de la base de données :

- NORMAL = l'instance autorise tous les utilisateurs à travailler.
- QUIESCING = l'instance est en train de passer dans l'état QUIESCE, elle attend que les sessions actives deviennent inactives.
- QUIESCED = l'instance est dans l'état QUIESCE

Nécessite que la fonctionnalité de gestion des plans de ressource soit activée (*Database Ressource Manager*). Positionner le paramètre RESOURCE_MANAGER_PLAN = nom du plan (INTERNAL_PLAN qui est le plan par défaut).

```
SQL> alter system quiesce restricted;
alter system quiesce restricted
*
ERREUR Ó la ligne 1 :
ORA-25507: le gestionnaire de ressources n'a pas ÚtÚ continuellement actif
```

```
•      Mettre l'instance dans l'état QUIESCE

ALTER SYSTEM QUIESCE RESTRICTED;
•      Arrêt de l'état QUIESCE

ALTER SYSTEM UNQUIESCE;
```

10.6 Vues du dictionnaire de données

Au niveau du dictionnaire de données, pour trouver des informations sur les bases identifiées sur un serveur, consultez les vues suivantes qui sont accessibles à un utilisateur de type administrateur.

- V$INSTANCE : informations sur l'instance
- V$DATABASE : informations sur la base
- V$SGA : informations sur la SGA

- V$PARAMETER : informations sur les paramètres actifs
- V$VERSION : informations sur la version d'Oracle
- V$OPTION : informations sur les options disponibles
- DATABASE_PROPERTIES : informations sur les propriétés par défaut de la base de données
- DATABASE_SUMMARY : informations de la base sur les service déclarés, le nom du serveur, et le characterset.
- NLS_DATABASE_PARAMETERS : paramètre NLS de la base
- V$MEMORY_DYNAMIC_COMPONENTS, permet de visualiser les différentes valeurs de chaque pool, entre autre la shared_pool, le database buffer cache, le large pool, etc …

```
Select component, current_size, min_size, max_size
from v$memory_dynamic_components ;
```

La vue dynamique V$MEMORY_TARGET_ADVICE

Cette vue dynamique de performances, permet de suivre l'allocation dynamique et visualiser les différentes valeurs de l'allocation dynamique de la mémoire.

Cette vue contient les colonnes :

- Memory size : taille réelle de la mémoire totale allouée à l'instance
- Size_factor : coefficient de taille
- Estd_db_time : taille de l'instance utilisée en mémoire en moyenne par rapport aux facteurs size-factor et time_factor.
- Time_factor : coefficient de temps
- Version :

```
Select * from v$memory_target_advice ;
```

11 Gestion de l'instance et SPFILE

Au démarrage, l'instance lit un fichier de paramètres binaire SPFILE qui contient des paramètres d'initialisation. Ce fichier est géré par le DBA.

Il s'agit d'un référentiel centralisé des paramètres d'initialisation de l'instance au démarrage de la base de données en binaire qui permet d'effectuer des modifications de paramètres pendant le fonctionnement de l'instance (sans avoir besoin d'arrêter la base de données.

Règles concernant l'écriture des paramètres :

- Les paramètres sont spécifiés sous la forme nom_paramètre = valeur
- Tous les paramètres sont optionnels et ont une valeur par défaut
- Des commentaires peuvent être inclus et commencent par le caractère #
- La valeur peut être spécifiée entre des guillemets doubles si elle contient des caractères spéciaux (égal, espace, …)
- Les valeurs multiples sont spécifiées entre parenthèses, séparées par des virgules

Ces paramètres sont pris en compte directement en mémoire (paramètres dynamiques) ou uniquement dans le SPFILE (paramètres statiques). Dans dernier ce cas il faut arrêter puis redémarrer la base de données pour que la modification soit prise en compte par l'instance.

11.1 Créer le fichier du paramètre SPFILE

Un fichier de paramètres serveur peut être exporté au format texte par l'ordre SQL :

```
CREATE SPFILE [ = 'nom_pfile' ] FROM PFILE [ = 'nom_spfile' ]
;
```

Cette action nécessite une connexion SYSDBA ou SYSOPER.

```
•      se connecter as sysdba
SQL> connect /@tahiti as sysdba
ConnectÚ.

•      créer le fichier de paramètres SPFILE
SQL> create spfile from pfile='d:\tahiti\pfile\inittahiti.ora';
Fichier crÚÚ.
```

Dans l'optique de l'utilisation d'un fichier de paramètres commun à plusieurs instances (par exemple en RAC), ceux-ci peuvent être spécifiés sous la forme : « instance.paramètre », le symbole « * » désignant n'importe qu'elle instance (*_SHARDE_POOL_SIZE).

- C'est cette syntaxe qui est utilisée lors de l'export d'un fichier SPFILE.

11.2 Exporter un fichier de paramètres serveur SPFILE

Le fichier généré peut être utilisé à des fins de simple consultation ou de modification, pour créer le SPFILE à partir du PFILE (init<SID>.ora) modifié ou pour effectuer des démarrages particuliers.

```
CREATE PFILE [ = 'nom_pfile' ] FROM SPFILE [ = 'nom_spfile' ]
;
```

```
•      Exporter le fichier de paramètres SPFILE
SQL> create pfile   from spfile ;
File created.
```

Le fichier « INITorcl.ORA » est généré dans le répertoire ORACLE_HOME/database sous Windows et dans le répertoire ORACLE_HOME/dbs sous unix.

INITorcl.ORA

```
orcl.__db_cache_size=96468992
orcl.__java_pool_size=4194304
orcl.__large_pool_size=4194304
orcl.__oracle_base='C:\app\oracle'#ORACLE_BASE set from environment
orcl.__pga_aggregate_target=146800640
orcl.__sga_target=281018368
orcl.__shared_io_pool_size=0
orcl.__shared_pool_size=163577856
orcl.__streams_pool_size=4194304
*.audit_file_dest='C:\app\oracle\admin\orcl\adump'
*.audit_trail='db'
*.compatible='11.2.0.0.0'
*.control_files='C:\app\oracle\oradata\orcl\control01.ctl',
                'C:\app\oracle\flash_recovery_area\orcl\control02.ctl'
*.db_block_size=8192
*.db_domain='26.1.15'
*.db_name='orcl'
*.db_recovery_file_dest='C:\app\oracle\flash_recovery_area'
*.db_recovery_file_dest_size=4039114752
*.diagnostic_dest='C:\app\oracle'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=orclXDB)'
*.local_listener='LISTENER_ORCL'
*.memory_target=425721856
*.open_cursors=300
*.processes=150
*.remote_login_passwordfile='EXCLUSIVE'
*.undo_tablespace='UNDOTBS1'
```

Les colonnes ISSES_MODIFIABLE et ISSYS_MODIFIABLE de la vue V$PARAMETER donnent des informations sur le type de paramètre.

- La colonne ISSES_MODIFIABLE vaut TRUE ou FALSE selon que le paramètre est modifiable ou non au niveau de la session.
- La colonne ISSYS_MODIFIABLE vaut FALSE si le paramètre n'est pas modifiable au niveau du système, et DEFERRED ou IMMEDIATE selon qu'il est modifiable en différé ou immédiatement.

```
SQL> set pagesize 100
SQL> col name format A16
SQL> col value format A40
SQL> select name, value, isses_modifiable, issys_modifiable
2   from v$parameter
3   where name='control_files'
4   or name='shared_pool_size'
5   or name='sort_area_size'
6  order by name;

NAME             VALUE                                    ISSES ISSYS_MOD
---------------- ---------------------------------------- ----- ---------
control_files    D:\Oracle\oradata\TAHITI\control01.ctl, FALSE FALSE
                 D:\Oracle\oradata\TAHITI\control02.ctl
shared_pool_size 16777216                                 FALSE IMMEDIATE
sort_area_size   65536                                    TRUE  DEFERRED
```

11.3 Modifier des paramètres de l'instance ou du SPFILE

L'ordre SQL `ALTER SYSTEM` permet de modifier dynamiquement la valeur des paramètres d'initialisation.

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

- Paramètre : nom du paramètre
- Valeur : valeur attribuée au paramètre
- « COMMENT = 'texte' » : commentaire associé à la modification du paramètre. Inséré dans le fichier de paramètres serveur si ce dernier est la cible de la modification (voir la clause SCOPE).
- DEFERRED : si présent, indique que la modification ne concerne que les futures sessions, pas celles actuellement connectées. N'a de sens que si la mémoire est la cible de la modification (voir la clause SCOPE). Peut être obligatoire pour certains paramètres.
- SCOPE : définit la cible de la modification.
- MEMORY : la mémoire seulement
- SPFILE : le fichier de paramètres serveur seulement
- BOTH : les deux

```
•      Modification d'un paramètre uniquement en mémoire
SQL> SELECT value FROM v$parameter WHERE name = 'shared_pool_size';

VALUE
167772168


SQL>  ALTER SYSTEM SET SHARED_POOL_SIZE = 80M
  2   SCOPE =  memory;
SystÞme modifiÚ.
```

11.4 Vues du dictionnaire de données

Plusieurs vues du dictionnaire permettent de visualiser les paramètres :

- `V$PARAMETER` = valeur actuelle des paramètres.
- `V$PARAMETER2` = identique à `V$PARAMETER` mais avec un affichage sur plusieurs lignes des paramètres qui ont une liste de valeurs (comme le paramètre `CONTROL_FILES` par exemple).
- `V$SPPARAMETER` = contenu actuel du fichier de paramètres serveur actif. (le contenu de la vue est vide si l'instance n'utilise pas de fichier de paramètres serveur). Donne la valeur du paramètre situé dans le `SPFILE`.
- `SHOW parameter SGA` : cette commande affiche tous les parametres contenant le mot SGA dans SQL*Plus

La vue dynamique V$SYSTEM_PARAMETER

Cette vue dynamique de performances, permet de la valeur des paramètres de l'instance.

Cette vue contient les colonnes :

- NAME : Nom du paramètre (en minuscule)
- VALUE : valeur du paramètre
- DISPLAY_VALUE : valeur du paramètre avec mise en forme à l'affichage
- ISDEFAULT : TRUE si le paramètre est égal à sa valeur par défaut, FALSE autrement.
- ISSES_MODIFIABLE : TRUE si le paramètre n'est pas modifiable au niveau de la session, FALSE sinon
- ISSYS_MODIFIABLE : FALSE si le paramètre n'est pas modifiable au niveau du système, et DEFERRED s'il est modifiable en différé et IMMEDIATE s'il est modifiable immédiatement.
- ISMODIFIED : indique si le paramètre a été modifié depuis le démarrage de l'instance.
- ISDEPRECATED : TRUE si le paramètre est déprécié.

12 Jeux de caractères et paramètres NLS

12.1 Introduction

NLS a pour fonction d'adapter automatiquement à la langue locale les utilitaires de base de données et les messages d'erreur, l'ordre de tri, la date, l'heure, les conventions monétiques, numériques et calendaires.

Les opérations liées à la langue sont gérées par un certain nombre de paramètres coté client et coté serveur.

Le serveur et le client peuvent se trouver à des emplacements différents.

Au cas où chacun d'entre eux utilise des caractères différents, ORACLE fait automatiquement la conversion.

Caractéristiques du NLS :

- Prise en charge de la langue
- Prise en charge du territoire
- Prise en charge du jeu de caractères
- Tri linguistique
- Prise en charge des messages
- Formats date et heure
- Formats numériques
- Formats monétaires

Jeux de caractères de la base et jeu de caractères national

Le jeu de caractères est créé à la création de la base de données par la commande :

```
CREATE DATABASE …    clause CARACTER SET
                     clause NATIONAL CARACTER SET
```

Les jeux de caractère de la base de données et le jeu de caractères national (client + serveur) doivent être très proches.

Une base oracle possède 2 jeux de caractères :

- Jeu de caractères standard : Pour les types SQL : CHAR, VARCHAR et LOB
- Jeu de caractères national : Pour les types SQL : NCHAR, NVARCHAR et NLOB

A partir de la version 9i, le jeu de caractères national doit impérativement être un jeu de caractères UNICODE.

- 2 valeurs possibles : UTF8 et AL16UTF16

La commande ALTER SESSION permet de modifier le comportement de la session en cours, on peut changer les caractères NLS de la session :

- `ALTER SESSION SET NLS_LANGUAGE='FRENCH' ;`
- `ALTER SESSION SET NLS_TERRITORY='FRANCE' ;`
- `ALTER SESSION SET NLS_DATE_FORMAT= »DD.MM.RRRR « ;`
- `ALTER SESSION SET NLS_TIMESTAMP_FORMAT="DD.MM.RRRR HH24:MI:SSXFF" ;`
- `ALTER SESSION SET NLS_LANGUAGE=FRENCH_FRANCE.WE8MSWIN1252;`

La globalisation NLS (National Langage Support) permet

- le support du traitement des données dans les différentes représentations de caractères utilisés par le matériel
- La transparence de la différence des jeux de caractères entre le serveur et le client
- Le support d'opérations dépendantes de la langue de l'utilisateur final permettant de les spécifier par session : messages du serveur, format des dates et des nombres, ou encore tris alphabétiques
- La variable d'environnement NLS_LANG qui définit l'encodage de caractères d'un terminal client :
  - Les données transmises entre le client et le serveur sont automatiquement converties
  - L'encodage de la base de données doit être un ensemble de niveau supérieur ou équivalent pour tous les encodages clients

Si le jeu de caractères du client est différent de celui du serveur alors une conversion est opérée dans les 2 sens, mais il est conseillé d'avoir le même jeu de caractères sur le client et le serveur, car si un caractère utilisé n'a pas de correspondant dans le jeu de caractères en face, alors une perte d'information est inévitable.

La variable d'environnement NLS_LANG sur le client joue un rôle déterminant dans la conversion des caractères.

Le jeu de caractères national a été ajouté à partir de la version 8i.

Une base oracle possède 2 jeux de caractères :

- **Jeu de caractère standard**
  Pour les types SQL : CHAR, VARCHAR et LOB
- **Jeu de caractères national**
  Pour les types SQL : NCHAR, NVARCHAR et NLOB

A partir de la version 9i, le jeu de caractères national doit impérativement être un jeu de caractères UNICODE.

- 2 valeurs possibles : UTF8 et AL16UTF16

La globalisation NLS (National Langage Support) permet :

- le support du traitement des données dans les différentes représentations de caractères utilisés par le matériel.
- La transparence de la différence des jeux de caractères entre le serveur et le client
- Le support d'opérations dépendantes de la langue de l'utilisateur final permettant de les spécifier par session : messages du serveur, format des dates et des nombres, ou encore tris alphabétiques.

Paramètres par défaut :

```
23/05/08                          OPTIONS PAR DEFAUT DE LA BASE DE DONNEES
page:  1




Option                             Valeur
---------------------------------- ------------------------------------------------
DICT.BASE                          2
DEFAULT_TEMP_TABLESPACE            TEMP
DEFAULT_PERMANENT_TABLESPACE       USERS
DEFAULT_TBS_TYPE                   SMALLFILE
NLS_LANGUAGE                       AMERICAN
NLS_TERRITORY                      AMERICA
NLS_CURRENCY                       $
NLS_ISO_CURRENCY                   AMERICA
NLS_NUMERIC_CHARACTERS             .,
NLS_CHARACTERSET                   WE8MSWIN1252
NLS_CALENDAR                       GREGORIAN
NLS_DATE_FORMAT                    DD-MON-RR
NLS_DATE_LANGUAGE                  AMERICAN
NLS_SORT                           BINARY
NLS_TIME_FORMAT                    HH.MI.SSXFF AM
NLS_TIMESTAMP_FORMAT               DD-MON-RR HH.MI.SSXFF AM
NLS_TIME_TZ_FORMAT                 HH.MI.SSXFF AM TZR
NLS_TIMESTAMP_TZ_FORMAT            DD-MON-RR HH.MI.SSXFF AM TZR
NLS_DUAL_CURRENCY                  $
NLS_COMP                           BINARY
NLS_LENGTH_SEMANTICS               BYTE
NLS_NCHAR_CONV_EXCP                FALSE
NLS_NCHAR_CHARACTERSET             AL16UTF16
NLS_RDBMS_VERSION                  10.2.0.1.0
GLOBAL_DB_NAME                     NSKEPP.REGRESS.RDBMS.DEV.US.ORACLE.COM
EXPORT_VIEWS_VERSION               8
DBTIMEZONE                         00:00
```

Extrait du fichier de paramètres :

```
nls_calendar
nls_comp
nls_currency
nls_date_format
nls_date_language
nls_dual_currency
nls_iso_currency
nls_language                       AMERICAN
nls_length_semantics               BYTE
nls_nchar_conv_excp                FALSE
```

```
nls_numeric_characters
nls_sort
nls_territory                          AMERICA
nls_time_format
nls_timestamp_format
nls_timestamp_tz_format
nls_time_tz_format
```

Certains paramètres influencent l'utilisation des index par l'optimiseur :

NLS_SORT : détermine le traitement des chaînes de caractères dans les tris :

- Pas d'effet sur les tris des index
- Pas d'effet sur les tris internes d'oracle (suppressions des doublons par exemple)

NLS_COMP : détermine le comportement lors des opérations de comparaison des chaînes de caractères :

- BINARY
- ANSI
- LINGUISTIC

12.2 Migration de jeux de caractères

Faites attention lors de la migration de jeux de caractères car une colonne définie avec une longueur convenable dans un jeu de caractères peut être tronquée dans un autre jeu de caractères.

Il faut vérifier que le paramètre NLS_LANG de la base a le même jeu de caractères du système d'exploitation du client.

⇨ **Faire une sauvegarde de la base avant une migration de jeu de caractères**

L'outil CSSCAN exécuté sous le système d'exploitation permet d'afficher les problèmes possibles de la base de données à convertir. Le résultat des problèmes affichés par l'outil peut nécessiter l'intervention du support Oracle.

L'installation de l'outil *csscan* nécessite l'exécution sous SYS du script :

⇨ `?/rdbms/admin/csminst.sql`

```
C:\oracle>csscan help=y
Character Set Scanner v2.1 : Release 10.2.0.0.0 - Production on Lun. Nov. 10 12:
12:51 2008
Copyright © 1982, 2005, Oracle.  All rights reserved.
You can let Scanner prompt you for parameters by entering the CSSCAN
command followed by your username/password:

Example: CSSCAN SYSTEM/MANAGER
Or, you can control how Scanner runs by entering the CSSCAN command
followed by various parameters. To specify parameters, you use keywords:

Example: CSSCAN SYSTEM/MANAGER FULL=y TOCHAR=utf8 ARRAY=1024000 PROCESS=3
Keyword    Default Prompt Description
---------- ------- ------ --------------------------------------------------
USERID             yes    username/password
FULL       N       yes    scan entire database
```

```
USER                yes    owner of tables to be scanned
TABLE               yes    list of tables to scan
COLUMN              yes    list of columns to scan
EXCLUDE                    list of tables to exclude from scan
TOCHAR              yes    new database character set name
FROMCHAR                   current database character set name
TONCHAR                    new national character set name
FROMNCHAR                  current national character set name
ARRAY       1024000 yes    size of array fetch buffer
PROCESS     1       yes    number of concurrent scan process
MAXBLOCKS                  split table if block size exceed MAXBLOCKS
CAPTURE     N              capture convertible data
SUPPRESS                   maximum number of exceptions logged for each table
FEEDBACK                   report progress every N rows
BOUNDARIES                 list of column size boundaries for summary report
LASTRPT     N              generate report of the last database scan
LOG         scan           base file name of report files
PARFILE                    parameter file name
PRESERVE    N              preserve existing scan results
LCSD        N       no     enable language and character set detection
LCSDDATA    LOSSY   no     define the scope of the detection
HELP        N              show help screen (this screen)
QUERY       N              select clause to scan subset of tables or columns
---------- ------- ------ --------------------------------------------------
Scanner terminated successfully.
C:\oracle>
```

La migration peut se faire par EXPORT/IMPORT ou en utilisant le script CSALTER, à condition que le nouveau jeu de caractères soit un sur-ensemble de l'ancien.

Le script `csalter.plb` remplace l'instruction SQL :

- **`ALTER DATABASE CHARACTER SET nouveau_jeu_de_caractères ;`**

12.2.1 Migration du jeu de caractères par EXPORT/IMPORT

Vérifiez la convertibilité du jeu de caractères avec CSSCAN. En effet, celui-ci peut rapporter un problème de troncature de certaines colonnes de la base.

Mode opératoire :

- Exportation de la base de données.
- Création d'une nouvelle base de données dans le jeu de caractères désiré.
- Recréation de tables si nécessaire avec des colonnes plus grandes pour les données tronquées
- IMPORT des données dans la nouvelle base de données.

la variable NLS_LANGS doit avoir une valeur qui correspond au jeu de caractères de la base source dans les 2 phases.

Vues du dictionnaire de données

Les vues du dictionnaire de données intéressantes sont :

- DATABASE_PROPERTIES : informations sur les propriétés par défaut de la base de données
- NLS_DATABASE_PARAMETERS : valeurs par défaut des paramètres NLS utilisés par la base de données (inclus les 2 jeux de caractères et la version de la base).
- NLS_INSTANCE_PARAMETERS : valeurs des paramètres utilisés par l'instance.
- NLS_SESSION_PARAMETERS : valeurs des paramètres utilisés par la session.
- V$NLS_PARAMETERS : valeurs des paramètres utilisés par la session (incluent les 2 jeux de caractères de la base)
- • V$NLS_VALID_VALUE : liste des valeurs valident pour certains paramètres.

13 Créer une base de données

La naissance d'une base de données Oracle se fait lors de la conception de celle-ci.

Toute erreur à ce niveau verra la base de données affligée par des dégradations de performances importantes. Et souvent seule une nouvelle conception permettra une optimisation réelle de celle-ci.

Le processus complet de création d'une nouvelle base de données pour une application comporte les étapes suivantes :

- Conception du modèle conceptuel de données (MCD)
- Conception du modèle logique puis physique de données (MLD et MPD)
- Création de la base proprement dite (présenté dans ce chapitre)

Les différentes étapes de la création de la base de données proprement dite sont :

- Créer les répertoires sur les disques
- Préparer un nouveau fichier de paramètres `init<SID>.ora`
- Créer un fichier de paramètres serveur à partir du fichier `init<SID>.ora`
- positionner `ORACLE_SID` , au nom de l'instance
- Sous Windows uniquement, créer le service associé à l'instance en utilisant l'outil ORADIM (qui gère les services rattachés aux instances des bases oracle)
- Démarrer l'instance en état `NOMOUNT`
- Créer la base en exécutant l'ordre CREATE DATABASE
- Installer le dictionnaire de données et packages destinés au bon fonctionnement de la base de données
- Remplir la base de données avec les objets de schéma destiné aux applications
  - Création des structures de stockage adaptées (tablespaces)
  - Création du compte Oracle qui va contenir les objets de l'application (utilisateur propriétaire des objets applicatifs)
  - Création des objets de l'application dans ce compte Oracle
  - Création des utilisateurs finaux de l'application
  - Sauvegarde de la base de données

A partir de la version 10g, il faut utiliser DBCA pour créer une base de données.
- c'est bien plus facile et surtout sécurisé !
- vous pouvez aussi générer les scripts de création de la base puis les exécuter !

13.1 Présentation du script de création de la base

```
set verify off
ACCEPT sysPassword CHAR PROMPT 'Enter new password for SYS: ' HIDE
ACCEPT systemPassword CHAR PROMPT 'Enter new password for SYSTEM: ' HIDE
ACCEPT sysmanPassword CHAR PROMPT 'Enter new password for SYSMAN: ' HIDE
ACCEPT dbsnmpPassword CHAR PROMPT 'Enter new password for DBSNMP: ' HIDE
host C:\app\oracle\product\11.2.0\dbhome_1\bin\orapwd.exe
file=C:\app\oracle\product\11.2.0\dbhome_1\database\PWDtahiti.ora force=y


OLD_UMASK='umask'
umask 0027
mkdir C:\app\oracle\admin\tahiti\dpdump
mkdir C:\app\oracle\admin\tahiti\pfile
mkdir C:\app\oracle\cfgtoollogs\dbca\tahiti
mkdir C:\app\oracle\flash_recovery_area
mkdir C:\app\oracle\flash_recovery_area\tahiti
mkdir C:\app\oracle\oradata\tahiti
mkdir C:\app\oracle\product\11.2.0\dbhome_1\database
umask ${OLD_UMASK}
set ORACLE_SID=tahiti
set PATH=%ORACLE_HOME%\bin;%PATH%
C:\app\oracle\product\11.2.0\dbhome_1\bin\oradim.exe -new -sid TAHITI -startmode manual -
spfile
C:\app\oracle\product\11.2.0\dbhome_1\bin\oradim.exe -edit -sid TAHITI -startmode auto -
srvcstart system
C:\app\oracle\product\11.2.0\dbhome_1\bin\sqlplus /nolog
@C:\app\oracle\admin\tahiti\scripts\tahiti.sql
```

- Creation de la base—

```
SET VERIFY OFF
connect "SYS"/"&&sysPassword" as SYSDBA
set echo on
spool C:\app\oracle\admin\tahiti\scripts\CreateDB.log append
startup nomount pfile="C:\app\oracle\admin\tahiti\scripts\init.ora";
CREATE DATABASE "tahiti"
MAXINSTANCES 8
MAXLOGHISTORY 1
MAXLOGFILES 16
MAXLOGMEMBERS 3
MAXDATAFILES 100
CHARACTER SET AL32UTF8
NATIONAL CHARACTER SET AL16UTF16
USER SYS IDENTIFIED BY "&&sysPassword"
USER SYSTEM IDENTIFIED BY "&&systemPassword"
DATAFILE 'C:\app\oracle\oradata\tahiti\system01.dbf' SIZE 700M REUSE
AUTOEXTEND ON NEXT  10240K MAXSIZE UNLIMITED
EXTENT MANAGEMENT LOCAL
SYSAUX DATAFILE 'C:\app\oracle\oradata\tahiti\sysaux01.dbf' SIZE 600M REUSE
AUTOEXTEND ON NEXT  10240K MAXSIZE UNLIMITED
SMALLFILE DEFAULT TEMPORARY TABLESPACE TEMP TEMPFILE
'C:\app\oracle\oradata\tahiti\temp01.dbf' SIZE 20M REUSE
AUTOEXTEND ON NEXT  640K MAXSIZE UNLIMITED
SMALLFILE UNDO TABLESPACE "UNDOTBS1" DATAFILE 'C:\app\oracle\oradata\tahiti\undotbs01.dbf'
SIZE 200M REUSE
AUTOEXTEND ON NEXT  5120K MAXSIZE UNLIMITED
LOGFILE
GROUP 1 ('C:\app\oracle\oradata\tahiti\redo01.log') SIZE 51200K,
GROUP 2 ('C:\app\oracle\oradata\tahiti\redo02.log') SIZE 51200K,
GROUP 3 ('C:\app\oracle\oradata\tahiti\redo03.log') SIZE 51200K
```

```
  ;
spool off
```

- **creation du tablespace USERS –**

```
SET VERIFY OFF
connect "SYS"/"&&sysPassword" as SYSDBA
set echo on
spool C:\app\oracle\admin\tahiti\scripts\CreateDBFiles.log append
CREATE SMALLFILE TABLESPACE "USERS" LOGGING DATAFILE
'C:\app\oracle\oradata\tahiti\users01.dbf' SIZE 5M REUSE AUTOEXTEND ON NEXT  1280K MAXSIZE
UNLIMITED EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT  AUTO;
ALTER DATABASE DEFAULT TABLESPACE "USERS";
spool off
```

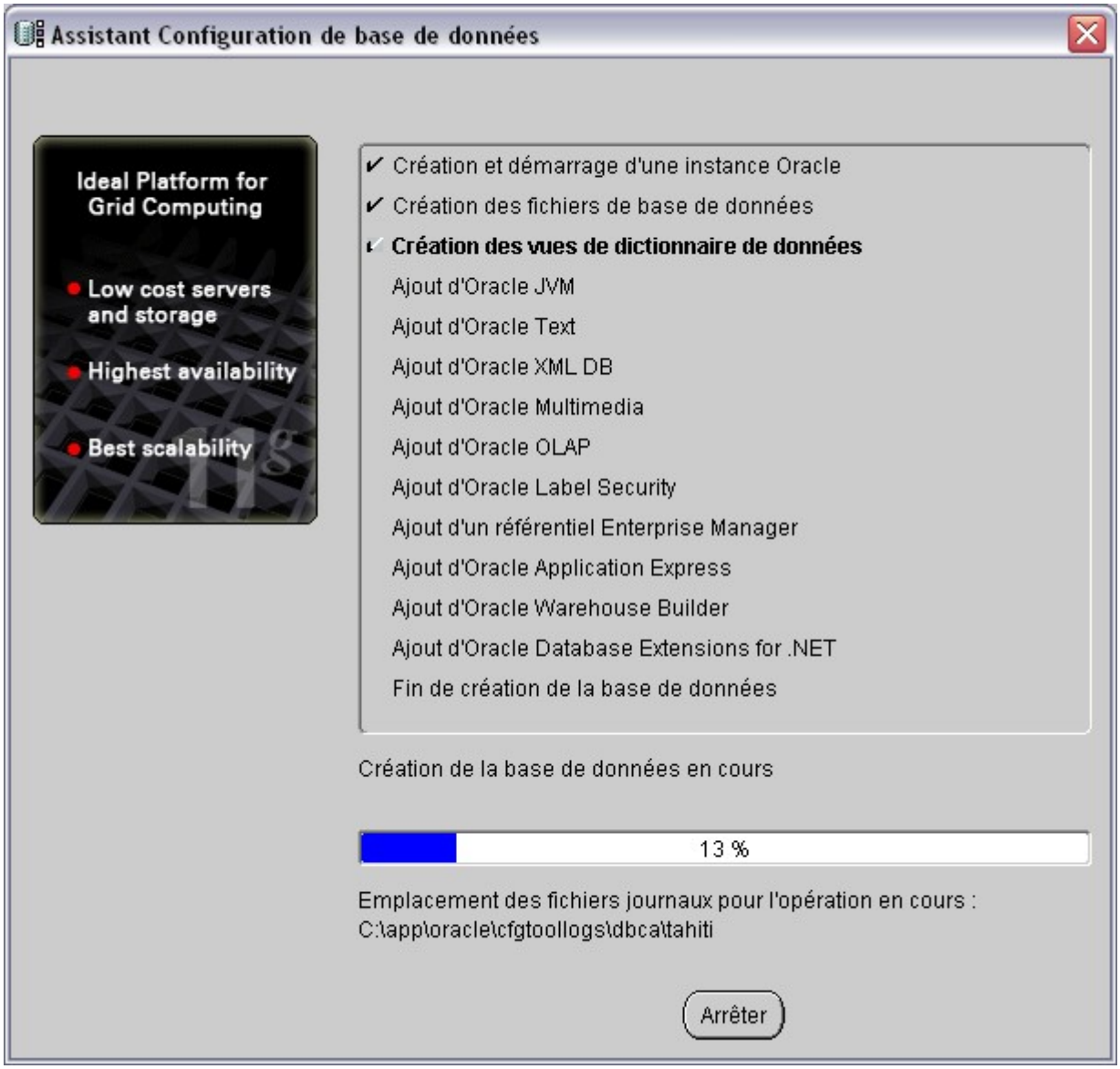
- **creation du dictionnaire de données –**

```
SET VERIFY OFF
connect "SYS"/"&&sysPassword" as SYSDBA
set echo on
spool C:\app\oracle\admin\tahiti\scripts\CreateDBCatalog.log append
@C:\app\oracle\product\11.2.0\dbhome_1\rdbms\admin\catalog.sql;
@C:\app\oracle\product\11.2.0\dbhome_1\rdbms\admin\catblock.sql;
@C:\app\oracle\product\11.2.0\dbhome_1\rdbms\admin\catproc.sql;
@C:\app\oracle\product\11.2.0\dbhome_1\rdbms\admin\catoctk.sql;
@C:\app\oracle\product\11.2.0\dbhome_1\rdbms\admin\owminst.plb;
connect "SYSTEM"/"&&systemPassword"
@C:\app\oracle\product\11.2.0\dbhome_1\sqlplus\admin\pupbld.sql;
connect "SYSTEM"/"&&systemPassword"
set echo on
spool C:\app\oracle\admin\tahiti\scripts\sqlPlusHelp.log append
@C:\app\oracle\product\11.2.0\dbhome_1\sqlplus\admin\help\hlpbld.sql helpus.sql;
spool off
```

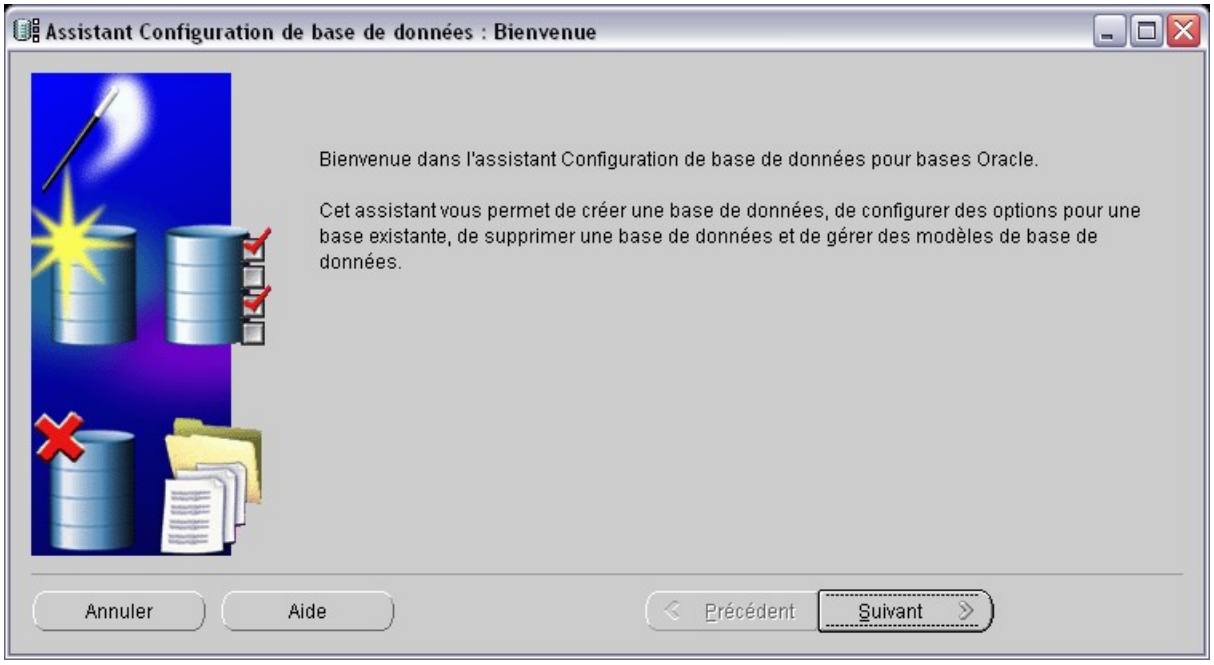
Après la création de la base de données et du dictionnaire de données vous pouvez installer des modules supplémentaires qui vous permettrons de gérer des bases stockant des données relatives à internent ou autorisant le datamining.

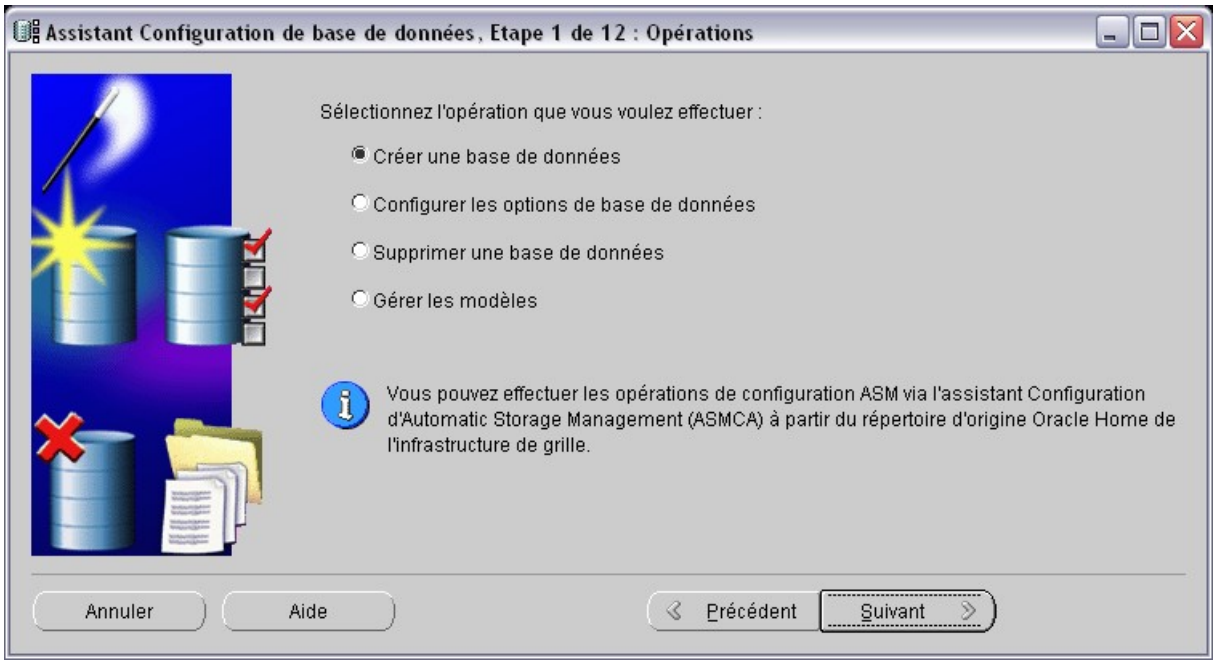
Ces scripts sont détaillés ci-dessous :

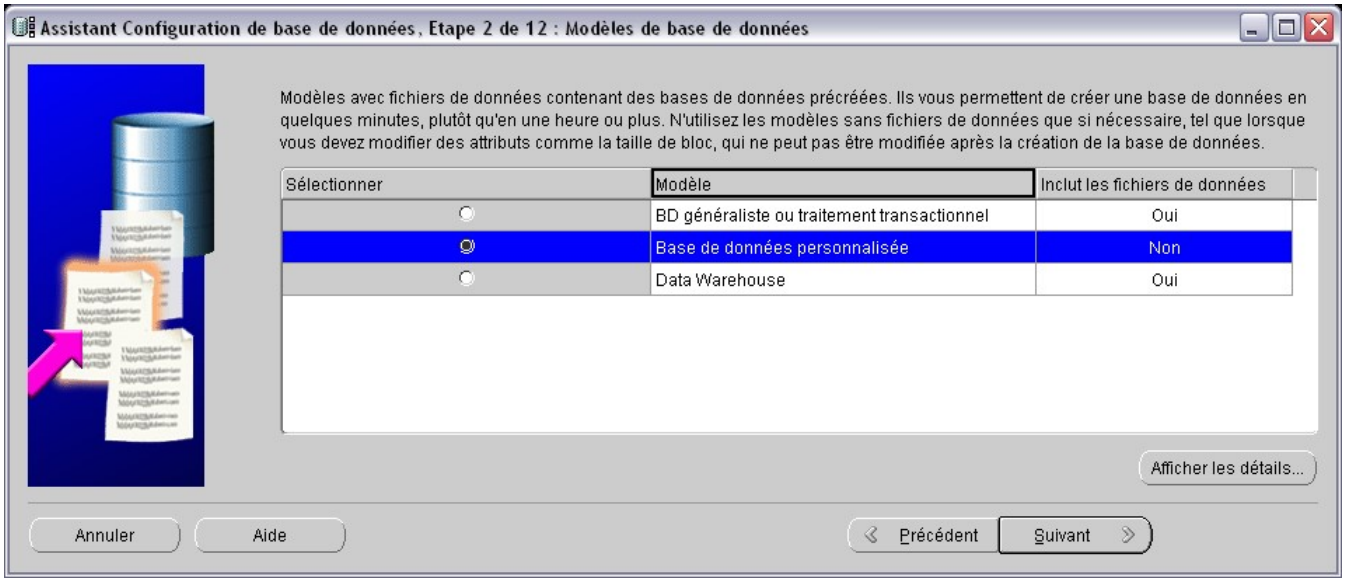
```
@C:\app\oracle\admin\tahiti\scripts\JServer.sql
@C:\app\oracle\admin\tahiti\scripts\context.sql
@C:\app\oracle\admin\tahiti\scripts\xdb_protocol.sql
@C:\app\oracle\admin\tahiti\scripts\ordinst.sql
@C:\app\oracle\admin\tahiti\scripts\interMedia.sql
@C:\app\oracle\admin\tahiti\scripts\cwmlite.sql
@C:\app\oracle\admin\tahiti\scripts\labelSecurity.sql
@C:\app\oracle\admin\tahiti\scripts\emRepository.sql
@C:\app\oracle\admin\tahiti\scripts\apex.sql
@C:\app\oracle\admin\tahiti\scripts\owb.sql
@C:\app\oracle\admin\tahiti\scripts\netExtensions.sql
@C:\app\oracle\admin\tahiti\scripts\lockAccount.sql
@C:\app\oracle\admin\tahiti\scripts\postDBCreation.sql
```

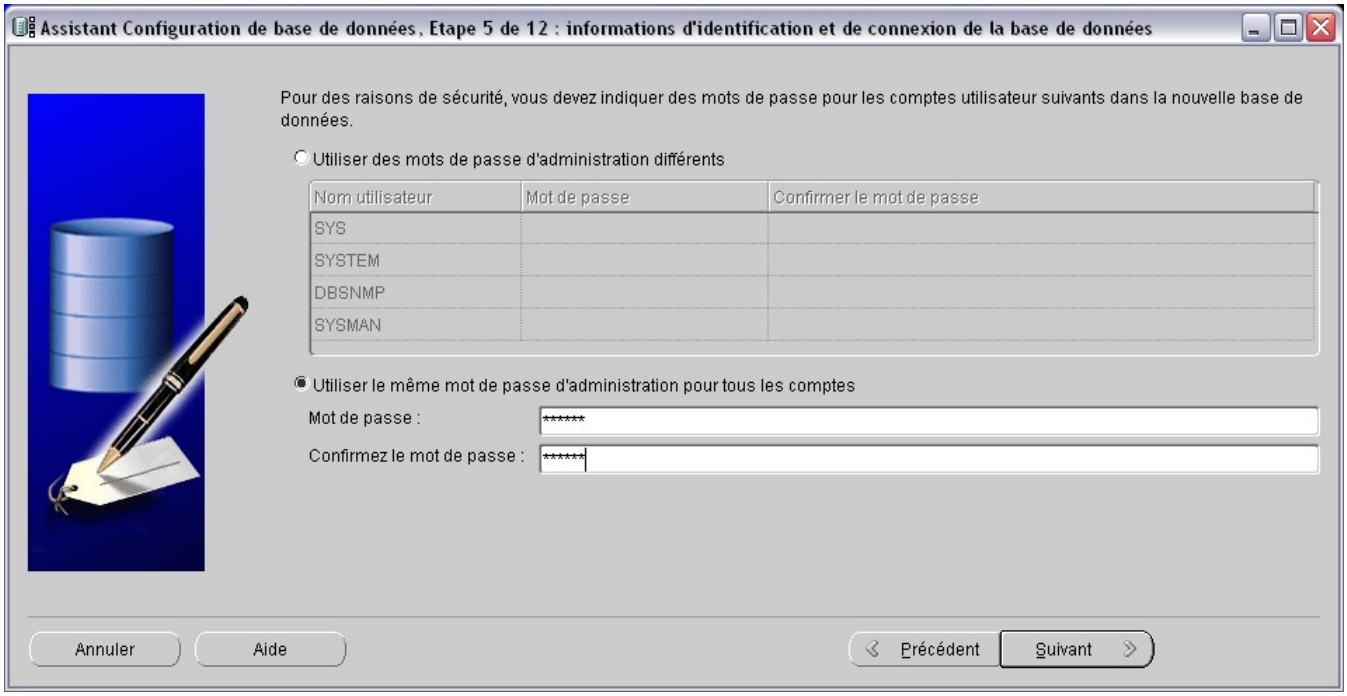

Assistant Configuration de base de données
Ideal Platform for Grid Computing
Low cost servers and storage
Highest availability
Best scalability
✔ Création et démarrage d'une instance Oracle
✔ Création des fichiers de base de données
✔ Création des vues de dictionnaire de données
Ajout d'Oracle JVM
Ajout d'Oracle Text
Ajout d'Oracle XML DB
Ajout d'Oracle Multimedia
Ajout d'Oracle OLAP
Ajout d'Oracle Label Security
Ajout d'un référentiel Enterprise Manager
Ajout d'Oracle Application Express
Ajout d'Oracle Warehouse Builder
Ajout d'Oracle Database Extensions for .NET
Fin de création de la base de données
Création de la base de données en cours
13 %
Emplacement des fichiers journaux pour l'opération en cours :
C:\app\oracle\cfgtoollogs\dbca\tahiti
Arrêter

13.2 Présentation de l'outil DBCA

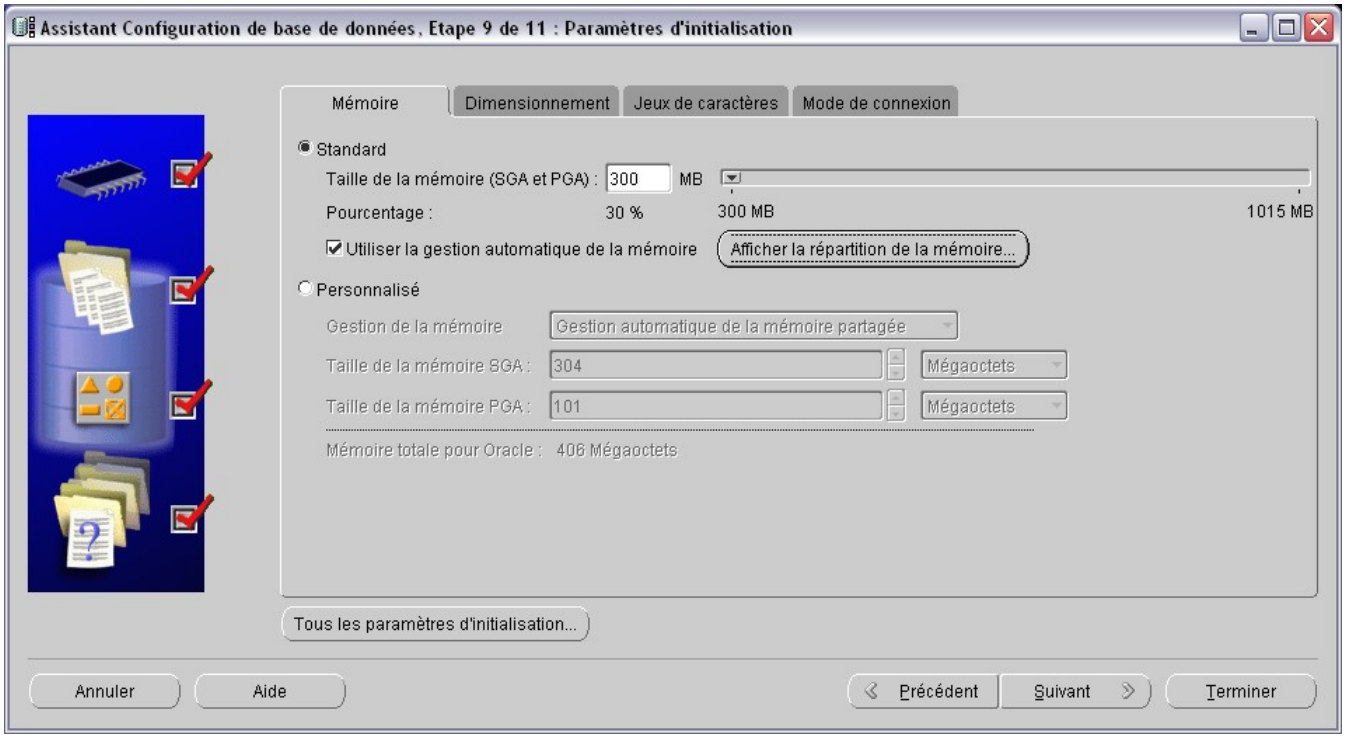


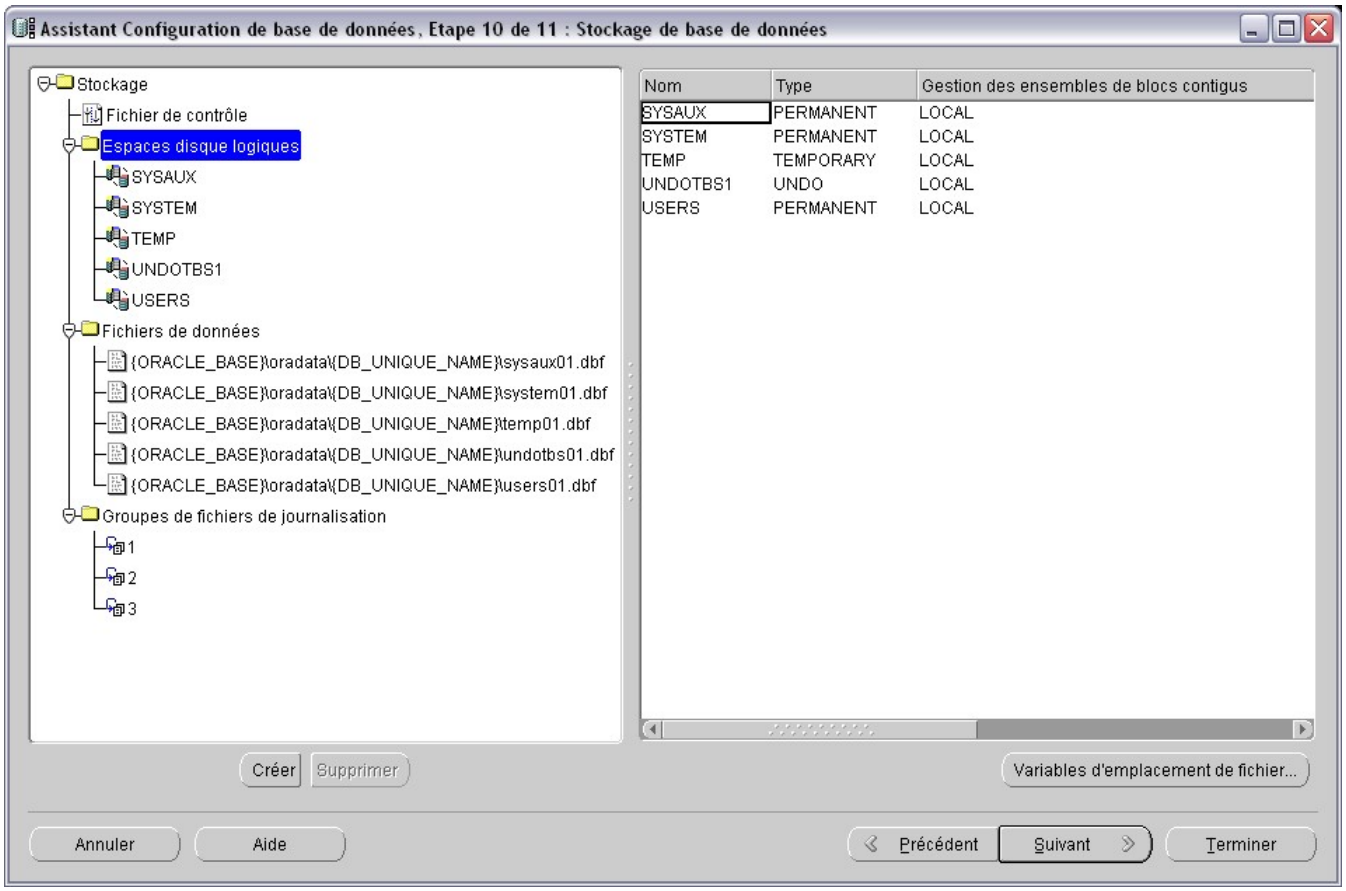






ATTENTION, à partir de la version 11g, les mots de passe sont sensibles à la casse.


Assistant Configuration de base de données, Etape 9 de 11 : Paramètres d'initialisation
Mémoire
Dimensionnement
Jeux de caractères
Mode de connexion
Standard
Taille de la mémoire (SGA et PGA) : 300 MB
Pourcentage : 30 %
300 MB
1015 MB
Utiliser la gestion automatique de la mémoire
Afficher la répartition de la mémoire...
Personnalisé
Gestion de la mémoire
Gestion automatique de la mémoire partagée
Taille de la mémoire SGA : 304 Mégaoctets
Taille de la mémoire PGA : 101 Mégaoctets
Mémoire totale pour Oracle : 406 Mégaoctets
Tous les paramètres d'initialisation...
Annuler
Aide
Précédent
Suivant
Terminer


Assistant Configuration de base de données, Etape 10 de 11 : Stockage de base de données
Stockage
Fichier de contrôle
Espaces disque logiques
SYSAUX
SYSTEM
TEMP
UNDOTBS1
USERS
Fichiers de données
{ORACLE_BASE}\oradata\{DB_UNIQUE_NAME}\sysaux01.dbf
{ORACLE_BASE}\oradata\{DB_UNIQUE_NAME}\system01.dbf
{ORACLE_BASE}\oradata\{DB_UNIQUE_NAME}\temp01.dbf
{ORACLE_BASE}\oradata\{DB_UNIQUE_NAME}\undotbs01.dbf
{ORACLE_BASE}\oradata\{DB_UNIQUE_NAME}\users01.dbf
Groupes de fichiers de journalisation
1
2
3
Nom
Type
Gestion des ensembles de blocs contigus
SYSAUX PERMANENT LOCAL
SYSTEM PERMANENT LOCAL
TEMP TEMPORARY LOCAL
UNDOTBS1 UNDO LOCAL
USERS PERMANENT LOCAL
Créer
Supprimer
Variables d'emplacement de fichier...
Annuler
Aide
Précédent
Suivant
Terminer

13.3 Valeurs des paramètres

Les paramètres utilisés dans le fichier SPFILE sont modifiables. Ils représentent les valeurs suivantes :

- **DB_NAME**
  Nom de la base (jusqu'à 8 caractères)
  Généralement égal au nom de l'instance (ORACLE_SID)

- **DB_DOMAIN**
  Localisation logique de la base sur le réseau (jusqu'à 128 caractères)
  Permet à Oracle de construire le nom global de la base = DB_NAME.DB_DOMAIN
  Important si la base appartient à un système distribué (ou est susceptible de l'être)
  ```
  Valeur par défaut : WORLD
     DB_DOMAIN = PARIS.ORA.FR
  ```

- **CONTROL_FILES**
  Emplacement des fichiers de contrôle de la base
  En spécifier au minimum 2, si possible sur des disques différents (dans l'idéal, un par disque)
  ```
  CONTROL_FILES = C:\ORACLE\PRODUCT\10.1.0\ORADATA\TAHITI\CONTROL01.CTL,
                  D:\ORACLE\PRODUCT\10.1.0\ORADATA\TAHITI\CONTROL02.CTL
  ```

- **NLS_LANGUAGE**
  Langage par défaut de l'instance, utilisé pour les messages, la date et l'heure. La valeur par défaut est dérivée du paramètre NLS_LANG.
  ```
  NLS_LANGUAGE = french
  ```

- **NLS_TERRITORY**
  Territoire par défaut de l'instance, utilisé pour la numérotation des jours et des semaines. Détermine également la valeur par défaut des formats de date, des séparateurs numériques et des symboles monétaires.
  ```
  NLS_TERRITORY = France
  ```

- **DB_BLOCK_SIZE**
  Taille en octets d'un bloc de données (compris entre 2 ko et 32 ko)
  Doit être un multiple de la taille de bloc du système d'exploitation
  Ne peut pas être modifié ultérieurement sans recréer la base
  ```
  DB_BLOCK_SIZE = 8192
  ```

- **COMPATIBLE**
  Paramètre de compatibilité, prend la valeur 11.2.0.0 par défaut.
  ```
  compatible = 11.2.0.0
  ```

- **DIAGNOSTIC_DEST**
  Apparu en version 11, ce paramètre définit la destination des fichiers de trace générés par la base de données.
  ```
  diagnostic_dest='C:\app\oracle'
  ```

- **MEMORY_TARGET**

  Apparu en version 11, si ce paramètre a une valeur différente de zéro, la gestion automatique de la mémoire est activée. Dans ce cas les paramètres SGA_TARGET et PGA_AGREGATE sont dépréciés. Oracle aura une réserve de mémoire vive en cas de besoin.

  ```
  memory_target=425721856
  ```

- **MEMORY_MAX_SIZE**

  Apparu en version 11, ce paramètre délimite la taille totale de la SGA et de la PGA utilisée par l'instance sur le serveur. Il doit être adapté à MEMORY_TARGET.

  ```
  memory_max_size=425721856
  ```

- **REMOTE_LOGIN_PASSWORDFILE**

  A positionner selon la stratégie adoptée pour l'identification SYSDBA

  NONE = pas de fichier de mots de passe – identification par l'OS

  EXCLUSIVE = utilisation d'un fichier de mots de passe dédié à une base

  SHARED = utilisation d'un fichier de mots de passe partagé entre plusieurs bases

  ```
  REMOTE_LOGIN_PASSWORDFILE = EXCLUSIVE
  ```

- **UNDO_TABLESPACE**

  Permet de spécifier le nom du tablespace contenant les segments d'annulation.
  Si le nom du tablespace spécifié ne correspond pas au nom du tablespace UNDO de la base une erreur apparaîtra dans le fichier des alertes.
  Valeur par défaut : chaque base de données contient 0 ou plusieurs espaces disque logiques d'annulation. En mode SMU, un seul espace disque logique de ce type est affecté à chaque instance ORACLE.

  ```
  UNDO_TABLESPACE = UNDOTBS
  ```

- **PROCESSES**

  Permet de limiter le nombre de processus simultanés sur le serveur.
  Pour connaître le nombre de processus d'arrière plan utilisez la vue V$BGPROCESS.

- **OPEN_CURSOR**

  Nombre maximum de curseurs ouverts en simultané. Compter 1 pour chaque session ouverte en simultanée et un pour chaque utilisateur interne à Oracle comme SYSMAN ou DBSNMP. .
  Ouvrir un grand nombre de curseurs évite une erreur de dépassement et n'a aucune incidence sur la base.

  ```
  OPEN_CURSOR = 500
  ```

- **CURSOR_SHARING =** EXACT

  Description : ce paramètre contrôle les instructions SQL qui peuvent partager le même curseur.
  Plage de valeurs :
  FORCE : oblige les instructions ne différant que par certains littéraux à partager un curseur, à moins que les littéraux ne modifient le sens de l'instruction.
  EXACT : seules les instructions SQL identiques partagent un curseur.
  Valeur par default : EXACT

➪ **STATISTICS_LEVEL**
Niveau de collecte des statistiques sur la base de données et le système utilisés.
Valeurs possibles : `BASIC, TYPICAL` (par défaut), `ALL`
`BASIC` désactive la gestion automatique des statistiques
`TYPICAL` permet de bénéficier des fonctionnalités de la gestion automatique de la version 10g
`ALL` collecte d'avantage de statistiques mais a un impact sur les performances

➪ **CLUSTER_DATABASE_INSTANCES = 1**
Description : nombre d'instances actuellement configurées comme éléments de la base de données de cluster. Ce paramètre permet de définir la taille des structures SGA, qui dépend du nombre d'instances configurées. L'attribution d'une valeur appropriée à ce paramètre optimisera l'utilisation de la mémoire SGA. Plusieurs paramètres sont calculés via ce nombre.
Plage de valeurs : toute valeur non nulle
Valeur par défaut : 1

➪ **CLUSTER_DATABASE = FALSE**
Description : paramètrer CLUSTER_DATABASE sur TRUE pour activer l'option Real Application Clusters.
Plage de valeurs : TRUE | FALSE
Valeur par défaut : FALSE

➪ **DB_RECOVERY_FILE_DEST**
Emplacement de la zone de récupération rapide (*flash recovery area*). Si ce paramètre est spécifié, il faut spécifier le paramètre `DB_RECOVERY_FILE_DEST_SIZE` .
`DB_RECOVERY_FILE_DEST = d:\oracle\Flash_recovery_area`

➪ **DB_RECOVERY_FILE_DEST_SIZE**
Taille maximum autorisée des fichiers stockés dans la zone de récupération rapide, définie en octets, Ko (K), Mo (M) ou en Go (G).
`DB_RECOVERY_FILE_DEST_SIZE = 30G`

➪ **AUDIT_FILE_DEST = {ORACLE_BASE}\ADMIN\{DB_UNIQUE_NAME}\ADUMP**
Description : chaque connexion SYSDBA ou INTERNAL à la base de données génère un fichier d'audit dans ce répertoire (UNIX uniquement).
Plage de valeurs : tout nom de répertoire valide
Valeur par défaut : ORACLE_HOME/rdbms/audit

➪ **AUDIT_TRAIL = DB**
Description : active ou désactive l'option d'audit de la base de données. Les enregistrements d'audit sont écrits dans la table SYS.AUD$ lorsque le paramètre a la valeur TRUE ou DB, ou dans un fichier du système d'exploitation lorsque le paramètre a la valeur OS.
Plage de valeurs : NONE | FALSE | DB | TRUE | OS
Valeur par défaut : NONE

➪ **CORE_DUMP_DEST = ?\RDBMS\TRACE**
Description : nom de répertoire, indiquant l'emplacement de vidage de la mémoire (sous UNIX). Plage de valeurs : tout nom de répertoire valide
Valeur par défaut : ORACLE_HOME/dbs

13.4 Vues du dictionnaire de données

Les vues du dictionnaire de données intéressantes sont :

- `V$INSTANCE` : informations sur l'instance
- `V$DATABASE` : informations sur la base de données
- `V$VERSION` : informations sur la version Oracle utilisée par la base de données
- `DATABASE_PROPERTIES` : informations sur les propriétés par défaut de la base de données

13.5 EMCA : Création de l'OEM repository (Database Control)

La plupart du temps l'installation d'entreprise manager se fait lors de la création de la base (si vous n'opter pas pour le grid control), mais vous pouvez toujours laisser l'installation d'entreprise manager après la création de la base.
Pour l'installer il y a plusieurs méthodes soit l'outil graphique DBCA, soit un programme en ligne de commande EMCA (*Entreprise Manager Configuration Assistant*) c'est celui dont nous allons parler.
la syntaxe générale de l'outil EMCA est :

```
Emca operation mode flag parameters
```

Vous pouvez voire la liste complète en tapant :

```
>emca –h
```

Pour créer votre console d'administration la création de la base tapez :

```
$ emca -config dbcontrol db -repos create

STARTED EMCA at jun 06, 2010 9:21:39 PM
EM Configuration Assistant, Version 10.2.0.1.0 Production
Copyright © 2003, 2005, Oracle. All rights reserved.

Enter the following information:
Database SID: db10
Listener port number: 1521
Password for SYS user: change_on_install
Password for DBSNMP user: manager
Password for SYSMAN user: manager
Email address for notifications (optional):
Outgoing Mail (SMTP) server for notifications (optional):
-----------------------------------------------------------------

You have specified the following settings

Database ORACLE_HOME ................ /u01/app/oracle/product/10.2.0/db_1

Database hostname ................ dbserver
Listener port number ................ 1521
Database SID ................ db10
Email address for notifications ...............
Outgoing Mail (SMTP) server for notifications ...............

-----------------------------------------------------------------
```

```
Do you wish to continue? [yes(Y)/no(N)]: Y
jun 06, 2010 10:00:12 PM oracle.sysman.emcp.EMConfig perform
INFO: This operation is being logged at
/u01/app/oracle/product/10.2.0/db_1/cfgtoollogs/emca/db10/emca_2010-01-06_09-21-
39-PM.log.
jun 06, 2010 10:00:15 PM oracle.sysman.emcp.EMReposConfig createRepository
INFO: Creating the EM repository (this may take a while) ...
jun 06, 2010 10:05:51 PM oracle.sysman.emcp.EMReposConfig invoke
INFO: Repository successfully created
jun 06, 2010 10:06:01 PM oracle.sysman.emcp.util.DBControlUtil startOMS
INFO: Starting Database Control (this may take a while) ...
jun 06, 2010 10:07:49 PM oracle.sysman.emcp.EMDBPostConfig performConfiguration
INFO: Database Control started successfully
jun 06, 2010 10:07:49 PM oracle.sysman.emcp.EMDBPostConfig performConfiguration
INFO: >>>>>>>>>>> The Database Control URL is http://dbserver:1158/em <<<<<<<<<<<
Enterprise Manager configuration completed successfully
FINISHED EMCA at jun 06, 2010 10:07:49 PM
```

Vous devez fournir quelques informations comme le SID le port listener.., cela prends quelques minutes et vous pouvez suivre les différentes étapes de création,

Une fois le programme terminé noter l'adresse URL qui apparait :

- http(s)://nommachine:port/em) est lancer la avec votre navigateur

Pour recréer la console il suffit de changer l'ordre CREATE par RECREATE dans la commande précédente.

Si vous êtes voulez désinstaller la DB Console, utilisez la commande suivante:

```
$ emca -deconfig dbcontrol db -repos drop

STARTED EMCA at jun 06, 2006 9:53:55 PM
EM Configuration Assistant, Version 10.2.0.1.0 Production
Copyright © 2003, 2005, Oracle. All rights reserved.

Enter the following information:
Database SID: db10
Listener port number: 1521
Password for SYS user: change_on_install
Password for SYSMAN user: manager

Do you wish to continue? [yes(Y)/no(N)]: Y
jun 06, 2006 9:54:15 PM oracle.sysman.emcp.EMConfig perform
INFO: This operation is being logged at
/u01/app/oracle/product/10.2.0/db_1/cfgtoollogs/emca/db10/emca_2006-01-06_09-53-
55-PM.log.
jun 06, 2006 9:54:16 PM oracle.sysman.emcp.util.DBControlUtil stopOMS
INFO: Stopping Database Control (this may take a while) ...
jun 06, 2006 9:54:35 PM oracle.sysman.emcp.EMReposConfig dropRepository
INFO: Dropping the EM repository (this may take a while) ...
jun 06, 2006 9:56:48 PM oracle.sysman.emcp.EMReposConfig invoke
INFO: Repository successfully dropped
Enterprise Manager configuration completed successfully
FINISHED EMCA at jun 06, 2006 9:56:48 PM
```

Pour vérifier le statut de la DB Console :

```
>emctl status dbconsole—qui fournit le statut de la console web.
```

Et :

```
>emctl status agent—pour le status de l'agent d'entreprise manager
```

Vous pouvez arrêter la console avec :

```
>emctl stop dbconsole
```

Ou la démarrer avec :

```
>emctl start dbconsole
```

14 Automatiser le démarrage de la base

Automatiser le démarrage et l'arrêt de la base lors du démarrage ou de l'arrêt du système dépend de la plate-forme.

14.1 Sous unix

Dans le fichier /etc/oratab, mettre une entrée pour chaque instance avec le format suivant :

```
<ORACLE_SID>:<ORACLE_HOME>:{Y|N}
```

```
TAHITI:/u01/app/oracle/product/10.1.0.3.:Y
```

Au démarrage et à l'arrêt, le système appelle les scripts `dbstart` et `dbshut` qui lisent le fichier `oratab` pour identifier les bases à démarrer ou arrêter, ces scripts peuvent éventuellement être appelés manuellement pour démarrer ou arrêter les bases configurées à « `Y` » dans `oratab`.

14.2 Sous Windows

Pour démarrer automatiquement une base au démarrage du système, il faut :

- Mettre le service (`OracleService<SID>`) associé à l'instance en démarrage automatique
- S'assurer que dans la base de registre (`HKEY_LOCAL_MACHINE\ SOFTWARE\ORACLE\HOMEx)`, `ORA_<SID>_AUTOSTART` est à `TRUE`
- `ORA_<SID>_PFILE` chemin + nom du fichier de paramètres texte standard, vide ou inexistant pour un fichier de paramètres serveur. Pour démarrer avec un autre fichier de paramètres serveur, utilisez la technique du fichier de paramètres texte contenant un paramètre `SPFILE`

Problèmes liés au fichier de paramètres serveur `SPFILE` :

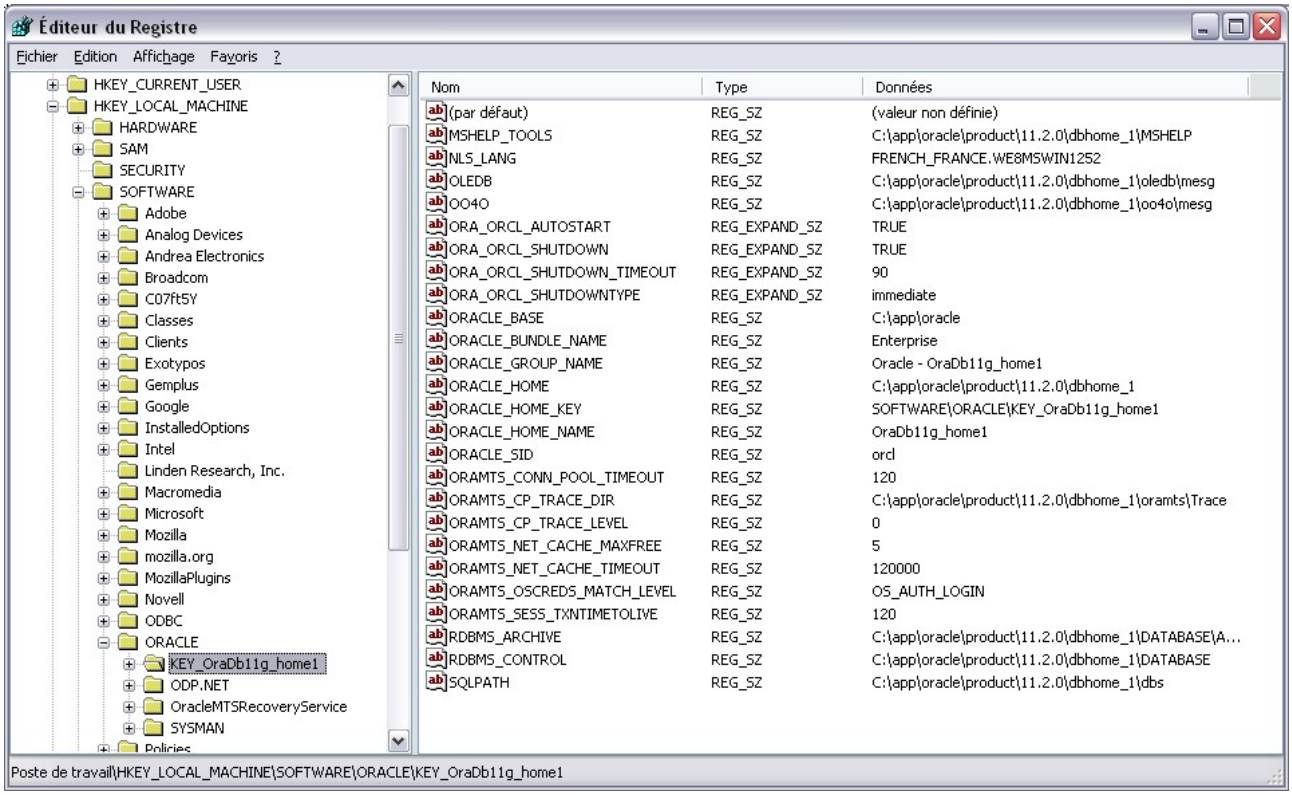
- Si le paramètre `ORA_<SID>_PFILE` contient une valeur erronée, l'instance ne redémarre pas.
- Si le paramètre `ORA_<SID>_PFILE` est vide ou n'existe pas, la séquence de recherche d'un fichier de paramètres texte ou serveur s'effectue en suivant la séquence du startup.
  - ⇨ `spfile<SID>.ora`
  - ⇨ `spfile.ora (!)`
  - ⇨ `init<SID>.ora`

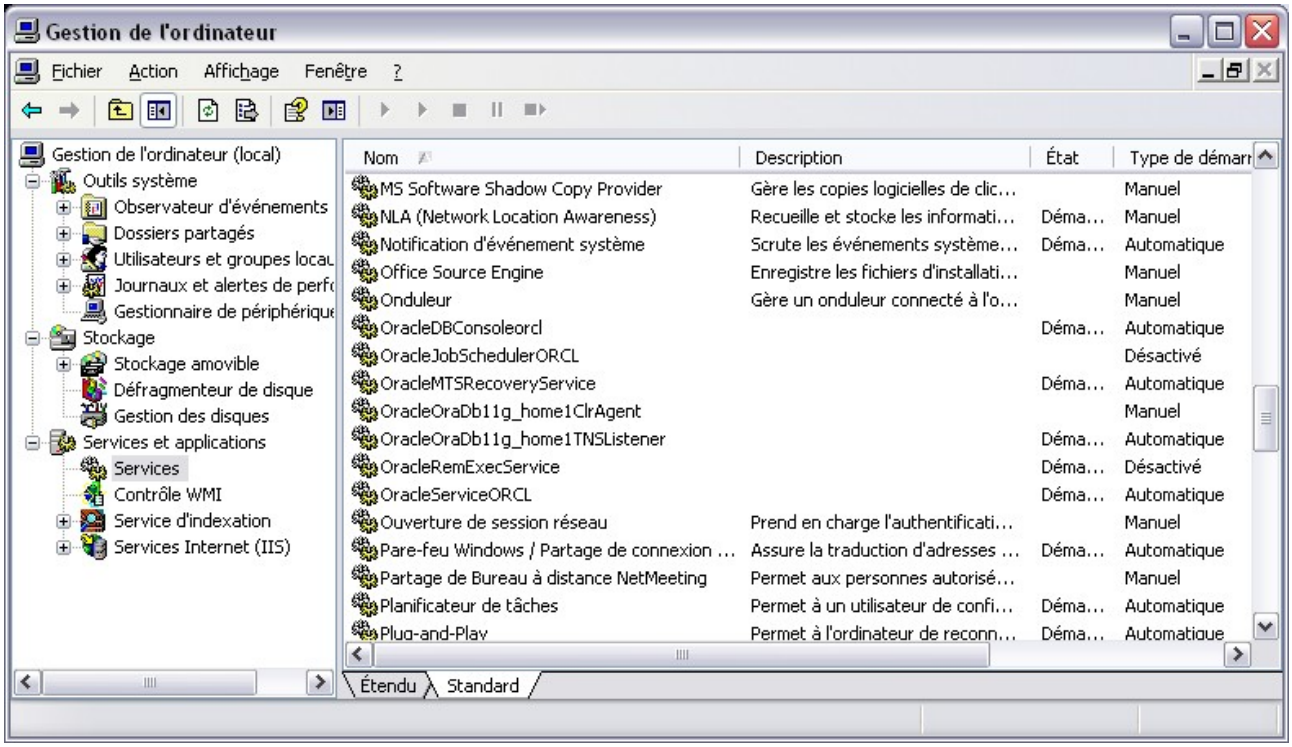
Pour arrêter automatiquement une base lors de l'arrêt du système, il faut :

- S'assurer que dans la base de registre :

`HKEY_LOCAL_MACHINE\ SOFTWARE\ORACLE\HOMEx, ORA_<SID>_SHUTDOWN` est à `TRUE`

et ajuster éventuellement `ORA_<SID>_SHUTDOWNTYPE` et `ORA_<SID>_SHUTDOWN_TIMEOUT`


Éditeur du Registre
Fichier Edition Affichage Favoris ?
HKEY_CURRENT_USER
HKEY_LOCAL_MACHINE
HARDWARE
SAM
SECURITY
SOFTWARE
Adobe
Analog Devices
Andrea Electronics
Broadcom
C07ft5Y
Classes
Clients
Exotypos
Gemplus
Google
InstalledOptions
Intel
Linden Research, Inc.
Macromedia
Microsoft
Mozilla
mozilla.org
MozillaPlugins
Novell
ODBC
ORACLE
KEY_OraDb11g_home1
ODP.NET
OracleMTSRecoveryService
SYSMAN
Nom Type Données
(par défaut) REG_SZ (valeur non définie)
MSHELP_TOOLS REG_SZ C:\app\oracle\product\11.2.0\dbhome_1\MSHELP
NLS_LANG REG_SZ FRENCH_FRANCE.WE8MSWIN1252
OLEDB REG_SZ C:\app\oracle\product\11.2.0\dbhome_1\oledb\mesg
OO4O REG_SZ C:\app\oracle\product\11.2.0\dbhome_1\oo4o\mesg
ORA_ORCL_AUTOSTART REG_EXPAND_SZ TRUE
ORA_ORCL_SHUTDOWN REG_EXPAND_SZ TRUE
ORA_ORCL_SHUTDOWN_TIMEOUT REG_EXPAND_SZ 90
ORA_ORCL_SHUTDOWNTYPE REG_EXPAND_SZ immediate
ORACLE_BASE REG_SZ C:\app\oracle
ORACLE_BUNDLE_NAME REG_SZ Enterprise
ORACLE_GROUP_NAME REG_SZ Oracle - OraDb11g_home1
ORACLE_HOME REG_SZ C:\app\oracle\product\11.2.0\dbhome_1
ORACLE_HOME_KEY REG_SZ SOFTWARE\ORACLE\KEY_OraDb11g_home1
ORACLE_HOME_NAME REG_SZ OraDb11g_home1
ORACLE_SID REG_SZ orcl
ORAMTS_CONN_POOL_TIMEOUT REG_SZ 120
ORAMTS_CP_TRACE_DIR REG_SZ C:\app\oracle\product\11.2.0\dbhome_1\oramts\Trace
ORAMTS_CP_TRACE_LEVEL REG_SZ 0
ORAMTS_NET_CACHE_MAXFREE REG_SZ 5
ORAMTS_NET_CACHE_TIMEOUT REG_SZ 120000
ORAMTS_OSCREDS_MATCH_LEVEL REG_SZ OS_AUTH_LOGIN
ORAMTS_SESS_TXNTIMETOLIVE REG_SZ 120
RDBMS_ARCHIVE REG_SZ C:\app\oracle\product\11.2.0\dbhome_1\DATABASE\A...
RDBMS_CONTROL REG_SZ C:\app\oracle\product\11.2.0\dbhome_1\DATABASE
SQLPATH REG_SZ C:\app\oracle\product\11.2.0\dbhome_1\dbs
Poste de travail\HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\KEY_OraDb11g_home1


Gestion de l'ordinateur
Fichier Action Affichage Fenêtre ?
Gestion de l'ordinateur (local)
Outils système
Observateur d'événements
Dossiers partagés
Utilisateurs et groupes locau
Journaux et alertes de perf
Gestionnaire de périphérique
Stockage
Stockage amovible
Défragmenteur de disque
Gestion des disques
Services et applications
Services
Contrôle WMI
Service d'indexation
Services Internet (IIS)
Nom Description État Type de démarr
MS Software Shadow Copy Provider Gère les copies logicielles de clic... Manuel
NLA (Network Location Awareness) Recueille et stocke les informati... Déma... Manuel
Notification d'événement système Scrute les événements système... Déma... Automatique
Office Source Engine Enregistre les fichiers d'installati... Manuel
Onduleur Gère un onduleur connecté à l'o... Manuel
OracleDBConsoleorcl Déma... Automatique
OracleJobSchedulerORCL Désactivé
OracleMTSRecoveryService Déma... Automatique
OracleOraDb11g_home1ClrAgent Manuel
OracleOraDb11g_home1TNSListener Déma... Automatique
OracleRemExecService Déma... Désactivé
OracleServiceORCL Déma... Automatique
Ouverture de session réseau Prend en charge l'authentificati... Manuel
Pare-feu Windows / Partage de connexion ... Assure la traduction d'adresses ... Déma... Automatique
Partage de Bureau à distance NetMeeting Permet aux personnes autorisé... Manuel
Planificateur de tâches Permet à un utilisateur de confi... Déma... Automatique
Plug-and-Play Permet à l'ordinateur de reconn... Déma... Automatique
Étendu Standard

15 Accéder à une base distante

Oracle Net permet à des produits Oracle situés sur des machines différentes de communiquer entre eux. Ainsi Oracle Net rend le réseau transparent et permet le transfert de données entre les 2 machines.

Oracle Net a pour objectif de rendre le réseau « transparent » pour les applications.
- Oracle Net doit être installé sur chaque machine du réseau.

Le processus d'écoute LISTENER placé sur le serveur (coté instance de base de données) permet la connexion des clients à l'instance. Celle-ci est vue comme un service (paramètre SERVICES_NAME) dans l'instance). Le LISTENER est configuré via le fichier LISTNER.ORA.

Plusieurs méthodes de connexions peuvent être utilisées :

- Locale : le fichier TNSNAMES.ORA est configuré sur le poste client et se charge de la résolution du service Oracle Net.
- Simplifiée, (easy connect naming), permettant une connexion via l'adresse du service à travers le réseau TCP/IP.
- LDAP (directory naming), un annuaire LDAP se charge de la résolution du nom de service. Cette méthode nécessite un produit tiers.

15.1 Configuration coté serveur

Pour permettre à un client de se connecter à une base de données distante, il faut d'abord configurer le LISTENER.

Le LISTENER se matérialise par un service (Oracle<NomHome>TNSListener) sur plate-forme Windows ou par un processus (tnslsnr) sur plate-forme Unix.

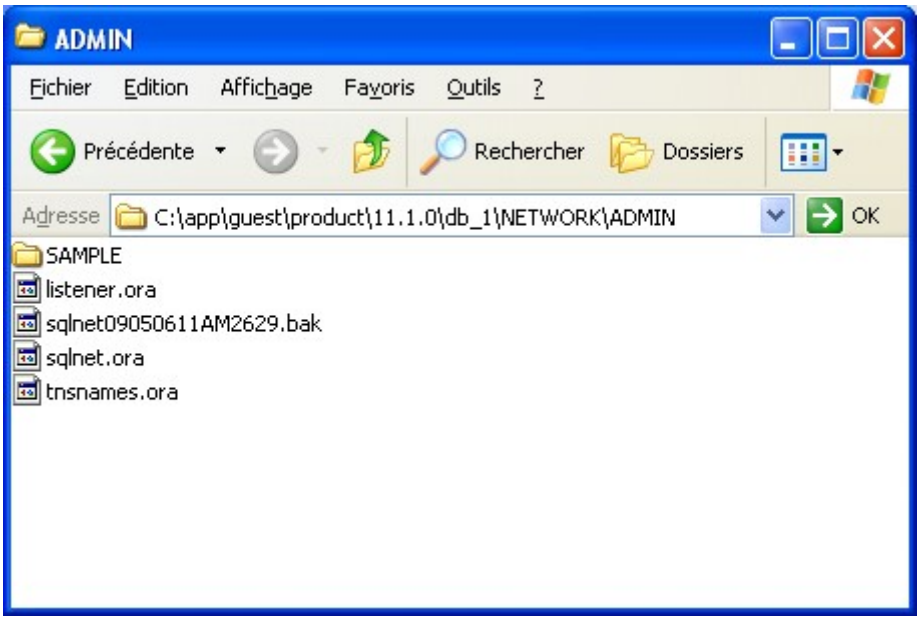
Il est configuré par le fichier listener.ora.

Les instances enregistrent automatiquement leur(s) service(s) auprès du processus d'écoute. Celui-ci est déclaré grâce au paramètre SERVICE_NAMES.

Le listener peut écouter à plusieurs emplacements (protocoles différents, ou variantes du même protocole par exemple 2 ports en TCP/IP), et peut écouter pour plusieurs bases de données et éventuellement pour des versions d'Oracle différentes.

Pour que des postes distants puissent se connecter à la base il faut que le LISTENER soit lancé.

Le LISTENER est utilisé à la première connexion d'un utilisateur. Après la connexion, un arrêt et un redémarrage du LISTENER ne déconnecte pas les utilisateurs déjà connectés.



Le LISTENER s'administre grâce à l'outil LSNRCTL :

```
C:\>LSNRCTL
LSNRCTL for 32-bit Windows: Version 10.2.0.1.0 - Production on 13-AO T -2006 02:
26:41
Copyright © 1991, 2005, Oracle.  All rights reserved.
Bienvenue Ó LSNRCTL, tapez « aide » pour plus d'informations.

LSNRCTL>
```

LSNRCTL permet notamment d'arrêter et de démarrer le LISTENER.

En cas de problème de connexion à partir d'un poste client, vérifier que le LISTENER est bien lancé (ne pas hésiter à le redémarrer).

La configuration côté serveur consiste à configurer le LISTENER, c'est à dire indiquer comment et pour quelles bases il écoute.

Cette configuration peut se faire en modifiant directement le fichier Listener.ora ou en utilisant l'assistant Oracle Net.

Listener.ora

```
# listener.ora Network Configuration File:
C:\oracle\product\10.2.0\db_1\network\admin\listener.ora
# Generated by Oracle configuration tools.
SID_LIST_LISTENER =
(SID_LIST =
(SID_DESC =
(SID_NAME = PLSExtProc)
(ORACLE_HOME = C:\oracle\product\10.2.0\db_1)
(PROGRAM = extproc)
    )
  )

LISTENER =
(DESCRIPTION_LIST =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
(ADDRESS = (PROTOCOL = TCP)(HOST = TELLORA01)(PORT = 1521))
    )
  )
```

Les principales commandes rattachées au LISTENER « lsnrctl » sont les suivantes :

- Start = démarrer le listener
- Stop = arrêter le listener
- Status = obtenir le statut du listener
- Reload = réinitialiser le listener
- Exit = sortir de lsnrctl
- Save_config = crée une sauvegarde du fichier listener.ora puis met à jour le fichier avec les paramètres modifiés à l'aide de lsnrctl.
- Services = affiche les services disponibles ainsi que l'historique de connexion.
- Help = affiche une liste d'options de commande de l'utilitaire lsnrctl.
- Quit = quitter l'utilitaire et revenir à l'invite du système d'exploitation.
- Version = affiche des informations de version sur le listener.
- Show = affiche les valeurs courantes des paramètres.
- Set password mot_de_passe = permet de se connecter au listener via un mot de passe.

Pour vérifier l'exécution du LISTENER, sous le système d'exploitation exécuter la commande :

```
Microsoft Windows XP [version 5.1.2600]
© Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrateur>lsnrctl
LSNRCTL for 32-bit Windows: Version 10.1.0.3.0 - Production on 23-MARS -2005 10:48:46
Copyright © 1991, 2004, Oracle.  All rights reserved.
Welcome to LSNRCTL, type "help" for information.

LSNRCTL> status
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC)))
STATUS of the LISTENER
Alias                     LISTENER
Version                   TNSLSNR for 32-bit Windows: Version 10.1.0.3.0 - Production
Start Date                23-MARS -2005 10:48:26
Uptime                    0 days 0 hr. 0 min. 24 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File   D:\oracle\product\10.1.0\Db_1\network\admin\listener.ora
Listener Log File         D:\oracle\product\10.1.0\Db_1\network\log\listener.log
Listening Endpoints Summary...
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(PIPENAME=\\.\pipe\EXTPROCipc)))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=ATTOUCHE)(PORT=1521)))
Services Summary...
Service "orcl" has 1 instance(s).
Instance "orcl", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully
```

15.2 Configuration coté client

La configuration coté client se fait en modifiant le fichier tnsname.ora, et en lui ajoutant l'accès à la nouvelle instance.

Il se trouve dans le répertoire :

➪ **D:\oracle\product\10.2.0\db_1\NETWORK\ADMIN**

```
# tnsnames.ora Network Configuration File:
C:\oracle\product\10.2.0\db_1\network\admin\tnsnames.ora
# Generated by Oracle configuration tools.
TAHITI =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = TCP)(HOST = TELLORA01)(PORT = 1521))
(CONNECT_DATA =
(SERVER = DEDICATED)
(SERVICE_NAME = tahiti)
    )
  )

EXTPROC_CONNECTION_DATA =
(DESCRIPTION =
(ADDRESS_LIST =
(ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
    )
(CONNECT_DATA =
(SID = PLSExtProc)
(PRESENTATION = RO)
    )
  )
```

Le fichier tnsname.ora ne contient aucune information relative au poste client, il est donc possible d'en créer un et de le diffuser sur d'autres postes.

15.3 Changer de machine automatiquement

Si vous avez installé une base de données de secoure, vous pouvez autoriser le changement de machine de façon automatique. Pour cela, mettre dans le Tnsnames l'adresse de la nouvelle machine ou est monté le Data Guard.

En cas de panne, la bascule se fait sans que l'utilisateur s'en aperçoive, et sans intervention de l'administrateur au niveau du TNSNAMES.ORA, par contre il faudra que l'administrateur passe le Data Guard en base primaire grâce à un Switchover.

```
TAHITI =
(DESCRIPTION =
(LOAD_BALANCE = OFF)          pour le RAC
(FAILOVER = ON)
(ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP)(HOST = poste01)(PORT = 1521))
(ADDRESS = (PROTOCOL = TCP)(HOST = poste02)(PORT = 1521))      pour le Data Guard
    )
(CONNECT_DATA =
(SERVER = DEDICATED)
(SERVICE_NAME = tahiti)
    )
  )
```

15.4 EZCONNECT

En version Oracle 10g vous pouvez vous connecter sans `tnsname.ora` grâce à `ezconnect` :

Il suffit de configurer le ficher `sqlnet.ora`.

⇨ **C:\oracle\product\10.2.0\db_1\NETWORK\ADMIN\sqlnet.ora**

```
# sqlnet.ora Network Configuration File:
C:\oracle\product\10.2.0\db_1\network\admin\sqlnet.ora
# Generated by Oracle configuration tools.
# This file is actually generated by netca. But if customers choose to
# install "Software Only", this file wont exist and without the native
# authentication, they will not be able to connect to the database on
NT.

SQLNET.AUTHENTICATION_SERVICES= (NTS)
NAMES.DIRECTORY_PATH= (TNSNAMES, EZCONNECT)
```

```
connect USER/MotPasse@//[host]:[port]/[service_name]
```

Se connecter en tapant la commande :

```
Sqlplus /nolog
•     connexion traditionnelle via le tnsnames
SQL> connect system/tahiti@tahiti
ConnectÚ.
SQL>


C:\Documents and Settings\ATTOUCHE Clotilde>hostname
TELLORA01

•     dans SQLPLUS
•     connexion via ezconnect
SQL> connect system/tahiti@//tellora01:1521/tahiti
ConnectÚ.
```

15.5 Bases distantes et database Links

Le Database Link est un lien qui permet l'accès à des objets situés dans une base de données distante.

Il est configuré à partir d'Oracle Net et correspond au « service » oracle Net défini dans le listener.

```
create   [ public ] database  link   nom_lien
[ connect  to  nom_user  identified  by  mot_passe ]
using   chaine_de_connection
```

Le nom de l'utilisateur de connexion précisé doit être déclaré dans la base de données distante.

La chaîne de connexion est déclarée dans le « listener.ora » sur une base pour permettre l'accès de celle-ci à la base distante, et dans le fichier « tsname.ora ».

Le fichier listener.ora permet aux postes client d'accéder au serveur Oracle via Oracle Net.

Le fichier tsname.ora permet au serveur Oracle d'accéder à une autre machine via Oracle Net.

Exemples

```
create public database link dli_classe
using  'calan_tcp_LYCE';
drop database link dli_classe ;
create database link dli_clastest.world
connect to admindba identified by oracle
using  'caladan_tcp_LYCE';
```

Le tsname.ora

```
caladan_tcp_LYCE=
(DESCRIPTION=
(ADDRESS=
(PROTOCOL=TCP)
(HOST=10.150.160.106)
(PORT=1521))
(CONNECT_DATA=(SID=LYCE)))
```

Utilisation d'un DB LINK

Exemples d'utilisation d'un `DB_Link` pour faire des insertions sur un site distant.

```
insert into clo01.classe@dli_classe
values (97, 'tyty7', 2);
insert into clo01.classe@dli_classe
values (98, 'tyty8t', 2);
```

16 Sécuriser la base de données

Assurer la sécurité des données est une des tâches principales de l'administrateur.

Cette sécurité est assurée par la mise en œuvre d'une protection des fichiers sensibles de la base de données :

- Fichiers de contrôle
- Fichiers de redo log

16.1 Le fichier de contrôle

Lorsqu'une instance est lancée pour ouvrir une base de données, le fichier de contrôle est le premier fichier ouvert, il permet ensuite à l'instance de localiser et d'ouvrir les autres fichiers de la base de données. Si on perd le fichier de control, la base de données reste à l'état NOMOUNT, et ne pourra pas s'ouvrir.

Le fichier de contrôle est automatiquement mis à jour par Oracle lors de chaque modification de la structure de la base de données (ajout ou déplacement de fichier, …).

Un fichier de contrôle contient les informations suivantes :

- Le nom et l'identifiant de la base de données
- Le nom et l'emplacement des fichiers de données et des fichiers Redo Log
- Le nom des tablespaces
- La date et l'heure de création de la base de données
- Le numéro de séquence du journal courant
- Des informations relatives au point de synchronisation
- L'historique du journal
- Les informations de sauvegarde de l'utilitaire Recovery Manager

Lors de sauvegardes faites avec l'utilitaire Recovery Manager (`RMAN`), certaines vues du dictionnaire et certaines commandes `RMAN` permettent d'interroger le fichier de contrôle pour connaître l'état des sauvegardes réalisées.

Quand `RMAN` sauvegarde une base de données (appelée base de données cible) des informations sont toujours consignées dans le fichier de contrôle de la base cible. Pour limiter la taille des fichiers de contrôle, les anciennes entrées de sauvegarde sont écrasées après un certain nombre de jour. Ce nombre de jours limité est défini par le paramètre `CONTROL_FILE_RECORD_KEEP_TIME`. Par défaut ce nombre de jour est égal à 7, soit une semaine. Il doit être augmenté lorsque `RMAN` utilise le fichier de contrôle de la base cible (15 jours).

Si le fichier de contrôle est trop petit il faudra le recréer en utilisant une trace du fichier de contrôle.

Les fichiers de contrôle doivent être sauvegardés lors de chaque sauvegarde complète ou partielle de la base de données, de plus entre deux sauvegardes « normales », il est conseillé de sauvegarder le fichier de contrôle après toute restructuration importante de la base (ajout/déplacement de fichiers de données ou de redo log).

Pour effectuer une sauvegarde du fichier de contrôle, utiliser l'ordre SQL : ALTER DATABASE

```
ALTER DATABASE BACKUP CONTROLFILE TO
to  [ nouveau_nom | trace ]  [ reuse ]
;
```

- Le paramètre TRACE génère un script dans les fichiers traces utilisateur (diagnostic_dest) .
- Cette trace peut être intéressante dans certaines situations.

Trace du fichier de control
optimum_ora_1396.trc

```
*** 2004-06-14 10:46:04.000
# The following are current System-scope REDO Log Archival related
# parameters and can be included in the database initialization file.
#
# LOG_ARCHIVE_DEST= »
# LOG_ARCHIVE_DUPLEX_DEST= »
#
# LOG_ARCHIVE_FORMAT=ARC%S.%T
# REMOTE_ARCHIVE_ENABLE=TRUE
# LOG_ARCHIVE_MAX_PROCESSES=2
# STANDBY_FILE_MANAGEMENT=MANUAL
# STANDBY_ARCHIVE_DEST=%ORACLE_HOME%\RDBMS
# FAL_CLIENT= »
# FAL_SERVER= »
#
# LOG_ARCHIVE_DEST_1='LOCATION=D:\oracle9\ora92\RDBMS'
# LOG_ARCHIVE_DEST_1='MANDATORY NOREOPEN NODELAY'
# LOG_ARCHIVE_DEST_1='ARCH NOAFFIRM SYNC'
# LOG_ARCHIVE_DEST_1='NOREGISTER NOALTERNATE NODEPENDENCY'
# LOG_ARCHIVE_DEST_1='NOMAX_FAILURE NOQUOTA_SIZE NOQUOTA_USED'
# LOG_ARCHIVE_DEST_STATE_1=ENABLE
#
# Below are two sets of SQL statements, each of which creates a new
# control file and uses it to open the database. The first set opens
# the database with the NORESETLOGS option and should be used only if
# the current versions of all online logs are available. The second
# set opens the database with the RESETLOGS option and should be used
# if online logs are unavailable.
# The appropriate set of statements can be copied from the trace into
# a script file, edited as necessary, and executed when there is a
# need to re-create the control file.
#
#     Set #1. NORESETLOGS case
#
# The following commands will create a new control file and use it
# to open the database.
# Data used by the recovery manager will be lost. Additional logs may
# be required for media recovery of offline data files. Use this
# only if the current version of all online logs are available.
STARTUP NOMOUNT
CREATE CONTROLFILE REUSE DATABASE "OPTIMUM" NORESETLOGS  NOARCHIVELOG
•      SET STANDBY TO MAXIMIZE PERFORMANCE
MAXLOGFILES 32
MAXLOGMEMBERS 5
MAXDATAFILES 128
MAXINSTANCES 16
MAXLOGHISTORY 1815
LOGFILE
GROUP 1 (
'D:\ORACLE9\ORADATA\OPTIMUM\REDO01A.LOG',
'D:\ORACLE9\ORADATA\OPTIMUM\REDO01B.LOG'
) SIZE 10M,
```

```
GROUP 2 (
'D:\ORACLE9\ORADATA\OPTIMUM\REDO02A.LOG',
'D:\ORACLE9\ORADATA\OPTIMUM\REDO02B.LOG'
) SIZE 10M,
GROUP 3 (
'D:\ORACLE9\ORADATA\OPTIMUM\REDO03A.LOG',
'D:\ORACLE9\ORADATA\OPTIMUM\REDO03B.LOG'
) SIZE 10M
•      STANDBY LOGFILE
DATAFILE
'D:\ORACLE9\ORADATA\OPTIMUM\SYSTEM01.DBF',
'D:\ORACLE9\ORADATA\OPTIMUM\UNDOTBS01.DBF',
'D:\ORACLE9\ORADATA\OPTIMUM\ELEVE01.DBF',
'D:\ORACLE9\ORADATA\OPTIMUM\INDX01.DBF',
'D:\ORACLE9\ORADATA\OPTIMUM\OPDEF01.DBF'
CHARACTER SET WE8ISO8859P15
;
# Recovery is required if any of the datafiles are restored backups,
# or if the last shutdown was not normal or immediate.
RECOVER DATABASE
# Database can now be opened normally.
ALTER DATABASE OPEN;
# Commands to add tempfiles to temporary tablespaces.
# Online tempfiles have complete space information.
# Other tempfiles may require adjustment.
ALTER TABLESPACE TEMP ADD TEMPFILE 'D:\ORACLE9\ORADATA\OPTIMUM\TEMP01.DBF'
SIZE 10485760  REUSE AUTOEXTEND OFF;
# End of tempfile additions.
#
```

La modification des paramètres définis dans le fichier de contrôle oblige à la recréation de celui-ci !

16.2 Protection du fichier de contrôle

La première action est le multiplexage du fichier de contrôle. Il s'agit de faire une copie du fichier de contrôle qui sera maintenue en miroir par Oracle.

Si la copie du fichier de contrôle n'est pas jugée cohérente par Oracle, une erreur se produira au redémarrage.

Le fichier de contrôle doit être multiplexé pour des raisons de sécurité :

- Sans fichier de contrôle, la base ne peut pas démarrer
- Au moins deux fichiers de contrôle, si possible sur des disques différents (et dans la pratique, un par disque)

16.2.1 Multiplexer le fichier de contrôle

Faire fonctionner la base avec au moins deux fichiers de contrôle, si possible sur des disques différents (et dans la pratique, un par disque).

Le multiplexage peut être mise en œuvre lors de la création de la base :

- ⇨ Spécifier la liste des fichiers de contrôle souhaités dans le paramètre CONTROL_FILES avant d'exécuter l'ordre SQL CREATE DATABASE

Mais il peut aussi être mise en œuvre ultérieurement :

- Arrêter la base proprement (pas ABORT !)
- Dupliquer un fichier de contrôle existant vers une nouvelle destination
- Mentionner le nouveau fichier de contrôle dans le paramètre CONTROL_FILES
- Redémarrer la base de données

La taille du fichier de contrôle est déterminée par Oracle.

Protéger les fichiers de contrôle est une opération simple. En plus ce sont de petits fichiers, donc on peut en avoir plusieurs copies sans problème.

MODE OPERATOIRE

- Exporter le fichier de paramètres serveur
  `CREATE PFILE FROM SPFILE`
- Modifier le fichier `SPFILE`
  `ALTER SYSTEM ... SCOPE=SPFILE`
- Arrêter la base
  `SHUTDOWN IMMEDIATE`
- Recopier le fichier de contrôle vers le nouvel emplacement
- Ouvrir la base
  `STARTUP`

```
•      Modifier les paramètres CONTROL_FILES dans le fichier de paramètres serveur (base
ouverte)
ALTER SYSTEM SET control_files = 'C:\oracle\oradata\tahiti\controlfile01.ctl',
'D:\tahiti\controlfile02.ctl'
SCOPE = SPFILE ;/
•      Arrêter la base de données
Shutdown immediate ;
•      Dupliquer le fichier de contrôle par une commande du système d'exploitation (base
fermée)
HOST COPY C:\oracle\oradata\tahiti\controlfile01.ctl
D:\tahiti\controlfile02.ctl
•      Redémarrer la base de données
Startup
```

16.3 Vues du dictionnaire de données

Interroger les vues :

- V$CONTROLFILE
- V$PARAMETER

```
Select name
From v$controlfile ;

NAME
/DISK1/control01.con
/DISK2/control02.con

select value
from v$parameter
where name = 'control_file' ;
```

Pour obtenir des informations sur les différentes sections des fichiers de contrôle, interrogez la vue dynamique sur les performances : V$CONTROLFILE_RECORD_SECTION

```
Select type, record_size, records_total, record used
From v$controlfile_record_section
Where type = 'DATAFILE' ;
```

La colonne « record_total » correspond au paramètre MAXDATAFILES de la commande CREATE DATABASE .

16.4 Protection des fichiers de Redo Log

Les fichiers de Redo Log enregistrent toutes les modifications apportées à la base.

Ils sont organisés en groupes composés d'un ou plusieurs membres. Oracle les utilise de manière circulaire, les informations sauvegardées sont donc par défaut périodiquement écrasées. Au minimum la base de données a besoin de 2 groupes.

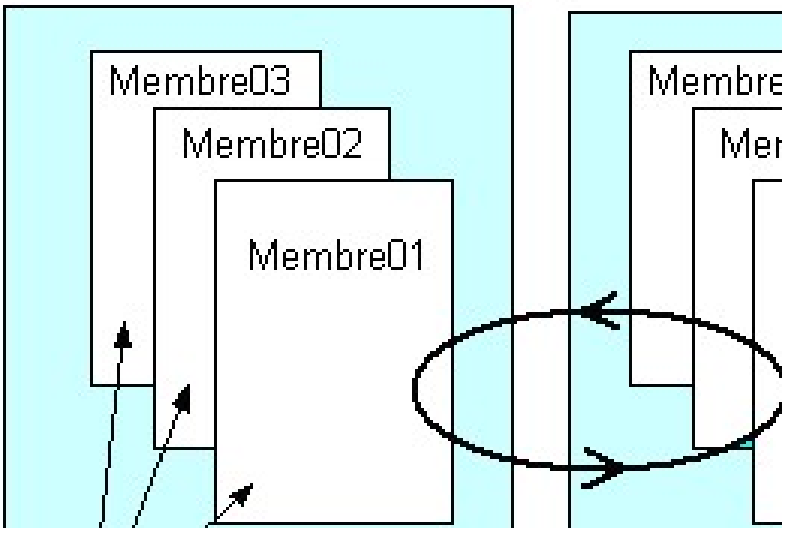
Ils sont utilisés pour la restauration de la base après un arrêt anormal de celle-ci.

Ils peuvent être réappliqués à une sauvegarde de fichier de données, pour rejouer toutes les modifications survenues entre la sauvegarde et un incident ayant endommagé le fichier (c'est la restauration de média).

LGWR écrit en parallèle dans chaque membre d'un même groupe.

Si un groupe de Redo Logs comporte plusieurs membres et qu'un des membres est indisponible, la base de données peut continuer à fonctionner.

Mettre au minimum deux ou trois membres par groupe.



Quelques possibilités sont apparues avec la version 9i :

- Possibilité de forcer l'archivage de façon périodique
  Grâce au paramètre ARCHIVE_LAG_TARGET
- Possibilité de garantir une durée maximale de restauration d'instance en cas d'arrêt anormal
  Grâce au paramètre FAST_START_MTTR_TARGET
- Possibilité d'utiliser plusieurs destinations d'archivage
  Grâce au paramètre LOG_ARCHIVE_DEST_n, n valant de 1 à 10
  Disponible uniquement avec l'édition Entreprise

16.4.1 Dimensionner les fichiers de Redo Log

L'objectif est d'avoir :

- Un basculement de fichier redo log toutes les 20 à 30 minutes environ.
- Aucune attente lors d'un basculement de fichier de Redo Log lié à un checkpoint ou a un archivage non terminé.

Dans le fichier des alertes de l'instance surveillez la fréquence des messages

- Relatif au basculement de fichiers redo log

```
Sat Mar 08 07:05:17 2003
Thread 1 advanced to log sequence 7
```

ET

- Relatif aux attentes lors d'un basculement de fichiers de redo log

```
Sat Mar 08 07:06:48 2003
Thread 1 cannot allocate new log, sequence 8
Checkpoint not complete

Sat Mar 08 07:07:59 2003
Thread 1 cannot allocate new log, sequence 9
All online logs needed archiving
```

Recommendations:

En cas de basculement trop rapide il faut augmenter la taille des fichiers de redo log.

En cas d'attente fréquente lors d'un basculement il faut ajouter un groupe de fichiers de redo log.

Si la base est en Archivelog, il faut prévoir un minimum de 3 groupes de Redo Logs.

Si la base possède un DATA Guard prévoir un quatrième groupe de redo Logs, voir un cinquième groupe.

Si la base est montée en RAC (bases en cluster), prévoir 2 groupes de Redo Logs par nœud.

Les opérations d'administration qui peuvent être effectuées sur les fichiers de redo log sont :

- Ajouter un nouveau groupe de fichiers de redo log permet d'améliorer la disponibilité des fichiers de redo log pour LGWR (en augmentant la durée d'un cycle complet de rotation).
- Multiplexer les membres de Redo Log pour sécuriser la base de données
- Déplacer les fichiers de redo log
- Supprimer un groupe de fichiers de redo log, peut être utilisé dans une opération d'augmentation de la taille des fichiers de redo log (ajout d'un nouveau groupe plus gros puis suppression d'un ancien).
- Supprimer un membre d'un groupe de fichiers de redo log
- Forcer le basculement du groupe courant au suivant peut être effectué avant les sauvegardes des archives de Redo Logs.

16.4.2 Multiplexer les fichiers de Redo Log

Le multiplexage des fichiers de redo log peut être mis en œuvre lors de la création de la base de données. Il consiste à dupliquer les membres de Redo Log d'un même groupe.

Il est possible de spécifier plusieurs membres pour chaque groupe listé dans la clause LOGFILE de l'ordre SQL CREATE DATABASE

Il peut aussi être mis en œuvre après création de la base de données, en utilisant de l'ordre SQL ALTER DATABASE.

```
ALTER DATABASE
ADD LOGFILE MEMBER 'nom_fichier' [,…] TO GROUP numéro
;
```

```
ALTER DATABASE
ADD LOGFILE MEMBER 'f:\oracle\oradata\HERMES\redo01.log' TO GROUP 1;
```

Remarques :

- La taille du fichier n'a pas besoin d'être spécifiée ; le nouveau fichier a forcément la même taille que les autres membres du groupe
- Normalement, tous les groupes doivent avoir le même nombre de membres

16.4.3 Ajouter un groupe de Redo Log

Pour ajouter un nouveau groupe à la base de données utilisez l'ordre SQL ALTER DATABASE :

```
ALTER DATABASE
ADD LOGFILE [GROUP numéro] spécification_fichier_redo [,…]
;
```

- spécification_fichier_redo

```
('nom_fichier' [,…]) [ SIZE valeur [K|M] ] [REUSE]
```

Sauf opération d'augmentation de la taille des fichiers de redo log, le nouveau groupe a normalement la même taille que les anciens.

Exemple

```
ALTER DATABASE
ADD LOGFILE
GROUP 4 ('d:\oracle\oradata\HERMES\redo04.log',
'e:\oracle\oradata\HERMES\redo04.log') SIZE 10240K;
```

ATTENTION

On ne peut pas modifier la taille des fichiers de Redo Log, il faut ajouter des groupes ayant une taille souhaitée et supprimer les anciens groupes.

Supprimer un groupe n'est pas possible si c'est le groupe courant. Dans ce cas, la commande permettant de forcer un SWITCH de fichier de Redo Log peut alors être utilisée pour éviter d'attendre. Le SWITCH permettra de changer de groupe.

16.4.4 Déplacer les fichiers de Redo Log

Lorsque l'on déplace un fichier de Redo Log il faut suivre le mode opératoire ci-dessous et veiller à recopier le fichier dans le nouvel emplacement avant d'effectuer la commande RENAME FILE.

Oracle ne s'est pas créer de fichiers ni déplacer des fichiers ou encore créer des répertoires !

Exemple

```
Connect / as sysdba
Shutdown immediate
Startup mount
C:\> host copy 'e:\oracle\oradata\tahiti\redo04.log'
'g:\oracle\oradata\tahiti\redo04.log'
Alter database
Rename file 'e:\oracle\oradata\tahiti\redo04.log'
to  'g:\oracle\oradata\tahiti\redo04.log' ;
Alter database open ;
Suppression de l'ancien fichier à l'aide d'une commande du système d'exploitation.
```

MODE OPERATOIRE

- Arrêter la base de données
  `SHUTDOWN IMMEDIATE`
- Déplacer les fichier de Redo Log vers le nouvel emplacement
  `COPIER .. +.. COLLER`
- Monter la base
  `STARTUP MOUNT`
- Indiquer à Oracle le nouvel emplacement
  `ALTER DATABASE RENAME FILE`
- Ouvrir la base
  `ALTER DATABASE OPEN`

16.4.5 Supprimer un groupe de fichiers redo log

La base doit avoir au moins 3 groupes de fichiers de redo log pour pouvoir en supprimer un (il doit en rester au moins 2).

Le groupe courant (celui dans lequel LGWR est en train d'écrire) ne peut pas être supprimé.

En mode ARCHIVELOG, un groupe pas encore archivé ne peut pas être supprimé.

Les fichiers concernés ne sont pas physiquement supprimés par Oracle, il faudra le faire après en utilisant une commande du système d'exploitation.

Utiliser l'ordre SQL ALTER DATABASE :

```
ALTER DATABASE
DROP LOGFILE GROUP numéro
;
```

Exemple

```
ALTER DATABASE
DROP LOGFILE      GROUP 4;
```

16.4.6 Supprimer un membre d'un groupe de redo log

Le groupe concerné doit avoir au moins 2 membres pour pouvoir en supprimer un (il doit en rester au moins un).

Un membre du groupe courant (celui dans lequel LGWR est en train d'écrire) ne peut pas être supprimé.

En mode ARCHIVELOG, un membre d'un groupe pas encore archivé ne peut pas être supprimé.

Les fichiers concernés ne sont pas physiquement supprimés par Oracle.

Pour supprimer tous les membres d'un groupe il faut supprimer le groupe.

Si un membre est corrompu, Oracle lui donne un statut « invalide » dans le dictionnaire de données mais ne le signale pas par un message affiché à l'écran.

Pour détecter qu'un membre est invalide, il faut consulter le fichier des Alertes.

Si un membre d'un groupe est perdu, il faut le supprimer pour mettre à jour le dictionnaire de données.

Utiliser l'ordre SQL ALTER DATABASE :

```
ALTER DATABASE
DROP LOGFILE MEMBER 'nom_fichier' [,…]
;
```

Exemple

```
ALTER DATABASE
DROP LOGFILE      MEMBER 'e:\oracle\oradata\HERMES\redo01.log';
```

16.4.7 Forcer le basculement du groupe courant

Le basculement d'un groupe courant peut être utilisé lorsque l'on a besoin d'effectuer une suppression de groupe ou lorsque l'on veut générer une archive avant d'effectuer une sauvegarde (la base de données doit alors être en ARCHIVELOG).

Utiliser l'ordre SQL ALTER SYSTEM :

```
ALTER SYSTEM SWITCH LOGFILE
;
```

Si les SWITCH sont trop fréquents, ce n'est pas bon pour les performances.

La vue V$LOG_HISTORY peut être utilisée pour analyser la fréquence de SWITCH des fichiers de Redo Log (colonne FIRST_TIME).

Le basculement manuel provoque les mêmes évènements qu'un basculement automatique

- Checkpoint : point de reprise
- Archivage (si l'archivage est activé)

Le paramètre FAST_START_MTTR_TARGET peut être utilisé pour obliger la base à effectuer des points de reprise régulièrement. En effet ce paramètre indique un nombre maximum de secondes pour le redémarrage de l'instance après un arrêt anormal.

Le positionnement de ce paramètre permet à l'instance d'ajuster des points de reprises de manière à pouvoir rejouer l'activité perdue sur la base en respectant les valeurs du paramètre. Attention à ne pas avoir de points de reprises trop fréquents ce qui dégraderait les performances.

16.4.8 Trouver des informations sur les fichiers Redo Log

Plusieurs vues du dictionnaire permettent d'obtenir des informations sur les fichiers de redo log :

- V$LOG : informations sur les groupes
- V$LOGFILE : informations sur les membres
- V$LOG_HISTORY : informations sur l'historique des fichiers de redo log
- V$INSTANCE__RECOVERY : informations sur les temps estimés de restauration d'instance.

V$LOG	
GROUP#	Numéro du groupe
SEQUENCE#	Numéro de séquence du groupe (s'incrémente à chaque SWITCH).
MEMBERS	Nombre de membres
ARCHIVED	Groupe archivé (yes, no)
STATUS	Statut du groupe Unused : groupe jamais utilisé Current : groupe courant Active : groupe encore nécessaire en cas de restauration d'instance (checkpoint non terminé) Inactive : groupe plus nécessaire pour une restauration d'instance (checkpoint terminé)
FIRST_CHANGE#	Plus petit numéro SCN (numéro de transaction) écrit dans le groupe
FIRST_TIME	Date et heure du plus petit numéro SCN

V$LOG_ FILE	
GROUP#	Numéro du groupe
STATUS	Statut du groupe Invalide : fichier inaccessible Stale : fichier incomplet (statut des nouveaux membres) Deleted : fichier supprimé Colonne vide : fichier utilisé
MEMBER	Nom complet du fichier membre

$LOG_ HISTORY	
SEQUENCE#	Numéro de séquence du groupe archivé
FIRST_CHANGE#	Plus petit numéro SCN (numéro de transaction) écrit dans le groupe archivé
NEXT_CHANGE#	Plus grand numéro SCN (numéro de transaction) écrit dans le groupe archivé
FIRST_TIME	Date et heure du plus petit numéro SCN

$INSTANCE_ RECOVERY	
TARGET_MTTR	Objectif réel de durée de récupération de l'instance recalculée par Oracle.
ESTIMATED_MTTR	Durée de récupération actuelle (tient compte de l'activité de la base)
OPTIMAL_LOGFILE_SIZE	Taille optimale de groupes de Redo Logs en Mo

17 Gestion du stockage

Dans ce chapitre, nous allons faire le lien entre les structures logiques d'Oracle et ces fichiers.

Nous allons étudier comment Oracle stocke les données des tables et des index dans les fichiers et comment ces données sont gérées physiquement sur le disque.

17.1 Notion de tablespace

Un tablespace est une unité logique de stockage composée d'un ou de plusieurs fichiers physiques.

Les fichiers de données sont découpés en blocs d'une taille définie à la création de la base de données (2 ko, 4 ko, 8 ko, etc …). La taille des blocks Oracle correspondent à un multiple du block géré par le système d'exploitation.

La taille d'un bloc Oracle correspond au paramètre `DB_BLOCK_SIZE`.

L'espace occupé par un objet dans un tablespace est désigné par le terme de segment.

Il y a 4 types de segments gérés dans une base Oracle :

- Les segments de table = espace occupé par les tables
- Les segments d'index = espace occupé par les index
- Les segments d'annulation = espace temporaire utilisé pour stocker les informations permettant d'annuler une transaction (ROLLBACK).
- Les segments temporaires = espace temporaire utilisé lors d'un tri dans une requête.

La règle est d'utiliser plusieurs tablespaces afin de séparer les différents objets de la base et d'assurer de meilleures performances à la base de données. Cela permettra également d'offrir une plus grande souplesse dans les tâches d'administration.

Chaque type de segment est stocké dans un tablespace qui lui est propre. Ainsi on pourra garantir un minimum de performances.

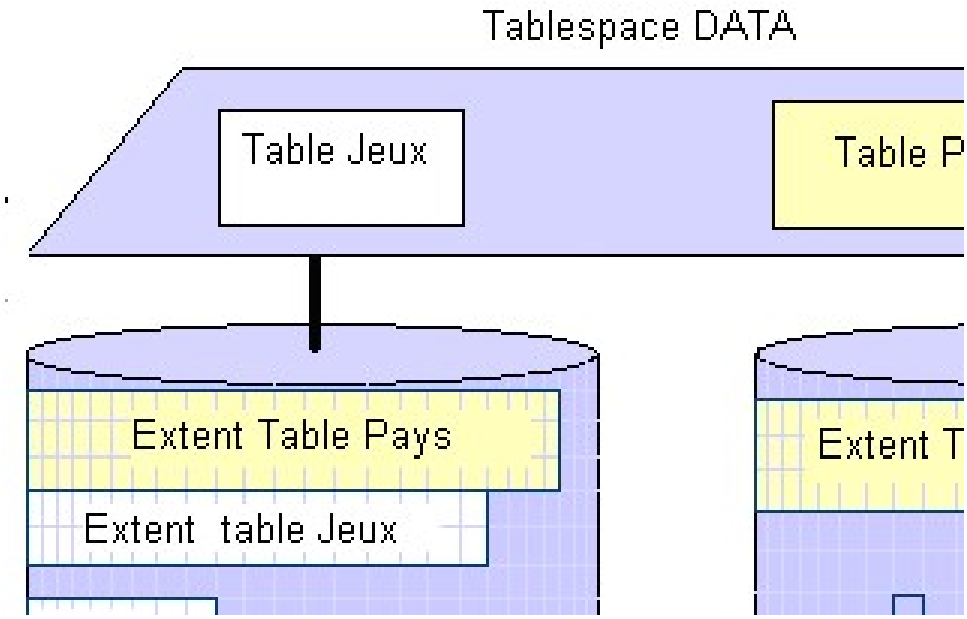
Ainsi on mettra les tables et les index dans des tablespaces dédiés aux tables ou aux index, les segments temporaires dans des tablespaces temporaires et les segments d'annulation dans le tablespace UNDO.

L'espace utilisé par les objets sur le disque est appelé segment (comme les tables par exemple).

A l'intérieur d'un tablespace, le stockage est organisé en segments comportant un ou plusieurs extents (ensemble de bloc Oracle contigus de taille `DB_BLOCK_SIZE` dans un fichier de données).

Lorsqu'un segment est créé dans un tablespace, Oracle lui alloue un premier extent dans un des fichiers du tablespace. Lorsque ce premier extent est plein, Oracle alloue un deuxième extent au segment et ainsi de suite.

Les extents complémentaires alloués au segment sont dans le tablespace d'origine du segment, mais pas forcément côte à côte ni forcément dans le même fichier. Lorsqu'un segment est supprimé, les extents qu'il occupe sont libérés et rendus disponibles pour d'autres segments.



Des créations/suppressions fréquentes de données appartenant à un segment ou de segments dans un tablespace conduisent souvent à une fragmentation de l'espace disponible dans ce tablespace.

La clause de stockage permet de gérer la taille du segment à sa création, puis lors de son évolution.

La clause de stockage est créée au moment de la création de l'objet puis peut être modifiée.

Syntaxe :

```
Storage (INITIAL     valeur
         NEXT        valeur
MINEXTENTS  valeur
MAXEXTENTS  valeur
PCTINCREASE valeur
         )
```

- INITIAL taille du premier extent créé à la création de l'objet
- NEXT taille des extents suivants
- MINEXTENTS nombre minimum d'extents créés à la création de l'objet
- MAXEXTENTS nombre maximum d'extents créés pendant la vie de l'objet
- PCTINCREASE pourcentage d'augmentation de la taille d'un extent par rapport au précédent.

La clause de stockage peut être modifiée après création de l'objet !

Exemple

```
-- ============================================================
•      Table : EMPLOYE
-- ============================================================
create table EMPLOYE
(
    ID_EMP        INTEGER                not null,
    NOM           VARCHAR2(30)           not null,
    SALAIRE       NUMBER(4)              not null,
    EMPLOI        VARCHAR2(20)           null    ,
    EMP_ID_EMP    INTEGER                null    ,
constraint PK_EMPLOYE primary key (ID_EMP)
using index
tablespace INDX
)
Tablespace tbs_local_uniform
Storage (initial 400K
next 100K
minextents 2
pctincrease 0
         )
/
```

Le tablespace est le plus petit niveau de sauvegarde & restauration.
Le `STRIPING` consiste à avoir 1 tablespace réparti sur plusieurs disques (le tablespace est alors composé de plusieurs *datafiles*).

Il est conseillé de répartir les différents tablespaces sur des disques différents afin d'éviter les contentions sur les entrées /sorties.

Directives

- Ne rien mettre dans les tablespace `SYSTEM et SYSAUX`.
- En plus des tablespaces UNDO, TEMP, et USERS, (penser au suivi des tables et aux sauvegardes car le tablespace est l'unité de stockage) créer :
  - Plusieurs tablespaces pour les tables
  - Plusieurs tablespaces pour les index
- En cas d'utilisation de `RAID`, préférez le `RAID 1` (*Miroring*).
- En cas d'utilisation de `RAID 5` , Beaucoup moins bon pour Oracle, vous pouvez mettre les fichiers de Redo Log sur des disques sans RAID pour éviter les contentions.

17.2 Organisation du stockage dans un tablespace

A partir de la 9i, les tablespaces sont gérés localement et plus par le dictionnaire de données. La gestion des extents est gérée par Oracle et la clause de stockage n'est plus spécifiée à la création du tablespace.

Un des objectifs des tablespaces gérés localement est d'optimiser l'utilisation de l'espace dans les tablespaces et d'éviter le phénomène de fragmentation de l'espace disponible.

Oracle tente d'allouer côte à côte des extents de même taille. Oracle optimise ainsi le stockage et les performances.

```
EXTENT MANAGEMENT
DICTIONARY | LOCAL [ AUTOALLOCATE | UNIFORM [ SIZE valeur [K|M] ] ]
[ SEGMENT SPACE MANAGEMENT AUTO ]
```

Attention la taille minimum acceptée par Oracle en `UNIFORM SIZE` est de 5 blocs.

- tablespace géré localement avec des extents uniformes

```
CREATE TABLESPACE tbs_uniform
DATAFILE 'C:\app\guest\oradata\BORA\tbs_uniform.dbf' SIZE 10M
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
```

- tablespace géré localement avec des extents gérés par Oracle

```
CREATE TABLESPACE tbs_auto
DATAFILE 'C:\app\guest\oradata\BORA\tbs_auto.dbf' SIZE 10M
EXTENT MANAGEMENT LOCAL AUTOALLOCATE
SEGMENT SPACE MANAGEMENT AUTO;
```

Les informations relatives à la gestion des espaces (extent libre/alloué) sont enregistrées dans un bitmap, dans l'en-tête de chaque fichier de données :

- Chaque bit du bitmap correspond à un extent
- Vaut 0 ou 1 selon que l'extent est libre ou alloué

La Bitmap permet d'identifier les extents adjacents libres.

Gestion automatique des extents par Oracle :

- 3 blocs pour l'en-tête de fichier plus la taille d'un extent
- Taille de l'extent :
  - valeur explicite ou par défaut de la clause `SIZE` pour un tablespace `UNIFORM`
  - 64 Ko, 1M, 8M pour un tablespace `AUTOALLOCATE`

La taille d'extents initialement choisis pour un segment dépend de la taille du segment spécifiée lors de sa création.

- 64Ko pour les segments de moins de 1Mo
- 1Mo pour les segments de moins de 64Mo
- 8Mo pour les segments de moins de 1024Mo

Exemple

Si la table suivante est créée dans un tablespace géré localement avec des extents UNIFORMES de 128 Ko

```
-- ============================================================
•      Table : EMPLOYE
-- ============================================================
create table EMPLOYE
(
    ID_EMP       INTEGER                not null,
    NOM          VARCHAR2(30)           not null,
    SALAIRE      NUMBER(4)              not null,
    EMPLOI       VARCHAR2(20)           null    ,
    EMP_ID_EMP   INTEGER                null    ,
constraint PK_EMPLOYE primary key (ID_EMP)
using index
tablespace INDX
)
Tablespace tbs_local_uniform
Storage (initial 400K
next 100K
minextents 2
pctincrease 0
         )
/
```

Oracle alloue (400 + 100) /128 = 3,5 arrondi à l'entier supérieur = 4 extents de 128 Ko pour la table.

Si la même table est créée dans un tablespace géré localement avec une gestion automatique des extents Oracle alloue (400 + 100)/64 = 7,8 arrondi à l'entier supérieur = 8 extents de 64 Ko pour la table.

En version 10.2 et en version 11g, la clause Segment Space Management est positionnée à AUTO par défaut.
Elle indique une gestion des segments dont l'en-tête est géré par Bitmap.
-- La Free List disparaît !

17.3 Notion de BIGFILE ou de SMALLFILE

A partir de la version 10g, Oracle introduit la notion de BIGFILE et de SMALLFILE.

La valeur SMALLFILE détermine une gestion de petits fichiers (pouvant contenir au maximum 1022 fichiers d'un taille de 2^{22} blocks = 32Go).alors que la valeur BIGFILE détermine la gestion d'un seul fichier de taille très importante (plus de 4 milliards de blocs) et sont préconisés pour l'utilisation en ASM.

Si aucun type de fichier n'est précisé au moment de la création d'un tablespace, la valeur de la propriété « DEFAULT_TBS_TYPE » est prise par défaut.

```
SQL> -- Afficher les tablespaces définis par défaut
SQL> set pagesize 200
SQL> set linesize 200

SQL> col property_value format A30
SQL> col property_name format A30


SQL> select property_value, property_name
2  from database_properties
3  where property_name like 'DEFAULT%'
4  /

PROPERTY_VALUE              PROPERTY_NAME
TEMP                    DEFAULT_TEMP_TABLESPACE
USERS                   DEFAULT_PERMANENT_TABLESPACE
SMALLFILE               DEFAULT_TBS_TYPE
```

La modification de cette option se fait par la commande :

```
ALTER DATABASE SET DEFAULT { SMALLLFILE | BIGFILE } TABLESPACE
/
```

La propriété `BIGFILE` ou SMALLFILE peut être spécifiée à la création du tablespace :

```
CREATE BIGFILE TABLESPACE TBS_GROS
DATAFILE ':\app\oracle\oradata\BORA\DATA_GROS.DBF'
SIZE  20G
/
```

Les recommandations d'oracle vis-à-vis du stockage sont *Strip and Mirror Everything* (SAME).
Soit le RAID 1 = Raid 0 (STRIP) pour les performances + Raid 1 (MIROR) pour la sécurité.

18 Tablespaces permanents

Les tablespaces permanents sont utilisés pour stocker des objets permanents tels que les tables ou les index. C'est l'unité logique de stockage d'une base Oracle, et donc le plus petit élément de sauvegarde.

18.1 Créer un tablespace permanent

L'ordre SQL `CREATE TABLESPACE` permet de créer un tablespace :

```
CREATE [ BIGFILE | SMALLFILE ] TABLESPACE Ts_nom
DATAFILE  'nom_fichier' [ SIZE valeur [K|M|G|T] ] [REUSE]
AUTOEXTEND { OFF | ON [ NEXT valeur [K|M|G|T] ]
[ MAXSIZE { UNLIMITED | valeur [K|M|G|T] } ] }
EXTENT MANAGEMENT
LOCAL { AUTOALLOCATE | UNIFORM [ SIZE valeur [K|M|G|T] ] }
SEGMENT SPACE MANAGEMENT
{ MANUAL | AUTO }
[ DEFAULT [COMPRESS [ FOR {ALL | DIRECT_LOAD} OPERATIONS] | NOCOMPRESS] storage ]
[ BLOCKSIZE  valeur (K) ]
[ LOGGING | NOLOGGING ]
[ FORCE LOGGING ]
[ FLASHBACK { ON | OFF } ]
[ ONLINE | OFFLINE ]
;
```

- **DATAFILE** = spécification (emplacement, taille, …) d'un (ou plusieurs) fichier(s) de données pour le tablespace, la taille peut être spécifiée en octets (pas de symbole), en Ko (symbole K) ou en Mo (symbole M) ou en Gigas (G) ou en terra pour les tablespace de type `BIGFILE` uniquement.
- **AUTOEXTEND** = indique si le fichier de données peut (`ON`) ou non (`OFF`) grandir lorsque tout l'espace initialement alloué est occupé.
- **NEXT** = espace minimum alloué lors d'une extension
- **MAXSIZE** = taille maximum du fichier, éventuellement non limitée (`UNLIMITED`)
- **EXTENT MANAGEMENT** = permet de définir le mode de gestion des extents à l'intérieur du tablespace.
- **SEGMENT SPACE MANAGEMENT** = permet de définir le mode de gestion de l'espace libre des segments à l'intérieur du tablespace.
- **DEFAULT [COMPRESSE | NOCOMPRESS]** = permet de définir un mode de compression automatique des segments dans le tablespace, permettant de réduire l'espace disque. Attention, cette compression se fait au détriment des temps des performances.
- **LOGGING | NOLOGGING** = enregistre les segments dans les Redo Log si `LOGGING` (par défaut) est défini.
- **FORCE LOGGING** = oblige l'enregistrement des segments dans les Redo Log même si la clause `NOLOGGING` est définie.
- **FLASHBACK { ON | OFF }** = indique si le tablespace participe aux opérations de flashback (`ON`).
- **ONLINE | OFFLINE** = indique si le tablespace est accessible (`ON`) ou non (`OFF`).
- **COMPRESS | NOCOMPRESS =** compression des données contenues dans le tablespace et possibilité de créer une clause de stockage par défaut pour les segments créés sans clause de stockage.

D'une manière générale toutes les syntaxes peuvent être spécifiées en octets (pas de symbole), en kilo octets (K), en méga octets (M), en giga octet (G) ou encore Terra octet (T).

18.2 Modifier un tablespace permanent

Un certain nombre de manipulations sont effectuées durant la vie de ces tablespaces :

- Allouer de l'espace supplémentaire à un tablespace
- Passer un tablespace `OFFLINE/ONLINE`
- Déplacer le(s) fichier(s) de données
- Passer un tablespace `READ ONLY/READ WRITE`
- Renommer un tablespace

Ces opérations s'effectuent avec la commande SQL :

- `ALTER TABLESPACE ou ALTER DATABASE.`

18.2.1 Agrandir un tablespace

Ajouter un fichier de données

Il est possible d'agrandir un tablespace par l'ajout d'un fichier de données.

```
ALTER TABLESPACE nom
ADD DATAFILE spécification_fichier_data [,…]
;
```

```
ALTER TABLESPACE data
ADD DATAFILE :\app\guest\oradata\BORA\data02.dbf' SIZE 100M
AUTOEXTEND ON NEXT 100M MAXSIZE 500M;
```

Modifier la taille du fichier rattaché au tablespace

Modifier la taille d'un fichier de données, à la hausse ou à la baisse dans la limite du dernier extent occupé dans le fichier de données.

```
ALTER DATABASE
DATAFILE 'nom_complet' [,…] RESIZE valeur [K|M]
;
```

L'option `RESIZE` peut être utilisée à la hausse mais aussi à la baisse en fonction de l'emplacement du dernier objet créé ou modifié, dans le tablespace, visible dans la vue `DBA_EXTENTS`, sinon il y a affichage d'un message d'erreur.

Cette option peut être lancée pendant que des utilisateurs travaillent sur la base. Elle a pour effet de modifier la taille du fichier immédiatement.

```
ALTER DATABASE
DATAFILE 'C:\app\oracle\oradata\BORA\data02.dbf' RESIZE 800M;
```

Modifier la clause AUTO EXTEND

L'option AUTOEXTEND ON a pour effet de permettre au fichier de s'étendre lorsqu'il est plein et qu'un nouvel objet est créé.

La clause AUTOEXTEND OFF annule une clause AUTOEXTEND ON.

```
ALTER DATABASE
DATAFILE 'nom_complet' [,…]
AUTOEXTEND [ON|OFF]
;
```

```
•      désactivation de la clause autoextend
ALTER DATABASE
DATAFILE 'C:\app\oracle\oradata\BORA\data01.dbf'
AUTOEXTEND OFF;

•      activation (ou modification) de la clause autoextend
ALTER DATABASE
DATAFILE 'C:\app\oracle\oradata\BORA\data02.dbf'
AUTOEXTEND ON NEXT 200M MAXSIZE 800M;
```

18.2.2 Passer un tablespace OFFLINE ou ONLINE :

Les clauses ONLINE ou OFFLINE permettent d'activer ou de désactiver un tablespace. Les objets contenus dans le tablespace ne sont plus accessibles par les utilisateurs si la clause OFFLINE est utilisée.

```
ALTER TABLESPACE 'nom_complet'
ONLINE | OFFLINE
;
```

```
•      désactivation
ALTER TABLESPACE data OFFLINE;
•      activation
ALTER TABLESPACE data ONLINE;
```

18.2.3 Passer un tablespace READ ONLY ou READ WRITE :

Le tablespace en READ ONLY ne permet plus l'écriture dans les objets qu'il contient.

```
ALTER TABLESPACE 'nom_complet'
READ ONLY | READ WRITE
;
```

```
•      désactivation
ALTER TABLESPACE data READ ONLY;
•      activation
ALTER TABLESPACE data READ WRITE;
```

18.2.4 Déplacer un fichier de données

Cette action est possible par un `ALTER TABLESPACE` ou un `ALTER DATABASE RENAME FILE`.

Le `ALTER DATABASE` est particulièrement utile pour déplacer le tablespace `SYSTEM` qui ne peut pas être mis `OFFLINE`.

Le tablespace concerné doit être `OFFLINE` ou la base en état `MOUNT` dans le cas du `ALTER DATABASE`.

Les commandes ne manipulent pas physiquement le fichier ;

Elles se contentent de mettre à jour le dictionnaire Oracle et les fichiers de contrôle, de plus le fichier doit être renommé + copié + déplacé avant de passer la commande.

```
ALTER TABLESPACE nom
RENAME DATAFILE 'ancien_nom_complet'
TO 'nouveau_nom_complet'
;
```

Le mode opératoire ci-dessous doit être appliqué pour effectuer ce genre d'opération.

MODE OPERATOIRE

- Mettre le tablespace `OFFLINE`

```
ALTER TABLESPACE nom_ts OFFLINE ;
```

- Déplacer les fichiers vers le nouvel emplacement

```
COPIER .. +.. COLLER
```

- Indiquer à Oracle le nouvel emplacement

```
ALTER TABLESPACE RENAME FILE ancien_nom
   TO nouveau_nom ;
```

- Mettre le tablespace `ONLINE`

```
ALTER TABLESPACE nom_ts ONLINE ;
```

Déplacer un fichier de données

```
•      Passer le tablespace OFFLINE
Sql> alter tablespace data OFFLINE ;
•      Déplacer le fichier du tablespace par une commande du système d'exploitation (copy)
•      Exécuter la commande : alter tablespace …
Sql> alter tablespace data
Sql>    rename datafile 'C:\app\guest\oradata\BORA\data01.dbf'
Sql>                to  'e:\oracle\oradata\BORA\data01.dbf' ;
```

```
•      Remettre le tablespace ONLINE
Sql> alter tablespace data01 ONLINE;
```

Le mode opératoire ci-dessous doit être appliqué pour effectuer ce genre d'opération sur le tablespace SYSTEM qui contient le dictionnaire de données et ne peut pas être mis OFFLINE.

MODE OPERATOIRE

- Arrêter la base
  `SHUTDOWN IMMEDIATE`
- Déplacer les fichiers vers le nouvel emplacement
  `COPIER .. +.. COLLER`
- Démarrer la base à l'état `MOUNT`
  `STARTUP MOUNT`
- Indiquer à Oracle le nouvel emplacement

```
ALTER DATABASE RENAME FILE ancien_nom
                        TO nouveau_nom
```

- Ouvrir la base

```
ALTER DATABASE OPEN
```

Déplacer le tablespace système

```
Le mettre à l'état MOUNT
Sql> Connect / as sysdba
Sql> Shutdown immediate
Sql> Startup mount

Déplacer le fichier du tablespace système par une commande du système d'exploitation
(copy)
Exécuter la commande : alter database rename file …
Sql> alter database
Sql> rename file 'C:\app\oracle\oradata\BORA\systeme01.dbf'
Sql>         to  'D:\app\oracle\oradata\BORA\systeme01.dbf' ;

Ouvrir la base
Sql> alter database open;
```

18.2.5 Renommer un tablespace

A partir de la version 10g, il est possible de changer le nom d'un tablespace en utilisant une requête SQL ou `OEM`. Quand vous renommez un tablespace, Oracle met à jour toutes les références au nom du tablespace dans le dictionnaire de données, le fichier de contrôle et les en-têtes de fichiers du tablespace concerné.

L'identifiant du tablespace n'est pas changé. Si le tablespace s'avère être le tablespace par défaut d'un utilisateur, alors le tablespace renommé apparaît toujours comme étant le tablespace par défaut de l'utilisateur.

Les en-têtes de fichiers ne peuvent être changés que si le tablespace est mis dans un mode `READ WRITE` (quand c'est possible).

Vous pouvez renommer à la fois des tablespaces permanents et temporaires.

```
ALTER TABLESPACE 'ancien_nom' RENAME TO 'nouveau_nom' ;
```

```
SQL> -- Renommer le tablespace DATABIS en DATA
SQL> ALTER TABLESPACE databis RENAME TO data
2  /

Tablespace altered.
```

Si vous recréez le fichier de contrôle, les « *record*s » de tablespace dans le nouveau fichier de contrôle sont construits à partir des informations des fichiers associés au tablespace.

Si vous utilisez des sauvegardes pour la restauration, les « *records* » de tablespace construits pourraient refléter les anciens noms du tablespace car la sauvegarde a été effectuée avant que le tablespace soit renommé.

La restauration en utilisant des sauvegardes de fichiers de données contenant des anciens noms de tablespace n'est pas un problème. Si un fichier de sauvegardes dont les en-têtes contiennent l'ancien nom du tablespace est restauré après avoir été renommé, l'en-tête de fichier possède le nouveau nom du tablespace après restauration.

Il n'est pas possible de renommer les tablespaces `SYSTEM`, `SYSAUX` ou `UNDO`. Le changement affecterait à la fois la mémoire et le `SPFILE`

Si un seul datafile du tablespace à renommer est `OFFLINE` ou si le tablespace est `OFFLINE`, alors le tablespace ne peut pas être renommé.

Avant la version 10g, si vous vouliez réorganiser et migrer un Tablespace « `DATA` » géré par le dictionnaire en un tablespace géré localement avec des extents uniformes,

vous deviez appliquer le mode opératoire ci-dessous :

- Créer un autre ts « `DATABIS` »,
- Copier tous les objets de « `DATA` » vers « `DATABIS` »
- Supprimer « `DATA` »
- Recréer « `DATA` » avec une gestion locale des extents,
- Copier tous les objets de « `DATABIS` » vers « `DATA` »,
- Supprimer « `DATABIS` ».

En utilisant la fonctionnalité de renommage d'un tablespace, vous pouvez simplifier ce mode opératoire :

- Créer un ts « DATABIS » géré localement
- Copier tous les objets de « DATA » vers « DATABIS »
- Supprimer « DATA »
- Renommer « DATABIS » en « DATA »

Le tablespace ne peut être renommé que si vous avez déjà supprimé le tablespace portant l'ancien nom.

Comme Oracle ne renomme pas les fichiers d'un tablespace, il faut faire attention aux noms des fichiers associés au tablespace. Faites attention à la méthode qui consistait à nommer les fichiers d'un tablespace du même nom suivi de numéros car après « renommage », ces noms deviennent obsolètes.

Cette fonctionnalité est à utiliser en cas de migration de oracle 8i vers une version oracle 10g.

18.2.6 Supprimer un tablespace

L'ordre SQL DROP TABLESPACE permet de supprimer un tablespace.

```
DROP TABLESPACE nom [ INCLUDING CONTENTS [ AND DATAFILES ]
[ CASCADE CONSTRAINTS ] ]
;
```

Si le tablespace contient des segments, la clause INCLUDING CONTENTS est nécessaire.

La clause CASCADE CONSTRAINTS permet en plus d'ignorer les contraintes d'intégrité référentielle définies sur des tables hors du tablespace et qui référencent les clés primaires des tables à l'intérieur du tablespace.

A partir de la 9i, la clause AND DATAFILES permet de supprimer les fichiers physiquement.

Un message est écrit dans le fichier des alertes de l'instance pour chaque fichier supprimé.

```
Alter tablespace data OFFLINE ;
DROP TABLESPACE data INCLUDING CONTENTS and DATAFILES;
```

Ne pas oublier de mettre le tablespace OFFLINE avant de le supprimer.

18.2.7 Créer un tablespace avec une taille de bloc non standard

La taille de bloc standard d'un tablespace correspond à la taille définie par le paramètre `DB_BLOCK_SIZE` lors de la création de la base de données. Cette taille ne peut plus être modifiée par la suite (pendant toute la vie de la base de données).

- ⇨ elle est utilisée pour le tablespace `SYSTEM` et les tablespaces créés à la création de la base tels que `SYSAUX`, `UNDO`, `TEMP` ou `USERS`

A partir d'Oracle 9i, il est possible d'utiliser plusieurs tailles de bloc dans la base de données.

Jusqu'à 5 autres tailles de bloc peuvent être utilisées.

Les valeurs permises sont 2 ko, 4 ko, 8 ko, 16 ko et 32 ko (certaines plates-formes sont plus restrictives), elles sont utilisées lors de la création des autres tablespaces.

Pour les utiliser, il suffit de configurer des sous-caches correspondants dans le *Database Buffer Cache*.

Un tablespace ne peut pas être créé avec une taille de bloc non standard si le cache correspondant n'est pas configuré auparavant (`DATABASE BUFFER CACHE`).

La colonne `BLOCK_SIZE` de la vue `DBA_TABLESPACES` donne la taille de bloc utilisée par les tablespaces

Cette possibilité d'utiliser plusieurs tailles de bloc est surtout intéressante pour la fonctionnalité de transport de tablespace.

Ainsi un tablespace ayant une taille de bloc de 4 ko peut être transportée dans une base utilisant des blocs de 8 ko.

La taille de bloc d'un tablespace ne peut pas être modifiée sans recréer le tablespace.

18.2.8 Le cryptage des données d'un tablespace (nouveauté 11g)

La version 11g permet de crypter la totalité des données contenues dans un tablespace. Cette option de la base de données est disponible en version Enterprise Edition et est payante.

Elle permet d'utiliser un algorithme de cryptage disponible dans la base de données.

Il existe différents algorithme de cryptage :

- 3DES168
- AES128 (par défaut)
- AES192
- AES256

Pour utiliser un algorithme de cryptage il faut le préciser au moment de la création du tablespace :

```
Create tablespace my_secure_tbs
Datafile '/oracle/oradata/Tahiti/my_secure_tbs01.dbf' size 100M
Encryption using '3DES168' default storage (encrypt) ;
```

Si vous utilisez l'algorithme par défaut il n'est pas nécessaire de préciser l'algorithme :

```
Create tablespace my_secure_tbs
Datafile '/oracle/oradata/Tahiti/my_secure_tbs01.dbf' size 100M
Encryption  default storage (encrypt) ;
```

En version 11g, si on utilise le cryptage des données d'un tablespace, il faut faire attention à ce que les données cryptées ne soient pas plus longues que la taille des colonnes définies pour les tables !
par exemple une colonne définie en VARCHAR(4000) peut avoir un contenu non crypté à 3899 et il devient supérieur à 4000 lorsqu'il est crypté.

Le cryptage des données dans un tablespace a des conséquences significatives sur les performances de la base de données. Il faut donc manipuler cette option de la base de données avec précaution.

18.2.9 Tablespace de travail par défaut

Chaque utilisateur de la base de données possède un tablespace permanent pour stocker des données permanentes et un tablespace temporaire pour les données temporaires (tris),

Oracle vous permet de définir un tablespace permanent par défaut qui est automatiquement utilisé à chaque fois qu'un nouvel utilisateur est créé sans tablespace spécifique permanent.

Le nouveau concept d'un tablespace permanent par défaut ne s'applique pas aux `USERS` system (`SYS`, `SYSTEM`, `OUTLN`). Ces `USERS` utilisent toujours le tablespace `SYSTEM` comme leur tablespace permanent.

Vous ne pouvez pas supprimer un tablespace désigné comme tablespace permanent par défaut. Vous devez d'abord réassigner un autre tablespace aux utilisateurs en tant que tablespace par défaut puis supprimer l'ancien tablespace par défaut.

Il est possible de modifier le tablespace permanent par défaut de la base de données. Le changement prend effet pour les nouveaux objets ou utilisateurs créés après la commande `ALTER DATABASE`.

Après l'exécution de cette commande tous les utilisateurs non « SYSTEM » sont rattachés au tablespace « newusers ».

```
ALTER DATABASE DEFAULT TABLESPACE 'newusers' ;
```

```
SQL> alter database default tablespace users
2  /
Database altered.
SQL> -- Afficher les tablespaces par dÚfinis par dÚfaut
SQL> select property_value, property_name
2  from database_properties
3  where property_name like 'DEFAULT%'
4  /

PROPERTY_VALUE            PROPERTY_NAME
------------------------- ----------------------------------
TEMP                      DEFAULT_TEMP_TABLESPACE
USERS                     DEFAULT_PERMANENT_TABLESPACE
SMALLFILE                 DEFAULT_TBS_TYPE
```

18.2.10 Vues du dictionnaire de données

Plusieurs vues du dictionnaire permettent d'obtenir des informations sur les tablespaces et les fichiers de données :

- `DBA_TABLESPACES` : informations sur les tablespaces
- `DBA_DATA_FILES` : informations sur les fichiers de données
- `V$TABLESPACE` : informations sur les tablespaces (à partir du fichier de contrôle)
- `V$DATAFILE` : informations sur les fichiers de données (à partir du fichier de contrôle)
- `V$SYSAUX_OCCUPANTS` : informations sur les objets des occupants stockés dans le tablespace `SYSAUX`.
- `DBA_FREE_SPACE` : informations sur l'espace disponible à l'intérieur d'un tablespace, un tablespace qui n'a pas d'extent libre n'a pas de ligne dans `DBA_FREE_SPACE`.
- `DBA_SEGMENTS` : informations sur les segments alloués à l'intérieur d'un tablespace
- `DBA_EXTENTS` : informations sur les extents alloués à l'intérieur d'un tablespace

19 Tablespaces SYSTEM & SYSAUX

19.1 Tablespace SYSTEM

Le tablespace SYSTEM est le tablespace qui contient le dictionnaire de données.

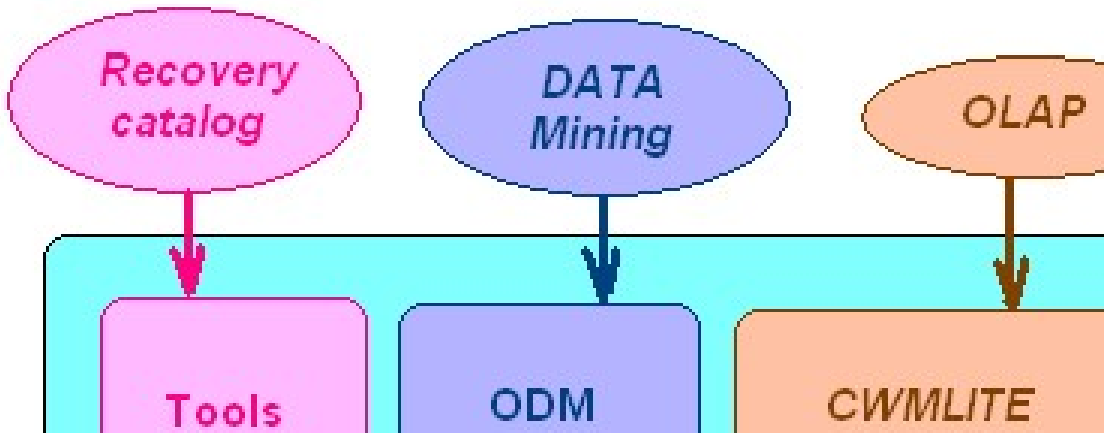
Le dictionnaire de données est installé à la création de la base de données dans le tablespace SYSTEM par l'utilisateur SYS.

Pour des raisons de performance et de taille de stockage, le tablespace SYSTEM ne contient plus d'autres occupants à partir de la version Oracle 10g. Ces occupants sont placés dans le tablespace SYSAUX.

19.2 Tablespace SYSAUX

Le tablespace SYSAUX est un tablespace auxiliaire au tablespace SYSTEM.

Des composants utilisaient dans les versions antérieures d'Oracle le tablespace SYSTEM ou leur propre tablespace. Depuis la version 10g, ces composants utilisent le tablespace SYSAUX comme destination par défaut pour stocker leurs données.



Ce tablespace est obligatoire et doit être créé au moment de la création de la base de données ou de son *upgrade*.

Ce tablespace SYSAUX est défini avec les attributs :

- Permanent
- Read Write
- Extent Management Local
- Segment Space Management Auto

Comme pour le tablespace SYSTEM, il est possible de créer vos propres tables à l'intérieur du tablespace SYSAUX ce qui est vivement déconseillé.

Ce nouveau tablespace fournit un emplacement centralisé pour toutes les « métas data » auxiliaires de la base de données qui ne résident pas dans le tablespace SYSTEM. Il réduit ainsi le nombre de tablespaces créés par défaut.

Le tablespace SYSAUX n'est pas un tablespace transportable.

Vous ne devez pas supprimer ou renommer le tablespace SYSAUX.
Si le tablespace SYSAUX n'est pas disponible (ENABLE) certaines fonctionnalités de la base de données ne fonctionneront plus.

19.2.1 Avantages du tablespace SYSAUX

- **Réduction du nombre de tablespaces :**
  Plusieurs fonctionnalités d'Oracle nécessitent un tablespace spécifique pour stocker les données. Le plus souvent un tablespace séparé est créé. En conséquence, les DBA se retrouvent dans l'obligation de gérer un grand nombre de tablespaces. Le tablespace SYSAUX est le tablespace par défaut pour ces produits Oracle en utilisant le tablespace SYSAUX, le travail du dba est simplifié.
- **Réduction de la charge sur le tablespace SYSTEM :**
  Certains produits ou fonctionnalités Oracle utilisent le tablespace SYSTEM comme tablespace par défaut pour stocker les données. De ce fait, les performances de la base sont dégradées à cause d'un large volume de données dans le tablespace SYSTEM. De plus, cela peut affecter la disponibilité de la base à cause des risques de corruption et de manque d'espace disque au fur et à mesure que celui-ci grandit.
- **Gestion simplifiée du RAC :**
  Pour les utilisateurs du RAC avec des raw devices (périphérique), un raw device doit être alloué pour chaque tablespace créé. Gérer un grand nombre de raw device peut être difficile. Regrouper ces tablespaces dans le tablespace SYSAUX réduit le nombre de raw device qu'un DBA doit allouer.

Le tablespace SYSAUX peut être modifié avec la commande « ALTER TABLESPACE ». Il n'y a pas de changement de syntaxe pour le tablespace SYSAUX.

Vous pouvez réaliser des tâches d'administration classiques telles que : ajouter des fichiers de données au tablespace SYSAUX.

Vous devez avoir le privilège SYSDBA pour modifier le tablespace SYSAUX.

19.2.2 Délocaliser les occupants du tablespace SYSAUX

Après la création vous pouvez surveiller l'utilisation de l'espace de chaque occupant à l'intérieur du tablespace SYSAUX en utilisant la console OEM. Si vous détectez qu'un composant prend trop d'espace dans le tablespace SYSAUX, vous pouvez déplacer l'occupant dans un tablespace différent.

```
•      Déplacer les données d'OLAP
•      du tablespace sysaux vers le nouveau tablepsace CWMLITE.

Select occupant_name, space_usage_kbytes
From v$sysaux_occupants ;
```

```
Select occupant_name, schema_name, move_procedure
From v$sysaux occupants ;

Exec WKSYS.MOVE_WK('CWMLITE');
Exec WKSYS.MOVE_WK('SYSAUX');
```

La seconde requête est utilisée pour déterminer quelle procédure doit être utilisée pour déplacer l'occupant correspondant hors du tablespace `SYSAUX`. Cette procédure gère les deux directions, entrant et sortant du tablespace `SYSAUX`.

- Les procédures fonctionnent base `ONLINE`.

Les occupants suivants ne peuvent pas être déplacés :

- `STREAMS, SMC, STATSPACK, ORDIM, ORDIM/PLUGINS, ORDIM/SQLMM, JOB_SCHEDULER.`

20 Tablespace UNDO

Depuis la version 9i, le tablespace UNDO permet la gestion automatique des segments d'annulation. Cette fonctionnalité est appelée *Automatic Undo Management* (AUM) ou encore *System Managed Undo* (SMU).

Utiliser la gestion MANUELLE des rollbacks segments en version 11g peut provoquer de graves problèmes de performance !

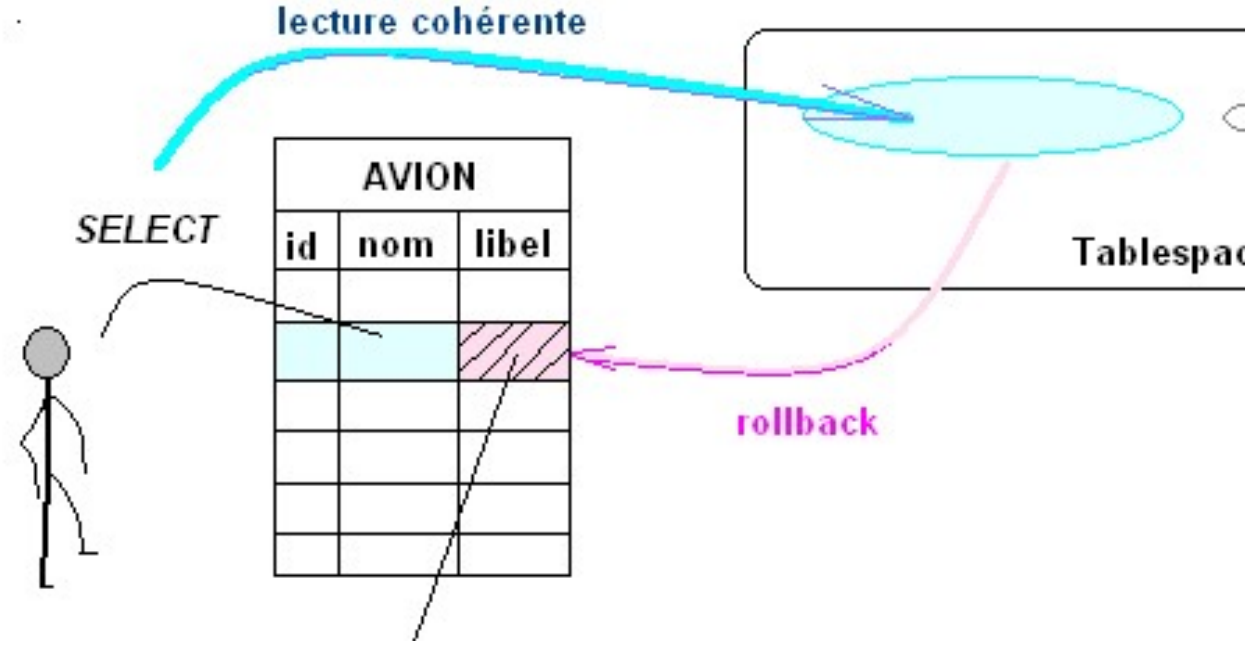
Un segment d'annulation est un segment utilisé pour stocker la version précédente (« *image avant* ») des données en cours de modification dans une transaction qui n'est pas encore validée (`non COMMITée`).

- Permet l'annulation de la transaction (`ROLLBACK`)
- Permet la lecture cohérente (même version des données lues dans un `SELECT`)
- Certaines fonctionnalités de FLASHBACK
- En cas de RECOVER pour annuler des modifications non commitées

Si la transaction est validée (`COMMIT`), l'espace sera libéré, par contre si elle est annulée (`ROLLBACK`) la version précédente des données sera réécrite.

La lecture cohérente est le fait que les données en cours de modification ne sont pas vues par les autres utilisateurs tant que la transaction n'est pas validée.

Lorsqu'un utilisateur interroge une table en cours de modification, Oracle utilise l'image avant des données stockées dans les segments d'annulation pour lui répondre.



Avant la version 9i les segments d'annulation n'existaient pas, les transactions étaient gérées par des *rollbacks segments*. Les rollbacks segments étaient gérés par les DBA.

Aujourd'hui encore une base de données a toujours au moins un *rollback segment* nommé `SYSTEM`, réservé aux transactions qui concernent les objets du tablespace `SYSTEM`, stocké dans le tablespace `SYSTEM`. Ce *rollback segment* est utilisé pour les accès au dictionnaire de données lors de la création de la base de données.

D'autres segments d'annulation sont nécessaires pour les transactions qui concernent les objets contenus dans les tablespaces (tables).

Un segment d'annulation est stocké dans un tablespace, composé d'extents dont les blocs sont chargés dans le Database Buffer Cache en fonction des besoins.
A partir de la version 9i, le tablespace UNDO remplace les rollbacks segments. Il contient des segments d'annulation gérés par Oracle.

Les segments d'annulation sont constitués d'extents.

Les modifications apportées aux blocs des segments d'annulation dans le *Database Buffer Cache* sont enregistrés dans les fichiers du tablespace UNDO si l'instance a besoin d'espace en mémoire vive.

Ils sont également enregistrés dans les redo Logs. Ainsi, en cas de restauration d'instance ou de restauration de média (restauration d'un tablespace), Oracle est en mesure de reconstituer les segments d'annulation et d'annuler les modifications déjà écrites dans les fichiers de données pour annuler les transactions non validées au moment de l'incident.

Les extents d'un segment d'annulation sont utilisés de la façon suivante :

- Les transactions écrivent dans l'extent courant
- Lorsque l'extent courant est plein, les transactions passent au suivant si celui-ci est libre (pas de transaction active à l'intérieur)
- Si l'extent suivant n'est pas libre, un nouvel extent est alloué au segment d'annulation

Par défaut, c'est Oracle qui alloue les segments d'annulation aux transactions en cherchant à répartir les transactions concurrentes sur les différents segments d'annulation.

Lorsqu'une transaction commence à utiliser un segment d'annulation, elle ne peut pas changer de segment en cours de déroulement. Elle inscrit son identifiant dans l'en-tête du segment puis utilise les blocs dont elle a besoin dans le segment.

Lorsque la transaction se termine, elle libère le segment d'annulation mais les informations de rollback ne sont pas supprimées immédiatement.

Ces informations peuvent encore être utiles pour une lecture cohérente.

⇨ Ces informations sont conservées pendant toute la durée de rétention définie par le paramètre `UNDO_RETENTION`.

Si le paramètre `UNDO_RETENTION` est trop faible (900 par défaut), l'erreur « `snapshot too old` » apparaît.
➔ il faut augmenter la durée de rétention !

Un segment d'annulation peut contenir plusieurs transactions simultanées, les extents pouvant même être partagés entre plusieurs transactions (mais pas les blocs).

Une transaction bloquée peut bloquer un extent et obliger le segment à grossir alors que d'autres extents sont libres derrière l'extent bloqué ; cette situation peut être détectée grâce à la vue `V$TRANSACTION`.

Lorsqu'un segment d'annulation grossit et atteint la limite de la taille du tablespace `UNDO`, il faut retailler le tablespace `UNDO` puis relancer la transaction.

20.1 Fonctionnement du tablespace UNDO

Dans un tablespace d'annulation les segments d'annulation sont automatiquement gérés par Oracle et lui seul.

- Nommés `_SYSSMUnn$`
- Dimensionnés automatiquement en fonction des besoins (nombre et taille).
- S'étendent et rétrécissent en fonction des besoins.

```
SQL>  COL name FOR A10
SQL> SELECT n.name,s.extends,s.shrinks,s.wraps,s.hwmsize,s.writes
2  FROM v$rollstat s,v$rollname n
3  WHERE n.usn = s.usn
4  ORDER BY n.usn
5  /

NAME          EXTENDS    SHRINKS      WRAPS    HWMSIZE     WRITES
---------- ---------- ---------- ---------- ---------- ----------
SYSTEM              0          0          0     385024       4520
_SYSSMU1$           0          0          0    1040384       7176
_SYSSMU2$           0          0          0    1171456        108
_SYSSMU3$           0          0          0    1236992        160
_SYSSMU4$           0          0          0    1499136        160
_SYSSMU5$           0          0          0    1302528        160
_SYSSMU6$           0          0          0     253952        236
_SYSSMU7$           0          0          0     319488         54
_SYSSMU8$           0          0          0     450560         54
_SYSSMU9$           0          0          0    1564672         54
_SYSSMU10$          0          0          0     319488        160

11 ligne(s) sÚlectionnÚe(s).
```

Les segments d'annulation stockés dans le tablespace d'annulation sont automatiquement activés.

Il est impossible de toucher aux segments d'annulation (ajout, suppression, activation, dimensionnement …), c'est Oracle qui les gère.

Lors de la création du tablespace d'annulation, Oracle crée 10 segments d'annulation de 2 extents chacun.

En fonction des besoins, Oracle alloue des extents supplémentaires aux segments d'annulation existants ou crée des segments d'annulation supplémentaires (un par un).

Les segments d'annulation créés ne sont pas supprimés. S'il y a baisse d'activité transactionnelle, Oracle n'en a plus l'utilité, il les passe `OFFLINE`. Si l'instance en a besoin ultérieurement, elle les repassera `ONLINE`.

20.2 Positionner les paramètres de gestion automatique

La gestion automatique des segments d'annulation se fait lors de la création de la base de données.

La gestion automatique des segments UNDO fait appel aux paramètres suivants :

- `UNDO_MANAGEMENT` : mode gestion souhaité pour les informations d'annulation (par défaut à `TRUE`).
- `UNDO_RETENTION` : durée de rétention/conservation des informations d'annulation dans les segments d'annulation (en seconde), 900 par défaut (soit ¼ d'heure).
- `UNDO_TABLESPACE` : nom du tablespace d'annulation à utiliser, si non spécifié Oracle prend le premier tablespace `UNDO` disponible.

```
SQL> show parameters undo
NAME                 TYPE                         VALUE
-------------------- ---------------------------- --------------------
undo_management      string                       AUTO
undo_retention       integer                      900
undo_tablespace      string                       UNDOTBS
```

Si le paramètre d'initialisation `UNDO_MANAGEMENT` est positionné à `AUTO` et que la clause `UNDO TABLESPACE` n'est pas présente, Oracle crée un tablespace d'annulation par défaut :

- Nommé `SYS_UNDOTBS`
- Avec un fichier de données de 10 Mo, positionné en `AUTOEXTEND`

20.3 Créer un tablespace UNDO

La création d'un tablespace `UNDO` se fait lors de la création de la base de données et s'effectue grâce à la clause `UNDO_TABLESPACE` de l'ordre SQL `CREATE DATABASE`.

Le tablespace d'annulation est forcément géré localement, avec une gestion automatique des extents, la clause `EXTENT MANAGEMENT` peut être indiquée, mais la seule valeur autorisée est `LOCAL AUTOALLOCATE`.

Si un nom a été indiqué dans le paramètre d'initialisation `UNDO_TABLESPACE`, utilisez le même nom dans la clause `UNDO TABLESPACE`, sinon :

- La base sera bien créée avec le tablespace spécifié dans la clause `UNDO TABLESPACE`
- Mais Oracle retournera une erreur à l'ouverture de celle-ci

20.4 Changer de tablespace UNDO actif

20.4.1 Créer un tablespace UNDO après création de la base

S'effectue grâce à l'ordre SQL `CREATE UNDO TABLESPACE` :

```
CREATE UNDO TABLESPACE nom
DATAFILE spécification_fichier_data [,…]
;
```

```
SQL> CREATE UNDO TABLESPACE undotbs_essai
2  DATAFILE 'C:\app\guest\oradata\BORA\undotbs_essai01.dbf'
3  SIZE 10M AUTOEXTEND ON NEXT 20M MAXSIZE 50M
4  /

Tablespace crÚÚ.
SQL> select TABLESPACE_NAME,STATUS,SEGMENT_NAME
2  from dba_rollback_segs;

TABLESPACE_NAME                STATUS           SEGMENT_NAME
------------------------------ ---------------- ------------------------------
SYSTEM                         ONLINE           SYSTEM
UNDOTBS                        ONLINE           _SYSSMU1$
UNDOTBS                        ONLINE           _SYSSMU2$
UNDOTBS                        ONLINE           _SYSSMU3$
UNDOTBS                        ONLINE           _SYSSMU4$
UNDOTBS                        ONLINE           _SYSSMU5$
UNDOTBS                        ONLINE           _SYSSMU6$
UNDOTBS                        ONLINE           _SYSSMU7$
UNDOTBS                        ONLINE           _SYSSMU8$
UNDOTBS                        ONLINE           _SYSSMU9$
UNDOTBS                        ONLINE           _SYSSMU10$
UNDOTBS_ESSAI                  OFFLINE          _SYSSMU11$
UNDOTBS_ESSAI                  OFFLINE          _SYSSMU12$
UNDOTBS_ESSAI                  OFFLINE          _SYSSMU13$
UNDOTBS_ESSAI                  OFFLINE          _SYSSMU14$
UNDOTBS_ESSAI                  OFFLINE          _SYSSMU15$
UNDOTBS_ESSAI                  OFFLINE          _SYSSMU16$
UNDOTBS_ESSAI                  OFFLINE          _SYSSMU17$
UNDOTBS_ESSAI                  OFFLINE          _SYSSMU18$
UNDOTBS_ESSAI                  OFFLINE          _SYSSMU19$
UNDOTBS_ESSAI                  OFFLINE          _SYSSMU20$

21 ligne(s) sÚlectionnÚe(s).
```

20.4.2 Changer de tablespace UNDO pendant l'activité de la base

Si la base dispose de plusieurs tablespaces d'annulation, il est possible de changer de tablespace actif en modifiant la valeur du paramètre UNDO_TABLESPACE qui est un paramètre dynamique :

```
ALTER SYSTEM SET UNDO_TABLESPACE = nouveau_nom [ clause_SCOPE ]
;
```

Lors d'un changement de tablespace UNDO actif, les segments d'annulation stockés dans l'ancien tablespace d'annulation sont désactivés (passés OFFLINE) et les segments d'annulation stockés dans le nouveau tablespace d'annulation sont activés (passés ONLINE).

Le changement de tablespace d'annulation actif n'attend pas que les transactions en cours se terminent.

Oracle met les segments d'annulation utilisés dans le statut PENDING OFFLINE (visible uniquement dans V$ROLLSTAT), empêchant ainsi qu'ils soient utilisés par de nouvelles transactions ; les segments d'annulation sans transaction active sont immédiatement passé OFFLINE.

Les segments d'annulation PENDING OFFLINE sont passés définitivement OFFLINE lorsqu'ils ne contiennent plus de transaction active.

Les nouvelles transactions utilisent les segments d'annulation du nouveau tablespace d'annulation actif.

```
SQL> alter system set undo_tablespace=undotbs_essai scope=memory;
SystÞme modifiÚ.
SQL> select TABLESPACE_NAME,STATUS,SEGMENT_NAME
2  from dba_rollback_segs;

TABLESPACE_NAME                STATUS           SEGMENT_NAME
------------------------------ ---------------- ------------------------------
SYSTEM                         ONLINE           SYSTEM
UNDOTBS                        OFFLINE          _SYSSMU1$
UNDOTBS                        OFFLINE          _SYSSMU2$
UNDOTBS                        OFFLINE          _SYSSMU3$
UNDOTBS                        OFFLINE          _SYSSMU4$
UNDOTBS                        OFFLINE          _SYSSMU5$
UNDOTBS                        OFFLINE          _SYSSMU6$
UNDOTBS                        OFFLINE          _SYSSMU7$
UNDOTBS                        OFFLINE          _SYSSMU8$
UNDOTBS                        OFFLINE          _SYSSMU9$
UNDOTBS                        OFFLINE          _SYSSMU10$
UNDOTBS_ESSAI                  ONLINE           _SYSSMU11$
UNDOTBS_ESSAI                  ONLINE           _SYSSMU12$
UNDOTBS_ESSAI                  ONLINE           _SYSSMU13$
UNDOTBS_ESSAI                  ONLINE           _SYSSMU14$
UNDOTBS_ESSAI                  ONLINE           _SYSSMU15$
UNDOTBS_ESSAI                  ONLINE           _SYSSMU16$
UNDOTBS_ESSAI                  ONLINE           _SYSSMU17$
UNDOTBS_ESSAI                  ONLINE           _SYSSMU18$
UNDOTBS_ESSAI                  ONLINE           _SYSSMU19$
UNDOTBS_ESSAI                  ONLINE           _SYSSMU20$

21 ligne(s) sÚlectionnÚe(s).
```

Si des transactions sont en cours dans l'ancien tablespace au moment du changement alors :

- Le changement est pris en compte immédiatement : les nouvelles transactions utilisent le nouveau tablespace d'annulation mais Oracle laisse les transactions actuelles se terminer dans l'ancien
- Les segments d'annulation non utilisés de l'ancien tablespace sont passés `OFFLINE`
- Les segments d'annulation utilisés de l'ancien tablespace sont marqués `PENDING OFFLINE` (dans `V$ROLLSTAT`)
- Lorsque les transactions sont terminées, les segments d'annulation utilisés sont définitivement passés `OFFLINE`
- L'ancien tablespace reste `ONLINE` : il n'est simplement plus le tablespace actif pour l'annulation (défini par le paramètre `UNDO_TABLESPACE`)

Un tablespace qui contient des segments d'annulation `ONLINE` ou `PENDING OFFLINE` ne peut pas être désactivé ou supprimé.

Bien noter que l'ancien tablespace d'annulation actif reste `ONLINE`, mais que les segments d'annulation qu'il contient sont `OFFLINE` (ils ne peuvent pas être passés `ONLINE` à la main) ; si besoin, il faut passer le tablespace d'annulation explicitement `OFFLINE`.

Activer ou désactiver le tablespace, uniquement s'il n'est pas le tablespace d'annulation **actif** et s'il ne contient pas de segments d'annulation `PENDING OFFLINE`

Le passage `OFFLINE` d'un tablespace d'annulation actif, ou contenant encore des segments d'annulation `PENDING OFFLINE`, est interdit.

20.5 Administrer un tablespace UNDO

S'effectue avec les ordres SQL habituels `ALTER TABLESPACE` ou `ALTER DATABASE` (pour la gestion des fichiers de données) comme un tablespace permanent ordinaire, mais restreints aux actions suivantes :

- Activer ou désactiver le tablespace (comme présenté ci-dessus)
- Ajouter un fichier de données
- Modifier la taille ou l'extension automatique d'un fichier de données
- Déplacer un fichier de données

20.5.1 Dimensionner le tablespace UNDO

Utiliser la vue V$UNDOSTAT pour dimensionner le tablespace d'annulation.

Les besoins en espace d'annulation sont liés à la durée de rétention des informations (paramètre UNDO_RETENTION).

La quantité totale d'espace d'annulation nécessaire pour satisfaire la durée de rétention peut être estimée par la formule.

⇨ UNDO_RETENTION * Quantité d'espace d'annulation par seconde

Le paramètre UNDO_RETENTION peut être défini en analysant la valeur de la colonne MAXQUERYLEN de la vue V$UNDOSTAT.

⇨ Donne la durée en seconde de la requête la plus longue sur chaque période analysée

La quantité d'espace d'annulation par seconde peut être estimée en analysant la valeur de la colonne UNDOBLKS de la vue V$UNDOSTAT.

⇨ Donne le nombre de blocs d'annulation utilisés sur chaque période analysée

Pour que le calcul soit pertinent, il est important de faire une analyse en pleine charge.

Exemple

```
3000 blocs utilisés en pleine charge, sur 10' ( Soit 3000 / 600 = 5 blocs par seconde
Durée de rétention = 30' = 1800 seconde ( Soit un besoin de 5 * 1800 = 9000 blocs
Taille de bloc = 4 K0 (  Soit une taille de 4 * 9000 / 1024 = environ 35 Mo
Prendre de la marge ( 50 Mo)
```

Exemple 2

```
SQL> SET LINESIZE 75
SQL> SET TRIMSPOOL ON
SQL> SET PAGESIZE 1000
SQL> SET VERIFY OFF
SQL> COL begin_time FOR A11
SQL> COL end_time FOR A11


SQL> -- interrogation de V$UNDOSTAT
SQL> SELECT
2         TO_CHAR(begin_time,'DD/MM HH24:MI') begin_time,
3         TO_CHAR(end_time,'DD/MM HH24:MI') end_time,
4         undoblks,
5         maxquerylen
6 FROM
7          v$undostat
8 /
BEGIN_TIME    END_TIME    UNDOBLKS MAXQUERYLEN
----------- ----------- ---------- -----------
14/08 11:03   14/08 11:09         0           3
14/08 10:53   14/08 11:03       526          77
14/08 10:43   14/08 10:53       601         192
```

```
14/08 10:33   14/08 10:43      635         72
14/08 10:23   14/08 10:33      666        147
14/08 10:13   14/08 10:23      609         84
14/08 10:03   14/08 10:13      618         65
14/08 09:53   14/08 10:03      150         44
14/08 09:43   14/08 09:53     1000         17
14/08 09:33   14/08 09:43      374          1
14/08 09:23   14/08 09:33        2          1

11 rows selected.

SQL> -- durée de la requête la plus longue, toutes périodes confondues
SQL> -- * bonne base de départ pour le paramètre UNDO_RETENTION

SQL> SELECT  MAX(maxquerylen) maxquerylen
3 FROM   v$undostat
5 /

MAXQUERYLEN
192
SQL> -- plus grand nombre de blocs d'annulation utilisés,
SQL> -- toutes périodes confondues
SQL> SELECT  MAX(undoblks) undoblks,
3            MAX(undoblks) / 600 « UNDOBLKS/SECONDE »
4 FROM   v$undostat
6 /
UNDOBLKS UNDOBLKS/SECONDE
---------- ----------------
1000         1,66666667
SQL> -- estimation des besoins minimums en espace d'annulation
SQL> SELECT MAX(maxquerylen)*(MAX(undoblks) / 600)*block_size_ko  « Taille (ko) »
4 FROM   v$undostat,
6        (SELECT value/1024 block_size_ko FROM v$parameter
7 WHERE name = 'db_block_size') p
8 GROUP BY  block_size_ko
10 /

Taille (ko)
1280
```

20.5.2 Supprimer un tablespace UNDO

S'effectue avec l'ordre SQL habituel `DROP TABLESPACE`, la clause `INCLUDING CONTENTS` est implicite

⇨ les segments d'annulation stockés dans le tablespace sont supprimés.

Par contre, la clause doit être mentionnée avec l'option `AND DATAFILES` pour supprimer les fichiers de données associés.

20.6 Vues du dictionnaire de données

Plusieurs vues du dictionnaire permettent d'obtenir des informations sur les tablespaces d'annulation et les segments d'annulation :

- `V$TABLESPACE` ou `DBA_TABLESPACES` : informations sur les tablespaces (dont les tablespaces d'annulation)
- `DBA_DATA_FILE` ou `V$DATAFILE` : informations sur les fichiers
- `DBA_FREE_SPACE` : espace disponible

- DBA_SEGMENTS : liste de segments
- DBA_UNDO_EXTENTS : liste des extents alloués dans le tablespace d'annulation
- DBA_ROLLBACK_SEGS : informations sur les segments d'annulation et/ou les rollback segments
- V$ROLLNAME : liste des segments d'annulation et/ou des rollback segments actuellement ONLINE ou PENDING OFFLINE
- V$ROLLSTAT : statistiques sur les segments d'annulation et/ou les rollback segments actuellement ONLINE ou PENDING OFFLINE
- V$UNDOSTAT : statistiques sur les informations d'annulation
- V$TRANSACTION : informations sur les transactions actives

La vue V$UNDOSTAT donne des statistiques sur les informations d'annulation générées sur les dernières 24 heures. La collecte est effectuée par plages de 10 minutes ; la vue contient donc 144 lignes, une pour chaque plage de 10 minutes sur les dernières 24 heures.

La première ligne de la vue correspond à la plage en cours (peut faire moins de 10 minutes). Les colonnes intéressantes de la vue sont les suivantes :

- BEGIN_TIME : Date/heure de début de la plage
- END_TIME : Date/heure de fin de la plage
- UNDOBLKS : Nombre de blocs d'annulation utilisés en cumulé sur la période
- TXNCOUNT : Nombre de transactions totales sur la période
- MAXQUERYLEN : Durée en seconde de la requête la plus longue sur la période
- MAXCONCURRENCY : Nombre maximum de transactions simultanées sur la période

La vue V$ROLLSTAT est utilisée pour l'optimisation des rollback segments.

Dans la pratique il faut surveiller le nombre de fois où le rollback segment a dû s'étendre (EXTENTS) par rapport au nombre de fois où il a pu tourner sans problème (WRAPS)

Si les rollback segments sont bien dimensionnés, le ratio EXTENDS/WRAPS est faible.

- Un ratio supérieur à 10% indique que les rollback segments sont trop petits.

Plusieurs vues du dictionnaire permettent d'obtenir des informations sur les rollback segments :

- DBA_ROLLBACK_SEGS : informations sur tous les rollback segments
- V$ROLLNAME : liste des rollback segments ONLINE
- V$ROLLSTAT : statistiques sur les rollback segments ONLINE
- V$TRANSACTION : informations sur les transactions actives

V$ROLLNAME	
USN	Numéro du segment
NAME	Nom du segment d'annulation ou de rollback

V$TRANSACTION	
SES_ADDR	Adresse de la session utilisateur qui exécute la transaction une jointure avec V$SESSION permet de récupérer des informations sur la session
START_TIME	Date et heure de démarrage de la transaction (sous forme de chaîne)
XIDUSN	Numéro du segment d'annulation ou de rollback utilisé pour la transaction

V$UNDOSTAT	
BEGIN_TIME	Date et heure de démarrage de la plage
END_TIME	Date et heure de fin de plage
UNDOBLKS	Nombre de blocs d'annulation utilisés en cumulé sur la période
TXNCOUNT	Nombre de transactions totales sur la période
MAXQUERYLEN	Durée en seconde de la requête la plus longue sur la période
MAXCONCURRENCY	Nombre maximum de transactions simultanées sur la période

V$ROLLSTAT	
USN	Numéro du rollback segment
EXTENT	Nombre d'extents dans le segment d'annulation ou de rollback
RSSIZE	Taille actuelle utile en octet (sans le bloc d'en-tête) du segment d'annulation ou de rollback
TABLESPACE_NAME	Nom du tablespace
WRITES	Nombre d'octets écrits dans le segment
OPTSIZE	Valeur OPTIMALE en octet du segment d'annulation ou de rollback
HWMSIZE	Plus grande taille jamais atteinte en octet par le segment
SHRINKS	Nombre de fois où le segment a rétréci
WRAPS	Nombre de fois où le segment a tourné (où il est venu réécrire dans le premier extent sans avoir eu besoin de s'étendre)
EXTENDS	Nombre de fois où le segment s'est étendu (allocation d'un nouvel extent)
AVESHRINK	Taille moyenne en octets des rétrécissements du segment d'annulation ou de rollback.

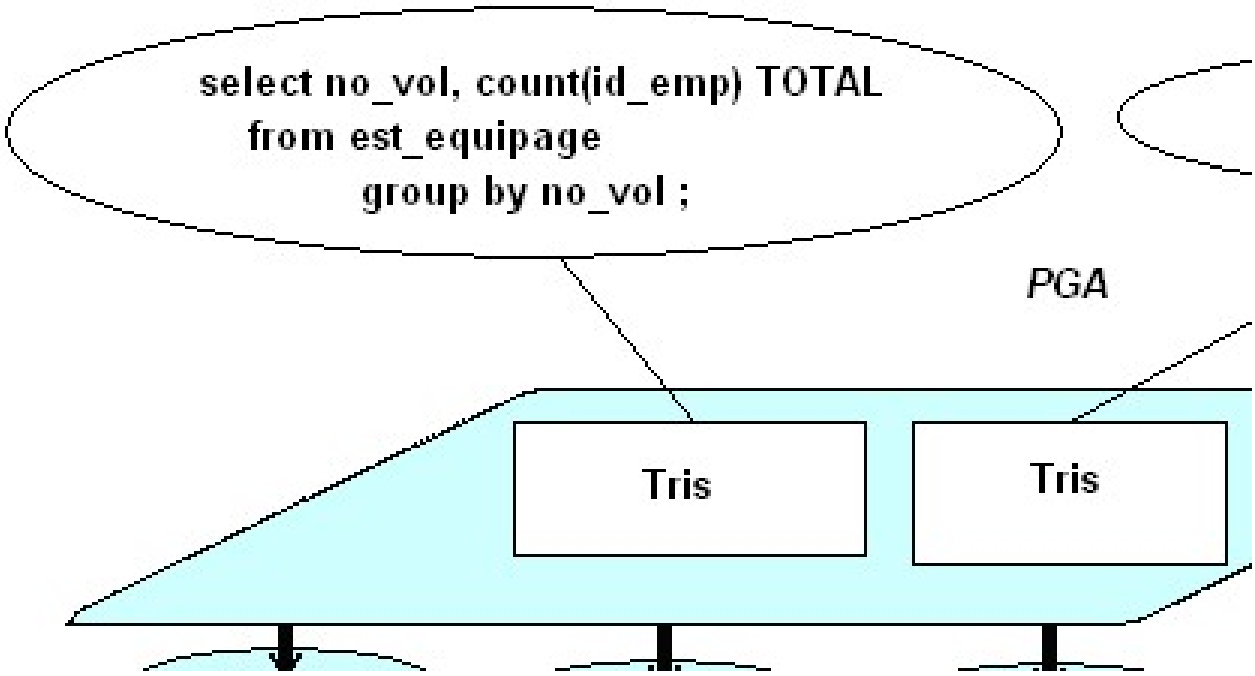
21 Tablespaces temporaires

Le tablespace temporaire héberge les segments temporaires issus des requêtes ou de commandes SQL telles que :

- select … order by
- select … group by
- select distinct ...
- les requêtes ensemblistes (`UNION, MINUS, INTERSECT`)
- create `INDEX` …
- gestion des statistiques (DBMS_STATS)
- les jointures par tri-fusion (`SORT, MERGE, JOIN`)
- les tables temporaires (CREATE GLOBAL TEMPORARY TABLE …)

Lorsqu'une requête nécessite un tri, Oracle tente de faire le tri en mémoire dans la PGA du processus serveur qui exécute la requête. S'il ne peut pas car le tri ne tient pas en mémoire, Oracle le découpe en morceaux et trie chaque morceau en stockant des résultats intermédiaires sur disque dans des segments temporaires.

Au bout d'un certain temps ces segments temporaires sont supprimés par Oracle.



Du point de vue des performances, il est conseillé de dédier un tablespace aux segments temporaires en utilisant le mot clé TEMPORARY à la création du tablespace.
obligatoire depuis la 9i.
- Evite la fragmentation et optimise les performances !

Exemple

```
Si 50 users font un tri de 1 Méga chacun, on aura 50 PGA de 1 Méga et si il n'y a pas
assez de place mémoire alors oracle utilisera le tablespace temporaire.
```

Les segments doivent avoir une taille correspondant à un multiple de SORT_AREA_SIZE (5 à 10 fois) + un bloc d'en-tête pour les données de contrôle.

Depuis Oracle9i, le paramètre PGA_AGGREGATE_TARGET permet la gestion dynamique de la PGA et des paramètres SORT_AREA_SIZE et autres. Depuis la version 9i il est également possible de définir un tablespace temporaire par défaut, dès la création de la base de données.

Il est attribué par défaut aux utilisateurs qui ont été créé sans tablespace temporaires par défaut (si la clause TEMPORARY TABLESPACE n'est pas précisée dans l'ordre SQL CREATE USER).

Le tablespace temporaire par défaut ne peut pas être supprimé.
Si possible, mettre le tablespace temporaire sur un disque à part.

Le tablespace temporaire par défaut ainsi créé est forcément géré localement. Il a une taille de bloc correspondant au paramètre DB_BLOCK_SIZE.

```
•      informations sur les tablespaces par défaut

SQL> col property_value for A15
SQL> select  property_name, property_value
2  from  database_properties
3  where property_name  like 'DEFAULT_%'
4  /

PROPERTY_NAME                  PROPERTY_VALUE
------------------------------ ---------------
DEFAULT_TEMP_TABLESPACE        TEMP
DEFAULT_PERMANENT_TABLESPACE   USERS
DEFAULT_TBS_TYPE               SMALLFILE
```

21.1 Créer un tablespace temporaire

Depuis Oracle 8i, un tablespace temporaire peut être géré localement. Dans ce cas, il utilise des fichiers temporaires.

Depuis la version 9i, il est possible de définir un tablespace temporaire par défaut à la création de la base de données afin de recueillir les tris des requêtes de tous les utilisateurs de la base de données.

Il est créé avec l'ordre SQL `CREATE TEMPORARY TABLESPACE`.

```
CREATE [ BIGFILE | SMALLFILE ] TEMPORARY TABLESPACE Ts_nom
TEMPFILE spécification_fichier_temp [,..]
[EXTENT MANAGEMENT LOCAL] [UNIFORM [SIZE valeur [K|M] ] ]
[ TABLESPACE GROUP  nom_group ]
;
```

- spécification_fichier_temp

```
'nom_fichier' [ SIZE valeur [K|M|G|T] ] [REUSE]
AUTOEXTEND { OFF|ON [ NEXT valeur [K|M|M|T] ]
[ MAXSIZE { UNLIMITED | valeur [K|M|G|T] } ]
                    }
```

  - par défaut la taille des extents est de 1Mo, la clause SIZE peut être utilisée pour spécifier une autre taille, dans ce cas le mot clé UNIFORM doit être mentionné.
  - BIGFILE, précise que le fichier est gros mais dans ce cas un seul fichier pourra être rattaché au tablespace.
  - TABLESPACE GROUP, spécifie le groupe d'appartenance du tablespace. Si le groupe n'existe pas, il est créé implicitement. Par défaut, le tablespace n'appartient à aucun groupe.

Les fichiers de données d'un tablespace temporaire géré localement sont particuliers : Ils sont dits temporaires « `TEMPFILE` ».

```
SQL> create temporary tablespace ts_essai
2  tempfile 'C:\app\guest\oradata\BORA\essai_temp.dbf'
3  size 5M;

Tablespace created.
```

21.2 Groupes de tablespaces temporaires

Avant la version 10g, une requête ne pouvait utiliser qu'un seul tablespace temporaire. Ceci posait des problèmes de performance pour les requêtes exécutées en parallèle (chaque processus serveur sollicite un accès au tablespace temporaire, d'où des problèmes de contentions et de performances).

Depuis la version Oracle 10g, il est possible de définir des groupes de tablespaces temporaires. Le nom d'un groupe peut être utilisé partout où un nom de tablespace temporaire est utilisé.

Ils respectent les règles suivantes :

- Un groupe ne peut pas porter le même nom qu'un tablespace temporaire.
- Un groupe est créé lorsqu'un tablespace est affecté au groupe.
- Un groupe est implicitement supprimé lorsque le dernier tablespace est retiré du groupe.

Vous pouvez changer le groupe d'appartenance du tablespace par la commande `ALTER TABLESPACE` en précisant le nouveau nom du groupe auquel il appartient, ou en spécifiant une chaîne vide, ce qui le retire du groupe.

```
ALTER TABLESPACE nom [ GROUP | '' ]
;
```

```
SQL> select * from dba_tablespace_groups ;
no rows selected
SQL> alter tablespace ts_essai tablespace group group_essai
2  /

Tablespace altered.
SQL> select * from dba_tablespace_groups;
GROUP_NAME           TABLESPACE_NAME
-------------------- ------------------------------
GROUP_ESSAI          TS_ESSAI

SQL> alter tablespace ts_essai tablespace group « ;
Tablespace altered.
SQL> select * from dba_tablespace_groups;
no rows selected
```

21.3 Administrer les tablespaces temporaires

L'administration d'un tablespace temporaire géré localement s'effectue avec les ordres SQL habituels, en remplaçant le mot clé `DATAFILE` par le mot clé `TEMPFILE` :

- `ALTER TABLESPACE`
- `ALTER DATABASE` pour la gestion des fichiers de données
- `DROP TABLESPACE` pour la suppression

Il existe néanmoins quelques restrictions :

- Un tablespace temporaire ne peut pas être mis `OFFLINE`.
- Les fichiers d'un tablespace temporaire ne peuvent pas être déplacés, ils doivent être supprimés puis recréés.
- Toujours en mode `NOLOGGING`, les modifications ne sont pas enregistrées dans les fichiers de Redo Log (intéressant pour les performances).
- Il ne peut pas être sauvegardé.

21.3.1 Agrandir un tablespace temporaire

Il est possible d'ajouter un fichier temporaire à un tablespace temporaire.

```
ALTER TABLESPACE nom_ts_temp
ADD TEMPFILE spécification_fichier_temp
;
```

```
SQL> alter tablespace ts_essai
2  add tempfile 'D:\Oracle\product\10.1.0\Oradata\orcl\essai_temp02.dbf'
3  size 5M;

Tablespace altered.
SQL> select t.name  ts, bigfile, enabled, f.name
2  from v$tablespace t, v$tempfile f
3  where t.ts#=f.ts#
4  UNION
5  select t.name ts, bigfile, enabled, f.name
6  from v$tablespace t, v$datafile f
7  where t.ts#=f.ts#
8  /

TS         BIG ENABLED     NAME
---------- --- ----------- ------------------------------------------------------
EXAMPLE    NO  READ WRITE  D:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\EXAMPLE01.DBF
SYSAUX     NO  READ WRITE  D:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\SYSAUX01.DBF
SYSTEM     NO  READ WRITE  D:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\SYSTEM01.DBF
TEMP       NO  READ WRITE  D:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\TEMP01.DBF
TS_ESSAI   NO  READ WRITE  D:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\ESSAI_TEMP.DBF
TS_ESSAI   NO  READ WRITE  D:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\ESSAI_TEMP02.DBF
UNDOTBS1   NO  READ WRITE  D:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\UNDOTBS01.DBF
USERS      NO  READ WRITE  D:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\USERS01.DBF

8 rows selected.
```

21.3.2 Modifier la clause AUTOEXTEND :

```
ALTER DATABASE TEMPFILE 'nom_complet' [, …] clause_autoextend
;
```

```
SQL> alter database
2  tempfile 'D:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\ESSAI_TEMP02.DBF'
3  autoextend on
4  /

Database altered.
SQL> select file_name, autoextensible, maxbytes, maxblocks
2  from dba_temp_files;
```

```
SQL> select file_name, autoextensible, maxbytes, maxblocks
2  from dba_temp_files;

FILE_NAME                                                  AUT   MAXBYTES  MAXBLOCKS
---------------------------------------------------------- --- ---------- ----------
D:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\TEMP01.DBF           YES 3,4360E+10    4194302
D:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\ESSAI_TEMP.DBF       NO           0          0
D:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\ESSAI_TEMP02.DBF     YES 3,4360E+10    4194302
```

21.3.3 Modifier la taille d'un fichier temporaire

```
ALTER DATABASE TEMPFILE 'nom_complet' [, …] RESIZE valeur [L|M]
;
```

```
SQL> alter database
2  tempfile 'D:\ORACLE\PRODUCT\10.1.0\ORADATA\ORCL\ESSAI_TEMP02.DBF'
3  resize 2M
4  /

Base de donnÚes modifiÚe.
```

21.3.4 Rétrécir un tablespace temporaire (Nouveautés 11g)

Certaines opérations d'administration peuvent demander beaucoup d'espace temporaire et donc d'avoir un tablespace temporaire de très grande taille.

Pour faire face à ce type de besoin, la version 11g propose la commande `SHRINK` qui permet de réduire la taille du tablespace ou de l'un de ses fichiers :

```
ALTER TABLESPACE nom_tbs SHRINK SPACE [ KEEP taille [K|M|G|T] ] ;
ALTER TABLESPACE nom_tbs SHRINK TEMPFILE '/chemin/nom_fichier.dbf' | No_fichier
                         [ KEEP taille [K|M|G|T] ] ;
```

- Cette commande permet de réduire la taille du tablespace temporaire ou la taille d'un de ses fichiers.
- La clause KEEP permet de préciser une taille minimum à concerver pour le tablespace ou le fichier

Exemples

```
ALTER TABLESPACE temp  SHRINK SPACE ;
ALTER TABLESPACE temp SHRINK TEMPFILE '/oracle/oradata/orcl/temp01.dbf' ;
```

Il est possible de préciser une quantité d'espace libre supplémentaire laissée dans le tablespace en utilisant la clause KEEP pour le tablespace ou le fichier temporaire, mais si la clause KEEP est trop basse une erreur est retournée.

Sans clause KEEP Oracle tente de libérer le maximum d'espace au niveau du tablespace (de tous les fichiers rattachés) ou du fichier spécifié.

Exemples

```
ALTER TABLESPACE temp  SHRINK SPACE KEEP 100M ;
ALTER TABLESPACE temp SHRINK
TEMPFILE '/oracle/oradata/orcl/temp01.dbf'
KEEP 100M ;
```

21.3.5 Supprimer un tablespace temporaire

S'effectue avec l'ordre SQL `DROP TABLESPACE`.

```
DROP TABLESPACE nom [ INCLUDING CONTENTS [ AND DATAFILES ] ]
;
```

```
SQL> drop tablespace ts_essai including contents and datafiles
2  /

Tablespace dropped.
```

21.4 Définir un tablespace temporaire par défaut

La création du tablespace temporaire par défaut peut se faire après création de la base de données avec l'ordre `CREATE TEMPORARY TABLESPACE`.

Il faut ensuite désigner ce tablespace temporaire comme tablespace temporaire par défaut en utilisant la commande ci-dessous.

```
ALTER DATABASE DEFAULT TEMPORARY TABLESPACE nom_tbs
;
```

Dés l'affectation du nouveau tablespace temporaire comme tablespace par défaut, Oracle mets les segments de temporaire dans le nouveau tablespace et libère l'ancien tablespace.

Il est possible de définir un groupe de tablespace temporaire par défaut à la place d'un tablespace temporaire par défaut.

En cas de perte de fichier temporaire supplémentaire, oracle ouvre la base normalement mais ne le signale pas directement.
Par contre une erreur est signalée dans le fichier de trace et le fichier des alertes : attention aux restaurations !

21.5 Vues du dictionnaire de données

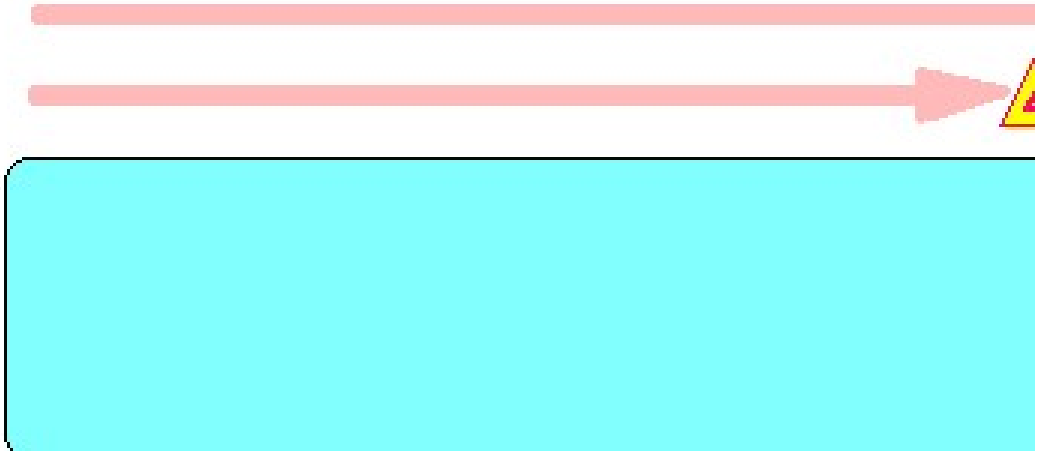
Plusieurs vues du dictionnaire permettent d'obtenir des informations sur les tablespaces et les fichiers de données :

- DBA_TABLESPACES : informations sur les tablespaces
- V$TABLESPACE : informations sur les tablespaces (à partir du fichier de contrôle)
- DBA_TABLESPACE_GROUPS : informations sur les groupes de tablespaces.
- DBA_TEMP_FILES : informations sur les fichiers de données des tablespaces temporaires gérés localement.
- V$TEMPFILE : informations sur les fichiers de données des tablespaces temporaires gérés localement.
- V$SORT_SEGMENT : supervision du stockage des segments temporaires.
- DATABASE_PROPERTIES : informations sur les propriétés par défaut de la base de données, pour le tablespace temporaire par défaut, le nom de la propriété est DEFAULT_TEMP_TABLESPACE.
- **DBA_TEMP_FREE_SPACE : nouvelle vue en version 11g**, qui permet de gérer plus facilement les tablespaces temporaires. Elle contient la taille totale utilisée pour le tablespace temporaire, ainsi que l'espace temporaire utilisé et l'espace temporaire disponible.

```
Select * from dba_temp_free_space ;
TABLESPACE_NAME    TABLESPACE_SIZE    ALLOCATED_SPACE    FREE_SPACE
---------------- ------------------ ------------------ ---------------
TEMP                    18,939,904          4,259,840      14,680,064
```

22 Monitoring de l'utilisation d'un tablespace

Les seuils du tablespace sont définis comme des pourcentages par rapport à la taille du tablespace. Quand le taux de remplissage du tablespace arrive à une de ces 2 limites (voir schéma), une alerte appropriée est déclenchée ou inhibée.



Les valeurs des seuils critiques et d'avertissement peuvent être configurées.

Si une valeur de seuil n'est pas spécifiée, les comportements suivants seront appliqués :

- Les valeurs par défaut sont 85% de taux de remplissage pour l'alerte d'avertissement et 97% pour le seuil critique.
- Les alertes sont désactivées par défaut, les valeurs de seuil sont donc à `NULL`. Ces valeurs peuvent être redéfinies par la suite, à tout moment.

Le process d'arrière plan `MMON` vérifie toutes les 10 minutes ces seuils d'alertes (dépassés ou redescendu). Une alerte est déclenchée dans chacun de ces cas (dépassé ou redescendu)

- Les alertes ne doivent pas être actives pour les tablespaces en lecture seule (Read-only) ou bien pour les tablespaces `OFFLINE` car ça n'a aucun sens.
- Dans les tablespaces temporaires, les valeurs de seuil doivent être définies comme une limite de l'espace utilisé dans le tablespace par les sessions.
- Pour le tablespace `UNDO`, un `EXTENT` est réutilisable s'il ne contient pas des segments `UNDO` actifs. Pour le calcul de la violation des seuils, la somme des `Extents` actifs est considérée comme espace utilisé.
- Pour les tablespaces avec des fichiers auto-extensibles, les seuils sont calculés en fonction de la taille maximale spécifiée du fichier ou de la taille maximale supportée par l'`OS`.

Le suivi de ces opérations se fait en utilisant les tables `DBA_THRESHOLDS` qui contient la configuration des alertes et `DBA_ALERT_HISTORY` qui contient l'historique des alertes.

```
--  visualiser les alertes min et max définies sur les tablespaces
--

select warning_value,critical_value
from dba_thresholds
where metrics_name='Tablespace Space Usage' and
object_name is null
/

select warning_value,critical_value
from dba_thresholds
where metrics_name='Tablespace Space Usage' and
object_name='DATA'
/

select reason,resolution
from dba_alert_history
where object_name='DATA'
/
```

22.1 Configuration des seuils de tablespace

Vous pouvez utiliser le package DBMS_SEVR_ALERT pour définir vos propres valeurs par défaut pour l'utilisation de l'espace dans le tablespace. Les procédures SET_THRESHOLD, et GET_THRESHOLD vous permettent de le faire.

```
•      définit un seuil d'avertissement à 80% du taux de remplissage
•      et un seuil critique à 95% pour le tablespace DATA.
•      Dès que le pourcentage de remplissage est égal ou supérieur à 80%
•      une alerte d'avertissement est déclenchée.
•      L'alerte critique est déclenchée quand le taux de remplissage est
•      égal ou supérieur à 95%;


EXECUTE DBMS_SERVER_ALERT.SET_THRESHOLD
    (
Dbms_server_alert.tablespace_pct_full ,
Dbms_server_alert.operator_ge, 80 ,
Dbms_server_alert.operator_ge, 95, 1, 1, NULL ,
Dbms_server_alert.object_type_tablespace, 'DATA'
    ) ;


•      remet les valeurs par défaut pour le tablespace appelé DATA.
EXECUTE DBMS_SERVER_ALERT.SET_THRESHOLD
    (
Dbms_server_alert.tablespace_pct_full ,
NULL, NULL, NULL, NULL, 1, 1, NULL,
Dbms_server_alert.object_type_tablespace, 'DATA'
    ) ;


•      désactiver le mécanisme de traçage
•      de l'utilisation de l'espace pour le tablespace DATA
EXECUTE DBMS_SERVER_ALERT.SET_THRESHOLD
    (
Dbms_server_alert.tablespace_pct_full ,
Dbms_server_alert.operator_do_not_check, '0' ,
Dbms_server_alert.operator_do_not_check, '0' , 1, 1, NULL ,
Dbms_server_alert.object_type_tablespace, 'DATA'
    ) ;
```

```
•      définir vos propres valeurs de seuil par défaut
•      elles deviennent la valeur de mesure « Tablespace Space Usage »
EXECUTE DBMS_SERVER_ALERT.SET_THRESHOLD
    (
Dbms_server_alert.tablespace_pct_full ,
Dbms_server_alert.operator_ge, 80 ,
Dbms_server_alert.operator_ge, 95, 1, 1, NULL ,
Dbms_server_alert.object_type_tablespace, NULL
    ) ;


•      revenir aux valeurs de 85% et 97% comme valeur de seuil par défaut
EXECUTE DBMS_SERVER_ALERT.SET_THRESHOLD
    (
Dbms_server_alert.tablespace_pct_full ,
NULL, NULL, NULL, NULL, 1, 1, NULL,
Dbms_server_alert.object_type_tablespace, NULL
    ) ;


•      utilisée pour lire les valeurs et les seuils
•      pour l'ensemble de la base de données
EXECUTE DBMS_SERVER_ALERT.GET_THRESHOLD
    (
Dbms_server_alert.tablespace_pct_full ,
:wo, :wv, :co, :cv, :op, :cocc, NULL,
Dbms_server_alert.object_type_tablespace, NULL
    ) ;


•      utilisée pour lire les valeurs et les seuils
•      pour le tablespace DATA
EXECUTE DBMS_SERVER_ALERT.GET_THRESHOLD
    (
Dbms_server_alert.tablespace_pct_full ,
:wo, :wv, :co, :cv, :op, :cocc, NULL,
Dbms_server_alert.object_type_tablespace, 'DATA'
    ) ;
```

23 Mémoire dynamique et performances

En version 10g d'Oracle, *Automatic PGA Memory Management* est activée par défaut, à moins que PGA_AGGREGATE_TARGET soit explicitement fixée à zéro ou WORKAREA_SIZE_POLICY soit explicitement posée à MANUAL.

PGA_AGGREGATE_TARGET est par défaut à 20% de la taille de la SGA.

Automatic PGA Memory Management permet de conseiller l'utilisateur de façon automatique sur les problèmes de performances de la base de données et de proposer des solutions adaptées.

Avec Oracle 9i, était apparue la notion de *granule*, ainsi que la possibilité de redimensionner la mémoire de façon interactive.

Elle peut toujours être modifiée dynamiquement alors que l'instance est en cours de fonctionnement :

- Augmentée ou diminuée
- Sans devoir arrêter la base
- En modifiant la valeur des paramètres qui la dimensionnent, par l'intermédiaire de l'ordre SQL ALTER SYSTEM

En cas d'augmentation, la taille maximum de la SGA est définie par le paramètre SGA_MAX_SIZE, qui n'est pas dynamique et calculé par défaut si non spécifié.

La taille de la SGA peut augmenter ou diminuer en fonction des besoins internes d'Oracle. C'est notamment le cas au démarrage où les valeurs des différents paramètres de dimensionnement de la SGA peuvent être adaptés par Oracle.

23.1 La notion de granule

Un granule est une quantité de mémoire qui peut être allouée à une structure de la SGA :

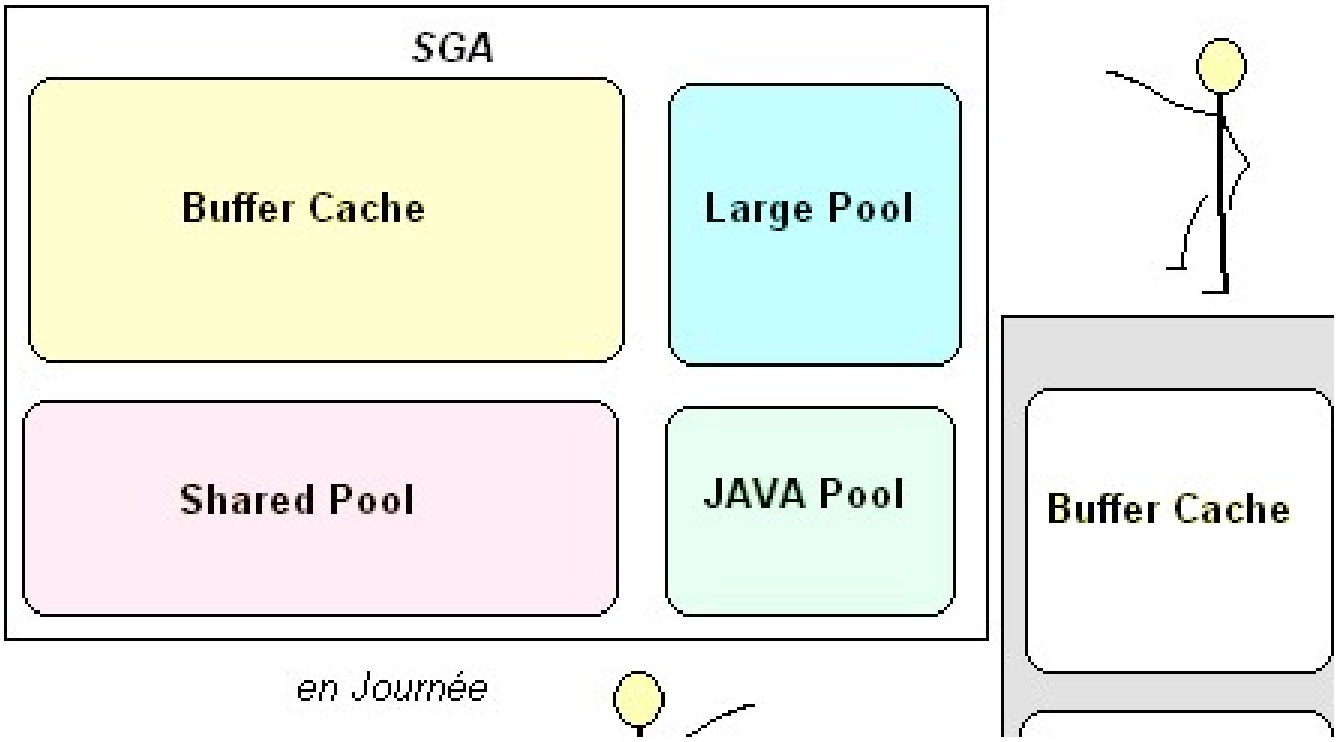
- D'une taille de 4 Mo si la taille totale de la SGA est inférieure à 128 Mo
- D'une taille de 16 Mo autrement

La vue V$SGA_COMPONENT permet de visualiser la taille du granule d'une base de données.

23.2 Gestion automatique du partage de la mémoire en 10g

La gestion automatique du partage de la mémoire est une amélioration importante d'autogestion d'Oracle 10g.

Cette fonctionnalité automatise la gestion des structures les plus importantes de mémoire partagée utilisée par toutes les instances d'une base de données Oracle.



Avec « *Automatic Shared Memory Management* » quand les jobs `OLTP` s'exécutent, le *Buffer Cache* prend la plus grande partie de la mémoire afin d'avoir une bonne performance d'entrée/sortie.

Par contre, lorsque le job batch démarrera plus tard, la mémoire sera automatiquement allouée vers le *Large Pool* afin d'être utilisée par des opérations parallèles de requêtes en évitant les erreurs de débordement de mémoire (*Memory overflow*).

23.2.1 Principes de tuning de la SGA

L'outil « *Automatic Shared Memory Management* » utilise un nouveau processus d'arrière plan appelé *Memory Manager* (`MMAN`).

`MMAN` agit comme un distributeur (*broker*) de mémoire et coordonne la taille allouée aux différents composants de la SGA. Il conserve une trace de la taille des composants et des opérations de dimensionnement en attente.

`MMAN` observe le système et le *Workload* (travail en cours) afin de déterminer la distribution de mémoire optimale. Il exécute ces vérifications toutes les « *quelques minutes* » afin que la mémoire soit toujours affectée là où on en a besoin.

En absence de gestion automatique du partage mémoire les composants doivent être dimensionnés afin d'anticiper leurs besoins en mémoire.

Basé sur les informations extraites du *Workload,* `MMAN` exécute les actions suivantes :

- Effectue des statistiques périodiques dans l'arrière plan.
- Utilise divers outils de conseils sur la mémoire.
- Exécute des analyses de type « *What-IF* » (SI) pour déterminer la meilleure distribution de la mémoire possible.
- Affecte de la mémoire là où elle en a besoin.
- Sauvegarde la taille des composants pendant les arrêts de la base (*shutdown*) si un `SPFILE` est utilisé. (La taille peut être récupérée à partir des valeurs existantes avant l'arrêt).

La mémoire est redistribuée entre les composants suivants :

- DB_CACHE_SIZE
- SHARED_POOL_SIZE
- LARGE_POOL_SIZE
- JAVA_POOL_SIZE

Avant la version 10g, la mémoire additionnelle était allouée à la SGA fixe et la mémoire additionnelle était entre 10 et 20Mo.

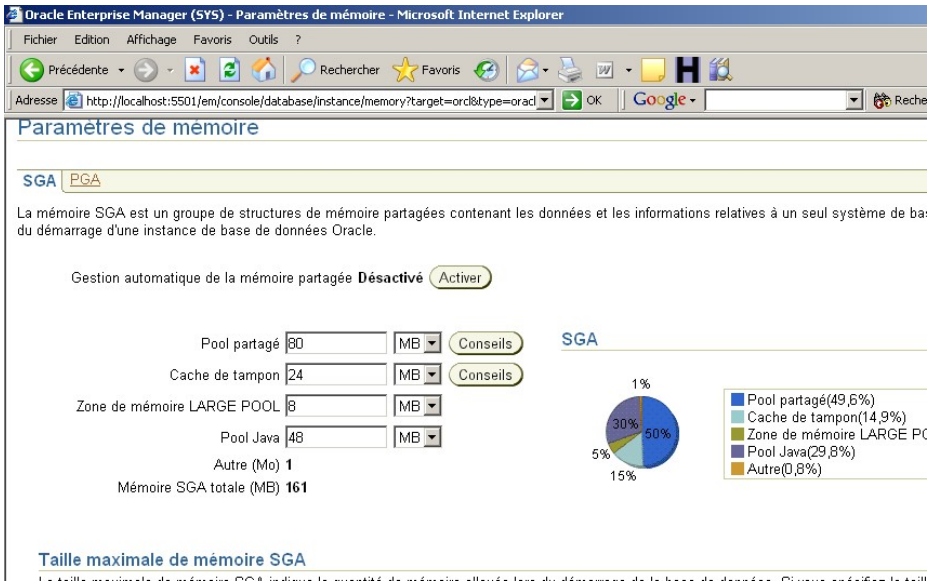
Le paramètre SGA_TARGET inclut toute la mémoire dans la SGA, comprenant aussi les composants dimensionnés automatiquement ou manuellement ainsi que toutes les allocations internes faites pendant le démarrage.

Mettre SGA_TARGET à la valeur SGA_MAX_SIZE !

23.2.2 SGA_TARGET et le Database Control (OEM)

Vous pouvez utiliser le Database Control pour configurer le « *Automatic shared Memory Management* » comme indiqué dans le mode opératoire ci-dessous (dans l'ordre) :

- Cliquez sur l'onglet « Administration »
- Sélectionner sous le nom de l'instance « Memory Parameter »
- Cliquez l'onglet « SGA »
- Cliquez sur le bouton « ENABLE » pour « *Automatic shared Memory Management* » puis saisir la taille totale de la SGA (en Mo). Cette valeur est affectée à SGA_TARGET.



23.2.3 Configuration manuelle de SGA_TARGET

« *Automatic shared Memory Management* » se configure avec le paramètre `SGA_TARGET`.

Si vous spécifiez une valeur différente de zéro, les 4 pools de mémoire seront dimensionnés automatiquement :

- Database buffer cache (pool par défaut)
- Shared pool (pool partagé)
- Large pool
- Java pool

Si `SGA_TARGET` = zéro (valeur par défaut), « *Automatic shared Memory Management* » est inactivé.

Les paramètres individuels utilisés dans les versions antérieures pour spécifier les tailles des composants dimensionnés automatiquement sont toujours actifs.

Les paramètres d'initialisation qui dimensionnent ces pools sont maintenant appelés paramètres « *Auto-tuned* » de la SGA.

Les buffers suivants sont des composants dimensionnés manuellement :

- Log buffer
- Autres buffer cache (`KEEP/RECYCLE`, autres dimensions de blocks associés au buffer cache).
- Streams pool
- SGA fixe et autre allocation interne

En version 10g et 11g le paramètre `STATISTICS_LEVEL` doit être positionné à la valeur `TYPICAL` !

23.2.4 Comportement des paramètres Auto-tuned

Quand `SGA_TARGET` n'est pas définie ou est égale à zéro les paramètres auto-réglés de la `SGA` se comportent comme dans les versions précédentes,

⇨ sauf `SHARED_POOL_SIZE` qui fait exception.

Les allocations des *overhead* internes (temps utilisé pour le système lui-même) pour les méta-données (comme les structures de données pour les processus, les sessions, etc..) sont maintenant incluses dans la valeur du paramètre `SHARED_POOL_SIZE`.

Par conséquent, vous pouvez avoir besoin d'augmenter la valeur du paramètre `SHARED_POOL_SIZE` pour migrer vers Oracle 10g afin de prendre en compte ces allocations.

Par exemple, si vous utilisiez une valeur de 256 Mo dans une version précédente et la valeur des allocations internes était de 32 Mo en Oracle 10g, vous devrez mettre la valeur de `SHARED_POOL_SIZE` à 288 Mo pour obtenir la même taille de pool.

```
•      calcule de la valeur totale de la Shared Pool
•      en incluant cet overhead (supplementaire) interne.

SELECT SUM(bytes)/1024/1024
FROM   V$SGASTAT
WHERE  pool = 'shared pool'
/
```

Cette requête permet de déterminer la nouvelle valeur de la *Shared Pool* en version 10g.

Quand SGA_TARGET est différent de 0, les paramètres auto-réglés de la SGA ont une valeur par défaut égale à zéro. Ces composants sont dimensionnés par l'algorithme du « *Automatic shared Memory Management* ».

Si leur valeur est différente de 0, ces valeurs sont utilisées comme une limite inférieure par l'algorithme d'autoréglage.

Par exemple si SGA_TARGET = 8 Go et SHARED_POOL_SIZE = 1Go, ceci indique à l'algorithme du « *Automatic Shared Memory Management* » de ne jamais baisser la valeur de la *Shared Pool* en dessous de 1Go et seules des valeurs supérieures seront acceptées.

Pour déterminer la taille effective des composants auto-réglés de la SGA, utilisez la requête suivante :

```
SELECT component, current_size/1024/1024
FROM   V$SGA_DYNAMIC_COMPONENTS
/
```

23.2.5 Comportement des paramètres Manuels

Les paramètres manuels de la SGA sont :

- DB_KEEP_CACHE_SIZE
- DB_RECYCLE_CACHE_SIZE
- DBn_CACHE_SIZE (n= 2,4,8,16,32)
- LOG_BUFFER
- STREAMS_POOL_SIZE

Ils sont spécifiés par l'utilisateur et leur taille détermine la taille de leur composant.

Quand SGA_TARGET a une valeur, la taille totale des paramètres manuels est soustraite à cette valeur. Le reste donne la valeur des composants auto-réglés de la SGA.

Par exemple si SGA_TARGET = 8 Go et DB_KEEP_CACHE_SIZE = 1Go alors la taille totale des 4 composants auto-réglés est limitée à 7 Go.

Les 7 Go incluent la SGA fixe et le Buffer Log et seulement après avoir alloué ceci, le reste de la mémoire est divisée entre les composants auto-réglés. La taille du KEEP_CACHE est de 1 Go comme spécifié par le paramètre.

Vues V$PARAMETER

Quand vous spécifiez une valeur différente de zéro pour SGA_TARGET et que vous ne spécifiez pas de valeur pour les paramètres auto-réglés, la valeur de ces paramètres dans la vue V$PARAMETER est 0 et la valeur de la colonne ISDEFAULT = TRUE.

Si vous avez spécifié une valeur pour au moins un des paramètres auto-réglés, la valeur affichée dans la vue V$PARAMETER est la valeur que vous avez spécifiée pour ce paramètre.

```
SELECT name, value, isdefault
FROM  V$PARAMETER
WHERE name like '%size'
/
```

23.2.6 Redimensionner SGA_TARGET

Le paramètre dynamique SGA_TARGET est :

- Dynamique
- Peut être réduit jusqu'à la valeur SGA_MAX_SIZE
- Peut être augmenté jusqu'à ce que tous les composants *auto_tuned* aient atteint leur taille minimale.

La modification du paramètre SGA_TARGET influence automatiquement la taille des autres composants.

Cette modification peut se faire *via* le *Database Control* ou une commande ALTER SYSTEM.

23.2.7 Désactiver la gestion automatique de la mémoire en version 10g

Il est possible de désactiver le partage de gestion automatique de la mémoire en mettant SGA_TARGET à zéro, dans ce cas les valeurs de tous les paramètres auto-dimensionnés prennent la taille définie en mémoire des composants correspondants, même si les utilisateurs ont spécifié des valeurs non nulles pour ces paramètres.

23.3 Gestion automatique de la mémoire en 11g

La gestion automatique du partage de la mémoire reste une amélioration importante d'autogestion d'Oracle 11g. Elle permet une gestion automatique de l'ensemble de la mémoire (SGA et PGA).

⇨ Le paramètre MEMORY_TARGET permet d'activer cette fonctionnalité.

Ce paramètre indique la taille globale de la SGA et de la PGA utilisée par oracle.

Les tailles respectives de ces zones peuvent changer en fonction des besoins.

La taille globale que la totalité de la mémoire oracle peut atteindre est donnée par le paramètre :

⇨ MEMORY_MAX_TARGET

MEMORY_MAX_TARGET est un paramètre statique alors que MEMORY_TARGET est un paramètre dynamique.

Le processus d'arrière plan *Automatic Memory Management* utilise et adapte automatiquement les paramètres SGA_TARGET et PGA_TARGET pour allouer la mémoire à l'instance.

Le paramètre MEMORY_MAX_TARGET défini une valeur maximum de mémoire utilisée par Oracle et la bloque. Il doit obligatoirement avoir une valeur supérieure ou égale à la valeur de MEMORY_TARGET.

23.3.1 Désactiver la gestion automatique de la mémoire en version 11g

Si ces deux paramètres sont positionnés à zéro, la gestion de la mémoire dynamique est désactivée.

Dans ce cas, les valeurs des paramètres `SGA_TARGET` et `PGA_AGREGTE_TARGET` sont désactivés (valeur à zéro), et il faut utiliser les anciens paramètres `DB_CACHE_SIZE, SHARED_POLL_SIZE, etc..`

23.3.2 La vue dynamique V$MEMORY_TARGET_ADVICE

Cette vue dynamique de performances, permet de suivre l'allocation dynamique et visualiser les différentes valeurs de l'allocation dynamique de la mémoire.

Cette vue contient les colonnes :

- Memory size : taille réelle de la mémoire totale allouée à l'instance
- Size_factor : coefficient de taille
- Estd_db_time : taille de l'instance utilisée en mémoire en moyenne par rapport aux facteurs size-factor et time_factor.
- Time_factor : coefficient de temps
- Version :

```
Select * from v$memory_target_advice ;
```

La vue `V$MEMORY_DYNAMIC_COMPONENTS`, permet de visualiser les différentes valeurs de chaque pool, entre autre la shared_pool, le database buffer cache, le large pool, etc …

```
Select component, current_size, min_size, max_size
from v$memory_dynamic_components ;
```

23.3.3 Nouveau cache en version 11g : `result_cache`

Ce cache est un nouveau composant de la SGA et est utilisé par Oracle pour initialiser le paramètre MEMORY_TARGET.

Par défaut ce paramètre est positionné à une valeur égale à 128K.

23.4 L'optimiseur Oracle

L'optimiseur de requêtes SQL permet de générer des plans d'exécution des requêtes performants. Pa exemple utiliser un index si une table a de nombreuses lignes sera plus rapide que d'effectuer un balayage complet de table (FULL TABLE SCAN).

Pour que l'optimiseur de requêtes génère des plans d'exécutions optimales, des statistiques doivent être générées sur les objets (tables et index).

Les statistiques donnent à l'optimiseur des indications sur la volumétrie des objets mais ces statistiques sont également utilisent pour connaître la dégradation des objets (fragmentation, profondeur des index)

Autrefois (avant la version 9i), il fallait conserver une trace des objets pour déterminer si une collecte de statistiques était nécessaire. Si un objet n'avait pas de statistiques ou si elles étaient périmées, des plans d'exécution SQL erronés étaient générés et provoquait une dégradation des performances lors de l'exécution des requêtes.

Avec oracle 9i, si la supervision était utilisée, la commande DBMS_STATS, pouvait être utilisée pour collecter des statistiques.

```
DBMS_STATS.GATHER_SCHEMA_STATS(schema_name, option=> 'GATHER AUTO')
;
```

Cette commande générait des statistiques optimisées, en incluant des histogrammes, sur les objets pour lesquels, les statistiques étaient considérées comme périmées. Mais dans ce cas il fallait activer le monitoring et DBMS_STATS régulièrement.

Avec la version 10g, l'outil de statistiques automatise ces taches et vous annonce s'il est nécessaire de générer des statistiques ou non.

Cette caractéristique réduit la probabilité d'exécuter du code SQL non performant causé par des statistiques inexistantes ou périmées.

23.4.1 Les optimiseurs RBO et CBO

L'optimiseur de requêtes RBO disparaît en version 9i. Cet optimiseur était basé sur des règles. Par exemple règle 1 si un index existe il faut l'utiliser.

RBO existe toujours dans la version 10g mais il n'est plus supporté, c'est-à-dire qu'aucun changement de code n'a été fait pour RBO et il n'y aura plus aucune correction des bugs.

Rappel de la version 9i

Le paramètre OPTIMIZER_MODE configure l'instance pour l'optimisation syntaxique RBO ou statistique CBO.

- OPTIMIZER_MODE = CHOOSE (CBO, valeur par défaut)
- OPTIMIZER_MODE = RULE (RBO)
- OPTIMIZER_MODE = FIRST_ROWS (CBO, avec l'objectif de minimiser le temps de réponse pour extraire la première ligne)
- OPTIMIZER_MODE = ALL_ROWS (CBO, avec l'objectif de minimiser le temps de réponse pour extraire toutes les lignes)

A partir de la version 10g seul l'optimiseur CBO est utilisé par Oracle. C'est un optimiseur basé sur les coûts. Cet optimiseur tient compte des statistiques génères dans la base pour générer les plans d'exécution des requêtes.

⇨ CBO est plus performant

A partir de la version 10g comme Oracle ne supporte que CBO, toutes les applications qui tournent sur la 10g doivent utiliser cet optimisateur (voir la notice oracle Metalink (189702.1) pour la décharge du support technique concernant le RBO). Cette notice fournie des détails sur l'abandon du support RBO et la migration des applications basées sur le RBO vers le CBO.

Voici certaines conséquences qu'il est possible de rencontrer :

- Les valeurs CHOOSE et RULE ne sont plus reconnues comme valeurs pour le paramètre d'initialisation OPTIMIZER_MODE. Les fonctionnalités de ces valeurs existent toujours mais seront supprimées dans une version future. Ceci est aussi valable pour les HINTS correspondants au RBO.
- ALL_ROWS est la valeur par défaut pour le paramètre d'initialisation OPTIMIZER_MODE.
- Les applications qui utilisent RBO doivent migrer vers CBO.

Dans la 1ere release de oracle 9i, le model de coût était limité seulement par les facteurs I/O. la 9i introduisait CPU *costing* pour rendre possible la comptabilisation des opérations CPU-intensive et CPU-only (CPU seule).

Par exemple une des opérations importante de CPU-only est d'extraire des données du buffer cache.

Aujourd'hui CBO a évolué et est devenu beaucoup plus performant.

Le « SQL *Tuning Advisor* » est une nouvelle fonctionnalité de l'optimisateur de requêtes depuis la version 10g, qui automatise l'ensemble des processus de Tuning du SQL.

En utilisant le nouvel optimiseur de requêtes CBO pour faire du Tuning le processus automatique remplace le Tuning manuel qui est complexe répétitif et consommateur de temps

23.4.2 Présentation du SQL Tuning Advisor

2 modes de fonctionnement :

- ⇨ Mode normal ➔ l'utilisateur compile le SQL et génère le plan d'exécution.
  - le mode normale génère les plans d'exécution qui correspondent à la majorité des cas. Sous le mode normal l'optimiseur opère avec des contraintes de temps très strictes, une fraction de secondes pour trouver un bon plan d'exécution
- ⇨ Mode Tuning ➔ l'optimiseur exécute une analyse supplémentaire pour vérifier si le plan d'exécution produit en mode normal peut être encore amélioré.
  - Le résultat de l'optimiseur de requête en mode Tuning n'est pas un plan d'exécution mais une série d'actions et leurs proratas de bénéfice attendus pour produire un plan bien meilleur. Quand il fonctionne dans le mode Tuning l'optimiseur est appelé ATO (*Automatic Tuning Optimizer*). Le *tuning* fait par ATO est appelé « *Tuning SQL Automatique* ».

Le « *Tuning automatique du SQL* » est une nouvelle fonctionnalité de l'optimisateur de requêtes qui automatise l'ensemble des processus de Tuning du SQL.

En utilisant le nouvel optimiseur de requêtes pour faire du Tuning le processus automatique remplace le Tuning manuel qui est complexe, répétitif et consommateur de temps.

Les caractéristiques du « *Tuning automatique du SQL* » sont disponibles via le *SQL Tuning Advisor*.

Dans le mode Tuning l'optimiseur peut prendre plusieurs minutes pour faire le Tuning d'une seule commande. L'ATO est destiné à être utilisé pour des requêtes SQL complexes avec un haut niveau de chargements qui ont un impact non trivial sur l'ensemble de la base.

Le *SQL Tuning Advisor* est en réalité le conducteur de processus de *tuning*. Il appelle l'ATO (*Automatic Tuning Optimizer*) pour exécuter les 4 types d'analyses suivantes :

- Statistics analysis (analyse des statistiques) : ATO vérifie chaque objet d'une requête afin d'identifier des statistiques manquantes ou figées (non générées depuis longtemps) et fournit des recommandations pour effectuer des statistique significatives. Il collecte aussi des informations

supplémentaires pour fournir les statistiques manquantes ou bien pour corriger les statistiques figées si les recommandations ne sont pas implémentées.

- SQL Profiling (créer un profile) : ATO vérifie ses propres estimations et collecte des informations supplémentaires pour corriger des erreurs d'estimation. Il collecte des informations sous forme de paramètres personnalisés de l'optimiseur comme « *First rows* » et « *All rows* » en s'appuyant sur l'historique d'exécution de la requête SQL. Il créé un profile SQL en utilisant ces informations et fait des recommandations pour la création de ce profil. Quand un profil SQL est créé, il permet à l'optimiseur de requêtes en mode normal de générer un plan ajusté.
- Acces path analysis (chemin d'accès) : ATO explore si un nouvel index peut être utilisé pour améliorer d'une manière significative l'accès à chaque table dans une requête et si c'est le cas il fait des recommandations pour la création de cet index.
- SQL structure Analysis : ATO tente d'identifier les commandes SQL qui utilisent de mauvais plans d'exécution et fait des recommandations pour les restructurer. La restructuration suggérée peut être au niveau syntaxique ou sémantique (la table existe, comment y accéder).

Le *tuning* SQL n'est pas simplement un des aspects les plus critiques pour la gestion des performances d'une base oracle mais est aussi une des taches les plus difficiles à accomplir.

Avec oracle 10g la tache d'identification de ces requêtes a été automatisée via l'Automatic *Database Diagnostic Monitor* (ADDM).

L'activité de tuning SQL représente une tache continue car la charge SQL peut changer en fonction des nouveaux modules applicatifs déployés.

Le « *SQL Tuning Advisor* » est un nouvel outil destiné à remplacer le tuning manuel des requêtes SQL.

Les requêtes SQL qui consomment beaucoup de ressource (CPU, I/O et espace temporaire de travail) sont de bons candidats pour *SQL Tuning Advisor.*

L'outil reçoit une ou plusieurs requêtes SQL en entrée et fourni ensuite les éléments suivants :

- des conseils sur l'optimisation des plans d'exécution
- les gains estimés de performances
- la commande effective pour implémenter ces conseils
- les résultats de ce conseil

Vous pouvez choisir d'accepter ces conseils et faire le tuning du SQL via l'optimiseur oracle.

23.4.3 Impacte sur les Statistiques

L'optimisateur de requêtes se base sur les statistiques d'un objet pour générer des plans d'exécution.

Si ces statistiques n'existent pas ou sont figées, l'optimiseur n'a pas l'information nécessaire et peut générer des plans d'exécution aberrants.

L'ATO vérifie chaque objet utilisé par une requête afin de déterminer s'il y a des statistiques manquantes ou figées et produit 2 types de résultats :

- Des informations supplémentaires sous forme de statistiques pour les objets qui n'en ont pas et un facteur d'ajustement des statistiques pour les objets avec des statistiques obsolètes.
- Des recommandations pour produire des statistiques cohérentes pour des objets sans statistiques ou possédant des statistiques aberrantes.

23.5 L'optimiseur Oracle et la gestion des statistiques

La version Oracle 10g vous informe automatiquement des performances et ressources attribuées aux problèmes de performance.

Les statistiques sont automatiquement générées par le job `GATHER_STATS_JOB`. Ce job génère des statistiques sur tous les objets de la base de données.

Ce job est créé automatiquement à la création de la base et est managé par le *scheduler*. Il se déclenche tous les soirs à 22H00.

• Liste des jobs automatisés.

```
select owner, job_name, program_name,
schedule_name, enabled
from dba_scheduler_jobs
;

OWNER  JOB_NAME          PROGRAM_NAME       SCHEDULE_NAME             ENABL
------ ------------------ ------------------ -------------------------- -----
SYS    PURGE_LOG         PURGE_LOG_PROG     DAILY_PURGE_SCHEDULE      TRUE
SYS    GATHER_STATS_JOB  GATHER_STATS_PROG  MAINTENANCE_WINDOW_GROUP  TRUE
```

`GATHER_DATABASE_STATS_JOB_PROC` attribue des priorités aux objets de la base de données afin que ceux qui en ont le plus besoin soient traités en priorité.

Si vous devez effectuer des chargements ponctuels de tables, régénérez manuellement les statistiques sur ces tables après chaque chargement.

L'optimiseur basé sur les coûts (`CBO`) est le seul optimiseur actif dans la version 10G.

CBO travaille en utilisant les statistiques générées sur les tables et les index. Alors il est fortement recommandé de ne pas laisser les objets qui sont souvent modifiés sans statistiques.

Ci-dessous sont présentés quelques conseils pour la collecte des statistiques sur le dictionnaire de données. Cette collecte se fait en utilisant le package `DBMS_STATS` :

- **Régulièrement** : la méthode recommandée est d'utiliser soit `GATHER_DATABASE_STATS` ou `GATHER_SCHEM_STATS`, avec le paramètre `OPTION = GATHER AUTO`.
  Naturellement, ceci présume que la supervision est activée.
  Avec ce moyen seulement, les objets à réanalyser sont traités à chaque fois.
  La nouvelle procédure `GATHER_DICTIONARY_STATS` vous permet d'analyser le dictionnaire de données après un nombre suffisant d'opérations `DDL`.
- **Cas particuliers** : n'oubliez pas d'analyser des objets indépendamment après une période de chargement caractéristique.

23.5.1 Statistiques sur les tables

Depuis la version 10g un job tourne en automatiquement de 22h à 6h les jours de la semaines et tout le week end, pour mettre à jour les statistiques des tables et des index de la base de données. C'est le job : GATHER_STATS_JOB.

Le job GATHER_STATS_JOBS lance une procédure interne :

- DBMS_STATS.GATHER_DATABASE_STATS_JOB_PROC (similaire à la procédure DBMS_STATS.GATHER_DATABASE_STATS avec option GATHER_AUTO) mais en prioritisant les objets de la base de données, c'est-à-dire les objets ayant le plus besoin de statistiques.

Les **statistiques mettent à jour** les colonnes de la vue du dictionnaire de données DBA_TABLES et ses consœurs.

DBA_TABLES

- ✓ **NUM_ROWS** = Nombre de lignes de la table.
- ✓ **BLOCKS** = Nombre de blocs sous la HWM (blocs utilisés).
- ✓ **EMPTY_BLOCKS** = Nombre de blocs au dessus de la HWM (blocs inutilisés).
- ✓ **AVG_SPACE** = Espace libre moyen en octets dans les blocs occupés (sous la HWM).
- ✓ **AVG_ROW_LEN** = Longueur moyenne d'une ligne en octets, en tenant compte des informations de contrôle (en-tête de lignes et en-tête de colonnes).
- ✓ **CHAIN_CNT** = Nombre de lignes chaînées ou migrées.
- ✓ **SAMPLE_SIZE** = Taille de l'échantillon utilisé en cas d'analyse réalisée sur la table.
- ✓ **LAST_ANALYZED** = Date et heure de la dernière analyse réalisée sur la table.

La gestion manuelle de la collecte des STATISTIQUES est faite en utilisant le package DBMS_STATS.

Si vous êtes connecté comme utilisateur SYS, vous pouvez collecter manuellement des statistiques sur tous les objets de la base avec la commande :

```
EXEC DBMS_STATS.GATHER_FIXED_OBJECTS_STATS('ALL') ;
```

Si vous êtes connecté comme utilisateur SYSTEM, vous pouvez collecter manuellement des statistiques sur les objets d'un schéma ou sur une table avec les commandes :

```
DBMS_STATS.GATHER_TABLE_STATS('schema','table') ;
```

```
DBMS_STATS.GATHER_SCHEMA_STATS('schema') ;
```

Exemples

```
Exec dbms_stats.gather_schema_stats('charly');
Exec dbms_stats.gather_table_stats('charly','avion');
```

23.5.2 Interpréter les statistiques générées sur les tables

Les problèmes pouvant être détectés sont l'espace inutilisé alloué à une table et le faible taux d'occupation des blocs.

La génération de statistiques alimente les colonnes de la vue DBA_TABLES.

La valeur de BLOCKS est toujours exacte même si les statistiques sont inexactes ou manquantes.

Espace utilisé par une table

Espace inutilisé alloué à une table :

- Le nombre de blocs inutilisés alloués à la table (EMPTY_BLOCKS) est important et la table ne va plus grossir (ou peu)
- Lié à une clause STORAGE mal adaptée

Faible taux d'occupation des blocs :

Le rapport (DB_BLOCK_SIZE-AVG_SPACE)/DB_BLOCK_SIZE est faible et les lignes actuelles ne vont pas grossir et peu de nouvelles lignes vont être insérées :

- Lié à des valeurs PCTFREE et PCTUSED mal adaptées ou à une suppression importante de données, pose un problème de performance dans le parcours complet de la table

Le package DBMS_SPACE

Le package DBMS_SPACE possède plusieurs procédures qui permettent de superviser le stockage d'un segment.

Les principales procédures sont :

- FREE_BLOCKS : informations sur les blocs libres dans un segment dont l'espace est géré manuellement.
- SPACE_USAGE : informations sur l'occupation des blocs dans un segment dont l'espace est géré automatiquement.
- UNUSED_SPACE : informations sur les blocs inutilisés d'un segment.

Ce package possède d'autres procédures qui permettent d'estimer la taille d'un segment (table ou index), ou la tendance de croissance de celui-ci.

```
•      liste des blocs vides et occupés dans la table AVION de CHARLY
Select t.blocks  Occupes, s.blocks  Alloues
From dba_tables t, dba_segments s
Where s.segment_name = t.table_name
And s.owner = t.owner
And t.table_name = 'AVION'
And t.owner = 'CHARLY'
;
```

23.5.3 Statistiques sur les index

Concernant les index et la dégradation de leur structure, le seul élément vraiment significatif est le nombre de *blocs de feuilles* qui doivent être lus.

Plus ce nombre est faible plus le nombre d'entrées-sorties le sera et plus grande sera la vitesse de lecture des lignes de tables.

⇨ Plus il y aura d'insertions ou de suppressions dans une table et plus le risque de fragmentation sera élevé.

Si les index sont analysés en même temps que les tables, le niveau de parallélisme qui peut s'appliquer au calcul des statistiques sur les tables ne s'applique pas aux index.

Si ceux-ci doivent être analysés en parallèle, il vaut mieux exécuter indépendamment la commande `GATHER_INDEX_STATS`.

```
Exec dbms.stats.gather_schema_stats (
'CHARLY',DBMS_STATS.AUTO_SAMPLE_SIZE
                                    )
;

Execute dbms_stats.gather_schema_statistics('clo01',
20,estimate_percent=>20);
```

Dans l'analyse finale, vérifiez que les statistiques ont été calculées pour tous les index afin d'éviter des statistiques incomplètes dans la base de données.

Il faut savoir qu'un bloc d'index vide n'est pas réutilisé tant que l'index n'est pas compacté ou reconstruit.

Les **statistiques générées sur les index** alimentent les colonnes de la table **DBA_INDEXES**

DBA_INDEXES
✓ **BLEVEL,** = Profondeur de l'arbre au niveau des branches (ne tient pas compte des feuilles). 0 si le bloc racine est égal au bloc feuille. Valeur exacte même en `ESTIMATE`
✓ **LEAF_BLOCKS** = Nombre de blocs feuilles dans l'index
✓ **NUM_ROWS** = Nombre de lignes dans l'index
✓ **DISTINCT_KEY** = Nombre de valeurs distinctes dans l'index
✓ **SAMPLE_SIZE** = Taille de l'échantillon utilisé en cas d'analyse `ESTIMATE`
✓ **LAST_ANALYZED** = Date et heure de la dernière analyse réalisée sur l'index

```
SQL> select  INDEX_NAME,BLEVEL,LEAF_BLOCKS,NUM_ROWS
2  from dba_indexes
3  where index_name='PK_AVION'
4  ;

INDEX_NAME               BLEVEL LEAF_BLOCKS   NUM_ROWS
-------------------- ---------- ----------- ----------
PK_AVION                      0           1          3
```

La hauteur d'un index est un élément clé de réduction du nombre d'entrées-sorties générées par cet index.

La génération de statistiques permet de donner suffisamment d'informations à l'optimiseur CBO sur les index.

23.5.4 Problèmes détectés sur les index

Deux problèmes peuvent être détectés :

- Faible taux d'occupation et/ou profondeur importante de l'index

Profondeur importante de l'index

BLEVEL est élevé (supérieur à 5).

- Lié à un PCTFREE mal adapté lors de la création ou à un index très volatile (beaucoup de mises à jour)
- Dégrade les performances de l'utilisation de l'index.

23.6 Outil de collecte des statistiques

Pour que l'optimiseur de requêtes génère des plans d'exécution optimale, des statistiques doivent être générées sur les objets.

Autrefois (avant la version 9i), il fallait conserver une trace des objets pour déterminer si une collecte de statistiques était nécessaire. Si un objet n'avait pas de statistiques ou si elles étaient périmées, des plans d'exécution SQL erronés étaient générés.

Avec Oracle 9i, si la supervision était utilisée, la commande DBMS_STATS, pouvait être utilisée pour collecter des statistiques.

```
DBMS_STATS.GATHER_SCHEMA_STATS(schema_name, option=> 'GATHER AUTO')
;
```

Cette commande générait des statistiques optimisées, en incluant des histogrammes, sur les objets pour lesquels, les statistiques étaient considérées comme périmées. Mais vous deviez activer le monitoring et DBMS_STATS régulièrement.

Avec la version 10g, l'outil de statistiques automatise ces tâches et vous annonce s'il est nécessaire de générer des statistiques.

Cette caractéristique réduit la probabilité d'exécuter du code SQL non performant causé par des statistiques inexistantes ou périmées.

23.6.1 GATHER_STATS_JOB

Par défaut GATHER_STATS_JOB est créé au moment de la création de la base de données, et exécute la procédure DBMS_STATS.GATHER_DATABASE_STATS_JOB_PROC. Il utilise le scheduler.

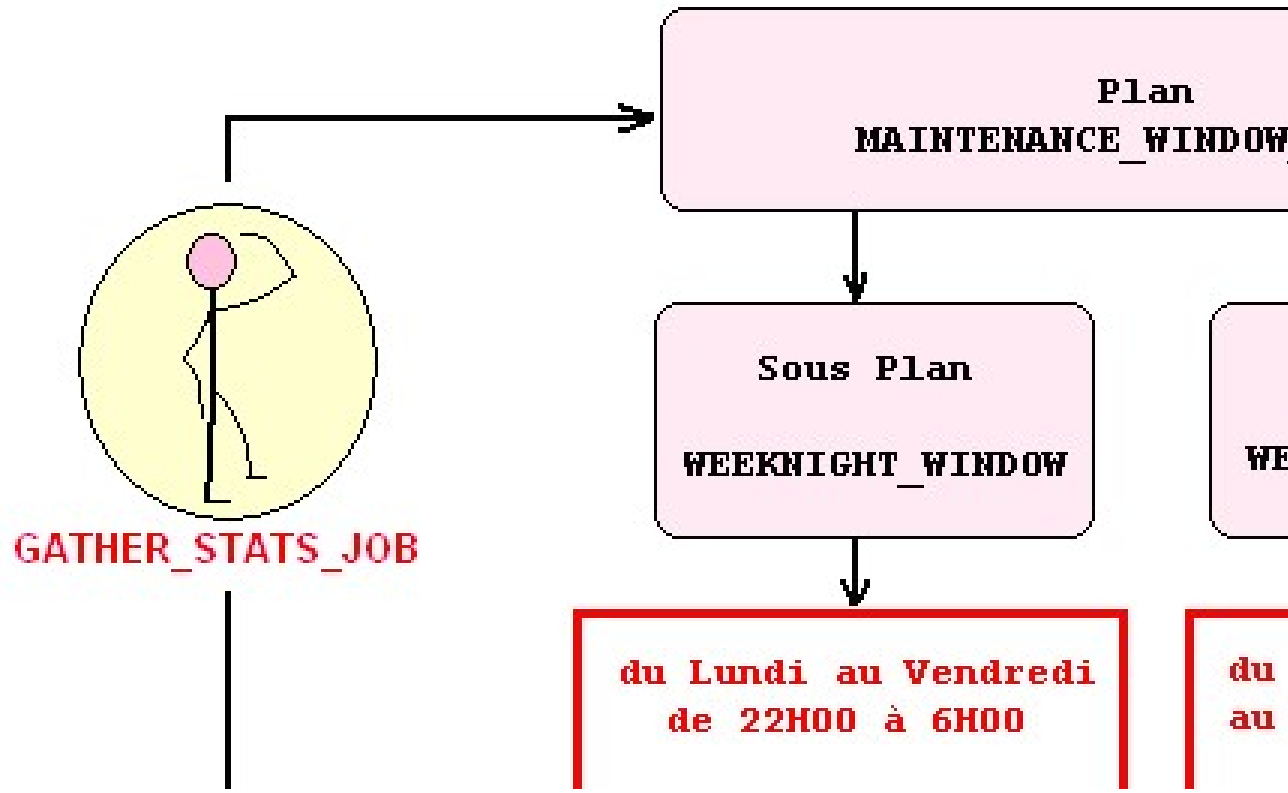
Deux fenêtres sont définies par défaut :

- WEEKNIGHT_WINDOW définie entre 22H00 et 6H00 tous les jours du lundi au vendredi
- WEEKEND_WINDOW définie du samedi midi au Dimanche midi

Un groupe de fenêtres appelées MAINTENANCE_WINDOW_GROUP est défini par défaut pour héberger ces fenêtres.

GATHER _STATS_JOB utilise une classe spécifique du scheduler appelée AUTO_TASKS_JOB_CLASSE. Cette classe est créée automatiquement et est associée à un groupe de consommateur de ressources appelé AUTO_TASKS_CONSUMER_GROUP.

Pour contrôler les ressources utilisées par GATHER_STATS_JOB, il suffit de définir un plan de gestion de ressources pour le MAINTENANCE_WINDOW_GROUP qui alloue les ressources pour l'AUTO_TASKS_CONSUMER_GROUP.



Pour que GATHER _STATS_JOB fonctionne correctement, vous devrez vous assurer que la valeur du paramètre d'initialisation : STATISTIC_LEVEL est égal à TYPICAL.

Si GATHER_STATS_JOB dépasse la fenêtre de temps allouée définie par MAINTENANCE_WINDOW_GROUP, le job continue jusqu'à ce qu'il soit terminé.

Vous devez générer des statistiques manuellement dans les cas suivants :

- Après une opération de chargement
- Pour les tables externes
- Pour collecter des statistiques systèmes
- Pour les objets en mémoire (*fixed objects*)

23.6.2 Modifier l'exécution des statistiques

Il est possible d'ajuster le temps d'ouverture des fenêtres de gestion prédéfinies. Par exemple, il est possible de changer l'intervalle de temps ou la fréquence de génération des statistiques. Il est également possible d'ajouter des plans de ressources à ces fenêtres pour contrôler les ressources utilisées par GATHER_STATS_JOB.

La page « Scheduler Windows » s'affiche. Dans cette page, apparaît une fenêtre :

⇨ cliquer sur le bouton « Edit » pour changer les paramètres.

A partir de cette page vous pouvez aussi ouvrir et fermer une fenêtre sélectionnée. Pour ce faire, sélectionnez l'action correspondante dans la liste déroulante de la fenêtre choisie puis :

⇨ cliquez sur « Go ».

Le bouton Modifier permet de modifier les paramètres.

Il est possible de désactiver la collecte automatique des statistiques en allant sur la page « Jobs » de l'onglet « Administration » puis désactiver GATHER_STATS_JOB.

23.7 Automatic Database Diagnostic Monitor (ADDM)

L'*Automatic Database Diagnostic Monitor* (ADDM), est un moteur d'autodiagnostic intégré directement dans la base de données Oracle.

L'ADDM met en œuvre les actions suivantes :

⇨ Il regarde l'ensemble du système
⇨ Il pose un diagnostic
⇨ puis propose des solutions ou vous envoie vers d'autres composants (par exemple *le SQL Tuning Advisor*)

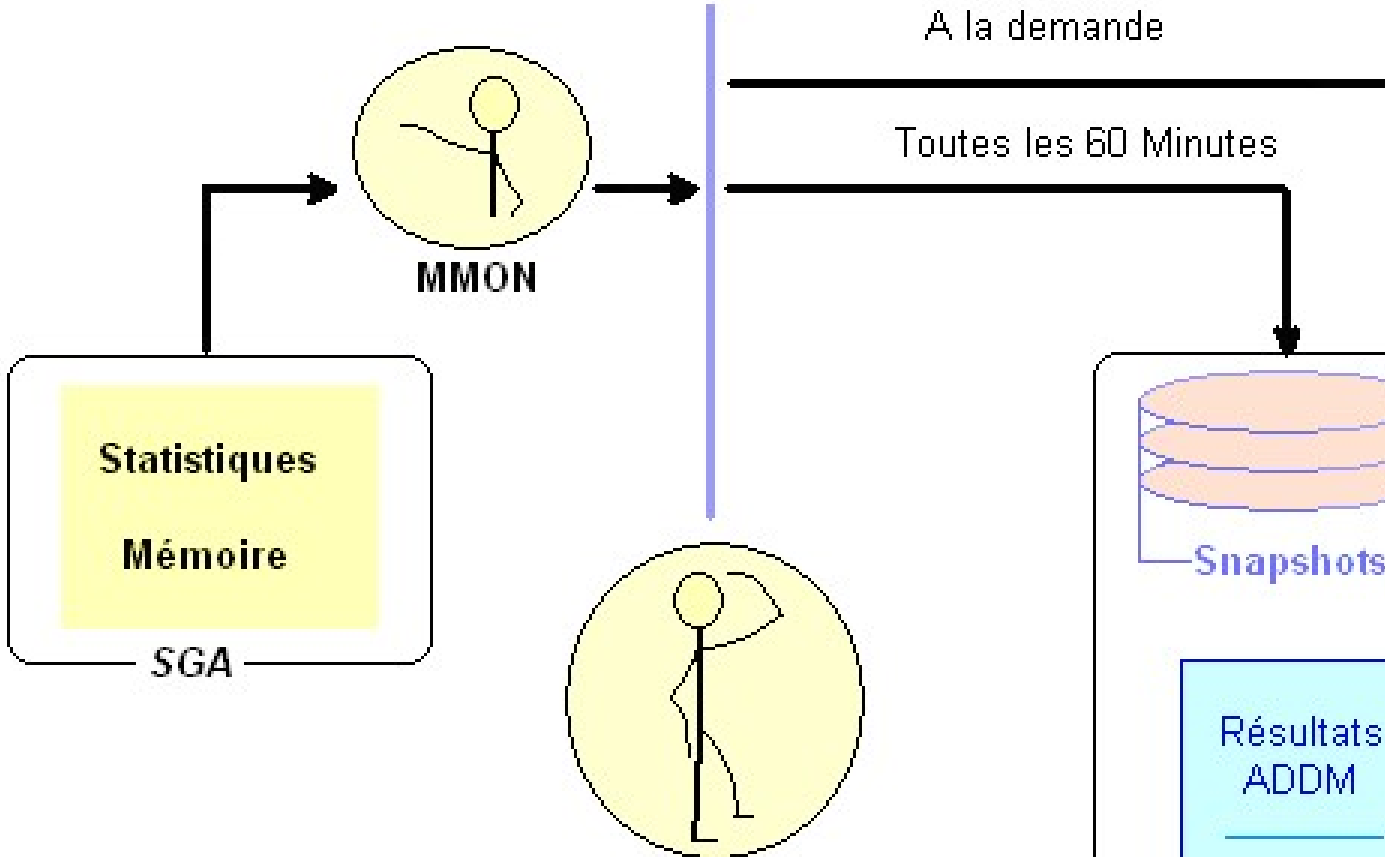
ADDM est appelé automatiquement par la base Oracle et effectue une analyse pour déterminer les problèmes potentiels principaux du système d'une manière préventive.

ADDM signale l'impacte qu'un problème particulier peut avoir sur le système global. Si une recommandation est faite, ADDM précise l'amélioration attendue.

ADDM documente également les portions du système qui n'ont aucun problème.

La méthodologie utilisée par ADDM peut être appliquée sur tous les types de bases de données :

- OLTP
- Data Warehouse
- Environnements mixtes



Par défaut la base Oracle capte automatiquement des informations statistiques à partir de la SGA toutes les 60 minutes et les stocke dans l'Automatic *Workload Repository* (AWR) sous forme de *snapshots*.

Ces *snapshots* sont stockés sur le disque.

ADDM est programmé par le processus MMON pour tourner automatiquement sur chaque instance de la base afin de détecter les problèmes d'une manière préventive.

Chaque fois qu'un *snapshot* est créé, ADDM est activé pour faire une analyse de la période correspondant aux 2 derniers *snapshots*. Cette approche supervise d'une manière préventive l'instance et détecte les goulots d'étranglement avant qu'ils ne deviennent des problèmes conséquents.

Les résultats de chaque analyse ADDM sont stockés dans l'AWR (dans le tablespace SYSAUX) et sont aussi accessibles *via* le *Database Control*.

Il est possible d'invoquer manuellement une analyse d'ADDM sur la période définie entre 2 *snapshots* quelconques même si ADDM analyse la performance de la base Oracle sur la période définie par les 2 derniers *snapshots*.

23.7.1 Méthode d'analyse utilisée par ADDM

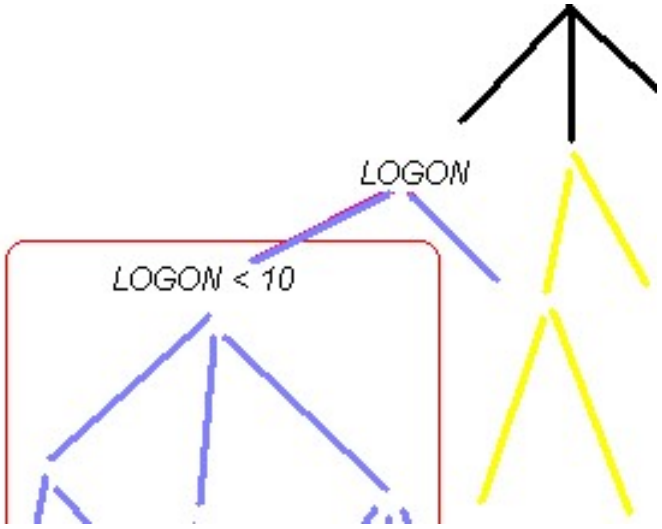
L'analyse ADDM utilise une approche *Top-Down* (de haut en bas) qui se focalise sur l'identification des goulots d'étranglements pour l'accès aux ressources.

Ceci est fait principalement en utilisant le nouveau modèle de statistiques de temps qui aide à déterminer la quantité de temps utilisée dans la base Oracle.

Cette méthode permet à ADDM d'identifier en premier le problème qui a l'impact le plus important sur toute la base.

En interne, ADDM utilise une structure arborescente pour représenter tous les problèmes possibles de *tuning*. L'arbre est basé sur le nouveau modèle de statistiques d'attente (*Wait*) utilisé par la base Oracle.

Cet arbre de classification est basé sur des décades d'expertises d'Oracle en tuning :



Le nœud initial de cet arbre, représente les symptômes ; en descendant vers les feuilles, ADDM identifie les problèmes principaux de performance.

ADDM parcourt l'arbre en utilisant des seuils de temps pour chaque nœud. Si le seuil de temps n'est pas dépassé pour un nœud en particulier, ADDM coupe le sous-arbre correspondant, alors il ne balaie pas l'arborescence sous ce nœud car si le nœud père ne pose pas de problème, alors les nœuds fils non plus. Cette structure arborescente permet à ADDM de déterminer efficacement la zone de recherche pour une identification rapide des problèmes.

L'exécution de l'analyse ADDM a un impact mineur sur le système et ne prend pas plus de 3 secondes pour s'exécuter.

Exemple

```
Examiner l'ouverture de sessions (LOGON) d'un système.
Si une des règles est que le ratio d'entrée en session ne doit pas dépasser 10 par
seconde, alors en utilisant les données du nouveau modèle temps ADDM peut déterminer d'une
manière quantitative que les connections (LOGON) prennent 20% du temps de l'utilisation de
la base oracle.
Ces problèmes étant quantifiés par ADDM (20% du temps) il ne vous reste plus qu'à les
résoudre, puisque vous connaissez les valeurs quantifiées et les impacts sur le système.
```

23.7.2 Résultats de l'analyse ADDM dans le grid Control

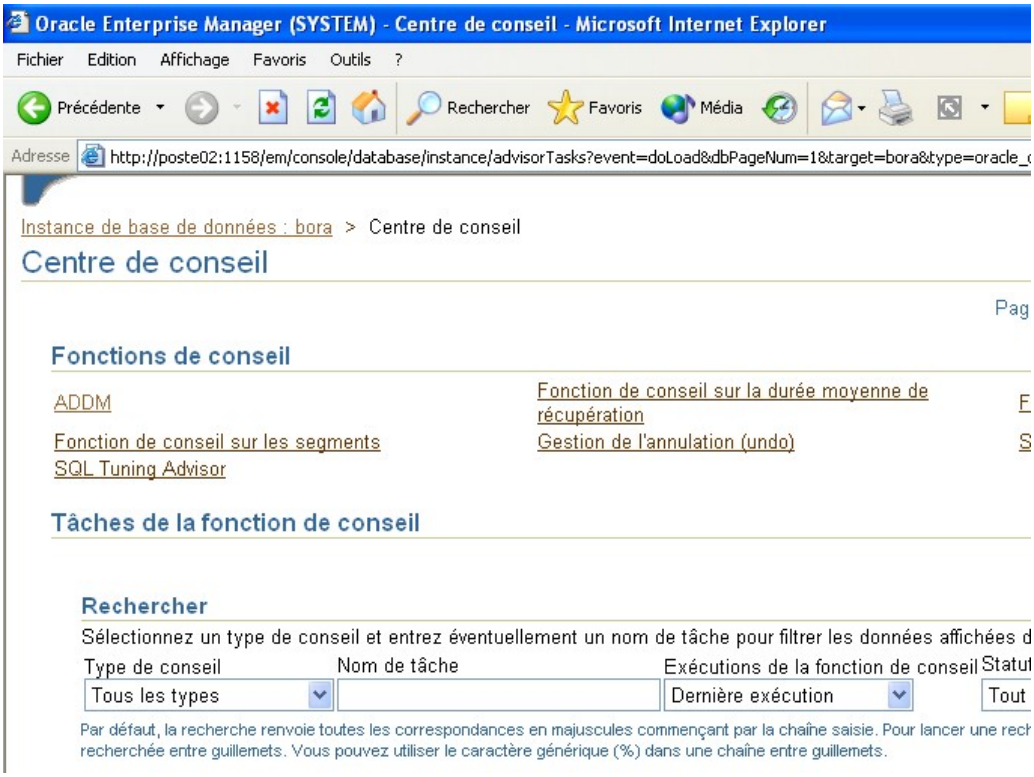
Sur la page « `Automatic Database Diagnostic Monitor (ADDM)` » sont affichés les résultats détaillés de la dernière exécution d'`ADDM`.

Le « `Database Time` » représente la somme de temps actif du système par session dans la base de données pour la période d'analyse.

Un pourcentage spécifique est donné par la colonne « `Impact` » pour chaque diagnostic. Cet impact représente le temps consommé par le problème comparé au temps total utilisé par la base de données pour la période analysée.

Une liste d'icône représente le *snapshot* pour lequel un problème a été identifié. Il est possible de détailler d'autres diagnostics correspondants à d'autres *snapshots*.

Une liste des diagnostics est présentée en bas de page et donne un court résumé de ce qu'`ADDM` a trouvé comme points d'améliorations de performance pour l'instance analysée.



23.7.3 Recommandations d'ADDM

Sur la page « Performance Finding Details », vous trouverez des recommandations pour résoudre le problème correspondant. Les recommandations sont regroupées par catégories comme : Schema, SQL Tuning, DB Configuration, et beaucoup d'autres. La colonne « Benefit (%) » affiche le temps d'exécution gagné par la base si cette recommandation est implémentée.

ADDM prend en compte un certain nombre de changements du système et ces recommandations peuvent inclure :

- Des changements Hardware : l'ajout de CPU ou le changement de la configuration du système d'entrée/sortie.
- La configuration de la base de données : changement des valeurs des paramètres d'initialisation.
- Un changement du schéma : partitionnement Hash d'une table ou d'un index ou l'utilisation de l'ASSM (*Automatic Segment Space Management*) qui correspond à la gestion automatique de la segmentation de l'espace.
- Des changements applicatifs : l'utilisation de l'option cache pour les séquences ou l'utilisation de variables BIND.

L'utilisation d'autres outils de conseil est possible : exécuter le *SQL Tuning Advisor* sur la base pendant une période de pointe d'exécution de requêtes SQL ou l'exécution du *Segment Advisor.*

23.7.4 Nouvelles vues en version 11g pour ADDM

En version 11g, ADDM voit apparaître de nouvelles vues du dictionnaire de données :

- DBA_ADDM_TASK contient la liste des taches ADDM exécutées
- DBA_ADDM_INSTANCES informations sur les instances dans lesquelles ADDM a été exécuté
- DBA_ADDM_FINDING fournit des informations supplémentaires sur ADDM et sont exécution
- DBA_ADDM_FDG_BREAKDOWN fournit des informations sur les performances de chaque instance
- DBA_ADDM_SYSTEM_DIRECTIVES informations concernant les directives (paramètres internes) prédéfinies
- DBA_ADDM_TASK_DIRECTIVES informations concernant les directives (paramètres internes) des tâches prédéfinies.

Le package DBMS_ADDM est optimisé en version 11g.

```
SQL> -- liste des taches ADVISOR effectuées pour une journée donnée
SQL>
SQL> select task_id, to_char(created,'DD/MM/YYYY HH24:MI:SS')  "Date Cree",
recommendation_count "NB recommendations"
2  from dba_advisor_tasks
3  where to_char(created,'DD/MM/YYYY HH24:MI:SS') > '30/07/2007 08:00:00'
4  order by "Date Cree";
```

```
TASK_ID Date Cree           NB recommendations
---------- ------------------- ------------------
      1428 30/07/2007 08:01:02                  0
      1429 30/07/2007 09:00:58                  0
      1430 30/07/2007 10:01:05                  0
      1431 30/07/2007 11:01:02                  0
      1432 30/07/2007 12:00:58                  0
      1433 30/07/2007 13:01:05                  0
      1434 30/07/2007 14:01:01                  0
      1435 30/07/2007 15:00:08                  0
      1436 30/07/2007 16:00:14                  0
      1437 30/07/2007 17:00:16                  0
      1438 30/07/2007 18:00:12                  0
      1439 30/07/2007 19:00:14                  0
      1440 30/07/2007 20:00:20                  0
      1441 30/07/2007 21:00:17                  0
      1442 30/07/2007 22:00:03                  0
      1443 30/07/2007 22:00:23                  0
      1444 30/07/2007 23:00:25                  0
      1445 31/07/2007 00:00:26                  0
      1446 31/07/2007 01:00:28                  0
      1447 31/07/2007 02:00:29                  0
      1448 31/07/2007 03:00:31                  0
      1449 31/07/2007 04:00:32                  0
      1450 31/07/2007 05:00:29                  0

23 ligne(s) sélectionnée(s).
SQL>
SQL> spool OFF
```

23.7.5 Exemple de génération de rapport ADDM

```
SQL> @addmrpt
Current Instance
DB Id    DB Name      Inst Num Instance
----------- ------------ -------- ------------
1885938199 LOWP01              1 LOWP01
Instances in this Workload Repository schema

DB Id     Inst Num DB Name      Instance     Host
------------ -------- ------------ ------------ ------------
•      1885938199        1 LOWP01       LOWP01       aolfrdb-lm01
.webdb.aol.c
om

Using 1885938199 for database Id
Using          1 for instance number

Specify the number of days of snapshots to choose from
Entering the number of days (n) will result in the most recent
(n) days of snapshots being listed.  Pressing <return> without
specifying a number lists all completed snapshots.

Listing the last 3 days of Completed Snapshots
Snap
Instance     DB Name        Snap Id    Snap Started    Level
------------ ------------ --------- ------------------ -----
LOWP01       LOWP01            1350 29 Juil. 2007 00:0     1
1
1351 29 Juil. 2007 01:0     1
0
```

```
1352 29 Juil. 2007 02:0     1
0
-----------------liste des taches-------------------
Snap
Instance     DB Name        Snap Id    Snap Started    Level
------------ ------------ --------- ----------------- -----
LOWP01       LOWP01            1378 30 Juil. 2007 04:0     1
0
1379 30 Juil. 2007 05:0     1
0
1380 30 Juil. 2007 06:0     1
0
1381 30 Juil. 2007 07:0     1
0
1382 30 Juil. 2007 08:0     1
0
1383 30 Juil. 2007 09:0     1
0
1384 30 Juil. 2007 10:0     1
0
1385 30 Juil. 2007 11:0     1
1
1386 30 Juil. 2007 12:0     1
0
1387 30 Juil. 2007 13:0     1
1
1388 30 Juil. 2007 14:0     1
1
1389 30 Juil. 2007 15:0     1
0
1390 30 Juil. 2007 16:0     1
0
Specify the Begin and End Snapshot Ids
Entrez une valeur pour begin_snap : 1382
Begin Snapshot Id specified: 1382
Entrez une valeur pour end_snap : 1384
End   Snapshot Id specified: 1384

Specify the Report Name
The default report file name is addmrpt_1_1382_1384.txt.  To use this name,
press <return> to continue, otherwise enter an alternative.
Entrez une valeur pour report_name : rpt_advisor.log
Using the report name rpt_advisor.log
Running the ADDM analysis on the specified pair of snapshots ...
Generating the ADDM report for this analysis ...
```

Exemple de rapport ADDM

```
DETAILED ADDM REPORT FOR TASK 'TASK_1640' WITH ID 1640

Analysis Period: 06-AOÛT -2007 from 11:00:19 to 15:00:25
Database ID/Instance: 1885938199/1
Database/Instance Names: LOWP01/LOWP01
Host Name: aolfrdb-lm01.webdb.aol.com
Database Version: 10.2.0.1.0
Snapshot Range: from 1553 to 1557
Database Time: 1304 seconds
Average Database Load: ,1 active sessions



FINDING 1: 44% impact (579 seconds)
SQL statements consuming significant database time were found.
RECOMMENDATION 1: SQL Tuning, 24% benefit (310 seconds)
```

```
ACTION: Investigate the SQL statement with SQL_ID "6nbtbvs44xdhj" for
possible performance improvements.
RELEVANT OBJECT: SQL statement with SQL_ID 6nbtbvs44xdhj
insert into EDITO_IMAGE (LAST_MODIFICATION_DATE, CREATED_BY,
CREATION_DATE, LAST_MODIFIED_BY, CONTENT, MIME_TYPE, AUTHOR,
CATEGORY, CHANNEL, COPYRIGHT, CONTENT_DATE, END_DATE, DESCRIPTION,
KEYWORDS, PROMOTION, PURGE_DATE, REFERENCE_ID, SOURCE, START_DATE,
SUB_CATEGORY, ALT, GALLERY_ID, IMAGE_IN_GALLERY_INDEX, IMAGE_TEXT,
THUMBNAIL_ID, ID) values (:1, :2, :3, :4, :5, :6, :7, :8, :9, :10,
:11, :12, :13, :14, :15, :16, :17, :18, :19, :20, :21, :22, :23, :24,
:25, :26)
RATIONALE: SQL statement with SQL_ID "6nbtbvs44xdhj" was executed 94
times and had an average elapsed time of 3.2 seconds.
RATIONALE: Waiting for event "SQL*Net more data from client" in wait
class "Network" accounted for 93% of the database time spent in
processing the SQL statement with SQL_ID "6nbtbvs44xdhj".

---------------------- suite du rapport -------------------------
RECOMMENDATION 3: SQL Tuning, 6,1% benefit (79 seconds)
ACTION: Run SQL Tuning Advisor on the SQL statement with SQL_ID
"aqzkzrwtzzfq8".
RELEVANT OBJECT: SQL statement with SQL_ID aqzkzrwtzzfq8 and
PLAN_HASH 3910361669
select this_.ID as ID7_0_, this_.CHILD_TYPE as CHILD2_7_0_,
this_.CHILD_ID as CHILD3_7_0_, this_.PARENT_TYPE as PARENT4_7_0_,
this_.PARENT_ID as PARENT5_7_0_, this_.RELATION as RELATION7_0_ from
CROSS_REFERENCE this_ where (this_.CHILD_TYPE=:1 and
this_.CHILD_ID=:2) and this_.RELATION=:3
RATIONALE: SQL statement with SQL_ID "aqzkzrwtzzfq8" was executed 75896
times and had an average elapsed time of 0.001 seconds.
RECOMMENDATION 4: SQL Tuning, 5,5% benefit (72 seconds)
ACTION: Run SQL Tuning Advisor on the SQL statement with SQL_ID
"4gnvmykfv138v".
RELEVANT OBJECT: SQL statement with SQL_ID 4gnvmykfv138v and
PLAN_HASH 1107743244
select * from ( select this_.ID as ID0_1_,
this_.LAST_MODIFICATION_DATE as LAST2_0_1_, this_.CREATED_BY as
CREATED1_8_1_, this_.CREATION_DATE as CREATION2_8_1_,
this_.LAST_MODIFIED_BY as LAST3_8_1_, this_.CONTENT as CONTENT10_1_,
this_.MIME_TYPE as MIME2_10_1_, this_.AUTHOR as AUTHOR10_1_,
this_.CATEGORY as CATEGORY10_1_, this_.CHANNEL as CHANNEL10_1_,
this_.COPYRIGHT as COPYRIGHT10_1_, this_.CONTENT_DATE as
CONTENT7_10_1_, this_.END_DATE as END8_10_1_, this_.DESCRIPTION as
DESCRIPT9_10_1_, this_.KEYWORDS as KEYWORDS10_1_, this_.PROMOTION as
PROMOTION10_1_, this_.PURGE_DATE as PURGE12_10_1_, this_.REFERENCE_ID
as REFERENCE13_10_1_, this_.SOURCE as SOURCE10_1_, this_.START_DATE
as START15_10_1_, this_.SUB_CATEGORY as SUB16_10_1_,
this_.AUTHOR_EMAIL as AUTHOR1_25_1_, this_.JOKE_CAT_ID as
JOKE4_25_1_, this_.MODERATION as MODERATION25_1_, this_.TITLE as
TITLE25_1_, jokecatego2_.ID as ID0_0_,
jokecatego2_.LAST_MODIFICATION_DATE as LAST2_0_0_,
jokecatego2_.LOGICAL_ID as LOGICAL1_24_0_, jokecatego2_.NAME as
NAME24_0_ from DIV_JOKE this_ inner join DIV_JOKE_CATEGORY
jokecatego2_ on this_.JOKE_CAT_ID=jokecatego2_.ID where
this_.MODERATION=:1 order by this_.LAST_MODIFICATION_DATE desc )
where rownum <= :2
RATIONALE: SQL statement with SQL_ID "4gnvmykfv138v" was executed 7070
times and had an average elapsed time of 0.01 seconds.
RECOMMENDATION 5: SQL Tuning, 3,9% benefit (51 seconds)
ACTION: Run SQL Tuning Advisor on the SQL statement with SQL_ID
"869fdws5s7k3w".
RELEVANT OBJECT: SQL statement with SQL_ID 869fdws5s7k3w and
PLAN_HASH 3366950133
select * from ( select showform0_.ID as col_0_0_ from DIV_SHOW
showform0_, RATING rating1_ where rating1_.TARGET_ID=showform0_.ID
group by showform0_.ID order by avg(rating1_.RATE) desc,
```

```
count(rating1_.ID) desc ) where rownum <= :1
RATIONALE: SQL statement with SQL_ID "869fdws5s7k3w" was executed 7938
times and had an average elapsed time of 0.0064 seconds.
------------------------------suite du rapport ----------------------------
RECOMMENDATION 3: SQL Tuning, 5,5% benefit (72 seconds)
ACTION: Run SQL Tuning Advisor on the SQL statement with SQL_ID
"4gnvmykfv138v".
RELEVANT OBJECT: SQL statement with SQL_ID 4gnvmykfv138v and
PLAN_HASH 1107743244
select * from ( select this_.ID as ID0_1_,
this_.LAST_MODIFICATION_DATE as LAST2_0_1_, this_.CREATED_BY as
CREATED1_8_1_, this_.CREATION_DATE as CREATION2_8_1_,
this_.LAST_MODIFIED_BY as LAST3_8_1_, this_.CONTENT as CONTENT10_1_,
this_.MIME_TYPE as MIME2_10_1_, this_.AUTHOR as AUTHOR10_1_,
this_.CATEGORY as CATEGORY10_1_, this_.CHANNEL as CHANNEL10_1_,
this_.COPYRIGHT as COPYRIGHT10_1_, this_.CONTENT_DATE as
CONTENT7_10_1_, this_.END_DATE as END8_10_1_, this_.DESCRIPTION as
DESCRIPT9_10_1_, this_.KEYWORDS as KEYWORDS10_1_, this_.PROMOTION as
PROMOTION10_1_, this_.PURGE_DATE as PURGE12_10_1_, this_.REFERENCE_ID
as REFERENCE13_10_1_, this_.SOURCE as SOURCE10_1_, this_.START_DATE
as START15_10_1_, this_.SUB_CATEGORY as SUB16_10_1_,
this_.AUTHOR_EMAIL as AUTHOR1_25_1_, this_.JOKE_CAT_ID as
JOKE4_25_1_, this_.MODERATION as MODERATION25_1_, this_.TITLE as
TITLE25_1_, jokecatego2_.ID as ID0_0_,
jokecatego2_.LAST_MODIFICATION_DATE as LAST2_0_0_,
jokecatego2_.LOGICAL_ID as LOGICAL1_24_0_, jokecatego2_.NAME as
NAME24_0_ from DIV_JOKE this_ inner join DIV_JOKE_CATEGORY
jokecatego2_ on this_.JOKE_CAT_ID=jokecatego2_.ID where
this_.MODERATION=:1 order by this_.LAST_MODIFICATION_DATE desc )
where rownum <= :2
RATIONALE: SQL statement with SQL_ID "4gnvmykfv138v" was executed 7070
times and had an average elapsed time of 0.01 seconds.
RATIONALE: Average CPU used per execution was 0.01 seconds.
RECOMMENDATION 4: SQL Tuning, 3,9% benefit (51 seconds)
ACTION: Run SQL Tuning Advisor on the SQL statement with SQL_ID
"869fdws5s7k3w".
RELEVANT OBJECT: SQL statement with SQL_ID 869fdws5s7k3w and
PLAN_HASH 3366950133
select * from ( select showform0_.ID as col_0_0_ from DIV_SHOW
showform0_, RATING rating1_ where rating1_.TARGET_ID=showform0_.ID
group by showform0_.ID order by avg(rating1_.RATE) desc,
count(rating1_.ID) desc ) where rownum <= :1
RATIONALE: SQL statement with SQL_ID "869fdws5s7k3w" was executed 7938
times and had an average elapsed time of 0.0064 seconds.
RATIONALE: Average CPU used per execution was 0.0064 seconds.
RECOMMENDATION 5: SQL Tuning, 3,5% benefit (46 seconds)
ACTION: Run SQL Tuning Advisor on the SQL statement with SQL_ID
"1w27xdpth04w3".
RELEVANT OBJECT: SQL statement with SQL_ID 1w27xdpth04w3 and
PLAN_HASH 979632492
select * from ( select this_.ID as ID0_0_,
this_.LAST_MODIFICATION_DATE as LAST2_0_0_, this_.CREATION_DATE as
CREATION1_1_0_, this_.HIDDEN as HIDDEN3_0_, this_.MODERATED as
MODERATED3_0_, this_.TEXT_CONTENT as TEXT3_3_0_, this_.USER_ID as
USER4_3_0_, this_.TARGET_TYPE as TARGET5_3_0_, this_.TARGET_ID as
TARGET6_3_0_ from REVIEW this_ where this_.TARGET_ID in (select
this0__.ID as y0_ from DIV_MOVIE this0__) and this_.HIDDEN=:1 order
by this_.CREATION_DATE desc ) where rownum <= :2
RATIONALE: SQL statement with SQL_ID "1w27xdpth04w3" was executed 7255
times and had an average elapsed time of 0.0063 seconds.
RATIONALE: Average CPU used per execution was 0.0063 seconds.
FINDING 4: 4,9% impact (64 seconds)
Wait event "SQL*Net more data to client" in wait class "Network" was consuming
significant database time.

RECOMMENDATION 1: Application Analysis, 4,9% benefit (64 seconds)
```

```
ACTION: Investigate the cause for high "SQL*Net more data to client"
waits. Refer to Oracle's "Database Reference" for the description of
this wait event.

RECOMMENDATION 2: Application Analysis, 4,9% benefit (64 seconds)
ACTION: Investigate the cause for high "SQL*Net more data to client"
waits in Service "SYS$USERS".
SYMPTOMS THAT LED TO THE FINDING:
SYMPTOM: Wait class "Network" was consuming significant database time.
(38% impact [491 seconds])
FINDING 5: 3,2% impact (42 seconds)
Soft parsing of SQL statements was consuming significant database time.
RECOMMENDATION 1: Application Analysis, 3,2% benefit (42 seconds)
ACTION: Investigate application logic to keep open the frequently used
cursors. Note that cursors are closed by both cursor close calls and
session disconnects.

RECOMMENDATION 2: DB Configuration, 3,2% benefit (42 seconds)
ACTION: Consider increasing the maximum number of open cursors a session
can have by increasing the value of parameter "open_cursors".
ACTION: Consider increasing the session cursor cache size by increasing
the value of parameter "session_cached_cursors".
RATIONALE: The value of parameter "open_cursors" was "700" during the
analysis period.
RATIONALE: The value of parameter "session_cached_cursors" was "20"
during the analysis period.


ADDITIONAL INFORMATION

Wait class "Application" was not consuming significant database time.
Wait class "Commit" was not consuming significant database time.
Wait class "Concurrency" was not consuming significant database time.
Wait class "Configuration" was not consuming significant database time.
Wait class "User I/O" was not consuming significant database time.
Session connect and disconnect calls were not consuming significant database
time.
Hard parsing of SQL statements was not consuming significant database time.

The analysis of I/O performance is based on the default assumption that the
average read time for one database block is 10000 micro-seconds.

An explanation of the terminology used in this report is available when you
run the report with the 'ALL' level of detail.
```

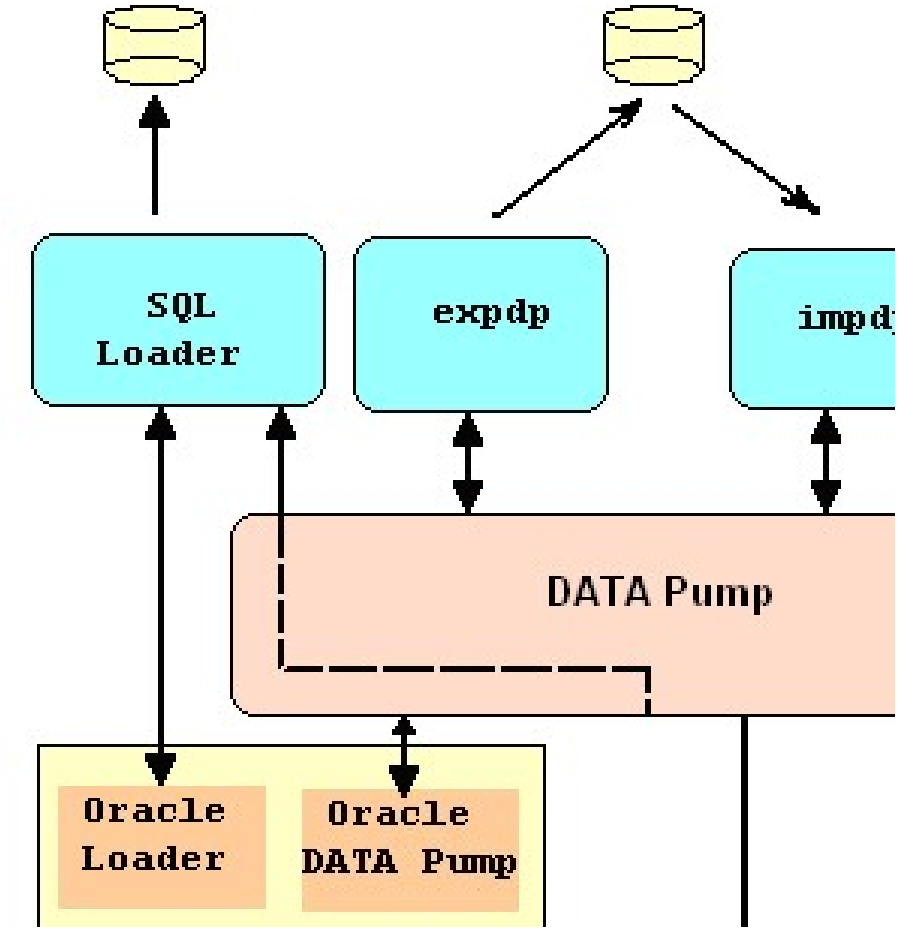
24 Présentation de l'utilitaire DATA Pump

`Data pump` est un nouvel outil qui permet de charger ou décharger des données à grande vitesse.

Il peut être appelé *via* le package PL/SQL, `DBMS_DATAPUMP`.

Oracle 10g introduit des nouveaux outils :

- De nouvelles commandes pour Export et Import appelées respectivement `EXPDP` et `IMPDP`.
- Une interface Web d'Import et d'Export accessible à partir du Database Control.



Les composants du DATA Pump sont :

- Direct Path API (DPAPI) : Oracle 10g supporte une interface de chemin directe qui minimise la conversion et le processus des données au moment du chargement et du déchargement des données.
- Services des Tables Externes : DATA Pump utilise le nouveau Driver d'accès ORACLE_DATAPUMP qui fournit des accès en lecture et écriture aux tables externes à des fichiers qui contiennent des chaînes de caractères binaires.
- Le package DBMS_METADATA est utilisé par les processus d'exécution pour tout chargement ou déchargement des métas-données (CLOB, BLOB). Les définitions des objets de la base sont stockées en utilisant XML plutôt que SQL.
- Le package DBMS_DATAPUMP inclut l'API pour les fonctions d'Import et Export rapides, ainsi que le déplacement de données de masse et des méta-données.
- Le client SQL*LOADER a été intégré avec les tables externes et fournit la migration automatique des fichiers de contrôle du SQL*LOAD vers les paramètres d'accès aux tables externes.
- Les clients EXPDP et IMPDP sont des clients légers qui font appel au package DBMS_DATAPUMP pour initialiser et gérer les opérations du DATA Pump. Même s'ils introduisent de nouvelles fonctionnalités, ils restent compatibles avec les clients import et export antérieurs qui sont toujours disponibles.
- Les applications comme Database Control, la réplication, les tablespaces transportables et les applications utilisateurs bénéficient de cette infrastructure. SQL*Plus peut aussi être utilisé comme un client du package DBMS_DATAPUMP pour des requêtes simples sur l'état des opérations en cours.

24.1 Opérations d'IMPORT et d'EXPORT du DATA Pump

L'import et l'export du DATA Pump sont de nouveaux outils propres à Oracle database 10G.

Ce sont des outils différents des outils d'import et d'export classiques même si les commandes sont similaires.

- DATA Pump Export est un outil pour décharger des données et des métadonnées dans des fichiers du système d'exploitation appelé fichiers de dump.
- DATA Pump import est utilisé pour charger des données et métadonnées qui sont stockées dans un fichier de dump vers une base cible.

Le mode de chargement ou de déchargement des outils DATA Pump export et import, est spécifié sur la ligne de commande en utilisant le paramètre approprié. Les divers modes disponibles sont listés ci-dessous, **ce sont les mêmes que ceux des utilitaires d'import et d'export des versions antérieures :**

- **FULL** **toute la base (sauf le dictionnaire de données)**
- **SCHEMA** **tous les objets d'un schéma**
- **TABLE** **une ou plusieurs tables**
- **TABLESPACE** **tous les objet contenus dans un tablespace**
- **TABLESPACE Transportable** **transport d'un tablespace entre é bases**

Le cœur de chaque opération DATA Pump est représenté par la table : MASTER TABLE (MT) qui est une table créée dans le schéma de l'utilisateur qui exécute un « job » DATA Pump.

La table MT est construite pendant l'exécution d'un job d'export. Par contre, le chargement de la table MT dans le schéma utilisateur en cours est la première action d'une opération d'import *via* un script, qui est utilisée pour créer la séquence de création de tous les objets importés.

La table MT représente l'élément clé pour pouvoir redémarrer l'outil DATA Pump dans le cas de l'arrêt planifié ou non planifié du job.

La table MT est supprimée quand le DATA Pump finit normalement.

24.2 Avantages de l'export et de l'import DATA Pump:

- DATA Pump décide automatiquement quelle est la méthode d'accès aux données à utiliser ; cette méthode peut être soit « Direct Pass » soit « Externals tables ».
- La possibilité de se connecter et se déconnecter à des jobs de longue durée sans affecter le job lui-même, vous permet de superviser ceux-ci à partir des différentes localisations pendant leur temps d'exécution.
- Les paramètres EXCLUDE, INCLUDE et CONTENT sont utilisés pour sélectionner des objets à des niveaux très fins.
- Le paramètre PARALLEL peut être utilisé pour spécifier le nombre maximal de « Threads » du serveur sur lequel s'exécute le job d'export.
- Le paramètre ESTIMATE_ONLY permet de savoir combien d'espace pourrait être consommé par le job d'export (sans exécuter l'export).
- Le mode réseau permet d'exporter des objets d'une base distante dans un fichier de dump. Ceci peut être effectué en utilisant un *lien de la base de données* distante vers le système source.
- Pendant l'import les noms des fichiers de données cibles, les noms des schémas et les noms des tablespaces de la base importée peuvent être changés.

Une fois que le job est déclenché, plusieurs « clients » peuvent se connecter et se déconnecter au job.

Le processus d'arrière plan « *Master Control Process* » (MCP) contrôle l'exécution et la séquence d'un job DATA Pump pendant son exécution.

Une fois le job en exécution, la tâche principale du processus d'arrière plan est de desservir les requêtes du « client ».

Si le « client » se déconnecte, le processus d'arrière plan s'arrête.

A la réception d'une requête START JOB, le MCP crée un nombre de processus de travail qui dépend de la valeur du paramètre PARALLEL. Le nom d'un processus de travail suit le format DWnn (il charge et décharge les données).

Si la méthode d'accès aux données est de type EXTERNAL TABLE PATH pour le chargement et le déchargement des données, le processus de travail DWnn coordonne un nombre parallèle des serveurs d'exécution en fonction du nombre de chargements ou de déchargements définis. Ceci permet le chargement et le déchargement intra-partition.

Remarques

Vous pouvez vous connecter à un job actif afin de l'arrêter, de changer son parallélisme ou bien de superviser son état d'avancement.

Vous pouvez utiliser l'information de la MT, dérivée du JOB_NAME pour redémarrer un job arrêté ou supprimer toutes les MT qui ne sont plus utiles.

Les jobs DATA Pump maintiennent une entrée dans la vue V$SESSION_LONGOPS sur les performances dynamiques.

Si aucun nom de job n'est spécifié, DATA Pump utilise le schéma du job afin de le générer automatiquement.

Ce nom utilise le format suivant : <USER>_<OPERATION>_<MODE>_%N, nom qui dépend du type d'opération exécutée et de son domaine.

Certains paramètres peuvent entrer en conflit avec d'autres. Par exemple, des valeurs du paramètre TABLES peuvent entrer en conflit avec le paramètre OWNER :

```
OWNER=ORCL
TABLES=opdef.client
```

Les erreurs n'arrêtent pas l'export, elles sont simplement signalées.

Les possibilités d'export incrémentaux (paramètre INCTYPE) sont obsolètes et conservées pour des raisons de compatibilité.

- ⇨ Utiliser les fonctionnalités similaires du Recovery Manager

Pour visualiser l'ensemble des paramètres d'export en ligne, utilisez la commande.

Exp help=y

24.3 Le mode intéractif du DATA Pump

DATA Pump peut s'exécuter en mode commande (sous dos ou sous unix) en utilisant les commandes expdp ou impdp, avec ou sans fichier de paramètre.

DATA Pump peut s'exécuter également en interactif, ce qui permet de sortir de l'affichage écran du travail en cours qui continue en arrière plan.

- ⇨ Ainsi en tapant les touches <CTRL+C>, pendant l'exécution du DATA Pump, l'affichage à l'écran s'arrête, mais le travail continue en arrière plan.

24.3.1 Commandes du mode interactif

Dans ce mode il existe un ensemble de commandes qui permettent de gérer les jobs DATA Pump en cours d'exécution.

- CONTINUE_CLIENT = redémarre l'affichage de l'avancement du job DATA Pump
- EXIT_CLIENT = quitte l'outil en laissant le travail continuer
- KILL_JOB = arrête et supprime le job
- START_JOB = redémarre le job
- STOP_JOB = arrête le travail sans le supprimer. Sans option, le job termine la tache en cours avant de s'arreter. Pour arrêter le job immédiatement, il faut utiliser l'option IMMEDIATE.

La vue DATAPUMP_JOBS permet de visualiser les travaux DATA Pump en cours !

24.4 Méthode d'extraction des données avant et aprés *data pump*

Dans le chemin conventionnel, l'export avant DATA Pump utilise des `SELECT` pour extraire les données, avec le mécanisme habituel de lecture.

Dans le chemin direct, certaines étapes du mécanisme de lecture sont éliminées, ce qui permet d'améliorer les performances.

Pour pouvoir utiliser le chemin direct, il faut que le jeu de caractères de la session qui réalise l'export soit le même que celui utilisé dans la base (clause `CHARACTER SET` du `CREATE DATABASE`).

La variable d'environnement `NLS_LANG` permet de respecter cette contrainte (par exemple `NLS_LANG = AMERICAN_AMERICA.WE8ISO8859P1`).

Du point de vue des performances, il est conseillé de spécifier un paramètre :

- `RECORDLENGTH` multiple du `DB_BLOCK_SIZE` de la base.

24.4.1 Méthode direct path (chemin direct) du *data pump*

`DATA Pump` permet 2 méthodes d'accès aux données :

- Chemin Direct Path en utilisant l'API direct-path
- External tables

`DATA Pump` sélectionne automatiquement la méthode d'accès la plus adaptée à chaque table.

`DATA Pump` utilise le chemin « Direct Path » quand la structure de la table le permet et quand il est demandé d'avoir un accès rapide aux données.

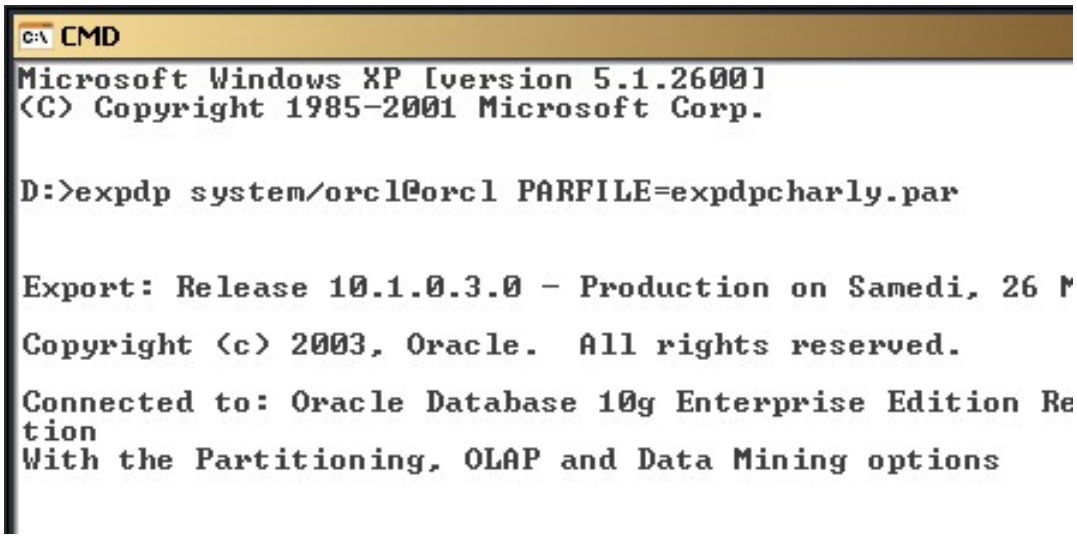
Si une des conditions énumérée ci-dessous est remplie, ou encore, si une table contient des colonnes cryptées, ou si les tables chargées sont partitionnées différemment au moment du chargement et du déchargement, le `DATA Pump` utilisera de préférence les tables externes plutôt que le chemin direct pour déplacer les données.

Des données chargées avec une méthode peuvent être déchargées en utilisant l'autre méthode.

24.4.2 Données conduisant un accès utilisant des tables externes :

- Tables avec contrôle d'accès fin pour le mode `INSERT` et `SELECT`
- Colonnes de type `LOB`
- Tables Clusterisées
- Tables avec triggers actifs
- Table partitionnées ou avec index globaux sur une seule partition
- `BFILE` ou colonnes contenant des tuples opaques (binaire)
- Contraintes d'intégrité référentielles
- Colonnes de type `VARRAY` avec un type opaque encapsulé

L'appel aux outils `DATA Pump` se fait sous le système d'exploitation, en précisant l'utilisateur de connexion à la base de données et un fichier de paramètres spécifié par le mot clé : `PARFILE`.



25 Export/Import Data Pump

25.1 Fichiers supportés par les outils *DATA Pump*

Il y a 3 types de fichiers gérés par les outils DATA Pump :

- Les fichiers de « dump » qui contiennent les données et les métadonnées à déplacer
- Les fichiers de « log » qui tracent les messages associés à chaque opération
- Les fichiers « SQL » qui enregistrent le résultat de chaque opération

DATA Pump autorise un accès aux fichiers à travers l'utilisation de chemins d'accès relatifs d'Oracle : les objets DIRECTORY. Les chemins absolus ne sont pas supportés pour des raisons de sécurité.

Enchaînement utilisé par les clients DATA Pump pour localiser ces fichiers :

- Les objets DIRECTORY par fichier peuvent être spécifiés pour chaque fichier « dump », de « log » et « SQL ». Si ils sont spécifiés, ils sont séparés du nom du fichier par « : »
- Les clients d'export/import du DATA Pump fournissent un paramètre DIRECTORY qui spécifie le nom d'un objet DIRECTORY. Ces objets DIRECTORY décrivent la localisation des fichiers.
- Une variable d'environnement DATA_PUMP_DIR peut être définie pour spécifier le nom de l'objet directory (*variable de chemin par défaut*). Les clients DATA Pump vont chercher cette variable d'environnement si aucun objet DIRECTORY n'est défini d'une façon explicite.

Il faut avoir les privilèges d'accès aux répertoires pour pouvoir accéder aux fichiers afin de pouvoir exécuter l'opération de chargement ou de déchargement.

Pour l'export, l'accès de type WRITE est nécessaire pour tous les fichiers.

Pour l'import, l'accès de type READ est nécessaire pour les fichiers de « dump » et l'accès de type WRITE est nécessaire pour les fichiers « log » et « SQL ».

```
•      création de la directory dir_charly pour le DATAPump

create or replace DIRECTORY dir_charly
as 'D:\datapump\'
/
```

```
•      APPEL DE LA DIRECTORY DANS LE FICHIER DE PARAMETRES
schemas = CHARLY
directory = dir_charly
dumpfile = ExpdpCharly.dmp
logfile = ExpCharly.log
```

Le paramètre DUMPFILE spécifie le nom (répertoires compris) des fichiers de « dump » dans lesquels sont situés les fichiers sur disque.

Le nom du fichier peut contenir la variable de substitution « %U » ce qui signifie que plusieurs fichiers pourront être générés. La variable « %U » sera traduit dans les noms de fichiers résultants, par un numéro à 2 chiffres incrémenté de 1 en 1 commençant à « 01 ».

Si le paramètre DUMPFILE n'est pas spécifié, le nom de fichier « expdat.dmp » est utilisé.

Ce sont des fichiers AUTOEXTENSIBLES, sauf si le paramètre FILESIZE est spécifié.

Chaque fichier aura alors une taille à la valeur de FILESIZE et sera non extensible.

S'il n'y a pas assez d'espace et si le format « %U » est défini, un nouveau fichier sera créé automatiquement à la valeur de FILESIZE ; sinon le client recevra un message en lui demandant d'ajouter un nouveau fichier.

Si le nom du fichier est généré avec « %U » le nombre de fichiers créés dés le début est égal à la valeur du paramètre PARALLEL.

25.2 Paramètres le l'export et de l'import DATA Pump

Ci-dessous sont présentés un ensemble de paramètres utilisés avec DATA Pump. Le type de données manipulées par DATA Pump sont :

- ALL : les données et les métadonnées
- DATA_ONLY : les données uniquement
- METADATA_ONLY : les métadonnées uniquement.

La liste complète des paramètres est expliquée dans la documentation « *Oracle Database Utilities* ».

25.2.1 Paramètres communs

Dans le fichier de paramètres, le caractère # en début de ligne met la ligne en commentaire.

ATTACHE[=schema.]nom_travail]

- Permet d'attacher sa session à un job DATA Pump en cours. Pour attacher un job d'un autre schema, il faut avoir le privilège EXP_FULL_DATABASE ou IMP_FULL_DATABASE. Si aucun nom n'est spécifié, la session est attachée au travail en cours dans le schéma courant. Si ce paramètre est utilisé, aucun autre paramètre ne peut être spécifié.

JOB_NAME=nom_job

- Permet de donner un nom au job DATA Pump execute. S'il n'est pas précisé le nom du job est SYS_<opération>_<mode>_<nn>.

CONTENT={ALL|DATA_ONLY|METADATA_ONLY}

- Permet de définir le contenu de l'export ou de l'import

DIRECTORY=objet_directory

- Permet de définir un répertoire appelé DIRECTORY déclaré dans la base de données (par la commande CREATE DIRECTORY …) Les fichiers du DATA Pump iront dans ce répertoire. DATA_PUMP_DIR définit la directory par défaut.

DUMPFILE=nom_fichier_dump

- Défini le num du fichier dump en sortie. Il est possible de parallèliser l'export ou l'import.

LOGFILE=nom_fichier_log

- Permet de préciser le nom du fichier de log (fichier journal). Par défaut ces fichiers sont nommés export.log ou import.log. Si le paramètre NOLOGFILE est positionné à « y » alors aucun fichier de log ne sera généré.

NOLOGFILE={y,n}

- Positionné à « y » ce paramètre précise qu'aucun fichier de log ne sera généré.

PARFILE=nom_fichier_paramêtres

- Permet de préciser le nom du fichier de paramêtre utilisé pour l'export ou l'import. Ce fichier contient les paramêtres appliqués lors de l'export ou de l'import. Ce fichier de paramêtre doit être présent sur le serveur qui effectue l'export ou l'import.

FULL={y|n}

- Permet de préciser s'il s'agit d'un export ou d'un import complet ou non.

SCHEMAS=nom_schema, …

- Permet de préciser le nom des schémas à exporter ou à importer

TABLES=[schema.]nom_table[:partition] [, …]

- Permet de préciser le nom des tables à exporter ou à importer, ainsi que des partition de tables si besoin.

QUERY=[schema.][nom_table :]clause_where

- Permet de filtrer les données à exporter ou à importer

TABLESPACES=nom_tablespace, …

- Permet de faire un export ou un import de niveau tablespace. Il permet de préciser plusieurs tablespaces.

TRANSPORT_FULL_CHECK={y,n}

- Si ce parameter est positionné à <y>, DATA Pump vérifie les dépendances entre les objets transportés à l'intérieur des tablespaces transportés.

TRANSPORT_TABLESPACES=nom_tablespace, …

- Permet de faire un export ou un import de niveau transport de tablespace. Il permet de préciser plusieurs tablespaces.

NETWORK_LINK=nom_database_link

- Précise le nom d'un database_link à utiliser pour l'export ou l'import.

EXCLUDE=type_objet[:filtre] [,…]

- Permet d'exclure des objets pour l'export ou l'import.

INCLUDE=type_objet[:filtre] [,…]

- Permet d'inclure des objets pour l'export ou l'import.

25.2.2 Paramètres de l'EXPORT DATA Pump

COMPRESSION={ALL|DATA_ONLY|METADATA_ONLY|NONE}

- Active la compression des données ou des « méta-data » du fichier d'export.

ESTIMATE_ONLY={y|n}

- Permet de vérifier l'espace que l'export va occuper sans faire l'export réellement.

25.2.3 Paramètres de l'IMPORT DATA Pump

SQLFILE=nom_fichier_SQL

- Précise le nom du fichier SQL généré au moment de l'import, contenant les ordres DDL correspondant à l'import sans réellement réaliser l'import.

REMAP_SCHEMA=nom_schema_source :nom_schema_cible

- Permet de préciser le nom du schéma qui a été exporté et le nom du schéma dans lequel se fera l'import. Si le schéma n'existe pas il sera créé avec les même préivilèges que celui de l'export, par contre le mot de passe devra être modifié avant de pouvoir se connecter.

REMAP_TABLESPACE=nom_tablespace_source :nom_tablespace_cible

- Permet de préciser le nom du tablespace cible lorsque l'on veut changer le tablespace d'origine. Plusieurs paramêtres peuvent être spécifiés pour effectuer plusieurs changement de tablespace.

REMAP_DATAFILE=nom_datafile_source :nom_datafile_cible

- Permet de préciser les chemins des fichiers de données des tablespace cible lorsque l'on veut changer le tablespace d'origine et donc de nom de fichier cible avec des chemins différentes. Plusieurs paramêtres peuvent être spécifiés pour effectuer plusieurs changement.

TABLE_EXISTS_ACTION=[SKIP|APPEND|TRUNCATE|REPLACE]

- Permet de préciser l'action à effectuer lorsque la table rencontrée au moment de l'import existe déjà :
  SKIP = ne rien faire et passer à l'objet suivant (non autoriser si CONTENT=data_only)
  APPEND = ajoute les données à la fin du contenu de la table (valeur par défaut si CONTENT=data_only)
  TRUNCATE = vide la table avant de charger les données.
  REPLACE=supprime la table puis la recrée, avant de charger les données. (non autoriser si CONTENT=data_only).

TRANSPORT_DATAFILES=nom_fichiers, …

- Permet de préciser l'emplacement des fichiers de données lors d'un transport de tablespaceLes fichiers de données (*datafiles*) doivent être recopier au préalable.

25.3 Filtrer les données à exporter

Le job `DATA Pump` peut inclure ou exclure pratiquement tous types d'objets grâce au paramètre `EXCLUDE`.

Ce paramètre `EXCLUDE` est spécifié dans un fichier SQL défini par le paramètre `SQLFILE`.

```
EXCLUDE=object_type [ :  « expression »]
```

Les 3 lignes du fichier de paramètre vont exclure toutes les vues, les packages et les index dont le nom commence par « EMP »

```
EXCLUDE=VIEW
EXCLUDE=PACKAGE
EXCLUDE=INDEX : "like 'EMP'"
```

Le paramètre INCLUDE permet d'inclure seulement les types d'objets spécifiés et les objets spécifiés pour l'opération.

```
INCLUDE=object_type [ :  « expression »]
```

Le paramètre CONTENT permet de sélectionner pour l'opération courante les métadonnées, les données ou les deux :

```
CONTENT= ALL | METADATA_ONLY | DATA_ONLY
```

Le paramètre QUERY fonctionne d'une façon similaire à l'utilitaire d'export antérieur avec 2 améliorations principales :

- Il peut être préfixé par un nom de table, pour être appliqué seulement sur cette table.
- Il peut être utilisé pour un import.

```
QUERY= [SCHEMA.][nom_table :] « QUERY »
```

```
QUERY=charly.employe:"where salaire in (1000,2000)
And nom_emp like 'Prof%'
Order by nom_emp"
```

Les paramètres EXCLUDE et INCLUDE sont mutuellement exclusifs.
Ils ne peuvent pas être utilisés si le paramètre CONTENT=DATA_ONLY, est spécifié.

Comme la métadonnée de l'objet est stockée sous un format XML dans un fichier de « dump », il sera facile d'appliquer une transformation quand une DDL est définie pendant l'import.

L'import DATA Pump permet plusieurs transformations :

- REMAP_DATAFILE est utile pour déplacer des bases de données vers différentes plateformes qui ont des systèmes de gestion de fichiers différentes.
- REMAP_TABLESPACE permet de déplacer des objets d'un tablespace dans un autre.
- REMAP_SHEMA fournit la fonctionnalité antérieure du FROMUSER/TOUSER, ceci afin de changer le propriétaire d'un objet.

```
•      EXEMPLE de Fichier PARFILE pour un import

directory = dir_charly
remap_schema = CHARLY:OPDEF
dumpFILE = ExpdpCharly.dmp
LOGfile = Imp.log
```

En utilisant le paramètre `TRANSFORM`, il peut être possible de ne pas générer les clauses de stockage dans la `DDL`. Ceci est utile si les caractéristiques de stockage de l'instance cible sont très différentes de celles de l'instance source.

Les vues du dictionnaire de données permettant de visualiser la liste des types d'objets pouvant être exporté dans un paramètre EXCLUDE ou INCLUDE sont :

- DATABASE_EXPORT_OBJECTS et
- SCHEMA_EXPORT_OBJECTS et
- TABLE_EXPORT_OBJECTS

25.4 Exemples d'export et d'import DATA Pump

25.4.1 Estimation de la taille de l'Export

Ce script permet de définir la taille que fera le fichier d'export.

```
> expdp hr/hr DIRECTORY=dpump_dir1 ESTIMATE_ONLY=y TABLES=employees,
departments, locations LOGFILE=estimate.log
```

25.4.2 Exports Parallélisés

C'est un export de base complet qui a 4 processus de travail parallèles. Les fichiers de dump sont créés dans des répertoires indiqués par les objets directory `DATADIR1, DATADIR2, DATADIR3, DATADIR4`.

Chaque fichier a une taille de 2 giga octets et 4 fichiers au moins seront créés.

Le nom du job et de la MT est le nom par défaut `SYSTEM_EXPORT_FULL_01`.

```
Expdp system/bora full = y
Parallel = 4
Dumpfile = DATADIR1:full1%U.dat,
DATADIR2:full2%U.dat,
DATADIR3:full3%U.dat,
DATADIR4:full4%U.dat,
```

Exemple 2 : EXPORT Full en Parallel

```
> expdp hr/hr FULL=y DUMPFILE=dpump_dir1:full1%U.dmp, dpump_dir2:full2%U.dmp
FILESIZE=2G PARALLEL=3 LOGFILE=dpump_dir1:expfull.log JOB_NAME=expfull
```

25.4.3 Import Parallélisé

Cet exemple est un exemple d'import complet du fichier de dump créé avec le 1er exemple.

Le fichier de dump a été envoyé vers un périphérique de stockage réseau spécifié par l'objet directory.

Le paramètre `NET_STORAGE_1.FULL=Y` n'est pas nécessaire car l'import par défaut est celui de l'import complet du fichier de dump. 4 lots parallèles de chargement sont créés.

Le job et la MT ont comme nom par défaut `SYSTEM_IMPORT_FULL_01`.

```
Impdp system/bora
Directory = NET_STORAGE_1
Parallel = 4
Dumpfile = full1%U.dat,
full2%U.dat,
Full3%U.dat,
full2%U.dat
sqlfile = gen08.sql
```

25.4.4 Export de schéma

Des procédures, des packages, des types et des vues dont le nom commence avec `PRODUCT` seront exportés à partir des schémas `CHARLY` et `OPDEF`.

Le fichier de dump « `schema_charly_opdef.dat` » est créé dans le répertoire indiqué dans l'objet directory `USR_DATA`.

Comme l'utilisateur `SYSTEM` détient le rôle `EXPORT_FULL_DATABASE`, il peut spécifier plusieurs schémas.

Les définitions des schémas et les autorisations concernant les privilèges *system* ne sont pas exportés (alors qu'elles devraient l'être normalement) car elles ne sont pas précisées, d'une manière explicite dans la clause `INCLUDE`.

```
Expdp system/bora schemas = CHARLY,OPDEF
Directory = USR_DATA
Dumpfile  = schema_charly_opdef.dat
include = function
include = procedure
include = package
include = view:"like 'PRODUCT%'"
```

Exemple 2– export de données déchargées des lignes de Tables

```
> expdp hr/hr PARFILE=exp.par
```

Le fichier de paramêtres contient :

```
DIRECTORY=dpump_dir1
DUMPFILE=dataonly.dmp
CONTENT=DATA_ONLY
EXCLUDE=TABLE:"IN ('COUNTRIES', 'LOCATIONS', 'REGIONS')"
QUERY=employees:"WHERE department_id !=50 ORDER BY employee_id"
```

25.4.5 Import de schéma

Cet exemple d'import montre comment vous pouvez générer un script SQL à partir d'un fichier d'export de dump qui contient toutes les définitions DDL que l'import exécutera en s'appuyant sur les autres valeurs des autres paramètres.

Le SQL ne sera pas exécuté et le système cible restera inchangé.

```
Impdp system/bora directory = USR_DATA
Dumpfile = schema_CHARLY_OPDEF.dat
Sqlfile = schema_CHARLY_OPDEF.sql
```

25.5 Remarques et modes opératoires

25.5.1 Export et jeux de caractères

DATA Pump écrit le dump dans le jeu de caractère de la base de données.

Des problèmes peuvent se produire si les jeux de caractères utilisés ne sont pas les mêmes entre la base d'où provient l'export et la base dans laquelle on fait l'import. De même des problèmes peuvent se produire si le jeu de caractère entre la base de données et le jeu de caractère du client ou de l'environnement qui fait l'export ne sont pas compatibles (typiquement, perte des caractères accentués).

Une variable d'environnement NLS_LANG correctement positionnée dans l'environnement qui lance l'outil et ouvre la session d'export permet de remédier à ce problème.

L'import peut provoquer une double conversion de jeu de caractères :

- Jeu de caractères du fichier d'import (variable d'environnement NLS_LANG lors de l'export)
  Jeu de caractères de la session qui effectue l'import (variable d'environnement NLS_LANG lors de l'import)
  Jeu de caractères de la base

Historiquement, les outils d'export/import peuvent être utilisés pour réorganiser le stockage de tout ou partie d'une base.

Ce sont toujours les bons outils pour reconstruire une base en changeant la taille de bloc, ou pour changer le jeu de caractère de la base de données.

25.5.2 Remarques sur les dépendances entre les objets

Lors de la récupération de tout ou partie d'un schéma (ou d'une base), si les objets n'existent pas dans la base cible et si le fichier d'export est importé tel quel (sans restructuration du stockage notamment), il n'y a pas de difficulté particulière car Oracle importe les objets dans un ordre « intelligent ».

Le seul problème potentiel concerne l'ordre d'import des vues et des objets stockés vis à vis des dépendances entre les objets ; si un objet est importé avant un autre objet dont il dépend, il est marqué `INVALID` (colonne `STATUS` de la vue `DBA_OBJECTS`) et il doit être recompilé.

Ce problème n'est pas grave, car la recompilation est automatique à la première utilisation et elle ne devrait pas échouer si tous les objets sont présents.

Pour anticiper et éviter tout problème, il est conseillé de vérifier s'il existe des objets invalides et de les recompiler soi-même avec la syntaxe :

```
ALTER { VIEW | PROCEDURE | FUNCTION | PACKAGE | PACKAGE BODY |
TRIGGER } nom_objet COMPILE
```

La présence de contraintes, d'index ou de triggers déjà existant dans la base cible peut poser plusieurs problèmes, notamment sur les performances et le risque d'avoir des données rejetées.

En ce qui concerne le risque de données rejetées, le problème peut ne pas venir des données proprement dites mais de l'ordre dans lequel l'import est fait : si les données de la table des commandes sont importées avant celles de la table des clients et qu'il y a une contrainte d'intégrité référentielle de la table des commandes vers la table des clients, les commandes risquent d'être rejetées car les clients n'auront pas encore été importés.

La technique classique consiste alors à supprimer ou désactiver les structures gênantes avant l'import et à les réactiver ou les recréer ensuite.

- Quels que soient les paramètres, l'import fait une activation des contraintes et des triggers qui étaient actifs au moment de l'export. Si vous aviez désactivé des contraintes et des triggers avant l'import, ils seront réactivés par l'import ; c'est un peu troublant parfois ...
- Lorsque l'objectif est de ne transférer que des données d'une base source vers une base cible (la structure est déjà prête dans la base cible), il est préférable de n'exporter que les données sans les contraintes, les index, les triggers, ... Cela permet de limiter les risques de comportements « bizarres » lors de l'import.

25.5.3 Export de niveau tablespace

L'export de niveau tablespace (notion de tablespace transportable) est intéressant pour transférer l'intégralité d'un tablespace d'une base à une autre, sous réserve :

- Que les bases soient sur la même plate-forme et aient le même jeu de caractères.
- Pas de tablespace portant le même nom dans la base cible.
- Que le tablespace soit auto-suffisant (pas de référence vers des objets stockés dans des tablespaces non transportés).

L'export s'effectue dans le jeu de caractères de la session qui effectue l'export, défini par la variable d'environnement NLS_LANG (exemple FRENCH_FRANCE.WE8ISO8859P1).

Une conversion automatique se produit si ce jeu est différent de celui de la base.

La fonctionnalité de tablespace transportable, introduite en version 8i, est particulièrement intéressante ; elle permet de transporter un tablespace d'une base à une autre, simplement en copiant directement les fichiers de données du tablespace et en le « branchant » sur la base d'arrivée.

Avec la version 9i, il est possible d'utiliser le paramètre TABLESPACE pour exporter toutes les tables situées dans un tablespace donné.

Si une table possède une partition dans le tablespace indiqué, elle sera exportée en totalité. Si l'option indexes=y est utilisée, les index associés aux tables seront également transportés.

Au moment du transport, le tablespace doit être READ ONLY, mais rien n'interdit, ensuite, dans la base source ou dans la base cible, de remettre le tablespace READ WRITE.

Le mode opératoire général est le suivant :

S'assurer que le tablespace concerné est READ ONLY ; au besoin, le passer READ ONLY.

Utiliser l'outil d'export sur la base source (qui contient le tablespace) pour extraire du dictionnaire les informations relatives à ce tablespace :

- Caractéristiques du tablespace
- Définition des objets qu'il contient
- Copier vers la base cible :
  - le fichier d'export contenant les définitions
  - les fichiers de données du tablespace.
- Utiliser l'outil d'import sur la base cible pour importer dans le dictionnaire les informations relatives au tablespace transporté :
  - Caractéristiques du tablespace
  - Définition des objets qu'il contient

Avec cette technique, à aucun moment, les données proprement dites ne sont ni lues (SELECT) par l'export ni insérées (INSERT) par l'import : le gain de temps est généralement appréciable.

Il faut *Oracle Entreprise Edition* sur la base source pour faire l'export ; par contre, l'import peut être réalisé sur n'importe qu'elle gamme Oracle.

25.6 Vues du dictionnaire de données de *DATA Pump*

Vous pouvez utiliser les vues du dictionnaire de données présentées ci-dessous pour obtenir des informations sur les jobs DATA Pump :

- V$SESSION : liste des sessions utilisateurs
- V$SESSION_LONGOPS : performances des sessions, indique l'état d'avancement du job représenté à travers le nombre de méga-bytes de données transférées. L'entrée contient la taille estimée du transfert et elle est mise à jour périodiquement pour refléter la quantité de données transférées.

- DBA_DATAPUMP_JOBS : identifie tous les jobs actifs du DATA Pump (quelque soit leur état) d'une instance ou de toutes les instances du RAC (Y sont aussi illustrées toutes les MT qui ne sont pas actuellement associées à un job actif)
- DBA_DATA_PUMP_SESSIONS : les sessions utilisateurs correspondant à un job.

```
set linesize 150
col program for A30
col module for A20
select username, program, module, action
from v$session
/

USERNAME                       PROGRAM                        MODULE               ACTION
------------------------------ ------------------------------ -------------------- -------
-------------------------

SYSTEM                         sqlplus.exe                    SQL*Plus
                               ORACLE.EXE (q000)
SYSTEM                         ORACLE.EXE (DW01)
SYSMAN                         OMS                            OEM.SystemPool
XMLLoader0
SYSMAN                         OMS                            OEM.SystemPool
SYSMAN                         OMS                            OEM.SystemPool
DBSNMP                         emagent.exe                    emagent.exe
SYSMAN                         OMS                            OEM.SystemPool
PingHeartbeatRecorder
SYSTEM                         expdp.exe                      expdp.exe
SYSMAN                         OMS                            OMS
SYSMAN                         OMS                            OEM.BoundedPool
SYSMAN                         OMS                            OEM.SystemPool
DBSNMP                         emagent.exe                    emagent.exe

ORACLE.EXE (MMNL)
ORACLE.EXE (MMON)
ORACLE.EXE (QMNC)
SYSMAN                         OMS                            OMS
ORACLE.EXE (CJQ0)
ORACLE.EXE (RECO)
ORACLE.EXE (SMON)
ORACLE.EXE (DBW0)
ORACLE.EXE (MMAN)
ORACLE.EXE (PMON)
30 rows selected.
```

26 SQL*Loader

Parmi les fonctionnalités de SQL*Loader, les suivantes sont particulièrement intéressantes :

- Il y a peu de limitation sur le format des données du fichier externe (largeur fixe, avec séparateur, ...).
- Plusieurs fichiers externes peuvent être chargés dans la même session.
- Plusieurs tables peuvent être chargées dans la même session.
- Des critères peuvent être définis pour éliminer certaines données du fichier externe.
- Les données peuvent être transformées avec des fonctions SQL pendant le chargement.
- Des numéros séquentiels uniques peuvent être générés pour certaines colonnes.

En entrée, SQL*Loader prend un fichier de contrôle qui pilote le chargement (rien à voir avec le fichier de contrôle d'une base) et un ou plusieurs fichiers de données `ASCII` (pas des fichiers de données d'une base Oracle).

En sortie, SQL*Loader alimente la base Oracle et génère un fichier de log, un fichier des rejets (*bad* - données erronées) et un fichier des refus (*discard* - données écartées).

Pour des petits volumes, les données peuvent être directement incluses dans le fichier de contrôle.

Le fichier *discard* contient des enregistrements qui ont été refusés (écartés) par SQL*Loader car ils ne respectaient pas des conditions, des critères, spécifiés dans le fichier de contrôle.

Le fichier *bad* contient des enregistrements qui ont été rejetés soit par SQL*Loader, soit par Oracle:

- Rejet par SQL*Loader
- Format de l'enregistrement non valide par rapport à la description du fichier de contrôle
- Rejet par Oracle
- Violation d'une contrainte d'intégrité
- Type de données non valide
- Rejet par un trigger

Les enregistrements refusés ou rejetés sont écrits tels quels dans les fichiers `bad` et `discard` qui ont donc la même structure que les fichiers de données utilisés en entrée ; après correction éventuelle des enregistrements, les fichiers `bad` et `discard` peuvent être utilisés comme fichiers d'entrée.

Le fichier de log donne énormément d'informations sur le résultat du chargement :

- Date
- Nom des fichiers utilisés
- Paramètres utilisés
- Tables cibles et mode d'alimentation
- Conditions éventuelles sur les enregistrements
- Nombre d'enregistrements chargés
- Nombre d'enregistrements écartés
- Nombre d'enregistrements rejetés
- Messages d'erreurs relatifs aux rejets

SQL*Loader peut effectuer l'import selon deux « chemins », le chemin direct et le chemin conventionnel.

- **Chemin direct** = Les données sont chargées en mémoire, formatées dans des blocs qui sont écrits directement dans la base.
- **Chemin conventionnel** = Les données sont chargées en mémoire et insérées dans les tables par des ordres `INSERT` classiques. Avec le chemin conventionnel, tous les mécanismes classiques sont appliqués (contraintes, triggers, ...).

Le chemin direct est plus performant mais a les conséquences suivantes :

- Seules les contraintes `NOT NULL, PRIMARY KEY` et `UNIQUE KEY` sont appliquées.
- Les triggers `INSERT` ne sont pas exécutés.
- D'autres utilisateurs ne peuvent pas apporter de modifications aux tables.
- Il faut lancer l'outil dans une fenêtre du système d'exploitation en mettant des paramètres sur la ligne de commande.

Les paramètres peuvent être listés dans un fichier de paramètres dont le nom seul est passé sur la ligne de commande :

```
C:\>sqlldr parfile=balance.par
```

Il y a deux catégories de paramètres à ne pas confondre :

- Les paramètres du fichier de contrôle
- Les paramètres de la ligne de commande qui peuvent être listés dans un fichier de paramètres (paramètre de ligne de commande `PARFILE`)
- Certains paramètres de la ligne de commande peuvent être inclus dans le fichier de contrôle (paramètre de fichier de contrôle `OPTIONS`) ou sont redondants avec des paramètres du fichier de contrôle !

Les paramètres du fichier de contrôle sont essentiellement destinés à décrire la structure des enregistrements en entrée, les tables cibles et la nature des contrôles/traitements à réaliser sur les enregistrements.

26.1 Fichier de paramètres

Les paramètres de la ligne de commande, ou du fichier de paramètres indiqué sur la ligne de commande, contrôlent le fonctionnement général de l'outil.

Les principaux paramètres de la ligne de commande sont les suivants :

- **`BAD =`** Nom du fichier *bad* (avec éventuellement un chemin complet). Par défaut égal au nom du fichier de contrôle, mais avec l'extension .bad.
- **`BINDSIZE`** = (65536) , Taille maximum en octets de la *bind array* (« zone de travail «). Contrôle la quantité de données traitée en un seul `INSERT` et la fréquence du `COMMIT` (en corrélation avec le paramètre `ROWS`).
- **`CONTROL =`** Nom du fichier de contrôle (avec éventuellement un chemin complet).
- **`DATA =`** Nom du fichier de données à traiter (généralement plutôt indiqué dans le fichier de contrôle). Par défaut égal au nom du fichier de contrôle, mais avec l'extension .dat.

- **DIRECT =** (FALSE) TRUE : chemin direct. FALSE : chemin conventionnel.
- **DISCARDFILE =** Nom du fichier discard (avec éventuellement un chemin complet). Par défaut, égal au nom du fichier de contrôle, mais avec l'extension .dsc.
- **DISCARDMAX =** Nombre maximum de rejets autorisés avant l'arrêt du chargement (1 = arrêt au premier). Par défaut, pas d'arrêt.
- **ERRORS** = (50), Nombre d'erreurs d'insertion autorisées avant l'arrêt du chargement (0 = aucune erreur autorisée - mettre un très grand nombre pour ne pas s'arrêter en cas d'erreur). Les enregistrements incriminés sont écrits dans le fichier bad.
- **LOAD =** Nombre maximum d'enregistrements à charger (après SKIP).
- **LOG =** Nom du fichier log (avec éventuellement un chemin complet). Par défaut, égal au nom du fichier de contrôle, mais avec l'extension .log.
- **PARFILE =** Nom du fichier de paramètres (avec éventuellement un chemin complet).
- **READSIZE** = (65536) , Taille en octets du buffer de lecture.
- **ROWS** = (64) , Nombre de lignes par COMMIT. Si ROWS x taille de la ligne est supérieure à BINDISZE, ROWS est automatiquement diminué. Si ROWS x taille de la ligne est inférieur à BINDISZE, ROWS est utilisé tel quel (il n'est pas augmenté).
- **SILENT =** Liste les catégories de message qui ne doivent pas être reportés à l'écran (HEADER, FEEDBACK, ERRORS, DISCARDS, PARTITIONS, ALL).
- **SKIP** = Nombre d'enregistrements à éliminer avant de commencer le chargement (aucun par défaut).
- **USERID =** Paramètres d'ouverture de la session sous la forme : nom_utilisateur[/mot_de_passe][@nom_service]. Une invite s'affiche pour saisir le mot de passe s'il est non spécifié.
- **SKIP_INDEX_MAINTENANCE** = Mode direct uniquement. YES : les index ne sont pas mis à jour (ils sont marqués UNUSABLE et il faut les reconstruire). NO : les index sont mis à jour.
- **SKIP_UNUSABLE_INDEXES** = YES, autorise le chargement même s'il existe des index préalablement UNUSABLE. NO : n'autorise pas le chargement s'il existe des index préalablement UNUSABLE. Si le cas est rencontré, l'enregistrement est simplement rejeté en chemin conventionnel mais le chargement s'arrête en chemin direct.

Exemples de fichiers de paramètres :

Cas où toutes les informations sont en fait dans le fichier de contrôle.

```
userid=system/manager
control=balance.ctl
```

Cas où le fichier de contrôle peut s'appliquer sur des fichiers de données de diverses origines (un seul fichier de contrôle et plusieurs fichiers de paramètres).

```
userid=system/manager
control=balance.ctl
data=balance_lyon.dat
log=balance_lyon.log
bad=balance_lyon.bad
discardfile=balance_lyon.dsc
```

26.2 Le fichier de contrôle

La syntaxe présentée ci-dessous n'est pas complète, mais les clauses les plus usuelles y sont présentes.

Les clauses doivent apparaître dans l'ordre indiqué.

Les lignes de commentaire doivent commencer par deux signes moins (--).

```
[ OPTIONS(liste d'options) ]
LOAD DATA
[ INFILE fichier | *
[ BADFILE fichier ] [ DISCARDFILE fichier ] [ DISCARDMAX valeur ] ]
[ INSERT | APPEND | REPLACE | TRUNCATE ]
INTO TABLE nom
[ INSERT | APPEND | REPLACE | TRUNCATE ]
[ WHEN condition ]
[ FIELDS TERMINATED BY 'x' [ OPTIONALLY ENCLOSED BY 'y' ] ]
[ TRAILING NULLCOLS ]
( colonne [ POSITION(x:y) ] [ type ] [ clause_SQL ],
    ...
    )
[ BEGINDATA données ]
```

- **LOAD DATA**

La clause `INFILE` donne l'emplacement d'un fichier de données à traiter ou est égal au caractère * si les données sont dans le fichier de contrôle (clause `BEGINDATA`).
De manière optionnelle, cette clause peut spécifier un fichier bad (option `BADFILE`), un fichier `DISCARD` (option `DISCARDFILE`) et un nombre maximum de rejets autorisés avant l'arrêt du chargement (option `DISCARDMAX`) ; si les paramètres équivalents de la ligne de commande ont été indiqués, ce sont ces derniers qui s'appliquent.
S'il y a plusieurs fichiers à charger en une seule session, plusieurs clauses `INFILE` peuvent être présentes, chaque clause pouvant spécifier ses propres options `BADFILE, DISCARDFILE` et `DISCARDMAX`.
Si le fichier de données est indiqué en paramètre de la ligne de commande (`DATA`), la clause est vide (mais il faut laisser le mot clé `LOAD DATA`).

INSERT | APPEND | REPLACE | TRUNCATE La clause suivante précise le mode de chargement dans les tables :

- `INSERT` = Ajout, mais uniquement pour une table vide (erreur sinon)
- `APPEND` = Ajout à la table (peut être vide ou non)
- `REPLACE` = Remplace tout le contenu de la table (un ordre `DELETE` est exécuté avant)
- `TRUNCATE` = Remplace tout le contenu de la table (un ordre `TRUNCATE` est exécuté avant)

- **INTO TABLE**

La clause `INTO TABLE` donne le nom d'une table à charger et décrit comment effectuer le chargement dans cette table. Si plusieurs tables sont chargées à partir d'un même fichier de données, plusieurs clauses `INTO TABLE` sont spécifiées. Pour chaque table, il est possible d'indiquer les options suivantes :

- `INSERT | APPEND | REPLACE | TRUNCATE` : Mode de l'import pour la table

`WHEN` : Indique une condition sur l'enregistrement pour qu'il soit effectivement chargé dans cette table. La condition peut porter soit sur une colonne de la table cible soit sur un champ de l'enregistrement source défini par la position de son caractère de début et la position de son caractère de fin sous la forme « début:fin ».

• FIELDS TERMINATED BY 'x' [OPTIONALLY ENCLOSED BY 'y'] : Pour des enregistrements de longueur variable (avec séparateur), indique comment sont délimités les champs avec :
TERMINATED BY 'x' : caractère séparateur des enregistrements (une virgule,...)
OPTIONALLY ENCLOSED BY 'y' : caractère pouvant entourer les enregistrements (apostrophes, ...)
TRAILING NULLCOLS : Les colonnes non présentes à la fin de l'enregistrement sont mises à NULL (si l'option est absente et que des colonnes vides existent à la fin, l'enregistrement est rejeté).
• Colonne [POSITION(x:y)] [type] [clause_SQL] Liste des colonnes à alimenter dans la table avec :
COLONNE = le nom de la colonne cible
POSITION(x:y) : position du champ correspondant dans l'enregistrement source (cas d'un enregistrement de longueur fixe) ; Pour des enregistrements avec séparateur, la correspondance colonne/enregistrement est en lien avec la position (1ère colonne de la liste = 1er enregistrement, ...)
TYPE : type de données (en cas d'ambiguïté)
CLAUSE_SQL : clause SQL à appliquer
Pour référencer une colonne X dans la clause SQL, utiliser la syntaxe :
X (caractère deux points devant le nom de la colonne).
Il existe d'autres options sur la spécification des colonnes.
BEGINDATA
La clause BEGINDATA marque le début des données si celles-ci sont incluses dans le fichier de contrôle (INFILE *).

26.3 Exemples de chargements

Les différents exemples sont présentés avec des données incluses (INFILE * + BEGINDATA) pour mieux visualiser la correspondance entre les données et les paramètres du fichier de contrôle.

Ces exemples sont très simples à adapter au chargement d'un fichier externe :

- Copier les données, sans le BEGINDATA, dans un fichier.
- Mettre le nom du fichier dans la clause INFILE (à la place du caractère *).
- Supprimer la clause BEGINDATA.

Pour ces exemples, nous supposons l'existence des tables suivantes :

```
•      tables des Employe_BIS
CREATE TABLE Employe_BIS (
Code  NUMBER(6),
nom VARCHAR2(40),
prenom VARCHAR2(40),
sexe CHAR(1),
date_naissance DATE,
adresse VARCHAR2(150)
)
/

•      tables des Employe_BIS masculins (pas de colonne sexe)
CREATE TABLE Employe_BIS_M (
Code  NUMBER(6),
nom VARCHAR2(40),
prenom VARCHAR2(40),
date_naissanceDATE,
adresse VARCHAR2(150)
)
/
```

```
•      tables des Employe_BIS féminins(pas de colonne sexe
•      ni de colonne date_naissance)
CREATE TABLE Employe_BIS_F (
code NUMBER(6),
nom VARCHAR2(40),
prenom VARCHAR2(40),
adresse VARCHAR2(80)
)
/

•      séquence utilisée pour alimenter les numéros d'employé
CREATE SEQUENCE SEQ_Employe_BIS
/
```

26.3.1 Exemples de fichiers de contrôle : Longueur variable enregistrements

```
LOAD DATA
INFILE *
INTO TABLE Employe_BIS
APPEND
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
TRAILING NULLCOLS
(
prenom ,
nom ,
sexe ,
date_naissance « TO_DATE(:date_naissance,'YYYYMMDD') »,
adresse,
code     "SEQ_ Employe_BIS.NEXTVAL"
)
BEGINDATA
Pierre,DUPOND,M,19660826, "2 rue de Matignon 78711 Mantes la Ville"
Marise,LEROY,F, "125 rue de Champion 75000 Paris"
Jean, »PEYRAC (de) »,M,19631204, "6 rue du Faubourg St Antoine 75000 Paris"
```

Pour cet exemple, les enregistrements ont une longueur variable, avec séparateur.

Spécifications des données en entrée :

- Les champs sont délimités par une virgule. Clause `TERMINATED BY ','`
- Ils peuvent éventuellement être entourés de guillemets (c'est le cas du nom dans la troisième ligne). Clause `OPTIONALLY ENCLOSED BY '"'`
- Ils peuvent être manquants en fin de ligne (pas de date de naissance sur la deuxième ligne) ; dans ce cas mettre un `NULL`. Clause `TRAILING NULLCOLS`
- La date de naissance est fournie sous forme de chaîne au format `YYYYMMDD` mais doit être stockée dans une colonne de type `DATE`. Clause SQL « `TO_DATE(:date_naissance,'YYYYMMDD')` » ; Noter le caractère deux points devant le nom de la colonne date_naissance pour la référencer dans le calcul.
- Le numéro d'adhérent n'est pas fourni ; il doit être calculé à l'aide d'une séquence. Clause SQL "`SEQ_Employe_BIS.NEXTVAL`"

Avec des enregistrements de longueur fixe, la correspondance entre les champs et les colonnes sont définies par la clause `POSITION` ; les colonnes qui ne sont pas alimentées par un champ de l'enregistrement peuvent être spécifiées n'importe où dans la liste des colonnes.

Dans cet exemple, les `EMPLOYE_BIS` de sexe féminin ne doivent pas être chargés.

Une clause `WHEN` (sexe = 'M') est ajoutée pour spécifier les enregistrements à conserver.

26.3.2 Exemples de fichiers de contrôle : Longueur fixe avec élimination d'enregistrements

```
LOAD DATA
INFILE *
INTO TABLE Employe_BIS
APPEND
WHEN (sexe = 'M')
TRAILING NULLCOLS
(
code    "SEQ_Employe_BIS.NEXTVAL",
prenom  POSITION(01:10),
nom     POSITION(11:22),
sexe    POSITION(23:23),
date_naissance POSITION(24:31) « TO_DATE(:date_naissance,'YYYYMMDD') »
)
BEGINDATA
Pierre   DUPOND     M19660826    "2 rue de Matignon 78711 Mantes la Ville"
Marise   LEROY      F            "125 rue de Champion 75000 Paris"
Jean     PEYRAC (de) M19631204   "6 rue du Faubourg St Antoine 75000 Paris"
```

26.3.3 Chargement dans deux tables

```
LOAD DATA
INFILE *
INTO TABLE Employe_BIS_M
APPEND
WHEN ((23) = 'M')
TRAILING NULLCOLS
(
code                                "SEQ_Employe_BIS.NEXTVAL",
prenom                              POSITION(01:10),
nom                                 POSITION(11:22),
date_naissance          POSITION(24:31) « TO_DATE(:date_naissance,'YYYYMMDD') »
)
INTO TABLE Employe_BIS_F
APPEND
WHEN ((23) = 'F')
TRAILING NULLCOLS
(
code                                "SEQ_Employe_BIS.NEXTVAL",
prenom                              POSITION(01:10),
nom                                 POSITION(11:22)
)
BEGINDATA
Pierre   DUPOND     M19660826    "2 rue de Matignon 78711 Mantes la Ville"
Marise   LEROY      F            "125 rue de Champion 75000 Paris"
Jean      PEYRAC (de) M19631204  "6 rue du Faubourg St Antoine 75000 Paris"
```

Pour cet exemple, les enregistrements ont une longueur fixe mais les EMPLOYE_BIS doivent être répartis entre deux tables en fonction de leur sexe.

Les deux tables n'ont pas de colonne sexe et la table des EMPLOYE_BIS de sexe féminin, pas de colonne date_naissance non plus.

Le fichier de contrôle utilise deux clauses INTO TABLE pour spécifier comment alimenter les deux tables.

Dans chaque clause INTO TABLE, une clause WHEN ((23) = 'X') permet d'indiquer si l'enregistrement courant doit être chargé dans la table ou non.

Comme les tables n'ont pas de colonne sexe, le seul moyen de désigner le champ correspondant est d'utiliser une notation par position du type (début:fin) ; le champ sexe commence (et finit) au 23ème caractère, soit (23:23) qui peut être abrégé en (23).

Chaque clause INTO TABLE a sa propre liste de colonnes.

26.3.4 Chargement dans deux tables avec utilisation d'une colonne FILLER

```
LOAD DATA
INFILE *
INTO TABLE Employe_BIS_M
APPEND
WHEN (sexe = 'M')
TRAILING NULLCOLS
(
code                                "SEQ_Employe_BIS.NEXTVAL",
prenom                              POSITION(01:10),
nom                                 POSITION(11:22),
sexe                    FILLER      POSITION(23:23),
date_naissance                      POSITION(24:31)
« TO_DATE(:date_naissance,'YYYYMMDD') »
)
INTO TABLE Employe_BIS_F
APPEND
WHEN (sexe = 'F')
TRAILING NULLCOLS
(
code                                "SEQ_Employe_BIS.NEXTVAL",
prenom                              POSITION(01:10),
nom                                 POSITION(11:22),
sexe                    FILLER      POSITION(23:23)
)
BEGINDATA
Pierre   DUPOND        M19660826    "2 rue de Matignon 78711 Mantes la Ville"
Marise   LEROY         F            "125 rue de Champion 75000 Paris"
Jean   ''PEYRAC (de)'' M19631204    "6 rue du Faubourg St Antoine 75000 Paris"
```

Cet exemple est une variante de l'exemple précédent dans lequel l'enregistrement correspondant au sexe est matérialisé et nommé pour faciliter sa manipulation, bien qu'il n'alimente pas une colonne des tables.

Colonne nommée dans la liste des colonnes avec la propriété FILLER (« remplissage »).

Cette « colonne » peut ensuite être manipulée comme si c'était une colonne de la table (utilisée dans une clause WHEN, dans une clause SQL) mais elle n'est pas chargée dans la table.

26.4 Chargement de données LOB

Cet exemple provient de la documentation Oracle « *b10825.pdf* ».

Chargement de données à partir de fichiers contenant les types LOB.

```
Control File Contents
LOAD DATA
INFILE 'sample.dat'
INTO TABLE person_table
FIELDS TERMINATED BY ','
(name CHAR(20),
1 ext_fname FILLER CHAR(40),
2 "RESUME" LOBFILE(ext_fname) TERMINATED BY EOF)
```

Datafile (sample.dat)

```
Johny Quest,jqresume.txt,
Speed Racer,'/private/sracer/srresume.txt',
Secondary Datafile (jqresume.txt)
Johny Quest
500 Oracle Parkway
...
```

Deusième Datafile (srresume.txt)

```
Speed Racer
400 Oracle Parkway
...
```

26.5 Chargement de formats XML

Il est possible de manipuler des formats XML en enregistrant un schéma xsl dans oracle comme type de donnée(par exemple personne.xsd).

IL est ensuite possible charger des données xml respectant Ce format dans la table.

Ce chargement peut se faire en utilisant les types suivant :

- Chargement de types XML à partir de types Primaires
- Chargement de types XML à partir de types BFILE
- Chargement de types XML à partir de types LOBFILEs

La documentation ci-dessous vous fournit un exemple d'utilisation.
HTTP://DOWNLOAD.ORACLE.COM/DOCS/CD/B19306_01/APPDEV.102/B14259/XDB25LOA.HTM

Example 29-1 Loading Very Large XML Documents Into Oracle Database Using SQL*Loader

Cet exemple utilize le fichier de control `load_data.ctl` pour charger des données de type XML dans la table `foo`. Le code enregistre le format `XMLTYPE` créé nommé `person.xsd` dans la table `foo`.

Cet exemple est extrait de la documentation oracle *xdb25loa.*

```
CREATE TYPE person_t AS OBJECT(name VARCHAR2(100), city VARCHAR2(100));/
BEGIN
•      Delete schema if it already exists (else error)
DBMS_XMLSCHEMA.deleteSchema('http://www.oracle.com/person.xsd', 4);
END;/
BEGIN
DBMS_XMLSCHEMA.registerschema('http://www.oracle.com/person.xsd',
,<schema xmlns="http://www.w3.org/2001/XMLSchema"' ||
, xmlns:per="http://www.oracle.com/person.xsd"' ||
, xmlns:xdb="http://xmlns.oracle.com/xdb"' ||
, elementFormDefault="qualified"' ||
, targetNamespace="http://www.oracle.com/person.xsd">' ||
, <element name="person" type="per:persontype"' ||
' xdb:SQLType= »PERSON_T »/>' ||
' <complexType name="persontype" xdb:SQLType="PERSON_T">' ||
' <sequence>' ||
' <element name="name" type="string" xdb:SQLName="NAME"' ||
' xdb:SQLType="VARCHAR2"/>' ||
' <element name="city" type="string" xdb:SQLName="CITY"' ||
' xdb:SQLType="VARCHAR2"/>' ||
' </sequence>' ||
' </complexType>' ||
' </schema>',
TRUE,
FALSE,
FALSE);
END;/

CREATE TABLE foo OF XMLType
XMLSCHEMA „http://www.oracle.com/person.xsd" ELEMENT „person";
```

Here is the content of the control file, `load_data.ctl`, for loading `XMLType` data using the registered XML schema, `person.xsd`:

```
LOAD DATA
INFILE *
INTO TABLE foo TRUNCATE
XMLType(xmldata)
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
(
xmldata
)
BEGINDATA
<person xmlns="http://www.oracle.com/person.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.oracle.com/person.xsd
http://www.oracle.com/person.xsd"> <name> xyz name 2</name> </person>
```

Here is the SQL*Loader command for loading the XML data into Oracle Database:

sqlldr [username]/[password] load_data.ctl (optional: direct=y)

In `load_data.ctl`, the data is present in the control file itself, and a record spanned only one line (it is split over several lines here, for printing purposes).

In the following example, the data is present in a separate file, `person.dat`, from the control file, `lod2.ctl`. File `person.dat` contains more than one row, and each row spans more than one line. Here is the control file, `lod2.ctl`:

```
LOAD DATA
INFILE *
INTO TABLE foo TRUNCATE
XMLType(xmldata)
FIELDS(fill filler CHAR(1),
xmldata LOBFILE (CONSTANT person.dat)
TERMINATED BY '<!-- end of record -->')
BEGINDATA
0
0
0
```

The three zeroes (`0`) after `BEGINDATA` indicate that three records are present in the data file, `person.dat`. Each record is terminated by `<!-- end of record -->`. The contents of `person.dat` are as follows:

```
<person xmlns="http://www.oracle.com/person.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.oracle.com/person.xsd
http://www.oracle.com/person.xsd">
<name>xyz name 2</name>
</person>
<!-- end of record -->
<person xmlns="http://www.oracle.com/person.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.oracle.com/person.xsd
http://www.oracle.com/person.xsd">
<name> xyz name 2</name>
</person>
<!-- end of record -->
<person xmlns="http://www.oracle.com/person.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.oracle.com/person.xsd
http://www.oracle.com/person.xsd">
<name>xyz name 2</name>
</person>
<!-- end of record -->
```

Here is the SQL*Loader command for loading the XML data into Oracle Database:

sqlldr [username]/[password] lod2.ctl (optional: direct=y)

27 Stratégie de Sauvegardes et Restaurations

La principale responsabilité du DBA est de prendre les mesures nécessaires pour assurer la sécurité et la disponibilité des données.

- ⇨ Il doit restituer les données en cas d'incident matériel ou d'erreur de manipulation.

Cette sécurité est assurée par :

- La mise en œuvre d'une protection des fichiers sensibles de la base
  - Fichiers de contrôle
  - Fichiers de Redo Log
- La mise en place d'une stratégie de sauvegarde/restauration
  - Adaptée aux contraintes de l'entreprise
  - Et qui aura été complètement testée et documentée

Le DBA doit faire des sauvegardes régulières de la base de données.

☺ **Sauvegardes Physiques ET**

😐 **Sauvegardes logiques**

Les questions à se poser pour définir la stratégie sont les suivantes :

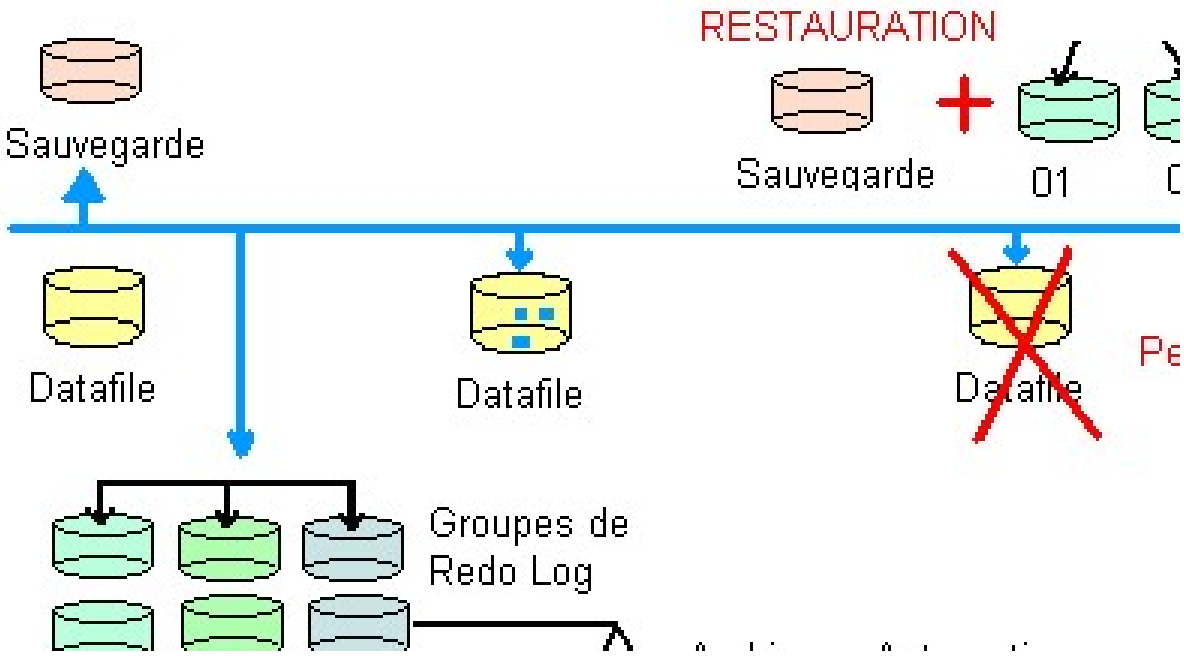
- ⇨ Est-il acceptable de perdre des données ?
  - si oui sur quelle période ?
- ⇨ Est-il possible d'arrêter périodiquement la base ?
  - si oui quand et combien de temps ?
  - Est-il possible de faire une sauvegarde complète de la base pendant l'arrêt de celle-ci ?

S'il n'est pas possible d'arrêter la base ou de faire une sauvegarde complète, il sera toujours possible de faire des sauvegardes partielles.

La sauvegarde d'un base de petits volumes, contenant des données stratégiques ou de volumes importants est différente. Par exemple, pour certaines bases de données, il sera préférable de recharger certaines tables plutôt que de les restaurer (gain de temps pour de gros volumes).

Ne pas oublier que le tablespace est l'unité de sauvegarde.

Il faudra penser au stockage des tables dans des tablespaces dédiés (les grosses tables ou les tables stratégiques) ou mutualisés (les petites tables de référence)



La restauration du fichier de données consiste à prendre la dernière sauvegarde du fichier de données, de remplacer par cette sauvegarde le fichier manquant ou endommagé et à appliquer sur cette sauvegarde les fichiers de Redo Log archivés.

Ceci afin de ramener le fichier de données dans l'état où il se trouvait juste avant l'incident.

27.1 Les modes NOARCHIVELOG et ARCHIVELOG

En fonction des possibilités d'arrêt et des pertes de données (acceptables), on choisira de mettre la base de données en mode `ARCHIVELOG` ou non.

En cas de passage en mode `ARCHIVELOG`, il faut étudier :

- Où mettre les archives (sur disque ou sur bande) ?
- Combien de jeux de sauvegarde conserve-t'on (épuration des sauvegardes et des archives) ?

Oracle écrit en général de manière cyclique dans les groupes de Redo Log en ligne qui forment le journal de reprise.

Oracle passe au fichier suivant lorsqu'il en a rempli un.

Lorsque le dernier fichier affecté au journal est plein, le processus d'arrière-plan `LGWR` (*Log Writer*) commence alors à écraser le contenu des membres du premier groupe de Redo Log.

Lorsqu'Oracle est exécuté en mode `ARCHIVELOG`, le processus `ARCH`, effectue une copie d'un membre de Redo Log après que le processus `LGWR` ait fini d'écrire.

27.1.1 Le mode NOARCHIVELOG

Un fichier de Redo Log peut être réutilisé immédiatement après qu'un point de synchronisation (*checkpoint*) ait eu lieu.

Après que le contenu d'un fichier de Redo Log ait été écrasé par une nouvelle écriture, les données à restaurer sont perdues.

Affichage du mode courant :

```
Select log_mode from v$database ;
LOG_MODE
NOARCHIVELOG
```

27.1.2 Le mode ARCHIVELOG

La synchronisation des fichiers de données est basée sur le numéro de séquence du fichier de Redo Log (sauf si le tablespace est en `READ ONLY`).

```
•      rechercher le dernier N° de SCN (transaction=9999999).
-- qui correspond à la colonne FIRST_CHANGE#.

SQL> select * from v$log_history;

RECID      STAMP    THREAD#   SEQUENCE# FIRST_CHANGE# FIRST_TI NEXT_CHANGE#
---------- ---------- ---------- ---------- ------------- -------- ------------
         1  568375848          1          1       2711661 07/09/05      2714944
         2  568375866          1          2       2714944 07/09/05      2715550
         3  568375884          1          3       2715550 07/09/05      2716153
         4  568375902          1          4       2716153 07/09/05      2716755
         5  568375919          1          5       2716755 07/09/05      2717359

6  568375937          1          6       2717359 07/09/05      2717961
         7  568375954          1          7       2717961 07/09/05      2718563
         8  568375971          1          8       2718563 07/09/05      2719168
         9  568375989          1          9       2719168 07/09/05      2719770
        10  568376006          1         10       2719770 07/09/05      2720373
        11  568376024          1         11       2720373 07/09/05      2720976
```

En cas de restauration, Oracle a besoin de tous les fichiers de Redo Log à partir de celui portant le numéro de séquence inscrit dans le fichier de données au moment de la sauvegarde. Autrement la restauration ne sera pas possible.

Si un fichier d'archive est perdu, Oracle ne pourra pas restaurer la base de données.

27.1.3 Mettre la base en mode ARCHIVELOG

Ce mode permet de garantir qu'un groupe de fichiers de Redo Log non archivés ne sera pas écrasé par LGWR. En version 9i, mettre la base en ARCHIVELOG ne démarrait pas automatiquement le processus ARCH. Il fallait le faire en positionnant le paramètre **LOG_ARCHIVE_START** ⑨ à TRUE.

⇨ En cas d'oublie de positionnement de ce paramètre, cela provoquait un crash de l'instance.

Pour débloquer la situation il fallait lancer la commande ARCHIVE LOG ALL, puis monter la base **(STARTUP MOUNT)** et lancer la commande **ARCHIVE LOG ALL.**

Démarrer le processus ARCH

⇨ Se charge de l'archivage proprement dit
⑩ A partir de la version 10g, placer la base en mode ARCHIVELOG démarre automatiquement les processus ARCH0 et ARCH1 lors de l'ouverture de la base de données.

Lors de l'activation de l'archivage, il est primordial de bien vérifier que tout fonctionne bien et que des archives sont bien générées.

27.1.4 Les paramètres du processus *ARCH*

Les différents paramètres relatifs au mode ARCHIVELOG sont présentés ci-dessous.

- **LOG_ARCHIVE_START** ⑨ (en version 9i)

Démarre (valeur TRUE) ou non (valeur FALSE, par défaut) le processus ARCH

- **LOG_ARCHIVE_DEST_i = 'LOCATION=chemin_local'**
- **• (i de 1 à 10)**

Destination de l'archivage pour une ENTERPRISE édition (au moins une obligatoire)
Syntaxe simplifiée pour une destination locale (au moins une obligatoire)
Exemple :
LOG_ARCHIVE_DEST_1 = 'LOCATION=d:\oracle\admin\BORA\arch' [*Pas de blanc pour 'location='*]
Le répertoire spécifié doit exister ; il n'est pas créé par Oracle.
ATTENTION, l'utilisation des derniers paramètres est destinée au DATA GUARD.

- **LOG_ARCHIVE_FORMAT**

Format souhaité pour le nom des archives (*les 3 sont obligatoires*)
%s ou %S : numéro de séquence du fichier de Redo Log
%t ou %T : numéro d'instance
%r ou %R : identifiant de remise à zéro des fichiers de journalisation
Exemple :
LOG_ARCHIVE_FORMAT = « 'Redo%S_%R%T.arch' »

- **ARCHIVE_LAG_TARGET**

Durée maximale en seconde qui doit séparer 2 archivages.
Une valeur nulle désactive la fonctionnalité (valeur par défaut).
Valeur autorisée : entre 60 (1 minute) et 7200 (2 heures) ; permet de forcer l'archivage de façon périodique et de garantir une périodicité d'archivage stable.
Exemple :

```
Archive_lag_target = 1800     #30 minutes
```

27.2 Passer la base en mode ARCHIVELOG

Avant de passer la base en mode ARCHIVELOG, arrêtez celle-ci et effectuez une sauvegarde à froid.

Ne pas oublier de sauvegarder le fichier SPFILE.

En version 11g passer une base de données ORACLE en mode ARCHIVELOG consiste à monter la base de données et à changer son état. En effet, les destinations d'archive et les noms des archives générées sont paramétrés par défaut.

MODE OPERATOIRE

- Arrêter la base de données
  `SHUTDOWN IMMEDIATE`
- Monter la base
  `STARTUP MOUNT`
- Passer la base en ARCHIVELOG
  `ALTER DATABASE ARCHIVELOG ;`
- Ouvrir la base
  `ALTER DATABASE OPEN`

Après avoir passé la base en mode ARCHIVELOG, vérifiez que les archives sont générées correctement et apparaissent à l'emplacement demandé, puis **arrêtez la base de données et faites une première sauvegarde.**

```
•     passer la base en mode ARCHIVELOG
Create pfile from spfile ;
alter system set
LOG_ARCHIVE_DEST_1 = 'location=f:\oracle\oradata\BORA\arch\'
scope=SPFILE ;

ALTER SYSTEM SET LOG_ARCHIVE_FORMAT='arch%S_%R%T.arc'
scope=SPFILE ;

Shutdown immediate;
Startup mount;
Alter database archivelog;
Alter database open;
```

27.3 Administrer le processus *ARCH*

La clause ARCHIVE LOG de l'ordre SQL : ALTER SYSTEM permet d'administrer le processus ARCH après démarrage de la base :

- Arrêter/Démarrer le processus ARCH
- Archiver manuellement un groupe de fichiers de Redo Log

```
ALTER SYSTEM ARCHIVE LOG
STOP
| START [ TO 'destination' ]
| GROUP numero [ TO 'destination' ]
;
```

```
•      Arrêter le processus ARCH
ALTER SYSTEM ARCHIVE LOG STOP;
•      Démarrer le processus ARCH
ALTER SYSTEM ARCHIVE LOG START;
•      Archiver le groupe 2 vers une autre destination
ALTER SYSTEM ARCHIVE LOG GROUP 2 TO 'd:\temp' ;
```

L'ordre SQL : ALTER SYSTEM ne dure que pendant l'instance en cours et ne dure pas après un arrêt de la base.

La commande ARCHIVE LOG LIST permet d'afficher des informations sur l'archivage :

```
Connect / as sysdba
Connected.
Archive log list
Database log mode              Archive Mode
Automatic archival             Enabled
Archive destination            h:\oracle\oradata\ORCL\archive
Oldest online log sequence     6355
Next log sequence to archive   6358
Current log sequence           6358
```

- **Database log mode** = mode de fonctionnement de la base
- **Automatic archival** = état du processus ARCH (enabled s'il est lancé, disabled s'il est arrêté)
- **Archive destination** = destination des archives
- **Oldest online log sequence** = plus ancien numéro de séquence des fichiers de Redo Log en ligne
- **Next log séquence to archive** = prochain numéro de séquence des fichiers de Redo Log à archiver (ligne absente si la base est en NOARCHIVELOG)
- **Current log séquence** = numéro de séquence des fichiers de Redo Log courants

27.3.1 Forcer l'archivage de façon périodique

En règle générale, il est conseillé d'avoir des basculements de fichiers de Redo Log (et donc des archivages en mode `ARCHIVELOG`) toutes les 20 à 30 minutes.

⇨ Plus l'activité est importante, plus il y a de basculement et génération d'archivages !

Pour garantir une périodicité stable, il est possible de spécifier une valeur dans le paramètre `ARCHIVE_LAG_TARGET` :

- Indique en secondes la durée maximale qui doit séparer deux archivages
- Une valeur `nulle` désactive la fonctionnalité (valeur par défaut)
- Valeurs autorisées : entre 60 (une minute) et 7200 (2 heures)
- Le paramètre est dynamique

28 Sauvegardes

Pour connaître l'emplacement et le nom des fichiers de la base, il existe une astuce qui consiste à générer la trace du fichier de contrôle.

Cette trace vous permettra de connaître l'emplacement et le nom des fichiers de la base de données mais aussi de recréer le fichier de contrôle en cas de celui-ci ou en cas de restauration partielle de la base de données.

```
SQL> show parameter user_dump_dest
NAME                 TYPE        VALUE
-------------------- ----------- ------------------------------
diagnostic_dest         string      D:\Oracle\admin\COLLEGE\trace
```

On possède un script de recréation qui a été réalisé avec la commande :

```
ALTER DATABASE BACKUP CONTROLFILE TO TRACE ;
```

```
*** 2004-06-14 10:46:04.000
# The following are current System-scope REDO Log Archival related
# parameters and can be included in the database initialization file.
#
# Below are two sets of SQL statements, each of which creates a new
# control file and uses it to open the database. The first set opens
# the database with the NORESETLOGS option and should be used only if
# the current versions of all online logs are available. The second
# set opens the database with the RESETLOGS option and should be used
# if online logs are unavailable.
# The appropriate set of statements can be copied from the trace into
# a script file, edited as necessary, and executed when there is a
# need to re-create the control file.
#
#     Set #1. NORESETLOGS case
#
# The following commands will create a new control file and use it
# to open the database.
# Data used by the recovery manager will be lost. Additional logs may
# be required for media recovery of offline data files. Use this
# only if the current version of all online logs are available.
STARTUP NOMOUNT
CREATE CONTROLFILE REUSE DATABASE "OPTIMUM" NORESETLOGS  NOARCHIVELOG
•      SET STANDBY TO MAXIMIZE PERFORMANCE
MAXLOGFILES 32
MAXLOGMEMBERS 5
MAXDATAFILES 128
MAXINSTANCES 16
MAXLOGHISTORY 1815
LOGFILE
GROUP 1 (
'D:\ORACLE\ORADATA\OPTIMUM\REDO01A.LOG',
'D:\ORACLE\ORADATA\OPTIMUM\REDO01B.LOG'
) SIZE 10M,
GROUP 2 (
'D:\ORACLE\ORADATA\OPTIMUM\REDO02A.LOG',
'D:\ORACLE\ORADATA\OPTIMUM\REDO02B.LOG'
) SIZE 10M,
GROUP 3 (
'D:\ORACLE\ORADATA\OPTIMUM\REDO03A.LOG',
'D:\ORACLE\ORADATA\OPTIMUM\REDO03B.LOG'
) SIZE 10M
```

```
•      STANDBY LOGFILE
DATAFILE
'D:\ORACLE\ORADATA\OPTIMUM\SYSTEM01.DBF',
'D:\ORACLE\ORADATA\OPTIMUM\UNDOTBS01.DBF',
'D:\ORACLE\ORADATA\OPTIMUM\ELEVE01.DBF',
'D:\ORACLE\ORADATA\OPTIMUM\INDX01.DBF',
'D:\ORACLE\ORADATA\OPTIMUM\OPDEF01.DBF'
CHARACTER SET WE8ISO8859P15
;
# Recovery is required if any of the datafiles are restored backups,
# or if the last shutdown was not normal or immediate.
RECOVER DATABASE
# Database can now be opened normally.
ALTER DATABASE OPEN;
# Commands to add tempfiles to temporary tablespaces.
# Online tempfiles have complete space information.
# Other tempfiles may require adjustment.
ALTER TABLESPACE TEMP ADD TEMPFILE 'D:\ORACLE9\ORADATA\OPTIMUM\TEMP01.DBF'
SIZE 10485760  REUSE AUTOEXTEND OFF;
# End of tempfile additions.
#
```

Le fichier trace « .TRC » qui est créé avec cette commande ne peut pas être lancé directement.

Il faut exclure toutes les lignes de début de fichier qui ne sont pas des commentaires et vérifier la syntaxe de la commande STARTUP.

Cette commande ne précise pas de fichier de paramètres. Il faut donc rajouter à la fin de la commande STARTUP le nom et le chemin du fichier de paramètres.

Pour se connecter sous l'instance où l'on souhaite ouvrir la base et lancer le script de création du fichier de contrôle, la base de données doit être à l'état NOMOUNT.

Après exécution, la base est ouverte et le fichier de contrôle récupéré.

28.1 Sauvegarde base arrêtée

Les sauvegardes hors ligne se produisent lorsque la base a été arrêtée proprement.

Pas de panne instance ou shutdown abort !

Les fichiers suivants doivent être sauvegardés :

- Fichiers de contrôle
- Fichiers de données
- Fichiers de Redo Log en ligne
- Fichier init.ora et spfile.ora (optionnel)

Une sauvegarde de tous les fichiers de la base lorsque la base de données est fermée permet d'obtenir une image complète de la base telle qu'elle existait au moment de son arrêt.

Une « copie » des fichiers avec la base de données ouverte ne serait pas valide, à moins de réaliser une sauvegarde en ligne !

Une sauvegarde hors ligne suivant un arrêt anormal de la base de données (`ABORT`) serait considérée comme incohérente.

- ⇨ Lors d'une restauration, son utilisation ne serait pas garantie et demanderait davantage d'efforts.

28.2 Sauvegarde base en ligne

Vous pouvez utiliser les sauvegardes en ligne pour n'importe quelle base de données qui fonctionne en mode `ARCHIVELOG`.

- ⇨ Dans ce mode, les fichiers de Redo Log en ligne sont archivés, ce qui génère un journal complet de toutes les transactions effectuées sur la base de données.

Les fichiers suivants doivent être sauvegardés :

- Fichier de contrôle
- Fichiers de données (sauvegardés à chaud)
- Fichier `init.ora` et `spfile.ora` (optionnel)

Une base de données peut être restaurée complètement à partir d'une sauvegarde en ligne, en ajoutant une récupération des données à partir des fichiers de Redo Log archivés.

Depuis la version 8i, Oracle propose un outil, le `LOGMINER` (package `DBMS_LOGMNR`), qui permet d'analyser les fichiers de Redo Log et de récupérer les ordres SQL de mise à jour exécutés par les transactions (ordres `REDO`) ainsi que les ordres SQL inverses (ordres `UNDO`).

Cet outil peut théoriquement être utilisé pour récupérer des ordres SQL permettant de rejouer les transactions pour une restauration incomplète.

Dans la pratique, la mise en œuvre n'est pas immédiate et le traitement de récupération peut être long.

28.2.1 Sauvegarde du fichier de contrôle

Le fichier de contrôle doit être sauvegardé lors de chaque sauvegarde complète ou partielle de la base de données.

Entre deux sauvegardes « normales », il est conseillé de sauvegarder le fichier de contrôle après toute restructuration importante de celle-ci (ajout/déplacement de fichiers de données ou de Redo Log)

Pour une sauvegarde base ouverte, utiliser l'ordre SQL : `ALTER DATABASE` :

```
ALTER DATABASE BACKUP CONTROLFILE TO
[ nouveau_nom | trace ]  [ reuse ]
;
```

En général, les sauvegardes binaires du fichier de contrôle sont suffisantes.

```
ALTER DATABASE BACKUP CONTROLFILE
TO 'f:\oracle\backup\ORCL\control.bak';
```

Ne pas faire une copie du fichier de contrôle au niveau du système d'exploitation alors que la base est ouverte car celle-ci serait inexploitable.

28.3 Sauvegarde partielle d'un tablespace ONLINE

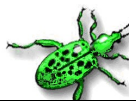
Lorsque le tablespace est passé en mode `BACKUP`, Oracle arrête d'écrire les *checkpoints* dans l'en-tête des fichiers de données du tablespace. Par contre, l'activité de lecture et d'écriture se poursuit normalement.

Une fois la sauvegarde terminée, le `END BACKUP` permet de sortir le tablespace du mode `BACKUP` et d'autoriser Oracle à reprendre les `CHECKPOINTS` dans l'en-tête des fichiers de données du tablespace.

Les fichiers sauvegardés sont incohérents mais Oracle a conservé la date et le numéro du dernier `CHECKPOINT` dans l'en-tête de chaque fichier. Ainsi, il pourra appliquer au fichier de données toutes les modifications postérieures au dernier `CHECKPOINT` (à partir des fichiers de Redo Log archivés) lors d'une restauration.

Cette sauvegarde n'est possible qu'en mode `ARCHIVELOG`. En supplément, il est important de faire une sauvegarde du fichier de contrôle et bien documenter ce qui a été sauvegardé.

Il est techniquement possible de mettre en parallèle des sauvegardes `ONLINE` de plusieurs `tablespaces` en mode `BACKUP`, mais ce n'est pas recommandé car l'activité sur les fichiers de Redo Log est augmentée.



Une sauvegarde `ONLINE` d'un tablespace qui n'est pas en mode `BACKUP` est généralement inexploitable.

L'oubli du END BACKUP ne génère aucun message de la part d'Oracle, pas même dans le fichier des alertes.

- **Ne pas faire**

```
Alter tablespace tbs1 begin backup ;
Alter tablespace tbs2 begin backup ;

•      sauvegarde des tablespaces ------

Alter tablespace tbs1 end backup ;
Alter tablespace tbs2 end backup ;
```

- **Faire plutôt**

```
Alter tablespace tbs1 begin backup ;
•      sauvegarde du tablespace tbs1 ---
Alter tablespace tbs1 end backup ;

Alter tablespace tbs2 begin backup ;
•      sauvegarde du tablespace tbs2 ---
Alter tablespace tbs2 end backup ;
```

Si un arrêt anormal de la base (ABORT) se produit alors qu'un tablespace est en mode BACKUP, une restauration du tablespace sera nécessaire au redémarrage

↱ à partir de la sauvegarde précédente du tablespace.

Un tablespace READ ONLY peut être sauvegardé ONLINE sans le mettre en mode BACKUP car il n'y a pas d'activité de mise à jour dans ce tablespace. Il n'y a pas de risque de données corrompues.

28.4 Sauvegarde de tous les tablespaces de la base ONLINE

La version 10g permet de placer tous les datafiles de la base de données dans le mode de sauvegarde ONLINE avec une seule commande. Vous n'avez plus besoin de placer chaque tablespace dans le mode de sauvegarde ONLINE un par un.

Pour cela, utiliser la commande :

```
ALTER DATABASE BEGIN BACKUP ;
-----   ------   -----   -----
ALTER DATABASE END BACKUP ;
```

Pour placer tous les fichiers de la base de données dans le mode de sauvegarde ONLINE, la base doit être ouverte (OPEN) et en mode ARCHIVELOG.

Pendant la sauvegarde, il n'est plus possible d'exécuter un SHUTDOWN normal, de placer un tablespace dans mode READ ONLY, ou dans le mode sauvegarde en ligne (ONLINE BACKUP MODE) ou de mettre un tablespace OFFLINE avec les options habituelles.

Toutefois, quand vous exécutez la commande ALTER DATABASE BEGIN BACKUP, tous les fichiers inexistants, OFFLINE ou READ ONLY, sont ignorés et le processus continue.

Si vous avez un « datafile » avec un statut OFFLINE un message d'avertissement est affiché.

28.5 Vues du dictionnaire de données

Plusieurs vues du dictionnaire permettent d'obtenir des informations sur l'archivage :

- V$DATABASE : mode de fonctionnement de la base (colonne LOG_MODE)
- V$INSTANCE : statut du processus ARCH (colonne ARCHIVER)
- V$LOG : statut des groupes vis à vis de l'archivage (colonne ARCHIVED)
- V$ARCHIVED_LOG : informations sur les fichiers de Redo Log archivés
- V$ARCHIVED_DEST : informations sur les destinations d'archivage

V$ARCHIVED_LOG	
RECID	Identifiant de l'enregistrement
NAME	Chemin complet de l'archive
SEQUENCE#	Numéro de séquence du fichier de Redo Log correspondant
FIRST_CHANGE#	Plus petit numéro de SCN (numéro de transaction) écrit dans l'archive
FIRST_TIME	Date et heure du plus petit numéro de SCN
NEXT_CHANGE#	Plus grand numéro SCN écrit dans l'archive
NEXT_TIME	Date et heure du plus grand numéro de SCN
COMPLETION_TIME	Date et heure de l'archivage

V$ARCHIVED_DEST	
DEST_NAME	Nom de la destination
DESTINATION	Chemin complet de la destination
STATUS	Statut de la destination (VALID, ERROR, …)
ERROR	Message d'erreur (en cas d'erreur)

28.6 Stratégie recommandée par Oracle

Le Grid Control permet de définir une stratégie de sauvegarde recommandée par Oracle, qui permet de protéger les données et fournit des possibilités de recouvrement de la base de données pour les 24 heures.

La stratégie recommandée par Oracle utilise la sauvegarde incrémentale et les caractéristiques de mises à jour des sauvegardes incrémentales proposées dans Recovery Manager (*RMAN*), en fournissant un processus de recouvrement plus rapide que si l'on applique les archives de Redo log.

La stratégie Oracle prend une copie complète de la base de données pour la première sauvegarde.

Il s'agit d'une sauvegarde complète de la base, suivie de sauvegardes incrémentales sur disque effectuées tous les jours à des heures précises via RMAN.

Comme ces sauvegardes sur disque sont mémorisées, vous pouvez toujours effectuer un recouvrement complet de la base ou un recouvrement de l'image faite à un moment donné le jour d'avant.

Une fois que vous avez complété et accepté les options de l'écran, votre base sera automatiquement sauvegardée.

28.7 Recover Manager (RMAN)

`RMAN` (*Recovery Manager*) est un outil qui accomplit une bonne partie du travail à la place du DBA dans le but de protéger la base de données.

Des opérations qui demandent du temps et qui sont délicates à réaliser sont exécutées avec `RMAN` au moyen de quelques commandes.

Lors de l'emploi de l'utilitaire `RMAN` (*Recovery Manager*), vous n'avez pas besoin de placer explicitement chaque tablespace dans un état de sauvegarde.

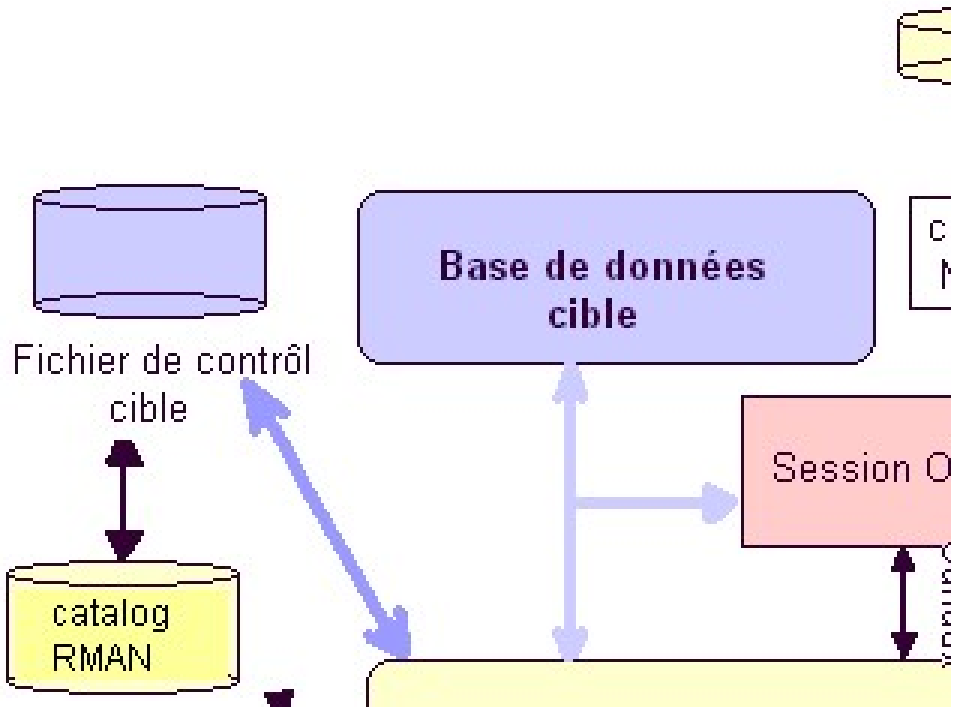
`RMAN` lit les blocs de données de la même manière qu'Oracle le fait lors de requêtes. Il n'y a plus aucun risque de corruption de fichier sauvegardé.

`RMAN` peut effectuer de nombreuses opérations identiques pour la plupart à celles que vous pouvez réaliser manuellement :

- Copie de fichiers de bases de données :
  `RMAN` peut créer des copies images de fichiers de données ou de contrôle ce qui revient à accomplir une sauvegarde à chaud.
- Création d'une base de données dupliquée :
  `RMAN` peut créer une copie de votre base de données sur la même machine ou sur une machine différente. Cette base peut avoir le même nom que votre base actuelle ou un nom différent.
- Création d'une base de données de secours :
  `RMAN` peut créer une base de données de secours qui servira à reprendre la fonction de base principale en cas de défaillance de cette dernière.
  Récupération de tablespace jusqu'à un point donné dans le temps : `TSPITR` (Tablespace Point-In-Time Recovery)
  `RMAN` peut effectuer une récupération d'un tablespace jusqu'à un point dans le temps. Il est possible ainsi de limiter la perte des données en rétablissant celui-ci jusqu'au point précédant l'incident. Les autres parties de votre base restent actuelles.

Exemple de sauvegarde RMAN

```
connect target
run {
configure device type DISK parallelism 1;
configure controlfile autobackup on;
configure controlfile autobackup format for device type disk to
'C:\sav_ora\autobackup\MABASE\SPFCTL_%d_%T_%F.BCK';
sql 'ALTER SYSTEM ARCHIVE LOG CURRENT';
sql 'ALTER SYSTEM ARCHIVE LOG CURRENT';
sql 'ALTER SYSTEM CHECKPOINT';
backup as compressed backupset archivelog
from time 'sysdate-7' until time 'sysdate'
format 'C:\sav_ora\arch\MABASE\Arch_%d_%T_%s%p'
tag = ARCH_MABASE
maxsetsize 4G ;
backup as compressed backupset database
format 'C:\sav_ora\db\MABASE\DBfull_%d_%T_%s%p'
tag = MABASE'
MAXSETSIZE 4G
FILESPERSET=3
include current controlfile;
sql 'ALTER SYSTEM CHECKPOINT';
sql 'ALTER SYSTEM ARCHIVE LOG CURRENT';
sql 'ALTER DATABASE BACKUP CONTROLFILE TO TRACE';
crosscheck backup device type disk;
delete expired backup;
delete obsolete redundancy 7 device type disk;
backup current controlfile format 'C:\sav_ora\db\MABASE\CTL_%d_%T_%s%p';
}
```



28.8 Le Flash Back

Le flashback permet de récupérer un ensemble de données ou d'objets dans le passé puis de les réinjecter dans la base de données. La technologie d'Oracle 10g ou 11g, offre la capacité d'interroger des versions anciennes de schéma d'objets ou de données.

Le « *flashback* » a été créé pour réparer facilement les données corrompues d'une table par un batch, en réinjectant dans la base de données les données récupérées avant le passage du batch grâce au flashback. Le *flashback* permet un retour arrière dans la base de données afin de sélectionner des objets ou parties d'objets pour les réinjecter dans la version actuelle de la base de données.

La version 9i introduisait la notion de « *flashback query* » pour fournir un mécanisme simple afin de réparer les erreurs humaines.

Oracle 10g et 11g étendent la technologie *flashback* pour assurer vite et facilement une réparation à tous les niveaux :

- **Flashback database** ; vous laisse rapidement ramener votre base à un point dans le temps en réparant toutes les modifications apportées depuis cet instant.
- **Flashback table** ; vous permet de retrouver rapidement une table et son contenu à un moment dans le passé.
- **Flashback Query** ; vous laisse voir les modifications apportées par une transaction à une ou plusieurs données, accompagnées de ses métadonnées.

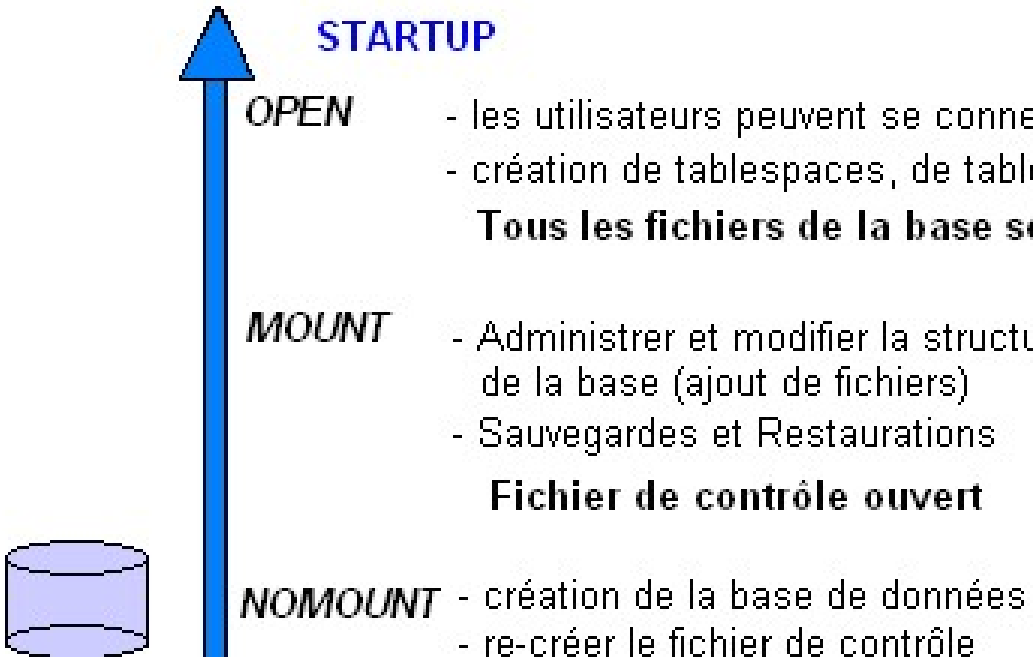
Objet	Scénario	Flashback
Database	`DROP USER`	Flashback database
	`TRUNCATE TABLE`	Flashback database
	Jobs Batchs	Flashback database
Table	`DROP TABLE`	Flashback Table
	`UPDATE` avec clause `WHERE` erronée	Flashback Table
Transaction	Comparer des données passées avec des données actuelles	Flashback Transaction
	Exécuter un job 2 fois car on n'est pas sûr de la validité des objets	Flashback Transaction

29 Restaurations

Dans une restauration, **c'est l'enchaînement des actions qui est compliqué**, pas la commande `RECOVER` en elle-même.

`NE PAS SE PRECIPITER`.
Avant de commencer toute opération de restauration, faire si possible une sauvegarde complète de la base endommagée. Elle fournit un point de retour en cas d'aggravation de la situation par une mauvaise manipulation.
Toutes les opérations de restauration nécessitent le privilège `SYSDBA`.

Lors d'un problème de perte de fichier, la base de données reste bloquée à l'état `NOMOUNT` s'il s'agit du fichier de contrôle, ou à l'état `MOUNT` s'il s'agit d'un fichier de données.



29.1 La commande RECOVER

Il existe trois variantes de la commande `RECOVER` :

- **`RECOVER DATABASE`** : Tous les fichiers de la base qui nécessitent une restauration sont traités.
- **`RECOVER TABLESPACE`** : Tous les fichiers des tablespaces cités en paramètre qui nécessitent une restauration, sont traités.
- **`RECOVER DATAFILE`** : Seuls les fichiers cités en paramètre, s'ils nécessitent une restauration, sont traités.

```
RECOVER [AUTOMATIC] [FROM 'location']
DATABASE [ UNTIL CANCEL
| UNTIL TIME 'YYYY-MM-DD:HH24:MI:SS'
| UNTIL CHANGE integer ]
[ USING BACKUP CONTROLFILE ]
```

Dans les trois cas, la commande RECOVER est capable de déterminer seule, quels sont les fichiers de Redo Log à appliquer sur les différents fichiers lors de la restauration.

Elle recherche les fichiers de Redo Log archivés qu'elle souhaite appliquer dans le répertoire défini par la valeur actuelle du paramètre LOG_ARCHIVE_DEST_1 du fichier spfile.

Si les fichiers de Redo Log archivés ne sont pas à l'emplacement attendu (soit parce que la destination des archives a changé, soit parce que les archives ont été archivées sur bande), il faudra intervenir pour aider la commande RECOVER à trouver l'emplacement.

Par défaut, la commande RECOVER fonctionne en mode manuel ; elle demande confirmation de l'emplacement de chaque fichier de Redo Log archivé qu'elle s'apprête à appliquer.

Il est possible alors :

- de valider (pour indiquer que l'archive est bien à l'emplacement indiqué)
- d'indiquer un autre emplacement si l'archive n'est pas/plus à l'endroit attendu
- de taper AUTO pour passer en mode automatique (voir ci-dessous) ou de taper CANCEL pour arrêter l'opération

29.1.1 Exemples de restaurations

- **AUTOMATIC** = la restauration se fera de façon automatique sans demander à l'opérateur de confirmer le passage de chaque fichier d'archive.

```
SQL> RECOVER DATABASE
ORA-00279: change 44629 generated at 08/07/2001 16:15:55 needed for thread 1
ORA-00289: suggestion : D:\ORACLE\ORADATA\ORCL\ARCH\ORCL00002.ARC
ORA-00280: change 44629 for thread 1 is in sequence #2
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

--- réponse saisie ---
d:\oracle\oradata\ORCL\temp\ORCL00002.arc
```

- **FROM 'location'** = précise à Oracle l'emplacement des archives de Redo Log

```
SQL> RECOVER DATABASE FROM 'd:\oracle\oradata\test\temp'
ORA-00279: change 44629 generated at 08/07/2001 16:15:55 needed for thread 1
ORA-00289: suggestion : D:\ORACLE\ORADATA\TEST\TEMP\TEST00002.ARC
ORA-00280: change 44629 for thread 1 is in sequence #2

ORA-00279: change 45013 generated at 08/08/2001 06:59:49 needed for thread 1
ORA-00289: suggestion : D:\ORACLE\ORADATA\TEST\TEMP\TEST00003.ARC
ORA-00280: change 45013 for thread 1 is in sequence #3
ORA-00278: log file 'D:\ORACLE\ORADATA\TEST\TEMP\TEST00002.ARC' no longer needed for this
recovery

Log applied.
Media recovery complete.
SQL>
```

- **UNTIL CANCEL** = Précise à Oracle que la restauration s'arrête lorsque l'opérateur saisit CANCEL

```
SQL> RECOVER DATABASE
ORA-00279: change 44629 generated at 08/07/2001 16:15:55 needed for thread 1
ORA-00289: suggestion : D:\ORACLE\ORADATA\ORCL\ARCH\ORCL00002.ARC
ORA-00280: change 44629 for thread 1 is in sequence #2
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

--- réponse saisie ---
d:\oracle\oradata\ORCL\temp\ORCL00002.arc

ORA-00279: change 45013 generated at 08/08/2001 06:59:49 needed for thread 1
ORA-00289: suggestion : D:\ORACLE\ORADATA\ORCL\ARCH\ORCL00003.ARC
ORA-00280: change 45013 for thread 1 is in sequence #3
ORA-00278: log file 'd:\oracle\oradata\ORCL\temp\ORCL00002.arc' no longer needed for this
recovery
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

--- réponse saisie ---
cancel

Media recovery cancelled.
```

- **UNTIL TIME** = Précise la date et l'heure d'arrêt de la restauration, en cas de restauration partielle.
  La condition d'arrêt est spécifiée par une date et une heure sous forme d'une chaîne entre apostrophes au format YYYY-MM-DD:HH24:MI:SS.
  La fonction TO_DATE peut être utilisée pour la mise en forme de la date :
  set until time "TO_DATE('02/24/2004 13:13:00','MM/DD/YYYY HH24:MI:SS')"

```
SQL> recover until time '2004-01-26:11:08:18';
ORA-00279: changement 72152 gÛnÛrÚ Ó 01/25/2004 15:51:23 requis pour thread 1
ORA-00289: suggestion : D:\ORACLE9\ADMIN\COLLEGE\ARCH\REDO00023.ARCH
ORA-00280: le changement 72152 pour le thread 1 se trouve au no de sÚquence 23
```

- **UNTIL CHANGE integer**= Précise le numéro de transaction (SCN) d'arrêt en cas de restauration partielle.
  La condition d'arrêt est spécifiée par un numéro de changement (numéro de transaction).
- **USING BACKUP CONTROLFILE** = indique à Oracle qu'une sauvegarde du fichier de contrôle est utilisée.

L'opération RECOVER peut être arrêtée temporairement puis reprise ultérieurement ; dans ce cas, elle redémarre à l'endroit où elle s'était arrêtée.

Si le RECOVER est interrompu avant la fin, la base ne peut pas être ouverte. Il est obligatoire, soit d'aller jusqu'au bout, soit de demander explicitement une restauration incomplète avec l'option UNTIL.

Cette séquence montre la possibilité de désigner un autre emplacement pour les archives, la possibilité d'arrêter l'opération en cours de route et l'impossibilité dans ce cas d'ouvrir la base (la restauration n'est pas complète).

```
SQL> RECOVER DATABASE
ORA-00279: change 44629 generated at 08/07/2001 16:15:55 needed for thread 1
ORA-00289: suggestion : D:\ORACLE\ORADATA\ORCL\ARCH\ORCL00002.ARC
ORA-00280: change 44629 for thread 1 is in sequence #2
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

--- réponse saisie ---
d:\oracle\oradata\ORCL\temp\ORCL00002.arc

ORA-00279: change 45013 generated at 08/08/2001 06:59:49 needed for thread 1
ORA-00289: suggestion : D:\ORACLE\ORADATA\ORCL\ARCH\ORCL00003.ARC
ORA-00280: change 45013 for thread 1 is in sequence #3
ORA-00278: log file 'd:\oracle\oradata\ORCL\temp\ORCL00002.arc' no longer needed for this
recovery
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

--- réponse saisie ---
cancel
Media recovery cancelled.
SQL> ALTER DATABASE OPEN ;
```

Les fichiers de Redo Log archivés attendus par Oracle peuvent être identifiés grâce à la vue V$RECOVERY_LOG !

Un RECOVER lancé en mode manuel, peut aussi être passé en mode automatique en tapant AUTO sur l'invite.

Si toutes les archives sont à un autre endroit que celui attendu, ou ramenées d'une bande à un autre endroit que celui attendu, il est possible de spécifier cet emplacement puis de lancer le RECOVER en mode automatique.

Le RECOVER utilise alors l'emplacement indiqué comme source des archives, ne demande pas confirmation et applique les archives.

Une source différente de l'emplacement attendu pour les fichiers de Redo Log archivés peut être indiquée (par exemple l'emplacement des fichiers d'archives) :

- Dans la commande RECOVER :

```
RECOVER FROM 'd:\temp' DATABASE
;
```

- Les commandes :

```
SET LOGSOURCE 'd:\temp'
RECOVER DATABASE
;
```

Si toutes les archives ne peuvent pas être ramenées d'un seul coup sur disque, il est possible de travailler série par série :

- ⇨ Récupérer la première série d'archives à un emplacement
- ⇨ Lancer le RECOVER en automatique en spécifiant l'emplacement

- Exemple de séquence pour une restauration à partir d'une source d'archives alternative.

```
SQL> SET LOGSOURCE 'd:\oracle\oradata\test\temp'
SQL> SET AUTORECOVERY ON
SQL> RECOVER DATABASE


ORA-00279: change 44629 generated at 08/07/2001 16:15:55 needed for thread 1
ORA-00289: suggestion : D:\ORACLE\ORADATA\TEST\TEMP\TEST00002.ARC
ORA-00280: change 44629 for thread 1 is in sequence #2


ORA-00279: change 45013 generated at 08/08/2001 06:59:49 needed for thread 1
ORA-00289: suggestion : D:\ORACLE\ORADATA\TEST\TEMP\TEST00003.ARC
ORA-00280: change 45013 for thread 1 is in sequence #3
ORA-00278: log file 'D:\ORACLE\ORADATA\TEST\TEMP\TEST00002.ARC' no longer needed for this
recovery


ORA-00279: change 45371 generated at 08/08/2001 07:00:01 needed for thread 1
ORA-00289: suggestion : D:\ORACLE\ORADATA\TEST\TEMP\TEST00004.ARC
ORA-00280: change 45371 for thread 1 is in sequence #4
ORA-00278: log file 'D:\ORACLE\ORADATA\TEST\TEMP\TEST00003.ARC' no longer needed for this
recovery


ORA-00279: change 45691 generated at 08/08/2001 07:00:15 needed for thread 1
ORA-00289: suggestion : D:\ORACLE\ORADATA\TEST\TEMP\TEST00005.ARC
ORA-00280: change 45691 for thread 1 is in sequence #5
ORA-00278: log file 'D:\ORACLE\ORADATA\TEST\TEMP\TEST00004.ARC' no longer needed for this
recovery

Log applied.
Media recovery complete.
```