

ORACLE
DBA1 10g

Essentiel

Sommaire

| | |
|---|-----------|
| 1. MODULE 01 : INTRODUCTION..... | 7 |
| 1.1. INTRODUCTION | 7 |
| 1.1.1. Les produits Oracle | 7 |
| 1.1.2. Systèmes de Base de données relationnels | 8 |
| 1.1.3. Comment les données sont-elles organisées ? | 9 |
| 1.1.4. Contraintes d'intégrités | 10 |
| 1.1.5. Structured Query Language | 11 |
| 1.1.6. Tâches de l'administrateur de base de données Oracle | 11 |
| 1.2. ARCHITECTURE ORACLE | 12 |
| 1.2.1. Structures de stockage | 12 |
| 1.2.1.1. Introduction..... | 12 |
| 1.2.1.2. Control Files | 13 |
| 1.2.1.3. Les fichiers de Redo Log | 13 |
| 1.2.1.4. Tablespaces et Fichiers de données..... | 14 |
| 1.2.1.5. Segments, Extents et Blocks | 14 |
| 1.2.2. Structures logiques | 15 |
| 1.2.2.1. Oracle Instance Management | 15 |
| 1.2.2.2. Structures de Mémoires Oracle | 16 |
| 1.2.2.3. Processus Oracle | 17 |
| 1.2.3. Le dictionnaire de données..... | 18 |
| 1.2.4. Database/Grid Control..... | 19 |
| 1.2.4.1. Database Control..... | 19 |
| 1.2.4.2. Grid Control..... | 19 |
| 2. MODULE 02: INSTALLER ORACLE ET CREER UNE BASE DE DONNEES..... | 20 |
| 2.1. INSTALLATION D'ORACLE 10G | 20 |
| 2.1.1. Pré requis | 20 |
| 2.1.2. Optimal Flexible Architecture (OFA)..... | 20 |
| 2.1.3. Paramétrage des variables d'environnement | 22 |
| 2.2. ORACLE UNIVERSAL INSTALLER | 23 |
| 2.2.1. Lancement de OUI..... | 23 |
| 2.2.1.1. Pré requis | 23 |
| 2.2.1.2. Oracle Universal Installer | 23 |
| 2.2.2. Configuration d'Oracle..... | 25 |
| 2.2.2.1. Inventory et nom de groupe UNIX | 25 |
| 2.2.2.2. orainstRoot.sh..... | 26 |
| 2.2.2.3. Emplacement des fichiers | 27 |
| 2.2.2.4. Type d'installation | 28 |
| 2.2.2.5. Pré requis nécessaires | 29 |
| 2.2.2.6. Base de données de départ | 30 |
| 2.2.2.7. Mot de passe et résumé | 32 |
| 2.2.3. Fin de l'installation | 33 |
| 2.3. CREATION DE LA BASE DE DONNEES ORACLE | 34 |
| 2.3.1. DBCA | 34 |
| 2.3.2. Créer une base de données..... | 35 |
| 2.3.2.1. Identification de la base de données..... | 36 |
| 2.3.2.2. Options de gestion..... | 37 |
| 2.3.2.3. Gestion des mots de passe..... | 38 |
| 2.3.2.4. Emplacement des fichiers et Backup / Recovery | 39 |
| 2.3.2.5. Paramètres d'initialisation..... | 40 |
| 2.3.2.6. Stockage de la base de données | 41 |
| 2.3.2.7. Les options de création | 42 |
| 2.3.3. Autres Actions avec DBCA..... | 43 |
| 3. MODULE 03 : GESTION DE LA BASE DE DONNEES | 44 |
| 3.1. CONTROLER LA BASE DE DONNEES | 44 |
| 3.1.1. Gestion des outils | 44 |
| 3.1.1.1. Démarrer et arrêter iSQL*Plus..... | 44 |
| 3.1.1.2. Gestion du Framework..... | 45 |
| 3.1.2. Démarrer et arrêter Database Control | 45 |
| 3.1.3. Etapes de démarrage | 46 |

| | |
|--|-----------|
| 3.1.3.1. Démarrer l'instance (NOMOUNT) | 46 |
| 3.1.3.2. Monter la base de données (MOUNT) | 46 |
| 3.1.3.3. Ouverture de la base de données (OPEN) | 47 |
| 3.1.4. <i>Mode de fermeture</i> | 48 |
| 3.1.4.1. Arrêter la base de données | 48 |
| 3.1.4.2. Options d'arrêt : base de données consistante..... | 48 |
| 3.1.4.3. Shutdown Options: Base de données inconsistante..... | 50 |
| 3.1.5. <i>Fichier de paramètres d'initialisation</i> | 51 |
| 3.1.6. <i>Accéder au Database Control</i> | 52 |
| 3.1.6.1. Authentification à Database Control | 52 |
| 3.1.6.2. SYSOPER et SYSDBA..... | 52 |
| 3.1.6.3. Page d'accueil de Database | 53 |
| 3.1.6.4. Démarrage et arrêt de la base de données | 54 |
| 3.1.6.5. Visualiser les logs d'alerte | 54 |
| 3.1.6.6. Modifier le statut du Listener | 55 |
| 3.1.6.7. Fichier de paramètres d'initialisation | 56 |
| 3.2. STRUCTURES DE STOCKAGE..... | 57 |
| 3.2.1. <i>Tablespaces et fichiers de données</i> | 57 |
| 3.2.2. <i>Gestion des tablespaces</i> | 58 |
| 3.2.2.1. Création d'un nouveau tablespace..... | 58 |
| 3.2.2.2. Espace de stockage des tablespaces gérés localement..... | 59 |
| 3.2.2.3. Tablespace dans une base de données pré configurer | 61 |
| 3.2.2.4. Modification d'un tablespace | 61 |
| 3.2.2.5. Actions sur les tablespaces..... | 63 |
| 3.2.2.6. Suppression d'un tablespace | 64 |
| 3.2.2.7. Information sur un tablespace | 65 |
| 4. MODULE 4 : GESTION DES UTILISATEURS ET GESTION DES OBJETS..... | 66 |
| 4.1. CREATION ET GESTION DES COMPTES UTILISATEURS | 66 |
| 4.1.1. <i>Comptes utilisateur de la base de données</i> | 66 |
| 4.1.2. <i>Création d'un utilisateur</i> | 67 |
| 4.2. CONTROLE DES RESSOURCES POUR LES UTILISATEURS | 68 |
| 4.2.1. <i>Profils et utilisateurs</i> | 68 |
| 4.2.2. <i>Authentification des utilisateurs</i> | 69 |
| 4.2.3. <i>Tablespace par défaut, tablespace temporaire et verrouillage</i> | 70 |
| 4.2.4. <i>Utilisateurs et schéma de la base de données</i> | 71 |
| 4.2.5. <i>Assignment de quota à un utilisateur</i> | 71 |
| 4.3. CREER ET GERER DES ROLES | 72 |
| 4.3.1. <i>Avantage des rôles</i> | 73 |
| 4.3.2. <i>Rôles prédéfinis</i> | 73 |
| 4.3.3. <i>Sécurité des rôles</i> | 74 |
| 4.3.4. <i>Assignment de rôles aux utilisateurs</i> | 75 |
| 4.4. PRIVILEGES GRANT ET REVOKE..... | 75 |
| 4.4.1. <i>Privilèges systèmes</i> | 76 |
| 4.4.2. <i>Privilèges objets</i> | 77 |
| 4.5. GESTION DES OBJETS DU SCHEMA | 78 |
| 4.5.1. <i>Qu'est-ce qu'un schéma ?</i> | 78 |
| 4.5.2. <i>Nomenclature des noms des objets de la base de données</i> | 80 |
| 4.5.3. <i>Les namespaces</i> | 80 |
| 4.5.4. <i>Types de données spécifique dans une table</i> | 81 |
| 4.5.5. <i>Création et modification d'une table</i> | 82 |
| 4.5.5.1. Création d'une table..... | 82 |
| 4.5.5.2. Modification d'une table..... | 82 |
| 4.5.6. <i>Gestion des contraintes</i> | 83 |
| 4.5.6.1. Intégrité des données..... | 83 |
| 4.5.6.2. Définitions des contraintes..... | 84 |
| 4.5.7. <i>Attribue d'une table</i> | 84 |
| 4.5.8. <i>Contenu d'une table</i> | 85 |
| 4.5.9. <i>Action avec les tables</i> | 85 |
| 4.5.10. <i>Création d'indexe</i> | 87 |
| 4.5.11. <i>Qu'est-ce qu'une vue ?</i> | 87 |
| 4.5.12. <i>Qu'est-ce qu'une séquence ?</i> | 88 |
| 5. MODULE 05 : GESTION DES DONNEES | 90 |

| | |
|---|------------|
| 5.1. DATA PUMP | 90 |
| 5.1.1. Gestion des données avec le SQL | 90 |
| 5.1.1.1. Introduction | 90 |
| 5.1.1.2. L'ordre INSERT | 90 |
| 5.1.1.3. L'ordre UPDATE | 91 |
| 5.1.1.4. L'ordre DELETE | 91 |
| 5.1.1.5. Les ordres COMMIT et ROLLBACK | 91 |
| 5.1.1.6. Contraintes d'intégrité et DML | 92 |
| 5.1.2. Exportation de données avec Data Pump | 93 |
| 5.1.3. Importation de données avec Data Pump | 94 |
| 5.1.4. Les objets de type répertoire | 95 |
| 5.2. CHARGER DES DONNEES AVEC SQL*LOADER | 97 |
| 5.2.1. Présentation | 97 |
| 5.2.2. Le fichier de contrôle de SQL*Loader | 98 |
| 5.2.3. Les méthodes de chargement des données | 100 |
| 5.2.3.1. Les deux méthodes | 100 |
| 5.2.3.2. Différences entre chargement direct et chargement conventionnel | 101 |
| 5.2.4. Chargement des données | 102 |
| 5.2.4.1. Données à charger et fichier de données | 102 |
| 5.2.4.2. Charger des données avec SQL*Loader | 105 |
| 6. MODULE 06 : SECURITE | 106 |
| 6.1. PARAMETRES ET PRIVILEGES | 106 |
| 6.1.1. Sécurité de la base de données | 106 |
| 6.1.1.1. Présentation | 106 |
| 6.1.1.2. Principe du moindre privilège | 106 |
| 6.1.2. Paramètres de sécurité | 106 |
| 6.1.2.1. Protection du dictionnaire de données | 106 |
| 6.1.2.2. Restreindre aux utilisateurs l'accès aux répertoires du système d'exploitation | 107 |
| 6.1.2.3. Désactiver l'authentification par système d'exploitation à distance | 107 |
| 6.1.3. Privilèges | 108 |
| 6.1.3.1. Retirer les privilèges non nécessaires à PUBLIC | 108 |
| 6.1.3.2. Limiter les utilisateurs ayant des privilèges administratives | 109 |
| 6.2. PROFILS | 110 |
| 6.2.1. Fonctionnalités | 110 |
| 6.2.1.1. Verrouillage du mot de passe | 110 |
| 6.2.1.2. Durée et expiration du mot de passe | 111 |
| 6.2.1.3. Historique des mots de passe | 111 |
| 6.2.1.4. Vérification du mot de passe | 112 |
| 6.2.2. Gestion des profils | 112 |
| 6.2.2.1. Création d'un profil de mot de passe | 112 |
| 6.2.2.2. Assigner un profil à un utilisateur | 113 |
| 6.2.2.3. Gestion des comptes par défaut | 113 |
| 6.3. AUDIT | 114 |
| 6.3.1. Surveillance d'activité suspecte | 114 |
| 6.3.1.1. Comparaison des outils d'audit | 114 |
| 6.3.1.2. Mise à jour de sécurité | 114 |
| 6.3.2. Audit standard de la base de données | 115 |
| 6.3.2.1. Mécanisme de l'audit standard de la base de données | 115 |
| 6.3.2.2. Spécification des options d'audit | 116 |
| 6.3.2.3. Visualiser les options d'audit | 117 |
| 6.3.2.4. Visualiser les résultats d'audit | 117 |
| 6.3.3. Audit basé sur des valeurs | 118 |
| 6.3.4. Fine-Grained Auditing (FGA) | 119 |
| 6.3.4.1. Stratégie d'audit FGA | 120 |
| 6.3.4.2. Le package DBMS_FGA | 121 |
| 6.3.4.3. Consultation des événements d'audit | 122 |
| 6.3.4.4. Les directives concernant la FGA | 122 |
| 6.3.4.5. Audit des utilisateurs de type SYSDBA et SYSOPER | 123 |
| 7. MODULE 07 : ORACLE NET SERVICES | 124 |
| 7.1. ORACLE NET LISTENER | 124 |
| 7.1.1. Définition | 124 |
| 7.1.2. Gestion du listener | 125 |
| 7.1.2.1. Utilitaire de contrôle du processus d'écoute (lsnrctl) | 125 |
| 7.1.2.2. Syntaxe du lsnrctl | 125 |

| | |
|--|------------|
| 7.1.2.3. Surveillance avec le lsnrctl | 127 |
| 7.1.3. <i>Création du listener</i> | 128 |
| 7.1.3.1. Création d'un listener..... | 128 |
| 7.1.3.2. Adresses d'écoute | 129 |
| 7.1.3.3. Configuration des paramètres optionnels | 130 |
| 7.1.4. <i>Enregistrement d'une base de données</i> | 131 |
| 7.2. CONNEXIONS RESEAU | 132 |
| 7.2.1. <i>Introduction</i> | 132 |
| 7.2.2. <i>Méthodes de connexion</i> | 132 |
| 7.2.2.1. Easy Connect | 132 |
| 7.2.2.2. Résolution locale (Local Naming) | 133 |
| 7.2.2.3. Résolution par répertoire (Directory Naming) | 133 |
| 7.2.3. <i>Configuration des connexions</i> | 134 |
| 7.2.3.1. Configuration les alias pour les services | 134 |
| 7.2.3.2. Options avancées | 135 |
| 7.3. ORACLE NET MANAGER | 136 |
| 7.3.1. <i>Présentation</i> | 136 |
| 7.3.1.1. Généralités | 136 |
| 7.3.1.2. Choisir une méthode de résolution | 137 |
| 7.3.2. <i>Configurations</i> | 138 |
| 7.3.2.1. Configuration les alias pour les services | 138 |
| 7.3.2.2. Options avancées | 139 |
| 7.3.3. <i>Oracle Net Connectivity</i> | 139 |
| 8. MODULE 08 : SERVEUR PARTAGE | 140 |
| 8.1. LES SESSIONS..... | 140 |
| 8.1.1. <i>Définition</i> | 140 |
| 8.1.2. <i>Processus serveur dédié</i> | 140 |
| 8.1.3. <i>Processus serveur partagé</i> | 141 |
| 8.1.4. <i>SGA et PGA</i> | 142 |
| 8.2. CONFIGURATION D'UN SERVEUR PARTAGE | 143 |
| 8.2.1. <i>DISPATCHERS</i> | 143 |
| 8.2.2. <i>SHARED_SERVERS</i> | 144 |
| 8.2.3. <i>MAX_SHARED_SERVERS</i> | 145 |
| 8.2.4. <i>CIRCUITS</i> | 145 |
| 8.2.5. <i>SHARED_SERVER_SESSIONS</i> | 146 |
| 8.3. MAINTENIR UN SERVEUR PARTAGE..... | 147 |
| 8.3.1. <i>Vérifier la configuration d'un serveur partagé</i> | 147 |
| 8.3.2. <i>Vues du dictionnaire de données</i> | 147 |
| 8.3.3. <i>Choisir un type de connexion</i> | 148 |
| 8.3.4. <i>Quand est ce qu'il ne faut pas utiliser un serveur partagé</i> | 148 |
| 9. MODULE 09 : SURVEILLANCE ET GESTION | 149 |
| 9.1. SUPERVISION DE LA PERFORMANCE..... | 149 |
| 9.1.1. <i>Supervision de la performance</i> | 149 |
| 9.1.2. <i>Méthodologie de supervision</i> | 149 |
| 9.1.3. <i>Métriques de la base de données et de l'instance</i> | 150 |
| 9.1.4. <i>Métriques du dictionnaire des données</i> | 150 |
| 9.1.5. <i>Objets INVALID et UNUSABLE</i> | 151 |
| 9.1.6. <i>Statistiques pour l'Optimizer</i> | 152 |
| 9.1.6.1. Collecter des statistiques manuellement..... | 153 |
| 9.1.6.2. Automatiser la collecte de statistiques | 154 |
| 9.1.6.3. Planifier la collecte de statistiques | 155 |
| 9.1.7. <i>Vues dynamiques de performance</i> | 155 |
| 9.1.8. <i>Obtenir des informations sur les performances</i> | 156 |
| 9.1.9. <i>Réagir aux problèmes de performance</i> | 157 |
| 9.2. MAINTENANCE PROACTIVE..... | 158 |
| 9.2.1. <i>Gestion des seuils d'alertes</i> | 158 |
| 9.2.1.1. Alertes générées par le serveur | 158 |
| 9.2.1.2. Seuils d'alertes..... | 159 |
| 9.2.1.3. Paramétrer les seuils d'alertes..... | 159 |
| 9.2.1.4. Mesure à l'aide des Baselines | 160 |
| 9.2.1.5. Utilisation des Baselines | 161 |
| 9.2.2. <i>SQL Tuning et autres outils</i> | 161 |

| | |
|--|------------|
| 9.2.2.1. SQL Tuning et Access Advisor..... | 162 |
| 9.2.2.2. Memory Advisor..... | 163 |
| 9.2.2.3. Segment Advisor..... | 163 |
| 9.2.3. <i>Automatic Workload Repository</i> | 164 |
| 9.2.3.1. Gestion des AWR..... | 164 |
| 9.2.4. <i>Automatic Database Diagnostic Monitor (ADDM)</i> | 165 |
| 9.2.4.1. Résultats d'ADDM..... | 166 |
| 9.2.4.2. Les recommandations d'ADDM..... | 166 |
| 10. MODULE 10 : GESTION DES UNDO | 167 |
| 10.1. CONCEPTS..... | 167 |
| 10.1.1. <i>Les données undo</i> | 167 |
| 10.1.2. <i>Les transactions et les undos</i> | 167 |
| 10.1.3. <i>Stockage des informations undo</i> | 168 |
| 10.2. GESTION DES UNDO..... | 169 |
| 10.2.1. <i>Administration des undos</i> | 169 |
| 10.2.2. <i>Rétention des undos</i> | 170 |
| 10.2.2.1. Configuration de la rétention des undos..... | 170 |
| 10.2.2.2. Garantir la rétention des undos..... | 170 |
| 10.2.3. <i>Tablespaces undo</i> | 171 |
| 10.2.3.1. Taille des tablespaces undo..... | 171 |
| 10.2.3.2. Utilisation de Undo Advisor..... | 172 |
| 11. MODULE 11 : SAUVEGARDE ET RECUPERATION | 173 |
| 11.1. CONCEPTS..... | 173 |
| 11.1.1. <i>Présentation</i> | 173 |
| 11.1.2. <i>Echecs</i> | 173 |
| 11.1.2.1. Types d'échecs..... | 173 |
| 11.1.2.2. Echecs de requêtes..... | 174 |
| 11.1.2.3. Echecs de processus utilisateurs..... | 174 |
| 11.1.2.4. Echecs de réseau..... | 175 |
| 11.1.2.5. Erreurs d'utilisateurs..... | 175 |
| 11.1.2.6. Echecs d'instances..... | 176 |
| 11.1.2.7. Echecs Média..... | 176 |
| 11.2. RECUPERATION D'UNE INSTANCE..... | 177 |
| 11.2.1. <i>Concepts de récupération</i> | 177 |
| 11.2.2. <i>Tuning</i> | 177 |
| 11.3. CONFIGURATION DE LA RECUPERATION..... | 179 |
| 11.3.1. <i>Règles à suivre</i> | 179 |
| 11.3.2. <i>Fichiers de contrôle</i> | 179 |
| 11.3.3. <i>Fichiers de Redo Log</i> | 180 |
| 11.3.4. <i>Multiplexage</i> | 180 |
| 11.3.5. <i>Fichiers de Redo Log archivés</i> | 181 |
| 11.3.5.1. Présentation..... | 181 |
| 11.3.5.2. Nommage et chemins des fichiers de Log archivés..... | 182 |
| 11.3.5.3. Mode ARCHIVELOG..... | 183 |

1. Module 01 : Introduction

1.1. Introduction

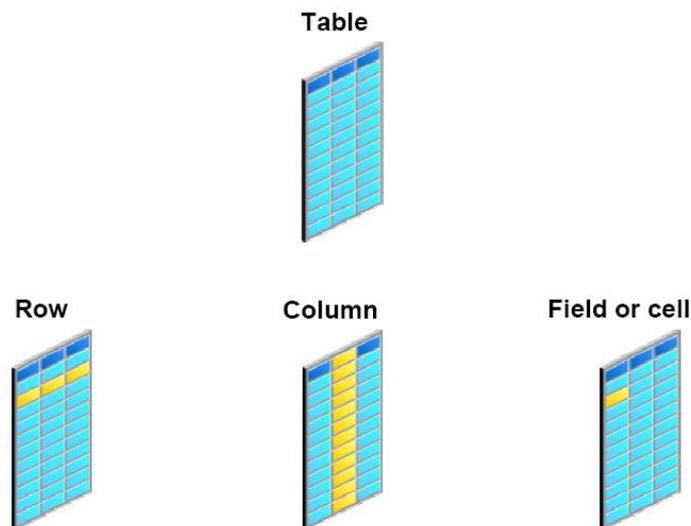
1.1.1. Les produits Oracle

- **Oracle Database** : Oracle Database est la première base de données conçue pour le Grid Computing en entreprise, la manière la plus flexible et la plus rentable pour gérer les informations et les applications. Oracle Database est disponible en trois éditions – Entreprise, Standard et Personnel.
- **Oracle Application Server** : Serveur Oracle certifié Java 2 Enterprise Edition (J2EE). Il intègre les outils nécessaires pour développer et déployer des applications Web. Le serveur d'application déploie les portails e-business, Web services, et des applications transactionnelles incluant PL/SQL, Oracle Forms et les applications basées sur J2EE.
- **Oracle Applications** : Oracle E-Business Suite est un ensemble complet d'applications permettant la gestion et l'automatisation des processus dans votre entreprise.
- **Oracle Collaboration Suite** : Oracle Collaboration Suite est un système simple et intégré pour toutes les données concernant les communications de votre entreprise : voix, e-mail, fax, wireless, calendrier et les fichiers.
- **Oracle Development Suite** : Oracle Developer Suite est un environnement complet et intégré qui combine le développement d'applications et d'utilitaires business intelligent.
- **Oracle Services** : Les services comme Oracle Consulting et Oracle University fournissent l'expertise dont vous avez besoin pour faire fonctionner votre projet Oracle correctement.

1.1.2. Systèmes de Base de données relationnels

En juin 1970, le Docteur E. F. Codd a publié l'article « A relational Model of Data for Large Shared Data Banks », dans le journal Association of Computer Machinery (ACM), *Communications of the ACM*. Le modèle de Codd est maintenant accepté comme le modèle définitif pour la gestion des bases de données relationnelles (SGBDR).

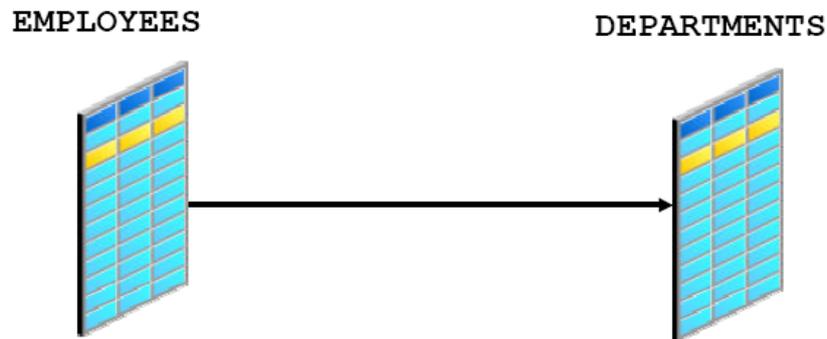
Le langage appelé Structured English Query Language (SEQUEL), a été développé par IBM Corporation, Inc., pour utiliser le modèle de Codd. SEQUEL est devenu plus tard SQL (toujours prononcé « SEQUEL »). En 1979, Relational Software, Inc. (aujourd'hui Oracle Corporation) a introduit la première version commerciale de SQL. Aujourd'hui, SQL est accepté comme le langage standard des SGBDR.



L'élément de base d'un système de base de données relationnelle est une table à deux dimensions. Une table est constituée de zéro ou plusieurs lignes de données. Chaque ligne contient une ou plusieurs colonnes. Chaque colonne d'une ligne est appelée une cellule ou un champ de données.

Chaque ligne de données est une collection de données relative à un sujet donné. Par exemple, vous pourriez avoir une table qui stocke les informations des employés dans votre compagnie. Chaque ligne représente un employé de la compagnie, et les colonnes peuvent être des données tels que le nom, le prénom, l'e-mail, le numéro de téléphone, la date d'embauche etc.... Chaque colonne a un nom et un type de données. Le type de données définit quel type de données peuvent être stockées dans la colonne, par exemple les nombres et les caractères.

1.1.3. Comment les données sont-elles organisées ?

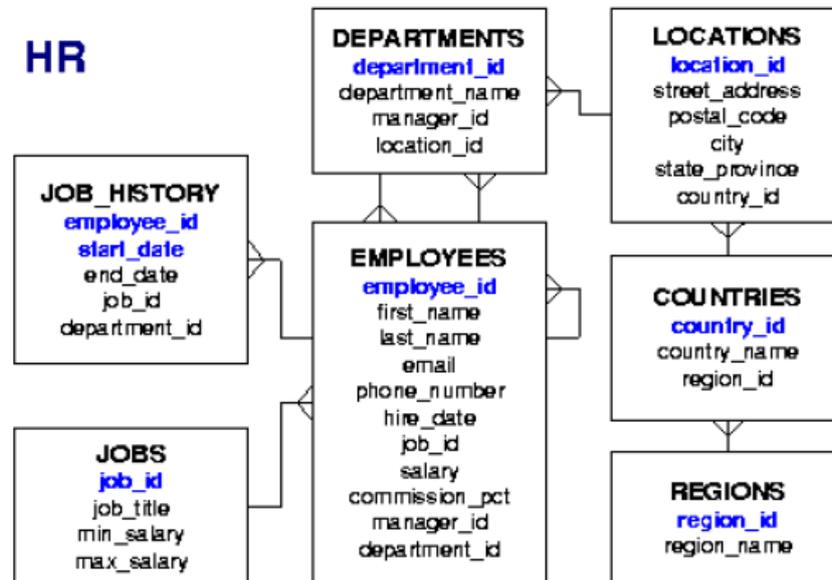


Une base de données relationnelle a plusieurs tables, qui peuvent être utilisées indépendamment ou jointes ensemble en utilisant une ou plusieurs colonnes communes entre les tables. Avec Oracle Database 10g vous pourrez avoir besoin que la colonne utilisée pour établir la relation avec une autre table contienne les valeurs qui existent déjà dans cette table. Cette relation obligatoire est connue comme une contrainte d'intégrité. En utilisant l'exemple des employés, chaque employé dans votre compagnie est assigné à un département particulier. Etant donné qu'un département doit avoir plusieurs employés, nous ne pouvons pas stocker les informations concernant le département dans chaque ligne de la table **employees**. La raison est simple, nous allons avoir les informations redondantes.

A la place, chaque ligne de la table **employees** possède une colonne qui sert d'identifiant pointant vers la table **departments**. Cela nous permet de joindre la table **employees** avec la table **departments** et de voir auquel département est assigné un employé.

Cet identifiant est un champ clé. Dans la table **employees**, le champ concernant les départements est appelé une clé étrangère (foreign key), car elle pointe vers une ligne d'une autre table. Dans la table **departments**, le champ référencé par une clé étrangère dans la table **employees** est la clé primaire (primary key), ou quelquefois la clé alternée (alternate key). Le champ correspondant à cette clé primaire doit avoir certaines propriétés dans la table **departments**, il doit être unique et ne doit jamais avoir une valeur NULL. Ce qui signifie que chaque ligne de la table **employees** pointera vers une et une seule ligne de la table **departments**.

1.1.4. Contraintes d'intégrités



Vous pouvez utiliser les contraintes pour définir les règles de votre base de données relationnelle. Le diagramme montre un modèle conceptuel de données (MCD) d'un service de ressources humaines (HR). Le MCD montre les tables et les relations entre elles. Le MCD est utilisé pour définir les contraintes qui seront utilisées pour garantir la validité des données stockées dans les tables de la base de données.

Les contraintes d'intégrités sont définies en tant qu'élément de définition des tables dans le dictionnaire de données, donc toutes les applications accédant à la table devront adhérer aux règles définies par les contraintes. Lorsque vous avez besoin de changer une règle, vous devez seulement la modifier au niveau de la base de données et non dans chaque application qui y accède.

Oracle Database 10g supporte les types de contraintes suivantes :

- **NOT NULL** : Interdit les valeurs nulles (entrées vides) dans la colonne de la table.
- **UNIQUE** : Interdit la duplication d'une valeur dans une ou plusieurs colonnes.
- **PRIMARY KEY** : Interdit la duplication et la valeur nulle dans une ou plusieurs colonnes.
- **FOREIGN KEY** : Exige que chaque valeur dans une colonne ou un jeu de colonnes corresponde avec une valeur des tables relationnelles **UNIQUE** ou **PRIMARY KEY**. Les contraintes d'intégrité **FOREIGN KEY** définissent également les actions d'intégrité référentielle qui indique ce qu'Oracle devrait faire sur des données dépendantes si les données sur lesquelles elles font références sont modifiées.
- **CHECK** : Interdit les valeurs qui ne satisfont pas l'expression logique de la contrainte.

1.1.5. Structured Query Language

Structured Query Language (SQL) est l'ensemble des requêtes permettant à tous les utilisateurs et applications d'accéder aux données de la base de données. Les programmes d'applications et les utilitaires Oracle permettent souvent aux utilisateurs d'accéder à la base de données sans utiliser directement SQL, mais ces applications doivent bien entendu obligatoirement utiliser SQL lorsqu'elles exécutent des requêtes utilisateur.

SQL permet de travailler avec les données au niveau logique. Par exemple, pour récupérer l'ensemble des lignes d'une table, vous utilisez une condition qui sert à filtrer les lignes. Toutes les lignes satisfaisant la condition sont retournées dans un premier temps et peuvent être passées comme une seule unité à une application ou à une autre requête SQL. Vous n'avez pas besoin de savoir comment les lignes sont stockées physiquement ou recherchées. Toutes les requêtes SQL utilisent Oracle Database optimizer pour déterminer la méthode la plus efficace pour accéder aux données.

1.1.6. Tâches de l'administrateur de base de données Oracle

Un administrateur de base de données Oracle (DBA) est responsable de l'installation du logiciel Oracle et de la création de la base de données. Vous pouvez être responsable de la création des structures de stockage de la base de données comme les tablespaces. De plus, vous pouvez créer le schéma ou l'ensemble des objets qui contient les données d'applications. Vous devez vous assurer que la base de données est disponible pour les utilisateurs. Vous pouvez faire ceci en démarrant la base de données, en sauvegardant la base de données de façon régulière, et en surveillant les performances de la base de données.

Approche prioritaire pour concevoir, mettre en application et maintenir une base de données Oracle :

1. Evaluer le matériel du serveur de base de données
2. Installer le logiciel Oracle
3. Concevoir la base de données
4. Créer et ouvrir la base de données
5. Sauvegarder la base de données
6. Inscrire les utilisateurs système
7. Mettre en application la conception de la base de données
8. Restaurer en cas d'erreur de la base de données
9. Surveiller les performances de la base de données

1.2. Architecture Oracle

1.2.1. Structures de stockage

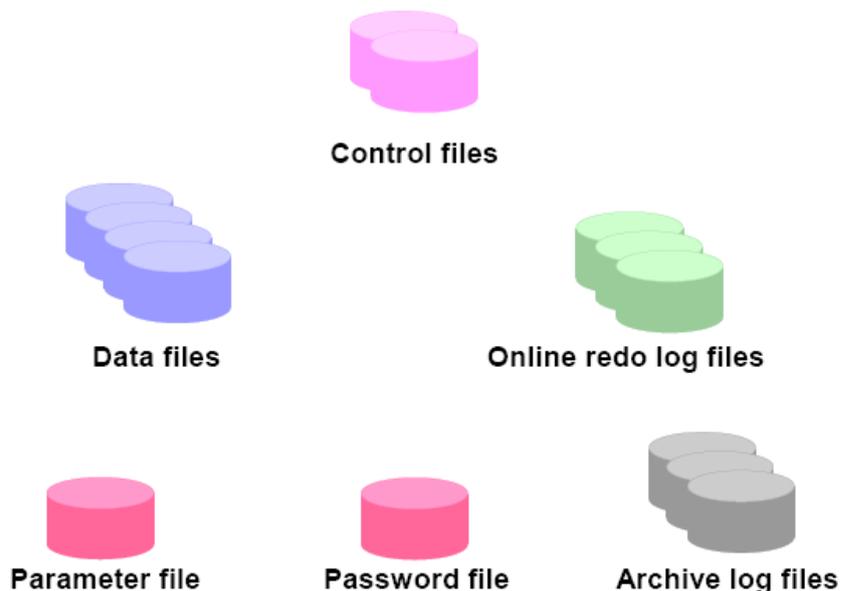
1.2.1.1. Introduction

Les fichiers qui constituent une base de données Oracle sont organisés de cette façon :

- **Control files** : Contiennent les données à propos de la base de données elle-même, appelés les méta-données. Ces fichiers sont critiques pour la base de données. Sans eux, vous ne pourrez pas ouvrir les fichiers de données et ainsi accéder aux données de la base de données.
- **Data files** : Contiennent les données de la base de données.
- **Online redo log files** : Permettent la récupération de l'instance de la base de données. Si la base de données crash et ne perd aucun fichier de données, l'instance peut récupérer la base de données grâce aux informations contenues dans ces fichiers.

Il existe d'autres fichiers qui ne font pas parti intégrante de la base de données mais ils sont importants pour le fonctionnement de la base :

- **Parameter file** : Utilisé pour définir comment l'instance sera configurée au démarrage.
- **Password file** : Permet aux utilisateurs de se connecter à distance à la base et d'effectuer des tâches administratives.
- **Archive log files** : Contiennent un historique des fichiers de redo générés par l'instance. Ces fichiers sont utilisés pour la récupération de la base de données ; en utilisant ces fichiers et une sauvegarde de la base de données, vous pouvez récupérer des fichiers perdus.



1.2.1.2. Control Files

Lorsque vous démarrez l'instance et montez la base de données, le fichier de contrôle est lu. Les entrées du fichier de contrôle spécifient les fichiers physiques que la base de données comprend.

Lorsque vous ajoutez des fichiers supplémentaires à votre base de données, le fichier de contrôle est automatiquement mis à jour. L'emplacement des fichiers de contrôle est spécifié dans un paramètre d'initialisation.

Pour éviter un éventuel échec de la base de données dû à la perte du fichier de contrôle, vous devez le multiplexer sur au moins trois disques physiques. En spécifiant dans le fichier d'initialisation les différents fichiers de contrôle, vous permettez au serveur Oracle de maintenir plusieurs copies du fichier de contrôle.

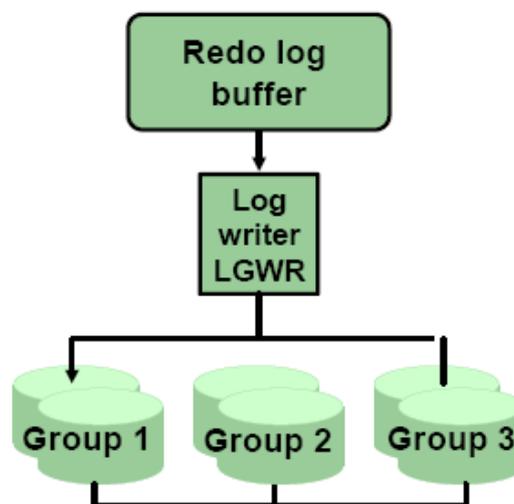


Control files

1.2.1.3. Les fichiers de Redo Log

Les fichiers de redo log sont utilisés pour enregistrer les changements sur la base de données comme le résultat d'une transaction ou d'une action interne au serveur Oracle. Ils protègent la base de données contre une perte d'intégrité de la base due à une panne du système, d'un disque ou autre. Les fichiers de redo log doivent être multiplexés pour éviter de perdre les informations qu'ils stockent dans le cas d'une panne d'un disque.

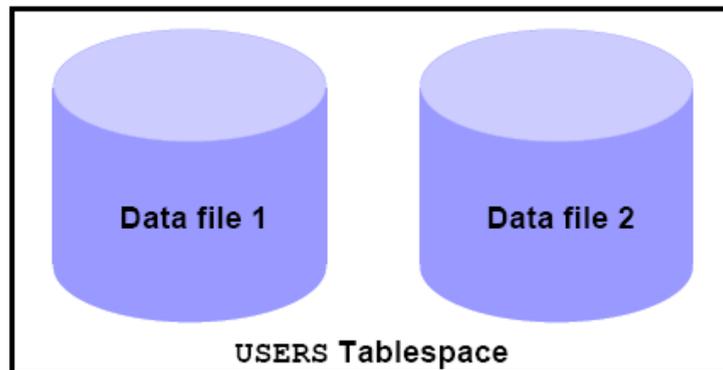
Les redo log sont organisés en groupe de fichiers de redo log. Un groupe est composé d'un fichier de redo log et de ces copies multiplexées. Chaque copie d'un fichier est appelée un membre et chaque groupe est identifié par un numéro. Le processus log writer (**LGWR**) écrit les enregistrements de redo depuis le redo log buffer dans un groupe de redo log jusqu'à ce que le fichier soit plein ou qu'un log switch soit exécuté. Il passe ensuite au groupe suivant. Les groupes de redo log sont utilisés de manière circulaire.



1.2.1.4. Tablespaces et Fichiers de données

Une base de données est divisée en unités de stockage logique appelées tablespaces, qui peuvent être utilisés pour regrouper des structures logiques ensemble. Chaque base de données est divisée de façon logique en un ou plusieurs tablespaces. Un ou plusieurs fichiers de données sont explicitement créés pour chaque tablespace afin de stocker les données physiques de toutes les structures logiques dans un tablespace.

Remarque : Vous pouvez également créer des tablespaces de type `bigfile`, qui sont constitué d'un seul, mais gros (plus de 4 Go) fichier de données. Le type par défaut est `smallfile`, les tablespaces de ce type peuvent contenir plusieurs fichiers de données mais pas aussi gros.



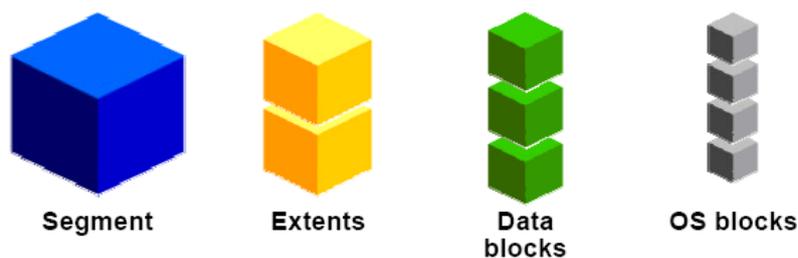
1.2.1.5. Segments, Extents et Blocks

Les objets de la base de données comme les tables et les index sont stockés dans des tablespaces en segments. Chaque segment contient un ou plusieurs extents. Un extent est composé de blocs de données contigus, ce qui signifie que chaque extent ne peut exister que dans un seul fichier de données. Le bloc de données est la plus petite unité d'entrée/sortie de la base de données.

Lorsque la base de données demande un ensemble de blocs de données en provenance de l'OS, celui-ci les attribue à ses blocs sur un périphérique de stockage. De ce fait, vous n'avez pas à vous soucier de l'adresse physique des données de votre base. Cela veut aussi dire qu'un fichier de données peut être striped ou mirrored sur plusieurs disques.

La taille d'un bloc de donnée peut être définie à la création de la base de données. La taille par défaut de 8 K est idéale pour de nombreuses bases de données. Si votre base de données supporte le data warehousing qui correspond à de large tables et index, alors une taille de bloc plus large peut être avantageuse. Si votre base supporte les applications transactionnelles où les lectures et écritures sont très aléatoires, alors un bloc de taille plus petite peut être avantageux. La taille maximale d'un bloc dépend de votre OS. La taille minimum est de 2 K et est rarement (voir jamais) utilisée.

Vous pouvez avoir des tablespaces avec des différentes tailles de bloc. Généralement, ceci est utilisé pour pouvoir transporter les tablespaces.



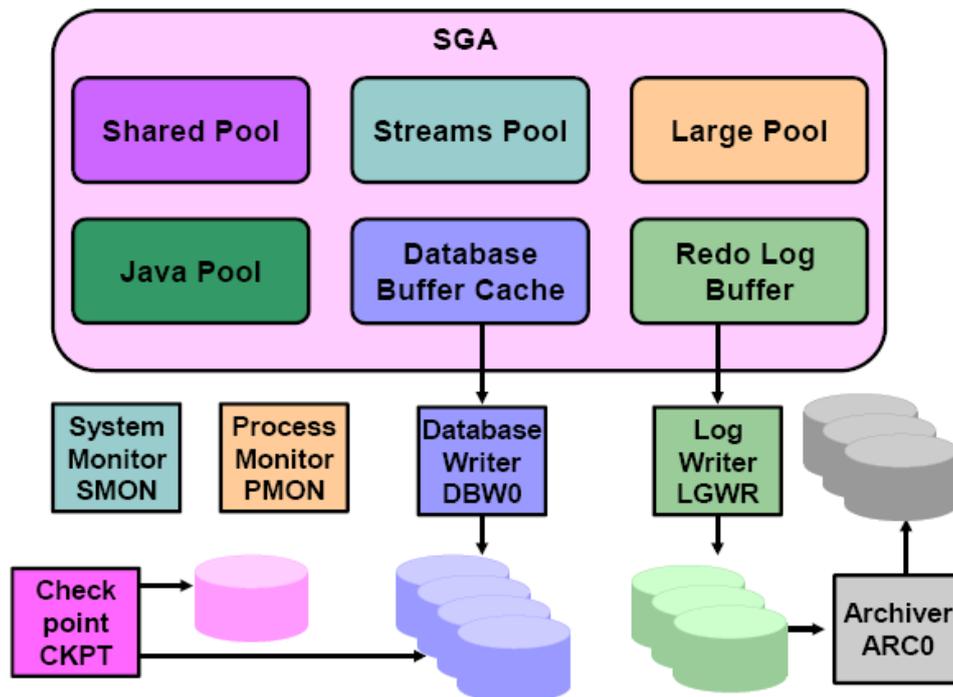
1.2.2. Structures logiques

1.2.2.1. Oracle Instance Management

Un serveur de base de données Oracle est composé d'une base de données Oracle et d'une instance Oracle. Une instance Oracle est composée de plusieurs structures mémoire connu sous le nom de System Global Area (SGA) et des processus d'arrière plan.

L'instance n'existe pas avant son démarrage. Lorsque l'instance est démarrée, le fichier d'initialisation est lu et l'instance est configurée selon les paramètres du fichier d'initialisation.

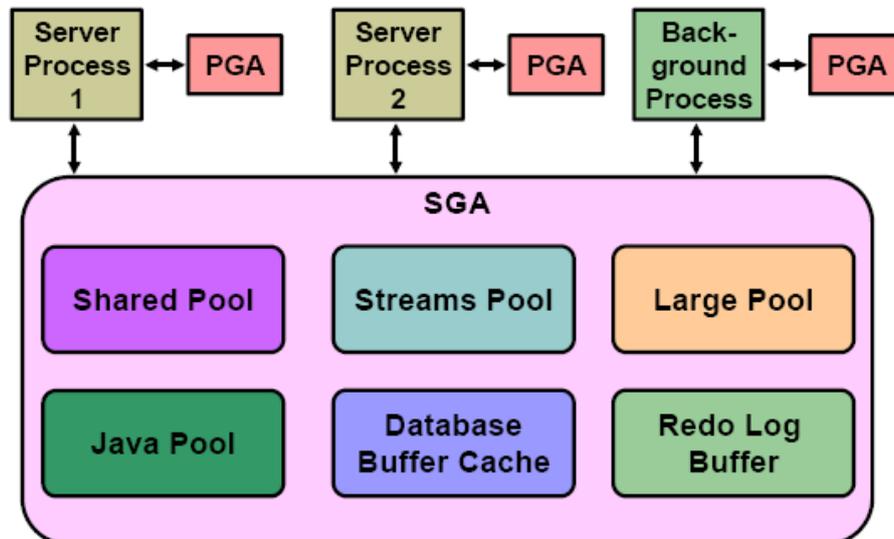
Une fois l'instance démarrée et la base de données ouverte, les utilisateurs peuvent accéder à la base de données.



1.2.2.2. Structures de Mémoires Oracle

Les structures mémoires de base associées à une instance Oracle sont :

- **System Global Area (SGA)** : Partagée par tous les processus serveurs et d'arrière plan.
- **Program Global Area (PGA)** : Privée pour chaque processus serveur et d'arrière plan ; il y a une PGA pour chaque processus.



La SGA est une zone mémoire partagée qui contient des données et des informations de contrôle de l'instance.

La SGA est constituée des éléments suivants :

- **Database buffer cache** : Garde en cache les blocs de données récupérés de la base de données.
- **Redo log buffer** : Garde en cache les informations de redo (utilisés pour récupérer l'instance) jusqu'à ce qu'elles soient écrites dans le fichier de redo log stocké sur le disque.
- **Shared pool** : Garde en cache différentes informations pouvant être partagées par plusieurs utilisateurs.
- **Large pool** : Zone mémoire optionnelle utilisée pour les grandes demandes d'entrées/sorties.
- **Java pool** : Utilisé pour toutes les sessions spécifiques à du code Java et aux données de la Machine Virtuel Java (JVM).
- **Streams pool** : Utilisé par Oracle Streams.

Lorsque vous démarrez l'instance en utilisant Enterprise Manager or SQL*Plus, la mémoire est allouée pour la SGA.

Avec une configuration dynamique de la SGA, la taille du **Database buffer cache**, du **Shared pool**, du **Large pool**, du **Java pool** et du **Streams pool** change sans avoir besoin d'éteindre l'instance.

La base de données pré configurée a été réglée avec les paramètres adéquats pour les structures mémoires. Cependant, si l'utilisation de votre base de données augmente vous pouvez avoir besoin de modifier les valeurs des paramètres de la mémoire.

Oracle fournit des alertes et des conseils pour identifier les problèmes de taille de la mémoire et vous aide à déterminer de nouvelles valeurs plus appropriées.

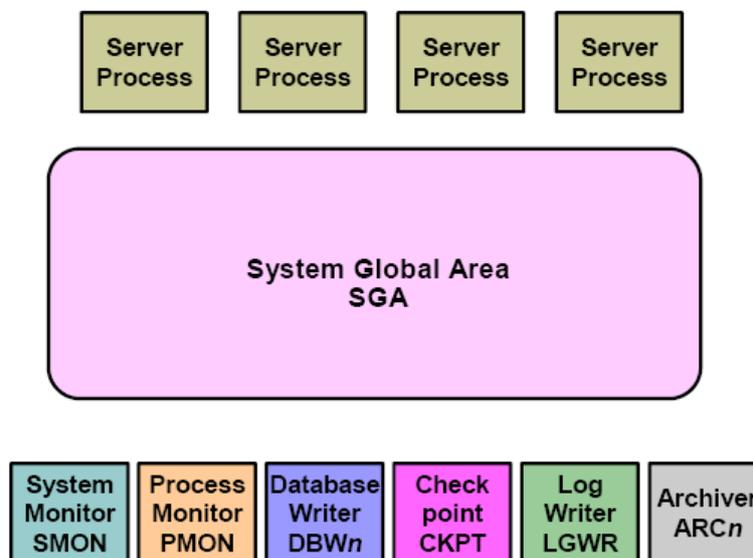
La PGA est une zone mémoire qui contient les données et contrôle les informations de chaque processus serveur. Un processus serveur est un processus qui reçoit les requêtes du client. Chaque processus serveur a sa propre zone PGA privée qui est créée lorsque le processus serveur est lancé. L'accès à cette zone est exclusivement réservé à ce processus serveur, et il n'est lu et écrit que par le code Oracle agissant en son nom. La quantité de mémoire PGA utilisé et son contenu dépendent de la configuration de l'instance, en mode partagé ou non.

Généralement, la PGA contient ceci :

- **Private SQL area** : Contient les données comme les informations de correspondances et le temps d'exécution des structures mémoires. Chaque session faisant des requêtes SQL a une zone privée SQL.
- **Session memory** : Mémoire allouée aux variables de sessions et aux autres informations relatives à la session.

1.2.2.3. Processus Oracle

Lorsque vous lancez une application ou un utilitaire Oracle comme Enterprise Manager, le serveur Oracle crée un processus serveur pour exécuter les commandes issues de l'application.



Oracle crée également plusieurs processus d'arrière plan pour une instance pour qu'ils interagissent les uns avec les autres et avec le système d'exploitation pour gérer les structures mémoires, effectuer des écritures de données asynchrones d'entrées/sorties sur le disque, et d'effectuer une gestion interne générale.

Quelques processus d'arrière plan dépendent des dispositifs utilisés dans la base de données. Les processus d'arrière plan les plus communs sont :

- **System monitor (SMON)** : Exécute la récupération de l'instance après le crash de celle-ci.
- **Process monitor (PMON)** : Exécute le nettoyage des processus lorsqu'un processus utilisateur échoue.
- **Database Writer (DBWn)** : Écrit les blocs modifiés du **database buffer cache** vers les fichiers stockés sur le disque.
- **Checkpoint (CKPT)** : Envoi un signal au processus **DBWn** lors d'un checkpoint et met à jour tous les fichiers de données et les fichiers de contrôle de la base de données afin d'indiquer le checkpoint le plus récent.
- **Log writer (LGWR)** : Écrit les entrées de redo log sur le disque.
- **Archiver (ARCn)** : Copie les fichiers de redo log dans un espace d'archivage quand les fichiers de log sont remplis ou lors de l'exécution d'un log switch.

1.2.3. Le dictionnaire de données

Le dictionnaire de données est un ensemble de tables et de vues qui est utilisé en lecture seule comme référence pour une base de données particulière. Le dictionnaire de données stocke des informations sur :

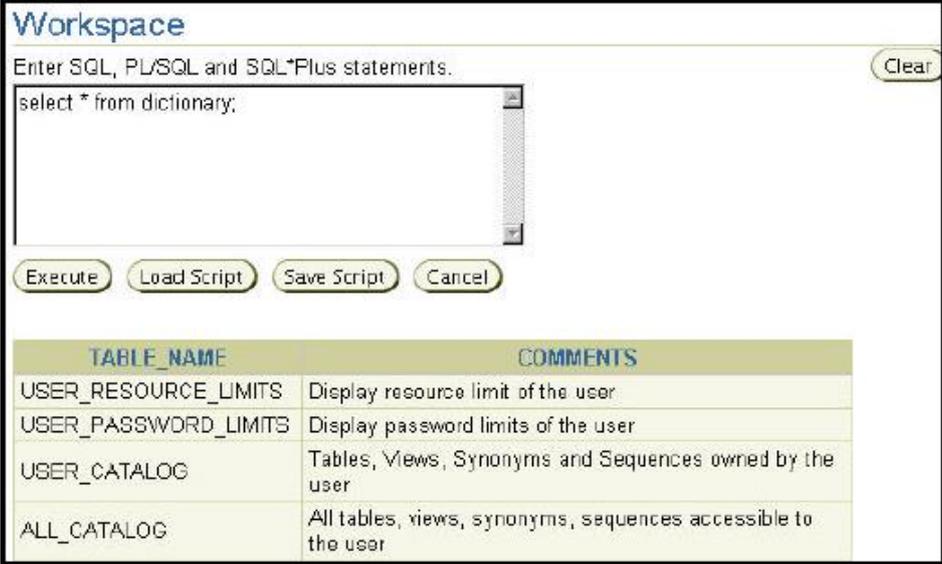
- La structure physique et logique de la base de données.
- Les utilisateurs valides de la base de données.
- Les informations sur les contraintes d'intégrité.
- L'espace alloué pour un objet d'un schéma et l'espace utilisé.

Le dictionnaire de données est créé lors de la création de la base de données et est automatiquement mis à jour lorsque la structure de la base de données est mise à jour.

Enterprise Manager trouve les informations à propos des objets de la base dans le dictionnaire de données. Vous pouvez également interroger les tables du dictionnaire de données pour obtenir des informations. Enterprise Manager le fait pour vous et présente les informations dans un format simple d'utilisation. La vue **DICTIONARY** contient les descriptions des tables et des vues du dictionnaire de données.

Ces tables et ces vues ont généralement un de ces préfixe :

- **USER** : Information sur les objets posséder par l'utilisateur courant.
- **ALL** : Information sur les objets accessible par l'utilisateur courant.
- **DBA** : Information sur tous les objets de la base de données.



The screenshot shows the Oracle Enterprise Manager Workspace interface. At the top, there is a text area for entering SQL, PL/SQL, and SQL*Plus statements. Below this text area are four buttons: Execute, Load Script, Save Script, and Cancel. The text area contains the query: `select * from dictionary;`. Below the buttons, a table displays the results of the query. The table has two columns: TABLE_NAME and COMMENTS. The results are as follows:

| TABLE_NAME | COMMENTS |
|----------------------|---|
| USER_RESOURCE_LIMITS | Display resource limit of the user |
| USER_PASSWORD_LIMITS | Display password limits of the user |
| USER_CATALOG | Tables, Views, Synonyms and Sequences owned by the user |
| ALL_CATALOG | All tables, views, synonyms, sequences accessible to the user |

1.2.4. Database/Grid Control

1.2.4.1. Database Control

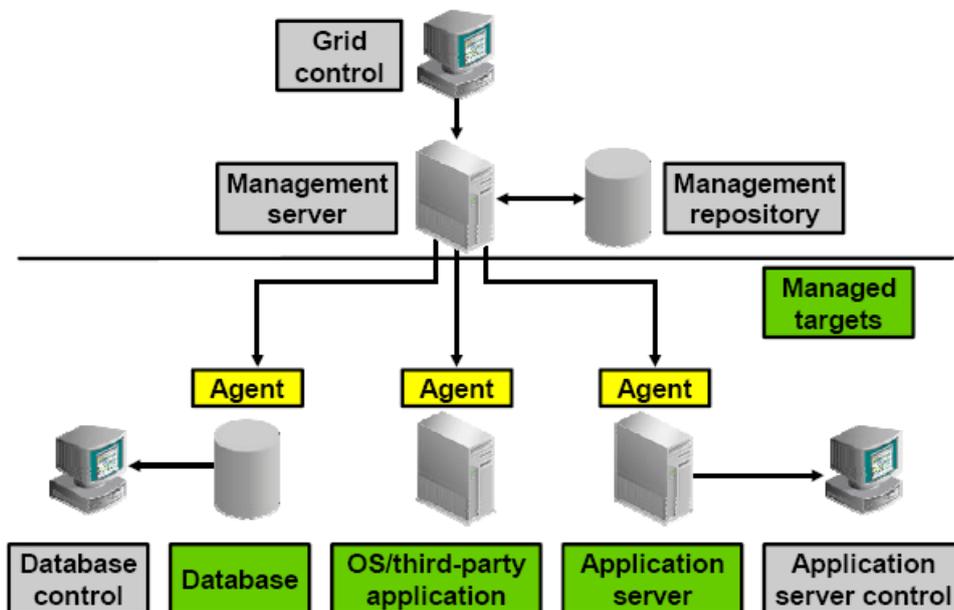
Oracle Database 10g est livré avec Oracle Enterprise Manager's Database Control. Database Control est une console de contrôle par le web que l'administrateur de la base de données peut utiliser pour :

- Surveiller les performances.
- Gérer les alertes.
- Contrôler la maintenance des agents et des conseillers.
- Administrer les objets des utilisateurs et de la base de données.
- Sauvegarder et récupérer la base de données.
- Gérer le stockage des données.
- Et bien plus.

Chaque base de données Oracle 10g que vous créez aura son propre Database Control.

1.2.4.2. Grid Control

Les capacités de Database Control peuvent être étendus et intégrés avec le reste de votre système en utilisant Oracle Enterprise Manager's Grid Control. L'architecture du Framework Grid Control permet un haut niveau de flexibilité et de fonctionnalité. Vous pouvez facilement personnaliser Enterprise Manager pour suivre la gestion et l'administration des besoins de votre environnement.



La configuration typique d'Enterprise Manager Framework contient les zones fonctionnelles suivantes :

- Les différentes cibles gérées
- La gestion des services
- Une console d'administration globale via le web
- Les outils Database control et Application server control

2. Module 02: Installer Oracle et créer une base de données

2.1. Installation d'Oracle 10g

2.1.1. Pré requis

Matériel:

- 512 Mo de mémoire physique (RAM)
- 1 Go d'espace de Swap (ou le double de la taille de la RAM)
- 400 Mo d'espace disque libre dans le dossier temporaire (`/tmp` ou `\temp`)
- 1,5 Go d'espace disque pour l'installation d'Oracle
- 1,5 Go d'espace disque pour la base de donnée préconfigurée

Système d'exploitation : Voir la documentation Oracle

2.1.2. Optimal Flexible Architecture (OFA)

OFA est une méthode permettant de configurer la base de données Oracle ainsi que d'autres bases de données. **OFA** tire profit des possibilités de l'OS et des disques pour faciliter la configuration. Ceci permet une flexibilité maximum pour la croissance et les performances des bases de données. Les méthodes décrites ici sont les fonctionnalités de base d'**OFA**.

OFA est utilisé pour:

- Organiser de grandes quantités de logiciel et de données sur le disque, pour éviter une baisse des performances.
- Faciliter les tâches administratives de routines comme la maintenance des logiciels et la sauvegarde des données, qui sont souvent vulnérable à la corruption de données.
- Faciliter le changement entre plusieurs bases de données Oracle.
- Contrôler et administrer la croissance des bases de données.
- Aider à éliminer la fragmentation de l'espace libre du dictionnaire de données, isoler les autres fragmentations et minimiser la corruption des ressources.

Le noyau d'**OFA** est un schéma nommé, qui donne une norme standard à appliquer aux points de montage (qui correspondent souvent aux disques physiques), aux répertoires, et aux fichiers eux-mêmes.

Syntaxe du point de montage : Nommer chaque point de montage en utilisant la syntaxe `/pm`, où **p** est une chaîne de caractères constante et **m** est une clé unique de taille fixe (généralement un nombre à deux chiffres) utilisée pour distinguer chaque point de montage. Exemple : point de montage `/u01` et `/u02`.

Syntaxe des répertoires de l'application : Nommer tous les répertoires de l'application en utilisant la syntaxe `/pm/h/u`, où **pm** est un point de montage nommé, **h** est un répertoire standard nommé et **u** est le nom du propriétaire du répertoire.

Exemples de répertoires de base d'OFA :

/u01/app/oracle
/u01/home/oracle

Syntaxe des répertoires des logiciels : Pour simplifier l'exécution simultanée de plusieurs versions d'OFA, stocker chaque version du logiciel d'Oracle dans des répertoires avec comme chemin **/pm/h/u/product/v**. Ici **product** représente un répertoire et **v** la version du logiciel.

Donc par exemple une installation de la version d'Oracle Database 10g version 10.1.0 serait stockée dans le répertoire :

/u01/app/oracle/product/10.1.0

Syntaxe des sous répertoires nommés : Pour faciliter l'organisation des données administratives, vous devez stocker les fichiers d'administration d'une base de données spécifiques dans des sous répertoires dont le chemin est **/h/admin/d/a/**, avec **h** le répertoire de base du propriétaire du logiciel oracle, **admin** le nom d'un répertoire, **d** le nom de la base de données, et **a** le sous répertoire dans lequel sont stockés les fichiers d'administration de la base. Voici une liste de sous répertoires pour les fichiers d'administration :

- **adhoc** : les scripts Ad hoc SQL pour une base de données particulière
- **arch** : les fichiers de redo log archivés
- **adump** : les fichiers d'audit (Indiquez dans le fichier de paramètre le chemin vers le dossier **adump** dans la variable AUDIT_FILE_DEST. Vider périodiquement ce sous répertoire.)
- **Bdump** : Les fichiers de trace des processus d'arrière plan
- **Cdump** : Core dump files
- **Create** : Programmes utilisés pour créer la base de données
- **Exp** : Les fichiers d'exportation de la base de données
- **Logbook** : Les fichiers enregistrant le statu et l'historique de la base de données
- **Pfile** : Les fichiers de paramètre de l'instance
- **udump** : Les fichiers utilisateurs de trace SQL

Syntaxe des fichiers nommés : La convention de nommage suivante permet d'identifier facilement les fichiers :

- **Control files** : **/pm/q/d/controln.ctl**
- **Redo log files** : **/pm/q/d/redon.log**
- **Data files** : **/pm/q/d/tn.dbf**

Les variables utilisées dans ces fichiers sont :

- **Pm** : Un point de montage nommé comme décrit auparavant
- **q** : Chaîne de caractère permettant de distinguer les données Oracle des autres fichiers (généralement appelé ORACLE ou oradata)
- **d** : La valeur du paramètre d'initialisation **DB_NAME** (le nom de la base de données)
- **t** : Le nom d'un tablespace Oracle
- **n** : Un nombre sur deux chiffres

2.1.3. Paramétrage des variables d'environnement

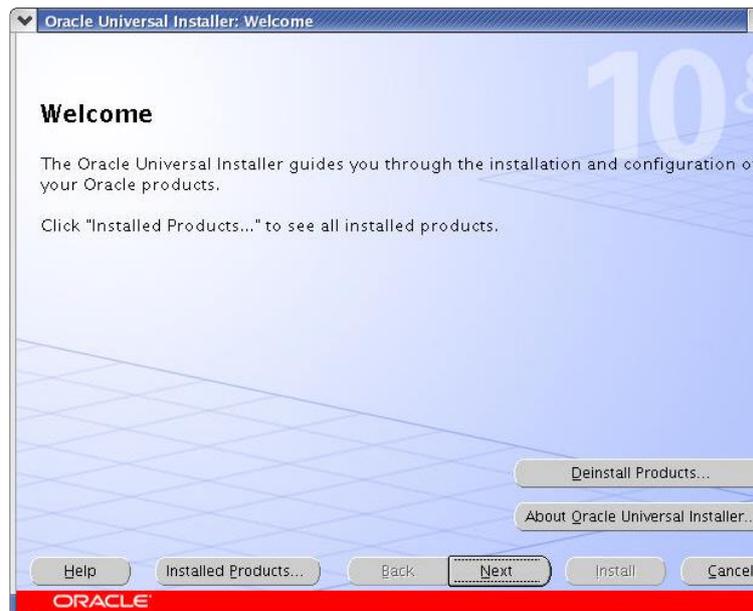
Il y a beaucoup de variables d'environnement Oracle, celles indiquées ici sont très importantes pour installer et utiliser une base de données Oracle correctement. Aucune d'elles ne requièrent une initialisation obligatoire mais les configurer avant une installation permet d'éviter des problèmes futurs.

- **ORACLE_BASE** : Indique le chemin du répertoire Oracle pour la structure **OFA**. Son utilisation est optionnelle, mais elle peut faciliter les installations et mises à jour futures. Il s'agit du chemin du répertoire, par exemple :
`/u01/app/oracle`
- **ORACLE_HOME** : Indique le répertoire contenant les logiciels Oracle. Il s'agit du chemin du répertoire, par exemple :
`§ORACLE_BASE/product/10.1.0`
- **ORACLE_SID** : Le nom initial de l'instance (par défaut ORCL). Il s'agit d'une chaîne de caractères et de nombres qui doit commencer par une lettre. Oracle recommande un maximum de 8 caractères.
- **NLS_LANG** : Indique le paramètre NLS initial pour une session. Il est de la forme **language_territory.character**.
Par exemple : `AMERICAN_DENMARK.WE8MSWIN1252`

Avec cet exemple, la session est configurée pour utiliser la langue **AMERICAN** pour les messages Oracle, les noms des jours et des mois. Le pays est **DENMARK**, ceci paramètre le format de la date, de l'heure, numérique et les conventions monétaires. Le jeu de caractères de **WE8MSWIN1252** oblige Oracle Net à convertir les informations de caractères dans ce jeu de caractères. Il s'agit d'une variable d'environnement UNIX et un registre sous Windows. Vous pouvez connaître le jeu de caractère défini sur la session courante grâce à la requête :

```
select * from nls_session_parameters;
```

Remarque : Sous Windows la valeur par défaut du paramètre **NLS_LANG** est la valeur de la clé de registre correspondant à la langue du clavier.



Oracle Universal Installer fournit les possibilités suivantes pour la gestion et la distribution des logiciels :

- Résolution automatique des dépendances et manipulation de logique complexe
- Installation à partir du Web
- Installation de composants et de suites
- Désinstallation facilitée
- Support pour plusieurs Oracle homes
- Support pour la globalisation (NLS)
- Support pour les installations distribuées
- Installation « silencieuse » en utilisant un fichier de réponse

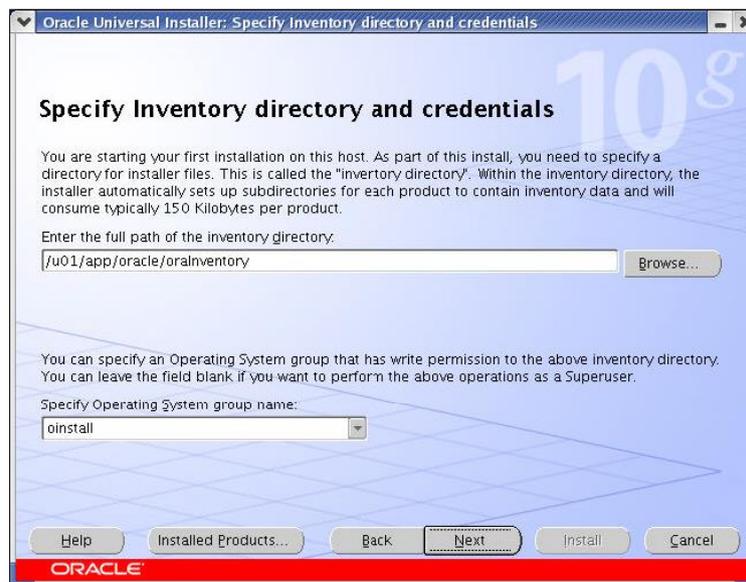
2.2.2. Configuration d'Oracle

2.2.2.1. Inventory et nom de groupe UNIX

Le répertoire appelé inventory est une zone utilisée pendant l'installation du logiciel. Les fichiers contenus dans ce répertoire ne doivent pas être supprimés. Le répertoire inventory contient les fichiers pour la gestion interne d'OUI. Ce répertoire stocke des informations à propos :

- Tous les produits Oracle installés sur la machine
- Les autres produits non Oracle inclut dans le logiciel Oracle, comme la Java Runtime Environment (JRE)

Vous devez réserver une place de 4Mo pour pouvoir stocker ces fichiers.



Si vous installez un produit sous un système UNIX, l'installateur vous demandera de spécifier le nom du groupe propriétaire des répertoires de base. Vous devrez spécifier le nom d'un groupe UNIX qui aura les permissions de modifier, installer et désinstaller les produits Oracle. Les membres de ce groupe devront avoir les droits d'écriture dans le répertoire de base qui aura été choisi. Seulement les utilisateurs appartenant à ce groupe pourront installer ou désinstaller les logiciels sur la machine.

2.2.2.2. orainstRoot.sh

Lors d'une installation sous une plateforme UNIX, vous devez lancer le script **orainstRoot.sh** en tant que root. Ce script crée le fichier pointant vers le répertoire inventory (sous Linux ce fichier est **/etc/oraInst.loc**). Le fichier pointant vers le répertoire inventory est utilisé par Oracle Universal Installer au démarrage pour trouver le chemin du répertoire inventory.

Un exemple de ce fichier :

```
inventory_loc = /oracle/oraInventory
inst_group = oinstall
```



```
# sh /u01/app/oracle/oraInventory/orainstRoot.sh
Creating the Oracle inventory pointer file (/etc/oraInst.loc)
Changing groupname of /u01/app/oracle/oraInventory to oinstall.
```

Lors d'une installation sous Windows le chemin du répertoire inventory est stocké dans le registre.

Le script `oraintRoot.sh` ressemble à ceci :

```
#!/bin/sh
INVPTR=/etc/oraInst.loc
INVLOC=/oracle/oraInventory
GRP=oinstall
PTRDIR="`dirname $INVPTR`";

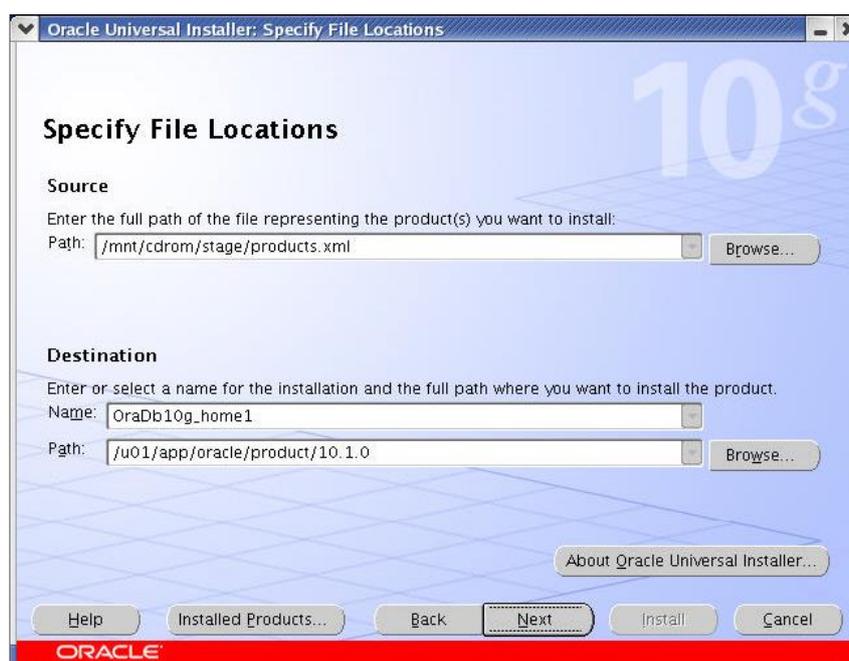
# Create the software inventory location pointer file
if [ ! -d "$PTRDIR" ]; then
    mkdir -p $PTRDIR;
fi
echo "Creating the Oracle inventory pointer file
($INVPTR)";
echo inventory_loc=$INVLOC > $INVPTR
echo inst_group=$GRP >> $INVPTR
chmod 644 $INVPTR

# Create the inventory directory if it doesn't exist
if [ ! -d "$INVLOC" ];then
    echo "Creating the Oracle inventory directory ($INVLOC)";
    mkdir -p $INVLOC;
    chmod 775 $INVLOC;
fi
echo "Changing groupname of $INVLOC to oinstall.";
chgrp oinstall $INVLOC;
if [ $? != 0 ]; then
    echo "WARNING: chgrp of $INVLOC to oinstall failed!";
fi
```

2.2.2.3. Emplacement des fichiers

Oracle Universal Installer fournit une valeur par défaut sur la page Specify File Locations qui pointe vers le chemin du kit d'installation du produit.

Les Oracle Home sont identifiés par un nom. Ils identifient le groupe de programme associé avec un Oracle Home particulier et les services installés avec l'Oracle Home associé. Le nom du Oracle Home doit être composé de 1 à 16 caractères et peut seulement inclure des caractères alphanumériques et des tirets-bas, et ne doit pas contenir d'espaces.



Oracle Universal Installer maintient les Oracle Home suivants sur une plateforme Windows :

- Tous les Oracle Home créés en utilisant Oracle Universal Installer
- Tous les Home créés avec la version précédente d'Oracle Universal Installer (ORCA-based)
- Les Home sauvegardés dans le registre d'entrée **ORACLE_HOME**

Oracle Universal Installer maintient les Oracle Home suivants sur une plateforme UNIX :

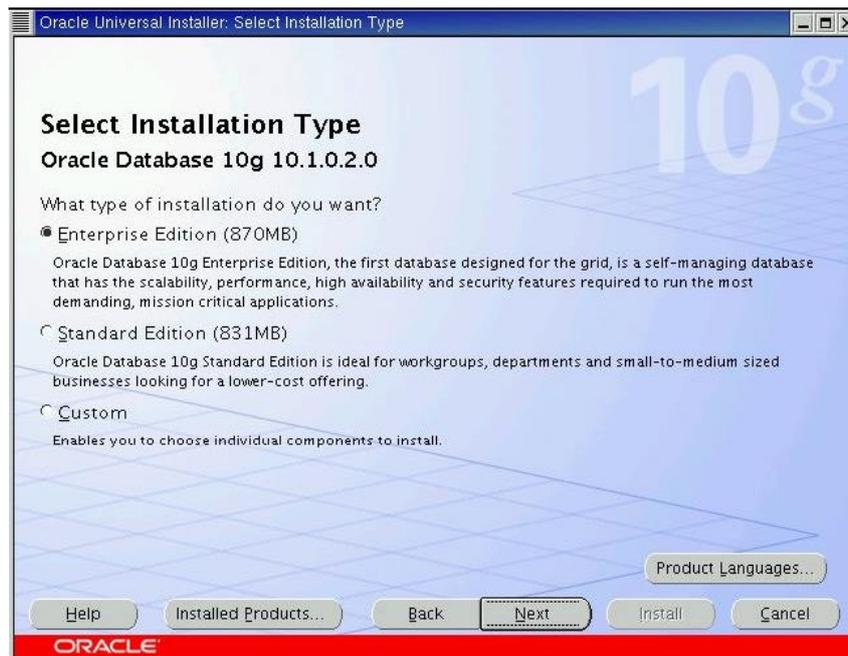
- Tous les Oracle Home créés en utilisant Oracle Universal Installer
- Tous les Home définis dans le fichier **/var/opt/oratab** (In linux, **/etc/oratab**)
- Le Home sauvegardé dans la variable d'environnement **ORACLE_HOME**

Si aucun de ces Home n'existe, un Home par défaut est calculé en se basant sur le volume du disque avec le plus d'espace libre possible.

2.2.2.4. Type d'installation

Oracle Database Enterprise Edition offre aux leaders du marché une fiabilité et une évolutivité sur les configurations des systèmes simple ou en cluster. Il fournit les dispositifs les plus complets pour les transactions en ligne.

Oracle Standard Edition est pour les petites et moyennes entreprises, ou les applications départementales qui requièrent toute la puissance, la fiabilité et la sécurité d'Oracle sans les options de la version Enterprise.



Les options suivantes ne sont pas disponibles avec la version Standard d'Oracle :

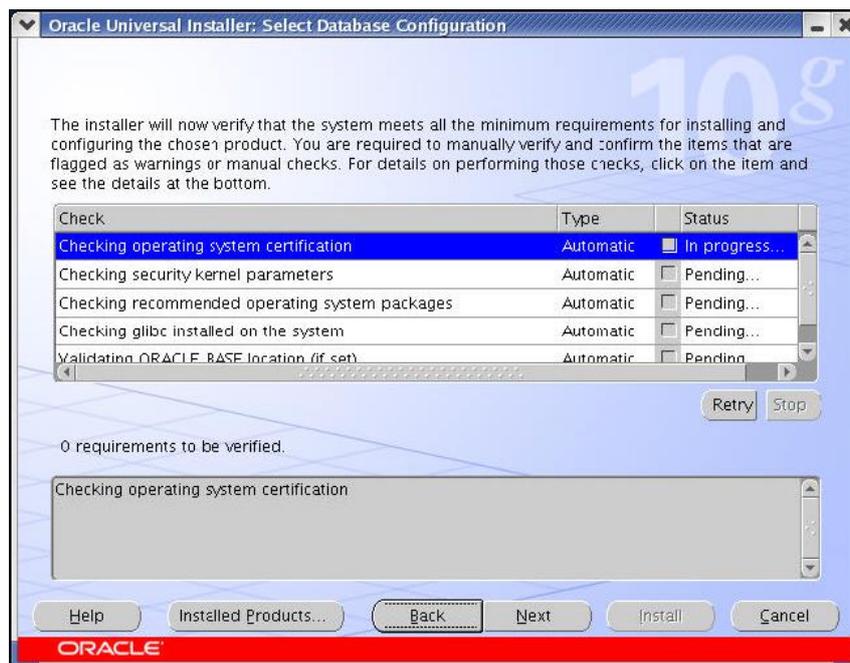
- **Oracle Data Guard:** Ensemble de services qui permettent de créer, gérer et surveiller une ou plusieurs bases de données.
- **Oracle Advanced Security, Oracle Label Security, Enterprise User Security, Virtual Private Database, N-tier authentication, et Fine Grained Auditing:** Ces options améliorent la sécurité d'une base de données Oracle 10g.
- **Oracle Partitioning, Oracle OLAP, Export Transportable Tablespace,** et autres utilitaires qui supportent l'Enterprise **Data Warehouses**.

L'installation **Custom** permet de sélectionner les composants que l'on veut installer.

2.2.2.5. Pré requis nécessaires

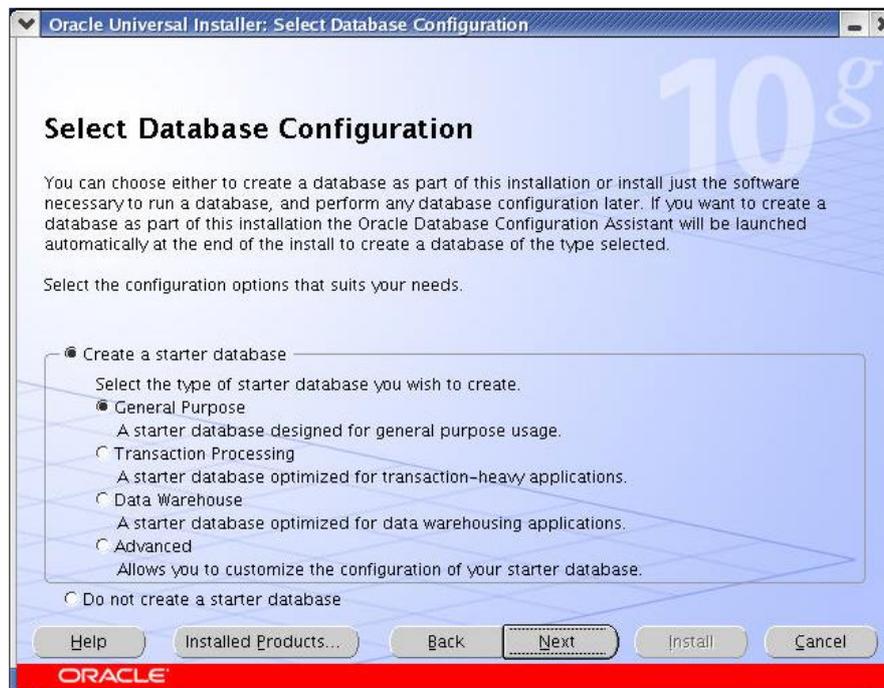
Avant l'installation, Oracle Universal Installer vérifie l'environnement pour s'assurer que les conditions nécessaires pour réussir l'installation sont réunies. Cette page montre le nom, le type et le statu pour tous les contrôles nécessaires. Les contrôles automatiques sont effectués en premier, si il y en a des optionnels vous pourrez les lancer une fois que les automatiques finis.

Une fois tous ces contrôles effectués, un résumé sera affiché dans le bas de la page. Un contrôle peut exiger une vérification si un avertissement est retourné pendant l'exécution ou lorsqu'un contrôle n'est pas confirmé. Pour vérifier un avertissement ou un contrôle manuel, cochez la case correspondant à l'avertissement ou au contrôle manuel.



2.2.2.6. Base de données de départ

Il existe trois template de base de données, que vous pouvez choisir, ou alors créer une base de données de départ personnalisée. Le template de création de base de données utilise des fichiers préconfigurés et crée une base de données rapidement. L'option advanced permet d'avoir une flexibilité totale dans la création de votre base de données de départ mais rallonge le temps de création.



Remarque : Si vous choisissez de ne pas créer une base de données de départ, la page de résumé suivante apparaît.

Si vous choisissez de créer une base de données pendant l'installation, vous aurez plusieurs pages vous demandant les informations de configuration de la base.

- **Database Naming:** Le nom de la base de données, par défaut **orcl**.
- **Database Character Set:** Le jeu de caractères utilisés pour stocker vos données. Vous devez prendre du temps pour définir quel jeu de caractères sera utilisé pour stocker vos données. Même s'il est possible de changer le jeu de caractère plus tard, cela prendra beaucoup de temps et exigera des temps d'arrêt de votre base de données. Si vous ne savez pas quel jeu de caractères utiliser et que vous savez que vous utiliserez plusieurs langues, le Standard Unicode **UTF-8 AL32UTF8** est un bon choix. Le jeu de caractères **WE8ISO8859P1** ne possède pas le symbole euro (€), mais le **WE8ISO8859P15** le possède. Pour plus de détails sur la sélection du jeu de caractères, voir *Oracle Database Globalization Support Guide*.
- **Sample Schemas:** Les schémas d'exemple sont utilisés pour les démonstrations et les entraînements.
- **Database Management:** Le Grid Control est utilisé pour gérer plusieurs bases de données situées sur une ou plusieurs machines. Database Control est utilisé pour gérer une seule base de données.

Specify Database Configuration Options

Database Naming

A Global Database Name, typically of the form "name.domain", uniquely identifies an Oracle database. In addition, each database is referenced by at least one Oracle System Identifier (SID). Specify the Global Database Name and SID for this database.

Global Database Name: SID:

Database Character Set

The database character set is determined based on the number of language groups that will be stored in your database. See "Help" for the definition of language groups. Select the character set that should be used in your database.

Select Database Character set:

Database Examples

You can choose to create a starter database with or without sample schemas. Note that you can plug in the sample schemas to your existing starter database after creation. See "Help" for more details.

Create database with sample schemas

Use Grid Control for Database Management

Management Service:

Use Database Control for Database Management

Enable Email Notifications

Outgoing Mail (SMTP) Server:

Email Address:

Options du stockage des fichiers :

- **File System:** Spécifie où stocker les fichiers sur votre OS.
- **Automatic Storage Management (ASM):** Dans ce mode les fichiers sont créés et gérés automatiquement, et vous avez les bénéfices supplémentaires comme le mirroring ou le striping.
- **Raw Devices (Partitions):** Il s'agit de la partition des disques de stockage. Généralement vous devez les utiliser seulement si vous êtes un utilisateur confirmé. Voir la documentation de votre OS pour plus de détails sur la configuration et la maintenance des Raw Partitions.

Specify Database File Storage Option

Select the storage mechanism you would like to use for database creation.

File System

Use the file system for database storage. For best database organization and performance, Oracle recommends installing database files and Oracle software on separate disks.

Specify Database file location:

Automatic Storage Management (ASM)

Automatic Storage Management simplifies database storage administration and optimizes database layout for I/O performance.

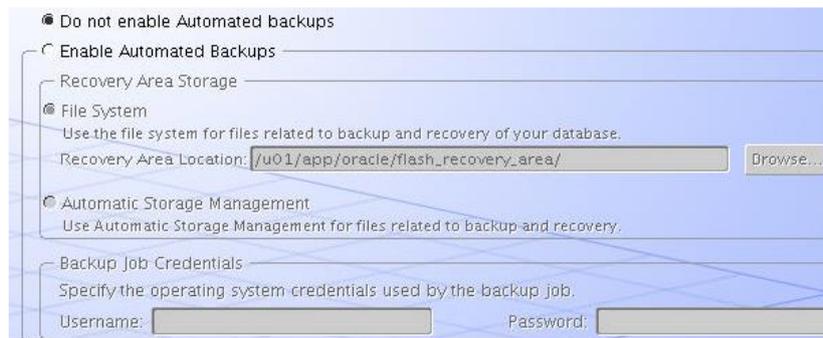
Raw Devices

Raw partitions can also provide the required shared storage for Real Application Clusters (RAC) databases. You will need to create one raw device for each data file, control file, and log file for the starter database and then provide a file that maps specific tablespaces, control files, and log files to rawvolumes.

Specify Raw Devices mapping file:

Options de Sauvegarde et de Récupération (Backup et Recovery) :

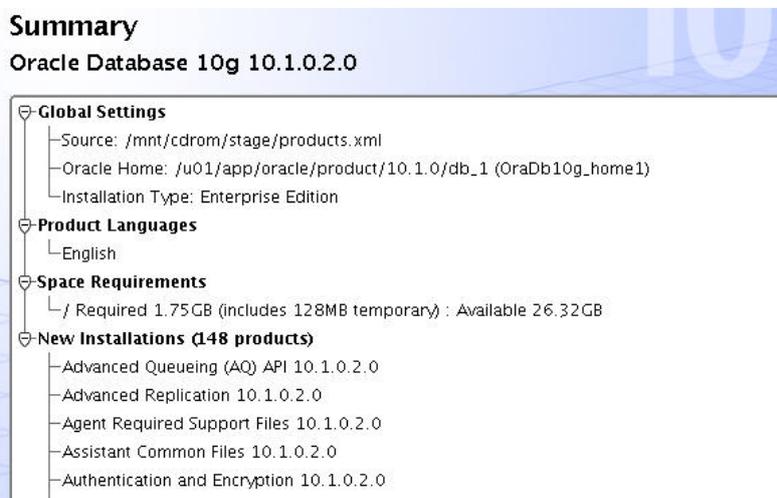
- **Do not enable Automated backups:** Vous aurez besoin de planifier manuellement vos sauvegardes. Vous pourrez toujours modifier ce choix plus tard.
- **Enable Automated Backups:** Lorsque vous choisissez d'automatiser vos sauvegardes, vous devrez spécifier un chemin où seront stockées vos sauvegardes, et les informations concernant sa planification.

**2.2.2.7. Mot de passe et résumé**

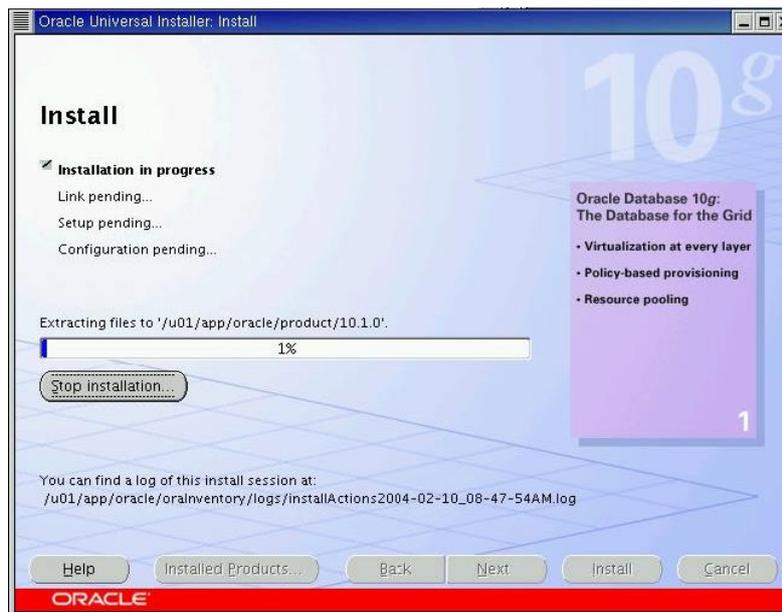
Database Schema Passwords: Permet de spécifier les mots de passe pour les utilisateurs administratifs **SYS**, **SYSTEM**, **SYSMAN**, et **DBSNMP**. Vous pouvez spécifier un mot de passe différent pour chacun d'entre eux.



Summary: Résume tous les produits et paramètres que vous avez choisi pour l'installation. L'étape suivante est de lancer l'installation. Acceptez le résumé et cliquez ensuite sur Install pour lancer l'installation.

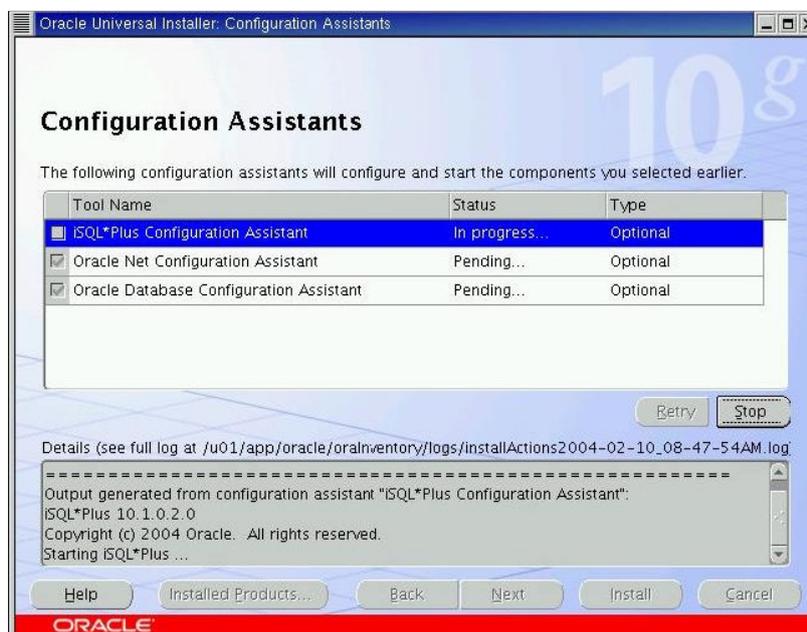


2.2.3. Fin de l'installation

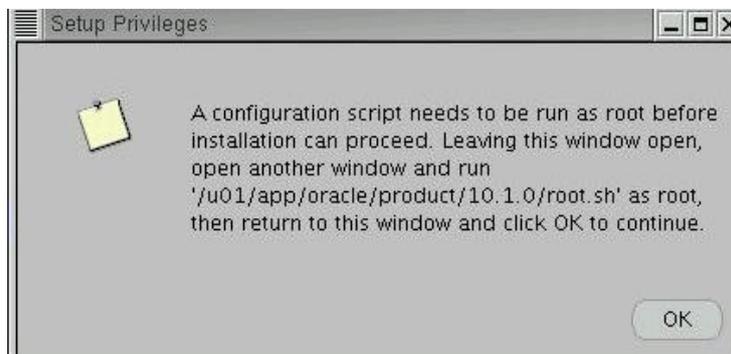


Si vous avez choisi de créer votre base de données lors de l'installation, la page de l'assistant de configuration vous sera proposée à la fin de l'installation du logiciel.

- **iSQLPlus Configuration Assistant:** Permet de configurer l'instance d'OC4J (Oracle Containers for Java) qui est utilisée par iSQLPlus et autres composants pour se connecter à la base de données Oracle.
- **Oracle Net Configuration Assistant:** Permet de configurer les composants réseaux de base durant l'installation, incluant (ces composants sont détaillés plus tard) :
 - Les noms des Listener et les adresses des protocoles.
 - La chaîne de connexion que l'utilisateur devra utiliser pour se connecter.
 - Les noms du service Net dans le fichier `tnsnames.ora`.
 - Le répertoire d'utilisation du serveur.
- **Oracle Database Configuration Assistant:** Permet de créer la base de données de départ que vous avez choisie. Il utilise les choix que vous avez faits pour créer la base de données. Lorsque l'assistant de configuration aura fini, vous pourrez bloquer des comptes et changer les mots de passe.



Lors d'une installation sous UNIX (ou Linux), ou si vous avez choisi de ne pas créer une base de données de départ, vous serez invité à lancer un ou plusieurs scripts en tant que root. Le script s'appelle `root.sh`. Il met à jour ou crée un fichier contenant des informations sur l'emplacement du répertoire `ORACLE_HOME` et sur les bases de données. Sous Linux, le fichier se nomme `/etc/oratab`.



Lors d'une installation sous Microsoft Windows, il n'y a pas de script `root.sh` car les informations à propos de la variable `ORACLE_HOME` et des bases de données sont stockées dans le registre. Le fichier `oratab` est un fichier où l'utilisateur inscrit des références pour toutes les bases de données afin qu'elles soient visibles par un Agent ou contrôlées par Oracle Enterprise Manager.

2.3. Création de la base de données Oracle

2.3.1. DBCA

Database Configuration Assistant (DBCA) vous permet de créer, de changer la configuration, ou de supprimer une base de données. Vous pouvez également créer une base de données à partir d'une liste de templates pré définis ou utiliser une base de données existante comme exemple pour créer une nouvelle base de données ou un template. Un template est une base de données pré configurée que vous utilisez pour avoir un point de départ pour une nouvelle base de données.



Create a database: Si vous sélectionnez cette option, vous pourrez créer une nouvelle base de données ou un nouveau template.

Configure options in a database: Si vous sélectionnez l'option « Change database configuration », vous pourrez configurer les options installées qui n'avaient pas été configurées pour qu'elles soient utilisées avec votre base de données. Vous pouvez également activer ou désactiver l'utilisation d'un serveur partagé.

Remarque : L'option « Change database configuration » n'est pas disponible pour Oracle Real Application Clusters.

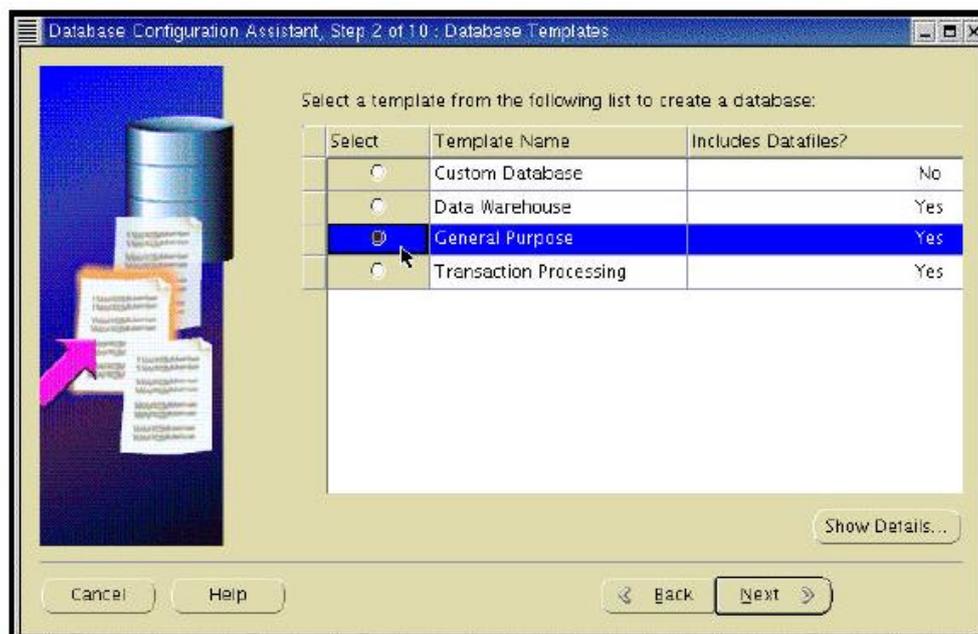
Delete a database: Si vous sélectionnez cette option, vous pourrez supprimer tous les fichiers de la base de données.

Manage Templates: Si vous sélectionnez cette option, vous aurez trois choix pour créer un template :

- Depuis un template existant.
- Depuis une base de données existante (seulement la structure).
- Depuis une base de données existante (Structure et données).

2.3.2. Créer une base de données

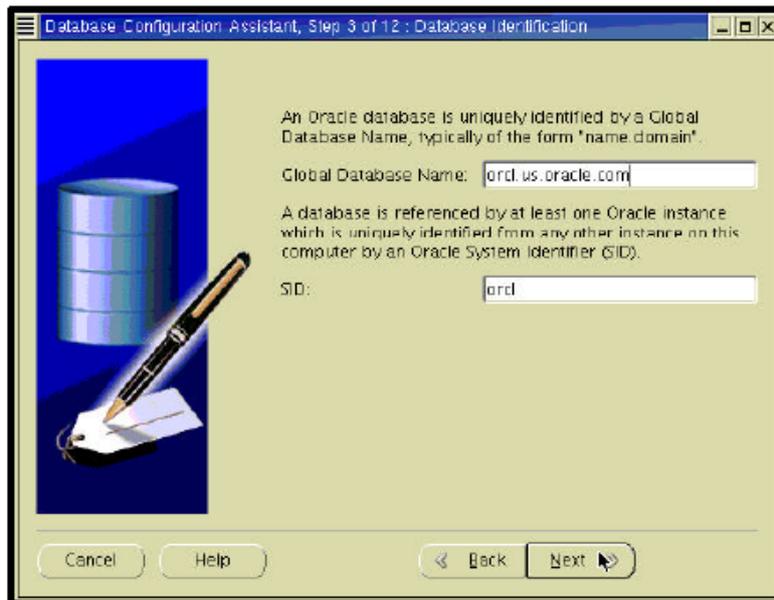
Lors de la création d'une base de données avec DBCA, vous pouvez sélectionner une des trois bases de données pré configurées, ou créer une base de données personnalisée. Lors de la création d'une base de données avec DBCA, vous pouvez sélectionner une des trois bases de données pré configurées, ou créer une base de données personnalisée. Oracle Corporation met à disposition des templates pré définis. Il existe des templates pour le data warehouse, « general purpose » pour une utilisation classique, et les bases de données « transactions processing » (OLTP). Le template contient des paramètres optimisés pour la charge de travail. Cliquez sur « Show Details » pour voir la configuration de chaque type de base de données. Choisissez un template en fonction de la charge de travail attendu du type d'activité de votre base de données. Si vous n'êtes pas sur, sélectionnez le template par défaut General Purpose.



Pour les environnements plus complexes, vous serez amenés à choisir l'option Custom Database. Dans ce cas, la création de votre base de données sera plus longue car un script devra être lancé.

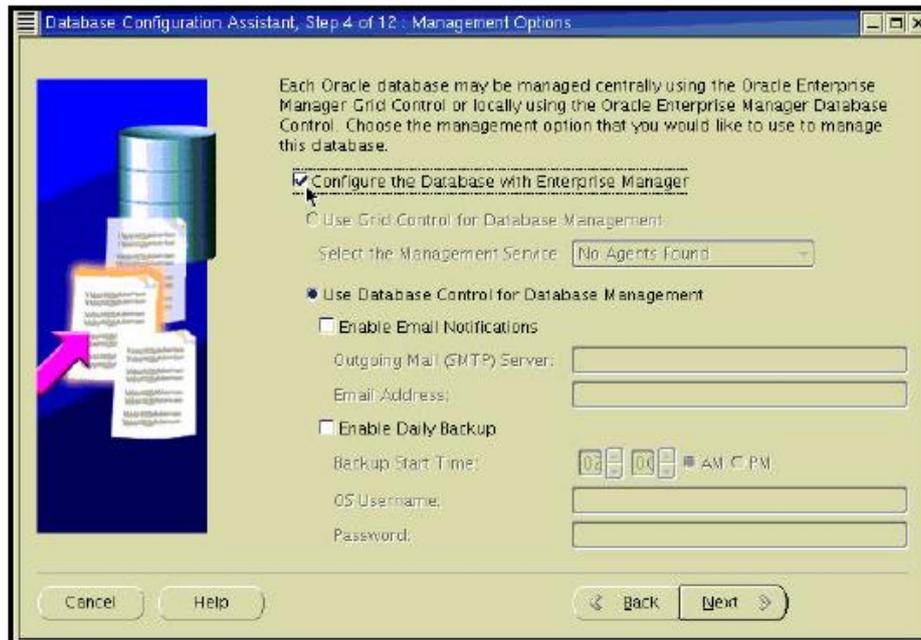
2.3.2.1. Identification de la base de données

Entrez le Global Database Name, de la forme `database_name.domain_name` et le SID (Oracle system identifier). Le SID par défaut correspond au nom de la base de données et identifie uniquement l'instance tournant avec la base de données. Il est important de comprendre que le SID est le nom de l'instance qui se connectera à la base de données et qu'il ne s'agit pas obligatoirement du nom de la base de données. Une instance et une base de données se connecte s'en pour autant avoir le même nom, même si c'est le mieux. Avec Real Application Clusters, plusieurs instances ouvrent la même base de données et le SID de chaque instance doit être différent.



2.3.2.2. Options de gestion

A travers cette page, vous pourrez paramétrer votre base de données afin qu'elle puisse être gérée par Oracle Enterprise Manager, fournissant des outils de gestion via le web pour des bases de données individuelles, comme l'utilitaire central management qui vous permet de gérer tout votre environnement Oracle. Pour utiliser Enterprise Manager, sélectionnez « Configure the Database with Enterprise Manager ».



Si Oracle Management Agent a été installé sur votre machine, vous pouvez activer l'option d'un management central en sélectionnant « Use Grid Control for Database Management ». Si vous sélectionnez ce type de gestion, vous devrez également préciser le service de gestion qui sera utilisé à l'aide du menu déroulant. Sinon, sélectionnez « Use Database Control for Database Management » pour gérer votre base de données individuellement. Si vous choisissez cette option, vous pourrez également ajouter les options telles que la notification par email (Email Notifications) ou la planification de sauvegarde journalière (Enable Daily Backup). Cliquez sur « Help » pour plus d'informations sur ces options.

2.3.2.3. Gestion des mots de passe

Database schema passwords: Fournit les mots de passe pour les utilisateurs administratifs, **SYS**, **SYSTEM**, **SYSMAN** et **DBSNMP**. Vous pouvez fournir un mot de passe pour chaque utilisateur séparément ou fournir un mot de passe pour tous.

The image shows two overlapping windows from the Oracle Database Configuration Assistant (DBCA). The background window is the 'Database Schema Passwords' screen, which has two radio buttons: 'Use the Same Password for All Accounts' (selected) and 'Use Different Passwords'. The 'Use the Same Password' section has 'Password:' and 'Confirm Password:' fields, both containing masked characters. The 'Use Different Passwords' section has a table with columns 'User Name', 'Password', and 'Confirm Password'. The table lists users: SYS, SYSTEM, DBSNMP, and SYSMAN. The foreground window is the 'Select the storage mechanism you would like to use for the database' screen. It has three radio buttons: 'File System' (selected), 'Automatic Storage Management (ASM)', and 'Raw Devices'. The 'File System' option is selected, and the text below it says 'Use the File System for Database storage.' The 'Automatic Storage Management (ASM)' option has a description: 'Automatic Storage Management simplifies database storage administration and optimizes database layout for I/O performance. To use this option you must either specify a set of disks to create an ASM disk group or specify an existing ASM disk group.' The 'Raw Devices' option has a description: 'Raw partitions or volumes can provide the required shared storage for Real Application Clusters (RAC) databases if you do not use Automatic Storage Management and a Cluster File System is not available. You need to have created one raw device for each datafile, control file, and log file you are planning to create in the database.' Below the 'Raw Devices' option is a checkbox 'Specify Raw Devices Mapping File' followed by a text box and a 'Browse...' button.

| User Name | Password | Confirm Password |
|-----------|----------|------------------|
| SYS | | |
| SYSTEM | | |
| DBSNMP | | |
| SYSMAN | | |

Options du stockage des fichiers :

- **File System:** Stocke les fichiers dans le système de fichier de l'OS.
- **Automatic Storage Management (ASM):** Dans ce mode les fichiers sont créés et gérés automatiquement, et vous avez les bénéfices supplémentaires comme le mirroring ou le stripping.
- **Raw Devices (partitions):** Il s'agit de la partition des disques de stockage. Généralement vous devez les utiliser seulement si vous êtes un utilisateur confirmé. Voir la documentation de votre OS pour plus de détails sur la configuration et la maintenance des Raw Partitions.

2.3.2.4. Emplacement des fichiers et Backup / Recovery

Specify locations for the Database files to be created:

Use Database File Locations from Template

Use Common Location for All Database Files

Database Files Location: /u01/oradata/orcl

Use Oracle-Managed Files

Database Area:

i If you want to specify different locations for any database files, pick either of the above options and use the Storage page to specify each location.

Choose the recovery options for the database:

Specify Flash Recovery Area

This is used as the default for all backup and recovery operations, and is also required for automatic backup using Enterprise Manager. Oracle recommends that the database files and recovery files be located on physically different disks for data protection and performance.

Flash Recovery Area: {ORACLE_BASE}/flash_reco

Flash Recovery Area Size: 2048 M Bytes

Enable Archiving

File Locations:

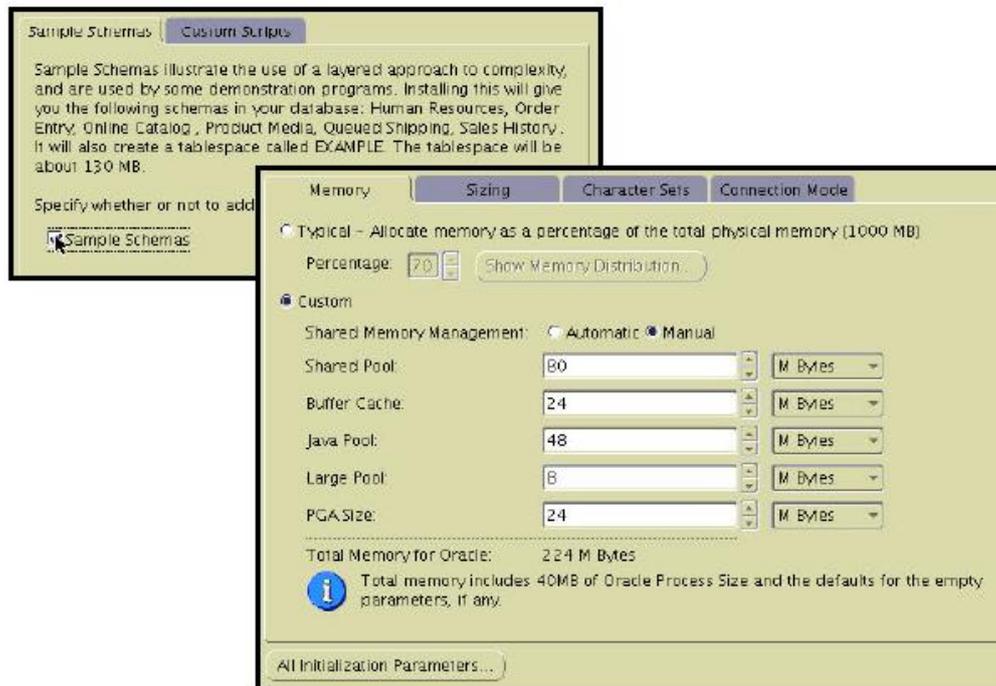
- **Use Database File Locations from Template:** Sélectionnez cette option signale à DBCA d'utiliser le répertoire d'information spécifié dans le template. Vous aurez l'opportunité de faire des modifications sur le nom des fichiers de la base de données et leur emplacement.
- **Use Common Location for All Database Files:** Cette option vous oblige à redéfinir une nouvelle zone commune pour tous les fichiers de la base de données. Vous aurez l'opportunité de faire des modifications sur le nom des fichiers de la base de données et leur emplacement plus tard dans la page Storage.
- **Use Oracle-Managed Files:** Sélectionnez cette option pour éliminer l'intervention du DBA dans la gestion des fichiers du système d'exploitation qu'une base de données Oracle comporte. Vous devez fournir le chemin de la zone de la base de données.

Options de sauvegarde et récupération :

- Une **flash recovery area** est un emplacement dans lequel Oracle peut stocker et gérer les fichiers relatifs à la sauvegarde et la restauration.

Pour activer l'archivage, mettez la base de données en mode archive log lors de la création. L'archivage sera vu plus en détail dans la leçon « Backup and Recovery ».

2.3.2.5. Paramètres d'initialisation



Sample schemas: Les schémas d'exemple sont un ensemble de schémas utilisés pour les démonstrations et les entraînements.

Custom scripts: Ici vous pouvez spécifier les scripts que vous voulez lancer pendant la création de la base de données.

Initialization parameters: Les quatre onglets peuvent être utilisés pour définir les paramètres les plus communs, et en cliquant sur « All Initialization Parameters » vous pouvez voir et configurer tous les paramètres.

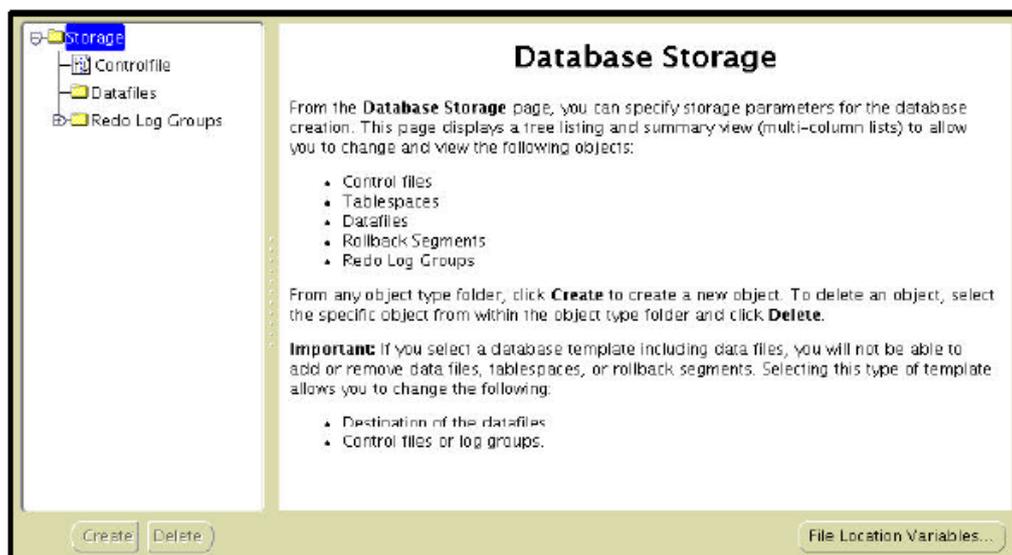
- **Memory:** Alloue la mémoire utilisé pour la SGA et pour chaque PGA d'un processus utilisateur.
- **Sizing:** Vous permet de définir la taille d'un bloc, mais si vous utilisez un template, la taille du bloc ne peut pas être changée. Vous pouvez également définir le nombre maximum de processus OS alloué à l'instance pour se connecter.
- **Character sets:** Configure le jeu de caractères par défaut pour la base de données et le jeu de caractères national. Le jeu de caractères par défaut est utilisé par beaucoup de type de données dans la base de données. Les types de données **NCHAR**, **NVARCHAR2** et **NCLOB** supportent uniquement le format Unicode, lequel est le jeu de caractères national. Vous pouvez utiliser le jeu de caractères **AL32UTF8** ou **AL16UTF16**.
- **Connection mode:** Une base de données Oracle crée des processus serveur pour traiter les requêtes des utilisateurs connectés à une instance. Un processus serveur peut être :
 - dédié, c'est-à-dire un processus serveur est attribué à un processus utilisateur.
 - partagé, c'est-à-dire un processus serveur peut être attribué à plusieurs processus utilisateur.

Les connections via des processus serveur en mode dédié sont toujours disponible au niveau d'une base de donnée, par contre vous devez configurer et activer le mode partagé en renseignant un ou plusieurs paramètres d'initialisation.

2.3.2.6. Stockage de la base de données

Sur cet écran, vous pouvez voir les paramètres de stockage des fichiers de contrôles, des fichiers de données, et des fichiers de redo. Lorsque vous utilisez un template, vous ne pouvez pas ajouter de fichiers de données à votre base de données, mais vous pouvez ajouter des fichiers de contrôles et de redo.

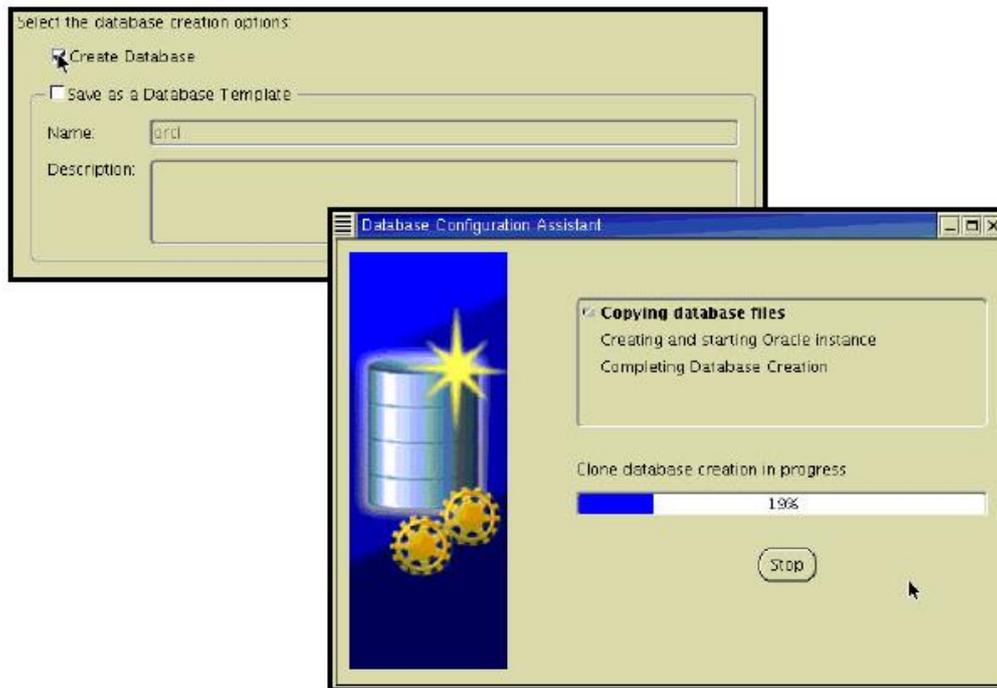
La maintenance des fichiers de contrôles multiplexés se fait de la sorte que chaque copie est une copie exacte des autres. DBCA multiplxe automatiquement les control files en trois fichiers. Vous pouvez en ajouter plus si vous le souhaitez. Les fichiers de redo log sont organisés en groupe et doivent également être multiplexés. DBCA ne multiplxe pas automatiquement les fichiers redo log. Chaque fichier dans un groupe est une copie exacte des autres membres du groupe. Vous pouvez ajouter des membres supplémentaires par groupe sur cet écran ou les ajouter plus tard après la création de la base de données.



2.3.2.7. Les options de création

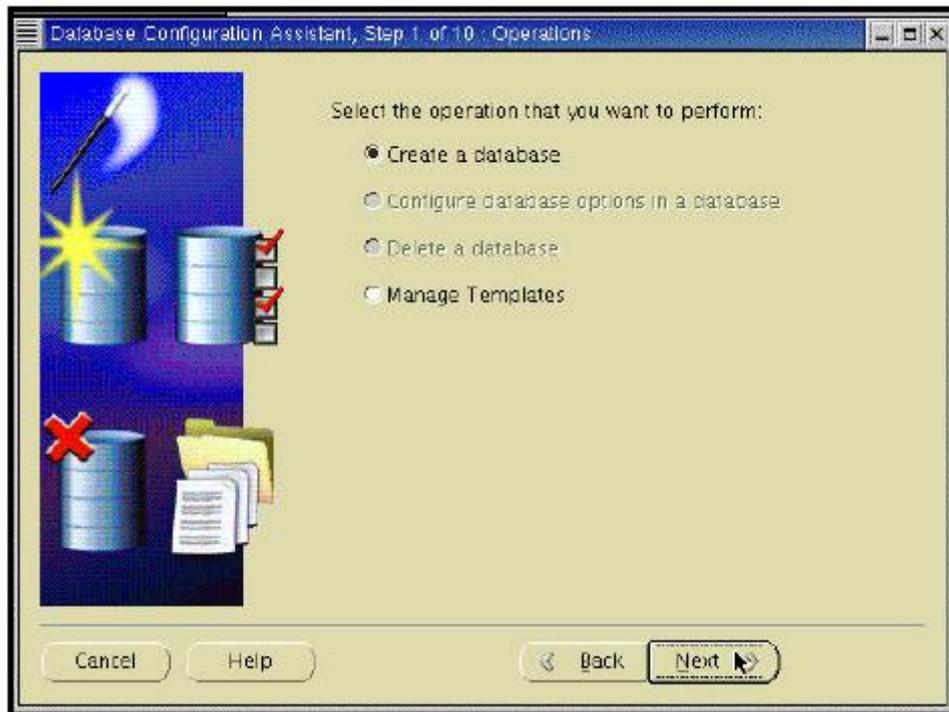
Vous avez l'option pour sauvegarder votre base de données en tant que template. Ce template peut être utilisé plus tard pour créer une base de données avec les mêmes options.

Après avoir cliqué sur Next sur la page de création, une page de confirmation apparaît vous résumant l'ensemble des options choisies. Vous pouvez encore effectuer des modifications, à ce moment, avant que le processus de création ne démarre. Vous pouvez sauvegarder la page de confirmation en tant que fichier HTML pour le revoir plus tard. Après avoir cliqué sur Ok sur la page de confirmation, la création de la base de données commence.



A la fin de l'installation, vous aurez la possibilité de débloquer les comptes créés et changer leur mot de passe si vous le désirez. Cliquez sur Password Management pour débloquer des comptes et changer les mots de passe. Cliquez sur OK si vous ne voulez pas gérer les comptes.

2.3.3. Autres Actions avec DBCA



Configure database options in a database: Permet d'ajouter des options dans une base de données existantes.

Delete a database: Supprimer une base de données.

Manage Templates: Permet de réaliser les tâches suivantes avec les templates.

- **Créer a Template:**
 - **From an existing Template:** Utilise un template existant pour créer un nouveau tempate.
 - **From an existing database (structure only):** Requier une connexion à une base de données existantes et utilise la structure de cette base de données comme point de départ.
 - **From an existing database (structure as well as data):** Requier une connexion à une base de données existantes et utilise la structure de cette base de données; cependant il capture les fichiers de données.
- **Delete a database template:** Permet de supprimer définitivement le template d'une base de données du système.

3. Module 03 : Gestion de la base de données

3.1. Contrôler la base de données

3.1.1. Gestion des outils

3.1.1.1. Démarrer et arrêter iSQL*Plus

L'Application Serveur iSQL*Plus doit être lancée avant de pouvoir lancer une session iSQL*Plus. L'utilisation d'une ligne de commande et d'un service Windows sont nécessaires pour démarrer et arrêter iSQL*Plus sous Windows. L'Application Serveur iSQL*Plus est démarrée par défaut lors de l'installation de la base de données Oracle.

Pour démarrer et arrêter l'Application Serveur iSQL*Plus sous Unix (Linux) :

- Démarrer une session en terminal
- Entrer cette commande pour démarrer : **isqlplus start**
- Entrer cette commande pour arrêter : **isqlplus stop**

Remarque : Vous devrez naviguer dans le répertoire `$ORACLE_HOME/bin` si ce répertoire n'est pas présent dans votre PATH.

```
$ isqlplusctl start
iSQL*Plus 10.1.0.2.0
Copyright (c) 2004 Oracle. All rights reserved.
Starting iSQL*Plus ...
iSQL*Plus started.
```

```
$ isqlplusctl stop
iSQL*Plus 10.1.0.2.0
Copyright (c) 2004 Oracle. All rights reserved.
Stopping iSQL*Plus ...
iSQL*Plus stopped.
```

Pour démarrer et arrêter l'Application Serveur iSQL*Plus sous Windows :

- Sélectionnez *Services* dans *Démarrer > Tous les programmes > Administrative Tools menu*.
- Localisez le service Windows iSQL*Plus, OracleOracleHomeNameiSQL*Plus.
- Démarrer le service Windows pour démarrer l'Application Serveur iSQL*Plus, ou arrêter le service pour l'arrêter.
- Alternativement, vous pouvez démarrer iSQL*Plus depuis en ligne de commande comme sous Unix (Linux).

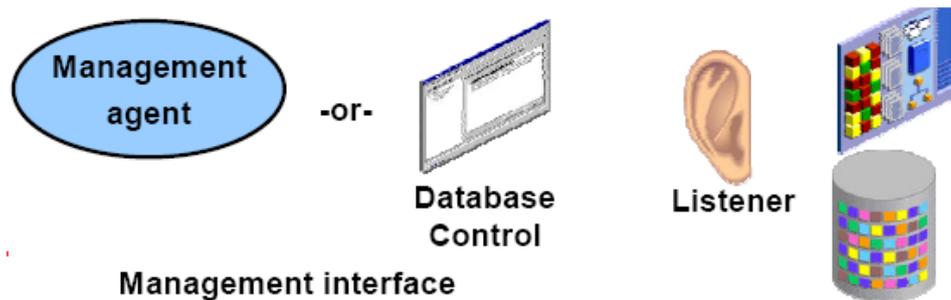
3.1.1.2. Gestion du Framework

Il y a trois composants majeurs dans la gestion du Framework Oracle Database 10g :

- La base de données et l'instance.
- Un listener permettant les connexions à la base de données.
- Une interface de gestion. Il peut s'agir d'un agent de gestion connectant le serveur à Oracle Enterprise Manager Grid Control, ou à Oracle Enterprise Manager Database Control.

Chacun de ces composants doit être démarré avant de pouvoir utiliser ses services et doit être arrêté proprement lorsque le serveur hébergeant la base de données Oracle 10g est arrêté.

Le premier composant devant être démarré est l'interface de gestion. Ensuite l'interface de gestion pourra être utilisée pour démarrer les autres composants.



3.1.2. Démarrer et arrêter Database Control

Pour les bases de données non connectées au Framework Grid Control, Oracle fournit une console de gestion individuelle appelée Database Control. Chaque base de données gérée avec Database Control possède une installation séparée de Database Control, et chaque Database Control ne peut gérer qu'une seule base de données. Database Control requiert qu'un processus **dbconsole** soit démarré avant l'utilisation.

Pour démarrer le processus **dbconsole** :

```
emctl start dbconsole
```

Pour arrêter le processus **dbconsole** :

```
emctl stop dbconsole
```

Pour voir le status du processus **dbconsole** :

```
emctl status dbconsole
```

Remarque : Vous devez vous placer dans le répertoire **\$ORACLE_HOME/bin** si ce répertoire n'est pas présent dans votre PATH.

Database Control utilise un processus agent du côté serveur. Ce processus agent démarre et s'arrête automatiquement lorsque le processus **dbconsole** est démarré ou arrêté.

```
$ emctl start dbconsole
Essential - ORACLE DBA1 10g 41 / 175
TZ set to US/Pacific
Oracle Enterprise Manager 10g Database Control Release 10.1.0.2. 0
Copyright (c) 1996, 2004 Oracle Corporation. All rights reserved.
http://edrsr9p1.us.oracle.com:5500/em/console/aboutApplication
Starting Oracle Enterprise Manager 10g Database Control
..... started.
-----
Logs are generated in directory
/u01/app/oracle/product/10.1.0/db_1/edrsr9p1.us.oracle.com_orcl/sy
an/log
```

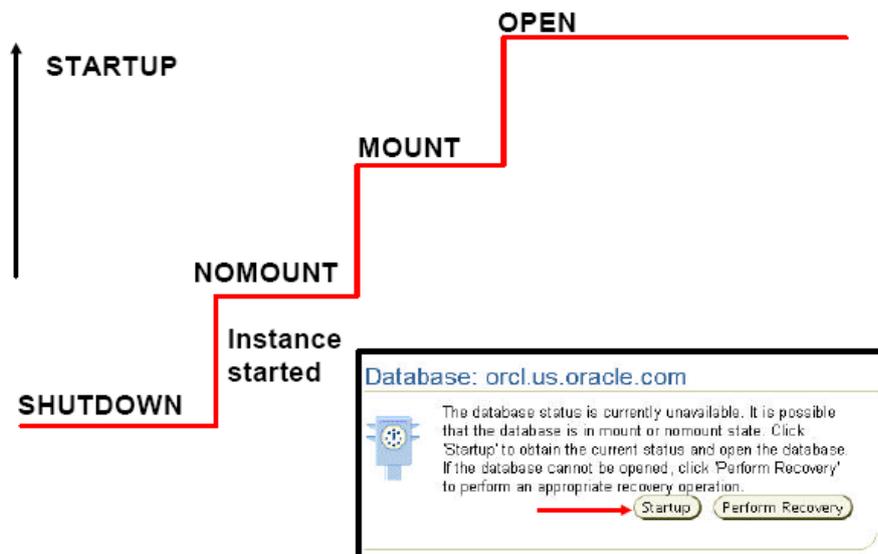
```

$ emctl stop dbconsole
TZ set to US/Pacific
Oracle Enterprise Manager 10g Database Control Release 10.1.0.2. 0
Copyright (c) 1996, 2004 Oracle Corporation. All rights reserved.
http://edrsr9p1.us.oracle.com:5500/em/console/aboutApplication
Stopping Oracle Enterprise Manager 10g Database Control ...
... Stopped.

```

3.1.3. Etapes de démarrage

3.1.3.1. Démarrer l'instance (NOMOUNT)



Une instance est démarrée en mode **NOMOUNT** seulement durant la création de la base de données ou la recréation des fichiers contrôles.

Démarrer une instance comprend les points suivants :

- Lecture du fichier d'initialisation situé dans `$ORACLE_HOME/dbs` dans l'ordre suivant :
 - En premier le `spfileSID.ora`
 - S'il n'est pas trouvé, `spfile.ora`
 - S'il n'est pas trouvé, `intiSID.ora`

Spécifier le paramètre **PFILE** avec la commande **STARTUP** remplace celui par défaut.

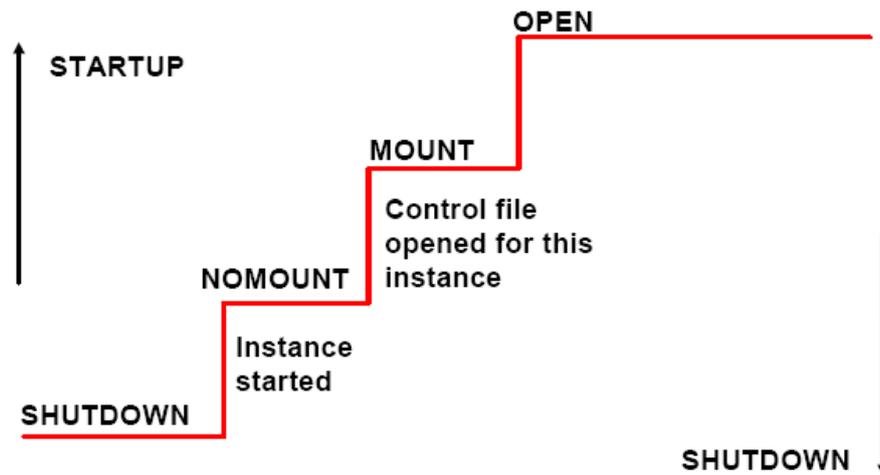
- Allocation de la SGA
- Démarrage des processus d'arrière plan
- Ouverture du fichier `alertSID.log` et des fichiers de trace.

3.1.3.2. Monter la base de données (MOUNT)

Pour réaliser des opérations de maintenances spécifiques, vous démarrez une instance et montez la base de données mais vous ne l'ouvrez pas.

Par exemple, la base de données doit être montée mais pas ouverte pendant les tâches suivantes :

- Renommer les fichiers de données (les fichiers de données pour un tablespaces offline peuvent être renommés lorsque la base est ouverte).
- Activer et désactiver les options d'archivage des fichiers de redo log archivés.
- Récupération complète de la base de données.

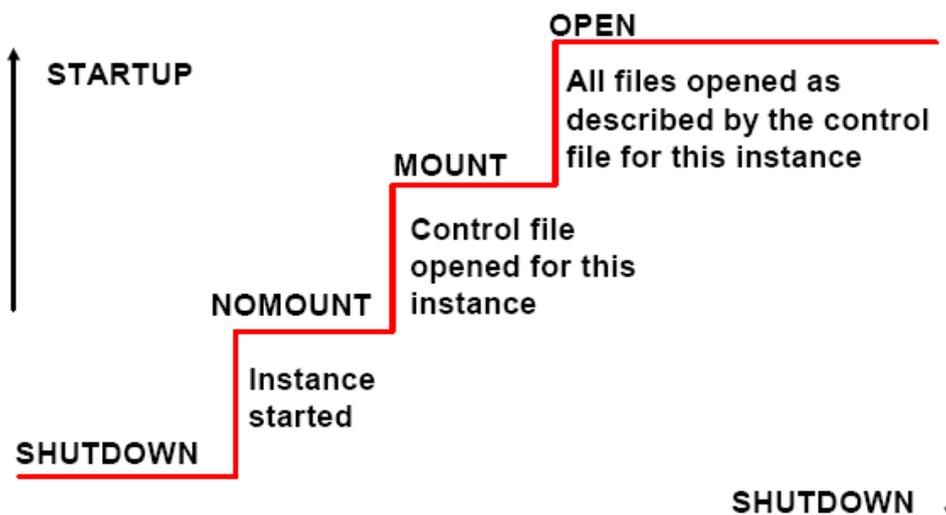


Montée une base de données comprend les étapes suivantes :

- Associer la base de données avec une instance démarrée précédemment.
- Situer et ouvrir les fichiers de contrôles spécifiés dans le fichier de paramètres.
- Lire les fichiers de contrôles pour obtenir les noms et statuts des fichiers de données et des fichiers de redo. Cependant, aucune vérification n'est faite concernant l'existence des fichiers de données et des fichiers de redo logs à ce moment.

3.1.3.3. Ouverture de la base de données (OPEN)

Un **startup** normal signifie que l'instance est démarrée et que la base de données est montée et ouverte. A la suite de cette opération, n'importe quel utilisateur valide peut se connecter à la base de données et réaliser des opérations sur les données.



L'ouverture d'une base de données comprend les tâches suivantes :

- Ouverture des fichiers de données.
- Ouverture des fichiers de redo logs.

Si aucun fichier de données ou de redo n'est présent lorsque vous essayez d'ouvrir la base, le serveur Oracle retourne une erreur.

Durant cette dernière étape, le serveur Oracle vérifie que tous les fichiers de données et de redo peuvent être ouverts et vérifie la consistance de la base de données. Si nécessaire, le processus d'arrière-plan **SMON** lance une récupération de l'instance.

3.1.4. Mode de fermeture

3.1.4.1. Arrêter la base de données

Il est nécessaire d'arrêter la base de données pour réaliser des sauvegardes offline de toutes les structures physiques et pour que les modifications apportées aux paramètres d'initialisation statiques soient appliqués.

| Shutdown Mode | A | I | T | N |
|-------------------------------------|----|-----|-----|-----|
| Allow new connections | No | No | No | No |
| Wait until current sessions end | No | No | No | Yes |
| Wait until current transactions end | No | No | Yes | Yes |
| Force a checkpoint and close files | No | Yes | Yes | Yes |

Shutdown mode:

- **A = ABORT**
- **I = IMMEDIATE**
- **T = TRANSACTIONAL**
- **N = NORMAL**



Pour arrêter une instance vous devez vous connecter en tant que *SYSOPER* ou *SYSDBA* et utiliser la commande suivante :

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

C'est cette commande qui est générée lorsque vous cliquez sur Shutdown.

3.1.4.2. Options d'arrêt : base de données consistante

SHUTDOWN NORMAL :

Normal est le mode d'arrêt par défaut. L'arrêt de la base de données dans le mode normal se déroule dans les conditions suivantes :

- Aucune nouvelle connexion n'est possible.
- Le serveur Oracle attend que tous les utilisateurs se déconnectent avant de s'arrêter.
- Le contenu du database buffer et du redo log buffer est écrit sur le disque.
- Les processus d'arrière plan sont arrêtés, et la mémoire utilisée par la SGA est désallouée.
- Oracle ferme et démonte la base de données avant d'arrêter l'instance.
- Le prochain démarrage n'aura pas besoin d'une récupération d'instance.

SHUTDOWN TRANSACTIONAL :

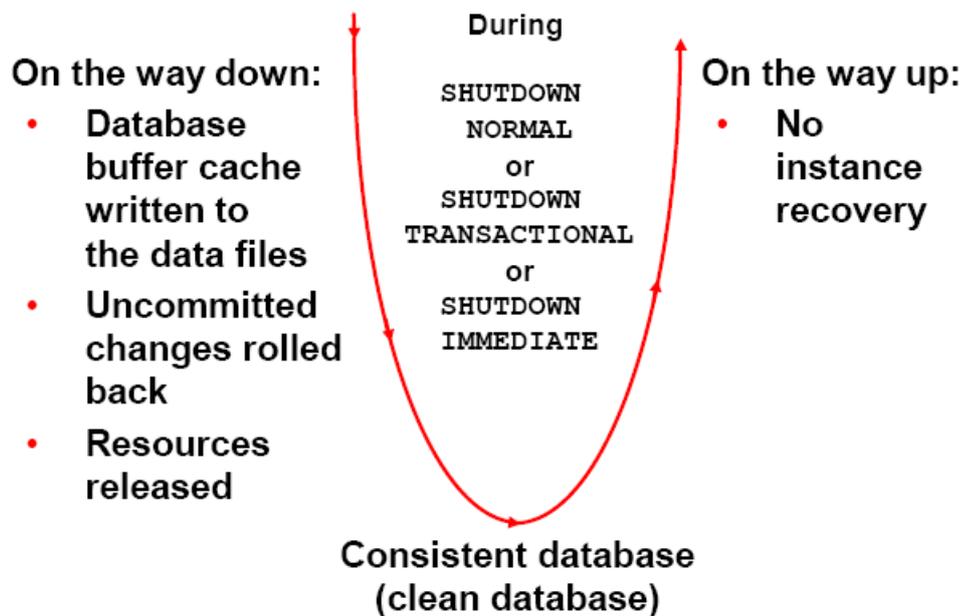
Un arrêt en mode transactionnel évite que les clients perdent leur travail. L'arrêt de la base de données dans le mode transactionnel se déroule dans les conditions suivantes :

- Aucun client ne peut lancer de nouvelles transactions.
- Un client est déconnecté lorsque sa transaction en cours se termine.
- Lorsque toutes les transactions sont terminées, arrêt est immédiat.
- Le prochain démarrage n'aura pas besoin d'une récupération d'instance.

SHUTDOWN IMMEDIATE :

L'arrêt de la base de données dans le mode immediate se déroule dans les conditions suivantes :

- Les instructions SQL en cours ne sont pas terminées.
- Le serveur Oracle n'attend pas que les clients, qui sont connectés à la base de données, se déconnectent.
- Oracle roll back les transactions actives et déconnectent tous les utilisateurs connectés.
- Oracle ferme et démonte la base de données avant d'arrêter l'instance.
- Le prochain démarrage n'aura pas besoin d'une restauration d'instance.

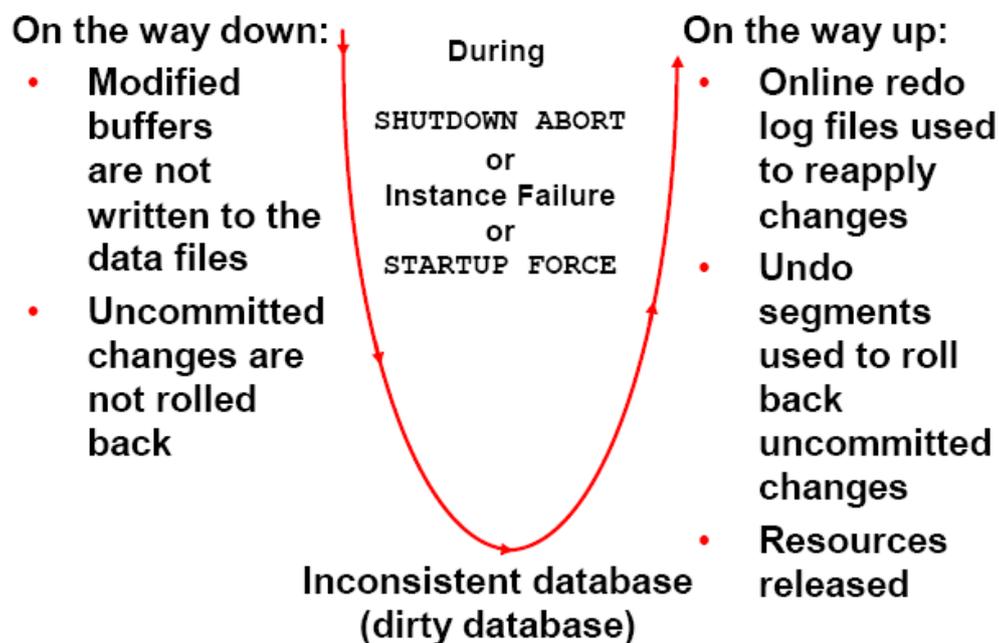


3.1.4.3. Shutdown Options: Base de données inconsistante

SHUTDOWN ABORT :

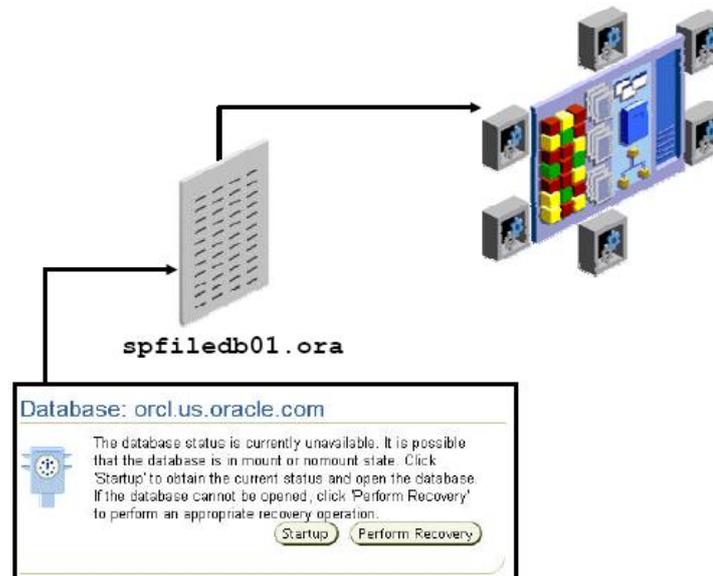
Si les options d'arrêt **NORMAL** et **IMMEDIATE** ne fonctionnent pas, vous pouvez arrêter l'instance. L'arrêt d'une instance se déroule dans les conditions suivantes :

- Les instructions SQL en cours de traitement sont immédiatement terminées.
- Le serveur Oracle n'attend pas que les clients, qui sont connectés à la base de données, se déconnectent.
- Le contenu du database buffer et du redo log buffer n'est pas écrit sur le disque.
- Les transactions non committées ne sont pas rollbackées.
- L'instance se termine sans fermer les fichiers.
- La base de données n'est pas fermée ni démontée.
- Le prochain démarrage nécessitera une récupération de l'instance, qui est exécutée automatiquement.



3.1.5. Fichier de paramètres d'initialisation

Pour démarrer une instance, Oracle doit lire un fichier de paramètres d'initialisation (PFILE) ou un fichier de paramètres serveur (SPFILE). Ces fichiers contiennent une liste des paramètres pour l'instance ou la base de données. Oracle stocke traditionnellement les paramètres d'initialisation dans un fichier texte. Vous pouvez également choisir de maintenir les paramètres d'initialisation dans un fichier binaire (SPFILE).



Les paramètres d'initialisation stockés dans le SPFILE sont persistants, dans ce cas les changements effectués, pendant que l'instance est lancée, peuvent persister pendant le démarrage et l'arrêt de celle-ci. Dans la majorité des cas, il est nécessaire de spécifier une valeur pour certains paramètres dits basiques pour avoir des performances raisonnables. Dans des situations rares, des modifications des paramètres avancés seront nécessaires pour optimiser les performances.

Oracle suggère de créer un SPFILE comme moyen dynamique pour maintenir les paramètres d'initialisation. En utilisant un SPFILE, vous pouvez stocker et gérer vos paramètres d'initialisation persistants dans le fichier situé du côté serveur.

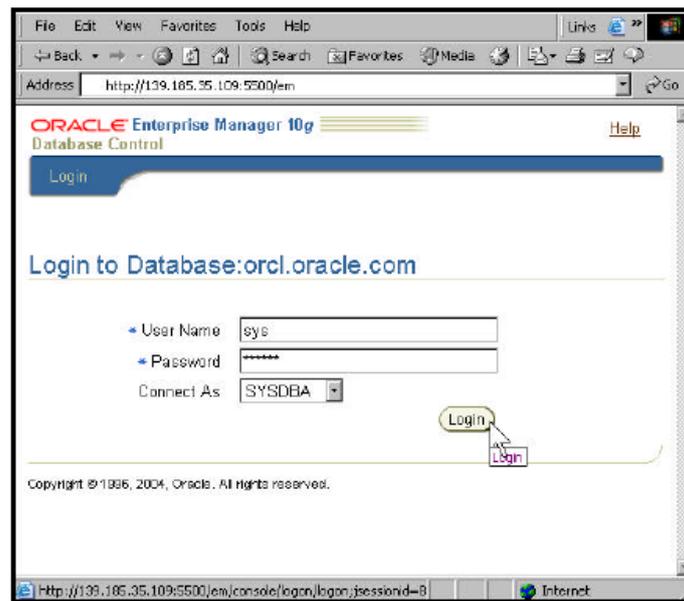
3.1.6. Accéder au Database Control

3.1.6.1. Authentification à Database Control

Ouvrez votre navigateur internet et entrez l'adresse suivante (le port par défaut est 5500) :

<http://hostname:portnumber/em>

Si la base de données est démarrée, Enterprise Manager affiche la page de connexion à Database Control. Connectez-vous avec le nom d'un utilisateur autorisé à se connecter à Database Control. Il s'agit initialement de *SYS*, *SYSMAN* or *SYSTEM*. Utilisez le mot de passe que vous avez spécifié pour le compte durant l'installation de la base de données.



Si la base de données est hors service et doit être démarrée, Enterprise Manager affiche une page de démarrage/arrêt de la base et une page permettant la récupération de la base (Perform Recovery). Si c'est le cas, cliquez sur le bouton démarrage/arrêt. Vous êtes ensuite invité à entrer, pour les bases de données locales et distantes, le login et le mot de passe. Pour le nom de l'utilisateur et le mot de passe de la base de données, utiliser le login **SYS** et le mot de passe qu'avez saisi durant l'installation. Cliquez sur Ok pour démarrer la base de données. Dans la page Confirmation, cliquez sur Yes pour démarrer la base de données en mode open.

3.1.6.2. SYSOPER et SYSDBA

SYSOPER : Il s'agit d'un rôle d'administration spécial de la base de données qui permet à l'administrateur de la base de données d'effectuer les actions **STARTUP**, **SHUTDOWN**, **ALTER DATABASE OPEN/MOUNT**, **ALTER DATABASE BACKUP**, **ARCHIVE LOG**, et **RECOVER**, et inclus le privilège **RESTRICTED SESSION**. Lorsque vous vous connectez en tant que **SYSOPER**, vous êtes dans le schéma public.

SYSDBA : Il s'agit d'un rôle d'administration spécial de la base de données qui contient tous les privilèges système des privilèges système **ADMIN OPTION** et **SYSOPER**. **SYSDBA** peut également utiliser la clause **CREATE DATABASE** et la récupération partielle. Lorsque vous vous connectez en tant que **SYSDBA**, vous vous trouvez dans le schéma de l'utilisateur **SYS**.

3.1.6.3. Page d'accueil de Database

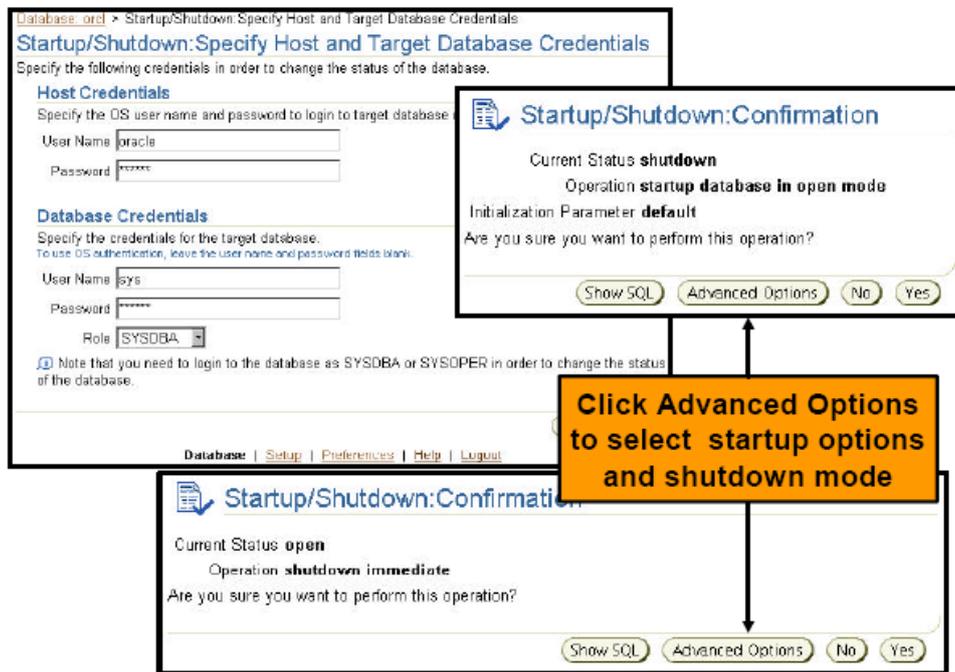
La page d'accueil vous donne accès aux pages de performances, d'administration et de maintenance de votre base de données. Les différentes sections de la page accueil, et les liens relatifs, fournissent une quantité d'informations à propos de l'environnement et de l'activité de votre base de données.



Pour donner l'accès à la page de gestion de la base de données à d'autres utilisateurs :

1. Démarrer votre navigateur Internet et connectez vous au Database Control en tant qu'utilisateur **SYS** ou **SYSTEM**.
2. Cliquez sur Setup en haut de la page accueil
3. Cliquez sur Administrators dans la barre de navigation de gauche.
4. Cliquez sur Create pour créer un nouvel utilisateur d'Enterprise Manager en donnant les privilèges de gestion à un utilisateur existant de la base de données.
5. Cliquez sur l'icône flashlight à côté du champ Name et sélectionnez un utilisateur de la base de données existant dans le pop-up Windows.
6. Entrer le mot de passe pour les utilisateurs sélectionnez et cliquez sur Finish.

3.1.6.4. Démarrage et arrêt de la base de données



Lorsque vous cliquez sur démarrer ou arrêter vous serez invité à entrer les identifiants/mots de passe pour se connecter sur la machine hôte et à la base de données.

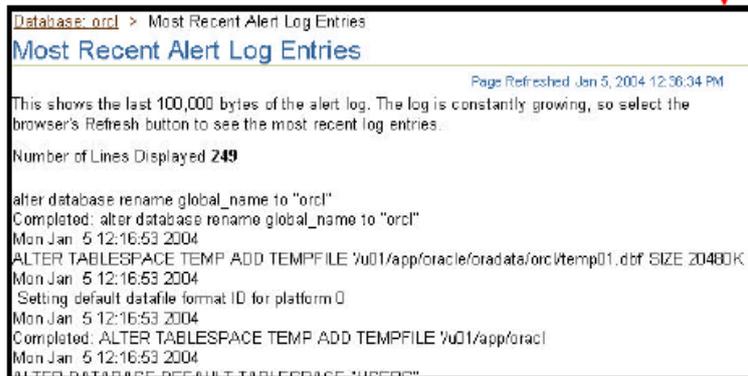
Vous pouvez ensuite cliquer sur Advanced Options pour sélectionner les options de démarrage ou le mode d'arrêt de la base de données souhaité. Vous pouvez cliquer sur Show SQL pour voir l'instruction qui sera utilisé pour le démarrage ou l'arrêt.

Lorsque vous démarrez la base de données, vous choisissez l'état dans lequel la base démarrera. Les scénarios suivants montrent les différentes étapes de démarrage d'une instance.

3.1.6.5. Visualiser les logs d'alerte

Chaque base de données possède un **alert_sid.log**. Ce fichier est sur le serveur avec la base de données et est stocké dans le répertoire spécifié par le paramètre d'initialisation **background_dump_dest**. Le fichier d'alerte d'une base de données est un fichier log chronologique de messages et d'erreurs, incluant :

- Toutes les erreurs internes (*ORA-600*), les erreurs dues aux blocs corrompus (*ORA-1578*), et les erreurs de deadlock (*ORA-60*) qui surviennent.
- Les opérations administratives, comme les instructions SQL **CREATE, ALTER, DROP DATABASE, TABLESPACE, ROLLBACK SEGMENT** et celles d'Enterprise Manager ou de SQL*Plus **STARTUP, SHUTDOWN, ARCHIVE LOG** et **RECOVER**.
- Plusieurs messages et erreurs à propos des fonctions du serveur partagé et des dispatchers
- Les erreurs durant le rafraîchissement automatique d'une vue matérialisée.

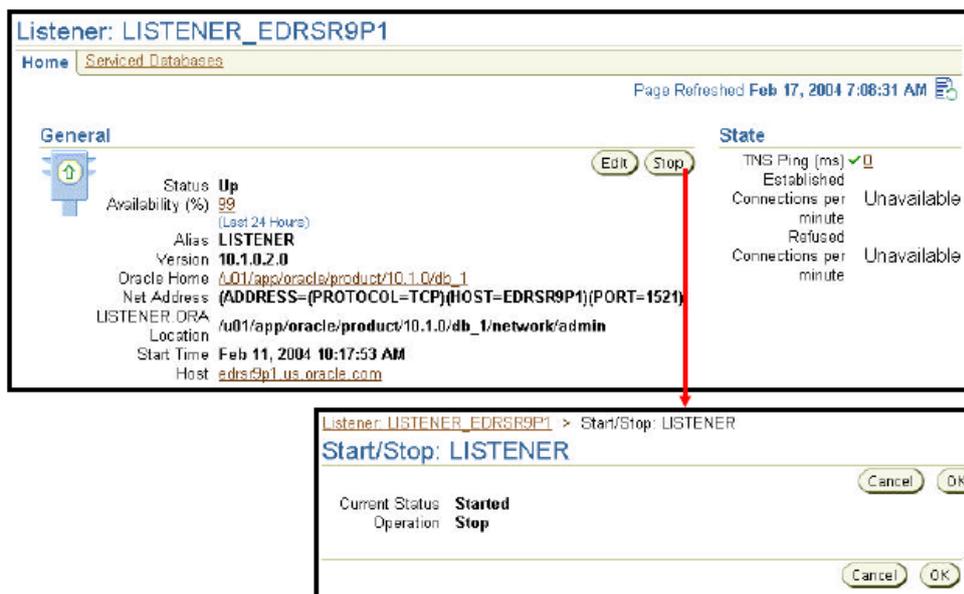


Enterprise Manager gère le fichier de log d’alerte et vous notifie de toutes les erreurs critiques. Vous pouvez également consulter les logs pour voir les erreurs non critiques et les messages informatifs. De plus le fichier peut grossir et devenir ingérable si il est laissé sans surveillance. Vous devez faire une sauvegarde du fichier d’alerte de temps en temps et supprimer le fichier d’alerte courant.

Si la base de données tente d’écrire à nouveau dans le fichier d’alerte, celle-ci en recréera un.

3.1.6.6. Modifier le statut du Listener

Depuis la page accueil, cliquez sur le nom du listener pour ouvrir la page du Listener. Cliquez sur Stop pour arrêter le listener s’il est lancé ou Start s’il est arrêté. Vous devez vous connectez sur la machine locale avec un utilisateur de l’OS si vous ne l’avez pas déjà fait. Ce sera cet utilisateur qui démarrera et arrêtera le listener.



La méthode en ligne de commande pour démarrer, arrêter, et voir le status du listener est :

```
lsnrctl START [listener_name]
```

```
lsnrctl STOP [listener_name]
```

```
lsnrctl STATUS [listener_name]
```

Où **listener_name** est le nom du listener défini dans le fichier **listener.ora**. Il n'est pas nécessaire d'indiquer le nom du listener si vous utilisez celui par défaut appelé **LISTENER**.

La commande **STATUS** fournit les informations sur le statut du listener, c'est-à-dire un résumé des paramètres de configuration du listener, les adresses des protocoles écoutés, et un résumé des services enregistrés avec le listener.

3.1.6.7. Fichier de paramètres d'initialisation

La base de données Oracle fournit un nombre de paramètres d'initialisation pour optimiser ces opérations dans divers environnements. Seulement une petite partie de ces paramètres doivent être explicitement déclarés à cause des valeurs par défauts des paramètres qui sont adéquats dans la majorité des cas. Il y a 28 paramètres basiques.

The image shows two screenshots from Oracle Enterprise Manager. The top screenshot shows the 'Administration' tab with 'All Initialization Parameters' highlighted. A red arrow points from this link to the bottom screenshot, which is a detailed view of the 'Initialization Parameters' page. This page shows a table of parameters with columns for Name, Help, Revisions, Value, Type, Basic, Default, Dynamic, and Category.

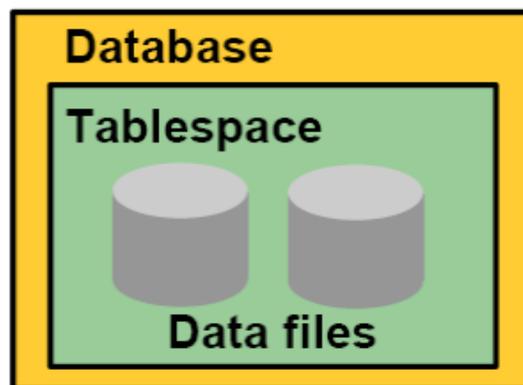
| Name | Help | Revisions | Value | Type | Basic | Default | Dynamic | Category |
|------------------|------|-----------|---|---------|-------|---------|---------|--------------------|
| cluster_database | | | FALSE | Boolean | ✓ | ✓ | | Cluster Database |
| compatible | | | 10.1.0.1.0 | String | ✓ | | | Miscellaneous |
| control_files | | | %D1/app/oracle/oradata/orcl/control01.ctl, %D1/app/oracle/oradata/orcl/control02.ctl | String | ✓ | | | File Configuration |

3.2. Structures de stockage

3.2.1. Tablespaces et fichiers de données

Les bases de données, les tablespaces et les fichiers de données sont étroitement liés, cependant il y a une différence importante :

- Une base de données Oracle est composée d'une ou plusieurs unités logiques de stockage appelées tablespaces. Ils stockent collectivement toutes les données de la base.
- Chaque tablespace d'une base de données comporte un ou plusieurs fichiers appelés fichiers de données. Lesquels sont des structures physiques conformes au système d'exploitation sur lequel le serveur Oracle fonctionne.
- Les données de la base sont stockées dans les fichiers de données, qui constituent chaque tablespace de la base. Par exemple : la plus simple base de données Oracle aurait un tablespace et un fichier de données. Une autre base de données peut avoir trois tablespaces, comprenant chacun deux fichiers de données chacun (donc un total de six fichiers de données). Une seule base de données peut avoir au maximum 65 535 fichiers de données



Les tablespaces assignent de l'espace aux extents. Les tablespaces peuvent être créés en utilisant une des deux méthodes suivantes pour gérer l'espace libre et utilisé.



Gestion locale des tablespaces :

Les extents sont gérés dans le tablespace via des bitmaps. Chaque bit, dans un bitmap, correspond à un bloc ou à un groupe de bloc. Lorsqu'un extent est alloué ou libre pour être réutilisé, le serveur Oracle change la valeur du bitmap pour mettre à jour le nouveau statut des blocs.

Gestion des tablespaces par le dictionnaire des données :

Les extents sont gérés par le dictionnaire des données. Le serveur Oracle met à jour les tables appropriées dans le dictionnaire des données dès qu'un extent est alloué ou désalloué. Pour une compatibilité avec les versions précédentes, vous devrez utiliser la gestion locale pour tous les tablespaces.

3.2.2. Gestion des tablespaces

3.2.2.1. Création d'un nouveau tablespace

Pour créer un tablespace, suivez les étapes suivantes :

1. Aller sur la page des tablespaces. Dans l'onglet Administration cliquer sur tablespace.
2. Cliquer sur le bouton Créer
Remarque : Si vous voulez recréer un tablespace, identique à un autre, sélectionnez un tablespace existant et cliquez sur Create Like dans la liste déroulante. Cliquez Go.
La page générale pour créer un tablespace apparaît.
3. Entrer le nom du tablespace
4. Sous la rubrique gestion des extents, sélectionner « locally managed ». Les extents d'un tablespace « locally managed » sont contrôlés efficacement par le serveur Oracle. Pour les tablespaces gérés par le dictionnaire, vous devez contrôler activement vos extents. Des tablespaces contrôlés par le dictionnaire sont désapprouvés et déconseillés par Oracle.
5. Sous la rubrique Type, sélectionner Permanent. Les tablespaces permanents stockent les objets permanents de base de données créés par le système ou les utilisateurs.
6. Sous la rubrique Statut, sélectionner Lecture/Ecriture. En sélectionnant ce statut, les utilisateurs peuvent lire et écrire dans le tablespace une fois que celui-ci est créé. C'est le mode par défaut.
7. Dans la zone « fichier de données », cliquer sur ajouter pour ajouter d'autres fichiers de données à votre tablespace. Un tablespace doit avoir au moins un fichier de données. Les tablespaces de type Bigfile sont utilisés avec de grosses bases de données utilisant des outils tels ASM (Automatic Storage Management) ou autres systèmes de gestion des volumes logiques supportant le striping ou le RAID, et permettent une extension dynamique de ces volumes de stockage logique.
8. Sur la page ajouter des fichiers de données, entrer le nom du fichier. Accepter le chemin du répertoire et la taille par défaut.
9. Dans la zone Stockage, sélectionner « Automatically extend datafile when full (AUTOEXTEND) » et indiquer la taille de l'extend supplémentaire qui sera ajouté au fichier de données chaque fois qu'il sera rempli. Laisser la taille du fichier maximum à illimitée. Cliquez sur OK pour terminer et retourner sur la page de création des Tablespaces.
10. Cliquer sur l'onglet Stockage pour afficher la page de stockage propre à la création des tablespaces.
11. Accepter tous les paramètres par défaut de cette page.
12. Cliquer sur l'onglet « thresholds ». Cette page vous permet de paramétrer les seuils d'alertes surveillés pour l'utilisation de l'espace. Vous recevrez une alerte et une aide sur l'action à réaliser quand le seuil sera atteint.
13. Après avoir paramétré ces seuils, cliquez sur OK pour ajouter le tablespace. Vous retournez sur la page des tablespaces où vous recevez la confirmation de la création du tablespace. Vous pouvez voir votre nouveau tablespace dans la section Résultats.

Database: orcl.oracle.com > Tablespaces > Create Tablespace Logged in As SYS

Create Tablespace

Show SQL Cancel OK

General Storage Thresholds

* Name

| Extent Management | Type | Status |
|--|--|---|
| <input checked="" type="radio"/> Locally Managed | <input checked="" type="radio"/> Permanent | <input checked="" type="radio"/> Read Write |
| <input type="radio"/> Dictionary Managed | <input type="checkbox"/> Set as default permanent tablespace | <input type="radio"/> Read Only |
| | <input type="radio"/> Temporary | <input type="radio"/> Offline |
| | <input type="checkbox"/> Set as default temporary tablespace | |
| | <input type="radio"/> Undo | |

Datafiles

Use bigfile tablespace
Tablespace can have only one datafile with no practical size limit.

Add

| Select | Name | Directory | Size (MB) |
|----------------------------------|-----------------|-------------------------------|-----------|
| <input checked="" type="radio"/> | inventory01.dbf | /u01/app/oracle/oradata/orcl/ | 50.00 |

Edit Remove

3.2.2.2. Espace de stockage des tablespaces gérés localement

Les extents d'un tablespace géré localement peut-être alloués de deux manières :

- Automatique : appelé aussi auto allocation, spécifie que la taille des extents dans le tablespace est gérée par le système. Vous ne pouvez pas spécifier la taille des extents et vous ne pouvez pas spécifier le paramètre automatique pour un tablespace temporaire.
- Uniforme : Les extents d'un tablespaces géré de cette manière ont une taille fixe que vous spécifiez. La taille par défaut est de 1 Mo. Tout les extents du tablespace temporaire ont une taille uniforme, cette option est facultative pour un tablespace temporaire. Vous ne pouvez pas spécifier cette option pour un tablespace undo.

Gestion de l'espace des segments dans un tablespace géré localement :

- Automatique : Oracle utilise les bitmaps pour gérer l'espace libre dans un segment. Dans ce cas, un bitmap est un tableau décrivant le statut de chaque bloc de données dans un segment. Que ce soit plus ou moins d'espace disponible dans les blocks de données, ce nouvel état est reflété dans un bitmap. Les bitmaps laissent Oracle gérer automatiquement l'espace libre. Cette forme de gestion de l'espace s'appelle la gestion automatique de l'espace des segments.
- Manuelle : Indique à Oracle que vous voulez utiliser les listes de bloc non alloué pour contrôler l'espace libre dans des segments. Les listes de blocs de données sont des listes qui ont l'espace disponible pour l'insertion de lignes. Cette forme de gestion de l'espace des segments s'appelle la gestion manuelle en raison de la nécessité de spécifier et de personnaliser les paramètres de stockage **PCTUSED**, **FREELIST**, **FREELIST GROUPS** pour des objets créés dans un tablespace. Cette permet une rétrocompatibilité.

Les tablespaces gérés localement apportent les avantages suivants par rapport à une gestion par le dictionnaire des données :

- La gestion locale évite des opérations récursives de gestion d'espace. Cela peut arriver dans une gestion par le dictionnaire des données si l'allocation ou la libération d'espace d'un extent survient à la suite d'une autre opération qui occupe ou libère, elle aussi, de l'espace d'un segment undo ou une table du dictionnaire.
- Les tablespaces gérés localement ne peuvent pas écrire dans les tables du dictionnaire des données, cela réduit donc les contentions (collisions) au niveau de ces tables.
- La taille des extents qui sont gérés localement peut être déterminée automatiquement par le système d'exploitation.
- Les changements d'état des blocs (libre/occupé) dans un bitmap ne génèrent pas d'entrées de undo parce que il n'y a pas de mis à jour des tables du dictionnaire des données (sauf quelque cas spéciaux tel que les informations sur le quota des tablespaces).

Database: orcl.us.oracle.com > Tablespaces > Create Tablespace Logged in As SYS

Create Tablespace

Show SQL Cancel OK

General Storage Thresholds

Extent Allocation

Automatic

Uniform

Size KB

Segment Space Management

Automatic

Objects in the tablespace automatically manage their free space. It offers high performance for free space management.

Manual

Objects in the tablespace will manage their free space using free lists. It is provided for backward compatibility.

3.2.2.3. Tablespace dans une base de données pré configurer

Les tablespaces suivant sont créé quand vous pré configurez la base de données :

- **SYSTEM** : le tablespace **SYSTEM** est utilisé par le serveur Oracle pour gérer la base de données. Il contient le dictionnaire des données et des informations administratives sur la base. Son contenu se trouve dans le schéma de **SYS** et ne peut être vu que celui-ci ou tout autre administrateur ayant les privilèges requis.
- **SYSAUX** : c'est un tablespace auxiliaire au tablespace **SYSTEM**. Quelques composants et produits qui utilisaient le tablespace **SYSTEM**, dans les versions précédentes d'Oracle, utilise maintenant le tablespace **SYSAUX**. Chaque base de donnée Oracle 10g doit posséder un tablespace **SYSAUX**.
- **TEMP** : Ce tablespace est utilisé pour stocker des tables temporaires et des indexes. Il est utilisé par exemple pour des opérations de tris. Chaque base de données devrait avoir un tablespace temporaire qui est assigné aux utilisateurs comme leur tablespace temporaire. Dans la pré-configuration de la base, le tablespace **TEMP** est considéré comme le tablespace temporaire par défaut. Dans ce cas, si aucun tablespace temporaire n'est spécifié à un utilisateur lors de sa création, Oracle lui assignera automatiquement celui-ci.
- **UNDOTBS1** : c'est le tablespace undo utilisé par le serveur Oracle pour stocker des entée de undo. Chaque base de données doit avoir un tablespace **UNDO** c'est pour cela qu'il est créé lors de la création de la base de données.
- **USERS** : Ce tablespace est utilisé pour stocker les objets et les données des utilisateurs. C'est le tablespace par défaut, pour une base de données pré configurée, pour tous les objets créées par un utilisateur non system. Pour les utilisateurs **SYS** et **SYSTEM**, le tablespace par défaut est le tablespace **SYSTEM**.
- **EXEMPLE** : Ce tablespace contient des exemples de schéma qui peuvent être installés lors de la création de la base de données.

| Select | Name | Type | Extent Management | Segment Management | Status | Size (MB) | Used (MB) | Used (%) |
|----------------------------------|----------|-----------|-------------------|--------------------|--------|-----------|-----------|----------|
| <input checked="" type="radio"/> | EXAMPLE | PERMANENT | LOCAL | AUTO | ONLINE | 150.000 | 66.875 | 44.58 |
| <input type="radio"/> | SYSAUX | PERMANENT | LOCAL | AUTO | ONLINE | 230.000 | 222.688 | 96.82 |
| <input type="radio"/> | SYSTEM | PERMANENT | LOCAL | MANUAL | ONLINE | 440.000 | 434.375 | 98.72 |
| <input type="radio"/> | TEMP | TEMPORARY | LOCAL | MANUAL | ONLINE | 26.000 | 25.000 | 96.15 |
| <input type="radio"/> | UNDOTBS1 | UNDO | LOCAL | MANUAL | ONLINE | 25.000 | 11.750 | 47.00 |
| <input type="radio"/> | USERS | PERMANENT | LOCAL | AUTO | ONLINE | 5.000 | 2.750 | 55.00 |

3.2.2.4. Modification d'un tablespace

Après avoir créé un tablespace, vous pouvez le modifier de plusieurs façons selon les besoins de votre système.

Renommage : Entrer simplement le nouveau nom du tablespace et cliquer sur Appliquer.

Changement de statut : Un tablespace peut avoir l'un des trois différents états. Selon le type du tablespace, certains statuts ne sont pas disponibles :

- **Lecture/écriture** (read write): Le tablespace est online, on peut donc lire et écrire dessus.
- **Lecture seulement** (read only) : il ne peut y avoir que des transactions de lecture. Dans cet état, les transactions existantes peuvent être terminées (commit ou rollback), mais aucun autre ordre DML n'est permis, excepté des roll backs sur des transactions existantes qui ont modifiés précédemment des blocs dans le tablespace. Le tablespace est online pendant qu'il est en mode Read-only. Vous ne pouvez pas mettre les tablespaces **SYSTEM** et **SYSAUX** en mode Read-only.

- **Hors ligne** (offline): Vous pouvez mettre offline un tablespace qui est online, celui-ci devient indisponible pour toute utilisation. Le reste de la base de données est accessible aux utilisateurs.

Vous avez différentes méthodes pour mettre un tablespace offline :

- **Normal** : un tablespace peut être mis offline normalement si il n'y a pas d'erreur sur les fichiers de données du tablespace. Oracle effectue un point de contrôle (checkpoint) pour tous les fichiers de données du tablespace avant de le mettre offline.
- **Temporaire** (temporary) : un tablespace peut être mis offline temporairement même si il y a des erreurs dans un fichier de données du tablespace. Oracle met offline les fichiers de données qui ne le sont pas déjà, et fait un checkpoint sur ces fichiers. Si aucun des fichiers n'est offline, mais vous utilisez la méthode temporaire, une récupération des fichiers n'est pas requise pour remettre votre tablespace online. Cependant, si un ou plusieurs fichiers du tablespace sont offline, à cause d'erreurs d'écriture, et que vous souhaitez quand mettre votre tablespace offline temporairement, une récupération de celui-ci sera nécessaire avant de le remettre online.
- **Immédiate** : un tablespace peut être mis offline immédiatement sans que Oracle effectue un point de sauvegarde (checkpoint) sur tous les fichiers de données. Quand vous spécifiez le mode immédiate, la récupération des fichiers du tablespace est requise pour pouvoir le remettre online. Vous ne pouvez pas mettre un tablespace offline immédiatement si votre base de données est en mode **NOARCHIVELOG**.
- **For Recover** : L'option **FOR RECOVER** est dépréciée. Cette syntaxe est encore supportée pour des questions de rétro compatibilité.

Edit Tablespace: EXAMPLE

Show SQL Revert Apply

General Storage Thresholds

Name

Bigfile tablespace **No**

Extent Management

Locally Managed

Dictionary Managed

Type

Permanent

Set as default permanent tablespace

Temporary

Set as default temporary tablespace

Undo

Status

Read Write

Read Only

Offline

Offline Mode

Normal

Temporary

Immediate

For Recover

Datafiles

Add

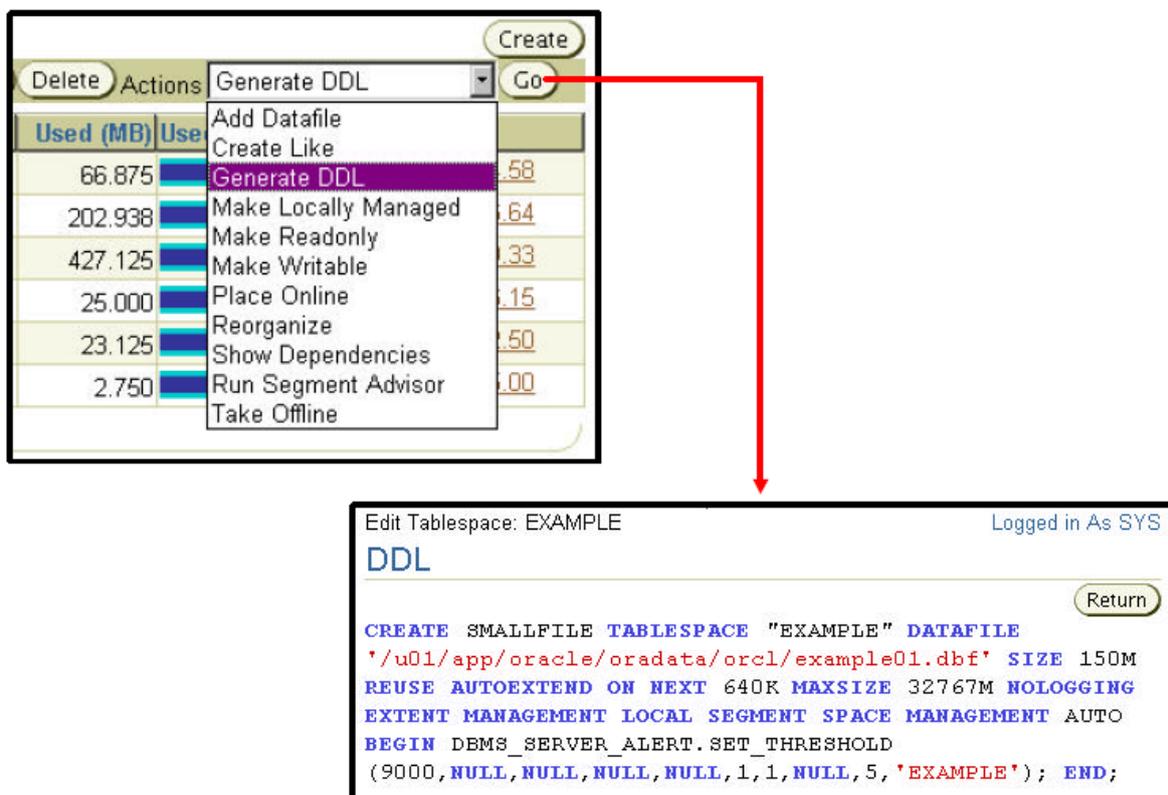
| Select | Name | Directory | Size (MB) | Used (MB) |
|-------------------------------------|---------------|--|-----------|-----------|
| <input checked="" type="checkbox"/> | example01.dbf | /u01/app/oracle/product/10.1.0/oradata/orcl/ | 150.00 | 66.88 |

Edit Remove

3.2.2.5. Actions sur les tablespaces

Grâce au menu Action, vous pouvez effectuer plusieurs tâches sur vos tablespaces. Sélectionnez un tablespace et une action à réaliser :

- **Add Datafile** : Ajoute un fichier de donnée au tablespace, permettant ainsi d'augmenter sa taille.
- **Create Like** : Crée un nouveau tablespace en prenant celui sélectionné comme modèle.
- **Generate DDL** : Affiche l'instruction SQL de la création du tablespace. Vous pouvez copier/coller l'ordre SQL dans un fichier texte pour l'utiliser comme script.
- **Make Locally Managed** : Si l'espace du tablespace est géré actuellement par le dictionnaire des données, la gestion se fera localement.
- **Make Readonly** : Arrête toute écriture sur le tablespace. Vous pouvez terminer les transactions en cours mais aucune autre nouvelle ne sera permise comme toutes autres tâches d'écriture.
- **Make Writable** : Les ordres DML seront désormais possibles sur le tablespace.
- **Place Online** : Si le tablespace est actuellement offline, celui-ci est mis online.
- **Reorganize** : Cela démarre l'outil de réorganisation, que vous pouvez utiliser pour déplacer des objets au sein du tablespace pour récupérer de l'espace qui n'aurait pas pu être utilisé. Cette tâche ne peut être réalisée quand il y a beaucoup d'activité sur les objets du tablespace.
- **Show Dependencies** : affiche les objets qui dépendent de ce tablespace ou ceux dont le tablespace dépend.
- **Run Segment Advisor** : L'outil Segment Advisor vous aide à déterminer si un objet a de l'espace disponible pour s'agrandir. Au niveau du tablespace, les conseils donnés s'appliquent à tous ses segments.
- **Take Offline** : Si un tablespace est actuellement online, cette action permet de le rendre indisponible. Le tablespace n'est pas supprimé, juste indisponible.



The screenshot shows the Oracle Enterprise Manager interface. On the left, a table lists tablespaces with columns for 'Used (MB)' and 'Use'. The 'Actions' menu is open, showing various options. The 'Generate DDL' option is highlighted. A red arrow points from the 'Go' button next to 'Generate DDL' to a separate window titled 'Edit Tablespace: EXAMPLE' which displays the generated SQL DDL statement.

| Used (MB) | Use | Actions |
|-----------|-----|----------------------|
| 66.875 | | Generate DDL |
| 202.938 | | Make Locally Managed |
| 427.125 | | Make Readonly |
| 25.000 | | Make Writable |
| 23.125 | | Place Online |
| 2.750 | | Reorganize |
| | | Show Dependencies |
| | | Run Segment Advisor |
| | | Take Offline |

```

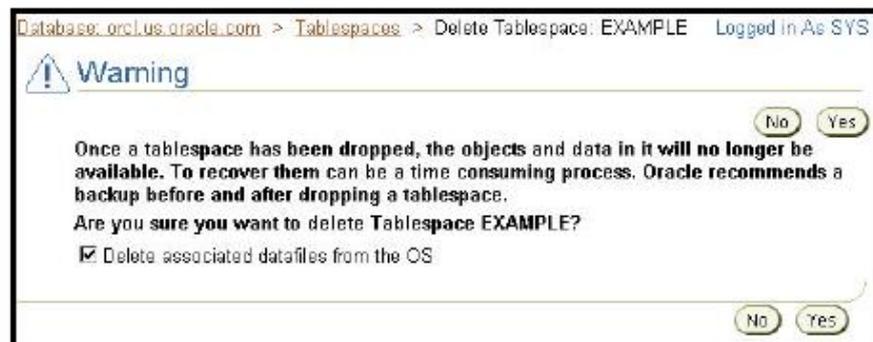
DDL
CREATE SMALLFILE TABLESPACE "EXAMPLE" DATAFILE
'/u01/app/oracle/oradata/orcl/example01.dbf' SIZE 150M
REUSE AUTOEXTEND ON NEXT 640K MAXSIZE 32767M NOLOGGING
EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO
BEGIN DBMS_SERVER_ALERT.SET_THRESHOLD
(9000, NULL, NULL, NULL, NULL, 1, 1, NULL, 5, 'EXAMPLE'); END;
  
```

Remarque : Toutes les actions ne sont pas disponibles pour chaque tablespace. Cela dépend du type de tablespace sélectionné, quelques actions ne peuvent pas être effectuées. Par exemple, vous ne pouvez pas mettre le tablespace **SYSTEM** offline, ou encore mettre le tablespace undo en lecture seule.

3.2.2.6. Suppression d'un tablespace

Vous pouvez supprimer un tablespace et son contenu (les segments contenus dans le tablespace) de la base de données si celui-ci est devenu inutile. Vous devez avoir les privilèges système **DROP TABLESPACE** pour supprimer un tablespace.

Quand vous supprimer un tablespace, le pointeur de fichier du fichier de contrôle qui est associé à la base de données est lui aussi supprimé. Vous pouvez directement supprimer les fichiers sur le système d'exploitation (fichiers de données) constituant le tablespace. Si vous ne supprimer pas directement les fichiers de données en même temps qu'Oracle supprime le tablespace, vous aller devoir utiliser plus tard les commandes appropriées de votre système d'exploitation pour les supprimer.



| Select | Name | Type | Extent Management | Segment Management | Status | Size (MB) | Used (MB) | Used (%) |
|----------------------------------|----------|-----------|-------------------|--------------------|--------|-----------|-----------|----------|
| <input checked="" type="radio"/> | EXAMPLE | PERMANENT | LOCAL | AUTO | ONLINE | 150.000 | 66.875 | 44.58 |
| <input type="radio"/> | SYSAUX | PERMANENT | LOCAL | AUTO | ONLINE | 220.000 | 211.313 | 96.05 |
| <input type="radio"/> | SYSTEM | PERMANENT | LOCAL | MANUAL | ONLINE | 430.000 | 427.313 | 99.38 |
| <input type="radio"/> | TEMP | TEMPORARY | LOCAL | MANUAL | ONLINE | 25.000 | 25.000 | 96.15 |
| <input type="radio"/> | UNDOTBS1 | UNDO | LOCAL | MANUAL | ONLINE | 25.000 | 11.938 | 47.75 |
| <input type="radio"/> | USERS | PERMANENT | LOCAL | AUTO | ONLINE | 5.000 | 2.750 | 55.00 |

Vous ne pouvez pas supprimer un tablespace qui contient des segments actifs. Par exemple, si une table du tablespace est en cours d'utilisation ou si le tablespace contient des données d'undo qui sont nécessaire à rollbacker des transactions non comités, vous ne pouvez pas supprimer le tablespace. Le tablespace peut être online ou offline mais il est préférable qu'il soit offline avant d'être supprimé.

3.2.2.7. Information sur un tablespace

Vous pouvez obtenir des informations sur les tablespaces ou sur les fichiers de données en interrogeant les vues suivantes :

- Information sur les tablespaces :
 - **DBA_TABLESPACES**
 - **V\$TABLESPACES**
- Information sur les fichiers de données :
 - **DBA_DATA_FILES**
 - **V\$DATAFILE**
- Information sur les fichiers temporaires :
 - **DBA_TEMP_FILES**
 - **V\$TEMPFILE**

Cliquez sur View pour voir des informations sur le tablespace sélectionné. Sur cette nouvelle page, vous pouvez cliquer sur Edit pour modifier le tablespace.

Database: orcl:oracle.com > Tablespaces > View Tablespace: EXAMPLE Logged in As SYS

View Tablespace: EXAMPLE

[Edit](#) [Return](#)

Name **EXAMPLE**
Bigfile tablespace **No**
Status **ReadWrite**
Type **Permanent**
Extent Management **local**

Storage

Allocation Type **Automatic**
Segment Space Management **Automatic**
Enable logging **No**
Block Size (B) **8192**

Datafiles

| Name | Directory | Size (MB) | Used (MB) |
|---------------|-------------------------------|-----------|-----------|
| example01.dbf | /u01/app/oracle/oradata/orcl/ | 150.00 | 80.25 |

Thresholds

Use Default Thresholds
Warning (% used) **85**
Critical (% used)

[Edit](#) [View](#) [Delete](#) Actions

| Select | Name | Type | Extent Management | Segment Management | Status | Size (MB) | Used (MB) |
|-------------------------------------|---------|-----------|-------------------|--------------------|--------|-----------|-----------|
| <input checked="" type="checkbox"/> | EXAMPLE | PERMANENT | LOCAL | AUTO | ONLINE | 150.000 | 80.250 |

4. Module 4 : Gestion des utilisateurs et gestion des objets

4.1. Création et gestion des comptes utilisateurs

4.1.1. Comptes utilisateur de la base de données

Pour accéder à la base de données, un utilisateur doit avoir un compte utilisateur valide dans la base de données et il doit s'authentifier.

Dans certains cas, chaque utilisateur a son propre compte de base de données. Dans d'autres, plusieurs utilisateurs partagent un compte commun pour se connecter à la base de données. Indépendamment de la manière dont vous organisez votre base de données, chaque compte utilisateur aura :

- Un nom unique : le nom ne peut pas excéder 30 caractères, ne peut pas contenir de caractère spécial et doit commencer par une lettre.
- Une méthode d'authentification : la méthode d'authentification la plus utilisée est le mot de passe, cependant Oracle 10g supporte plusieurs autres méthodes comme la biométrie ou l'utilisation de certificats.
- Un tablespace par défaut : il s'agit d'un endroit où seront stockés les objets de l'utilisateur si il ne précise pas explicitement où il veut stocker ses objets. Cependant, le fait qu'un utilisateur ait un tablespace par défaut pour stocker ses objets n'implique pas forcément qu'il ait les *privileges* suffisants ou un *quota* d'espace sur celui-ci pour créer des objets. Ces deux conditions sont accordées séparément.
- Un tablespace temporaire : il s'agit d'un endroit où un utilisateur peut créer des objets temporaires comme les tris et les tables temporaires.
- Un profil utilisateur : Il s'agit d'un ensemble de restrictions sur les ressources et les mots de passe assignés à un utilisateur.

4.1.2. Création d'un utilisateur

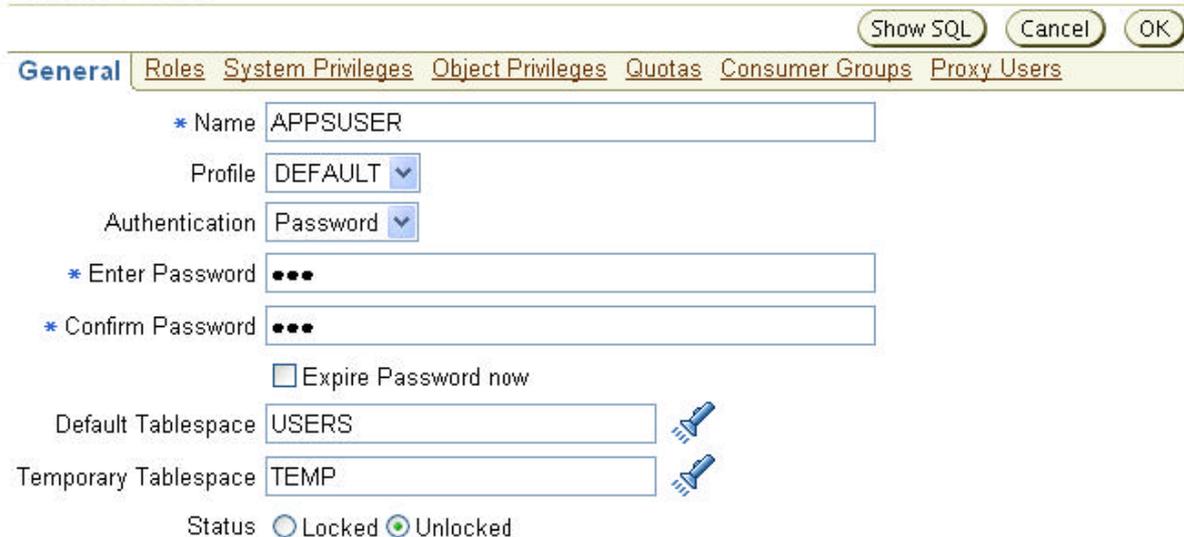
Grâce à Oracle Enterprise Manager, vous pouvez gérer la liste des utilisateurs de la base de données pouvant accéder celle-ci en éditant les propriétés des utilisateurs. Vous pouvez donc créer, supprimer ou modifier les paramètres d'un utilisateur.

La page de gestion des utilisateurs comporte plusieurs onglets. Vous pouvez créer ou éditer les paramètres de sécurité pour un utilisateur à l'aide des différents onglets disponibles. Cliquer sur l'aide pour avoir obtenu d'avantage de précisions concernant une page spécifique.

Pour créer un utilisateur :

1. Accéder à la page d'accueil d'Enterprise Manager.
2. Sur la page d'administration, cliquer sur Users. La page affichée contient une liste de tous les utilisateurs de la base.
3. Cliquer sur Create pour ajouter un nouvel utilisateur.

Create User



The screenshot shows the 'Create User' dialog box in Oracle Enterprise Manager. The 'General' tab is active, and the following fields are visible:

- Name:** APPSUSER
- Profile:** DEFAULT
- Authentication:** Password
- * Enter Password:** (masked with three dots)
- * Confirm Password:** (masked with three dots)
- Expire Password now:**
- Default Tablespace:** USERS
- Temporary Tablespace:** TEMP
- Status:** Locked Unlocked

At the top right of the dialog, there are buttons for 'Show SQL', 'Cancel', and 'OK'. Below the 'General' tab, there are links for 'Roles', 'System Privileges', 'Object Privileges', 'Quotas', 'Consumer Groups', and 'Proxy Users'.

4.2. Contrôle des ressources pour les utilisateurs

4.2.1. Profils et utilisateurs

Les profils permettent de limiter la consommation des ressources au niveau de l'instance et l'utilisation de la base de données. Par ailleurs, ils établissent des restrictions sur les mots de passes des utilisateurs (longueur du mot de passe, durée de vie avant expiration...). Un utilisateur à obligatoirement un et un seul profil qui lui a été assigné.

Le profil par défaut sert de base à tous autres profils. Dans l'exemple suivant, les valeurs pour les différents paramètres d'un profil peuvent être soit implicitement spécifiées (comme pour CPU/Session), illimitées (comme pour CPU/Call) ou alors configurées avec les paramètres du profil par défaut (comme pour Connect Time).

Les profils ne peuvent pas imposer de limitation au niveau des ressources pour un utilisateur tant que le paramètre d'initialisation **RESSOURCES_LIMIT** n'est pas à TRUE. Ce paramètre est par défaut initialisé à FALSE, donc les limitations de ressources sont ignorées. Les profils permettent à l'administrateur de contrôler les ressources systèmes suivantes :

- CPU : les ressources CPU peuvent être limitées « per-session » (par session) ou « per-call » (par appel). Une limitation de CPU/session de 1000 signifie que si n'importe quelle session configurée avec ce profil consomme plus de 10 secondes de temps processeur (ce paramètre est exprimé en centième de seconde), alors un message d'erreur est retourné et la session fermée :
 - `ORA-02392: exceeded session limit on CPU usage, you are being logged off.`
 - Une limitation par appel effectue le même contrôle, mais s'applique pas à l'ensemble de la session, il empêche que n'importe quelle commande de consommer trop de temps CPU. Ce paramètre est limité et si l'utilisateur dépasse cette valeur, la commande échoue et l'utilisateur reçoit le message d'erreur suivant : `ORA-02393: exceeded call limit on CPU usage`
- Réseaux/Mémoire: chaque session de base de données consomme des ressources mémoires systèmes et des ressources réseaux (si celle-ci n'est pas locale) :
 - Connect Time : Temps en minutes pendant lequel les utilisateurs peuvent rester connectés avant d'être automatiquement déconnectés.
 - Idle Time : Temps en minute avant qu'une session utilisateur soit automatiquement fermée par le serveur s'il n'y a pas d'échanges de données. Le temps d'inactivité est calculé seulement au niveau du processus serveur. Il ne prend pas en compte l'activité d'une application. La limite IDLE_TIME n'est tient pas compte du temps d'exécution de longues requêtes ou autres opérations.
 - Concurrent session : Nombre de sessions pouvant être ouvertes simultanément en utilisant le même compte utilisateur.
 - Private SGA : Limite l'espace utilisé dans la SGA pour trier, ou encore réorganiser des bitmaps. Cette restriction est effective seulement si la session utilise un serveur partagé.
- Disk I/O : Limite le volume de données qu'un utilisateur peut lire soit pour la session, soit pour un appel (« per session » et « per call »). Les paramètres Reads/Session et Reads/Call mettent en place des restrictions sur le nombre total d'accès, que ce soit pour la mémoire ou pour le disque. Cela permet de s'assurer qu'aucune requête n'utilise de manière intensive les entrées/sorties au niveau de la mémoire et des disques. Les profils peuvent aussi être configurés avec les limites « composées ». Les limites composées se basent sur la combinaison de CPU/session, reads/session, connect time, et de la SGA privée.

Create Profile

General Password

* Name

Details

| | | |
|------------------------|--|---|
| CPU/Session (Sec./100) | <input style="width: 90%;" type="text" value="1000"/> |  |
| CPU/Call (Sec./100) | <input style="width: 90%;" type="text" value="UNLIMITED"/> |  |
| Connect Time (Minutes) | <input style="width: 90%;" type="text" value="DEFAULT"/> |  |
| Idle Time (Minutes) | <input style="width: 90%;" type="text" value="60"/> |  |

Database Services

| | | |
|---------------------------------|--|---|
| Concurrent Sessions (Per User) | <input style="width: 90%;" type="text" value="DEFAULT"/> |  |
| Reads/Session (Blocks) | <input style="width: 90%;" type="text" value="DEFAULT"/> |  |
| Reads/Call (Blocks) | <input style="width: 90%;" type="text" value="DEFAULT"/> |  |
| Private SGA (KBytes) | <input style="width: 90%;" type="text" value="DEFAULT"/> |  |
| Composite Limit (Service Units) | <input style="width: 90%;" type="text" value="DEFAULT"/> |  |

4.2.2. Authentification des utilisateurs

L'authentification consiste à vérifier l'identité de quelqu'un (un utilisateur, un dispositif ou une autre entité) qui veut utiliser des données, des ressources, ou des applications. Le fait de valider l'identité établit une relation de confiance pour d'autres interactions. L'authentification permet aussi l'imputabilité en rendant possible la liaison entre les accès et les actions spécifique liés à certaines identités. Après l'authentification, les processus d'autorisation vont autoriser ou limiter les niveaux d'accès et action à cette entité. Quand vous créer un utilisateur, vous devez choisir le mode d'authentification à utiliser, qui pourra être modifié par la suite.

Mot de passe : C'est l'authentification par la base de données. Chaque utilisateur est associé à un mot de passe qui doit être fourni quand l'utilisateur veut d'établir une connexion. Lors de la configuration du mot de passe, vous pouvez spécifier que celui-ci expire à la première connexion, ce qui oblige l'utilisateur à le changer. Si vous projetez d'activer l'option `EXPIRE`, il faut s'assurer que les utilisateurs ont la possibilité de le changer. Certaines applications n'ont pas cette fonctionnalité.

Externe : C'est l'authentification par le système d'exploitation, l'utilisateur se connecte à Oracle sans spécifier son login et son mot de passe. Avec cette authentification, votre base de données dépend du système d'exploitation ou d'une authentification par le réseau pour limiter l'accès à la base de données. Un mot de passe de base de données n'est pas utilisé pour ce type d'authentification. Si votre système d'exploitation ou les services réseaux le permettent, vous pouvez donc authentifier des utilisateurs. Si vous procédez ainsi, renseignez le paramètre `OS_AUTHENT_PREFIX` et utilisez ce préfixe dans les noms d'utilisateurs d'Oracle. Le paramètre `OS_AUTHENT_PREFIX` définit un préfixe qu'Oracle ajoute au début du nom du compte du système d'exploitation de chaque utilisateur. La valeur par défaut de ce paramètre est `OP$`.

Par exemple, le paramètre `OS_AUTHENT_PREFIX` peut être initialisé comme suit :

```
OS_AUTHENT_PREFIX=OP$
```

Si un utilisateur du système d'exploitation porte le nom « tsmith » et se veut se connecter à la base de données en utilisant l'authentification par le système d'exploitation, Oracle vérifie qu'il existe un utilisateur correspondant dans la base de données du nom de : OPS\$tsmith. Si c'est le cas, l'utilisateur est autorisé à se connecter. Toutes références à un utilisateur authentifié par le système d'exploitation doit être préfixé, comme OPS\$tsmith.

Globale : C'est l'authentification forte par l'intermédiaire des options d'Oracle Advanced Security. L'authentification globale permet à des utilisateurs d'être identifiés en utilisant la biométrie, les certificats x509, ou encore Oracle Internet Directory.

* Name

Profile

Authentication

* Enter Password

* Confirm Password

Expire Password now

* Default Tablespace

Temporary Tablespace

Status Locked Unlocked

4.2.3. Tablespace par défaut, tablespace temporaire et verrouillage

Le tablespace par défaut est le tablespace dans lequel les objets seront créés si aucun tablespace n'est spécifié pour la création de l'objet. Si vous ne choisissez pas de tablespace par défaut, alors le tablespace permanent par défaut défini au niveau de la base de données est utilisé.

Le tablespace temporaire est le tablespace qui permettra d'effectuer des opérations de tris (clause ORDER BY, création d'un index...). Si vous n'en spécifiez pas, alors le tablespace temporaire par défaut défini au niveau de la base est utilisé.

Quand vous créez un utilisateur, vous pouvez verrouiller ou déverrouiller son compte. Si le compte d'un utilisateur est verrouillé, cela signifie que personne ne peut ouvrir une session en utilisant ce compte. Le compte d'un nouvel utilisateur est par défaut déverrouillé.

Remarque : Il est possible de définir un tablespace par défaut et un temporaire au niveau de la base de données. Si vous les définissez au niveau de la base de données, alors n'importe quel utilisateur pour lequel il n'est pas explicitement spécifié de tablespace par défaut et temporaire, se voit attribuer ces tablespaces par défauts définis au niveau de la base de données.

Default Tablespace

Temporary Tablespace

Status Locked Unlocked

4.2.4. Utilisateurs et schéma de la base de données

Un schéma est une collection nommée d'objet dont l'utilisateur est propriétaire. Un schéma porte le même nom que l'utilisateur qui en est propriétaire. Les objets appartenant au schéma sont les tables, les vues, les indexes, ou encore le code stocké d'une application Java ou PL/SQL. Il n'y a aucun lien entre un tablespace et un schéma.

Les objets d'un même schéma peuvent être stockés dans différents tablespaces, et un tablespace peut stocker les objets de différents schémas. Quand un utilisateur est créé, un schéma du même nom est créé pour cet utilisateur. Un utilisateur ne peut être associé qu'à un seul schéma. Par ailleurs, les *schémas* et les *noms d'utilisateurs* sont souvent utilisés de manière interchangeable.

Les objets de schéma qui occupent l'espace dans la base de données (telle que des tables et des index) sont créés dans le tablespace par défaut de l'utilisateur si ils ne sont pas spécifiés autre part lors de leur création. Le propriétaire du schéma a tous les droits sur les objets de son schéma et il peut donner des privilèges à d'autres utilisateurs pour utiliser ses objets.

Certains comptes utilisateurs sont créés spécifiquement pour fournir un schéma spécifique et ne sont pas prévus pour permettre aux utilisateurs d'ouvrir une session en les utilisant. Ces comptes doivent être verrouillés, ce qui signifie que personne ne pourra se connecter en les utilisant. Pour verrouiller un compte, sélectionnez Locked dans la page de gestion des utilisateurs.

Objets:

- Tables
- Triggers
- Indexe
- Vues
- Séquences
- Programmes stockés
- Synonymes
- Types de données définies par les utilisateurs
- Liens de base de données

4.2.5. Assignment de quota à un utilisateur

Un quota est une l'allocation d'espace dans un tablespace donné. Par défaut, un utilisateur n'a aucun quota sur aucun tablespace. Vous avez trois options pour fournir un quota d'utilisateur sur un tablespace :

- Illimité : cela signifie que l'utilisateur peut utiliser toute la place disponible sur le tablespace.
- Une valeur : c'est un nombre en Ko ou Mo. C'est la valeur du quota que l'utilisateur pourra utiliser sur le tablespace. Cette valeur peut être plus grande ou plus petite que l'espace actuellement disponible.
- Le privilège système **UNLIMITED TABLESPACE** : Ce privilège système outrepassé tous les différents quotas sur les tablespaces et donne à l'utilisateur un quota illimité sur tous les tablespaces, y compris **SYSTEM** et **SYSAUX**. Ce privilège doit être donné avec précaution.

D'une manière générale, vous ne devez pas donner de quota aux utilisateurs sur le tablespace **SYSTEM** ou **SYSAUX**. Normalement, seul les utilisateurs **SYS** et **SYSTEM** sont capables de créer des objets dans les tablespaces **SYSTEM** et **SYSAUX**. Les utilisateurs n'ont pas besoin de quota sur le tablespace temporaire et sur les tablespaces undo.

Edit User: HR

Show SQL Revert Apply

General Roles System Privileges Object Privileges **Quotas** Consumer Groups Proxy Users

| Tablespace | Quota | Value | Unit |
|-----------------|-------------|-------|----------|
| EXAMPLE | Value ▾ | 250 | MBytes ▾ |
| SYSAUX | None ▾ | 0 | MBytes ▾ |
| SYSTEM | None ▾ | 0 | MBytes ▾ |
| TEMP | None ▾ | 0 | MBytes ▾ |
| UNDOTBS1 | None ▾ | 0 | MBytes ▾ |
| USERS (Default) | Unlimited ▾ | 0 | MBytes ▾ |

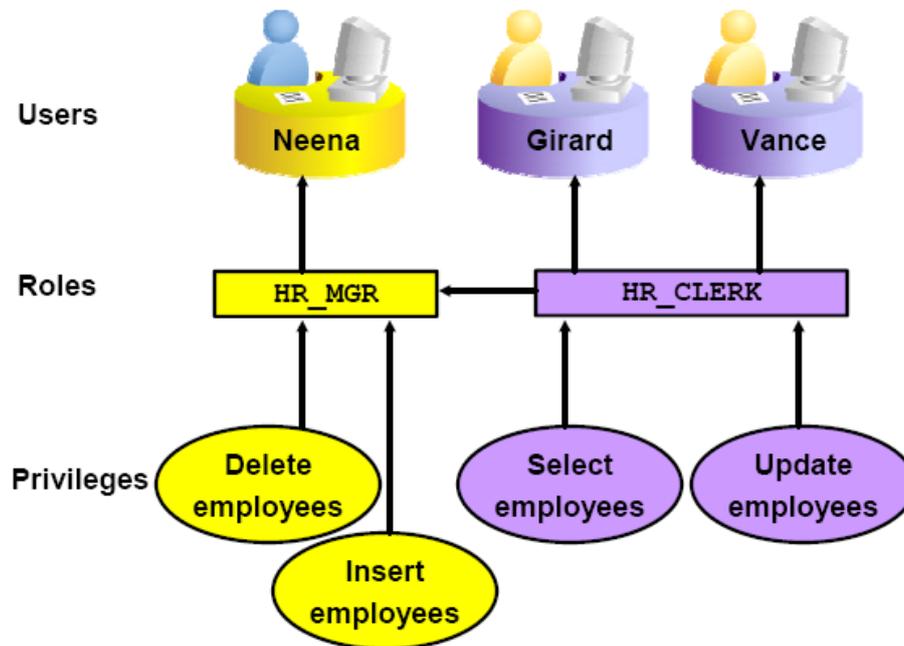
4.3. Créer et gérer des rôles

Pour la plupart des systèmes, il faut passer beaucoup de temps pour accorder les privilèges nécessaires à chaque utilisateur individuellement, augmentant de ce fait le risque d'erreur. Oracle fournit une gestion des privilèges contrôlée et facile grâce aux rôles. Les rôles sont des groupes de privilèges qui sont accordés aux utilisateurs ou à d'autres rôles. Ils permettent de faciliter l'administration des privilèges dans la base de données et par conséquent d'améliorer la sécurité.

Caractéristiques des rôles :

- Les privilèges sont accordés et révoqués des rôles comme si le rôle était un utilisateur.
- Les rôles peuvent être accordés ou révoqués à un utilisateur ou à un autre rôle comme si ils étaient des privilèges système.
- Un rôle peut contenir des privilèges systèmes et objets.
- Un rôle peut être activé ou désactivé pour chaque utilisateur à qui le rôle a été attribué.
- Un rôle peut nécessiter un mot de passe pour être activé.
- Les rôles n'appartiennent à personne et ne sont dans aucun schéma.

Dans l'exemple ci-dessous, le rôle **HR_CLERK** contient les privilèges **SELECT** et **UPDATE** sur la table employees. Le rôle **HR_MGR** contient les privilèges **DELETE** et **INSERT** sur la table employees et le rôle **HR_CLERK**. Le manager qui a le rôle **HR_MGR** peut donc faire des **SELECT**, **DELETE**, **INSERT** et **UPDATE** sur la table employees.



4.3.1. Avantage des rôles

Une gestion plus facile des privilèges : Utiliser les rôles pour simplifier la gestion des privilèges. Plutôt que d'accorder le même ensemble de privilèges à plusieurs utilisateurs, vous pouvez accorder les privilèges à un rôle, et puis accordez ce rôle à chaque utilisateur

Une gestion des privilèges dynamique : Si un privilège associé à un rôle est modifié, tous les utilisateurs qui ont ce rôle auront le privilège modifié automatiquement et immédiatement.

Disponibilité sélective des privilèges : Des rôles peuvent être activés et désactivés pour ajouter ou enlever des privilèges temporairement. L'activation d'un rôle peut être utilisée pour vérifier que ce rôle a bien été attribué à un utilisateur.

Attribué par le système d'exploitation : Des commandes du système d'exploitation ou des utilitaires peuvent être utilisés pour assigner un rôle à un utilisateur.

4.3.2. Rôles prédéfinis

Il y a plusieurs rôles qui sont définis automatiquement par Oracle quand vous lancez les scripts de création de la base de données. Le rôle **CONNECT** est accordé automatiquement à un utilisateur qui est créé avec Enterprise Manager. Le rôle **SELECT_CATALOG_ROLE** est accordé pour accéder aux vues du dictionnaire des données ainsi qu'aux packages (ce rôle est dévalué en faveur du privilège système **SELECT_ANY_DICTIONARY**). Le rôle **DBA** inclut presque tous les privilèges et ne doit pas être donné à un utilisateur non administrateur.

D'autres rôles qui vous permettant d'administrer des fonctions spéciales sont aussi créés. Par exemple le rôle **XDBADMIN** contient les privilèges pour administrer la base de données avec du XML si cette fonctionnalité est installée. Le rôle **AQ_ADMINISTRATOR_ROLE** fournit des privilèges pour gérer des requêtes avancées. Le rôle **HS_ADMIN_ROLE** inclut des privilèges nécessaires pour administrer des services. Vous ne devez pas changer les privilèges accordés à ces rôles fonctionnels sans un appui d'Oracle parce que vous pourriez par erreur désactiver des fonctionnalités nécessaires.

| RÔLE | PRIVILEGE |
|-----------------|---|
| CONNECT | CREATE SESSION, CREATE TABLE, CREATE VIEW, CREATE SYNONYM, CREATE SEQUENCE, CREATE DATABASE LINK, CREATE CLUSTER, ALTER SESSION |
| RESOURCE | CREATE TABLE, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TRIGGER, CREATE TYPE, CREATE CLUSTER, CREATE INDEXTYPE, CREATE OPERATOR |
| SCHEDULER_ADMIN | CREATE ANY JOB, EXECUTE ANY CLASS, EXECUTE ANY PROGRAM, MANAGE SCHEDULER |
| DBA | La plupart des privilèges système et plusieurs autres rôles Ne doit pas être accordé aux utilisateurs. |
| SELECT_CATALOG | Ne contient aucun privilège système mais plus de 1600 privilèges objet sur le dictionnaire des données. |

4.3.3. Sécurité des rôles

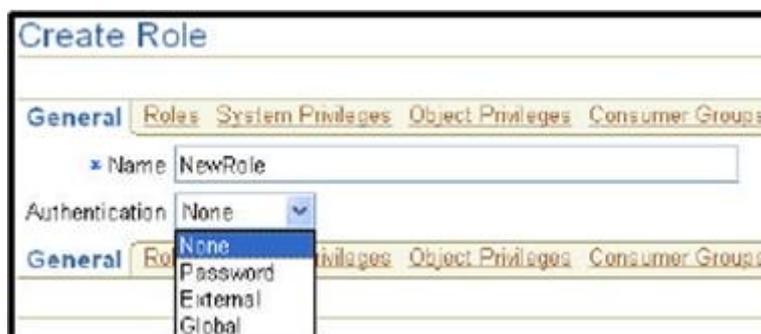
Les rôles sont normalement activés par défaut, ce qui signifie que si un rôle est accordé à un utilisateur, l'utilisateur peut exécuter les privilèges donnés à ce rôle. Il est possible de :

- Désactiver le rôle par défaut : quand un rôle est accordé à un utilisateur, décochez la case DEFAULT. Maintenant l'utilisateur doit explicitement activer le rôle avant que les privilèges du rôle puissent être exercés.
- Avoir un rôle qui requière une authentification supplémentaire. L'authentification par défaut pour un rôle est configurée à NONE mais il est possible que le rôle requière une authentification supplémentaire avant d'être utilisé.
- Créer des rôles d'application sécurisés qui peuvent être activés seulement en exécutant un bloc PL/SQL. Le bloc PL/SQL peut vérifier certaines choses telles que l'adresse réseau de l'utilisateur, quel programme l'utilisateur a lancé ou toute autre chose qui est nécessaire pour sécuriser correctement un groupe de permissions.

Pour activer un rôle qui n'est pas activé par défaut:

```
SET ROLE vacationdba;
```

Un rôle peut être protégé par une authentification:



Un rôle peut aussi être sécurisé par programmation:

```
CREATE ROLE secure_application_role
IDENTIFIED USING <security_procedure_name>;
```

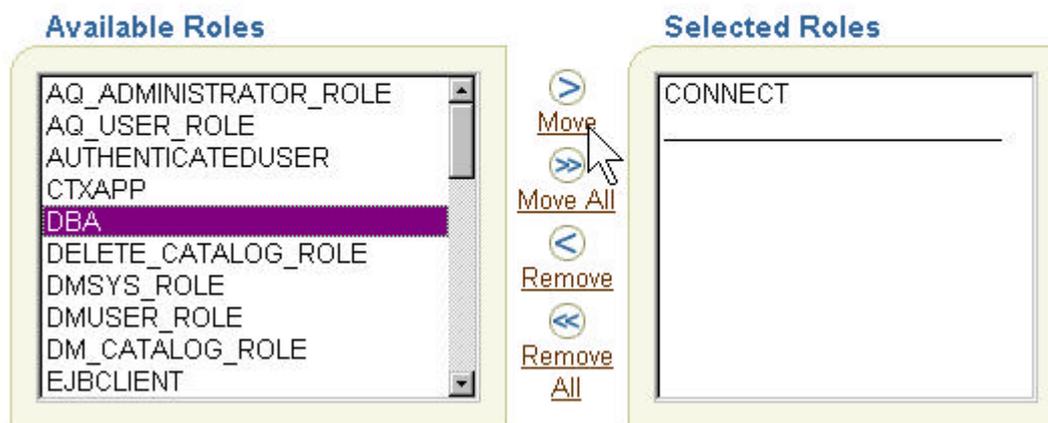
4.3.4. Assignment de rôles aux utilisateurs

Un rôle est un ensemble de privilèges qui peut être accordé à un utilisateur ou à un autre rôle. Vous pouvez utiliser les rôles pour gérer les privilèges de la base de données. Vous pouvez ajouter un privilège à un rôle et accorder ce rôle à un utilisateur. Un utilisateur peut alors activer le rôle, et bénéficier des privilèges de ce rôle. Un rôle contient tous les privilèges qui lui ont été accordés et tous les privilèges des autres rôles qui lui ont été attribués.

Par défaut, Enterprise Manager accorde automatiquement le rôle **CONNECT** à un nouvel utilisateur. Ceci permet aux utilisateurs de se connecter à la base de données et de créer des objets dans leur propre schéma.

Database: orcl > Users > Edit User: RIC

Modify Roles



4.4. Privilèges GRANT et REVOKE

Un privilège est un droit qui permet d'exécuter un ordre SQL particulier ou d'accéder à l'objet d'un autre utilisateur. Oracle permet le contrôle pointu de ce que les utilisateurs peuvent ou ne peuvent pas faire dans la base de données. Les privilèges sont divisés en deux catégories :

- Privilèges systèmes : Chaque privilège système permet à un utilisateur d'exécuter une opération particulière sur la base de données, par exemple, le privilège **CREATE TABLESPACE** pour créer un tablespace. Les privilèges systèmes peuvent être accordés à un administrateur. Il y a plus de 100 privilèges systèmes.
- Privilèges objets : ce sont des privilèges qui permettent à l'utilisateur d'effectuer une action particulière sur un objet, comme une table, une vue, une séquences, une procédure, une fonction ou un package. Sans permission particulière, les utilisateurs ne peuvent avoir accès qu'à leurs objets. Les privilèges objets peuvent être accordés à un utilisateur, par un administrateur, ou pas quelqu'un à qui l'on a explicitement donné la permission d'accorder des droits sur cet objet.

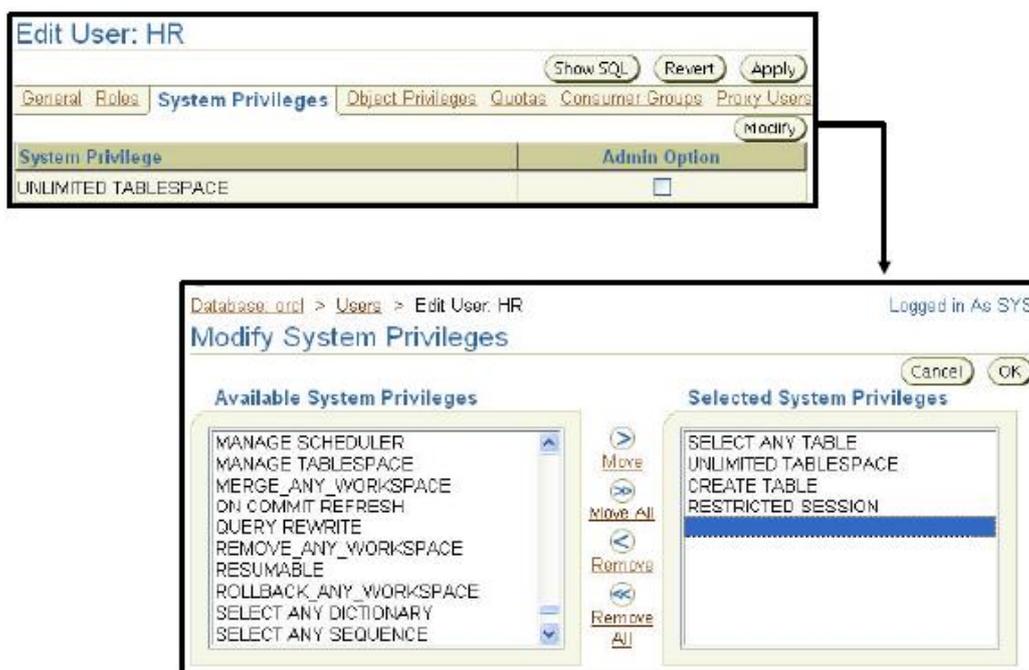
4.4.1. Privilèges systèmes

Pour donner des privilèges systèmes, cliquer sur le lien Systems Privileges, sélectionnez le privilège approprié à partir de la liste des privilèges disponibles et déplacez le dans la liste des privilèges choisis en cliquant sur la flèche.

Accorder un privilège avec la clause **ANY** signifie que le privilège est accordé sur d'autres schémas. Par exemple, le privilège CREATE TABLE ne vous permet de créer des tables que dans votre schéma. Le privilège SELECT ANY TABLE vous donne le droit d'exécuter des ordres SELECT sur les tables des autres schémas. La check box Admin Option permet à cet utilisateur d'administrer le privilège et d'accorder le privilège système à d'autres utilisateurs.

Prenez bien en compte les conditions de sécurité avant d'accorder des privilèges systèmes. Certains privilèges systèmes ne sont généralement accordés qu'aux administrateurs :

- RESTRICTED SESSION : ce privilège donne droit aux utilisateurs de se loguer même quand la base de données est ouverte en mode restreint.
- SYSDBA et SYSOPER : ces privilèges permettent aux utilisateurs d'éteindre, de démarrer, de restaurer et opérer d'autres tâches administratives sur la base de données.
- DROP ANY *objet* : ce privilège permet d'effacer des objets même si ils ne nous appartiennent pas.
- CREATE, MANAGE, DROP, ALTER TABLESPACE : un non administrateur ne devrait pas avoir de contrôle sur les tablespaces.
- CREATE ANY DIRECTORY : Oracle permet à des développeurs d'appeler du code externe (exemple : une librairie C) depuis du PL/SQL. C'est une mesure de sécurité, les répertoires du système d'exploitation où le code est stocké doivent être reliés à un répertoire virtuel d'Oracle. Avec ce privilège, un utilisateur peut potentiellement appeler du code non sécurisé.
- EXEMPT ACCESS POLICY : Ce privilège permet à un utilisateur de dévier des fonctions de sécurité placées sur des tables ou des vues.
- GRANT ANY OBJECT PRIVILEGE : Ce privilège permet à des utilisateurs d'accorder des privilèges objets sur des objets qu'ils ne possèdent pas.
- ALTER DATABASE et ALTER SYSTEM : un non administrateur ne doit pas avoir le moyen d'effectuer des changements sur la base ou sur l'instance.

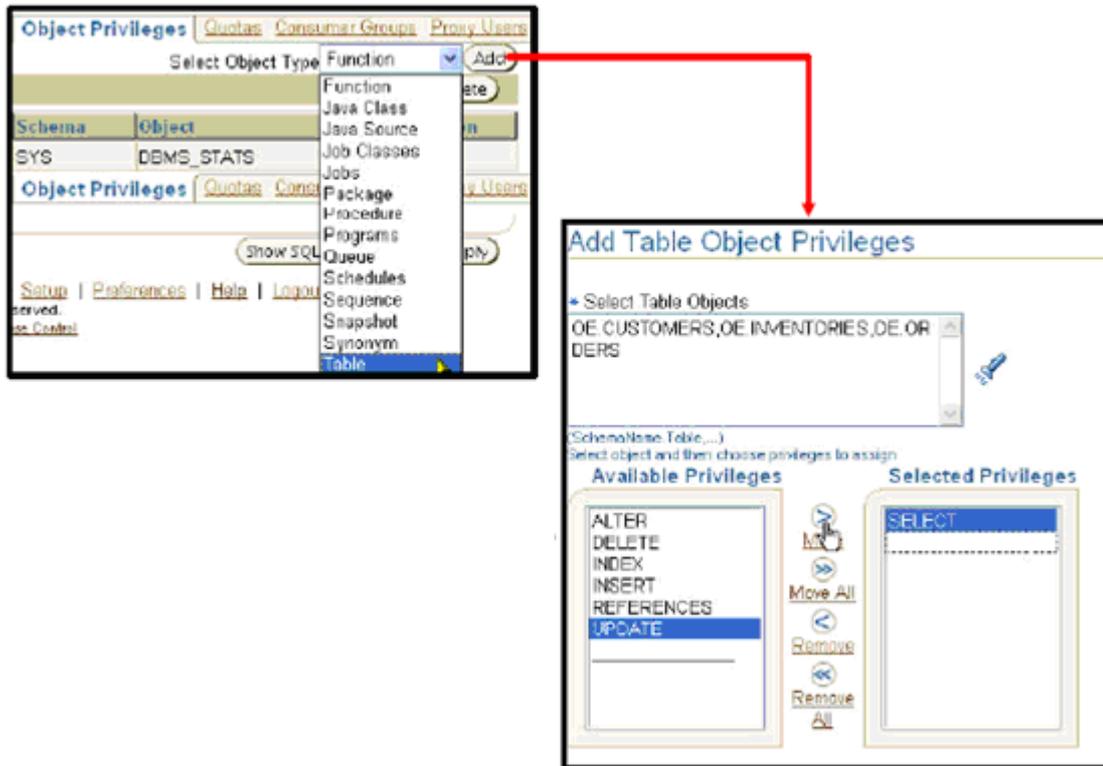


4.4.2. Privilèges objets

Pour accorder un privilège objet, cliquez sur le lien Object Privileges et sélectionnez le type d'objet sur lequel vous voulez accorder un privilège puis cliquez sur le bouton Add.

Après, sélectionnez le privilège approprié depuis la liste des privilèges disponible et cliquez sur le bouton Move.

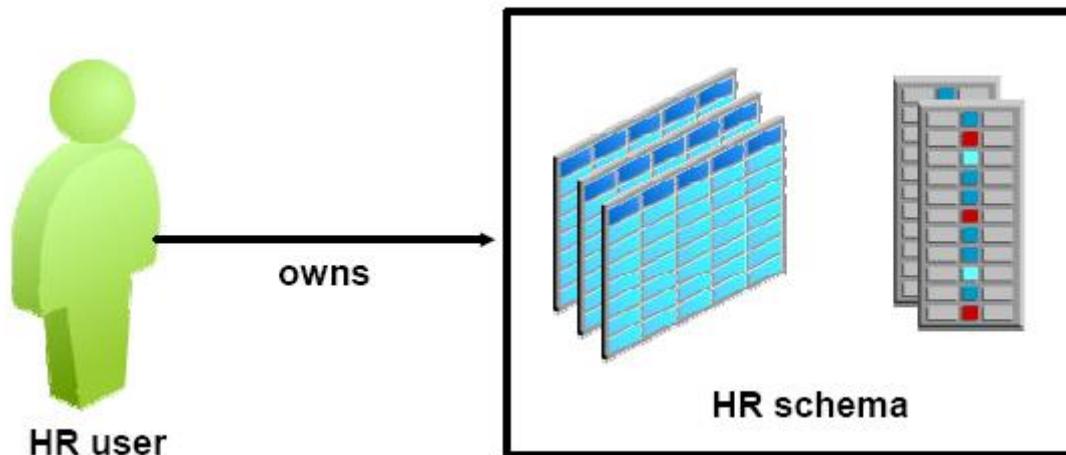
Cocher la check box Grant à partir de la liste de privilèges d'objet pour permettre à l'utilisateur d'accorder à d'autres utilisateurs le même accès.



4.5. Gestion des objets du schéma

4.5.1. Qu'est-ce qu'un schéma ?

Un schéma est une collection nommée d'objet. Un schéma porte le même nom que l'utilisateur du schéma. Les objets de schéma sont des structures logiques qui se rapportent directement aux données de la base de données. Les objets de schéma incluent des structures tels que des tables, des vues, des indexes.



Schémas automatiquement créés pendant le processus de création de la base de données:

- SYS
- SYSTEM
- Des schémas d'exemple

Quand vous créer une base de données, un certain nombre de schémas sont créé. Deux d'entre eux sont particulièrement importants :

Le schéma SYS : Toutes les tables de base et les vues qui constituent le dictionnaire des données de la base de données sont créées dans le schéma SYS. Le dictionnaire des données est une collection de tables qui décrit la base de données Oracle. Le dictionnaire de données est créé dans le tablespace SYSTEM lors de la création de la base et il est mis à jour par le serveur de la base de données Oracle quand un ordre DDL est exécuté. Le dictionnaire de données contient des informations sur les utilisateurs, les objets des schémas et les structures de stockage. Vous pouvez utiliser le dictionnaire de données, comme référence, pour obtenir des informations sur la base de données. Les objets du schéma SYS ne doivent pas être modifiés par un utilisateur ou par un DBA, et personne n'est censé créer de tables dans le schéma de l'utilisateur SYS.

- **Le schéma SYSTEM :** il contient les tables additionnelles et les vues qui stockent des informations administratives ainsi que des tables internes et des vues utilisées par diverses options et outils d'Oracle. Vous ne devez pas créer d'objet dans le schéma SYSTEM.

Pendant l'installation de la base de données Oracle, des schémas d'exemples sont installés automatiquement avec le noyau de la base de données. Les schémas d'exemple ont pour but de servir de base pour les exemples dans les documentations d'Oracle ou autres :

- HR : le schéma « Human Resources » est un schéma simple pour se familiariser avec les notions de bases.
- OE : le schéma « Order Entry » traite des sujets de complexité intermédiaire. Une multitude de types de données est disponible dans le schéma d'OE. Le sous schéma OC (Online Catalog) est une collection d'objets relationnels de base de données établis à l'intérieur du schéma d'OE.
- PM : Le schéma « Product Media » est dédié aux types de données multimédia.
- QS : Le schéma « Queued Shipping » contient un ensemble de schémas qui sont utilisés pour démontrer les possibilités avancées d'Oracle.
- SH : Le schéma « Sales History » est conçu pour permettre des démonstrations avec de plus grands volumes de données. Une extension de ce schéma fournit un support pour le traitement analytique avancé.

Accès aux objets du schéma

Vous pouvez rapidement accéder à différents types d'objets de schéma à partir de la page Administration.

Après avoir cliqué sur un des liens, dans la zone de recherche, vous pouvez mettre un nom de schéma ou un nom d'objet pour rechercher un objet spécifique. En outre, vous pouvez rechercher d'autres types d'objets dans la zone de recherche en choisissant le type d'objet à partir du menu déroulant. Le menu déroulant inclut les types d'objets qui ne sont pas affichés comme lien sur la page Administration.



The screenshot shows the Oracle Administration page for the database 'orcl.us.oracle.com'. The 'Administration' tab is selected. Under the 'Instance' section, there are links for Memory Parameters, Undo Management, All Initialization Parameters, and Controlfiles. Under the 'Storage' section, there are links for Tablespace, Datafiles, Rollback Segments, Redo Log Groups, Archive Logs, and Temporary Tablespace Groups. Under the 'Schema' section, there are links for Tables, Indexes, Views, Synonyms, Sequences, Database Links, Packages, Package Bodies, Procedures, Functions, Triggers, Java Sources, and Java Classes. A pink callout box with the text 'Click a link to access the schema objects.' has an arrow pointing to the 'Tables' link under the 'Schema' section.

| Instance | Storage | Schema |
|---|---|--------------------------------|
| Memory Parameters | Controlfiles | Tables |
| Undo Management | Tablespaces | Indexes |
| All Initialization Parameters | Datafiles | Views |
| Parameters | Rollback Segments | Synonyms |
| | Redo Log Groups | Sequences |
| | Archive Logs | Database Links |
| | Temporary Tablespace Groups | Packages |
| | | Package Bodies |
| | | Procedures |
| | | Functions |
| | | Triggers |
| | | Java Sources |
| | | Java Classes |

4.5.2. Nomenclature des noms des objets de la base de données

Quand vous nommez un objet de la base de données vous avez la possibilité de mettre les noms entre guillemets ("). Toutefois ceci n'est pas recommandé, parce que si vous appelez un objet de cette façon vous devez toujours vous référer à celui-ci en utilisant son nom entre guillemets. Par exemple, si vous nommez une table "Local Temp":

```
SQL> select * from "Local Temp";
TEMP_DATE LO_TEMP HI_TEMP
-----
01-DEC-03 30 41
If you mistype the case you will get:
SQL> select * from "local temp";
select * from "local temp"
*
ERROR at line 1:
ORA-00942: table or view does not exist
```

Les noms qui ne sont pas entre guillemets sont stockés en majuscule et ne sont pas sensibles à la casse. Quand une requête SQL est traitée, les noms qui ne sont pas entre guillemets sont convertis en majuscule.

Un identifiant qui n'est pas entre guillemets peut comprendre seulement des caractères alphanumériques du jeu de caractères de votre base de données ainsi que les caractères « _ », « \$ », « # ». Les liens de base de données peuvent également contenir des « . », « @ ». Il est dangereux d'utiliser les caractères « \$ » et « # » avec des noms qui ne sont pas entre guillemets. Les identifiants qui sont entre guillemets peuvent contenir tous les caractères et les ponctuations et même les espaces.

4.5.3. Les namespaces

La base de données utilise des namespaces comme référence aux objets et schémas. Quand vous vous référez à un objet, Oracle va localiser l'objet dans le namespace approprié. Après avoir localisé l'objet, Oracle effectue l'opération de la requête sur l'objet. Si l'objet n'est pas trouvé dans le bon namespace, Oracle vous retournera une erreur.

Les tables et les vues dans un même schéma ne peuvent pas porter le même nom car ces objets sont dans le même namespace. Cependant, les tables et les indexes sont dans différents namespaces donc une table et un index dans un même schéma peuvent avoir le même nom.

Les objets suivants sont dans le même namespace:

- Tables
- Vues
- Séquences
- synonymes
- Procédures stand-alone
- Fonctions stand-alone stockées
- Package
- Vues matérialisées
- Types définis par l'utilisateur

Les objets suivants ont leur propre namespace:

- Indexe
- Contraintes
- Clusters
- Triggers de la base de données
- Lien de base de données
- Dimensions

4.5.4. Types de données spécifique dans une table

Quand vous créer une table, vous devez spécifier le type de données pour chaque colonne. Quand vous créer une procédure ou une fonction, vous devez spécifier le type de données pour chaque argument. Ces types de données définissent un domaine de valeur que chaque colonne ou argument peut stocker. Voici les types de données fournis par Oracle:

- **CHAR** : chaîne de caractère de taille fixe. La taille maximum est 2000 octets. Par défaut la taille est de 1.
- **VARCHAR2** : chaîne de caractère de taille variable. La taille maximum est de 4000 octets. Vous devez spécifier une taille pour une donnée VARCHAR2.
- **DATE** : stocke une date valide à partir du 1er janvier 4712 AVANT JC au 31 décembre 9999.
- **NUMBER (p,s)** : nombre avec une précision *p* et une échelle *s*. la plage de valeur pour la précision est de 1 à 38, et celle de l'échelle de -84 à 127.
- **FLOAT (p)** : C'est une donnée de type ANSI. Le type de données FLOAT est un nombre à virgule flottante avec une précision binaire *p*.
- **INTEGER** : c'est l'équivalent de NUMBER (p,0)
- **NCHAR (taille)** : le type de données NCHAR est un type de données Unicode. Quand vous créer une table avec une colonne NCHAR, vous allez spécifier le nombre de caractère pour la colonne. Vous définissez le jeu de caractère quand vous créez votre base de données. La taille de la colonne est déterminée par le jeu de caractère. La taille maximum allouée est 2000 octets. Si vous insérez une valeur dont la taille est plus petite que celle de la colonne, alors Oracle comble le reste par des blancs. Vous ne pouvez pas insérer une valeur CHAR dans une colonne NCHAR, ni insérer une valeur NCHAR dans une colonne de type CHAR.
- **NVARCHAR2 (taille)** : le type de données NVARCHAR2 est un type de données Unicode. C'est comme pour le type NCHAR sauf que la taille maximum est de 4000 octets.
- **LONG** : chaîne de caractère variable de maximum 2 Go.
- **LONG RAW** : type de donnée binaire de longueur variable pouvant stocker jusqu'à 2 Go.
- **RAW (size)** : type de donnée binaire de longueur fixe déterminé par le paramètre size (maximum 2000 octets).
- **ROWID** : Chaîne de caractères codée en base 64 représentant l'adresse physique unique d'une ligne dans sa table. Valeur retournée en interrogeant la pseudo-colonne ROWID.
- **UROWID** : Chaîne de caractères codée en base 64 représentant l'adresse logique d'une ligne dans une table organisée en index. la taille optionnelle est la taille de la colonne de type UROWID. La taille maximum et par défaut est 4000 octets
- **BLOB** : Chaîne d'éléments au format binaire dont la capacité de stockage maximum est de 4 giga-octets (9i) et 8 téraoctets (10g).
- **CLOB** : Chaîne composée de caractères mono octet ou multi octets dont la capacité maximum de stockage est de 4 giga-octets (9i) et 8 téraoctets (10g).
- **NCLOB** : Chaîne composée de caractères au format Unicode mono octet ou multi octets dont la capacité maximum de stockage est de 4 giga-octets (9i) et 8 téraoctets (10g).
- **BFILE** : Pointeur vers un fichier stocké en dehors de la base.
- **TIMESTAMP (fractional_second_precision)** : Donnée de type date stockant l'année, le mois, le jour, l'heure, les minutes, les secondes et les fractions de secondes dont le nombre de chiffres est exprimé par le paramètre (fractional_seconds_precision) Ce chiffre peut être compris entre 0 et 9 (par défaut 6).

4.5.5. Création et modification d'une table

4.5.5.1. Création d'une table

Les tables sont des unités de stockage basique. Elles contiennent des données accessibles par les utilisateurs. Chaque table est composée de lignes et de colonnes.

Vous pouvez créer une table avec Entreprise Manager :

1. Cliquez sur Tables dans la partie Schéma de la page Administration. La page Tables apparaît.
2. Si vous connaissez le nom du schéma, entrez son nom dans le champ Schéma. Si vous ne connaissez pas le nom du schéma, cliquez sur l'icône à côté du champ schéma, pour ensuite sélectionner votre schéma.
3. Cliquez sur créer. La table est créée puis la page Create Table: Table Organization apparaît.
4. Acceptez les options par défaut, et cliquez sur Continue. La page Create Table apparaît.
5. Entrez le nom de la table dans le champ Name.
6. Entrez le nom du Tablespace dans le champ Tablespace ou cliquez sur l'icône à côté du champ pour choisir un Tablespace.
7. Dans la section Colonnes, entrez le nom des colonnes et les types de données.
8. Cliquez sur OK. Un message apparaît vous indiquant que la table est créée.

4.5.5.2. Modification d'une table

Vous pouvez modifier la table en utilisant Entreprise Manager. Dans cet exemple, on va ajouter une colonne à la table.

1. Dans la page Tables, sélectionnez la table dans la liste des résultats et cliquez sur Edit.
2. Cliquez sur le bouton Add 5 Table Columns.
3. Entrez le nom de la nouvelle colonne, le type de données et sa taille. Cliquez sur Appliquer.
4. Un message apparaît vous indiquant que votre table a été modifiée avec succès.

Database: orcl.us.oracle.com > Tables > Create Table

Create Table

General Constraints Storage Options Partitions

Specify the table name and schema.

* Name jobs
 Schema shopowner
 Tablespace <Default> Estima
 Organization Standard, Heap Organized

Specify the column names, data types, and lengths.

| Select | Name | Data Type | Size |
|----------------------------------|------------|-----------|------|
| <input checked="" type="radio"/> | job_id | NUMBER | 5 |
| <input type="radio"/> | job_title | VARCHAR2 | 30 |
| <input type="radio"/> | min_salary | NUMBER | 6 |
| <input type="radio"/> | max_salary | NUMBER | 6 |
| <input type="radio"/> | | VARCHAR2 | |

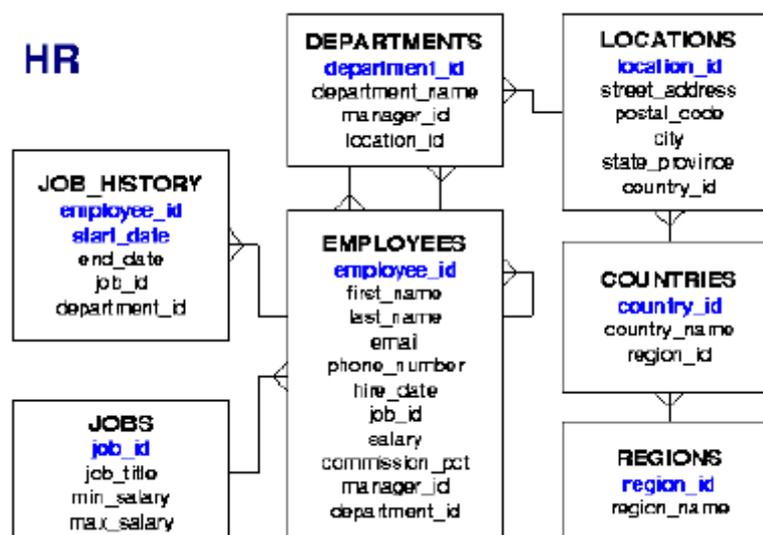
Add 5 Table Columns

4.5.6. Gestion des contraintes

4.5.6.1. Intégrité des données

Vous pouvez imposer des contraintes pour l'insertion de données dans vos colonnes en configurant des contraintes d'intégrité :

- **NOT NULL** : Par défaut, toutes les colonnes peuvent avoir des valeurs **NULL**. La valeur **NULL** signifie une absence de valeur. La contrainte **NOT NULL** contraint la colonne de la table à ne pas contenir de valeur **NULL**. Par exemple, vous pouvez définir une contrainte **NOT NULL** pour vous assurer que pour chaque ligne de la table *employees*, la colonne *last_name* contient une valeur.
- **UNIQUE key** : La contrainte d'intégrité **UNIQUE** requière que chaque valeur de la colonne soit unique. Il ne peut pas y avoir des lignes avec les mêmes valeurs. Par exemple, une contrainte **UNIQUE** est définie sur la colonne *department_name* de la table *department*, interdisant les doublons pour cette colonne.
- **PRIMARY KEY** : chaque table de la base peut avoir au plus une clé primaire. Les valeurs formées par d'une ou plusieurs colonnes sujet à cette contrainte constituent un identifiant unique de la ligne.
- Une clé primaire garantit les deux conditions suivantes :
 - Il ne peut pas y avoir deux lignes avec les mêmes valeurs dans la colonne spécifiée.
 - Il ne peut pas y avoir de valeur **NULL** dans cette colonne.
- Contrainte d'intégrité référentielle : Différentes tables dans une base de données relationnelle peuvent être reliées par des colonnes communes et des règles existent pour garantir que ce lien entre ces colonnes doit être maintenu. Les règles d'intégrités référentielles garantissent que ces rapports sont préservés. Une contrainte d'intégrité référentielle exige que pour chaque ligne d'une table, la valeur des clés étrangères corresponde à une valeur de la clé primaire. Une clé étrangère est définie sur la colonne *department_id* de la table *employee*. Cette contrainte garantit que chaque valeur de cette colonne correspond à une valeur de la clé primaire de la table *departments* (la colonne *department_id*). Un autre type de contrainte référentielle d'intégrité s'appelle une contrainte d'intégrité autoréférentielle. Ce type de clé étrangère fait référence à une clé primaire présente dans la même table.
- **CHECK** : Une contrainte d'intégrité **CHECK** sur une colonne ou un ensemble de colonnes exige que la condition soit vérifiée pour chaque ligne de la table. Si un ordre DML ne remplit pas cette condition, alors celui-ci est annulé.



4.5.6.2. Définitions des contraintes

Vous pouvez ajouter des contraintes à vos colonnes :

1. Sélectionnez la table dans la page Tables et cliquez sur Edit.
2. Cliquez sur l'onglet Constraints. La page Constraints vous montre toutes les contraintes que vous avez définies sur la table.
3. Sélectionnez le type de contrainte que vous voulez ajouter et cliquez sur ajouter.
4. Entrez l'information appropriée pour le type de contrainte que vous avez définies. Cliquez sur OK.

Database: orcl.us.oracle.com > Tables > Edit Table: HR.COUNTRIES

Add UNIQUE Constraint

Up to 32 columns can make up a UNIQUE key constraint. The unique key columns constitute a unique

Definition

Name

Table Columns

| Available Columns | | Selected Columns |
|-------------------------|--|------------------|
| COUNTRY_ID REGION_ID | <input type="button" value="Move"/> Move <input type="button" value="Move All"/> Move All <input type="button" value="Remove"/> Remove <input type="button" value="Remove All"/> Remove All | COUNTRY_NAME |

4.5.7. Attribue d'une table

Vous pouvez utiliser Enterprise Manager pour voir les attributs d'une table :

1. Cliquez sur le lien Tables dans la zone Schéma de la page Administration.
2. Sélectionnez la table et cliquez sur le bouton View pour voir vos attributs.

Database: orcl.us.oracle.com > Tables > Edit Table: HR.DEPARTMENTS

Edit Table: HR.DEPARTMENTS

[General](#)
[Constraints](#)
[Segments](#)
[Storage](#)
[Options](#)

* Name

Schema

Tablespace

Organization **Standard, Heap Organized**

Columns

| Select | Name | Data Type | Size |
|----------------------------------|-----------------|-----------|------|
| <input checked="" type="radio"/> | DEPARTMENT_ID | NUMBER | 4 |
| <input type="radio"/> | DEPARTMENT_NAME | VARCHAR2 | 30 |
| <input type="radio"/> | MANAGER_ID | NUMBER | 6 |
| <input type="radio"/> | LOCATION_ID | NUMBER | 4 |

Add 5 Table Columns

4.5.8. Contenu d'une table

Vous pouvez facilement voir les lignes de votre table en utilisant Enterprise Manager :

1. Sélectionnez la table sur la page Tables.
2. Sélectionnez View Data dans le menu Action et cliquez sur GO.

Les lignes de la table sont montrées dans la section de résultats. Le prompt vous montre la requête SQL qui a été exécutée pour produire ces résultats. Sur cette page, vous pouvez cliquer sur n'importe quelle colonne et trier vos données par ordre ascendant ou descendant. Si vous voulez modifier la requête, cliquez sur le bouton Refine Query.

Database: orcl.oracle.com > Tables > View Data for Table: HR.REGIONS Logged in As SYS

View Data for Table: HR.REGIONS

Refine Query OK

Query `SELECT "REGION_ID", "REGION_NAME" FROM "HR"."REGIONS"`

Result

| REGION_ID | REGION_NAME |
|-----------|------------------------|
| 1 | Europe |
| 2 | Americas |
| 3 | Asia |
| 4 | Middle East and Africa |

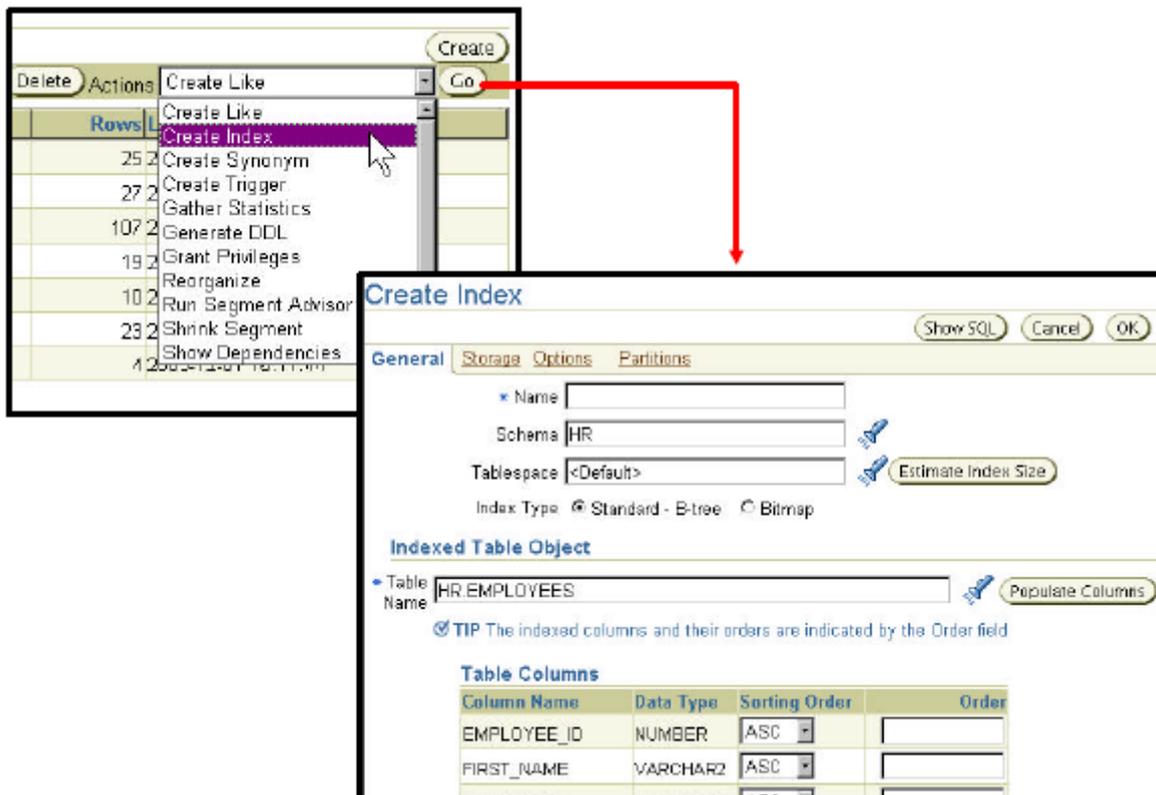
Refine Query OK

4.5.9. Action avec les tables

Vous pouvez choisir une table et puis effectuer plusieurs actions sur cette table :

- **CREATE LIKE** : Permet de créer une table avec la même structure que la table sélectionnée. Vous devez changer le nom de la table et le nom des contraintes. Vous pouvez aussi ajouter ou supprimer des colonnes ou faire des changements sur la structure de la table. Quand la table sera créée, elle sera vide.
- **CREATE INDEX** : Vous pouvez utiliser cette commande pour créer un index sur votre table. Des index doivent être créés uniquement sur les colonnes qui sont utilisées avec la condition WHERE d'un ordre SELECT et d'une instruction DML.
- **CREATE SYNONYM** : Un synonyme est un nom qui est utilisé à la place du nom complet de la table. Le synonyme peut être privé ou public.
- **CREATE TRIGGER** : Un trigger est un bloc PL/SQL qui est exécuté lorsqu'un événement particulier est effectué sur une table. Par exemple vous voulez garder une copie des données dans une table si on efface des données de la table employees. Vous allez donc créer un trigger « before delete » qui va insérer les lignes dans une autre table avant d'être effacé sur la table employees.
- Collecter des statistiques : cet assistant « Gather Statistics Wizard » vous aide à collecter et modifier des statistiques d'optimisation. Les statistiques sont stockées dans le dictionnaire de données et utilisées par l'optimiseur basé sur le coût (cost-based optimizer). Des statistiques mises à jour peuvent accroître les performances de vos requêtes SQL sur vos objets. Oracle suggère que vous automatisiez ces tâches pour générer de statistiques régulièrement.

- Générer des DDL : Affiche les ordres DDL de création pour la table sélectionnée. Vous pouvez copier/coller le contenu dans un fichier texte pour vous en servir de script.
- Accorder des privilèges : Par défaut quand vous créez une table, seul le propriétaire peut faire ce qu'il veut. Le propriétaire doit accorder des privilèges à d'autres utilisateurs pour pouvoir exécuter des ordres DML ou DDL sur celle-ci.
- Réorganiser : Vous pouvez employer l'outil Reorganize pour reconstruire des indexes ou des tables, pour déplacer des objets dans un tablespace différent, ou pour optimiser les attributs de stockage des objets indiqués.
- Segment Advisor : Segment Advisor détermine si les objets qui ont de l'espace inutilisé peuvent libérer cet espace, en prenant en compte une estimation future de l'espace requis.
- Shrink Segment : L'opération shrink fragmente l'espace et optionnellement libère de l'espace.
- Dépendance : Montre les objets dont la table dépend ou les objets qui dépendent de celle-ci.
- Vue des données : Exécute un ordre SELECT sur la table, vous pouvez redéfinir l'ordre SELECT. Par contre vous ne pouvez pas changer les données.
- **Flashback Table** : Flashback Table permet de récupérer une table à un moment précis.. Il fournit une méthode de récupération pour une table qui a été accidentellement modifiée ou supprimée par un utilisateur dans le passé. Vous pouvez restaurer les données de la table ainsi que ses attributs, comme les indexes, triggers et autres. Cette opération est possible tant que la base de données est online en annulant les changements qui ont été apportés à la table. Vous pouvez revenir à un état de la table et à son contenu à un certains temps ou à un numéro de SCN (System Change Number) spécifié. Utilisez Flashback Table avec Flashback Query et Row Versions pour trouver un temps à partir duquel la table doit être restaurée.
- **Flashback by row version** : Il vous permet d'interroger les métadonnées et un historique des données pour une période donnée.



4.5.10. Création d'indexe

Les indexes sont des structures optionnelles associées à une table. Ils sont créés afin d'améliorer les performances pendant la récupération des données. Un indexe Oracle fournit un chemin d'accès direct aux données de table. Des indexes peuvent être créés sur une ou plusieurs colonnes d'une table. Une fois l'indexe créé, il est automatiquement géré et utilisé par le serveur Oracle. Les mises à jour des données d'une table, comme l'ajout de nouvelles lignes, la mise à jour de lignes, ou la suppression de lignes, sont automatiquement propagées à tous les index appropriés avec une transparence complète pour l'utilisateur qui fait la modification.

Vous pouvez cliquer sur le lien Indexe de la page Administration pour voir la page des indexes. Vous pouvez voir les attributs des indexes ou utiliser le menu Action pour voir les dépendances de ces indexes.

Les indexes peuvent être créé explicitement ou implicitement.

Create Index

Show SQL Cancel OK

General Storage Options Partitions

* Name

Schema

Tablespace Estimate Index Size

Index Type Standard - B-tree Bitmap

Indexed Table Object

* Table Name Populate Columns

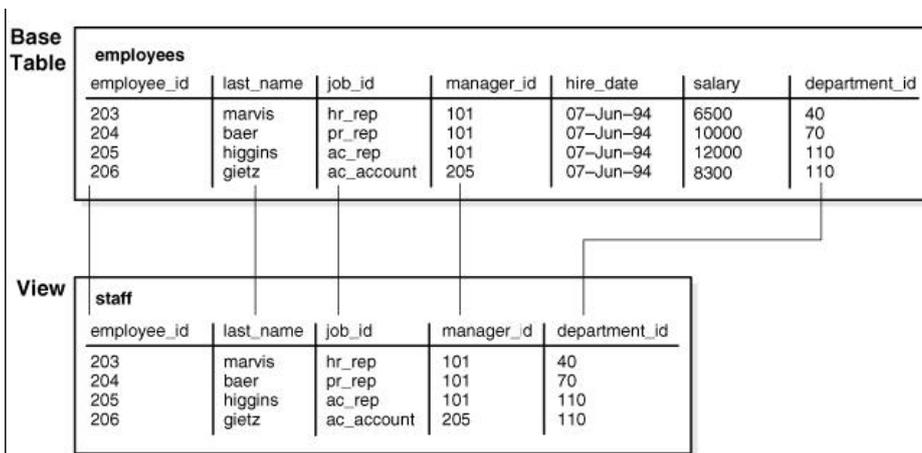
TIP The indexed columns and their orders are indicated by the Order field

Table Columns

| Column Name | Data Type | Sorting Order | Order |
|-------------|-----------|---------------|----------------------|
| EMPLOYEE_ID | NUMBER | ASC | <input type="text"/> |
| FIRST_NAME | VARCHAR2 | ASC | <input type="text"/> |
| LAST_NAME | VARCHAR2 | ASC | <input type="text"/> |

4.5.11. Qu'est-ce qu'une vue ?

Les vues sont des représentations logiques des données d'une ou plusieurs table ou d'une ou plusieurs vues. Les vues ne contiennent pas de données, mais à la place elles dérivent leurs données des tables sur lesquelles elles sont basées. Ces tables sont appelées tables de base d'une vue.



Création d'une vue

Comme pour une table, sur une vue vous allez pouvoir faire des SELECT, UPDATE, INSERT mais avec quelques restrictions. Toutes les opérations exécutées sur une vue affectent réellement les tables de base de la vue. Les vues fournissent un niveau additionnel de sécurité en restreignant l'accès à un ensemble prédéfini de lignes et colonnes d'une table. Vous allez pouvoir stocker des résultats de requêtes complexes.

Cliquez sur le lien View de la page Administration pour voir les vues qui sont dans la base de données.

Database: [orcl.us.oracle.com](#) > [Views](#) > Create View Logged in As SYS

Create View

Show SQL Cancel OK

General Options Object

* Name

* Schema 

Aliases

Replace the view if exists

* Query Text

```
SELECT  
EMPLOYEE_ID, LAST_NAME, JOB_ID, MANAGER_ID, DEPARTMENT_ID  
FROM  
EMPLOYEES
```

4.5.12. Qu'est-ce qu'une séquence ?

Une séquence est un objet de la base qui permet de générer des entiers uniques. Vous allez généralement utiliser des séquences pour générer les valeurs de vos clés primaires :

- Nom : Utilisez les conventions de nommage pour nommer une séquence.
- Schéma : C'est le propriétaire de la séquence.
- Type : Une séquence peut être en ordre croissant ou décroissant.
- Valeur maximum : Spécifie la valeur maximum que la séquence peut générer. Elle doit être supérieure à la valeur minimale ou initiale. Utilisez la clause Unlimited pour la valeur maximum de 10^{27} pour une séquence croissante ou -1 pour une décroissante. La valeur par défaut est unlimited.
- Valeur minimum : Spécifie la valeur minimum que la séquence peut générer. Elle doit être inférieure ou égale à la valeur initiale et inférieure à la valeur maximale. Par défaut, il n'y a pas de limite. Utilisez le paramètre Unlimited pour une valeur minimum de 1 pour une séquence croissante ou -10^{26} pour une décroissante.
- Intervalle : Spécifie l'intervalle entre deux nombres de séquences. Sa valeur peut être un entier positif ou négatif, mais ne peut pas être 0. Par défaut la valeur est 1 et doit être codée sur maximum 28 chiffres.
- Initial : Spécifie la première valeur que la séquence va générer. Ce chiffre doit être plus grand que la valeur minimum pour une séquence croissante et inversement pour une séquence décroissante.

- Cycle : Un fois que la séquence a atteint sa valeur maximum, elle générera la valeur minimum. Pour une séquence décroissante, une fois le minimum atteint, celle-ci recommencera par la valeur maximum. Si vous ne choisissez pas cette option, une erreur sera retournée si vous demandez une nouvelle valeur à une séquence épuisée.
- Cache : Spécifie combien de valeurs de la séquence sont mises en mémoire par Oracle pour avoir un accès plus rapide aux valeurs. Cette valeur peut être codée sur 28 chiffres ou moins. La valeur minimum pour ce paramètre est 2.

Create Sequence

Show SQL

Cancel

OK

General

* Name * Schema Type Ascending Descending

Values

* Maximum Value Value Unlimited* Minimum Value Value Unlimited* Interval * Initial

Options

 Cycle Values - Sequence will wrap around on reaching limit Order Values - Sequence numbers will be generated in order

Cache Options

 Use CacheCache Size

Utilisation d'une séquence

Pour manipuler vos séquences, vous avez des pseudo-colonnes :

- CURRVAL : retourne la valeur courante de la séquence
- NEXTVAL : incrémente la séquence et retourne la prochaine valeur

Vous devez mettre CURRVAL et NEXTVAL après le nom de la séquence :

```
sequence.CURRVAL
sequence.NEXTVAL
```

La première utilisation de NEXTVAL vous retournera la valeur initiale de la séquence. Les futures références à NEXTVAL incrémentent les valeurs de la séquence par le pas défini et retourne la nouvelle valeur. N'importe quel appel de CURRVAL, cela renvoie la valeur courante de la séquence, qui est la valeur retournée par le dernier NEXTVAL.

5. Module 05 : Gestion des données

5.1. Data Pump

5.1.1. Gestion des données avec le SQL

5.1.1.1. Introduction

Les ordres basiques du SQL permettent de manipuler les données dans la base de données.

```
INSERT INTO employees
VALUES (9999, 'BOB', 'Builder', 'bob@abc.net', NULL, SYSDATE,
       'IT_PROG', NULL, NULL, 100, 390);
```

```
1 row created.
```

```
UPDATE employees SET salary = 6000
WHERE employee_id = 9999;
```

```
1 row updated.
```

```
DELETE FROM employees
WHERE employee_id = 9999;
```

```
1 row deleted.
```

5.1.1.2. L'ordre INSERT

L'ordre **INSERT** créé une ligne à la fois. En utilisant une sous requête, vous pouvez utiliser l'ordre **INSERT** pour copier les lignes d'une table à une autre. L'exemple suivant illustre l'instruction **INSERT** :

```
INSERT INTO dept_80
(SELECT * FROM employees
 WHERE department_id = 80);
```

Dans cet exemple, la table `dept_80` a exactement la même structure que la table `employees`. Si ce n'est pas le cas vous pouvez explicitement nommer les colonnes dans chaque table. Les colonnes dans les ordres **INSERT** et **SELECT** devront obligatoirement correspondre (respect de l'ordre dans lequel vous avez nommé les colonnes et des types de données des colonnes).

Par exemple :

```
INSERT INTO just_names (first, last)
(SELECT first_name, last_name
 FROM employees);
```

Ici la table `just_name` possède seulement deux colonnes qui sont du même type que les colonnes `first_name` et `last_name` de la table `employees`. Utilisez l'ordre insert avec une sous requête pour charger un volume de données d'une ou plusieurs tables vers une autre table.

5.1.1.3. L'ordre UPDATE

L'ordre **UPDATE** est utilisé pour modifier les lignes d'une table. Le nombre de lignes modifiées par l'ordre **UPDATE** dépendra de la condition **WHERE**. Si la clause **WHERE** est omise, alors toutes les lignes seront modifiées. Si aucune ligne ne remplit la condition, alors aucune ligne ne sera modifiée. Par exemple :

```
UPDATE employees
SET    salary = salary * 1.1
WHERE  department_id = 90;
```

5.1.1.4. L'ordre DELETE

L'ordre **DELETE** est utilisé pour supprimer des lignes d'une table. Le nombre de lignes modifiées par l'ordre **DELETE** dépendra de la condition **WHERE**. Si la clause **WHERE** est omise, alors toutes les lignes seront supprimées. Si aucune ligne ne remplit la condition alors aucune ligne ne sera supprimée. Remarquez que si aucune ligne n'est supprimée, vous n'aurez pas un message d'erreur, le message retourné vous spécifiera qu'aucune ligne n'a été supprimée de la table.

Par exemple :

```
DELETE FROM employees
WHERE  department_id = 200;
```

5.1.1.5. Les ordres COMMIT et ROLLBACK

Par défaut tous les ordres DML ne sont pas validés de manière définitive (commit). Plusieurs outils (dont iSQL*Plus) dispose d'une option qui permet de configurer le commit pour chaque ordre ou sur un groupe d'ordre.

Avant qu'un **COMMIT** ou qu'un **ROLLBACK** ne soit effectué, les changements sont dans un état intermédiaire (en suspens). Seul l'utilisateur qui a effectué les modifications est autorisé à voir les données changées. Les autres utilisateurs peuvent afficher ces données, mais verront uniquement les données telles qu'elles étaient avant la modification. Les autres utilisateurs ne peuvent pas non plus effectuer d'ordre DML sur les données modifiées non validées.

Par défaut lorsqu'un utilisateur modifie une ligne, si un second veut modifier cette même ligne il devra attendre que le premier effectue un commit ou un rollback pour sa transaction. Ceci est géré automatiquement par les mécanismes de verrouillage d'Oracle. La base de données ne peut fonctionner sans verrous parce que le verrouillage est effectué sur la ligne elle-même.

Par exemple :

```
COMMIT;
```

```
ROLLBACK;
```

5.1.1.6. Contraintes d'intégrité et DML

Toutes les contraintes d'intégrité sont vérifiées lorsqu'un ordre DML est exécuté.

Pour les colonnes **FOREIGN KEY** :

- **INSERT** et **UPDATE** : La valeur doit exister dans la table parente.
- **DELETE** : Une ligne dans la table parente qui possède au moins une ligne qui lui fait référence ne peut être supprimée.

Pour les colonnes **NOT NULL** :

- **INSERT** : oblige l'insertion d'une valeur
- **UPDATE** : empêche de mettre à jour une valeur à **NULL**

Pour les colonnes avec une clé **UNIQUE** :

- **INSERT** et **UPDATE** : empêche d'avoir la même valeur dans une colonne, sauf pour les valeurs **NULL**. Chaque valeur **NULL** est considérée comme unique, donc toutes les lignes peuvent avoir une valeur **NULL** dans une colonne avec une contrainte **UNIQUE**.

Pour les colonnes **PRIMARY KEY** :

- Application des contraintes **UNIQUE** et **NOT NULL**

Pour les colonnes **CHECK** :

- **INSERT** et **UPDATE** : Les valeurs insérées ou mises à jour doivent respecter la condition spécifiée.

Exemple lorsque nous essayons de mettre à jour la colonne **department_id** à **99** :

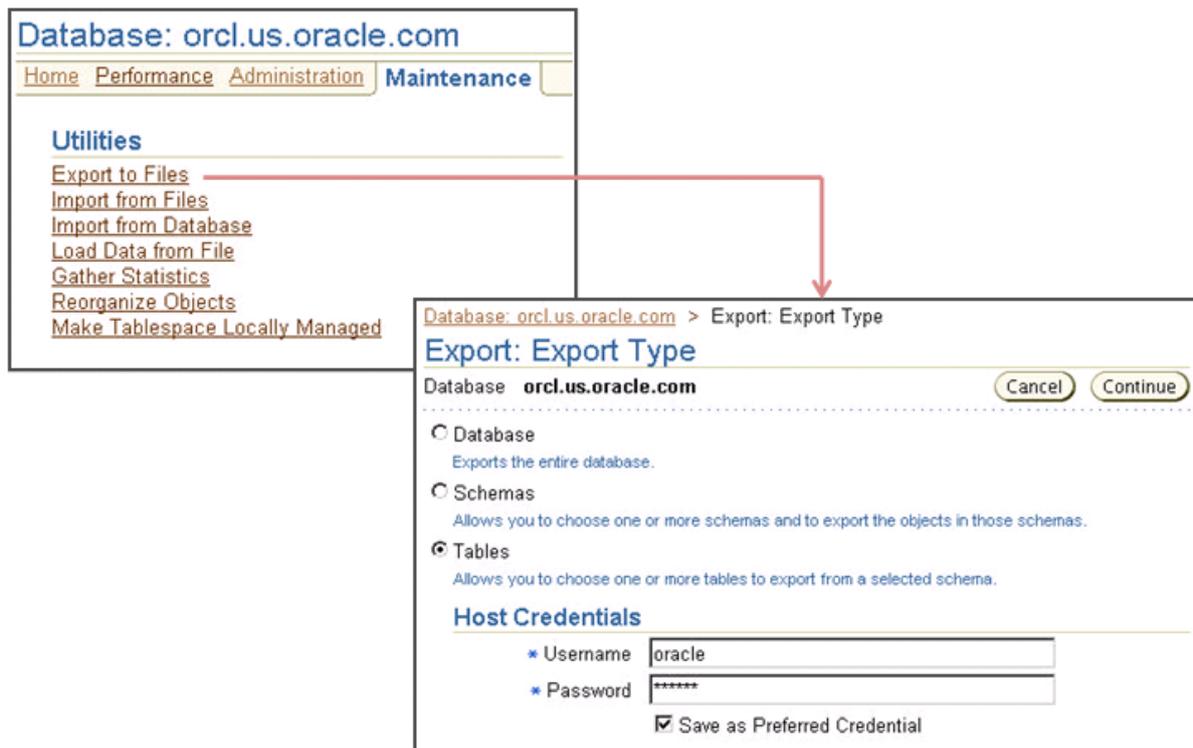
```
UPDATE employees
SET    department_id = 99
WHERE  last_name = 'KING';

UPDATE employees
*
ERROR at line 1:
ORA-02291: integrity constraint (HR.EMP_DEPT FK) violated - parent
key not found
```

5.1.2. Exportation de données avec Data Pump

L'outil Oracle Data Pump permet le transfert très rapide de données d'une base de données à une autre. Par exemple, vous pourrez l'utiliser si vous voulez exporter une table et ses indexes d'une base de données et l'importer dans une autre.

Sur la page Maintenance cliquez sur Export to Files pour utiliser Data Pump et écrire les données dans un fichier.



Sur la première page, sélectionnez ce que vous voulez exporter, et dans la section Host credentials, entrez votre nom et votre mot de passe de l'utilisateur du système.

- **Database** : exporte la base de données dans son ensemble, aussi appelé full export. Les schémas système suivants ne sont pas exportés dans le full export car les métadonnées qu'ils contiennent sont exportées en tant qu'objets dans le fichier de dump : **SYS**, **ORDSYS**, **ORDPLUGINS**, **CTXSYS**, **MDSYS**, **LBACSYS**, et **XDB**. Ce mode impose que vous ayez le rôle **EXP_FULL_DATABASE**.
- **Schemas** : exporte tous les objets d'un ou plusieurs schémas. Si vous n'avez pas le rôle **EXP_FULL_DATABASE**, alors vous ne pourrez spécifier que votre schéma. Les objets référencés présents dans un autre schéma ne sont pas exportés à moins que le schéma référencé soit aussi spécifié dans la liste des schémas à exporter.
- **Tables** : dans ce mode, seulement un ensemble de tables, de partitions, et leurs objets dépendants sont exportés. Vous devez avoir le rôle **EXP_FULL_DATABASE** pour spécifier les tables qui ne sont pas dans votre schéma, et un seul schéma peut être spécifié. Remarquez que la définition des types des colonnes n'est pas exportée dans ce mode. Il faut donc que la définition des types existe déjà dans l'instance cible au moment de l'import. Par ailleurs, comme dans l'export de schéma, les références aux schémas croisés ne sont pas exportées.

Si vous sélectionnez Schémas ou Tables pour le type d'export, la page suivante vous permettra de sélectionner un ensemble de schéma ou de tables à exporter. Cliquez sur Add pour sélectionner les schémas ou les tables à exporter.

La page Options est la prochaine étape, si vous sélectionnez le mode d'exportation Database vous serez redirigé vers cette page après que vous ayez sélectionné le mode d'exportation. Le nombre maximum de threads dans un export constitue le niveau de parallélisme à utiliser. Cliquez sur Estimate Disk Space Now pour afficher les résultats. Vous pouvez aussi spécifier des informations concernant les fichiers de log. Cliquez sur Advanced Options pour avoir les options suivantes :

- **Content** : Vous pouvez exporter uniquement les métadonnées, les données, ou les deux. Vous pouvez aussi configurer une condition pour filtrer les données selon le nom de l'objet.
- **Flashback** : Effectue une exportation consistante à partir d'un moment donné.
- **Query** : Récupère uniquement les données qui répondent à la condition **WHERE** que vous avez fournie.

La page **Files** est l'étape suivante. Cette page permet de spécifier le nom du répertoire, le nom du fichier, et la taille maximum pour le fichier d'export. Le %U est une variable de substitution qui commence avec 01 pour le premier fichier et continue avec 02, 03, etc.... Un nombre suffisant de fichiers de dump est créé pour permettre aux processus spécifiés par le paramètre **PARALLEL** d'être actifs. Par exemple, si expa%U, expb%U, et expc%U ont été configurés pour un job ayant un parallélisme de 6, les fichiers de dump initiaux créés seront expa01.dmp, expb01.dmp, expc01.dmp, expa02.dmp, expb02.dmp, et expc02.dmp.

La page suivante est la page **Schedule**. Vous pouvez ici planifier l'exécution du job.

La dernière page est la page **Review**. La page Review affiche le code PL/SQL généré suite aux choix effectués dans les pages précédentes. Vous ne pouvez pas changer le code généré. Vous pouvez copier le code ailleurs pour la création d'un script si nécessaire.

Cliquez sur Submit Job pour commencer l'exécution de la tâche.

Une fois le job validé, cliquez sur View Job pour surveiller la progression du job.

5.1.3. Importation de données avec Data Pump

Utilisez Data Pump Import pour charger des données extraites par Data Pump Export. Sur la page Maintenance, cliquez sur Import from Files pour lire les fichiers créés par Data Pump Export et les importer dans une base de données.

Files : La page Files permet de définir le chemin et le nom des fichiers à importer sur le serveur de base de données. Après, vous aller devoir choisir une des trois possibilités pour importer (vous pourrez importer un fichier entier, des schémas spécifiques, ou des tables spécifiques). Vous pouvez aussi choisir de générer un fichier de log et entrer les informations concernant votre nom d'utilisateur et le mot de passe.

Select Objects : Cette page est affichée seulement lorsque vous choisissez d'importer des objets spécifiques (à partir de la page Files). Utilisez cette page pour choisir les utilisateurs que vous voulez importer à partir de la liste Available Users, et en les déplaçant dans la liste Selected Users.

Re-Mapping : Cette page est affichée uniquement si vous avez précédemment sélectionné des objets à partir de la page Select Objects. Utilisez cette page pour définir si vous voulez importer les données de chaque utilisateur dans le même schéma pour chaque utilisateur, ou dans un schéma utilisateur différent de celui désigné dans les champs Source User et Destination User.

Options : La page Options permet de configurer les options pour la réutilisation d'un fichier de données, les erreurs de création d'objet, et les indexes non utilisables.

Schedule : Permet de planifier la procédure d'importation comme une tâche dans le système de gestion des jobs.

Review : Permet d'afficher le code PL/SQL généré pour l'ordre d'importation selon vos choix dans les pages précédentes. Vous pouvez éditer le code PL/SQL manuellement ou cliquez sur Import pour commencer l'importation des données.

Une fois la tâche validée, cliquez sur View Job pour surveiller la progression de la tâche.

Database: [orcl.us.oracle.com](#) > Import: Files

Import: Files

Database **orcl.us.oracle.com** Cancel Continue

Database Version of Files to Import **10g or later** Go
Changing the version affects attributes below.

Files

Specify the directory name and file name of the import files on the database server machine. Create Directory Object

Remove

| Select Directory Object | File Name |
|-------------------------------------|--------------|
| <input type="radio"/> DATA_FILE_DIR | EXPDAT%U.DMP |

Add Another Row

You can wildcard a set of dump files using "%U" in the filename.

Import Type

Entire files

Schemas
Allows you to choose one or more schemas and to import the objects in those schemas.

Tables
Allows you to choose one or more tables to import from a selected schema.

Host Credentials

* Username

* Password

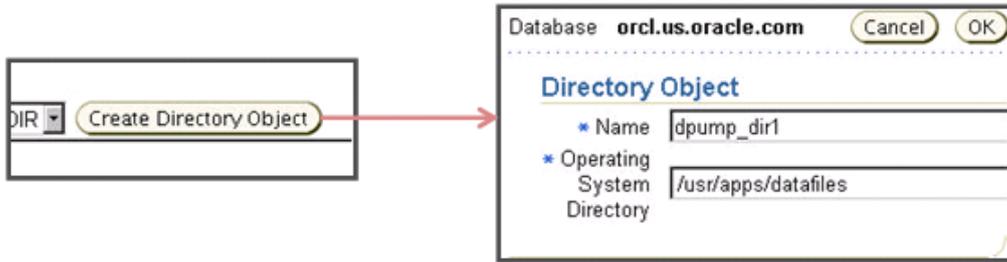
Save as Preferred Credential

5.1.4. Les objets de type répertoire

Les fichiers de dump, les fichiers de log, et les fichiers SQL sont accessibles à partir de répertoires situés sur le serveur parce que Data Pump est basé du côté serveur. Pour Data Pump, vous devez spécifier les chemins des répertoires en tant qu'objet. Un objet de type répertoire permet de faire correspondre le nom d'un objet de type répertoire avec le chemin d'un répertoire présent sur le système de fichier du serveur.

Utilisez l'ordre **CREATE DIRECTORY** pour créer un répertoire objet, ou cliquez sur Create Directory Object. Un objet répertoire spécifie un alias pour le répertoire présent sur le système de fichier du serveur. Vous devez disposer du privilège système **CREATE DIRECTORY** pour créer des objets répertoires. Quand vous créez un objet répertoire, vous obtenez automatiquement les droits en lecture/écriture sur ce répertoire, et vous pouvez attribuer ces droits à d'autres utilisateurs et rôles. Pour voir la définition du répertoire objet utilisez la vue **DBA_DIRECTORIES**.

Remarque : Oracle ne vérifie pas que le répertoire que vous avez spécifié existe. Ainsi, assurez-vous d'avoir spécifié un répertoire existant sur votre système de fichier. De plus, si votre système d'exploitation est sensible à la casse, assurez-vous de spécifier le chemin dans le bon format. (Cependant, vous n'avez pas à inclure un slash à la fin du chemin). Retenez que le système d'exploitation contrôlera la sécurité pour les fichiers dans le répertoire.



```
CREATE DIRECTORY dpump_dir1
AS      '/usr/apps/datafiles';

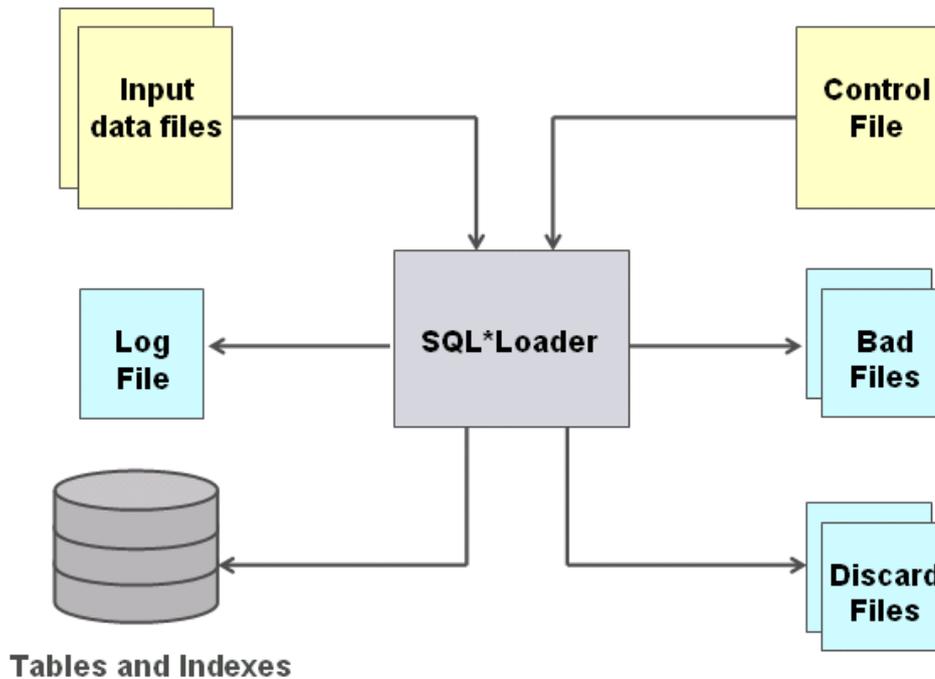
Directory created.
```

```
SELECT *
FROM   DBA_DIRECTORIES
WHERE  DIRECTORY_NAME = 'DPUMP_DIR1';
```

| OWNER | DIRECTORY_ | DIRECTORY_PATH |
|-------|------------|---------------------|
| SYS | DPUMP_DIR1 | /usr/apps/datafiles |

5.2. Charger des données avec SQL*Loader

5.2.1. Présentation



SQL*Loader charge les données issues de fichiers externes dans des tables de la base de données. Il dispose d'un puissant moteur avec peu de limitations concernant le format du fichier de données à charger.

Les fichiers utilisés par SQL*Loader sont les suivants :

- **Input Data Files** : SQL*Loader lit les données d'un ou plusieurs fichiers (à partir du système de fichier) spécifiés dans le fichier de contrôle. A partir de SQL*Loader, les données dans les fichiers de données sont organisées sous forme d'enregistrements. Un fichier de données particulier peut être dans un format d'enregistrement fixe, un format d'enregistrement variable, ou un format d'enregistrement stream. Le format d'enregistrement peut être spécifié dans le fichier de contrôle avec le paramètre **INFILE**. Si aucun format n'est spécifié, le format par défaut est stream.
- **Control File** : Le fichier de contrôle est un fichier texte écrit dans un langage que SQL*Loader comprend. Le fichier de contrôle indique à SQL*Loader où trouver les données, comment les parcourir, comment les interpréter, et où les insérer. Bien que cela ne soit pas décrit précisément, un fichier de contrôle peut comporter trois parties différentes.
 - La première partie contient les informations concernant la session, par exemple :
 - Les options globales comme la taille des enregistrements, les enregistrements à ignorer, etc ...
 - La clause **INFILE** spécifie le chemin du fichier de données
 - Les données à charger
 - La deuxième partie est constituée d'un ou plusieurs blocks **INTO TABLE**. Chaque bloc contient des informations (comme le nom de la table et le nom des colonnes de la table) à propos des tables dans lesquelles les données seront chargées.
 - La troisième partie est optionnelle et contient les données à charger.

- **Log File** : Quand SQL*Loader commence son exécution, il crée un fichier de log. S'il ne peut pas créer de fichier de log, l'exécution se termine. Le fichier de log contient un sommaire détaillé du chargement, incluant une description des erreurs qui sont survenues pendant le chargement.
- **Bad File** : Le bad file contient les enregistrements qui ont été rejetés, soit par SQL*Loader soit par le serveur Oracle. Les enregistrements sont rejetés par SQL*Loader quand le format en entrée est invalide. Après qu'un enregistrement ait été accepté pour être traité par SQL*Loader, il est envoyé vers la base de données Oracle pour être inséré dans une table en tant que ligne. Si le serveur Oracle détermine que la ligne est valide, alors la ligne est insérée dans la table. Si la ligne est invalide, alors l'enregistrement est rejeté et SQL*Loader l'inscrit dans le bad file.
- **Discard File** : Ce fichier est créé seulement si cela est nécessaire, et seulement si vous avez spécifié qu'il devra être utilisé. Le discard file contient les enregistrements qui ont été filtrés lors du chargement car ils ne correspondaient à aucun critère de sélection spécifié dans le fichier de contrôle.

5.2.2. Le fichier de contrôle de SQL*Loader

Le fichier de contrôle de SQL*Loader est un fichier texte qui contient des instructions de définition des données (DDL). Les ordres DDL sont utilisés pour contrôler les aspects suivants d'une session SQL*Loader :

- L'emplacement des données à charger
- Le formatage attendu pour ces données
- Comment charger les données (gestion de la mémoire, enregistrements rejetés, interruption du chargement, etc....)
- Comment manipuler les données qui vont être chargés.

L'exemple ci-dessous illustre un fichier de contrôle typique :

```
1 -- This is a sample control file
2 LOAD DATA
3 INFILE 'SAMPLE.DAT'
4 BADFILE 'sample.bad'
5 DISCARDFILE 'sample.dsc'
6 APPEND
7 INTO TABLE emp
8 WHEN (57) = '.'
9 TRAILING NULLCOLS
10 (hiredate SYSDATE,
    deptno POSITION(1:2) INTEGER EXTERNAL(3)
    NULLIF deptno=BLANKS,
    job POSITION(7:14) CHAR TERMINATED BY WHITESPACE
    NULLIF job=BLANKS "UPPER(:job)",
    mgr POSITION(28:31) INTEGER EXTERNAL
    TERMINATED BY WHITESPACE, NULLIF mgr=BLANKS,
    ename POSITION(34:41) CHAR
    TERMINATED BY WHITESPACE "UPPER(:ename)",
    empno POSITION(45) INTEGER EXTERNAL
    TERMINATED BY WHITESPACE,
    sal POSITION(51) CHAR TERMINATED BY WHITESPACE
    "TO_NUMBER(:sal, '$99,999.99')",
    comm INTEGER EXTERNAL ENCLOSED BY '(' AND '%' ":comm * 100"
)
```

Explication de l'exemple (par numéro de ligne) :

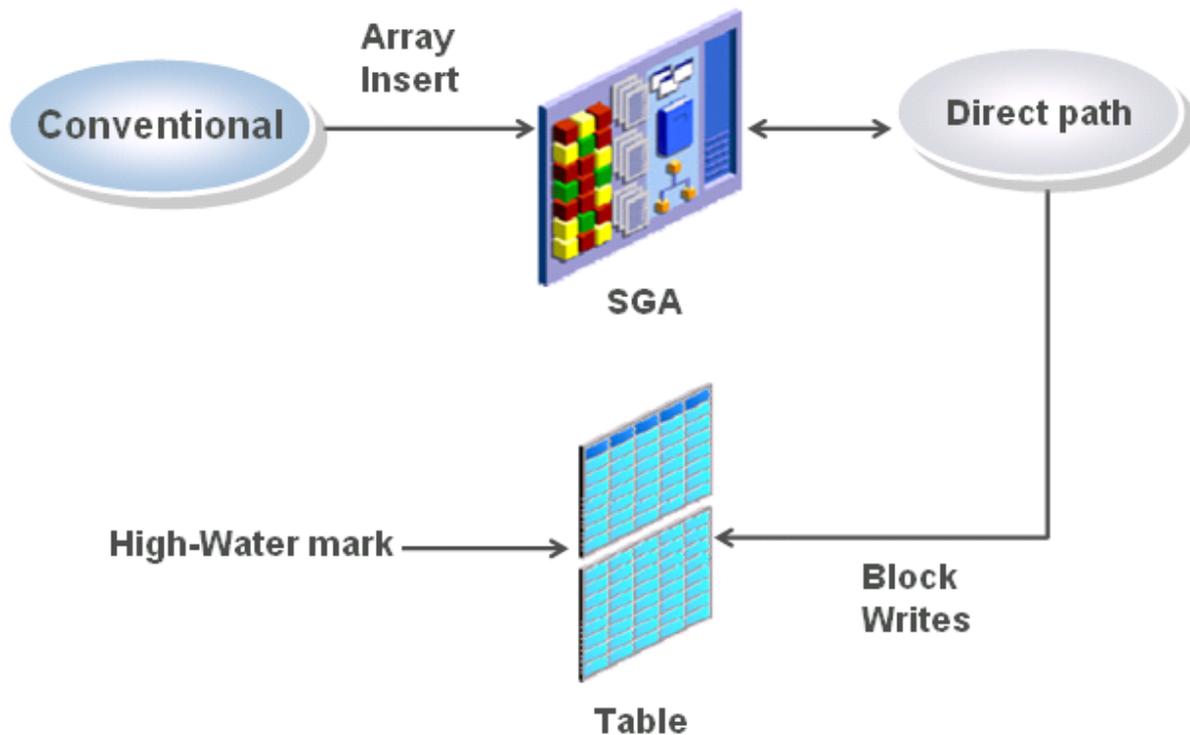
- 1 : Les commentaires peuvent apparaître n'importe où dans la partie de contrôle du fichier, mais ne doivent pas apparaître avec les données. Précédez tous les commentaires par deux traits d'union. Tout le texte à droite des deux traits d'union est ignoré, jusqu'à la fin de la ligne.
- 2 : L'ordre **LOAD DATA** indique à SQL*Loader qu'il s'agit d'un nouveau chargement de données. Si vous continuez un chargement qui a été interrompu, alors utilisez l'ordre **CONTINUE LOAD DATA**.
- 3 : Le mot clé **INFILE** spécifie le nom du fichier de données contenant les données que vous voulez charger.
- 4 : Le mot clé **BADFILE** spécifie le nom du fichier dans lequel les enregistrements rejetés seront placés.
- 5 : Le mot clé **DISCARDFILE** spécifie le nom du fichier dans lequel les enregistrements ne répondant pas aux contraintes sont placés.
- 6 : Le mot clé **APPEND** est une option que vous pouvez utiliser pour charger des données dans une table qui n'est pas vide. Pour charger des données dans une table vide, utilisez le mot clé **INSERT**.
- 7 : Le mot clé **INTO TABLE** vous permet d'identifier les tables, les champs, et les types de données. Cela définit la relation entre les enregistrements dans le fichier de données et les tables de la base de données.
- 8 : La clause **WHEN** spécifie une ou plusieurs conditions que les enregistrements doivent satisfaire avant d'être chargés par SQL*Loader.
- 9 : La clause **TRAILING NULLCOLS** indique à SQL*Loader comment traiter toute colonne dont la position est relative, qui ne sont pas présentes dans l'enregistrement en tant que colonnes **NULL**.
- 10 : Le reste du fichier de contrôle contient la liste des champs, qui fournit des informations concernant la structure de la table où les données vont être chargées.

Remarques sur la syntaxe du fichier de contrôle :

- La syntaxe est libre. (Les ordres peuvent être sur plusieurs lignes.)
- Le contenu n'est pas sensible à la casse; cependant, les chaînes de caractères entre simples et doubles apostrophes sont prises en tant que chaînes de caractères, et prennent en compte la casse.
- Dans la syntaxe du fichier de contrôle, les commentaires sont précédés de deux traits d'union (--) et marque le début du commentaire jusqu'à la fin de la ligne. La troisième partie optionnelle du fichier de contrôle est interprétée pour les données; et par conséquent, les commentaires dans cette partie ne sont pas permis.
- Le mot clé **CONSTANT** a une signification spéciale pour SQL*Loader et est donc réservé. Pour éviter d'éventuels conflits, n'utilisez pas le mot clé **CONSTANT** comme nom des tables ou des colonnes.

5.2.3. Les méthodes de chargement des données

5.2.3.1. Les deux méthodes



Le chargement conventionnel construit un tableau pour les lignes à insérer et utilise l'ordre **INSERT** pour charger les données. Pendant le chargement conventionnel, les enregistrements en entrée sont parcourus selon la spécification des champs, et un tableau d'enregistrement est construit et inséré dans la table spécifiée dans le fichier de contrôle. Les enregistrements qui ne sont pas conformes à la spécification des champs sont rejetés et les enregistrements qui ne satisfont pas aux critères de sélection sont rejetés.

Le chargement conventionnel peut être utilisé pour charger les données dans les tables en cluster ou non. La génération des redo est contrôlée par l'attribut **LOGGING** pour la table où les données seront chargées.

Un chargement direct construit les blocks de données dans la mémoire et sauvegarde ces blocks directement dans les extents alloués pour la table à charger. Les entrées de redo log ne sont pas générées à moins que la base de données ne soit en mode **ARCHIVELOG**. Le chargement direct utilise la spécification des champs pour construire l'ensemble des blocks de données, et écrit les blocks directement dans les fichiers de données en dessous du High water mark. Le chargement direct outrepassa le database buffer cache et à accède à la SGA seulement pour la gestion locale des extents et ajuste le High water mark.

5.2.3.2. Différences entre chargement direct et chargement conventionnel

| Chargement conventionnel | Chargement direct |
|---|--|
| Utilisez l'ordre COMMIT pour valider vos modifications | Utilise des données sauvegardées |
| Génère toujours des entrées redo | Génère des entrées redo selon des conditions spécifiques |
| Toutes les contraintes sont exécutées | Exécute seulement les contraintes PRIMARY KEY, UNIQUE, et NOT NULL |
| Les triggers INSERT se déclenchent | Les triggers INSERT triggers ne se déclenchent pas |
| Les tables en cluster peuvent être chargées | Les tables en clusters peuvent être chargées |
| D'autres utilisateurs peuvent modifier les tables | D'autres utilisateurs ne peuvent pas modifier les tables |

Le chargement conventionnel utilise le SQL et l'ordre **COMMIT** pour sauvegarder les données. L'insertion d'un tableau d'enregistrement est suivie par un commit. Chaque donnée chargée peut comporter plusieurs transactions.

Le chargement direct utilise les données sauvegardées pour écrire les blocks dans les fichiers de données. Les composants suivants différencient une donnée sauvegardée d'un **COMMIT** :

- Pendant la sauvegarde d'une donnée, seul les blocks entiers sont écrits dans la base de données.
- Les blocks sont écrits après le High water mark de la table.
- Après la sauvegarde d'une donnée, le High-water mark s'est déplacé.
- Les ressources internes ne sont pas libérées après la sauvegarde d'une donnée.
- Une donnée sauvegardée ne termine pas la transaction.
- Les indexes ne sont pas mis à jour à chaque sauvegarde de donnée.

Enregistrer les informations de logs pour les modifications :

Le chargement conventionnel génère des entrées similaires à un ordre DML. Quand on utilise un chargement direct, les entrées redo sont générées si :

- La base de données est en mode **NOARCHIVELOG**
- La base de données est en mode **ARCHIVELOG**, mais LOGGING est désactivé. LOGGING peut être désactivé en configurant l'attribut **NOLOGGING** pour la table ou en utilisant la clause **UNRECOVERABLE** dans le fichier de contrôle.

Exécution des contraintes :

Pour un chargement conventionnel, toutes les contraintes activées sont exécutées de la même façon qu'elles le sont avant l'exécution de n'importe quelle ordre DML.

Pour un chargement direct, les contraintes sont manipulées comme suit :

- Les contraintes **NOT NULL** sont vérifiées quand les tableaux sont construits.
- Les contraintes **FOREIGN KEY** et **CHECK** sont désactivées, et peuvent être activées à la fin de l'exécution en utilisant les commandes appropriées dans le fichier de contrôle. Les contraintes **FOREIGN KEY** sont désactivées car elles référencent d'autres lignes ou tables, et les contraintes **CHECK** sont désactivées car elles peuvent utiliser des fonctions SQL. Si un nombre limité de lignes doit être inséré dans une table volumineuse, utilisez le chargement conventionnel.
- Les contraintes **PRIMARY KEY** et **UNIQUE** sont vérifiées pendant et à la fin de l'exécution, et peuvent être désactivées si elles sont violées.

Déclenchement des triggers INSERT :

Les triggers **WHILE INSERT** sont déclenchés pendant les chargements conventionnels; ils sont désactivés avant le chargement direct et réactivés à la fin de l'exécution. Ils peuvent restés désactivés si un objet référencé n'est pas accessible à la fin de l'exécution. Utilisez le chargement conventionnel pour charger des données dans des tables avec des triggers **INSERT**.

Charger des données dans des tables en cluster :

Le chargement direct ne peut pas être utilisé pour charger des lignes dans des tables en clusters. Seul le chargement conventionnel permet de charger les tables en cluster.

Verrouillage :

Quand un chargement direct est en cours d'exécution, les autres transactions ne peuvent pas effectuer de modifications aux tables qui sont en cours de chargement. La seule exception à cette règle est quand plusieurs sessions de chargements parallèles directs sont utilisées simultanément.

5.2.4. Chargement des données

5.2.4.1. Données à charger et fichier de données

Format d'enregistrement fixe :

Un fichier est en d'enregistrement fixe lorsque tous les enregistrements dans un fichier de données sont de la même longueur (taille en octet). Même si ce format est le moins flexible, les performances sont meilleures que pour les formats variable ou stream. Le format d'enregistrement fixe est aussi simple à spécifier. Par exemple :

```
INFILE <datafile_name> "fix n"
```

Cet exemple montre que SQL*Loader devra interpréter le fichier de données en tant qu'enregistrement fixe où chaque enregistrement est de « n » octets.

L'exemple suivant est un fichier de contrôle qui spécifie un format d'enregistrement fixe pour le fichier de données. Le fichier de données contient 4 enregistrements physiques. Le premier enregistrement est [0001, abcd], qui est d'une taille de 9 octets exactement (en utilisant un octet pour chaque caractère), le retour chariot étant le dixième octet.

```
LOAD DATA
INFILE 'example.dat' "fix 10"
INTO TABLE example
FIELDS TERMINATED BY ','
(col1, col2)
```

Contenu du fichier de données "example.dat" :

```
0001,abcd
0002,fg hi
0003,klmn
```

Format d'enregistrement variable :

Un fichier est en format d'enregistrement de longueur variable quand la taille de chaque enregistrement est incluse au début de chaque enregistrement dans le fichier de données. Ce format fournit plus de flexibilité par rapport au format d'enregistrement fixe et procure davantage de performance que le format d'enregistrement stream. Par exemple, vous pouvez spécifier un fichier de données qui sera interprété en tant qu'enregistrement variable comme suit :

```
INFILE "datafile_name" "var n"
```

Dans cet exemple, **n** spécifie le nombre d'octet pour la taille d'un champ. Si **n** n'est pas spécifié, SQL*Loader considère une taille de 5. On aura une erreur si nous spécifions un nombre **n** plus grand que 40.

L'exemple suivant est un fichier de contrôle qui indique à SQL*Loader comment traiter les données du fichier **example.dat** et attend un format d'enregistrement variable où la taille des champs d'enregistrement est donnée par les 3 premiers caractères. Le fichier de données **example.dat** contient trois enregistrements physiques. Le premier enregistrement occupe 9 octets (009), le deuxième en occupe 10 (incluant un caractère pour le caractère de retour à la ligne), et le troisième 12. Cet exemple prend aussi en compte un caractère simple octet pour le fichier de données.

```
LOAD DATA
INFILE 'example.dat' "var 3"
INTO TABLE example
FIELDS TERMINATED BY ',' optionally enclosed by '"'
(col1 char(5), col2 char(7))
```

Le contenu du fichier de données "**example.dat**" est :

```
009hello,cd,
010world,im,
012my,name is,
```

Format d'enregistrement Stream:

Un fichier est en format d'enregistrement Stream quand les enregistrements n'ont pas de taille spécifiée ; à la place, SQL*Loader construit les enregistrements en analysant l'enregistrement qui doit se terminer par un caractère bien précise. Le format d'enregistrement Stream est le plus flexible, mais il peut avoir un effet négatif sur la performance. Pour spécifier qu'un de données sera traité comme un fichier organisé au format stream, nous allons utiliser la syntaxe suivante :

```
INFILE <datafile_name> ["str terminator_string"]
```

Le paramètre **terminator_string** est spécifié en tant que '**char_string**' ou **X'hex_string'** où :

- '**char_string**' est une chaîne de caractères entre simples ou doubles apostrophes
- **X'hex_string'** est une chaîne de caractère au format hexadécimal

Quand le paramètre **terminator_string** contient un caractère spécial (non affichable), il devrait être spécifié en tant que **X'hex_string'**. Cependant, des caractères non affichables peuvent être spécifiés en tant que ('**char_string**') en utilisant un backslash.

Par exemple nous avons les caractères non affichables suivants :

`\n` retour à la ligne
`\t` tabulation horizontale
`\f` formfeed
`\v` tabulation verticale
`\r` retour chariot

Si le jeu de caractère avec le paramètre **NLS_LANG** pour votre session est différent du jeu de caractère du fichier de données, les chaînes de caractères sont converties dans le jeu de caractère du fichier de données. Les chaînes de caractères hexadécimales sont certaines d'être dans le jeu de caractère du fichier de données, donc aucune conversion n'est effectuée. Si aucun `terminator_string` n'est spécifié, le caractère de nouvelle ligne (end-of-line) sera utilisé. Le caractère de nouvelle ligne est connecté au jeu de caractère du fichier de données. L'exemple suivant illustre le chargement de données au format d'enregistrement stream où le `terminator_string` est spécifié en utilisant une chaîne de caractère, `'|\n'`. L'utilisation du caractère backslash permet à la chaîne de caractère de spécifier le caractère non affichable de retour à la ligne.

```
LOAD DATA
INFILE 'example.dat' "str '\n'"
INTO TABLE example
FIELDS TERMINATED BY ',' optionally enclosed by '"'
(col1 char(5),
col2 char(7))
```

Le fichier de données "**example.dat**" va contenir:

```
hello,world,|
james,bond,|
```

5.2.4.2. Charger des données avec SQL*Loader

Utilisez l'assistant Load Data from file pour charger des données ne provenant pas d'une base de données Oracle vers une base de données Oracle. Pour afficher l'assistant, sélectionnez Load Data from File dans la partie Utilities de la page Maintenance (dans Enterprise Manager).

Vous pouvez charger les données comme suit :

- Connectez-vous à Enterprise Manager en tant que **SYS** ou en utilisant un autre compte avec des privilèges d'administration.
- Cliquez sur Load Data from File sur la page Maintenance.
- Spécifiez le nom et chemin complet de votre fichier de contrôle sur le serveur pour le chargement des données : spécifiez aussi le nom d'utilisateur et le mot de passe. Cliquez sur Next.
- Sélectionnez "Provide the full path and name on the database server machine" sur la page Load Data : Data File et spécifiez le chemin. Cliquez sur Next.
- Par défaut la méthode par défaut pour charger les données est conventionnelle sur la page Load Data : Load Method. Vous pouvez choisir un chargement direct si cela est nécessaire.
- Sélectionnez "Generate log file where logging information is to be stored" sur la page Load Data : Options. Vous pouvez choisir le nom et chemin du fichier par défaut ou spécifier des paramètres différents. Remarquez qu'à partir de cette page vous pouvez limiter le nombre de ligne à charger. Cliquez sur Next.
- Spécifiez un nom et une description pour le job sur la page Load Data : Schedule. Sélectionnez **Immediately** pour exécuter le job immédiatement. Cliquez sur Next.
- La page Load Data : **Review** vous permet d'avoir un récapitulatif du nom du fichier et des méthodes de chargements. Si vous voulez modifier quelque chose, vous pouvez cliquer sur Back. Sinon, cliquez sur Submit Job pour démarrer le chargement.
- la page Status s'affiche avec le message Load Data Submit Successful. Cliquez sur View Job pour afficher le récapitulatif du job. La page Summary vous indique si le job a réussi. Vous pouvez afficher le contenu du fichier de log en cliquant sur votre job.
- Vous pouvez confirmer le chargement des données en navigant sur la page Tables, sélectionnez la table, et choisissez View Data.

ORACLE Enterprise Manager 10g Database Control Database

Control File Data File Load Method Options Schedule Review

Load Data: Control File

Database orcl.oracle.com Cancel Finish Step 1 of 6 Next

A control file is used to describe what will be loaded and how. Specify the full path and name of the control file on the database server machine.

/u01/app/oracle/oradata/orcl/LOAD.CTL

Host Credentials

* Username oracle

* Password *****

Save as Preferred Credential

6. Module 06 : Sécurité

6.1. Paramètres et Privilèges

6.1.1. Sécurité de la base de données

6.1.1.1. Présentation

Tous les utilisateurs ne doivent pas avoir accès à toutes les données. En fonction de ce qui est stocké dans votre base de données, des limites d'accès peuvent être imposées en tenant compte des nécessités de son activité professionnelle, des demandes des clients et des restrictions de la loi.

Pour renforcer le contrôle sur l'accès des données sensibles, le système devra s'assurer de l'identité de ceux qui essaient d'y accéder. Compromettre l'authentification rend tout le système de sécurité inutile.

Le mot de passe est un moyen simple de s'assurer de l'identité de nos utilisateurs. On pourra aussi s'assurer de la complexité de ce mot de passe en utilisant des fonctions de vérification.

Oracle supporte aussi des techniques d'authentification avancées comme la biométrie mais aussi les certificats.

Les comptes qui ne sont pas utilisés doivent être bloqués.

Il faut aussi savoir que même authentifiés correctement, les utilisateurs peuvent compromettre le système, pour cela Oracle offre aussi une riche variété d'outils qui permettent de surveiller les actions des utilisateurs et d'identifier les activités suspectes.

6.1.1.2. Principe du moindre privilège

Oracle Database 10g est le leader de l'industrie de la sécurité. Pour cela, afin de maximiser la sécurité des fonctionnalités offertes par Oracle Database 10g dans n'importe quel domaine, il est impératif que Oracle Database 10g elle-même soit bien protégée et bien configurée.

Le principe du moindre privilège signifie que les utilisateurs n'auront accès qu'aux privilèges qui leur permettront d'effectuer correctement leurs tâches. Cela réduit les risques qu'un utilisateur modifie ou ait accès à des données qui ne les concernent pas.

6.1.2. Paramètres de sécurité

6.1.2.1. Protection du dictionnaire de données

Les personnes qui ne sont pas administrateurs n'ont pas besoin d'accéder au dictionnaire de données, mais elles peuvent y accéder si vous leur donnez les privilèges système du type *** ANY TABLE** tels que **SELECT ANY TABLE** ou **UPDATE ANY TABLE**. Le dictionnaire de données contient des informations qu'un utilisateur mal intentionné pourrait utiliser pour pénétrer le système ou le corrompre. Pour protéger ce dictionnaire de données de ces privilèges système *** ANY TABLE**, il suffit de mettre la valeur du paramètre d'initialisation **O7_DICTIONARY_ACCESSIBILITY** à **FALSE**. Et s'il y a des personnes non administrateur qui ont besoin d'accéder au dictionnaire de données, vous pouvez les y autoriser en:

- utilisant la commande **GRANT** pour un objet bien défini du dictionnaire de données
- donnant le privilège système **SELECT ANY DICTIONARY** pour donner un accès à l'ensemble du dictionnaire de données.

Dans Oracle Database 10g et Oracle 9i database, la valeur par défaut du paramètre **O7_DICTIONARY_ACCESSIBILITY** est **FALSE**. Par contre pour les versions 8i et antérieures, la valeur par défaut est **TRUE** donc il faut penser à la régler manuellement à **FALSE**.

Remarque : Si ce paramètre est à **TRUE** tout utilisateur ayant le privilège **DROP ANY TABLE** sera capable de supprimer inconsciemment ou volontairement des données du dictionnaire.

6.1.2.2. Restreindre aux utilisateurs l'accès aux répertoires du système d'exploitation

Le paramètre d'initialisation **UTL_FILE_DIR** permet de désigner sur quels répertoires les programmes PL/SQL auront le droit d'écrire ou de lire. Par défaut aucun répertoire n'est accessible, les privilèges du système d'exploitation prévalent. Les répertoires non accessibles par la personne ayant démarré la base de données ne le seront que s'il les configure grâce à **UTL_FILE_DIR**. Pour spécifier plusieurs répertoires il suffit de les mettre entre apostrophes puis de les séparer par des virgules.

Ce paramètre n'est pas dynamique donc l'instance doit être redémarrée pour que les changements prennent effet. Aussi lorsque que vous spécifiez les chemins d'accès vers les répertoires vous ne pouvez pas utiliser les variables d'environnement.

L'instance ne vérifie pas si le répertoire spécifié existe ou pas donc on peut renseigner le paramètre **UTL_FILE_DIR** avant de créer les répertoires.

Les informations contenues dans les dossiers autorisés sont accessibles par tous les utilisateurs PL/SQL donc le contenu de ces dossiers ne sera pas confidentiel pour aucun d'entre eux.

Remarque: Il ne faut jamais utiliser: **UTL_FILE_DIR = ***, parce que cela permettrait aux utilisateurs d'accéder à tous les répertoires accessibles par l'instance Oracle, les répertoires des fichiers de données et de redo log inclus.

6.1.2.3. Désactiver l'authentification par système d'exploitation à distance

Cette fonctionnalité est désactivée par défaut. Quand elle est activée l'authentification externe des utilisateurs est déléguée au système d'exploitation du client. Cela veut dire que l'instance aurait confiance implicitement aux utilisateurs qui se sont correctement authentifiés sur leur machine cliente. Dans ce cas les utilisateurs pourront se connecter à l'instance sans donner de mot de passe à condition que le login soit le même pour la base de données et le système d'exploitation et qu'il soit correctement authentifié par son système d'exploitation.

L'authentification par le système d'exploitation est très déconseillée surtout pour les utilisateurs qui sont sur des PCs. Activer cette fonctionnalité réduit de beaucoup la sécurité du système.

6.1.3. Privilèges

6.1.3.1. Retirer les privilèges non nécessaires à PUBLIC

Tous les privilèges dont dispose le groupe d'utilisateur **PUBLIC** seront automatiquement affectés à tout utilisateur de la base de données, pour cela il est important de retirer certains privilèges et rôles non nécessaires au groupe **PUBLIC**.

Parmi ces privilèges nous avons **EXECUTE** qui concerne plusieurs packages PL/SQL du type **DBMS_*** et **UTL_***. Pour respecter le principe du moindre privilège vous devrez enlever du groupe **PUBLIC** le privilège **EXECUTE** sur certains de ces packages **DBMS_*** et **UTL_***. Les restrictions affectant le groupe **PUBLIC** affectent tous les utilisateurs. Certaines restrictions peuvent ne pas être pratiques dans ce cas il faudra préférer l'audit à ces restrictions.

Pour certains packages il est vivement recommandé de restreindre l'accès ; parmi eux nous pouvons citer :

- **UTL_SMTP** : qui peut permettre à la base de données d'œuvrer comme un serveur de mail SMTP. En donnant ce privilège à **PUBLIC** il pourrait y avoir des échanges de mail non autorisés.
- **UTL_TCP** : qui permet d'établir les connexions réseau entre la base de données Oracle et tout service réseau en attente. Ce package pourrait permettre l'envoi frauduleux de données présentes dans la base de données.
- **UTL_HTTP** : qui permet à la base de données de répondre ou d'envoyer des requêtes HTTP. Il pourrait permettre l'envoi de données via les formulaires HTML à des sites web pirates.
- **UTL_FILE** : Une mauvaise configuration pourrait permettre d'accéder à tous les fichiers présents dans le serveur. Et même s'il était bien configuré un problème pourrait survenir du fait que deux applications pourraient écrire en même temps dans un même répertoire.
- **DBMS_OBFUSCATION_TOOLKIT** : utilisé pour l'encryption de données. L'affectation de ce privilège est très restreinte parce que les données encryptées ne pourront être restaurées que si les clés d'encryptage sont stockées en sécurité et bien gérés.

Ces packages sont très pratiques pour les applications qui en ont besoin mais requièrent une configuration appropriée pour être utilisés en sécurité. Il ne faudra en aucun cas les affecter à **PUBLIC** mais les affecter individuellement aux utilisateurs qui en ont besoin et uniquement en cas de nécessité absolue.

Vous pourrez utiliser la requête suivante afin de lister les objets appartenant à **SYS** et sur lesquels **PUBLIC** détient le privilège **EXECUTE** :

```
SELECT table_name
FROM   dba_tab_privs
WHERE  owner='SYS'
AND    privilege = 'EXECUTE'
AND    grantee='PUBLIC'
```

```
TABLE_NAME
-----
AGGXMLIMP
AGGXMLINPUTTYPE
...
XMLTYPEEXTRA
XMLTYPEPI
437 rows selected.
```

6.1.3.2. Limiter les utilisateurs ayant des privilèges administratives

Il ne faut jamais donner aux utilisateurs plus de privilèges que ceux nécessaires. Pour implémenter le principe du moindre privilège, il faut procéder à des restrictions sur les privilèges suivants :

- privilèges système et objet.
- privilèges de connexion de l'utilisateur **SYS** comme **SYSDBA** et **SYSOPER**.
- autres privilèges pour les DBA comme **DROP ANY TABLE**.

Vous pourrez utiliser la requête ci-dessous pour avoir la liste des utilisateurs qui ont le rôle DBA. Seuls les administrateurs ont besoin du rôle DBA. Vous pourrez étudier les privilèges dont les utilisateurs ont besoin et leur donner seulement ces privilèges.

```
SELECT grantee
FROM dba_role_privs
WHERE granted_role = 'DBA';
```

| GRANTEE |
|---------|
| SYS |
| SYSTEM |

Exemple : Pour voir la liste des utilisateurs qui ont les privilèges **SYSDBA** ou **SYSOPER** nous pouvons utiliser la requête suivante :

```
SELECT *
FROM V$PWFILE_USERS;
```

| USERNAME | SYSDBA | SYSOPER |
|----------|--------|---------|
| SYS | TRUE | TRUE |

Les utilisateurs outre que **SYS** ont rarement besoin des privilèges **SYSDBA**.

6.2. Profils

6.2.1. Fonctionnalités

La gestion des mots de passe sous Oracle se fait grâce aux profils. Quand les utilisateurs sont créés ils ont le profil **DEFAULT** à moins qu'un autre profil leur soit attribué.



6.2.1.1. Verrouillage du mot de passe

| Paramètre | Description |
|------------------------------|---|
| FAILED_LOGIN_ATTEMPTS | Nombre de connexion sans succès avant le verrouillage du compte |
| PASSWORD_LOCK_TIME | Nombre de jours durant lequel le compte restera verrouillé |

Le serveur Oracle verrouille automatiquement le compte après que l'utilisateur ait effectué un certain nombre de tentatives de connexion sans succès. Ce nombre de connexion sans succès est limité par le paramètre **FAILED_LOGIN_ATTEMPTS**. Ce compte sera automatiquement déverrouillé après la période de temps déterminée par le paramètre **PASSWORD_LOCK_TIME** ou l'administrateur devra utiliser la commande **ALTER USER**.

La commande **ALTER USER** peut être utilisée pour verrouiller un compte; Enterprise Manager peut aussi être utilisé. Par ailleurs lorsque le compte est verrouillé manuellement, il ne sera pas automatiquement déverrouillé après la période **PASSWORD_LOCK_TIME**, donc le DBA devra le faire manuellement.

```
ALTER USER hr ACCOUNT LOCK;
User altered.
```

```
CONNECT hr/hr
ERROR: ORA-28000: the account is locked
Warning: You are no longer connected to ORACLE.
```

```
CONNECT / as sysdba
Connected.
```

```
ALTER USER hr ACCOUNT UNLOCK;
User altered.
```

6.2.1.2. Durée et expiration du mot de passe

| Paramètre | Description |
|----------------------------|--|
| PASSWORD_GRACE_TIME | Période de grâce pour changer le mot de passe après la première connexion après l'expiration du mot de passe |
| PASSWORD_LIFE_TIME | Durée en jours après lequel le mot de passe expire automatiquement |

L'administrateur de la base de données peut spécifier une période de grâce à l'aide du paramètre **PASSWORD_GRACE_TIME**, qui commence après la première connexion après l'expiration du mot de passe. Durant cette période un message d'alerte affichera à chaque connexion de l'utilisateur. Si l'utilisateur ne change pas de mot de passe, le compte sera verrouillé.

Remarque: Dans le cas d'un compte d'application (pas accessible via SQL*PLUS), il faudra vérifier que l'application est capable de récupérer les messages d'erreurs avant d'activer l'expiration des mots de passe.

La plupart des DBA assigne des profils séparés pour les comptes d'utilisateurs et les outils de monitoring. Un compte d'utilisateur peut être forcé à expirer en faisant expirer le mot de passe.

```
ALTER USER hr PASSWORD EXPIRE;
User altered.
```

```
CONNECT hr/hr
ERROR: ORA-28001: the password has expired
Changing password for hr
New password: *****
Retype new password: *****
Password changed
```

6.2.1.3. Historique des mots de passe

| Paramètre | Description |
|----------------------------|--|
| PASSWORD_REUSE_TIME | Nombre de jours durant lequel le mot de passe ne pourra pas être utilisé |
| PASSWORD_REUSE_MAX | Nombre de fois que le mot de passe actuel pourra être changé |

L'historique des mots de passe permet de s'assurer qu'un mot de passe ne sera pas réutilisé pendant une période donnée.

Les deux paramètres cités ci-dessus sont mutuellement exclusives, donc si l'une des paramètres a une valeur autre que **UNLIMITED**, l'autre paramètre devra être à **UNLIMITED**.

6.2.1.4. Vérification du mot de passe

Avant d'assigner un nouveau mot de passe à un utilisateur, une fonction PL/SQL peut être invoquée afin de vérifier la validité du mot de passe. La fonction de mot de passe doit être créée dans le schéma **SYS**.

Le serveur Oracle fournit une fonction de vérification par défaut que l'on peut visualiser en éditant le script `$ORACLE_HOME/rdbms/admin/utlpwdmg.sql`. Le script `utlpwdmg` permet de créer la fonction **VERIFY_FUNCTION** mais aussi de changer les paramètres du profil **DEFAULT** grâce à la commande **ALTER PROFIL**.

L'administrateur de la base de données peut écrire une fonction PL/SQL qui répond aux besoins de sécurité.

Les fonctions de vérification doivent respecter les spécifications suivantes pour la déclaration de variables en entrée :

```
function_name(userid_parameter      IN VARCHAR2,
              password_parameter    IN VARCHAR2,
              old_password_parameter IN VARCHAR2)
RETURN BOOLEAN
```

Si la fonction de mot de passe soulève une exception ou devient invalide, un message d'erreur est retourné et les commandes **ALTER USER** ou **CREATE USER** sont arrêtées.

6.2.2. Gestion des profils

6.2.2.1. Création d'un profil de mot de passe

The screenshot shows the 'Create Profile' dialog box with the following settings:

- General** / **Password** tab selected.
- Password** section:
 - Expire in (days): 90
 - Lock (days past expiration): 10
- History** section:
 - Number of passwords to keep: UNLIMITED
 - Number of days to keep for: 120
- Complexity** section:
 - Complexity function: VERIFY_FUNCTION
- Failed Login** section:
 - Number of failed login attempts to lock after: 3
 - Number of days to lock for: 5/1440

Pour créer un profil de mot de passe, il faut ouvrir Enterprise Manager et naviguer dans la page d'administration. Sélectionnez Profils et puis appuyez sur le bouton Create.

Les valeurs courantes de chaque paramètre peuvent être choisies dans une liste déroulante sinon vous pouvez aussi saisir la valeur du paramètre.

Toutes les périodes sont exprimées en jours mais les fractions pourront aussi être utilisées. Par exemple comme il y a 1440 minutes par jour, 5/1440 désignera 5 minutes.

Enterprise Manager peut aussi être utilisé pour éditer les profils existants.

6.2.2.2. Assigner un profil à un utilisateur

Edit User: NGREENBERG

[General](#)
[Roles](#)
[System Privileges](#)
[Object Privileges](#)
[Quotas](#)
[Consumer Groups](#)
[Proxy Users](#)

Name **NGREENBERG**

Profile

Authentication

* Enter Password

* Confirm Password

Expire Password now

* Default Tablespace 

Temporary Tablespace 

Status Locked Unlocked

Pour assigner un profil à un utilisateur:

- Ouvrir Enterprise Manager puis naviguer dans la page d'administration.
- Sélectionner Users. Sélectionner l'utilisateur auquel vous souhaitez assigner un profil puis cliquez sur le bouton Edit.
- Dans la liste déroulante Profil, sélectionnez le profil que vous voulez lui attribuer et appuyer sur le bouton Apply.

Un utilisateur ne pourra avoir qu'un profil à un moment donné. Si les utilisateurs sont déjà logués et que des changements dans leur profil ont lieu, alors ces changements ne prendront effet qu'après leur prochaine connexion.

Les comptes utilisateurs peuvent aussi être verrouillés ou mis à expiration à partir de la page d'édition des utilisateurs.

6.2.2.3. Gestion des comptes par défaut

Oracle Database 10g compte plusieurs comptes créés par défaut. Ces comptes sont destinés à stocker des données mais aussi à posséder les objets PL/SQL et des programmes Java. Quand nous utilisons DBCA (Database Creation Assistant) pour créer la base de données; il verrouille et fait expirer tous les comptes par défaut sauf :

- **SYS**
- **SYSTEM**
- **DBSNMP**
- **SYSMAN**

Il faut savoir que vous pouvez créer une base de données grâce à un script. Les bases de données créées grâce à des scripts n'ont pas les comptes par défaut bloqués. Aussi plusieurs applications créent des comptes qui nécessitent d'être verrouillés.

6.3. Audit

6.3.1. Surveillance d'activité suspecte

6.3.1.1. Comparaison des outils d'audit

Oracle Database 10g offre trois différents types d'audit. L'administrateur peut auditer toutes les actions effectuées dans la base de données. Il est bien de se rappeler que faire de l'audit augmente la charge de travail du système. Pour cela l'audit doit être bien ciblé afin que seule la trace des événements importants soit conservée.

| Type d'audit | Qu'est qui est audité? | Quelles sont les données enregistrées? |
|--------------------------------------|---|--|
| Audit Standard de la base de données | Usage des privilèges et l'accès aux données. | Ensemble fixe de données |
| Audit basé sur la valeur | Données modifiées par des requêtes DML. | Définies par l'administrateur |
| Fine-Grained Auditing (FGA) | Requêtes SQL (insert, update, delete, and select) basées sur le contenu | Ensemble fixe de données en incluant les requêtes SQL. |

6.3.1.2. Mise à jour de sécurité

Les alertes de sécurité Oracle contiennent une courte description sur les vulnérabilités, une évaluation du risque et le degré d'exposition lié à ces vulnérabilités mais aussi des patches ou les démarches à effectuer pour résoudre les problèmes. Oracle fournira aussi les informations sur la personne ou l'organisation qui l'a informé de la vulnérabilité.

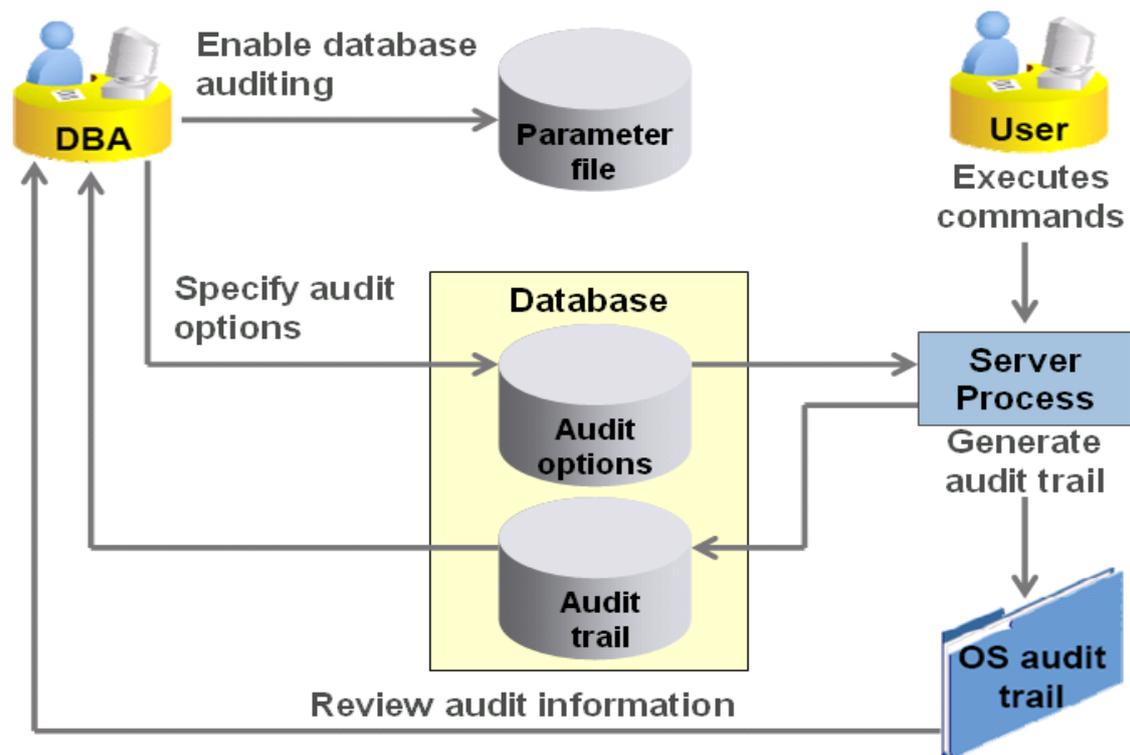
Les alertes de sécurité sont à poster sur le site web d'Oracle Technology Network (<http://otn.oracle.com/deploy/security/alerts.htm>) et sur OracleMetaLink (Metalink). Bien que les alertes de sécurité soient diffusées pour tout le monde, seules les personnes disposant d'un numéro d'identification de support clientèle (CSI : Customer Support Identification) pourront télécharger les patches.

Si vous découvrez une faille de sécurité dans n'importe quel produit Oracle vous pourrez le signaler sur Metalink ou en envoyant un mail à secalert_us@oracle.com.

6.3.2. Audit standard de la base de données

L'audit standard de la base de données permet de capturer plusieurs informations telles qu'un événement a eu lieu, les informations sur l'auteur de cet événement et les informations sur la machine utilisée.

6.3.2.1. Mécanisme de l'audit standard de la base de données



Après avoir activé l'audit de la base de données (grâce au paramètre **AUDIT_TRAIL**) et spécifié les options d'audit (grâce à des requêtes SQL que nous verrons par la suite), la base de données pourra commencer à collecter les informations d'audit.

Si le paramètre **AUDIT_TRAIL** est réglé à **OS**, alors les enregistrements d'audit seront stockés au niveau du système d'audit du système d'exploitation. Dans un environnement Windows ce sera un log d'événement. Dans le cas d'un environnement UNIX ou Linux, les enregistrements d'audit seront stockés dans un fichier. L'emplacement est déterminé grâce au paramètre **AUDIT_FILE_DEST**. Si le paramètre **AUDIT_TRAIL** est réglé à **DB**, les enregistrements d'audit seront dans une table qui appartiendra au schéma **SYS**.

L'audit est une tâche administrative très importante. En fonction des événements qui seront ciblés, la taille des données collectées peut croître de manière rapide. S'il n'est pas bien ciblé, l'audit peut consommer une si grande quantité d'espace que cela risque d'altérer les performances du système.

6.3.2.2. Spécification des options d'audit

- **Audit des requêtes SQL:** la requête **AUDIT TABLE** permettra d'auditer toutes les requêtes DDL (Data Definition Language) exécutées sur les tables en y incluant **CREATE TABLE**, **DROP TABLE**, **TRUNCATE TABLE**. Ce type d'audit peut être ciblé par rapport au login ou au succès ou échec.

```
AUDIT TABLE BY hr WHENEVER NOT SUCCESSFUL;
```

- **Audit des privilèges système:** utilisé pour contrôler l'utilisation des privilèges système comme **DROP ANY TABLE**. Il peut être ciblé par rapport au login ou au succès ou échec. Par défaut, à chaque fois qu'un privilège système est utilisé, un enregistrement est effectué. Vous pouvez choisir de regrouper ces enregistrements afin qu'un seul enregistrement soit généré par session (on pourra utiliser cela lorsqu'un utilisateur met à jour 100 000 enregistrements présents dans une table qui ne lui appartient pas). Si la clause **BY SESSION** n'est pas spécifiée, la valeur par défaut est **BY ACCESS**. Vous pourrez utiliser la clause **BY SESSION** pour limiter les impacts de l'audit des privilèges système sur la performance et les capacités de stockage.

```
AUDIT select any table, create any trigger;
```

```
AUDIT select any table BY hr BY SESSION;
```

- **Audit des privilèges objet:** utilisé pour auditer les actions sur les tables, les vues, procédures, séquences, répertoires et les types de données utilisateur. Il peut être ciblé par rapport au login ou au succès ou échec et groupé par session ou par accès. Contrairement à l'audit des systèmes privilèges, les enregistrements sont regroupés par session donc vous devrez spécifier implicitement **BY ACCESS** pour que chaque utilisation d'un privilège objet du même type soit enregistré.

```
AUDIT ALL on hr.employees;
```

```
AUDIT UPDATE, DELETE on hr.employees BY ACCESS;
```

L'option **AUDIT SESSION** permet d'auditer les créations de session d'un utilisateur. Il peut être ciblé par rapport au login ou au succès ou échec. Cette option sera seule parce qu'il permettra de générer un seul enregistrement pour chaque connexion à l'instance. L'enregistrement d'audit sera effectué lors de la connexion et mis à jour lors de la déconnexion. Toutes les informations sur une session comme la date de connexion et de déconnexion, les entrées sorties physiques et logiques générées ... seront stockées dans un enregistrement unique. Dans la plupart des bases de données il est plus recommandé d'utiliser la commande **AUDIT SESSION** (non ciblé). Parfois il est préférable d'utiliser **AUDIT SESSION WHENEVER NOT SUCCESSFUL**, parce que cela permettra de détecter les tentatives d'intrusion dans la base de données. Si l'option **AUDIT ALL** est utilisée avec le login:

```
AUDIT ALL BY hr;
```

Toutes les requêtes DDL de l'utilisateur (hr dans ce cas) et concernant les objets ci-dessous seront auditées :

| | | | |
|----------------------|----------------|--------------|------------------|
| Alter System | Cluster | Context | Create Session |
| Database Link | Dimension | Directory | Index |
| Materialized View | Not Exists | Procedure | Profil |
| Public Database Link | Public Synonym | Role | Rollback Segment |
| Sequence | Synonym | System Audit | System Grant |
| Table | Tablespace | Trigger | Type |
| User | View | | |

6.3.2.3. Visualiser les options d'audit

Pour voir les options sélectionnées concernant l'audit, vous pourrez utiliser la liste des vues suivantes:

| Vue du dictionnaire de données | Description |
|--------------------------------|--|
| ALL_DEF_AUDIT_OPTS | Options par défaut |
| DBA_STMT_AUDIT_OPTS | Options concernant l'audit des requêtes. |
| DBA_PRIV_AUDIT_OPTS | Options pour les privilèges |
| DBA_OBJ_AUDIT_OPTS | Options pour les l'audit des objets |

Pour chaque option d'audit nous aurons deux statuts possibles : SUCCES/PAS DE SUCCES avec trois valeurs possibles pour chaque statut:

- : pour audit pas activé

S: collecte des enregistrements d'audit par session

A: collecte des enregistrements d'audit par accès.

Exemple :

```
SELECT object_name, object_type, ins, upd
FROM dba_obj_audit_opts WHERE object_name = 'EMPLOYEES'
```

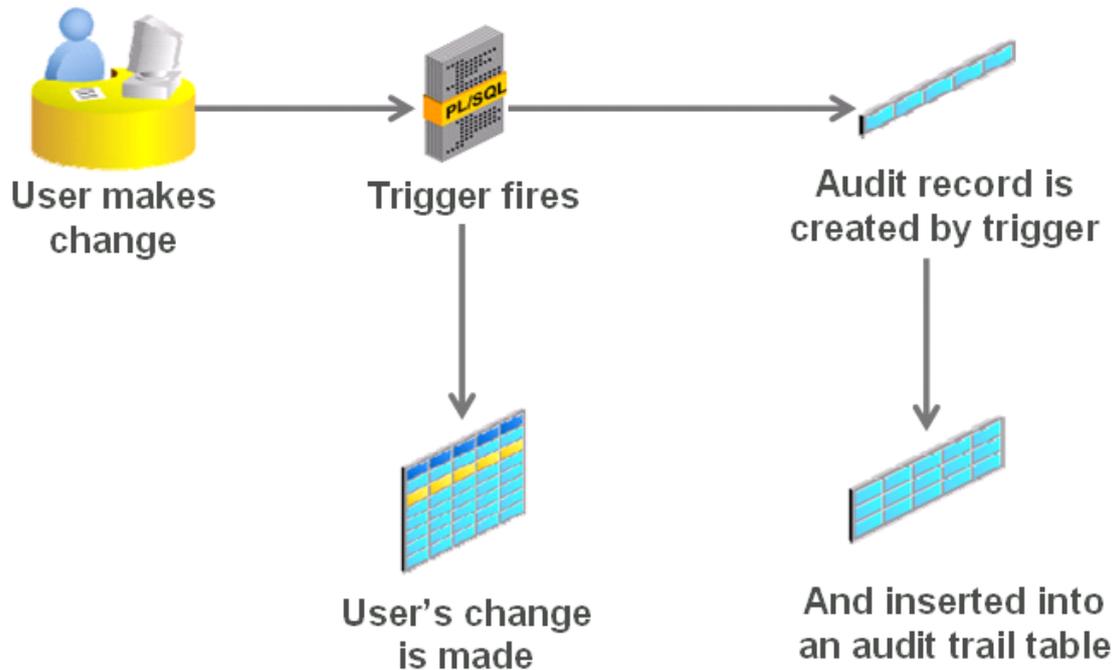
| OBJECT_NAME | OBJECT_TYPE | INS | UPD |
|-------------|-------------|-----|-----|
| EMPLOYEES | TABLE | A/S | -/- |

6.3.2.4. Visualiser les résultats d'audit

L'accès aux enregistrements d'audit doit être très bien contrôlé par qu'ils peuvent contenir des informations sensibles. En général la tâche de gestion des enregistrements d'audit est effectuée par l'administrateur mais en cas de nécessité le rôle **DELETE_CATALOG_ROLE** peut être accordé pour permettre la suppression d'enregistrement d'audit.

| Vue contenant les données d'audit | Description |
|-----------------------------------|--|
| DBA_AUDIT_TRAIL | Tous les enregistrements d'audit |
| DBA_AUDIT_EXISTS | Enregistrements pour AUDIT EXISTS/NOT EXISTS |
| DBA_AUDIT_OBJECT | Enregistrement concernant les objets de schéma |
| DBA_AUDIT_SESSION | Toutes les entrées sur les connexions et les déconnexions. |
| DBA_AUDIT_STATEMENT | Enregistrements sur l'audit des requêtes |

6.3.3. Audit basé sur des valeurs



Vous pouvez utiliser l'audit basé sur les valeurs pour auditer les changements sur les données (insertion, mises à jour, suppressions). L'audit basé sur la valeur étend l'audit standard de la base de données, il permet d'auditer un événement mais aussi de garder les valeurs qui ont été insérées, mises à jour ou supprimées. Ce type d'audit est mis en place grâce aux triggers (déclencheurs).

Quand un utilisateur insère, met à jour ou supprime des données ayant un trigger associé, le trigger va travailler en arrière plan pour copier les informations d'audit dans une table créée pour cela. L'audit basé sur la valeur a plus tendance à dégrader les performances que l'audit standard parce que les triggers seront exécutés à chaque suppression, mise à jour ou insertion. Le degré de dégradation dépendra aussi l'efficacité du code du trigger.

L'audit basé sur la valeur sera utilisé uniquement si les informations capturées par l'audit standard ne sont pas suffisantes. La clé de l'audit basé sur la valeur est le trigger.

Exemple :

```
CREATE OR REPLACE TRIGGER system.hrsalary_audit
AFTER UPDATE OF salary
ON hr.employees
REFERENCING NEW AS NEW OLD AS OLD
FOR EACH ROW
BEGIN
  IF :old.salary != :new.salary THEN
    INSERT INTO system.audit_employees
    VALUES (sys_context('userenv','os_user'), sysdate,
    sys_context('userenv','ip_address'),
    :new.employee_id || ' salary changed from
    '||:old.salary|| ' to '||:new.salary);
  END IF;
END;
/
```

Ce trigger cible l'audit afin de conserver les informations sur les changements apportés sur la colonne **hr.employees**.

Quand le salaire sera mis à jour, si la nouvelle valeur est différente de l'ancienne valeur alors le trigger va insérer un nouvel enregistrement dans la table **audit_employees** (qui a été créée dans le schéma **SYSTEM** précédemment). Dans cet enregistrement nous aurons le login, l'adresse IP de la machine à partir de laquelle le changement a été effectué, la clé primaire de la ligne mis à jour et la valeur actuelle qui a été changée.

Les triggers de base de données peuvent aussi permettre de capturer des informations sur les connexions des utilisateurs au cas où l'audit standard ne donnerait pas d'informations suffisantes.

Grâce au triggers de connexions, l'administrateur pourra avoir :

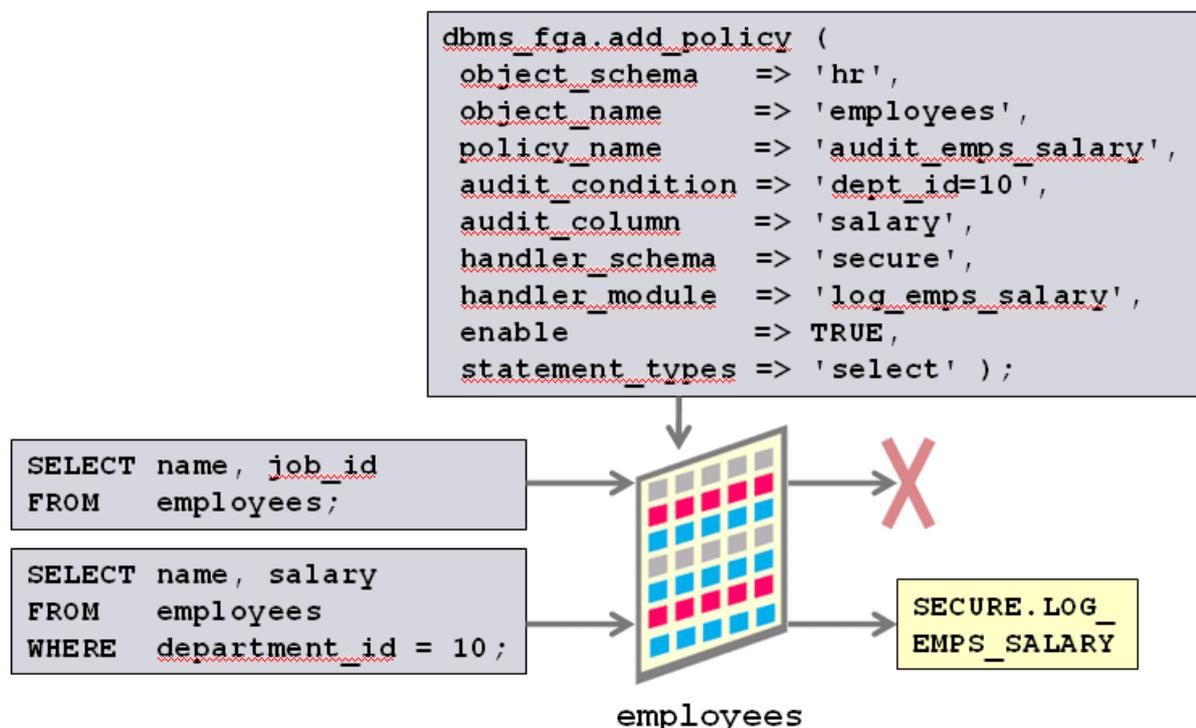
- L'adresse IP de la personne qui s'est connectée
- Les 48 premiers caractères du nom du programme que l'utilisateur a utilisé pour se connecter à l'instance.
- Le nom du terminal utilisé pour se connecter à l'instance.

6.3.4. Fine-Grained Auditing (FGA)

Les enregistrements de l'audit standard de la base de données permettent de savoir qu'une opération a eu lieu mais ne donnent pas la requête qui a été utilisée. L'audit Fine-Grained étend cette capacité en conservant les requêtes SQL permettant de manipuler des données. La FGA permet de cibler plus en détail l'audit que les deux autres types d'audit.

Contrairement à l'audit basé sur la valeur, la FGA n'utilise pas de trigger et a le même impact sur la performance que l'audit standard.

L'administrateur va utiliser le package PL/SQL **DBMS_FGA** pour créer une stratégie d'audit sur une table ou une vue. Si une des lignes provenant du résultat d'une requête répond à la condition d'audit, alors un enregistrement d'audit sera créé et stocké. Optionnellement nous pourrions faire appel à une procédure qui sera exécutée lorsque la condition d'audit a été satisfaite. Le FGA peut avoir comme cible une requête **SELECT** qui retourne des milliers de lignes mais ne va générer qu'un seul enregistrement d'audit.



6.3.4.1. Stratégie d'audit FGA

Dans l'exemple ci-dessus nous voyons comment créer une stratégie FGA grâce à la procédure **DBMS_FGA.ADD_POLICY**. Cette procédure accepte les arguments suivants:

- **Nom de la stratégie** : Chaque stratégie de type FGA doit être nommée. Dans l'exemple la stratégie a été nommée **AUDIT_EMPS_SALARY** en utilisant la syntaxe suivante :

```
policy_name => 'audit_emps_salary'
```

- **Condition d'audit** : La condition d'audit est un prédicat SQL qui permet de définir quand est ce que l'événement d'audit est déclenché. Dans notre exemple, toutes les lignes concernant le département « sales » seront auditées en utilisant :

```
audit_condition => 'department_id = 10'
```

- **Type de requête** : Cet argument permettra de spécifier le type de requêtes à auditer. Les choix possibles sont : **SELECT**, **INSERT**, **UPDATE**, **DELETE**.
- **Colonne à auditer** : Cet argument permet de définir les données qui seront auditées. Donc un événement d'audit aura lieu si seulement la colonne en argument est spécifiée dans la clause **SELECT**. Dans notre exemple c'est la colonne **salary** qui est auditée :

```
audit_column => 'salary'
```

Cet argument est optionnel. S'il n'est pas spécifié seul l'argument **AUDIT_CONDITION** sera pris en compte.

- **Objet** : L'objet est la table ou la vue qui sont auditées. Il utilise deux arguments :
 - Le schéma qui contient l'objet
 - Le nom de l'objet.

Dans notre exemple nous allons auditer la table **employees** qui est dans le schéma **hr** en utilisant les arguments suivants :

```
object_schema => 'hr'
object_name => 'employees'
```

- **Handler** : Le handler (capteur) est optionnel et est une procédure PL/SQL qui permet de définir toute action supplémentaire qui pourrait être effectuée durant l'audit. Par exemple, le handler d'événement pourrait envoyer une page d'alerte à l'administrateur. Si un handler d'événement est défini, alors les enregistrements d'audit seront stockés et puis le handler sera exécuté. Les enregistrements d'audit vont inclure la stratégie qui a causé l'événement, l'utilisateur qui a exécuté la requête SQL et la requête SQL avec les variables hôtes qu'elle a utilisées. Le capteur d'événement utilise deux arguments :
 - Le schéma qui contient le programme PL/SQL
 - Le nom du programme PL/SQL.

Dans notre exemple c'est la procédure **LOG_EMPS_SALARY** localisée dans le schéma **SECURE** qui sera appelé grâce aux deux arguments :

```
handler_schema => 'secure'
handler_module => 'log_emps_salary'
```

- **Statut** : Le statut indique si la stratégie est activée ou pas. Dans notre exemple notre stratégie est activée grâce à :

```
enable => TRUE
```

Par défaut la stratégie d'audit est activée lorsqu'elle est créée.

6.3.4.2. Le package DBMS_FGA

Le package **DBMS_FGA** est l'outil d'administration pour les fonctions utilisées pour l'audit Fine-Grained. Le privilège **EXECUTE** sur le package **DBMS_FGA** est nécessaire pour administrer les stratégies de ce type d'audit, par contre ce privilège devra être attribué uniquement qu'aux administrateurs. En désactivant la stratégie d'audit, on arrête aussi de générer des événements d'audit. Vous pourrez redémarrer la génération d'événements d'audit en réactivant la stratégie.

- **Pour activer une stratégie :**

```
EXEC dbms_fga.enable_policy (
  object_schema => 'hr',
  object_name => 'employees',
  policy_name => 'audit_emps_salary');

PL/SQL procedure successfully completed.
```

- **Pour désactiver une stratégie :**

```
EXEC dbms_fga.disable_policy(
  object_schema => 'hr',
  object_name => 'employees',
  policy_name => 'audit_emps_salary' );

PL/SQL procedure successfully completed.
```

- **Pour supprimer une stratégie :**

```
EXEC dbms_fga.drop_policy(
  object_schema => 'hr',
  object_name => 'employees',
  policy_name => 'audit_emps_salary' );

PL/SQL procedure successfully completed.
```

6.3.4.3. Consultation des événements d'audit

Les enregistrements d'audit pour la FGA sont stockés dans des tables différentes en fonction du type d'audit. Les trois dernières vues citées ci-dessous contiennent la définition des stratégies.

| Nom de la vue | Description |
|----------------------------|---|
| DBA_FGA_AUDIT_TRAIL | Tous les événements de type FGA |
| ALL_AUDIT_POLICIES | Toutes les stratégies pour les objets dont l'utilisateur actuel a accès. |
| DBA_AUDIT_POLICIES | Toutes les stratégies dans la base de données |
| USER_AUDIT_POLICIES | Toutes les stratégies pour les objets appartenant au schéma de l'utilisateur actuel |

Le tableau ci-dessous donne une liste des colonnes contenues dans la vue **DBA_FGA_AUDIT_TRAIL**.

| Nom | Description |
|----------------------|--|
| TIMESTAMP | La date et l'heure d'exécution |
| DB_USER | Le login dans la base de données de l'utilisateur |
| OS_USER | Le login dans le système d'exploitation de l'utilisateur |
| OBJECT_SCHEMA | Le propriétaire de l'objet audité |
| OBJECT_NAME | Le nom de l'objet audité |
| POLICY_NAME | Le nom de la stratégie qui a causé l'événement d'audit. |
| SCN | Le SCN de la transaction |
| SQL_TEXT | La requête SQL qui a causé l'événement d'audit |
| SQL_BIND | La variable hôte de l'événement audité, sous le format suivant : #n(s):v: où n est la position de la variable hôte dans la requête, s la longueur de la variable et v la valeur de la variable |
| COMMENT\$TEXT | Un commentaire |

6.3.4.4. Les directives concernant la FGA

- **Condition d'audit:** Lorsque vous créez une nouvelle stratégie d'audit, la condition d'audit est à **NULL** par défaut, ce qui signifie que toutes les requêtes seront auditées.
- **Erreur sur le nom de la stratégie :** Le nom de la stratégie doit être unique dans la base de données. Si vous voulez nommer une stratégie par un nom qui existait déjà vous allez avoir l'erreur suivante :

```
ORA-28101: policy already exists
```

- **Erreur sur l'audit des objets:** La table ou la vue que vous voulez auditer doit exister lors de la création de la stratégie sinon vous aurez l'erreur :

```
ORA-00942: table or view does not exist
```

- **Erreur sur la condition d'audit:** Si la syntaxe de la condition d'audit est invalide, la stratégie sera créée sans erreur mais une erreur va être affichée lorsque l'objet audité sera utilisé :

```
ORA-28112: failed to execute policy function
```

Si la condition d'audit a une syntaxe correcte mais est sémantiquement incorrecte, les bonnes lignes ne seront pas auditées.

- **Erreur sur la colonne d'audit :** Si la colonne à auditer n'existe pas dans la table, la stratégie sera créée mais aucune ligne sera auditée par il n'y aura jamais d'accès à la colonne qui n'existe pas.
- **Erreur sur le capteur d'événement:** Lorsque la stratégie fait référence à un capteur d'événement invalide ou non existant, la stratégie sera créée mais aucune ligne ne sera auditée.

6.3.4.5. Audit des utilisateurs de type **SYSDBA** et **SYSOPER**

Les utilisateurs de type **SYSDBA** et **SYSOPER** ont les privilèges de démarrer et d'arrêter la base de données. Les enregistrements d'audit pour ces utilisateurs doivent être stockés en dehors de la base de données car ces utilisateurs peuvent effectuer des modifications même si la base de données est éteinte. Oracle capture par défaut les événements de connexion pour les utilisateurs de type **SYSDBA**, les autres événements ne seront audités que lorsque cela sera explicitement spécifié.

Vous pourrez autoriser l'audit des autres événements pour les utilisateurs de type **SYSDBA** et **SYSOPER** en mettant le paramètre d'initialisation **audit_sys_operations** à **TRUE** en sachant que la valeur par défaut est **FALSE**.

Si les opérations de l'utilisateur **SYS** sont auditées, le paramètre **AUDIT_FILE_DEST** indiquera l'endroit où les enregistrements d'audit seront stockés. Dans un environnement de type Windows les enregistrements d'audit seront stockés dans les logs d'événements Windows. Dans les plateformes de type UNIX ou Linux les enregistrements d'audit seront stockés dans le répertoire :
\$ORACLE_HOME/rdbms/audit.

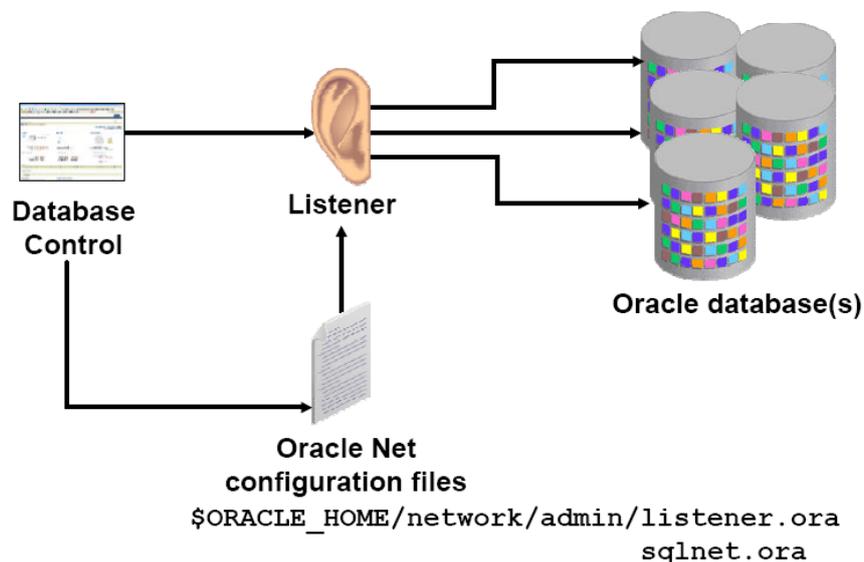
7. Module 07 : Oracle Net Services

7.1. Oracle Net Listener

7.1.1. Définition

Au sein de la base de données, Oracle Net se matérialise par un processus actif appelé le listener. Oracle Net Listener est responsable de la coordination des connexions entre la base de données et les applications externes. Sans listener aucune connexion depuis l'extérieur vers la base de données ne serait possible.

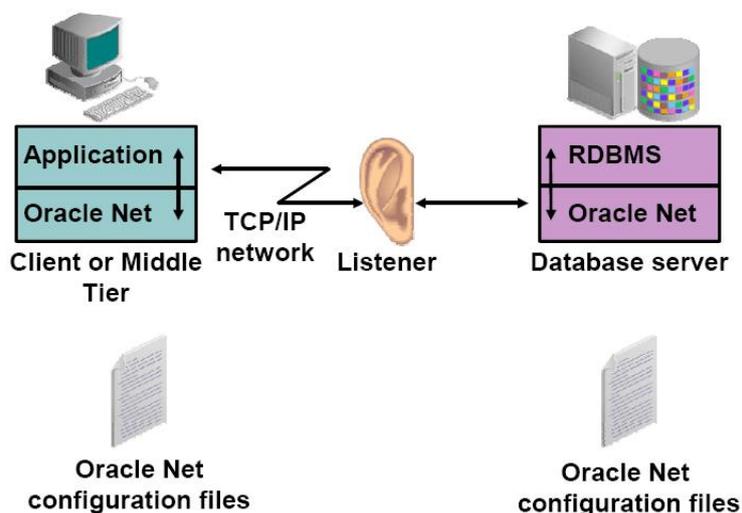
Oracle Net Listener sera donc la passerelle pour l'instance Oracle pour toutes les connexions à distance. Un Listener peut être utilisé par plusieurs instances de base de données et gérer des milliers de connexions de clients.



Le Listener est contrôlé grâce à Database Control ou à l'outil en ligne de commande `lsnrctl`.

Database Control permet de configurer les paramètres généraux du listener en utilisation comme la protection des mots de passe et la localisation des fichiers de logs.

Les administrateurs expérimentés pourront aussi configurer Oracle Net Services en éditant manuellement le fichier de configuration grâce à un éditeur de texte comme vi ou gedit sous Linux ou Notepad sous Windows.



7.1.2. Gestion du listener

7.1.2.1. Utilitaire de contrôle du processus d'écoute (lsnrctl)

L'utilitaire de control du listener permet de contrôler le listener. Grâce à **lsnrctl** vous pourrez :

- Démarrer le listener
- Arrêter le listener
- Voir le statut du listener
- Voir le statut des services du listener
- Configurer dynamiquement plusieurs listener
- Changer le mot de passe du listener

```
$ lsnrctl
```

```
LSNRCTL for Linux: Version 10.1.0.1.0 on 05-NOV-2003 13:27:51  
Copyright (c) 1991, 2003, Oracle. All rights reserved.  
Welcome to LSNRCTL, type "help" for information.
```

```
LSNRCTL> help
```

```
The following operations are available  
An asterisk (*) denotes a modifier or extended command:
```

| | | |
|-----------------|---------|--------|
| start | stop | status |
| services | version | reload |
| save_config | trace | spawn |
| change_password | quit | exit |
| set* | show* | |

7.1.2.2. Syntaxe du lsnrctl

La syntaxe basique en ligne de commande pour cet utilitaire est la suivante :

```
LSNRCTL> command [listener_name]
```

Quand la commande **lsnrctl** est utilisée seule, elle va agir sur le processus d'écoute par défaut nommé **LISTENER** à moins qu'un nom différent soit spécifié ou que la commande **SET LISTENER** soit exécutée. Les deux exemples ci-dessous auront les mêmes effets.

Nous pouvons utiliser la syntaxe en ligne de commande :

```
/home/oracle> lsnrctl start
```

Nous aussi utiliser le prompt :

```
/home/oracle> lsnrctl
```

```
LSNRCTL for Linux: Version 10.1.0.1.0 on 05-NOV-2003  
Copyright (c) 1991, 2003, Oracle. All rights reserved.  
Welcome to LSNRCTL, type "help" for information.
```

```
LSNRCTL> start
```

En général, la syntaxe par ligne de commande est utilisée pour exécuter des commandes à l'unité ou des scripts. Si vous prévoyez d'exécuter successivement plusieurs commandes la syntaxe avec le prompt sera plus pratique. Vous pouvez remarquer qu'aucun nom de processus d'écoute n'a été donné en argument lors du démarrage, donc la commande stop arrête le processus d'écoute nommé **LISTENER**. Si votre processus d'écoute est protégé par un mot de passe, la syntaxe avec le prompt sera utilisée.

Il faut se rappeler que si le processus d'écoute est nommé autrement que **LISTENER**, vous devez le spécifier avec toutes les commandes que vous allez exécutées ou à moins de le spécifier au début et de manière définitive grâce à la commande **SET CURRENT_LISTENER**.

Supposons que le processus d'écoute soit nommé **BACKUP**.

En utilisant la syntaxe du prompt nous aurons :

```
LSNRCTL> stop backup
```

```
Connecting to  
(DESCRIPTION= (ADDRESS= (PROTOCOL=TCP) (HOST=rhel) (PORT=5521)))  
The command completed successfully
```

On aura le même résultat en faisant :

```
LSNRCTL> set cur backup
```

```
Current Listener is backup
```

```
LSNRCTL> stop
```

```
Connecting to  
(DESCRIPTION= (ADDRESS= (PROTOCOL=TCP) (HOST=rhel) (PORT=5521)))  
The command completed successfully
```

Avec la syntaxe par ligne de commande nous allons avoir:

```
/home/oracle> lsnrctl stop backup
```

```
LSNRCTL for Linux: Version 10.1.0.1.0 on 05-NOV-2003 15:19:33  
Copyright (c) 1991, 2003, Oracle. All rights reserved.  
Connecting to  
(DESCRIPTION= (ADDRESS= (PROTOCOL=TCP) (HOST=rhel) (PORT=5521)))  
The command completed successfully
```

7.1.2.3. Surveillance avec le lsnrctl

La commande **lsnrctl services** donne le statut de tous les services Oracle Net que le processus doit gérer, qu'ils soient enregistrés statiquement et dynamiquement. Dans le résultat nous allons avoir les informations suivantes :

- Nom du service
- Le statut du service
- Le nombre de connexions établies ou refusées concernant les connexions dédiées ou partagées (les connexions partagées seront traitées dans le chapitre suivant).

La commande **lsnrctl status** retourne des informations détaillées sur le processus mais aussi un résumé des informations sur les services que ce processus d'écoute gère.

```
LSNRCTL> SERVICES
```

```
Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC) (KEY=EXTPROC)))
Services Summary...
Service "dba10g" has 1 instance(s).
  Instance "dba10g", status READY, has 1 handler(s) for this
  service...
  Handler(s):
    "DEDICATED" established: 12 refused: 0 state: ready
...
The command completed successfully
```

7.1.3. Création du listener

7.1.3.1. Création d'un listener

Chaque processus d'écoute doit avoir au moins une adresse d'écoute. Dans le panneau de configuration cliquez sur le bouton Ajouter Adresse pour renseigner l'adresse d'écoute. Sélectionnez le protocole réseau. TCP/IP est le plus utilisé mais aussi le protocole par défaut. Vous pouvez aussi utiliser IPC pour se connecter aux applications locales (résidant dans le serveur de base de données) ou les bibliothèques externes (EXTPROC), NMP pour les « pipelines » nommées et TCP/IP avec SSL. Entrez l'adresse IP du serveur sur lequel on retrouve le listener et le port d'écoute de ce listener. Le port d'écoute par défaut est le 1521. Si vous décidez de choisir un autre port, des configurations supplémentaires seront nécessaires pour le listener et l'instance.

Toutes les autres étapes de la configuration sont optionnelles. La seule étape obligatoire est la configuration de l'adresse d'écoute. Il faudra sélectionner le menu Fichier puis Enregistrer les configurations réseau. Pour créer un listener il suffit de cliquer sur Administration de Services Réseau à partir des liens de la page de propriétés du listener.

The screenshot illustrates the Oracle Enterprise Manager interface for creating a listener. It is divided into four numbered steps:

- Step 1:** The 'Administer Listeners' page is shown with the 'Go' button highlighted.
- Step 2:** The 'Listeners' table is displayed, showing a listener named 'LISTENER' with protocol 'TCP/IP', host 'c640.lowenthal.com', and port '1521'. The 'Create' button is highlighted.
- Step 3:** The 'Create Listener' dialog box is shown with the 'Listener Name' field containing 'BACKUP'.
- Step 4:** The 'Addresses' section of the dialog is shown, with the 'Add' button highlighted.

- Choisissez Listeners de la liste déroulante Administerer puis appuyez sur Go.
- Cliquez sur Créer
- Entrez le nom du listener. Ce nom doit être unique pour ce serveur.
- Ajouter l'adresse d'écoute.

7.1.3.2. Adresses d'écoute

Add Address

5

Cancel OK 8

Protocol TCP/IP

* Port 5561 6

* Host <hostname or IP address of server>
The host name or IP address of the computer.

7

Listeners: /u01/app/oracle/product/10.1.0/network/admin

Create

Edit Delete Actions Start/Stop 9

| Select | Name | Protocol Details | Status | Enterprise Manager Target |
|----------------------------------|----------|--|---------|-----------------------------|
| <input type="radio"/> | LISTENER | Protocol IPC Key EXTPROC Protocol TCP/IP Host c640.lowenthal.com Port 1521 | Started | LISTENER_c340.lowenthal.com |
| <input checked="" type="radio"/> | BACKUP | Protocol TCP/IP Host c640.lowenthal.com Port 5561 | Stopped | Not a target |

Chaque listener doit avoir au moins une adresse d'écoute. Pour cela :

- Sélectionnez le protocole réseau. TCP/IP est le plus utilisé et le protocole par défaut.
Remarque: Les protocoles NMP et EXTPROC sont configurés en utilisant l'onglet Autres Services.
- Si vous choisissez d'utiliser un autre port d'écoute il faudra faire les configurations supplémentaires.
- Entrez le nom et l'adresse IP du serveur dans lequel nous allons localiser le listener, et le port d'écoute.
- Toutes les autres configurations sont optionnelles. Cliquez sur OK pour sauvegarder vos changements.
- Pour démarrer le nouveau, sélectionnez Start/Stop de la liste déroulante Actions et appuyez sur OK

7.1.3.3. Configuration des paramètres optionnels

Vous pouvez aussi configurer les paramètres optionnels:

- **Logging:** dans le cas où vous voulez conserver les logs des connexions, indiquez aussi l'endroit où ces fichiers seront stockés.
- **Tracing:** utilisé dans le cas où vous voulez conserver des informations détaillées sur les connexions Oracle Net. Il faut spécifier dans ce cas la quantité d'information à collecter (niveau de trace) et l'endroit où ces informations seront stockées.

Edit Listener: BACKUP

Cancel OK

General Authentication **Logging & Tracing** Static Database Registration Other Services

Logging

Logging Disabled
 Logging Enabled

* Log File

Tracing

Tracing Disabled
 Tracing Enabled

Select a trace level.

User
Basic information useful for identifying user-induced error conditions

Administrator
Detailed information useful for identifying installation-specific problems

Support
Trace information useful for Oracle Support Services

* Trace File

7.1.4. Enregistrement d'une base de données

Pour qu'un listener transfère les connexions de clients à une instance, il doit connaître le nom et l'emplacement du répertoire **ORACLE_HOME** de l'instance. Le listener peut trouver les informations de deux manières :

- **L'abonnement dynamique** : les instances Oracle8i, Oracle9i et Oracle Database 10g vont s'abonner automatiquement au listener par défaut à leur démarrage. Le listener par défaut n'a besoin d'aucune configuration pour fonctionner.
- **L'abonnement statique** : Les anciennes versions d'Oracle Database ne s'enregistrent pas automatiquement auprès du listener et donc nécessitent un fichier de configuration pour le listener qui va contenir la liste des services de toutes les bases de données qu'il doit servir. Vous pourrez utiliser ce type d'abonnement avec les nouvelles versions d'Oracle dans les cas suivants :
 - Le port d'écoute du processus d'écoute n'est pas le port par défaut 1521.
 - Votre application a besoin obligatoirement d'un abonnement en statique. Pour configurer un abonnement statique, sélectionnez Static Database Registration dans le panneau d'édition du listener et appuyez sur Add Database button. Entrez le nom du service (qui est équivalent au nom global de la base de données **<DB_NAME>.<DB_DOMAIN>**), le chemin d'accès du répertoire **ORACLE_HOME**, et le **SID** (qui est équivalent au nom de l'instance). Cliquez sur Ok et redémarrer le listener pour que les changements prennent effet.

The image shows a sequence of three Oracle Enterprise Manager dialog boxes:

- Edit Listener: BACKUP**: The "Static Database Registration" tab is selected. It contains a table with columns "Select Service Name", "Oracle Home Directory", and "Oracle System Identifier (SID)". The table is currently empty, showing "(No items found.)". An "Add" button is visible in the bottom right corner.
- Add Database Service**: This dialog box has three input fields:
 - Service Name:
 - Oracle Home Directory:
 - Oracle System Identifier (SID):The "OK" button is highlighted.
- Edit Confirmation: BACKUP**: This dialog box contains the text: "If the listener is not stopped and restarted, the changes will not take effect. Choose whether or not you want the listener to be restarted." There are two radio buttons:
 - Restart
 - Don't Restart

7.2. Connexions réseau

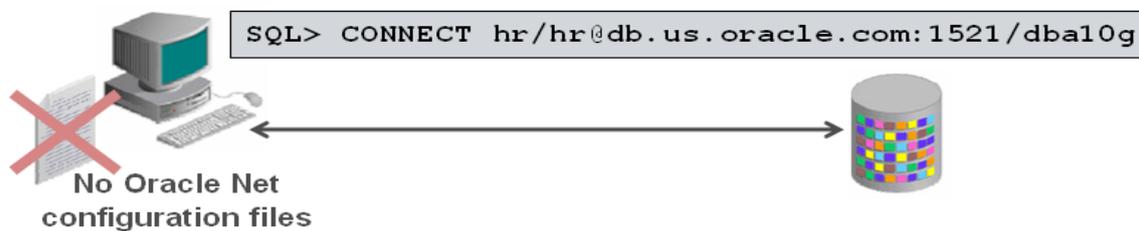
7.2.1. Introduction

Avant de se connecter à un service par l'intermédiaire du processus d'écoute Oracle Net, l'application doit avoir des informations sur ce service comme l'adresse et l'hôte sur lesquels résident le processus d'écoute, le protocole que le processus d'écoute utilise et le port d'écoute. Après que le processus d'écoute soit localisé la dernière information que l'application aura besoin d'avoir sera le nom du service auquel il veut se connecter.

Le processus utilisé pour déterminer les informations sur la connexion est appelé Résolution de Nom. Oracle Net supporte plusieurs méthodes pour résoudre les informations de connexions.

7.2.2. Méthodes de connexion

7.2.2.1. Easy Connect



Avec Easy Connect c'est l'utilisateur qui fournit toutes les informations nécessaires pour la connexion Oracle Net grâce à la chaîne de connexion. La chaîne de connexion d'Easy Connect aura le format suivant :

<username>/<password>@<hostname>:<listener port>/<service name>

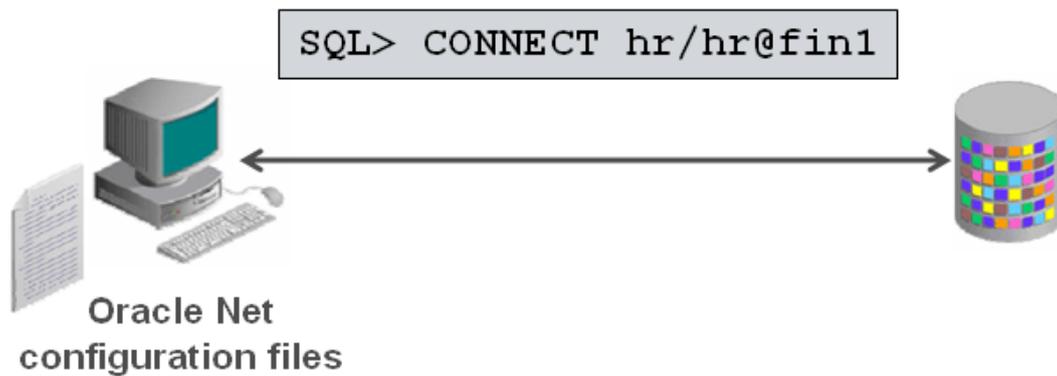
Le port du processus d'écoute et le nom du service sont optionnels. Si le port du processus d'écoute n'est pas fourni, ce sera le port par défaut d'Oracle Net **1521** qui sera utilisé. Si le nom du service n'est pas fourni, Oracle Net utilisera le nom d'hôte donné dans la chaîne de connexion comme le nom du service de base de données.

Si le processus d'écoute utilise le protocole TCP et écoute sur le port 1521 et que les paramètres d'instance **SERVICE_NAMES=db, DB_DOMAIN=us.oracle.com**, alors la chaîne ci-dessous pourra être utilisée :

```
connect hr/hr@db.us.oracle.com
```

Remarque: Le paramètre **SERVICE_NAMES** peut accepter plusieurs valeurs qui seront séparées par des virgules. Pour que la chaîne de connexion ci-dessous marche il faut que la valeur **db** soit présente dans la liste de valeurs.

7.2.2.2. Résolution locale (Local Naming)

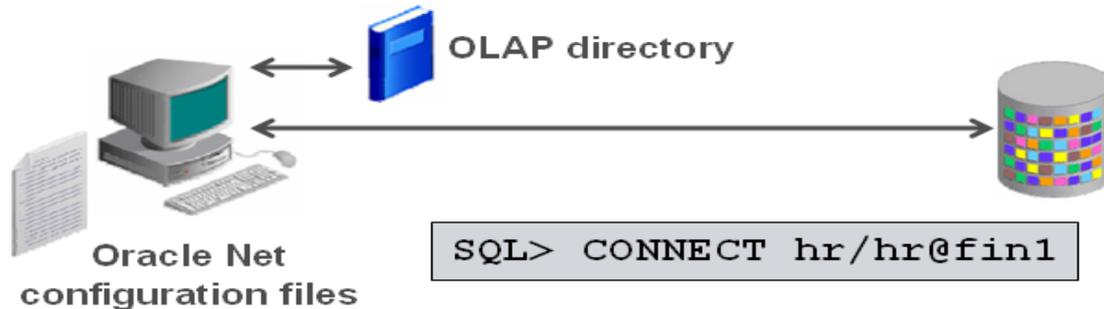


Avec la résolution locale, l'utilisateur aura besoin de l'alias du service Oracle Net. Oracle Net vérifiera alors que l'alias est présente dans la liste de services connus et si c'est le cas l'alias sera résolu afin d'obtenir le nom d'hôte, le protocole, le port et le nom du service.

Parmi les avantages de la résolution locale on peut citer le fait que les utilisateurs n'auront besoin d'avoir que l'alias et non une chaîne de connexion plutôt complexe et obligatoire pour Easy Connect. La liste des services connus est stockée dans un fichier de configuration texte :

\$ORACLE_HOME/network/admin/tnsnames.ora. La résolution locale est appropriée pour les organisations dont les configurations d'Oracle Net service ne changent pas souvent.

7.2.2.3. Résolution par répertoire (Directory Naming)



Avec la résolution par répertoire, l'utilisateur n'aura besoin que d'un alias du service Oracle Net. Oracle Net va résoudre l'alias grâce à une liste de services qui ne sera pas stockée en local mais dans un annuaire LDAP, cette résolution va permettre de résoudre l'alias afin d'obtenir le nom d'hôte, le protocole, le port et le nom du service.

Comme pour la résolution en local les utilisateurs auront besoin que de l'alias au lieu de la chaîne de connexion plutôt longue d'Easy Connect.

L'avantage de la résolution par répertoire est que dès qu'un nouveau service est ajouté au répertoire LDAP, il sera disponible pour tous les utilisateurs. Avec la résolution en local le DBA est obligé de mettre à jour individuellement tous les fichiers **tnsnames.ora** si les informations d'un service venaient à changer.

La résolution par répertoire est appropriée pour les organisations dont la configuration Oracle Net change souvent.

7.2.3. Configuration des connexions

7.2.3.1. Configuration les alias pour les services

Pour créer un alias pour un service Oracle Net, sélectionnez Local Naming de la liste déroulante Administer et puis cliquer sur Go ensuite il faudra cliquer sur Create.

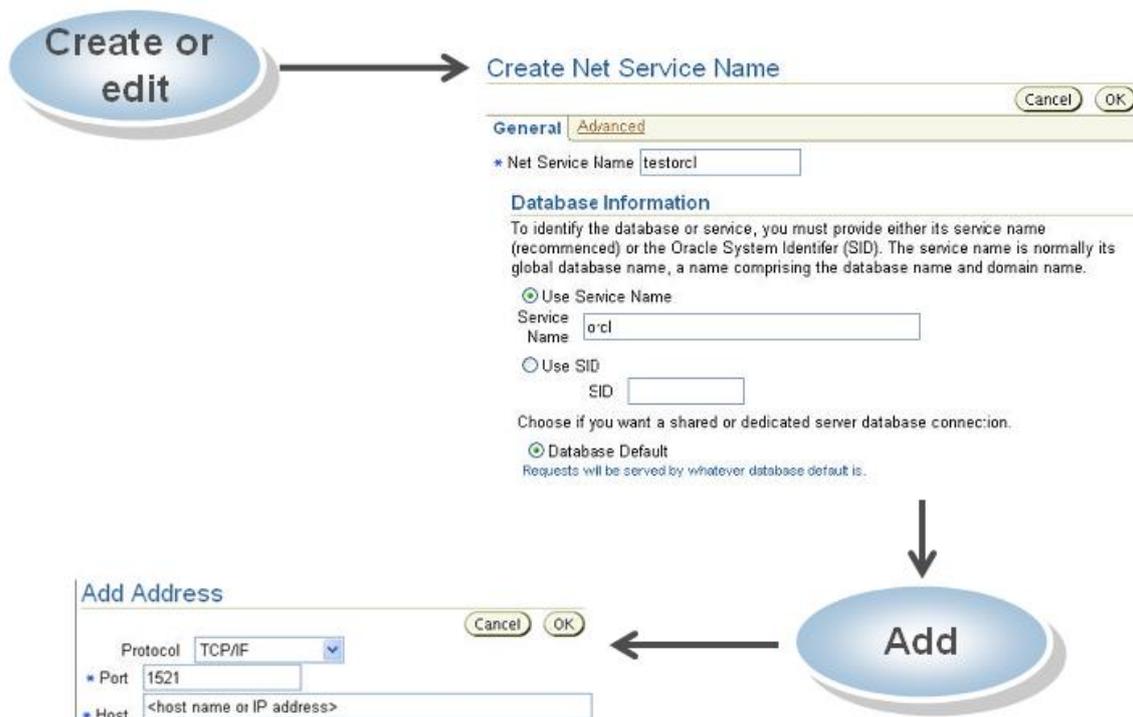
Pour configurer un alias pour une résolution par répertoire, il faudra donc choisir Directory Naming au lieu de Local Naming.

Remarque : Si la résolution par répertoire n'est pas configurée, vous ne pourrez pas sélectionner l'option Directory Naming.

La résolution par répertoire est approfondie dans le cours Oracle Enterprise Identity Management mais aussi dans le manuel Oracle Advanced Security Administration.

Dans la page de création de nom de service réseau, il faudra saisir un nom qui sera unique dans le champ Nom de service réseau (ce nom correspondra à l'alias). Entrez ensuite le nom du service ou le SID de la base de données à laquelle vous désirez vous connecter puis appuyez sur le bouton ajouter pour entrer l'adresse du service réseau.

Concernant l'adresse il faudra donner le nom du protocole à utiliser, le port et l'hôte utilisé par le processus d'écoute que le service auquel vous désirez vous connecter utilise.



7.2.3.2. Options avancées

| Select | Protocol | Protocol Details |
|----------------------------------|----------|--|
| <input checked="" type="radio"/> | TCP/IP | Host 192.168.2.62 Port 1521 |
| <input type="radio"/> | TCP/IP | Host 192.168.2.62 Port 5561 |

Connect-time Failover and Client Load Balancing

Configure whether addresses are tried randomly or sequentially during connections to the service. This setting is applicable only if there are more than one addresses configured.

- Try each address, in order, until one succeeds
- Try each address randomly, until one succeeds
- Try one address, selected at random
- Use each address in order until destination is reached
- Use only the first address

Avec les options de connexion avancées, Oracle Net peut faire profiter des avantages du failover et du load balancing.

Si l'option failover est activée, l'alias aura deux processus d'écoute ou plus. Dans le cas où le premier processus d'écoute ne serait pas disponible ce sera le second qui serait utilisé. Oracle Net essaiera d'utiliser les processus d'écoute dans l'ordre qui sera spécifié et il essaiera d'utiliser le processus suivant que si le premier qu'il essaie d'utiliser ne répond pas.

Si le load balancing est activé, Oracle Net utilisera un service en choisissant de manière aléatoire son adresse dans la liste des adresses. Dans un environnement Real Application Cluster le travail d'une base de données sera effectué par une multitude d'instances. Source routing est utilisée avec Oracle Connexion Manager. Oracle Connexion Manager permet de configurer un serveur proxy pour le trafic Oracle Net, et permet au trafic Oracle Net d'être routé avec sécurité grâce à un firewall. Oracle Net va traiter les adresses comme une liste de relais, les requêtes vont être envoyées à la première adresse qui va s'occuper de l'envoyer à la deuxième et la requête va être ainsi propagée pour atteindre la destination voulue. La différence avec le failover et le load balancing c'est qu'avec Source routing toutes les adresses peuvent être utilisées lors d'une requête.

| Option | Fonctionnalité avancée |
|---|----------------------------|
| Essaie chaque adresse dans l'ordre à jusqu'à ce qu'une n'ait réussi. | Failover |
| Essaie chaque adresse de manière aléatoire jusqu'à ce qu'une n'ait réussi. | Failover Load balancing |
| Essaie une adresse sélectionnée au hasard | Load balancing |
| Utilise chaque adresse dans l'ordre jusqu'à ce la destination soit atteinte | Source routing |
| Utilise seulement les premières adresses | Aucune |

7.3. Oracle Net Manager

7.3.1. Présentation

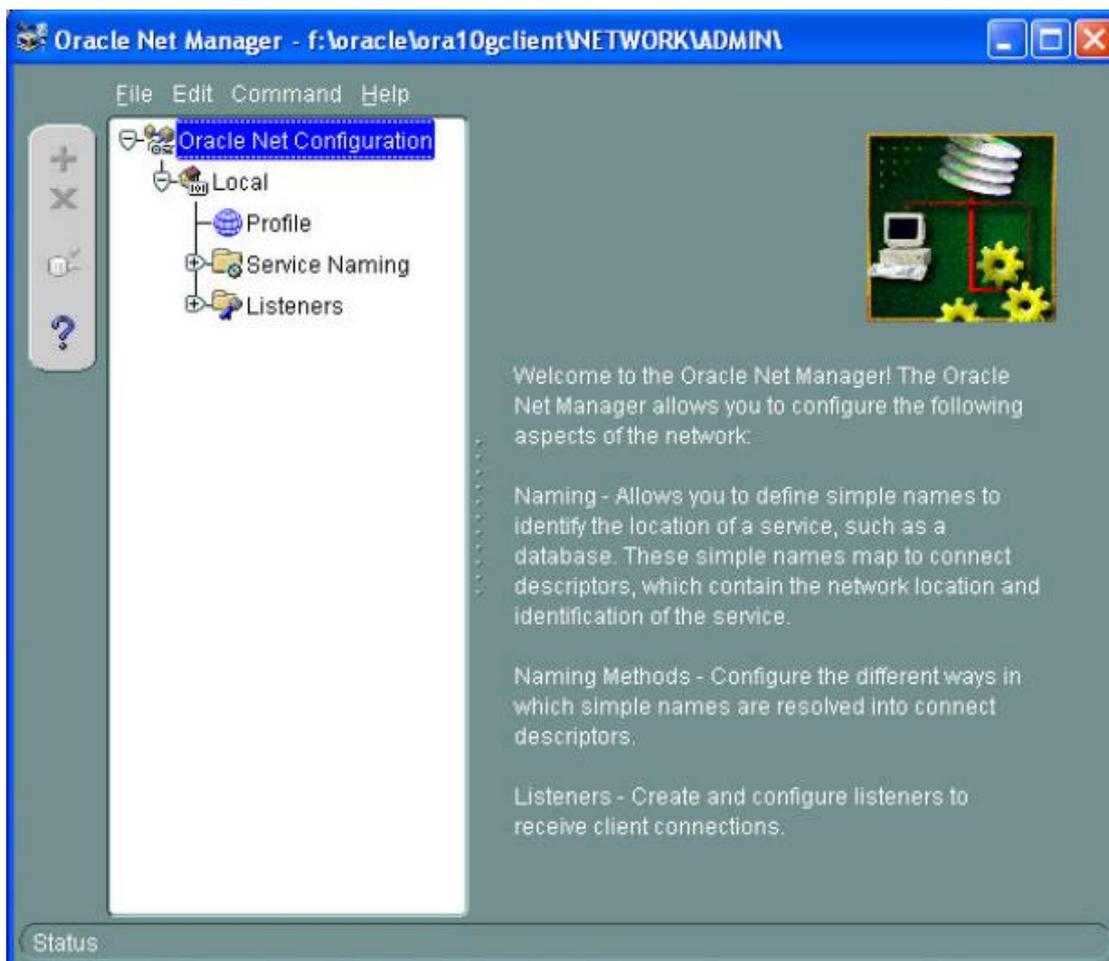
7.3.1.1. Généralités

Enterprise Manager Database Control permet de configurer Oracle Net du côté du serveur, mais ne permet pas de le faire au niveau du client ou du serveur d'application. Oracle Net Manager est un outil graphique qui permet de configurer les connexions Oracle Net au niveau du client. Ainsi vous pourrez gérer :

- Le nom des services Oracle Net.
- Les profils Oracle Net
 - La sécurité
 - Le troubleshooting (logging et la conservation des traces)
 - Les méthodes de résolution de noms.

Oracle Net Manager peut aussi être utilisé pour créer et configurer les processus d'écoute Oracle Net mais aussi de configurer la résolution de nom du service LDAP (Lightweight Directory Access Protocol). Pour démarrer Oracle Net Manager il suffit d'exécuter la commande **run netmgr**.

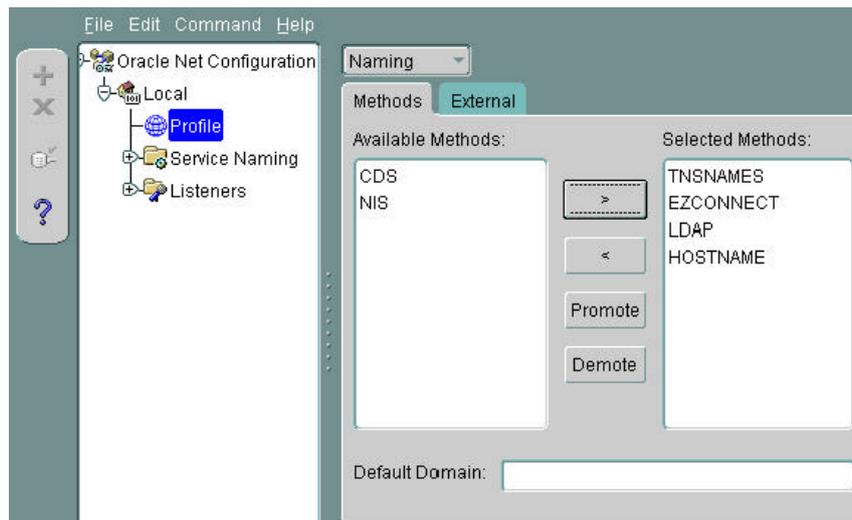
Lorsque vous allez ouvrir Oracle Net Manager, vous allez voir un arbre de navigation à votre gauche et la fenêtre des détails à votre droite. Vous pouvez consulter les options de chacune en dépliant le contenu grâce aux symboles + et -.



7.3.1.2. Choisir une méthode de résolution

Les méthodes de résolution disponibles par défaut sont la résolution locale (**TNSNAMES**), Easy connecta (**EZCONNECT**) et la résolution locale (prédécesseur moins performant de Easy Connect).

Avec la configuration montrée ci-dessous, Oracle Net va d'abord examiner les fichiers de configuration de la résolution locale, essayer d'utiliser Easy Connect, vérifier le serveur LDAP, et enfin essayer d'utiliser la résolution par nom d'hôte. Dans le cas précédent Oracle Net essayer d'utiliser la méthode de résolution suivante que si la méthode précédente échoue. Si vous voulez qu'un service soit utilisé avant un autre sélectionnez le service en question dans la liste et appuyez sur le bouton Promouvoir.



7.3.2. Configurations

7.3.2.1. Configuration les alias pour les services

Vous pouvez aussi utiliser Oracle Net Manager pour ajouter des alias de service Oracle Net pour la résolution locale ou la résolution par répertoire. Pour la résolution locale, dépliez le nœud Local et sélectionnez Service Names. Ensuite vous pourrez cliquer sur l'icône + (en haut à gauche) ou sélectionnez le menu en sélectionnant Edit puis Create. Une nouvelle fenêtre de configuration va s'ouvrir, où vous allez choisir le nom de l'alias, puis le protocole, ensuite l'adresse et le port en enfin le nom actuel du service que le processus d'arrière doit servir.

Pour configurer l'alias de service pour une résolution par répertoire, dépliez le nœud Directory au lieu du nœud Local.

Après avoir créé votre alias vous pouvez tester la connexion. Contrairement à Database Control, Oracle Net Manager va tenter de se connecter au service avec le nom d'utilisateur SCOTT et le mot de passe TIGER. Par ailleurs vous connecter avec des informations correctes en cliquant sur le bouton Change Login.

The image displays four sequential screenshots of the Oracle Net Manager configuration wizard, numbered 1 through 4, illustrating the steps to create a net service name.

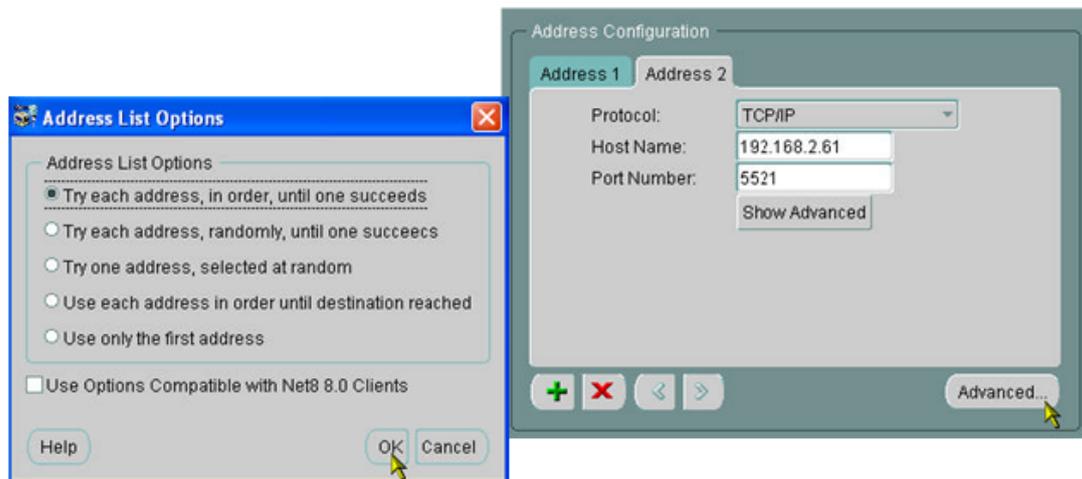
Step 1: The first screenshot shows the initial configuration screen. It instructs the user to enter a net service name. The text reads: "To access an Oracle database, or other service, across the network you use a net service name. This wizard will help you create a net service name. Enter the name you want to use to access the database or service. It can be any name you choose." The "Net Service Name" field contains the value "foo".

Step 2: The second screenshot shows the protocol selection screen. It asks the user to select a network protocol. The text reads: "To communicate with the database across a network, a network protocol is used. Select the protocol used for the database you want to access." The "TCP/IP (Internet Protocol)" option is selected.

Step 3: The third screenshot shows the host name and port number configuration screen. It instructs the user to enter the TCP/IP host name and port number. The text reads: "To communicate with the database using the TCP/IP protocol, the database computer's host name is required. Enter the TCP/IP host name for the computer where the database is located." The "Host Name" field contains "db.us.oracle.com" and the "Port Number" field contains "1521".

Step 4: The fourth screenshot shows the service name configuration screen. It asks the user to identify the database or service. The text reads: "To identify the database or service you must provide either its service name, for Oracle8i 8.1 or later, or system identifier (SID), for Oracle8 8.0 database versions. The service name for an Oracle8i or later database is normally its global database name." The "Service Name" field contains "dba10g" and the "SID" field contains "ORCL". The "Connection Type" is set to "Database Default".

7.3.2.2. Options avancées



Oracle Net Manager permet de sélectionner les options avancées telles que le failover ou le load balancing ou bien source routing pour Oracle Connexion Manager

| Options | Fonctionnalités avancées |
|--|----------------------------|
| Essayer chaque adresse, dans l'ordre, jusqu'au succès | Failover |
| Essayer chaque adresse, aléatoirement, jusqu'au succès | Failover Load balancing |
| Essayer une adresse, sélectionnée aléatoirement | Load balancing |
| Utilisez chaque adresse dans l'ordre jusqu'à atteindre le destinataire | Source routing |
| Utilisez seulement la première adresse | None |

7.3.3. Oracle Net Connectivity

tnsping est l'équivalent de l'utilitaire ping de TCP/IP. Il offre un test rapide qui permet de vérifier qu'une connexion réseau vers une destination donnée fonctionne correctement.

```
tnsping db.us.oracle.com:1521/dba10g
```

```
TNS Ping Utility for Linux: Version 10.1.0.1.0 - Production on 06-NOV-20
Copyright (c) 1997, 2003, Oracle. All rights reserved.
Used parameter files:
 /oracle/product/ora10g/network/admin/sqlnet.ora
Used EZCONNECT adapter to resolve the alias
Attempting to contact
 (DESCRIPTION=
  (CONNECT_DATA=(SERVICE_NAME=dba10g))
  (ADDRESS=(PROTOCOL=TCP)(HOST=192.168.2.61)(PORT=1521)))
OK (380 msec)
```

Cet utilitaire permet de vérifier qu'un nom d'hôte, un port, et un protocole donnés permettent d'accéder à un processus d'écoute. Actuellement le **tnsping** ne vérifie pas le service dont le nom est spécifié est bien abonné au listener. Un autre avantage du **tnsping** c'est le fait de donner l'emplacement des fichiers de configuration ce qui peut être intéressant pour un système avec plusieurs répertoires **ORACLE_HOME**.

8. Module 08 : Serveur Partagé

8.1. Les sessions

8.1.1. Définition

Une fois la résolution de nom soit effectuée, une requête de connexion est envoyée par l'utilisateur ou l'application middle-tiers vers le processus d'écoute Oracle Net.

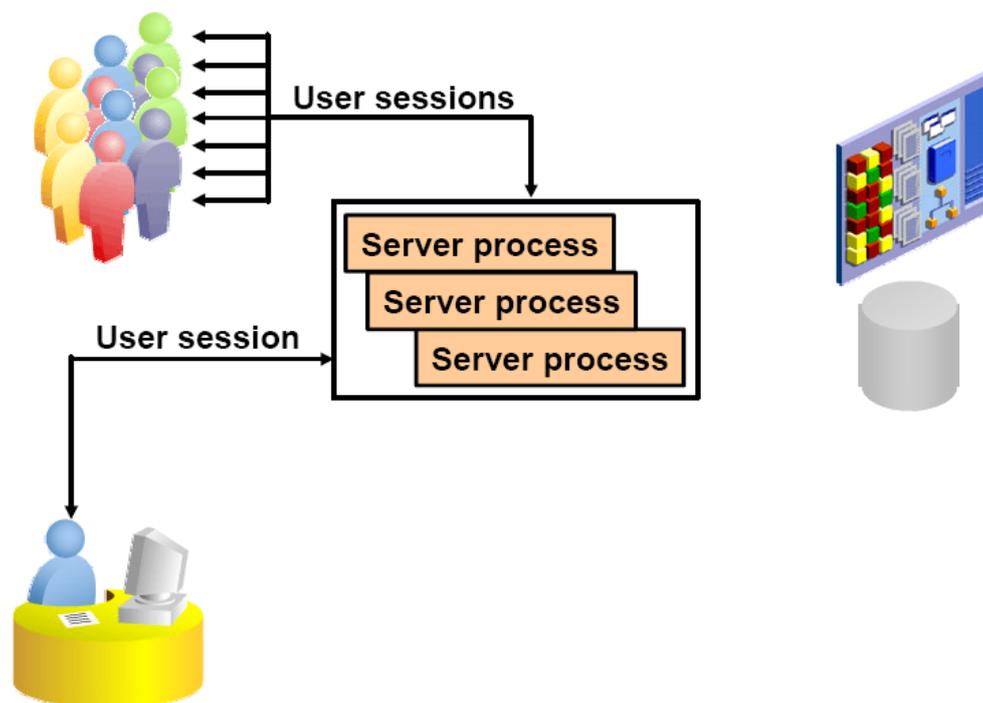
Le processus d'écoute (listener) va donc recevoir un paquet **CONNECT** et va vérifier que ce paquet demande un nom de service Oracle Net valide.

Si aucun nom de service n'est pas concerné par la requête (cas d'une requête **TNSPING**), le processus d'écoute ignore la requête de connexion et ne fait aucune action supplémentaire. Si le nom du service est invalide, le processus d'écoute transmet un code d'erreur au processus utilisateur.

Après que la session utilisateur ait reçu le paquet **RESEND**, il va renvoyer un paquet **CONNECT**. Le processus serveur va vérifier les paramètres d'authentification de l'utilisateur (généralement le mot de passe) et s'ils sont valides, la session sera créée.

Les processus serveur vont aussi réserver de la mémoire pour les tâches spéciales comme les indexes bitmaps ou les jointures. La quantité de mémoire consommée par les processus serveur dédiés dépend de plusieurs paramètres d'initialisation. Il peut être contrôlé automatiquement grâce aux paramètres **PGA_AGGREGATE_TARGET** et **WORKAREA_SIZE_POLICY** ou grâce à l'utilisation de tuning avancé.

8.1.2. Processus serveur dédié



Si le paquet **CONNECT** fait référence à un nom de service valide, le processus d'écoute va créer un nouveau processus qui va gérer la connexion. Ce nouveau processus sera appelé « processus serveur » mais peut aussi être nommé « shadow process ». Après que le processus serveur soit créé, le processus d'écoute va se connecter à lui pour lui donner les informations d'initialisation comportant les informations sur l'adresse du processus utilisateur.

A partir de cette étape le listener ne va plus intervenir dans la communication entre le processus utilisateur et le serveur Oracle. Le processus serveur va transmettre un paquet **RESEND** au processus utilisateur.

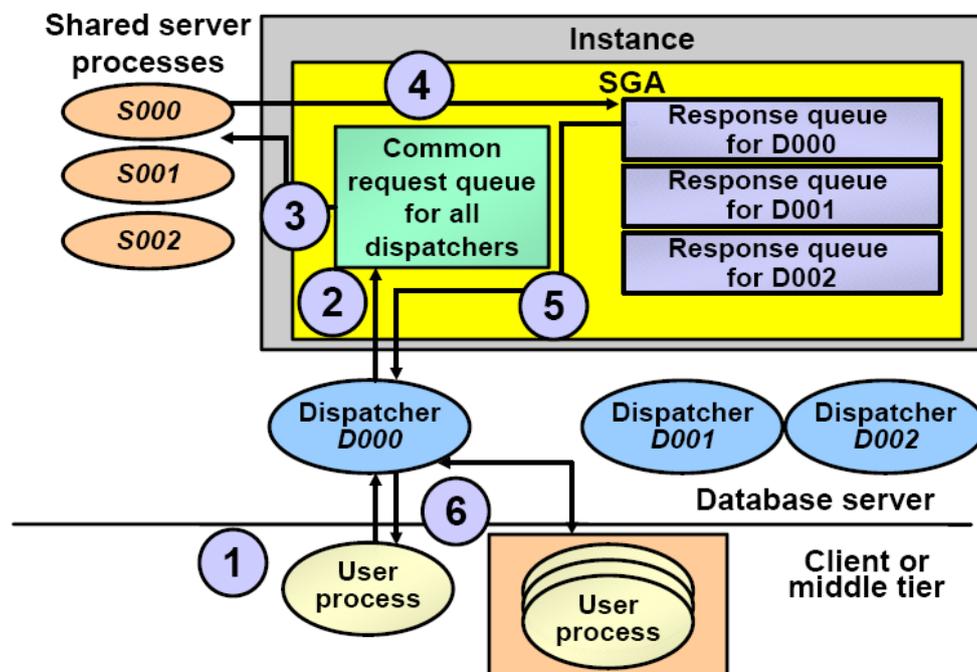
Avec le principe du processus serveur dédié, un processus va être utilisé exclusivement pour traiter les requêtes d'un processus utilisateur. Chaque processus serveur consomme des ressources système comprenant des cycles CPU et de la mémoire.

Dans un système avec une charge de travail importante, la mémoire et les ressources CPU consommé par le processus serveur dédié peuvent affecter la stabilité de ce système. Dans ce cas vous aurez deux possibilités pour régler ce problème :

- Augmenter les ressources système en ajoutant plus de mémoire et de la capacité CPU.
- Utiliser l'architecture serveur partagé d'Oracle.

8.1.3. Processus serveur partagé

Chaque service participant à l'architecture Serveur Partagé dispose au moins d'un dispatcher. Dans cette architecture, lorsqu'une requête de connexion arrive, le processus d'écoute ne va pas directement lui attribuer un processus serveur dédié. En effet, le processus d'écoute va conserver une liste de dispatchers disponibles pour chaque service, il va aussi conserver le nombre de connexions concurrentes pour chaque dispatcher. Les requêtes de connexion seront renvoyées vers un dispatcher qui traite les requêtes pour un service donné. Contrairement à un processus serveur dédié, un dispatcher peut gérer des milliers de connexions utilisateur. Les dispatchers n'effectuent pas les tâches demandés par les requêtes utilisateur, ils vont s'occuper de mettre les requêtes dans une **queue** de demande commune localisée dans une portion du shared pool de la **SGA**. Plusieurs structures de la mémoire généralement stockées dans la **PGA** (Program Global Area) seront stockées dans une zone mémoire partagée du serveur, parce que les requêtes d'une même session seront traitées par des processus serveur partagés différents. Ces structures mémorielles seront stockées dans la plupart du cas dans le large pool.



Lorsqu'un utilisateur connecté à une architecture de serveur partagé envoie une requête:

1. Le processus utilisateur va transmettre sa requête à son dispatcher.
2. Le dispatcher va placer sa requête dans une **liste** de demande commune présente dans la **SGA**.
3. Le premier processus serveur partagé disponible va récupérer la requête de la queue et va effectuer le travail demandé
4. Le processus serveur partagé va placer la réponse dans la queue de réponse du dispatcher responsable de la requête. On pourra noter que chaque dispatcher aura sa propre queue pour les réponses.
5. Le dispatcher va récupérer la réponse de sa queue de réponse.
6. Le dispatcher va retourner la réponse à l'utilisateur.

Une fois la requête de l'utilisateur traitée, le processus serveur partagé sera libre et sera donc disponible pour exécuter une autre requête de n'importe quel autre utilisateur.

Queue des requêtes :

- Cette queue sera partagée par tous les dispatchers
- Les processus serveurs partagés vont toujours vérifier la présence de nouvelles requêtes dans la queue.
- Les requêtes seront traitées en se basant sur le principe du premier arrivé premier servi (FIFO). Il n'existe pas de priorités.

8.1.4. SGA et PGA

Le contenu de la **SGA** (System Global Area) et de la **PGA** (Program Global Area) diffère en fonction du type d'architecture serveur utilisé (dédié ou partagé). Ce contenu concerne :

- Le texte et les formulaires analysés pour toutes les requêtes SQL stockées dans la **SGA**.
- L'état des curseurs qui va contenir les résultats en temps réel des requêtes
- Les données des sessions utilisateurs comme les informations sur la sécurité et l'usage des ressources.
- La pile qui contient les variables locales nécessaires pour les traitements.

Les changements dans la **SGA** et la **PGA** seront transparents pour l'utilisateur, mais si vous avez beaucoup d'utilisateurs il faut penser à augmenter la valeur du paramètre **LARGE_POOL_SIZE**. Chaque processus serveur partagé aura besoin d'avoir accès aux données de toutes les sessions afin de pouvoir traiter leurs requêtes. Vous pouvez contrôler la quantité de mémoire allouée à chaque session grâce au paramètre d'initialisation **PRIVATE_SGA**.

Dans le cas d'un serveur dédié les données de la session de l'utilisateur étaient stockées dans le **PGA**. Dans ce cas la **SGA** contient le shared pool et les autres structures mémorielles, le **PGA** va contenir la pile mémoire, les données de la session de l'utilisateur et l'état des curseurs.

Dans le cas d'un serveur partagé, les données de la session de l'utilisateur seront stockées dans la **SGA**. Dans ce cas la **SGA** contient le shared pool, les autres structures mémorielles, les données de la session et l'état des curseurs. Le **PGA** va contenir seulement la pile mémoire.

| Espace Mémoire | Serveur dédié | Serveur Partagé |
|---|---------------|-----------------|
| Nature de la mémoire pour les sessions | Privé | Partagé |
| Localisation des données persistantes | PGA | SGA |
| Localisation des résultats en temps réel pour les requêtes SELECT | PGA | PGA |
| Localisation des résultats en temps réel pour les requêtes DML/DDDL | PGA | PGA |

Différences d'allocation de la mémoire entre serveurs dédiés et serveurs partagés

8.2. Configuration d'un serveur partagé

Pour configurer Oracle Shared Server, vous devez éditer les paramètres d'initialisation de votre instance. La plupart des paramètres optionnels ont des valeurs par défaut correctes. Dans plusieurs systèmes, le seul paramètre qui doit être configuré est **DISPATCHERS**.

En fonction des options sélectionnées lors de la création de la base de données, le paramètre **DISPATCHERS** pourrait déjà être configuré pour démarrer un dispatcher servant la base de données XML. Le paramètre **DISPATCHER** accepte plusieurs valeurs qui seront entre apostrophes et séparées par une virgule : '*<Première valeur>*', '*<deuxième valeur>*'

Ci-dessous nous verrons les paramètres d'initialisation utilisés pour configurer un serveur partagé :

8.2.1. DISPATCHERS

Lorsqu'un utilisateur effectue une requête, cette dernière est placée dans la queue des requêtes, le premier processus serveur partagé disponible va alors récupérer cette requête pour la traiter. La queue des requêtes est présente dans la **SGA** et est commune à tous les dispatchers d'une instance. Le processus serveur partagé va vérifier la présence de nouvelles requêtes dans la queue commune, si c'est le cas, ces requêtes seront traitées en se basant sur le principe du premier arrivé premier servi. Un processus serveur partagé va récupérer une requête dans la queue et va ensuite faire tous les appels nécessaires vers la base de données afin de traiter cette requête.

Lorsque le processus serveur va finir le traitement de la requête, il va placer le résultat dans la queue de réponse du dispatcher qui gère la requête. Chaque dispatcher aura sa propre queue de réponse dans la **SGA**. Le dispatcher lorsqu'il recevra la réponse va retourner le traitement demandé au processus utilisateur approprié.

Le paramètre **DISPATCHERS** permet à chaque groupe dispatcher d'avoir des attributs variés. Oracle Database 10g supporte une syntaxe clé-valeur (similaire à la syntaxe utilisée par Oracle Net Services) qui va permettre la spécification de nouvelles attributs ou d'attributs existants. Lors de la spécification des attributs ni la position ni la casse ne sont prises en compte.

Exemple :

```
DISPATCHERS=' (PROTOCOL=TCP) (DISPATCHERS=3) '
```

| | |
|---------------------|--|
| Type de paramètre | Chaîne de caractère (spécifiée entre apostrophe) |
| Classe du paramètre | Dynamique |
| Valeur par défaut | Null (aucun dispatcher ne sera démarré) |

Bien que le nombre de connexions qu'un dispatcher puisse gérer varie énormément en fonction des contextes, une solution correcte serait d'utiliser un dispatcher pour chaque vingt-cinq nouvelle connexion à la base de données.

Le seul attribut obligatoire est **PROTOCOL**. Tous les autres sont optionnels. Dans ce tableau ci-dessous nous avons les attributs pour le paramètre **DISPATCHERS**. Ces attributs peuvent être abrégés en trois lettres à la place du nom entier du paramètre.

| Attribut | Description |
|------------------------------|---|
| PROTOCOL (PRO ou PROT) | Spécifie le protocole réseau pour lequel le dispatcher va générer un point d'écoute (généralement TCP). |
| DISPATCHERS (DIS ou DISP) | Le nombre initial de dispatchers à démarrer (par défaut 1) |
| SERVICE (SER ou SERV) | Le nom de service Oracle Net auquel le dispatcher va s'abonner avec le processus d'écoute. Si cet attribut n'est pas renseigné le dispatcher va s'abonner au service dont le nom est contenu dans le paramètre SERVICES_NAMES |
| LISTENER (LIS ou LIST) | Spécifie le nom du processus d'écoute (listener) avec lequel le processus PMON va récupérer les informations sur le dispatcher. L'alias est le même que celui donné lors de la configuration de la résolution de nom. Cet attribut sera spécifié si le processus d'écoute est un processus d'écoute local qui n'utilise pas le port par défaut (1521) et n'est pas spécifié avec le paramètre LOCAL_LISTENER ou bien le processus d'écoute est dans un autre nœud. |
| SESSIONS (SES ou SESS) | Le nombre maximum de sessions réseau pour chaque dispatcher. La valeur par défaut est spécifique au système d'exploitation. Dans la plupart des cas elle est à 16K. |
| CONNECTIONS (CON ou CONN) | Spécifie le nombre maximum de connexions réseau autorisées pour chaque dispatcher. La valeur par défaut est spécifique au système d'exploitation. Par exemple elle est de 1024 pour Sun Solaris et Windows. |

8.2.2. SHARED_SERVERS

SHARED_SERVERS spécifie le nombre minimum de processus serveur qui sont actifs. Le renseignement de ce paramètre n'est pas indispensable car l'instance va contrôler la queue des requêtes et va démarrer ou arrêter automatiquement des processus serveur en fonction de la demande.

| | |
|---------------------|--|
| Type de paramètre | Entier |
| Classe du paramètre | Dynamique |
| Valeur par défaut | 0 si DISPATCHERS est à NULL, 1 si DISPATCHERS a une autre valeur |
| Plage des valeurs | Dépend du système d'exploitation |

Une bonne mesure pour ce paramètre serait d'utiliser un processus serveur partagé pour vingt-cinq connexions.

8.2.3. MAX_SHARED_SERVERS

MAX_SHARED_SERVERS spécifie le nombre maximum de processus serveur partagés autorisés à fonctionner au même moment. Donner une valeur à ce paramètre est important parce que par défaut l'instance créera de manière automatique de nouveaux processus serveur en fonction des besoins.

| | |
|---------------------|----------------------------------|
| Type de paramètre | Entier |
| Classe du paramètre | Dynamique |
| Valeur par défaut | None (pas de limitation) |
| Plage de valeurs | Dépend du système d'exploitation |

En général, ce paramètre correspond au nombre de processus serveur partagés lorsque le serveur est en pleine utilisation. Il faut donc expérimenter cette valeur tout en contrôlant les processus serveur afin de déterminer la valeur idéale pour ce paramètre. Pour trouver le nombre de processus serveur vous pourrez interroger la vue de dictionnaire de données **V\$SHARED_SERVER_MONITOR**.

8.2.4. CIRCUITS

Les circuits virtuels correspondent aux connexions utilisateur à la base de données via les dispatchers et les processus serveur. Le paramètre **CIRCUITS** spécifie le nombre total de circuits virtuels disponibles pour les connexions réseau.

| | |
|---------------------|---|
| Type de paramètre | Entier |
| Classe du paramètre | Dynamique |
| Valeur par défaut | Si Oracle Shared Server est configure, alors le paramètre CIRCUITS aura la même valeur que SESSIONS , sinon la valeur sera 0. |

Ce paramètre aura une valeur seulement si vous voulez imposer une limite sur le nombre total de connexion qu'un utilisateur pourra utiliser sur une architecture serveur partagé.

8.2.5. SHARED_SERVER_SESSIONS

Ce paramètre contrôle le nombre total de sessions ouvertes en même temps à n'importe quel instant. Ce paramètre va permettre de contrôler la quantité de ressources qui sera réservée aux utilisateurs utilisant une connexion dédié.

| | |
|---------------------|--------------------------|
| Type de paramètre | Entier |
| Classe du paramètre | Dynamique |
| Valeur par défaut | None (pas de limitation) |

L'architecture Serveur Partagé élimine le besoin d'avoir un processus serveur dédié à chaque connexion. Le dispatcher va rediriger les requêtes des sessions réseaux vers un ensemble de processus serveur. Un processus serveur va récupérer la requête de la queue de requête commune et va faire le traitement. Ceci montre qu'un nombre limité de processus serveur partagés peut effectuer une quantité de travail équivalente à celle effectuée par plusieurs processus serveur dédiés. Aussi vu que la quantité de mémoire nécessaire à chaque utilisateur est relativement faible, moins de mémoire et moins de gestion des processus seront nécessaires et ainsi plus d'utilisateurs seront supportés.

Les autres paramètres affectant Oracle Shared Server et nécessitant d'être contrôlés sont :

- **LARGE_POOL_SIZE** qui permet de spécifier la taille en octets du large pool. Oracle Shared Server utilise le large pool pour stocker les informations sur la session qui seront dans la **PGA** dans le cas d'une connexion avec un processus serveur dédié.
- **SESSIONS** spécifie le nombre maximum de sessions qui peuvent être créées. Avec Oracle Shared Server le système va supporter encore plus de sessions.
- **PROCESSUS** qui contrôle le nombre de processus au niveau du serveur.
- **LOCAL_LISTENER** qui définit le port et le protocole utilisés par les processus d'écoute. Si le processus d'écoute n'utilise pas TCP/IP avec le port par défaut 1521, ou que vous avez plusieurs processus d'écoute, alors vous allez devoir configurer le paramètre **LOCAL_LISTENER** afin que les dispatchers puissent s'abonner aux processus d'écoute.

Si vous ne donnez pas de valeur au paramètre **LARGE_POOL_SIZE**, alors Oracle va utiliser le Shared Pool pour allouer la mémoire nécessaire pour les sessions utilisateur. Cela peut dégrader les performances du PL/SQL, du SQL ou bien des autres services qui utilisent aussi le Shared pool. Oracle allouera des quantités de mémoires fixes (environ 10K) du Shared Pool pour chaque session configurée, à moins que le large pool soit configuré.

8.3. Maintenir un serveur partagé

8.3.1. Vérifier la configuration d'un serveur partagé

Quant Oracle Shared Server est utilisé, vous devrez d'abord démarrer le processus d'écoute, ensuite la base de données, ainsi les dispatchers pourront immédiatement s'enregistrer avec le processus d'écoute. Si vous redémarrez le processus d'écoute, il faudra attendre un moment (environ une minute) afin que les services se réenregistrent. Pour vérifier que l'abonnement des services a été effectué vous pourrez utiliser la commande ci-dessous :

```
$ lsnrctl services
Service "TEST" has 1 instance(s).
Instance "TEST", status READY, has 3 handler(s) for this service.
Handler(s):
"DISPATCHER" established:1 refused:0 curr:0 max:1022 state:ready
D001 <machine: db.us.oracle.com, pid: 8705>
  (ADDRESS=(PROTOCOL=tcp) (HOST=db.us.oracle.com) (PORT=35230))
"DISPATCHER" established:1 refused:0 curr:0 max:1022 state:ready
D000 <machine: db.us.oracle.com, pid: 8703>
  (ADDRESS=(PROTOCOL=tcp) (HOST=db.us.oracle.com) (PORT=35229))
"DEDICATED" established:0 refused:0
```

Pour vérifier que les connexions utilisent les processus serveur partagés, vous pouvez consulter la vue **V\$CIRCUIT** afin de visualiser une entrée correspondant à chaque connexion à travers un processus serveur partagé.

```
SQL> SELECT dispatcher, circuit, server, status
2> FROM v$circuit;

DISPATCH CIRCUIT  SERVER  STATUS
-----
82890064 8257BA64 8288F6A4 NORMAL
8288F9E4 8257BBB0 00      NORMAL
8288FD24 8257BCFC 00      NORMAL
```

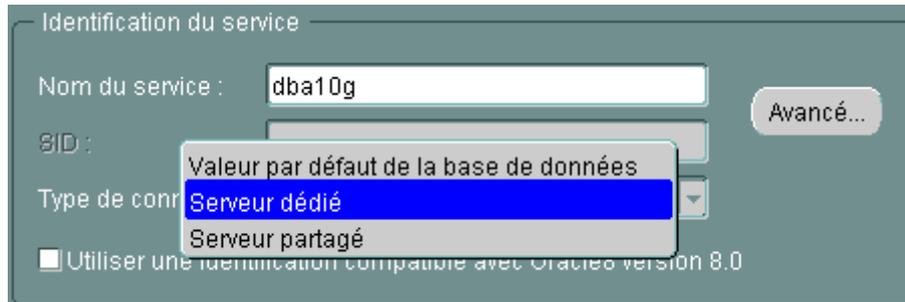
8.3.2. Vues du dictionnaire de données

| | |
|--------------------------|--|
| V\$CIRCUIT | Cette vue contient les informations concernant les circuits virtuels qui sont les connexions utilisateur à travers les dispatchers et les processus serveur. N'importe quel connexion au serveur va générer une entrée dans cette vue. |
| V\$SHARED_SERVER | Cette vue contient les informations sur les processus serveur partagés. |
| V\$DISPATCHER | Cette vue donne des informations sur les dispatchers. |
| V\$SHARED_SERVER_MONITOR | Cette vue contient les informations nécessaires pour tuner les processus serveur partagés. |
| V\$QUEUE | Cette vue contient les informations sur les queues de réponse et de requêtes. |
| V\$SESSION | Cette vue liste les informations pour chaque session en cours. |

8.3.3. Choisir un type de connexion

La connexion par serveur partagé est le type de connexion par défaut pour Oracle Net. Si l'instance a été configurée comme un serveur partagé et qu'une requête de connexion ne demande pas explicitement un serveur dédié, la connexion sera une connexion de type partagé.

Grâce à Oracle Net Manager nous pouvons spécifier le type de connexion pour la résolution locale et ou la résolution par annuaire. Pour cela il suffit de sélectionner le type désiré grâce à la zone de liste déroulante Type de connexion.



8.3.4. Quand est ce qu'il ne faut pas utiliser un serveur partagé

L'architecture serveur partagé d'Oracle est un modèle efficace d'utilisation de la mémoire et de gestion des processus.

Les processus serveur partagés ne sont pas appropriés pour certaines opérations qui manipulent des quantités importantes de données, comme les requêtes portant sur des millions de lignes ou les tâches batch. Cette architecture n'est pas appropriée pour ces traitements parce que toutes les requêtes seront placées dans une même file d'attente et que plusieurs utilisateurs peuvent se partager la même queue de réponse d'un dispatcher.

Les sessions de sauvegarde et de restauration utilisant Oracle Recovery Manager manipulent aussi des quantités importantes de données donc doivent aussi utiliser des connexions dédiées.

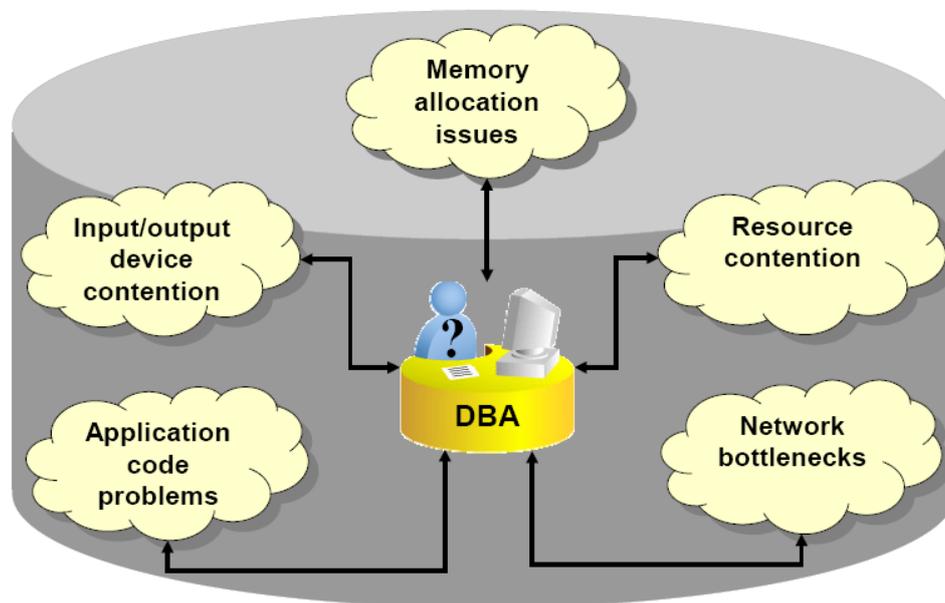
Plusieurs tâches d'administration ne peuvent pas ou ne doivent pas être effectuées en utilisant des processus serveur partagés. Parmi ces tâches nous pouvons citer les opérations de démarrage et d'arrêt de l'instance, la création de tablespaces ou des fichiers de données, la maintenance des tables et des index, l'analyse des statistiques et d'autres tâches réservées au DBA. Toutes les sessions des DBA doivent s'effectuer en utilisant des processus serveur dédiés.

9. Module 09 : Surveillance et Gestion

9.1. Supervision de la performance

9.1.1. Supervision de la performance

Pour administrer la base de données Oracle 10g, l'administrateur de base de données doit régulièrement superviser la performance pour déceler les problèmes et les corriger. Le DBA dispose de centaines de mesures de gestion de la performance, couvrant l'ensemble du serveur, de la performance du réseau à la vitesse des entrées/sorties, en passant par le temps processeur utilisé pour une opération spécifique. Ces mesures de performances sont communément appelées métriques de base de données.



9.1.2. Méthodologie de supervision

La base de données Oracle 10g comprend la supervision proactive et réactive. La supervision réactive est la réponse à un problème connu. Vous pouvez superviser les métriques de performances en rapport aux plaintes d'un utilisateur concernant le temps de réponse, l'échec d'une instance, ou les erreurs trouvées dans les fichiers de trace. La supervision réactive est parfois nécessaire, mais votre objectif est de détecter et résoudre les problèmes avant qu'ils n'affectent le système. Résoudre les problèmes avant qu'ils ne surviennent, ou alors juste avant qu'ils soient vraiment notifiés, est une approche proactive concernant la maintenance du système. La base de données Oracle 10g inclut différents outils pour une supervision proactive. Deux de ces outils sont les alertes générées au niveau du serveur et l'outil de diagnostic automatique de la base de données (ADDM).

9.1.3. Métriques de la base de données et de l'instance

Il existe des centaines de mesures différentes pour superviser votre système. Le dictionnaire des données fournit des informations concernant l'espace utilisé et le statut des objets. Les métriques en temps réel supervise la gestion de la mémoire, les événements en attentes, les entrées/sorties sur les périphériques et l'activité de l'instance via les vues dynamiques de performance, aussi connu sous le nom de vues « **v\$** ». La base de données Oracle peut être extrêmement volumineuse (il est aujourd'hui courant que les bases de données stockent des téraoctets ou hexa-octets de données). Avec de tels volumes de données il est important que l'instance puisse accéder très rapidement aux données qu'un utilisateur réclame. La partie de l'instance qui décide comment accéder aux données est connue sous le nom d'**Optimizer**. Les métriques concernant la distribution des données sont utilisées pour permettre à l'Optimizer de choisir le moyen le plus rapide de récupérer les données, et ainsi donc communément appelé « **Optimizer Statistics** ».

« **Optimizer Statistics** » montre la distribution des données pour chaque table en incluant:

- Le nombre de lignes
- La taille moyenne d'une ligne
- L'espace libre alloué à la table
- Le nombre de lignes « chaînées » (existantes dans deux ou plusieurs blocks de données à cause de la taille de la ligne ou de l'inefficacité des structures de stockage)

Un exemple de choix de l'Optimizer est la différence entre le scan complet d'une table (lecture de la table en entier pour trouver une ligne particulière) et le scan d'un indexe (les lignes sont d'abord cherchées dans l'indexe, puis les blocks sont directement trouvés dans la table qui contient la ligne). Pour les tables petites, le scan complet d'une table est bien plus efficace que de regarder dans un indexe pour récupérer la position de la ligne et trouver le bon bloc. Pour les tables plus grandes (n'oubliez pas que cela peut inclure plusieurs centaines de millions de lignes) l'utilisation d'un index prend davantage de sens.

9.1.4. Métriques du dictionnaire des données

Un des objectifs les plus importants des métriques du dictionnaire des données est de déterminer le statut des objets. Les indexes et les procédures stockées doivent être **VALID** pour être utilisés. Les indexes peuvent devenir **UNUSABLE** à cause des opérations de maintenance sur les tables. Les indexes stockent des pointeurs vers l'adresse physique d'une ligne spécifique. Si le DBA déplace les lignes pour des opérations de maintenance comme par exemple avec la commande **MOVE**, les pointeurs ne sont alors plus valides et l'indexe devra être reconstruit pour être à nouveau utilisé. Le code PL/SQL des objets peut avoir un statut **INVALID** pour différentes raisons. Si le code contient des erreurs, l'objet PL/SQL ne sera pas compilé correctement et sera marqué **INVALID**. Une procédure valide peut devenir invalide si un objet référencé devient invalide. Supposons qu'une procédure référence la colonne salary de la table employees. Si la définition de la colonne change, ou si la colonne est supprimée, rien ne dit que la procédure qui fonctionnait avec l'ancienne définition de la colonne fonctionnera avec la nouvelle. Les indexes **UNUSABLE** demande l'intervention du DBA pour les rétablir dans un état valide. Les objets PL/SQL **INVALID** sont normalement recompilés automatiquement la première fois qu'ils sont appelés, mais parfois demande l'intervention du DBA lorsque cette recompilation échoue.



9.1.5. Objets INVALID et UNUSABLE

Si vous trouvez des objets dont le statut est **INVALID**, la première question à vous poser est « l'objet a-t-il déjà été **VALID** ? » Souvent, les développeurs négligent de retravailler du code qui ne fonctionne pas. Si l'objet PL/SQL est **INVALID** à cause d'un code erreur, alors il y a peu de chose à faire pour résoudre le problème.

Si la procédure était valide à un moment dans le passé et est devenue récemment invalide, vous avez alors deux possibilités pour résoudre le problème:

- **Ne rien faire.** La plupart des objets PL/SQL sont automatiquement recompilés si nécessaire quand ils sont appelés. Les utilisateurs auront un léger délai d'attente pendant la recompilation des objets (dans la plupart des cas ce temps d'attente est transparent pour les utilisateurs).
- **Recompiler manuellement les objets INVALID.** Les objets PL/SQL invalides peuvent être recompilés manuellement en utilisant Enterprise Manager ou avec les commandes SQL:

```
ALTER PROCEDURE HR.updateSalary COMPILE;

ALTER PACKAGE HR.maintainemp COMPILE; -- for a package
ALTER PACKAGE HR.maintainemp COMPILE BODY; --for a package
```



Les indexes **UNUSABLE** sont marqués valides une fois reconstruit (recalcule des pointeurs). La reconstruction d'un index inutilisable recrée l'index à un nouvel emplacement et supprime l'index inutilisable.

Cela peut être fait soit par Enterprise Manager soit par les commandes SQL :

```
ALTER INDEX HR.emp_empid_pk REBUILD;
ALTER INDEX HR.emp_empid_pk REBUILD ONLINE;
ALTER INDEX HR.email REBUILD TABLESPACE USERS;
```

Si la clause **TABLESPACE** est omise, l'indexe est reconstruit dans le même tablespace que celui où il existait précédemment. La clause **REBUILD ONLINE** permet aux utilisateurs de continuer à mettre à jour l'indexe alors que la reconstruction s'effectue (sans le mot clé **ONLINE** les utilisateurs doit attendre que la reconstruction soit terminée avant d'effectuer tout DML sur la table). Enterprise Manager utilise la réorganisation pour réparer les indexes inutilisables.

Remarque: La reconstruction d'un indexe nécessite de l'espace libre. Vérifier qu'il y a suffisamment d'espace libre avant la reconstruction. Enterprise Manager vérifie automatiquement l'espace nécessaire.

9.1.6. Statistiques pour l'Optimizer

Les statistiques pour les tables et les indexes sont stockées dans le dictionnaire des données. Ces statistiques ne sont pas prévues pour fournir des données en temps réel. Elles fournissent à l'Optimizer des snapshots des données concernant le stockage et la distribution des données qu'il utilisera pour prendre les bonnes décisions concernant l'accès aux données.

Les métriques collectées inclus:

- La taille de la table ou de l'indexe en bloc de base de données
- Le nombre de ligne
- La taille moyenne d'une ligne et le nombre de chaînage (pour les tables uniquement)
- La hauteur et le nombre de feuilles de lignes supprimés (pour les indexes uniquement)

Comme des données sont insérées, supprimées, et modifiées, les valeurs changent. L'impact sur les performances pourrait être significative à cause de la collecte des statistiques, ainsi les statistiques sont mises à jour périodiquement pour les tables et les indexes.

Si votre base de données a été créée avec Database Creation Configuration Assistant (**DBCA**), les statistiques pour l'Optimizer sont collectées automatiquement une fois par jour entre 22 heures et 6 heures. Si votre base de données n'a pas été créée avec DBCA, ou si votre base de données contient des tables et des indexes qui sont très volatiles, vous devez collecter les statistiques manuellement. Une table volumineuse dont le taux d'augmentation (ou de réduction) est de 10% pendant une période de 24 heures est considérée trop volatile pour une seule collecte de statistiques par jour. Pour ces tables dont les changements sont nombreux, Oracle recommande de collecter des statistiques plus fréquemment, de préférence assez souvent pour que la table ne subisse pas de modification de plus 10% entre deux collectes. Les statistiques peuvent être collectées par Enterprise Manager, ou en utilisant le package **DBMS_STATS** comme suit :

```
SQL> EXEC dbms_stats.gather_table_stats('HR','EMP');
```

```
SQL> SELECT num_rows  
FROM dba_tables  
WHERE owner='HR'  
AND table_name = 'EMP';
```

NUM_ROWS

107

Remarquez que le nombre de lignes reflète maintenant ce qui était dans la table au moment où les statistiques ont été collectées. **DBMS_STATS** permet aussi la collecte manuelle de statistiques pour un schéma entier, ou même pour la base de données toute entière.

9.1.6.1. Collecter des statistiques manuellement

L'assistant de collection de statistiques fournit dans Enterprise Manager (disponible à partir de la page Maintenance) est la manière recommandée pour collecter ou mettre à jour des statistiques.

L'assistant vous propose les options de configuration et crée le job pour collecter les statistiques:

- **Default Method:** Offre le choix de supprimer, évaluer, ou de calculer des statistiques. Le calcul de statistiques analyse chaque ligne dans la table; l'évaluation prend un échantillon pour les statistiques. Pour les indexes et le petites tables la méthode par calcul peut être utilisée. Pour les tables plus volumineuses une évaluation est plus intéressante. Dans la plupart des cas le pourcentage par défaut d'évaluation peut être utilisé pour l'évaluation (remarque: une évaluation de 50% ou plus déclenche un calcul. Lorsque l'on choisit manuellement une taille pour l'évaluation, 5 à 15% suffise.)
- **Object selection:** Choisissez les objets pour lesquelles vous voulez collecter des statistiques. Vous pouvez choisir des tables seules ou des indexes ou diffuser la portée pour l'ensemble des schémas ou même la base de données entière.
- **Schedule:** Fournit une authentification au niveau du serveur et détermine quand les statistiques seront collectées. Le choix de planification peut être **immediately**, **scheduled time**, ou **named maintenance**. Ci-dessous vous est montré le code généré par l'assistant et vous permet de valider le job.

The image shows two screenshots from Oracle Enterprise Manager. The left screenshot displays the 'Database: dba10g' interface with a 'Utilities' menu where 'Gather Statistics' is highlighted with a red arrow. The right screenshot shows the 'Gather Statistics Wizard: Review' step, which is Step 5 of 5. It includes a progress bar at the top with steps: Introduction, Default Method, Object Selection, Schedule, and Review (highlighted). Below the progress bar are buttons for 'Cancel', 'Back', 'Step 5 of 5', and 'Submit'. The main content area contains the following text:

Review the following for accuracy and make changes as necessary:
Job ID GatherStats1908451 **Execution Time** Immediately

Generated SQL for gathering statistics

```
begin
dbms_stats.gather_schema_stats(
ownname=> 'HR',
granularity=> 'DEFAULT',
block_sample=> FALSE,
cascade=> TRUE,
method_opt=> 'FOR COLUMNS SIZE AUTO',
options=> 'GATHER AUTO');
end;
```

9.1.6.2. Automatiser la collecte de statistiques

Le planificateur d'Oracle automatise les tâches répétitives comme la collecte de statistique. A partir de la page **Administration**, cliquez sur **Jobs** et créez un nouveau job. Si le PL/SQL ne vous est pas familier, vous pouvez copier coller le code généré pour collecter manuellement les statistiques.

Database: dba10g > Jobs > Create Job

Create Job

Show SQL Cancel OK

General Schedule Options

* Name

* Owner

Enabled Yes No

Description

Logging Level
Set how much logging pertaining to this job should be done

Job Class
Whether the job should remain after having completed

Auto Drop
Whether the job should remain after having completed

Restartable
Whether the job can be safely restarted (and should be restarted in case of failure)

Vous pouvez aussi collecter des informations en utilisant du code PL/SQL. Par exemple:

Command

Select the command type for this job, and specify the required information for that command type.

Command Type **PL/SQL Block**

PL/SQL

```
begin
dbms_stats.gather_schema_stats(
ownname=> 'HR',
granularity=> 'DEFAULT',
block_sample=> FALSE,
cascade=> TRUE,
method_opt=> 'FOR COLUMNS SIZE AUTO',
options=> 'GATHER AUTO');
end;
```

9.1.6.3. Planifier la collecte de statistiques

Planifier la collecte de statistiques à des intervalles fréquents pour que l'Optimizer puisse prendre les meilleures décisions pour l'accès aux données.

Souvenez vous que le fait de collecter des statistiques implique du travail pour l'instance. Eviter de planifier la collecte de statistiques pendant les pics d'utilisation du serveur pour minimiser l'impact sur les performances.

Create Job

Show SQL Cancel OK

General **Schedule** Options

Schedule Type Standard

Time Zone GMT -07:00 Change Time Zone

Repeating

Repeat Do Not Repeat

Start

Immediately
 Later

Date Feb 16, 2004
(example: Feb 16, 2004)

Time 5 05 00 AM PM

9.1.7. Vues dynamiques de performance

Il y a plus de 300 vues dynamiques de performance dans la base de données Oracle 10g qui fournissent des informations détaillées sur les sessions, les statistiques systèmes, les entrées/sorties sur les fichiers, et sur l'activité de l'instance.

Les vues dynamiques de performance fournissent un snapshot des données en temps réel. Comme vous pouvez le voir dans l'exemple ci-dessous, chaque accès à une vue dynamique de performance donne des résultats à jour. Pour les statistiques montrées ci-dessous, le fait de sélectionner les informations avec la clause **ORDER BY** abouti à un classement (mémoire) pour incrémenter.

```
SELECT name, value
FROM v$sysstat
WHERE name='sorts (memory)'
ORDER BY name;
```

| NAME | VALUE |
|----------------|-------|
| sorts (memory) | 4476 |

SQL> /

| NAME | VALUE |
|----------------|-------|
| sorts (memory) | 5021 |

Les données dynamiques de performance sont non-persistantes. Une fois l'instance arrêtée, les vues sont remises à zéro et recommence leur incrémentation. Cela signifie que les données de performances obtenues à partir des vues dynamiques de performance ne devraient pas être utilisées pour faire des choix de configuration tant que l'instance n'a pas été opérationnelle pendant un long moment. Les métriques dynamiques de performance sont peu souvent significatives comme seule mesure. Elles ne devraient être utilisées que pour obtenir des informations sur la « tendance » des performances de la l'instance (augmentation, déclin).

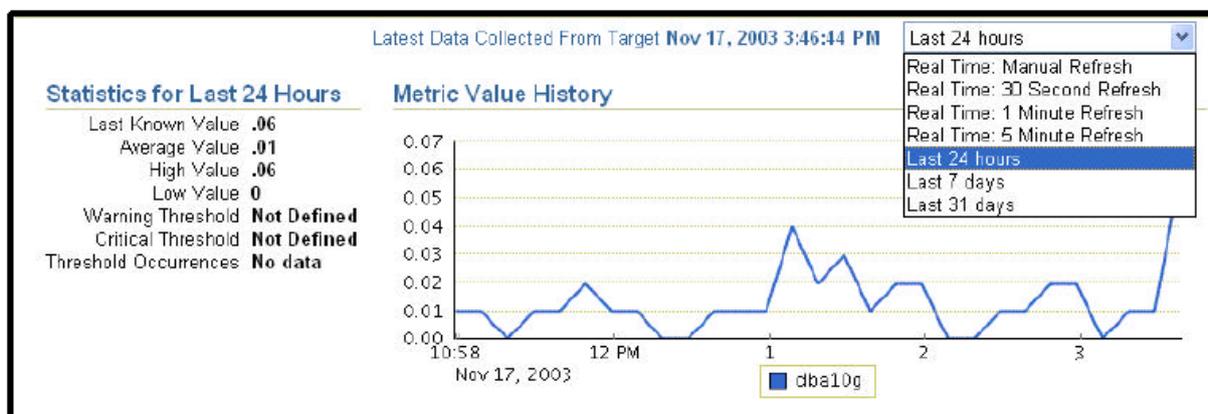
9.1.8. Obtenir des informations sur les performances

La page **All Metrics** d'Enterprise Manager groupe les métriques par fonctionnalités. Déroulez la zone qui vous intéresse et cliquez sur lien pour la métrique que vous souhaitez obtenir.

| Metrics | Thresholds |
|--------------------------|------------|
| orcl.oracle.com | |
| ▶ Alert Log | Some |
| ▶ Alert Log Content | None |
| ▶ Alert Log Error Status | All |

La page sur les détails des métriques affiche:

- La dernière valeur connue
- Les valeurs des High-water mark et Low-water mark
- Les accès moyens en lecture
- La représentation graphique de la métrique tracée sur une période définie par l'utilisateur



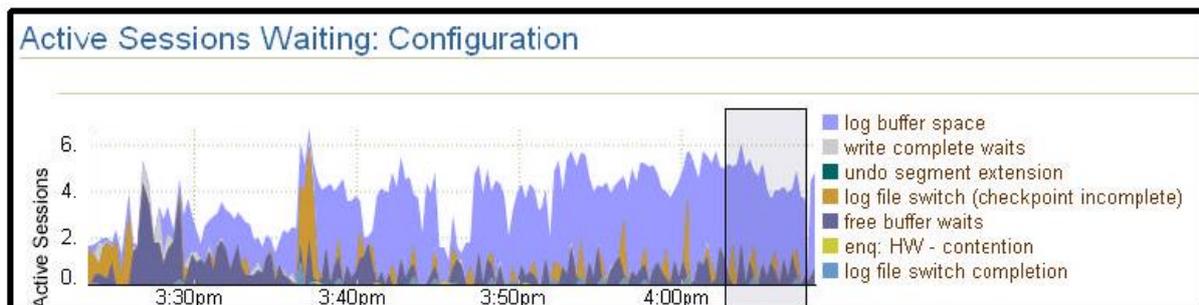
9.1.9. Réagir aux problèmes de performance

La page **Performance** dans Enterprise Manager est le meilleur moyen de superviser le serveur si vous remarquez une dégradation des performances.

La page Performance se divise en 3 principales parties:

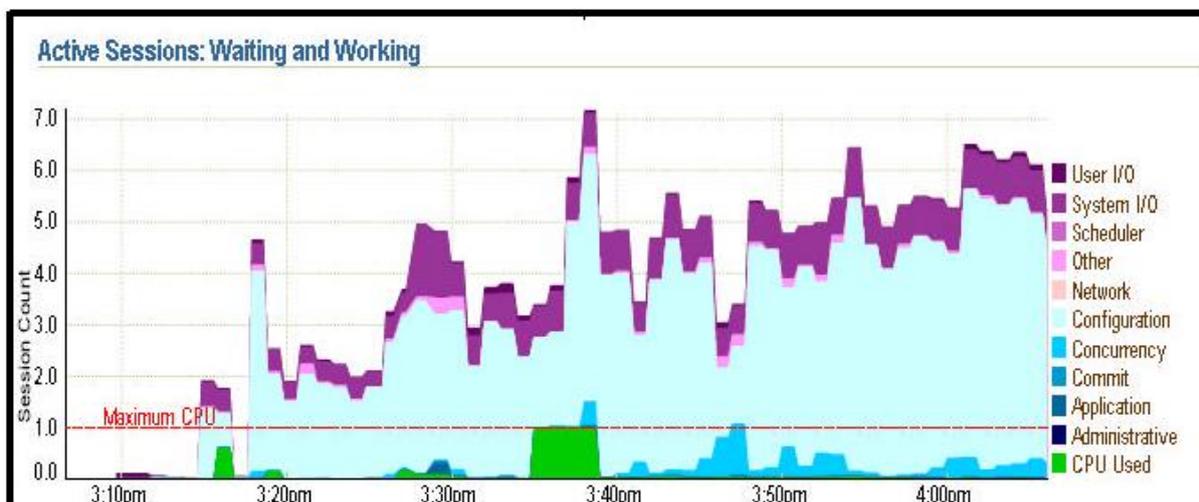
- **Host:** Métriques du serveur montrant le nombre de processus en attente et la taille de la mémoire paginée
- **Active Sessions:** En attentes et en cours de traitement: Une vue d'ensemble des performances de l'instance montrant des métriques de groupes de différentes catégories. Si une catégorie consomme une partie significative du temps d'attente pour l'instance, alors c'est à partir de cette catégorie que vous devriez commencer à résoudre les problèmes.
- **Instance Throughput:** Informations concernant les sessions en cours, la génération des redo, et l'activité concernant la lecture des fichiers de données.

Chacune des ces parties peut être forcée en cliquant sur la métrique désirée. Dans l'exemple ci-dessous, les utilisateurs utilisent une partie significative de leur temps à attendre suite à une configuration. Cliquez sur la mesure de configuration pour voir d'où vient le problème.



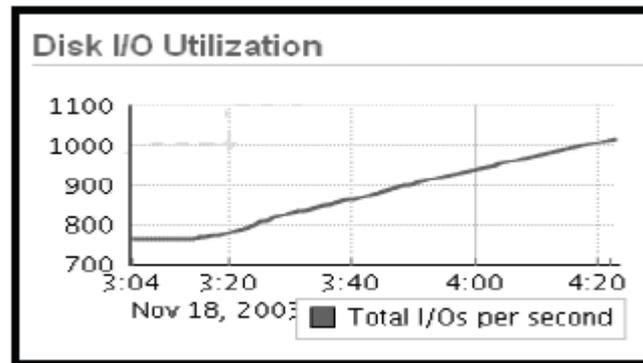
Avancer plus profondément dans la configuration du temps d'attente montre que la majorité du temps d'attente pour l'espace provient du redo log buffer. Dans ce cas, le DBA doit déterminer d'où vient le problème, du redo log buffer lui-même (nécessitant davantage d'espace) ou dans les fichiers de redo (les temps d'attente pour écrire dans les fichiers de redo est trop long).

Avancer plus profondément dans le graphique de **Run Queue** dans la région Host de la page Performance montre que les entrées/sorties sont élevées et augmentent. Après de plus amples investigations en utilisant les outils du système d'exploitation vous trouverez sans doute que les groupes de redo (les fichiers de redo) sont dans les mêmes disques.



La contention des entrées/sorties est souvent problématique. Oracle recommande:

- Mettre les fichiers de redo log sur des disques séparés
- Séparer les indexes et les données dans des fichiers et des tablespaces différents



9.2. Maintenance proactive

9.2.1. Gestion des seuils d'alertes

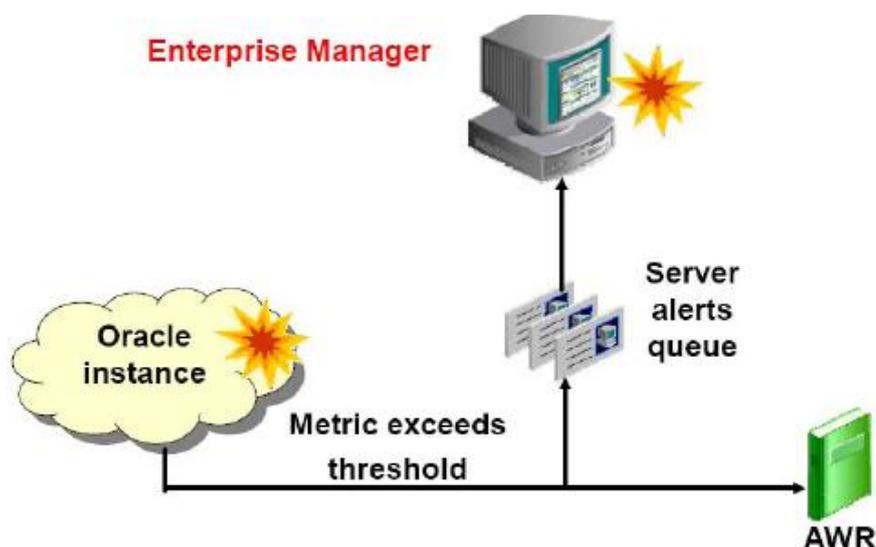
9.2.1.1. Alertes générées par le serveur

Oracle Database 10g remonte des alertes à l'administrateur via Enterprise Manager Database Console. Enterprise Manager peut être configuré optionnellement pour envoyer une notification par e-mail à l'administrateur sur un problème, de la même sorte qu'il est notifié dans la console.

Vous pouvez aussi sélectionner les seuils d'alertes (thresholds) sur plusieurs métriques de votre système. Oracle Database 10g peut vous prévenir si l'activité de la base dévie de la normale atteignant ainsi ces seuils d'alertes. Plus tôt les notifications aux problèmes potentiels sont remontées, plus vous pouvez y répondre rapidement, et les résoudre avant que ceux-ci n'affectent les utilisateurs.

Quelques métriques principales qui peuvent fournir des notifications plutôt :

- Le temps d'écriture moyen (centième de secondes)
- L'évolution de la taille du Dump Area (en pourcentage)
- Le temps de réponse (par transaction)
- Le temps de réponse des requêtes SQL (en pourcentage)
- L'espace utilisé des tablespaces (en pourcentage)
- Le temps d'attente (en pourcentage)



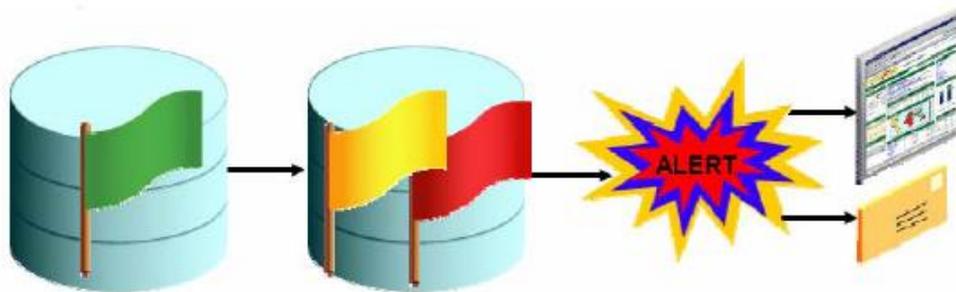
9.2.1.2. Seuils d'alertes

Les seuils d'alertes sont des valeurs limites qui sont comparées avec celles des métriques. Vous pouvez spécifier des seuils d'alertes (warning) ou critiques qui une fois dépassés une alerte est générée. Cette alerte vous informe des problèmes imminents.

Les seuils d'alerte de type warning sont généralement utilisés pour fournir un avis quand les performances de la base de données dévient de leur activité normale. Aucune action n'est requise habituellement pour ces warning, ils servent à avertir le DBA de surveiller cette situation.

Les alertes critiques doivent être utilisées pour indiquer un problème qui requière une attention immédiate. Des actions de réponses peuvent être configurées pour s'activer dès que les seuils d'alertes critiques sont atteints.

Le but de l'administrateur est de recoudre les problèmes avant que ces seuils d'alertes critiques soient atteints.



9.2.1.3. Paramétrer les seuils d'alertes

Cliquez sur le bouton Edit Threshold sur la page Manage Metrics pour déterminer les seuils d'alertes warning et critiques. Pour certaines métriques vous pouvez choisir de définir des cibles spécifiques. Par exemple, vous pouvez définir des valeurs différentes pour la métrique Tablespace Space Used (%) en fonction de chaque du tablespace. L'alerte de vos tablespaces en écriture sera à 70% de sa capacité mais celle de vos tablespace en lecture seule sera à 95%.

Les seuils d'alertes détaillés peuvent être définis en sélectionnant la métrique appropriée et cliquez sur Manage Metric Indexes.

| Manage Metrics | | | |
|---|---------------------|-------------------|--------------------|
| Thresholds | | Baselines | |
| Edit Thresholds | | | |
| Pending changes: 0 | | | |
| Metric | Comparison Operator | Warning Threshold | Critical Threshold |
| Archive Area Used (%) | > | 80 | |
| Archiver Hung Alert Log Error | Contains | | ORA- |
| Archiver Hung Alert Log Error Status | > | 0 | |
| Audited User | = | SYS | |
| Average File Read Time (centi-seconds) | > | | |
| Average File Write Time (centi-seconds) | > | | |

9.2.1.4. Mesure à l'aide des Baselines

Vous pouvez baser vos seuils d'alertes pour les métriques de performances en fonction de l'activité de votre base et non sur des nombres. Par exemple, si la base de données fonctionne sous la charge de travail normal avec des performances acceptables à une heure et un jour donné, vous pouvez définir des seuils d'alertes tels que pour une variation de 10% vous êtes alerté et pour une variation de 25% vous recevez une alerte critique.

L'assistant Create Baseline vous permet de capturer des statistiques de performances moyennes à une heure donnée, ou si vous ne sélectionnez pas d'heure sur une période de 24h indiqué par le champ Date. Cela calcule les déviations de warning et critique à partir d'une moyenne des performances.

Pour déterminer un Baseline :

1. Ouvrez l'assistant Create Baseline depuis la page Manage Metric.
2. Spécifiez une date ou une heure quand les performances et la charge de travail sont normales
3. Fournissez un pourcentage par défaut pour les seuils critique et warning et cliquez sur le bouton Go.
4. Choisissez la ou les métriques concernées par ces seuils (par défaut toutes les métriques sont sélectionnées). Les valeurs des seuils d'alertes sont calculées automatiquement en utilisant les pourcentages spécifiés dans l'étape 3. vous pouvez toujours les modifier si nécessaire.
5. Après que tous les changements désirés aient été faits sur les seuils d'alertes, cliquez le bouton OK.
- 6.

Create Metric Baseline

Create a metric baseline by specifying the date whose performance metric data will be used as a basis to calculate thresholds. If you choose, Warning and Critical thresholds will be calculated based on the percentages specified.

* Name

* Date

(Specify a date where performance was acceptable for this target.)
(example: Dec 15, 2003)

Hour of day AM PM

Warning Percentage

Critical Percentage

(The Warning and Critical percentages will be used against metric baseline Low and High Values to calculate metric thresholds. See help for details.)

9.2.1.5. Utilisation des Baselines

Après avoir créé une baseline, vous l'appliquez en cliquant sur le bouton Copy Thresholds From Baseline dans la page Edit Thresholds, et sélectionnez celle que vous voulez utiliser. Les valeurs calculées de ces seuils d'alertes depuis la baseline sont appliquées pour toutes les métriques concernées par celle-ci. Vous pouvez bien sûr modifier ces valeurs.



9.2.2. SQL Tuning et autres outils

Automatic Database Diagnostic Monitor (ADDM) : un outil coté serveur qui analyse les performances de la base de données toutes les 60 minutes. Le but d'ADDM est de détecter les goulots d'étranglements possibles dans le système et propose des solutions avant que les performances du serveur ne se dégradent.

SQL Tuning Advisor : Ce conseiller analyse les requêtes SQL individuellement et propose des recommandations pour améliorer les performances. Ces actions peuvent être une réécriture de la requête, un changement de la configuration de l'instance, ou l'ajout d'index. SQL Tuning Advisor n'est pas appelé directement. Cependant, il est appelé par d'autres outils tels que Top SQL ou Top Sessions, pour que l'optimisation sur les requêtes SQL ait un meilleur impact.

SQL Access Advisor : Cet outil analyse toutes les requêtes SQL exécutées pendant une période donnée et recommande la création d'index additionnels ou de vues matérialisées qui augmenteraient les performances.

Memory Advisor : Memory Advisor est un ensemble de fonctions conseillères qui aide à déterminer les meilleurs paramètres pour le shared pool, database buffer cache, et la PGA. En plus de ces diverses fonctions, cette page fournit un point de contrôle pour le large pool et le Java pool.

Mean-Time-To-Recover (MTTR) Advisor : MTTR advisor vous guide pour définir la durée nécessaire à la base de données pour effectuer sa récupération après un crash de l'instance. La notion MTTR et l'utilisation de ce conseiller sera abordé plus tard.

Segment Advisor : Il définit les tables et indexes qui consomment plus d'espace que nécessaire. Cet outil vérifie l'espace alloué inutilement au niveau d'un schéma ou d'un tablespace et fournit un script pour réduire celui-ci si possible.

Undo Management Advisor : Undo Management Advisor vous aide à déterminer la taille du tablespace undo qui est requise pour supporter la période de rétention définie.

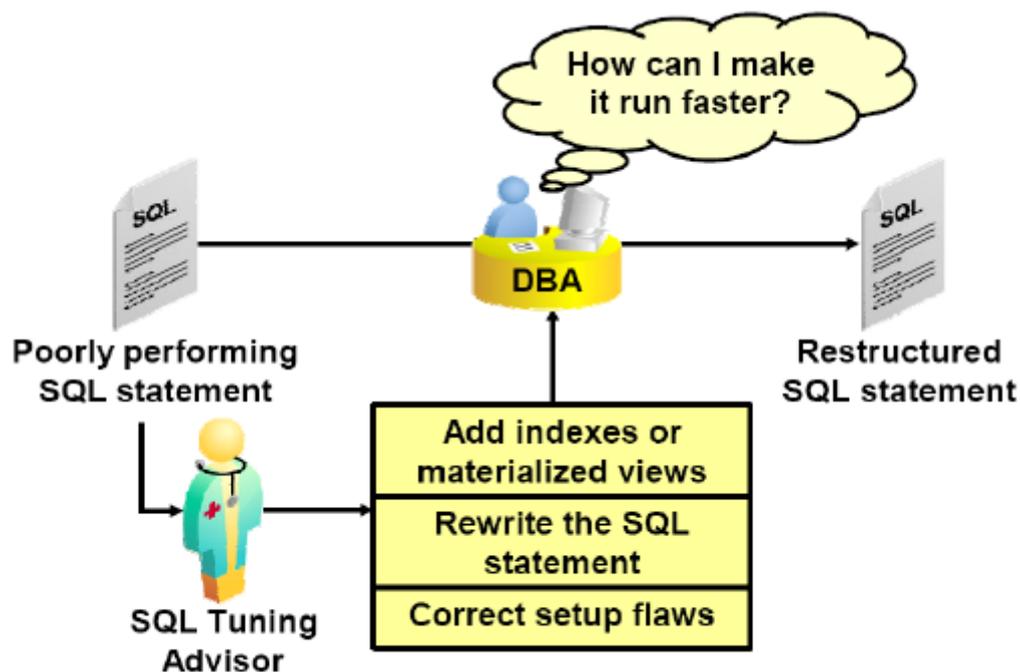
9.2.2.1. SQL Tuning et Access Advisor

Le but de l'analyse du SQL est de vous aider à identifier les requêtes SQL hautement consommatrices en ressources et de proposer des solutions pour augmenter leurs performances.

La performance des requêtes SQL peut être améliorée en ajoutant ou modifiant des indexes, en créant des vues matérialisées, en réécrivant le code SQL afin de le rendre plus performant, et en ajustant les paramètres d'initialisation.

L'analyse et le tuning manuel du SQL est un processus consommateur de ressources qui requiert de l'expertise. Les assistants SQL Tuning et SQL Access vous aident à :

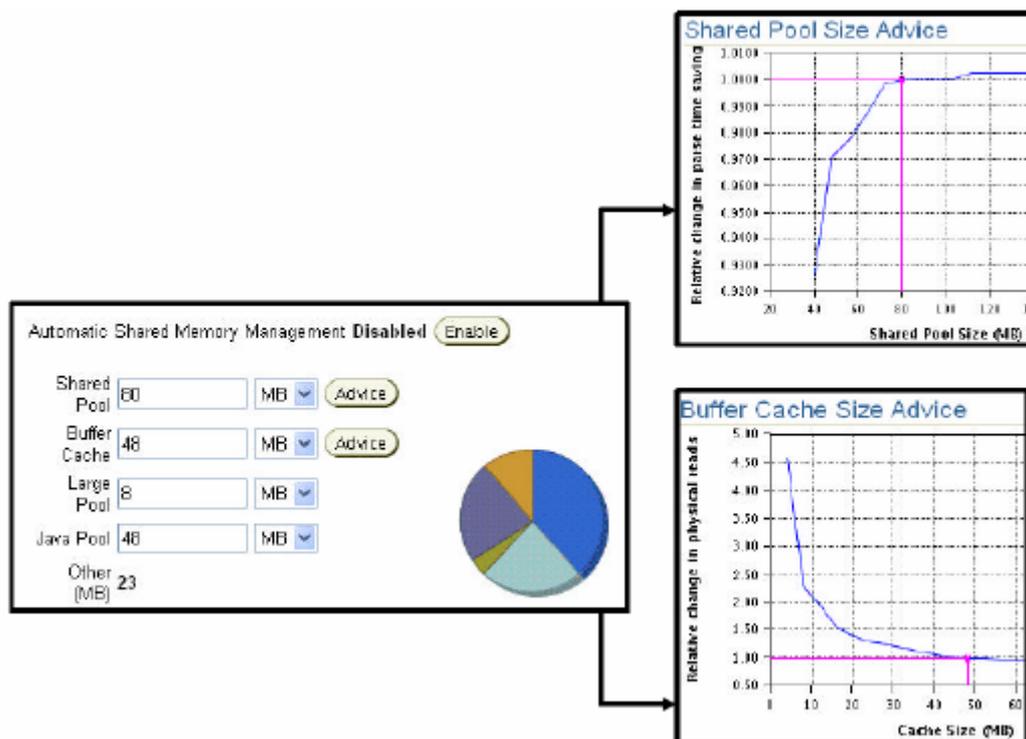
- Rechercher facilement les requêtes SQL hautement consommatrices.
- Reconnaître les erreurs courantes dans la construction des requêtes SQL
- Déterminer les emplacements pour les indexes ou les vues matérialisées, tout en informant sur l'éventuel gain de performance.



9.2.2.2. Memory Advisor

Une taille de SGA trop petite peut dégrader les performances de votre système, mais il est souvent difficile de connaître la quantité de mémoire adaptée. De la même manière, une allocation trop importante des ressources mémoire entraîne une dégradation des performances.

Memory Advisor vous permet facilement de trouver et définir les composants de la SGA avec des tailles appropriées. Chaque conseil propose un graphique montrant les bénéfices d'une augmentation de la mémoire en contraste avec la quantité de mémoire requise pour atteindre plus de bénéfice. L'administrateur devra, à l'aide du graphique, définir une taille pour les différents buffers aux alentours de la « cassure » de la courbe.

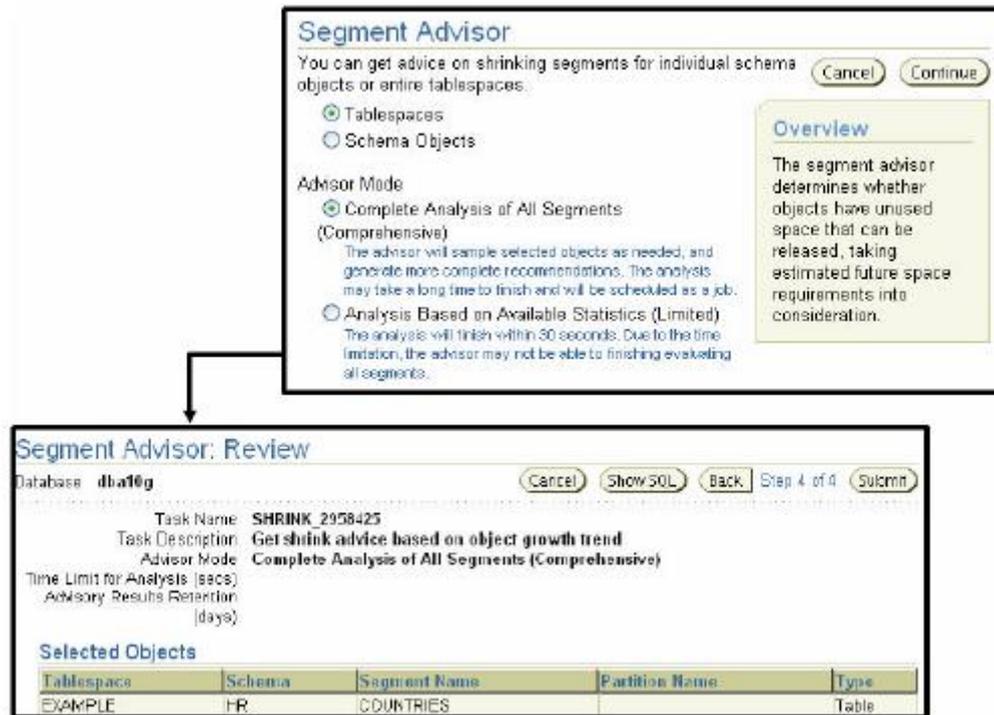


9.2.2.3. Segment Advisor

Quand des lignes sont supprimées d'une table, ou quand elles sont mise à jour avec une valeur qui occupe moins de place que l'originale, l'espace alloué pour cette ligne est retourné à la table pour une utilisation future. Au fil du temps, ceci implique une allocation de mémoire supérieur à ce qui est nécessaire. Segment Advisor regarde les tables et les indexes qui ont un excès d'espace libre et suggère des solutions pour retourner cet espace au reste de la base de données.

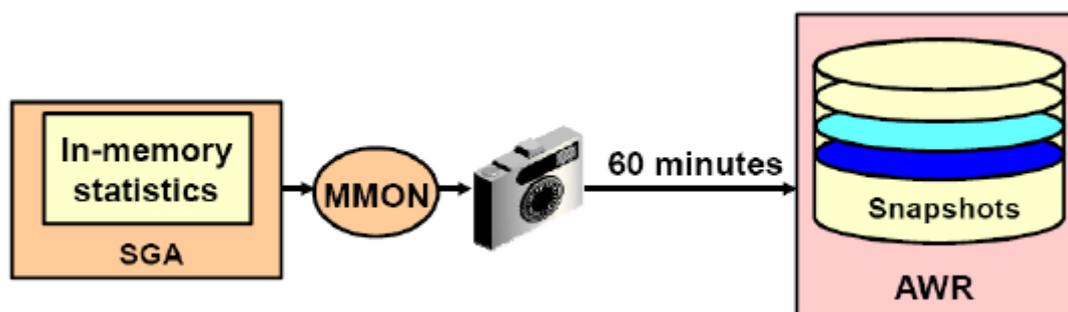
Segment Advisor peut être démarré en utilisant des statistiques existantes d'optimiseur (mode limité) ou peut réellement scanner les tables et les indexes (mode complet) pour avoir de meilleurs conseils. La portée de l'analyse peut porter sur des tables ou indexes individuels ou couvrir un tablespace entier. Les conseils de Segment Advisor requièrent que l'option « row mouvement » soit activée sur la table. Vous pouvez l'activer par Enterprise Manager en ouvrant la page Administration, cliquez sur Tables, et éditez la table que vous voulez modifier. Activez le paramètre Row Movement dans l'onglet Options de la page Edit Table. Vous pouvez aussi l'activer à l'aide de la commande suivante :

```
ALTER TABLE owner.table_name ENABLE ROW MOVEMENT;
```



9.2.3. Automatic Workload Repository

Par défaut, toutes les 30 minutes, la base de données capture automatiquement des statistiques sur la SGA, et les stocke à l'intérieur d'**Automatic Workload Repository (AWR)** sous la forme de snapshot (image de l'état à un instant T). Ces snapshots sont stockés sur le disque et sont conservés pendant sept jours. L'intervalle entre les snapshots et la période de rétention sont modifiables. Les **AWR** contiennent des centaines de tables, qui appartenant à **SYSMAN** et sont situées dans le tablespace **SYSAUX**. Oracle ne supporte pas les accès directs en SQL au repository. Par contre, utilisez Enterprise Manager ou le package **BDM\$_WORKLOAD_REPOSITORY** pour exploiter les **AWR**.



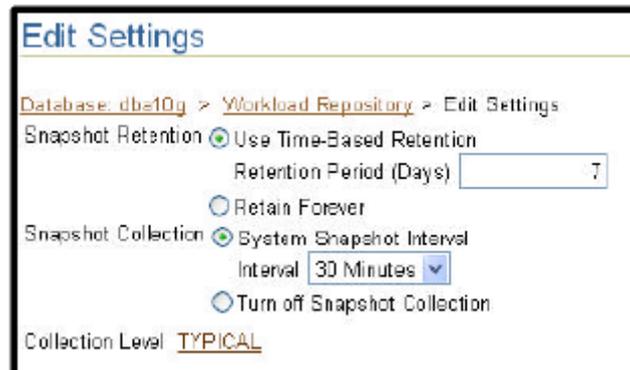
9.2.3.1. Gestion des AWR

Ouvrez la page Administration et cliquez sur le lien Workload Repository pour accéder à la page de gestion des AWR.

Les paramètres des AWR incluent la période de rétention, l'intervalle entre les prises de vues, et le niveau de collection. Rappelez-vous que si vous diminuez un de ces paramètres

Rappelez-vous que diminuer un de ces paramètres agit sur la fonctionnalité des composants qui dépendent des AWR, comme les conseillés par exemple. L'augmentation des valeurs de ces paramètres peut améliorer ces recommandations, mais nécessite plus d'espace pour le stockage des snapshots et sollicite plus de ressources système.

Préférez le niveau de collection (Collection Level) **ALL** pour effectuer le tuning d'une nouvelle application. Le paramètre **ALL** accumule des plans d'exécution SQL et des statistiques pour fournir des recommandations plus précises. Une fois le tuning achevé, ce paramètre doit être remis à **TYPICAL**.



9.2.4. Automatic Database Diagnostic Monitor (ADDM)

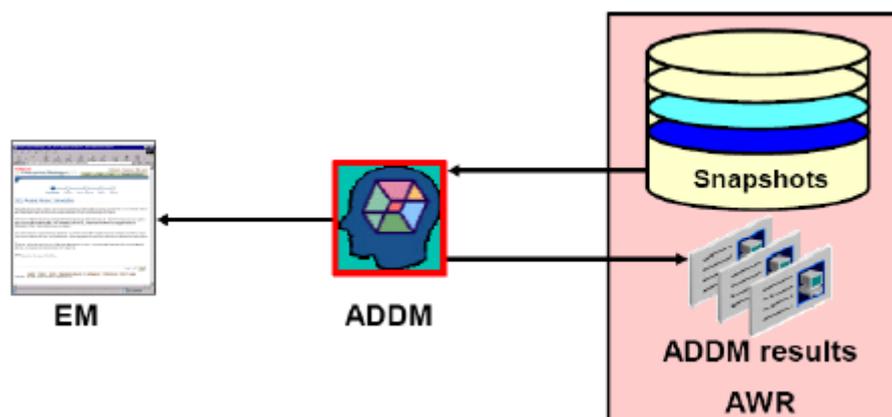
Contrairement aux autres conseillé, ADDM fonctionne automatiquement après chaque snapshot AWR. Chaque fois qu'un snapshot est pris, ADDM fait une analyse d'une période correspondant aux deux derniers snapshots. ADDM fournit une surveillance préventive, et détecte la plupart des goulots d'étranglement avant qu'ils ne deviennent des problèmes concrets.

Dans beaucoup de cas, ADDM propose des solutions aux problèmes détectés et mesure les bénéfices apportés suite à l'application des recommandations.

Quelques problèmes détectés par ADDM :

- les goulots d'étranglement du CPU
- gestion des connexions faibles d'Oracle Net
- les collisions dues aux verrous
- les capacités d'entrée/sortie
- manque de mémoire alloué à Oracle
- Les requêtes SQL hautement consommatrices
- latence du code PL/SQL ou Java
- Opérations de checkpoint trop longues et leurs causes, par exemple, des fichiers de redo trop petits.

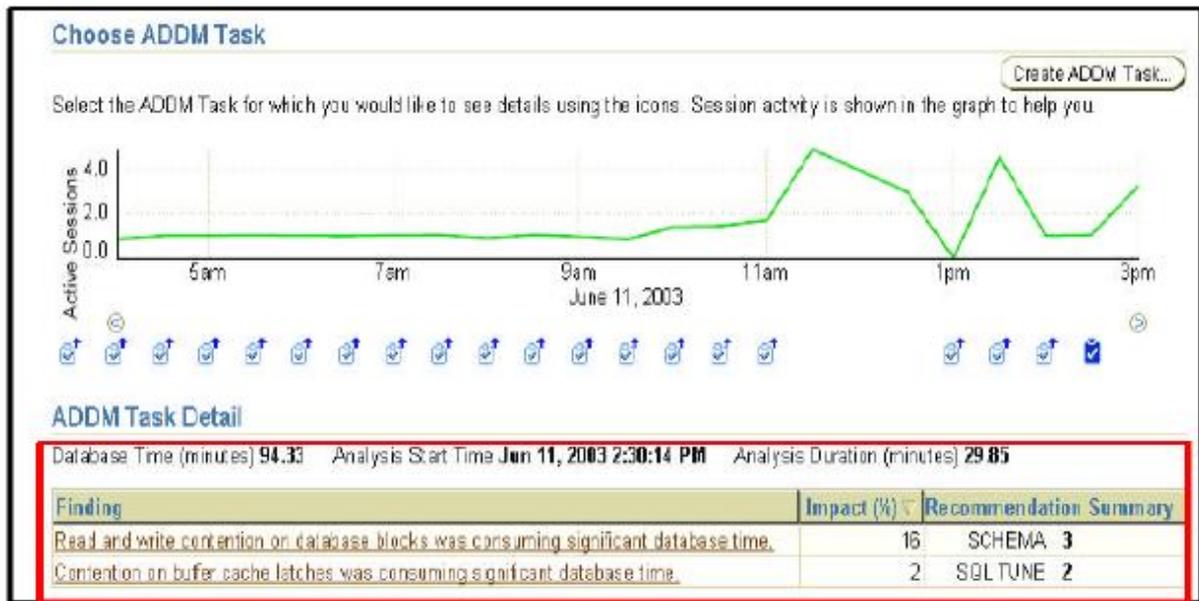
Les résultats de chaque analyse d'ADDM sont stockés dans les AWR, et sont accessibles seulement par Enterprise Manager.



9.2.4.1. Résultats d'ADDM

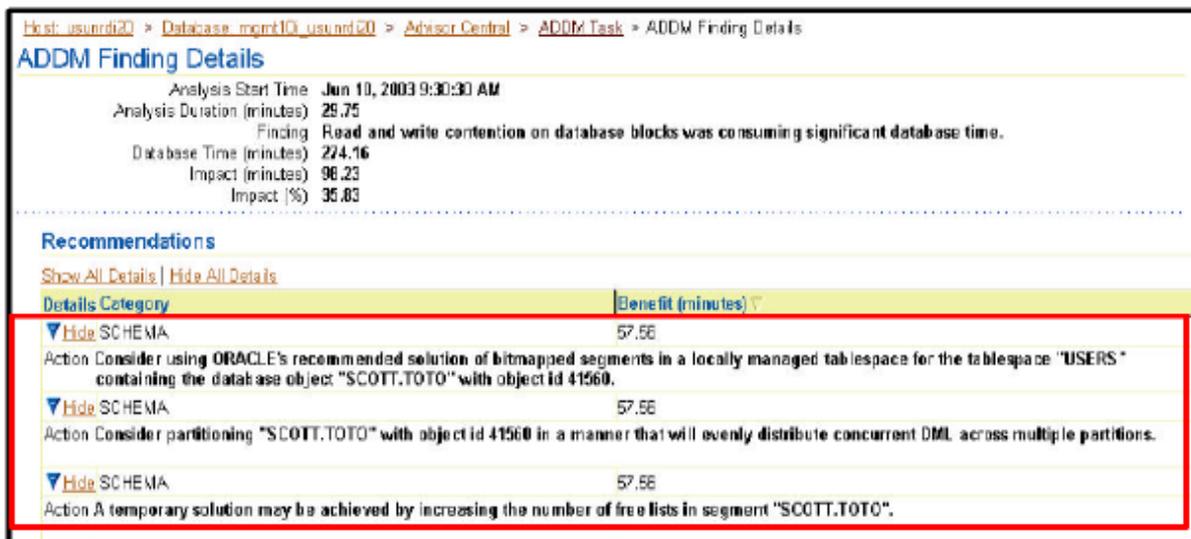
Sur la page ADDM Findings, vous pouvez voir les résultats détaillés de la dernière exécution d'ADDM. Database Time représente la somme des durées d'activités des sessions au niveau de la base de données pour la période analysée. Un pourcentage d'impact spécifique est donné pour chaque résultat. Un impact représente le temps consommé par le problème comparé au temps moyen sur la période d'analyse.

En cliquant sur un résultat particulier, vous êtes redirigés sur la page ADDM Finding Details, qui contient plus d'informations telles que toutes les recommandations qu'ADDM a pu générer.



9.2.4.2. Les recommandations d'ADDM

Sur la page ADDM Finding Details, vous avez certaines recommandations pour résoudre le problème correspondant. Les recommandations sont catégorisées par **SCHEMA**, SQL Tuning, configuration de la base, et autres. La colonne Benefit vous donne le gain apporté au temps base de données si les recommandations sont implémentées.



10. Module 10 : Gestion des Undos

10.1. Concepts

10.1.1. Les données undo

Oracle sauvegarde l'ancienne valeur (donnée undo) quand un processus modifie une donnée dans la base de données. Il stocke la donnée telle qu'elle existait avant qu'elle ne soit modifiée. Stocker les données undo permet aux utilisateurs de changer d'avis (rollback). Les undo permettent aussi la lecture consistante des données et les requêtes flashback.

Les requêtes consistantes sont des requêtes qui débutent avant une modification de données et se terminent parfois après modification des données. Oracle fournit des résultats consistants qui sont les données au moment de l'exécution de la requête. Pour qu'une requête consistante soit réussie, l'information originale doit exister en tant que donnée undo. Aussi longtemps que les informations undo sont conservées, Oracle peut reconstruire les données pour permettre des requêtes consistantes. Les requêtes flashback sont des requêtes qui permettent de récupérer une image des données telle qu'elles étaient dans le passé. Aussi longtemps que les informations undo nécessaires existent, les requêtes flashback peuvent s'exécuter avec succès.

Les données undo sont également utilisées pour récupérer des données après l'échec de transactions. Une transaction échoue quand un utilisateur quitte anormalement sa session (peut être dû à des problèmes réseau ou un problème sur la machine cliente) avant que l'utilisateur n'ait décidé de valider ou d'annuler la transaction. Les transactions peuvent aussi échouer à la suite du crash de l'instance. Au cas où une transaction échoue, l'action la plus sûre choisie par Oracle est d'inverser les modifications effectuées par un utilisateur, donc de restaurer les données initiales.

Des informations undo sont retenues pour chaque transaction jusqu'à ce qu'elle ne se termine par :

- Une annulation des modifications par les utilisateurs (rollback)
- Une validation de la transaction par les utilisateurs (commit)
- La fin anormale de la session d'un utilisateur (rollback)
- La fin de la session d'un utilisateur par le mot clé **EXIT** (commit)

Le temps pendant lequel seront conservées les informations undo dépend du taux d'activité et de la configuration de la base de données.

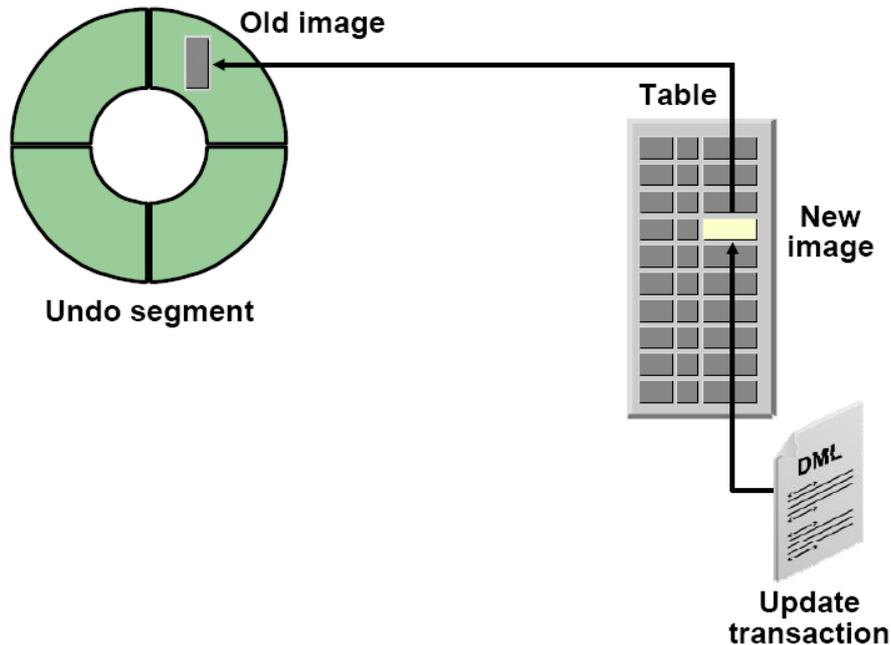
10.1.2. Les transactions et les undos

Quand une transaction débute, un segment undo lui est assigné. Pendant toute la vie de la transaction, n'importe quelle modification effectuée pendant la transaction copiera les valeurs originales dans le segment undo. Vous pouvez voir quelles transactions sont assignées à quels segments undo en consultant la vue dynamique de performance **v\$transaction**.

Les segments undo sont des segments spéciaux qui sont automatiquement créés par l'instance pour pouvoir gérer les transactions. De même que tous les autres segments, les segments undo se composent d'extents qui eux-mêmes se composent de blocks. La taille des segments undo augmente ou diminue automatiquement selon les besoins, fonctionnant comme un buffer de stockage circulaire pour les transactions qui leur sont assignées.

Les transactions remplissent les extents contenus dans leur segment undo jusqu'à ce que la transaction se termine ou que tout l'espace disponible soit utilisé. Si un extent se remplit et que davantage d'espace est requis, la transaction va obtenir cet espace à partir de l'extent suivant dans le segment.

Une fois que tous les extents sont remplis, la transaction réutilise le premier extent (elle réécrit par-dessus les données undo du premier extent qui ne sont plus nécessaires) ou demande à ce qu'un nouvel extent soit alloué au segment (si les données undo du premier extent sont toujours requises).



10.1.3. Stockage des informations undo

Les segments undo peuvent uniquement exister seulement dans un tablespace spécifique appelé "tablespace undo". Alors que la base de données peut avoir plusieurs tablespaces undo, chaque instance ne peut ouvrir qu'un seul de ces tablespaces. Les segments undo sont des segments de type "**TYPE 2 UNDO**" et leur propriétaire est toujours **SYS**. Comme les segments fonctionnent de manière circulaire, chaque segment sera composé de deux extents au minimum. Le nombre d'extents maximum par défaut dépend de la taille des blocks de la base de données mais est très élevé (32765 pour une taille de bloc de 8K).

Les tablespaces undo sont permanent, gérés localement avec allocation automatique des extents. Ils sont gérés comme n'importe quel autre tablespace, excepté la gestion de la restauration. Comme les données undo sont requises pour la restauration après une transaction échouée (comme un crash d'instance), les tablespaces undo peuvent être restaurés seulement quand l'instance est en mode **MOUNT**.

10.2. Gestion des Undos

10.2.1. Administration des undos

En général, les données undo sont gérées automatiquement par l'instance et le DBA intervient peu. Seules certaines opérations peuvent nécessiter l'intervention du DBA:

- Espace insuffisant pour les undo
- Des erreurs **ORA-01555 snapshot too old** sont retournées aux utilisateurs

Les données undo sont toujours conservées jusqu'à ce que la transaction se termine. Cela signifie que si une grande quantité de données est supprimé ou mise à jour (les opérations d'insertions consomment très peu d'espace undo car l'image initiale des données est à **NULL**) sans que les modifications soient périodiquement validées, le tablespace undo doit être suffisamment grand pour contenir l'image originale des données. Prenez par exemple le cas où toutes les lignes d'une table de 50 Go sont supprimées :

```
DELETE FROM reallybigtable;
```

Le tablespace undo aura besoin de 50 Go d'espace pour l'image initiale au cas où l'utilisateur veuille annuler ses modifications par un rollback. Quand le tablespace undo n'a pas assez d'espace pour stocker les données undo, l'erreur suivante est retournée aux utilisateurs :

```
ORA-01650: unable to extend rollback segment
```

Remarque : la maintenance proactive, vue précédemment, détecte les problèmes d'espace liés au tablespace undo avant que cela n'affecte les utilisateurs.

Un autre problème que le DBA peut rencontrer avec les informations undo est lorsqu'une requête tente d'accéder à des informations undo sur lesquelles d'autres informations ont été réécrites. Cela peut arriver pour une requête consommatrice de ressources ou une requête flashback. Quand une requête a besoin d'une capture (snapshot) des données à un moment donné dans le passé et que la reconstruction de la capture nécessite des données undo qui n'existe plus, alors l'erreur suivante est retournée:

```
ORA-01555: snapshot too old
```

Oracle recommande l'utilisation de la gestion automatique des undos en configurant le paramètre **UNDO_MANAGEMENT** à **AUTO**. La gestion manuelle des undos est supportée pour la compatibilité avec les versions Oracle 8i et précédentes, mais requiert une plus grande intervention du DBA. Avec une gestion automatique des undos, le DBA gère les undos au niveau des tablespaces et définit quel tablespace sera utilisé par une instance avec le paramètre d'initialisation **UNDO_TABLESPACE**. Après avoir choisit un tablespace undo, le DBA doit seulement vérifier qu'il dispose d'assez d'espace et configurer le temps de rétention des undos.

Avec une gestion manuelle des undos, le DBA doit également :

- Contrôler la taille des segments en incluant le nombre maximum d'extents et leur taille
- Identifier et supprimer les transactions bloquantes
- Créer suffisamment de segments de rollback (en mode manuel les segments undo sont connus sous le nom de segment de rollback) pour manipuler les transactions
- Choisir un tablespace pour stocker les segments de rollback (les undo tablespaces sont seulement utilisés pour la gestion automatique des undo)

10.2.2. Rétention des undos

10.2.2.1. Configuration de la rétention des undos

La rétention des undo est configurée avec le paramètre **UNDO_RETENTION**. Ce paramètre définit le temps pendant le lequel les informations undo ne peuvent pas être écrasées.

La gestion automatique des undo essaiera de conserver les données undo jusqu'à expiration, mais si une transaction active a besoin d'espace dans le tablespace undo, les données validées (non-expirées) seront écrasées sans se soucier du temps de rétention, pour permettre la restauration de la transaction si elle échoue.

Si le paramètre **UNDO_RETENTION** est initialisé à 0 alors la gestion de la rétention est gérée automatiquement. Dans ce mode, l'instance conserve autant d'informations undo que nécessaires pour satisfaire la plus longue requête en cours. Si le tablespace undo est trop petit pour répondre à la rétention des undo requise par la plus longue requête en cours, alors la gestion automatique de la rétention conserve aussi longtemps que possible les undo en utilisant l'espace disponible (sans étendre les fichiers de données undo) à moins que les informations retenues ne soit supprimées avant 900 secondes (15 minutes). La rétention automatique des undo garde si possible pendant au moins 15 minutes les informations undo sans qu'il y ait d'erreur au niveau de l'espace.

Les informations undo sont classées en trois catégories:

- Les informations undo non validées : concerne une transaction en cours et sont requises si l'utilisateur veut annuler la transaction ou si la transaction échoue. Les données non validées ne sont jamais écrasées.
- Les informations undo validées: les données undo ne sont plus nécessaires pour la transaction en cours, mais sont quand même conservées pour la rétention. Elles sont aussi connues sous le nom de "informations undo non expirées". Les données validées sont conservées lorsque c'est possible sans que cela ne fasse échouer une transaction active par manque d'espace.
- Les informations undo expirés: les données undo ne sont plus nécessaires pour la transaction en cours. Elles sont écrasées si de l'espace est requis pour une transaction active.

10.2.2.2. Garantir la rétention des undos

Le comportement par défaut des undos est d'écraser les transactions validées qui n'ont pas encore expiré plutôt que de permettre à une transaction d'échouer à cause du manque d'espace. Ce comportement peut être changé en garantissant la rétention. En garantissant la rétention, les paramètres de rétention des undos sont imposés même si la transaction échoue. **RETENTION GUARANTEE** est un attribut de tablespace et non un paramètre d'initialisation. Cet attribut ne peut être modifié qu'à partir d'ordres SQL.

La syntaxe utilisée pour changer un tablespace undo afin de garantir la rétention est la suivante :

```
ALTER TABLESPACE undotbs1 RETENTION GUARANTEE;
```

Pour que le tablespace retrouve ses paramètres initiaux, utilisez la commande suivante :

```
ALTER TABLESPACE undotbs1 RETENTION NOGUARANTEE;
```

La garantie de la rétention s'applique uniquement aux tablespaces undo. L'erreur suivante vous est renvoyée si vous essayez cet ordre SQL sur un tablespace non undo :

```
ALTER TABLESPACE example RETENTION GUARANTEE;
ERROR at line 1:
ORA-30044: 'Retention' can only specified for undo tablespace
```

Vous pouvez utiliser la vue **dba_tablespaces** pour obtenir des informations :

```
SELECT contents, retention
FROM dba_tablespaces
WHERE TABLESPACE_NAME = 'UNDOTBS1';
```

| CONTENTS | RETENTION |
|----------|-----------|
| UNDO | GUARANTEE |

10.2.3. Tablespaces undo

10.2.3.1. Taille des tablespaces undo

Les tablespaces undo doivent être dimensionnés de manière à stocker les informations originales pour toutes les transactions. Cliquez sur le lien Undo Management dans la page des propriétés d'administration d'Enterprise Manager pour obtenir une vue d'ensemble des undos incluant les paramètres de configuration actuels, la consommation des undos par minute et le temps d'exécution de la plus longue requête observée pendant une période donnée.

Les fichiers de données qui appartiennent à un tablespace undo peuvent être automatiquement étendus quand ils manquent d'espace. A la différence des autres tablespaces, Oracle recommande que les fichiers de données associés aux tablespaces undo ne soient pas configurés avec l'extension automatique.

Quand on détermine pour la première fois les besoins d'espace, vous pouvez activer l'extension automatique des fichiers de données, mais une fois le tablespace correctement dimensionné vous devriez désactiver cette option.

Désactiver l'extension automatique des fichiers dans un tablespace undo empêche un utilisateur d'utiliser par inadvertance beaucoup d'espace en négligeant de valider ses transactions.

Undo Management Undo Advisor

Configuration

| | | | |
|--------------------------|----------------|-----------------|--|
| Automatic Undo Retention | Enabled | Undo Tablespace | UNDOTBS1 Change Tablespace |
| Undo Retention | Automatic | Size (MB) | 485 ← Current Tablespace Size |
| Undo Retention Guarantee | No | Auto-Extensible | Yes |

Recommendations

Choose the time period that best represents the system activity to get the recommendations for undo retention length and undo tablespace size. Edit Undo Tablespace

Analysis Time Period: Last One Hour Update Analysis

Selected Analysis Time Period: **11/23/03 11:00 AM - 11/23/03 12:00 PM**

Potential Problems: **No Problem Found**

Recommendations: **No Recommendation**

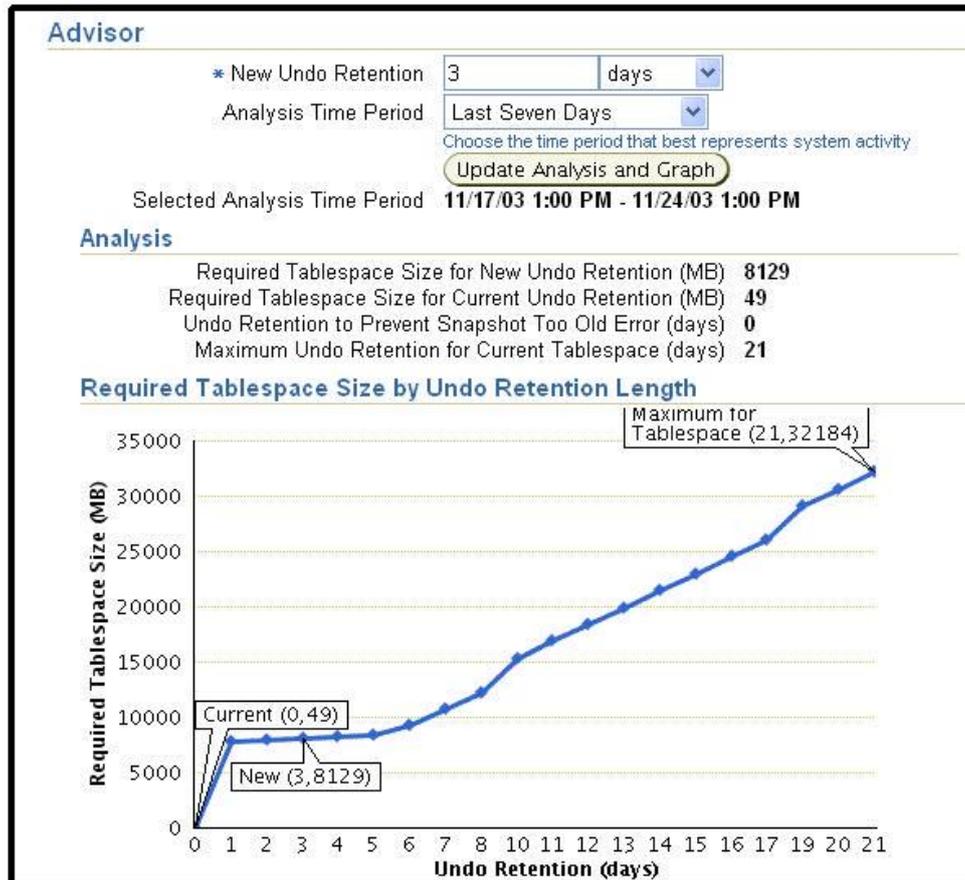
System Activity and Tablespace Usage

The recommendations are based on system activity and undo tablespace usage for the selected analysis time period.

| | |
|--|---|
| Longest Running Query (seconds) | 2 |
| Average Undo Generation Rate (KB/minute) | 29.0 |
| Maximum Undo Generation Rate (KB/minute) | 50.0 ← Undo Consumption Rate |

10.2.3.2. Utilisation de Undo Advisor

Undo Advisor est disponible à partir de la page Undo Management. Une estimation de la taille du tablespace undo requise pour satisfaire la rétention des undos vous est fournie. Entrez la période de rétention désirée et cliquez sur le bouton Update Analysis and Graph. La section Analysis affiche la taille du tablespace requise pour ce temps de rétention. Vous pouvez aussi cliquer sur un point du graphe pour sélectionner une période. Après que vous ayez sélectionné la période de rétention, cliquez sur OK pour implémenter la nouvelle période de rétention.



11. Module 11 : Sauvegarde et Récupération

11.1. Concepts

11.1.1. Présentation

Le rôle du DBA est de s'assurer que la base de données est ouverte et disponible pour tout utilisateur qui en a besoin :

- Anticiper et prévenir les dysfonctionnements les plus courants.
- Travailler pour augmenter Mean-Time-Between-Failures **MTBF** (le temps moyen entre les échecs), en s'assurant que le matériel est aussi fiable que possible, que les données critiques sont protégés par de la redondance, et que la maintenance du système d'exploitation est faite aux moments opportuns.

Oracle fournit des options de configuration avancée pour augmenter le MTBF incluant :

- Real Application Clusters
- Streams
- Diminuer Mean-Time-To-Recover **MTTR** (le temps moyen de restauration), grâce à des procédures de restauration anticipées et en configurant les sauvegardes pour qu'elles soient disponibles au moment voulu.
- Minimiser la perte de données. Les DBA peuvent configurer leurs bases de données de manière optimale afin d'éviter toute perte de données validée (**COMMIT**);

Les outils fournis dans cet optique sont :

- L'archivage des redo log
- Les bases de données standby et Oracle Data Guard (vu dans un prochain module)

11.1.2. Echecs

11.1.2.1. Types d'échecs

Les dysfonctionnements peuvent être répartis en plusieurs catégories :

- **Echec d'un requête**: Une opération sur les données (select, insert, update, delete) échoue.
- **Echec d'un user process**: une session échoue.
- **Echec réseau**: la connectivité à la base de données est perdue.
- **Erreur humaine**: un utilisateur réalise une opération, mais celle-ci était incorrecte (suppression d'une table, entrée de données incorrectes).
- **Echec de l'instance**: l'instance de la base s'arrête de façon inattendue.
- **Echec d'un media**: un ou plusieurs fichiers de donnée sont perdus (supprimés, crash d'un disque).

11.1.2.2. Echecs de requêtes

Quand une opération sur les données échoue, le DBA devra intervenir pour corriger les erreurs concernant les privilèges de l'utilisateur ou l'allocation d'espace pour cet utilisateur au niveau de la base de données.

| Problèmes récurrents | Solutions à envisager |
|---|--|
| Des données invalides ont été enregistrées dans une table | Travailler avec les utilisateurs pour valider et corriger les données |
| Effectuer des opérations avec des privilèges insuffisants | Fournir les privilèges objets ou systèmes appropriés |
| L'allocation d'espace échoue | Activer la reprise après un problème d'allocation d'espace Augmenter les quotas des utilisateurs Ajouter de l'espace à un tablespace |
| Les erreurs logiques dans les applications | Travailler avec les développeurs pour corriger les erreurs dans les programmes |

11.1.2.3. Echecs de processus utilisateurs

Les utilisateurs qui ont été déconnectés anormalement de l'instance peuvent ne pas avoir validé leur travail en cours ; les ressources utilisées par les processus utilisateur doivent être libérées. Le processus d'arrière plan **PMON** scrute périodiquement les processus server pour s'assurer que leurs sessions sont toujours connectées. Si **PMON** trouve un processus server dont l'utilisateur n'est plus connecté, **PMON** récupère les transactions, annule les transactions non validées (rollback), et libère tous les verrous créés par la session qui a échoué.

L'intervention du DBA n'est normalement pas nécessaire suite aux échecs de processus utilisateur, mais il doit garder un œil sur cette opération. Un ou deux utilisateurs déconnectés anormalement n'est pas un sujet d'inquiétude. Un faible pourcentage de processus utilisateur qui échouent est normal. Des problèmes systématiques à ce niveau peuvent indiquer d'autres problèmes. Un grand nombre de déconnexions anormales signale peut être le besoin de montrer aux utilisateurs comment se déconnecter soigneusement plutôt que de terminer brutalement leur programme. Cela peut aussi indiquer des problèmes au niveau du réseau ou des applications.

| Problèmes récurrents | Solutions à envisager |
|---|---|
| Les utilisateurs se déconnectent anormalement | L'intervention du DBA n'est pas systématiquement nécessaire pour résoudre les problèmes des processus utilisateur. Les processus d'arrière plan annulent les transactions non validées et libèrent les ressources |
| La session d'un utilisateur s'est terminée anormalement | |
| Suite à l'exécution d'un programme, la session d'un utilisateur s'est terminée anormalement | |
| | Observer les tendances |

11.1.2.4. Echecs de réseau

La meilleure solution pour résoudre les problèmes réseaux est de créer des chemins redondants pour l'acheminement des données. Les listeners de secours, les connexions réseaux, et les interfaces réseaux diminuent le nombre de problème réseau qui pourrait survenir et affecter la disponibilité du système.

| Problèmes récurrents | Solutions à envisager |
|-------------------------------|--|
| Le listener échoue | Configurer un listener de secours avec basculement automatique |
| La carte réseau à un problème | Configurer plusieurs cartes réseau |
| La connexion réseau échoue | Configurer une connexion réseau de secours |

11.1.2.5. Erreurs d'utilisateurs

Parfois, les utilisateurs suppriment ou modifient des données par inadvertance. Quand cela arrive, l'utilisateur peut avoir besoin d'être assisté par le DBA pour corriger les erreurs. Si l'utilisateur n'a pas encore validé ou quitté le programme, ils peuvent tout simplement annuler la transaction par un **ROLLBACK**. Si l'utilisateur a déjà validé les changements, les requêtes flashback peuvent être utilisées pour déterminer la valeur des données avant validation (**COMMIT**) (ainsi les données peuvent être mises à jour pour restaurer les données originales).

```
SELECT salary
FROM employees
WHERE employee_id = 100;
```

| SALARY |
|--------|
| 25 |

```
SELECT salary
FROM employees AS OF TIMESTAMP(SYSTIMESTAMP-INTERVAL '10' minute)
WHERE employee_id=100;
```

| SALARY |
|--------|
| 24000 |

Si l'utilisation des requêtes flashback n'est pas possible à cause de la période de rétention des undo, le DBA peut quand même restaurer les données initiales grâce à l'outil Oracle LogMiner.

LogMiner vous permet d'interroger les redo log online et les redo log archivés grâce à des commandes SQL. Les données des transactions se trouvent plus longtemps dans les redo log online que dans les undos, et si vous avez configuré l'archivage des informations de redo, alors les redos sont disponibles jusqu'à ce que vous supprimiez les fichiers.

Les utilisateurs qui suppriment des tables peuvent restaurer ces tables à partir de la corbeille en récupérant les informations de la table avant qu'elle n'ait été supprimée.

```
TABLE hr.job_history;
Table dropped.
```

```
SELECT COUNT(*) FROM hr.job_history;
```

```
SELECT COUNT(*) FROM hr.job_history
*
ERROR at line 1:
ORA-00942: table or view does not exist
```

```
FLASHBACK TABLE hr.job_history TO BEFORE DROP;
Flashback complete.
```

```
SELECT COUNT(*) FROM hr.job_history;
```

| COUNT(*) |
|----------|
| 10 |

Si la corbeille a déjà été purgée ou si l'utilisateur a supprimé la table avec l'option **PURGE**, la table supprimée peut quand même être restaurée en utilisant le Point-In-Time-Recovery (**PITR**) si la base de données a été configurée correctement.

| Problèmes récurrents | Solutions à envisager |
|---|--|
| Un utilisateur a supprimé ou modifié des données par inadvertance | Annuler la transaction ou utiliser les requêtes flashback |
| Un utilisateur supprime une table | Restaurer la table à partir de la corbeille (recycle bin) |

11.1.2.6. Echecs d'instances

Un échec d'instance se produit quand l'instance est arrêtée avant que tous les fichiers de la base de données ne soient synchronisés. Cela peut être dû à des problèmes matériels ou logiciel, ou à cause de l'utilisation des commandes d'arrêt d'urgence **SHUTDOWN ABORT** et **STARTUP FORCE**.

Le rôle du DBA dans ce cas de figure est limité, car ce dernier devra simplement redémarrer l'instance et travailler pour éviter ce type de problème à l'avenir.

| Problèmes récurrents | Solutions à envisager |
|---|--|
| Panne électrique | Redémarrer l'instance en utilisant la commande STARTUP . La restauration de l'instance est automatique et inclue les opérations Rolling Forward (applique les modifications validées et non validées) et Rolling Back (annule les modifications non validées). Chercher les causes de ce dysfonctionnement en utilisant les fichiers d'alertes, les fichiers de trace, et Enterprise Manager. |
| Panne matérielle | |
| Dysfonctionnement d'un des processus d'arrière plan | |
| Exécution d'une procédure d'arrêt d'urgence | |

11.1.2.7. Echecs Média

Oracle définit un échec média par n'importe quelle perte ou corruption d'un ou plusieurs fichiers de la base de données (données, contrôle, redo log). Pour palier à ce problème vous devez restaurer et récupérer les fichiers manquants.

| Problèmes récurrents | Solutions à envisager |
|---|---|
| Dysfonctionnement d'un disque | <ol style="list-style-type: none"> 1. Restaurer les fichiers affectés à partir de sauvegardes 2. Si nécessaire, informer la base de données d'un nouvel emplacement pour les fichiers désignés 3. Si nécessaire, restaurer les fichiers en appliquant les informations de redo |
| Dysfonctionnement d'un contrôleur de disque | |
| Suppression ou corruption de fichiers | |

11.2. Récupération d'une instance

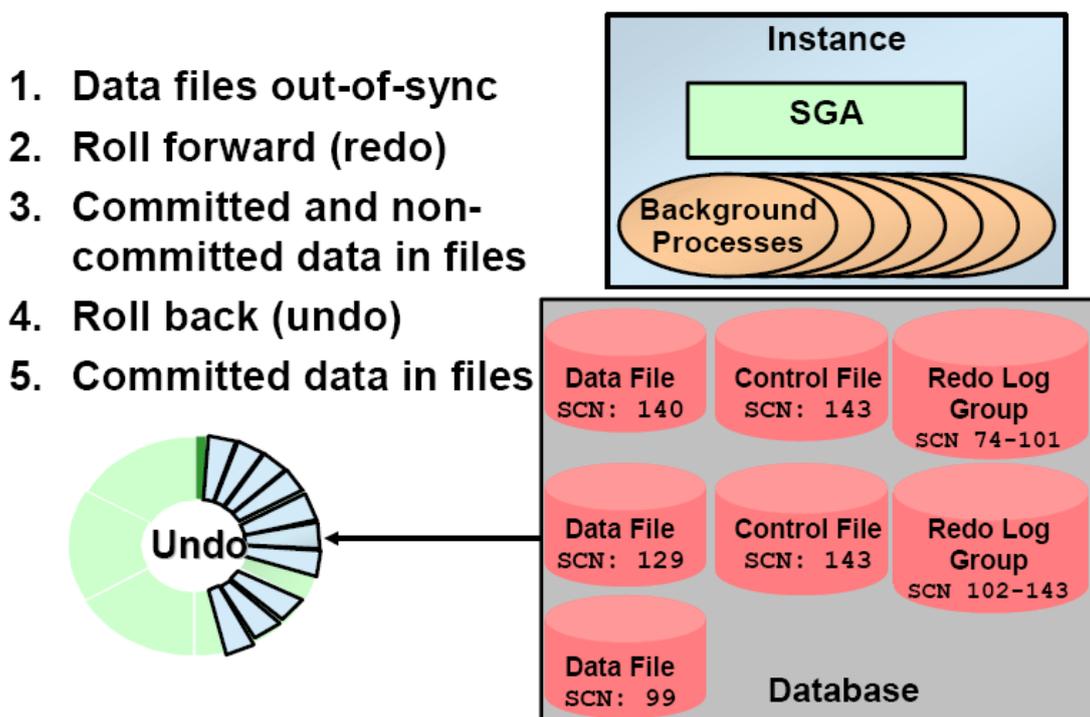
11.2.1. Concepts de récupération

La récupération de la base de données Oracle 10g est automatique lors de l'échec de l'instance. Tout ce qu'a à faire le DBA est de démarrer l'instance normalement. L'instance montera les fichiers de contrôle et essaiera d'ouvrir les fichiers de données. Lorsque que l'instance découvre que les fichiers de données n'ont pas été synchronisés pendant l'arrêt de la base, elle utilise les informations contenues dans les groupes de redo log pour mettre à jour la base comme elle était au moment où l'instance s'est arrêtée (**Rolling Forward**) et annule toutes les transactions qui n'ont pas été validées (**Rolling Back**).

Pour que l'instance puisse ouvrir un fichier de données, le System Change Number (**SCN**) contenu dans l'en-tête des fichiers de données doit correspondre au SCN enregistré dans le fichier de contrôle de la base de données.

Si les SCN ne correspondent pas, l'instance applique les redo à partir des redo log online, et rejoue les transactions de manière séquentielle jusqu'à ce que les fichiers de données soient à jour. Après que les fichiers aient été synchronisés avec les fichiers de contrôle, la base de données est ouverte et les utilisateurs peuvent ouvrir une session.

Quand les redo sont appliqués, toutes les transactions sont appliquées pour que la base de données soit dans le même état qu'elle était au moment du dysfonctionnement. Cela inclut en général les transactions qui étaient en cours et qui n'ont pas été validées. Après que la base de données ait été ouverte, ces transactions sont annulées. A la fin de cette phase de rollback pour la récupération de l'instance, les fichiers de données contiennent uniquement les données validées.

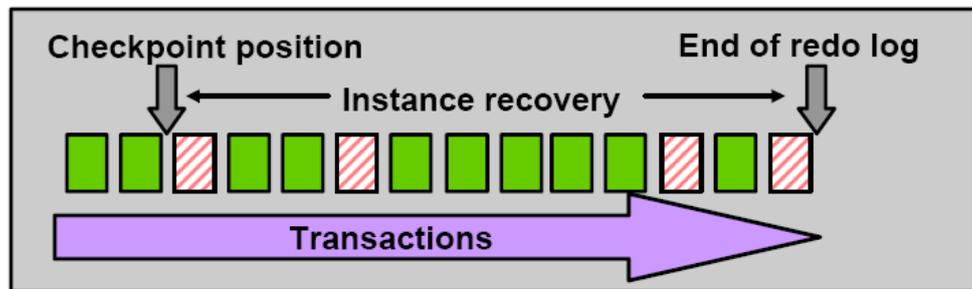


11.2.2. Tuning

Les informations sur les transactions sont toujours enregistrées dans les groupes de redo log avant que l'instance ne retourne le message **Commit complete** à l'utilisateur. Les informations dans les groupes de redo log garantissent que la transaction peut être récupérée en cas de dysfonctionnement. Les informations de cette même transaction doivent aussi être écrites dans les fichiers de données. L'écriture dans les fichiers de données arrive souvent après que les informations sur la transaction aient été enregistrées dans les groupes de redo log car le processus d'écriture dans les fichiers de

données est plus lent que celui d'écriture dans les redo (l'écriture aléatoire dans les fichiers de données est plus lente que l'écriture séquentielle dans les fichiers de redo log).

Pour garder une trace de ce qui a déjà été écrit dans les fichiers de données, la base de données utilise des points de sauvegarde (checkpoints). Un checkpoint garantit qu'au moment où celui-ci s'effectue, tous les fichiers de données sont synchronisés avec le même SCN. Les transactions après la position du point de sauvegarde peuvent ou non avoir été écrites dans le fichier de données approprié. Dans le graphique ci-dessous, les blocks rayés n'ont pas encore été écrits sur le disque.



Le temps requis pour la récupération de l'instance correspond au temps requis pour mettre à jour les fichiers de données à partir de leur dernier checkpoint jusqu'au dernier SCN enregistré dans le fichier de contrôle. Le DBA définit le temps imparti à cette opération en configurant le paramètre `MTTR target` (en secondes), et selon la taille des groupes de redo log.

L'intervalle entre la position d'un checkpoint et la fin des groupes de redo log ne peut jamais être supérieur à 90% du groupe de redo log le plus petit.

Cliquez sur `MTTR Advisor` à partir de la section `Advisor Central` de `Enterprise Manager` pour configurer le `MTTR target`. L'Advisor vous permet de spécifier le temps pour le `Mean-Time-To-Recover` afin de configurer le paramètre d'initialisation `FAST_START_MTTR_TARGET`.

La valeur par défaut 0 désactive le `MTTR target`, et réduit la probabilité qu'il faille d'attendre que l'écriture dans les fichiers de données soit effectuée pour écrire dans les groupes de log. `MTTR target` devrait être configuré pour pouvoir satisfaire un niveau d'équilibre pour le système.

Donner une valeur trop faible à `MTTR target` implique que l'écriture dans les groupes de redo attendra l'écriture dans les fichiers de données (ce qui affecte les performances). Donner une valeur trop grande à `MTTR target` implique que l'instance mettra plus de temps à être restaurée suite à un crash.

Advisor Central

Page Refreshed Dec 1, 2003 5:09:54 AM Refresh

Advisors

| | | |
|------------------------------------|--------------------------------|---------------------------------|
| ADDM | Memory Advisor | Segment Advisor |
| SQL Tuning Advisor | MTTR Advisor | Undo Management |
| SQL Access Advisor | | |

Instance Recovery

The `FAST_START_MTTR_TARGET` initialization parameter specifies the number of seconds estimated for crash recovery. Oracle converts this number into a set of internal parameters and sets the recovery time as close as possible to these parameters. Setting `FAST_START_MTTR_TARGET` to 0 will disable this functionality.

Current Estimated Mean Time To Recover (seconds) **13**

Desired Mean Time To Recover Minutes ▼

Remarque :

- Spécifiez le temps choisi en secondes ou minutes.
- La valeur par défaut est 0 (désactivé).
- La valeur maximale est de 3600 secondes (une heure).

11.3. Configuration de la récupération

11.3.1. Règles à suivre

Pour une protection optimale de vos données vous devez :

- Planifier des sauvegardes régulièrement. La plupart des dysfonctionnements matériels impliquent que vous devez restaurer les fichiers manquants ou corrompus à partir de sauvegardes.
- Multiplexer les fichiers de contrôle. Tous les fichiers de contrôle associés à une base de données sont identiques. La restauration d'un fichier contrôle est une opération simple et rapide. La restauration d'un fichier de contrôle si tous les fichiers de contrôle ont été perdus est une opération plus délicate. Pour prévenir ces problèmes, utilisez plusieurs multiplexés (au moins trois).
- Multiplexer les groupes de redo log. Pour récupérer la base suite à un échec d'instance ou de média, les informations de redo log sont utilisées pour le rolling forward des données jusqu'à la dernière transaction validée. Si vos groupes de redo log se basent sur un fichier unique, alors la perte de ce fichier entraîne la perte de données. Assurez vous de disposer d'au moins deux copies par groupe de redo log.
- Conserver les groupes de redo log archivés. Si un fichier de données est perdu et restauré à partir d'une sauvegarde, l'instance doit appliquer les informations de redo pour que les fichiers de données soient synchronisés avec le dernier SCN enregistré dans le fichier de contrôle. La configuration par défaut est d'écraser les informations contenues dans les redo une fois que ces informations ont été enregistrées dans les fichiers de données. Votre base de données peut être configurée pour conserver les informations de redo en archivant des copies des groupes de redo log. Cette possibilité peut être effectuée en paramétrant la base en mode **ARCHIVELOG**.

11.3.2. Fichiers de contrôle

Le fichier de contrôle est un petit fichier binaire qui décrit la structure de la base de données. Il doit être disponible pour l'écriture par le serveur Oracle quand la base est montée ou ouverte. Sans ce fichier, la base de données ne peut pas être montée et la restauration ou la recréation du fichier de contrôle est nécessaire. Votre base de données devrait disposer d'au moins deux fichiers de contrôle (trois recommandés) sur des disques différents pour minimiser l'impact de la perte d'un fichier de contrôle.

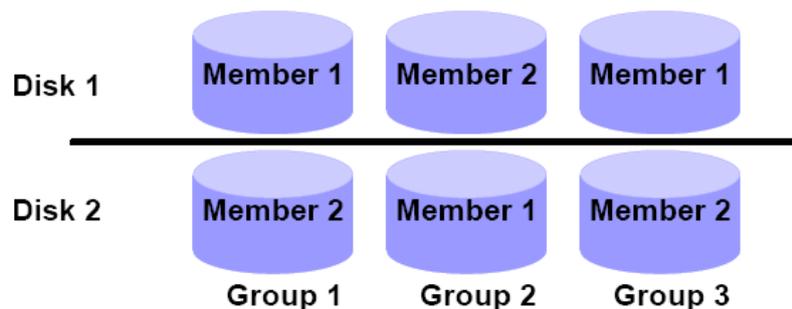
Si votre base de données a été créée avec DBCA, vous devriez déjà disposer de trois fichiers de contrôle. La perte d'un seul fichier de contrôle entraînera le dysfonctionnement de l'instance car les fichiers de contrôle doivent être tous disponibles, mais la restauration de l'un d'eux se fait simplement en copiant un autre des fichiers de contrôle au niveau de l'OS. La perte de tous les fichiers de contrôle est plus problématique pour la restauration, mais pas forcément catastrophique.

11.3.3. Fichiers de Redo Log

Les groupes de redo log se composent d'un ou plusieurs fichiers de redo log. Chaque fichier de redo log dans un groupe est identique aux autres fichiers du même groupe. Oracle recommande d'utiliser au moins deux fichiers par groupe en répartissant les fichiers sur des disques/contrôleurs différents pour qu'un dysfonctionnement matériel ne détruise pas un groupe de redo log tout entier.

La perte d'un groupe entier est le problème le plus critique car il peut en résulter la perte de données. La perte d'un simple membre (fichier appartenant à un groupe) dans un groupe composé de plusieurs membres est négligeable, car elle n'empêchera pas les opérations sur la base de données (une erreur sera signalée dans le fichier de log d'alerte).

Rappelez vous que les redo ont une influence considérable sur les performances de la base de données car un commit ne peut se terminer tant que les informations concernant la transaction n'ont pas été écrites dans les redo log. Vous devriez stocker vos fichiers de redo log sur les disques les plus rapides de vos contrôleurs les plus puissants. Si possible, ne placez pas d'autres fichiers de la base sur ces mêmes disques. Comme il n'y a qu'un seul groupe en écriture à la fois, il n'y a pas de problème à ce que vous stockiez les membres de différents groupes sur le même disque.

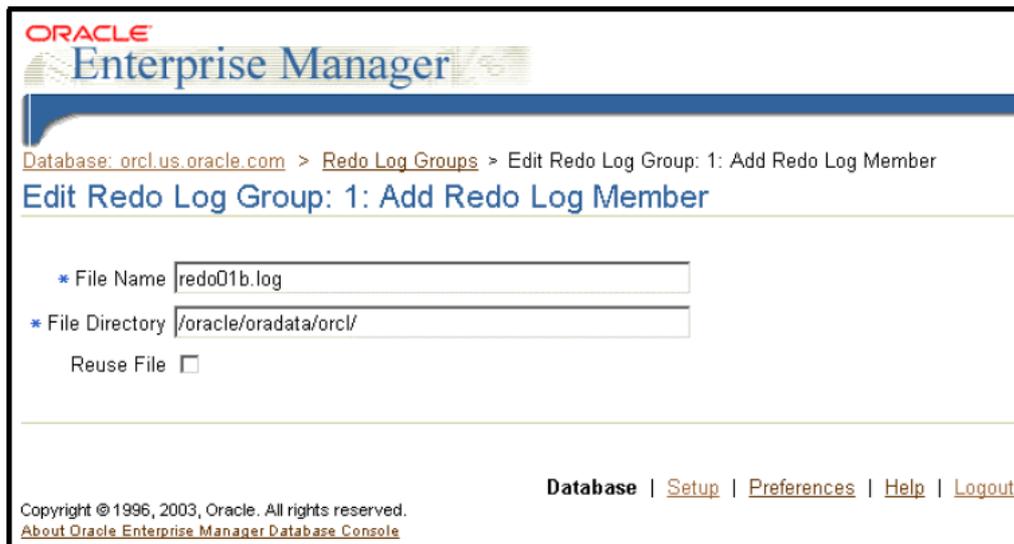


11.3.4. Multiplexage

Vous pouvez multiplexer les logs de redo en ajoutant un ou plusieurs membres à un groupe de logs existant. Exécutez les instructions suivantes pour ajouter un membre à un groupe, ceci peut être fait alors que la base de données est ouverte sans que cela n'affecte les performances :

1. Accédez à la page Redo Log Groups.
 2. Choisissez un groupe et cliquez sur le bouton Edit, ou cliquez sur le lien group number. La page Edit Redo Log Group s'affiche.
 3. Dans la section Redo log Member, Cliquez sur Add. La page Add Redo Log Member s'affiche.
 4. Spécifiez le nom du fichier et le répertoire. Cliquez sur OK.
- Remarque :** Il est recommandé que vous stockiez les membres sur des disques séparés pour ne pas perdre des données redo en cas de défaillance d'un disque.

Répétez cette opération pour tous les groupes.



ORACLE
Enterprise Manager

Database: orcl.us.oracle.com > Redo Log Groups > Edit Redo Log Group: 1: Add Redo Log Member

Edit Redo Log Group: 1: Add Redo Log Member

* File Name

* File Directory

Reuse File

Database | [Setup](#) | [Preferences](#) | [Help](#) | [Logout](#)

Copyright © 1996, 2003, Oracle. All rights reserved.
[About Oracle Enterprise Manager Database Console](#)

Quand vous ajoutez un membre à un groupe, le statut du groupe est marqué **INVALID** (car le membre du groupe n'a pas encore été utilisé). Quand un changement de log est effectué et que le groupe **INVALID** devient le groupe courant, le statut est marqué **CURRENT**.

11.3.5. Fichiers de Redo Log archivés

11.3.5.1. Présentation

L'instance utilise les groupes de redo log online de façon circulaire pour enregistrer les informations sur les transactions, en remplissant un groupe et en passant ensuite au suivant. Après que tous les groupes aient été remplis, l'instance écrase les informations contenues dans le premier groupe.

Pour configurer la base de données afin d'obtenir une efficacité maximale pour la restauration, vous devez configurer la base de données pour garder une copie des groupes avant qu'ils ne soient écrasés. Ces copies sont connues sous le nom de logs archivés. Pour faciliter l'archivage des logs vous devez :

1. Adopter une convention de nommage pour les logs archivés.
2. Spécifier une ou plusieurs destinations pour stocker les logs archivés.
3. Mettre la base de données en mode **ARCHIVELOG**.



Remarque : Le répertoire de destination doit exister avant que la base de données ne soit placée en mode **ARCHIVELOG**. Quand un répertoire est spécifié comme destination, il doit y avoir un slash à la fin du chemin.

11.3.5.2. Nommage et chemins des fichiers de Log archivés

Configurez le nom et le chemin des fichiers de log archivés en cliquant sur Configure Recovery Settings à partir de la page Maintenance.

Chaque fichier de log archivé doit avoir un nom unique pour éviter d'écraser des anciens fichiers de log. Vous pouvez spécifier le format comme montré ci-dessous. Pour vous permettre de créer des fichiers dont le nom est unique, Oracle propose différents caractères spéciaux pour le format du nom :

- **%s** : Ajoute le numéro de séquence au nom du fichier
- **%t** : Ajoute le numéro du thread au nom du fichier
- **%r** : Resetlogs ID. S'assure que les fichiers de log archivés restent uniques même après certaines opérations de restauration avancées qui réinitialisent le numéro de séquence des logs
- **%d** : Ajoute d'identifiant de la base de données au nom du fichier

Log Archive Filename Format*

The naming convention for the archived log files. %s: log sequence number; %t: thread number; %S and %T: padding the filename to the left with zeroes.

| Number | Archive Log Destination | Quota (512B) | Status | Type |
|--------|---------------------------|--------------|--------|-------|
| 1 | /oracle/ARCHIVE/ | 0 | VALID | Local |
| 2 | | | | Local |
| 3 | | | | Local |
| 4 | | | | Local |
| 5 | | | | Local |
| 6 | | | | Local |
| 7 | | | | Local |
| 8 | | | | Local |
| 9 | | | | Local |
| 10 | USE_DB_RECOVERY_FILE_DEST | n/a | VALID | Local |

TIP It is recommended that archive log files be written to multiple locations spread across the different disks.

TIP You can specify up to 10 archive log destinations.

Le format doit inclure les caractères spéciaux **%s**, **%t**, et **%r**. L'utilisation de **%d** est optionnelle mais devrait être incluse si plusieurs bases de données partagent le même chemin de destination pour les fichiers de log archivés.

Les fichiers de log archivés peuvent être écrits dans dix destinations différentes. Les destinations peuvent être locales (un répertoire) ou distantes (un alias Oracle Net pour une base de données standby). Les destinations locales doivent se terminer par un slash (/), ou un backslash (\) si vous utilisez Windows.

La destination par défaut (nombre 10) envoie les fichiers de log archivés à un emplacement déterminé par le paramètre d'initialisation **DB_RECOVERY_FILE_DEST**. **DB_RECOVERY_FILE_DEST** est aussi appelé flash recovery area. Cette destination est visible en bas de la page Configure Recovery Settings en tant que Flash Recovery Area Location. Si vous ne voulez pas envoyer les archives à cet emplacement, supprimez simplement **USE_DB_RECOVERY_FILE_DEST**.

Pour changer les paramètres de restauration vous devez être connecté en tant que **SYSDBA** ou **SYSOPER**.

11.3.5.3. Mode ARCHIVELOG

Configurer la base de données en mode **ARCHIVELOG** évite que les fichiers de redo log soient écrasés tant qu'ils ne sont pas archivés.

La commande SQL pour configurer la base de données en mode **ARCHIVELOG** est la suivante :

```
ALTER DATABASE ARCHIVELOG;
```

Cette commande peut être exécutée seulement quand la base de données est dans l'état **MOUNT**, l'instance doit donc être redémarrée pour réaliser cette dernière étape. Pour redémarrer la base de données les informations de connexion (nom d'utilisateur et mot de passe) au niveau de l'OS vous seront demandées. Vous devez disposer du privilège **SYSDBA**.

Après que l'instance ait été démarrée, les modifications que vous avez effectuées pour le processus d'archivage, le format des logs, et la destination des logs seront prises en compte.

Quand la base de données est en mode **NOARCHIVELOG**, (mode par défaut), la restauration est possible uniquement jusqu'au moment de la dernière sauvegarde. Toutes les transactions opérées après cette sauvegarde sont perdues.

En mode **ARCHIVELOG**, la restauration est possible jusqu'au moment du dernier commit. La plupart des bases de données d'entreprises sont configurées en mode **ARCHIVELOG**.

Pour mettre la base de données en mode **ARCHIVELOG** procédez comme suit :

- Cliquez sur la case à cocher ARCHIVELOG mode
- Cliquez sur Apply. Le mode **ARCHIVELOG** peut seulement être activé lorsque la base de données est en mode **MOUNT**.
- Cliquez sur Yes lorsque cela vous l'est demandé si vous voulez redémarrer la base de données

Media Recovery

The database is currently in NOARCHIVELOG mode. In ARCHIVELOG mode, hot backups and recovery to the latest time is possible, but you must provide space for logs. If you change the database to ARCHIVELOG mode, you should make a backup immediately. In NOARCHIVELOG mode, you can make only cold backups and data may be lost in the event of database corruption.

ARCHIVELOG Mode*