Bases de Données

Oracle : administration (cours)

Rappelons que l'administrateur d'une base de données (*Data Base Administrator*) a pour rôles la définition des objets d'une part et veiller à leur bonne utilisation d'autre part. Cela recouvre un rôle organisationnel (gestion des utilisateurs, gestion des droits d'accès, création des vues au niveau externe ; participation à la conception des données au niveau conceptuel) et technique (pour assurer la correspondance entre le schéma conceptuel et le système d'exploitation au niveau interne, garantissant les performances, l'intégrité et la sécurité des données). Le rôle technique comprend :

- l'installation du SGBD (Système de Gestion de Bases de Données) et des outils associés,
- la création de la base de données et son évolution,
- la gestion des privilèges d'accès par l'attribution ou le retrait de droits aux différents utilisateurs,
- l'amélioration des performances par une implantation optimale tenant compte de l'utilisation qui est faite par les traitements de la base de données,
- la sécurité et la cohérence des données par la mise en place des structures et procédures permettant de faire face à tout incident et de retrouver une base de données intègre et cohérente,
- l'échange de données entre la base de données et le monde extérieur, soit par intégration de données en provenance d'autres applications ou bases de données, soit par migration de données vers d'autres applications ou bases de données.

Ce document présente l'essentiel de l'architecture et de l'administration du SGBDRO (SGBD Relationnel-Objet) Oracle ORACLE, logiciel leader mondial des SGBD, commercialisé par la société Oracle Corporation (dirigée par Larry Ellison). Une première partie présente la version 7 d'Oracle tandis que la seconde partie présente certaines des nouveautés des versions plus récentes d'Oracle. Ce document est complété d'une série d'exercices couvrant largement les différents aspects de l'administration d'Oracle ; pour effectuer correctement ces exercices, la lecture (et même la relecture) de ce document est indispensable.

Oracle est un SGBDRO permettant de gérer des bases de données jusqu'à 65 536 fichiers de 128 To (téra-octets i.e. 10^{12} octets) chacun soit jusqu'à quelques trillions d'octets.

Oracle est essentiellement écrit en langage C et certains codes sont actuellement écrits ou réécrits en Java. Oracle est ainsi portable sur près d'une centaine de plate-formes matérielles (des très gros ordinateurs aux micro-ordinateurs) et sur une soixantaine de systèmes d'exploitation (HP-UX, Linux, Mac OS X, Microsoft Windows, Sun Solaris, IBM AIX, IBM z/OS, etc.).

La gamme Oracle se décline en deux principales licences :

- Standard Edition (version de base) ;
- Enterprise Edition (version de référence, avec toutes les fonctionnalités).

D'autres licences existent ou ont existé : Oracle Workstation (version mono-utilisateur pour Windows d'Oracle Enterprise Edition), Oracle Lite (pour travailler sur une base allégée et nomade), Oracle Standard One (version standard limitée à quatre processeurs), etc.
Il est également possible d'acquérir certaines options : Partitionning, OLAP, Data Mining, Real Application Cluster, Advanced Security.
Depuis fin 2005, Oracle offre une version gratuite : Oracle Database Express Edition (version 10g Standard Edition One limitée : base de 4 Go maximum, mémoire d'1 Go de mémoire et 1 processeur actif).

L'intégralité du produit Oracle est constituée de la base de données (RDBMS), le serveur d'application (IAS), la suite collaborative (OCS), l'environnement de développement (ODS) et la suite applicative (eBusiness suite).
L'achat d'Oracle Enterprise Developer Suite vous permet d'avoir Oracle Designer, Oracle Developer, Oracle Developer Server, Oracle Application Server et Oracle Database Server.

La version 11g d'Oracle, commercialisée depuis le 11 juillet 2007, est la plus récente (au moins au moment où ces lignes sont rédigées c'est-à-dire à la fin du mois de novembre 2007).
Voici l'historique des sorties des versions précédentes : 1979 pour le premier prototype (assembleur, séparation des espaces d'adressage entre les programmes utilisateurs et le noyau Oracle) nommé RDBMS - RSI1 comme *Relational Database Management System* et *Relational Software Incorporated* (premier nom de la société Oracle Corporation) mais pas de 1, 1979 pour la 2 (portage sur d'autres plates-formes, SQL : requête et jointure), 1983 pour la 3 (entièrement-ré-écrite en C, transaction), 1984 pour la 4 (cohérence en lecture), 1985 pour la 5 (client/serveur et requête distribuée avec SQL*Net), 1988 pour la 6 (un Progiciel de Gestion Intégré, PL/SQL, verrouillage de lignes, sauvegarde à chaud, amélioration des performances, option Parallel Server), 1992 pour la 7 (contrainte d'intégrité, procédure stockée et déclencheur, meilleure gestion mémoire et CPU et entrées-sorties, SQL*DBA) dénommée Oracle Universal Server pour la 7.3, 1997 pour la 8 (orienté-objet, multimédia) et 1999 pour la 8i (internet), 2001 pour la 9i (400 nouvelles fonctionnalités, XML, Online Analytical Processing (OLAP) i.e. bases de données multidimensionnelles), et 2003 pour la 10g (grid computing).

De nombreuses explications sont fournies dans les deux ouvrages suivants qui constituent des références pour tout administrateur d'une base de données gérée par Oracle :

- toute la documentation d'Oracle (plus de 16000 pages, en langue anglaise) : **incontournable !**
- A. ABDELLATIF, M. LIMAME et A. ZEROUAL, *Oracle7 : Langages – Architecture – Administration*, EYROLLES, 1994 : **indispensable !**

Ces ouvrages ont largement contribué à la réalisation de ce mini-cours d'administration d'Oracle et des exercices associés.
Quelques sites Internet et groupes de discussion (*news*) sont également conseillés :

- www.osborne.com/oracle/index.htm : collection (de livres) officielle d'Oracle présentée par l'éditeur Oracle Press (Osborne/McGraw-Hill)
- www.oracle.com : site officiel d'Oracle
- www.oracle.fr : site français d'Oracle
- home.nordnet.fr/~bdesquesne/index.html : site indépendant français de B. Desquesne sur Oracle
- www.onwe.co.za/frank/faq.htm : site indépendant proposant des réponses aux questions les plus couramment posées sur Oracle ainsi que des pointeurs sur d'autres sites sur Oracle
- `comp.databases.oracle.`◁`extension`▷ : groupe de discussion sur Oracle (moteur, outils, etc.)
- `comp.databases` : groupe de discussion sur les SGBD

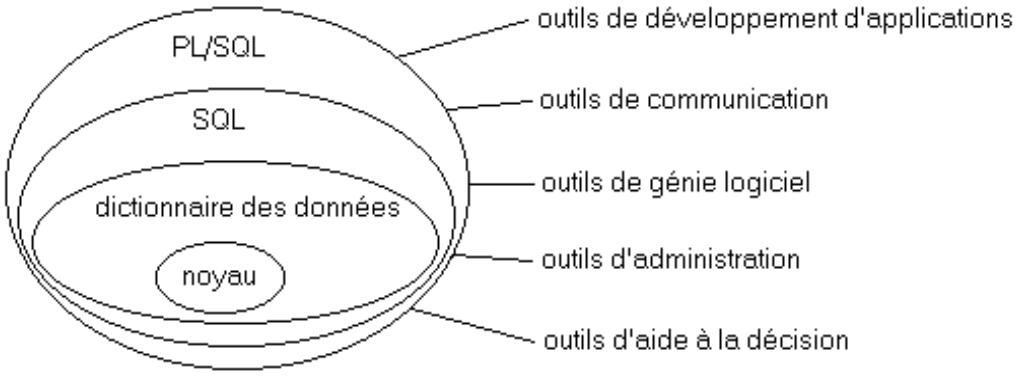
1ère partie : Architecture et administration d'Oracle7

Cette partie présente tout d'abord l'architecture du SGBDR Oracle7 (architectures fonctionnelle et interne) et ensuite l'administration (création et démarrage de la base de données, transactions et accès concurrents, sécurité des données, sauvegarde et restauration des données, l'optimisation et la présentation de quelques outils d'administration).

1. Architecture fonctionnelle

L'architecture fonctionnelle d'Oracle repose sur un modèle en couches : les applications se servent d'outils (de développement d'applications, de communication, de génie logiciel, d'administration ou d'aide à la décision) qui utilisent directement SQL ou qui s'appuient sur le PL/SQL (*Programming Language* / SQL), langage procédural d'Oracle qui transforme ses instructions en SQL, les ordres SQL sollicitant finalement le moteur (ou noyau) d'Oracle qui quant à lui accède au dictionnaire des données.

1.1. Principe général



1.2. Description

- noyau : communication avec la base de données, connexion à d'autres noyaux (cas d'une base répartie)
  - intégrité et cohérence
  - confidentialité
  - sauvegarde et restauration
  - gestion des accès concurrents
  - optimisation de l'exécution des requêtes (analyse, optimisation, exécution)
  - gestion des trois accélérateurs (index, clusters, hash clusters)
  - stockage physique des données
- dictionnaire des données : métabase (stockée sous la forme d'un schéma relationnel) décrivant de façon dynamique la base de données (i.e. structure centralisée contenant la description de tous les objets gérés par Oracle)
  - objets de la base :
    - ✓ schéma : ensemble de structures logiques de données (cluster, lien de base de données, index, paquetage, procédure, séquence, cliché, journal de cliché, table, vue, déclencheur, fonction)
    - ✓ cluster (ou groupement) : contient une ou plusieurs tables ayant une ou plusieurs colonnes communes
    - ✓ lien de base de données (*database link*) : permet l'accès à une base (Oracle ou non) distante
    - ✓ index : structure contenant l'adresse physique de chaque ligne d'une table ou d'un cluster
    - ✓ paquetage (*package*) : collection de fonctions, de procédures et autres objets stockés
    - ✓ procédure : programme PL/SQL stocké dans la base de données ne retournant pas de valeur
    - ✓ séquence : permet de générer des entiers uniques
    - ✓ cliché (*snapshot*) : table contenant le résultat d'une requête définie sur une base distante
    - ✓ journal de cliché (*log snapshot*) : table associée à la table maître utilisée par le cliché
    - ✓ table : structure de données contenant les données, composée de lignes (occurrences) et de colonnes (champs)
    - ✓ vue : représentation logique combinant une ou plusieurs tables ou vues
    - ✓ déclencheur (*trigger*) : procédure stockée dans la base associée à un événement
    - ✓ fonction : programme PL/SQL stocké dans la base de données retournant une valeur
    - ✓ synonymes : re-dénomination de certains des objets de la base
    - ✓ segment d'annulation : sauvegarde des données permettant de valider ou de défaire les transactions
    - ✓ espace de tables (*tablespace*) : allocation d'espace disque dans la base de données pour stocker les objets
    - ✓ clé primaire : une ou plusieurs colonnes permettant d'identifier de manière unique chaque ligne de la table
    - ✓ clé unique : une ou plusieurs colonnes permettant d'identifier de manière unique chaque ligne de la table, autorisant la valeur indéterminée
    - ✓ clé étrangère : une ou plusieurs colonnes dont les valeurs dépendent d'une clé primaire

    - ✓ intégrité référentielle : consistance des relations des clés étrangères référençant les clés primaires
  - utilisateurs avec leurs privilèges sur les différents objets
    - ✓ profil : ensemble de limitations de ressources, attribué explicitement ou implicitement à chaque utilisateur
    - ✓ rôle : ensemble de privilèges attribuables à des utilisateurs ou autres rôles
    - ✓ privilège d'accès : droit accordé à un utilisateur sur la base de données
    - ✓ utilisateur : compte administré par Oracle ayant accès à tout ou partie de la base de données
  - informations relatives à l'activité de la base (connexions, ressources utilisées, verrouillages, etc.)

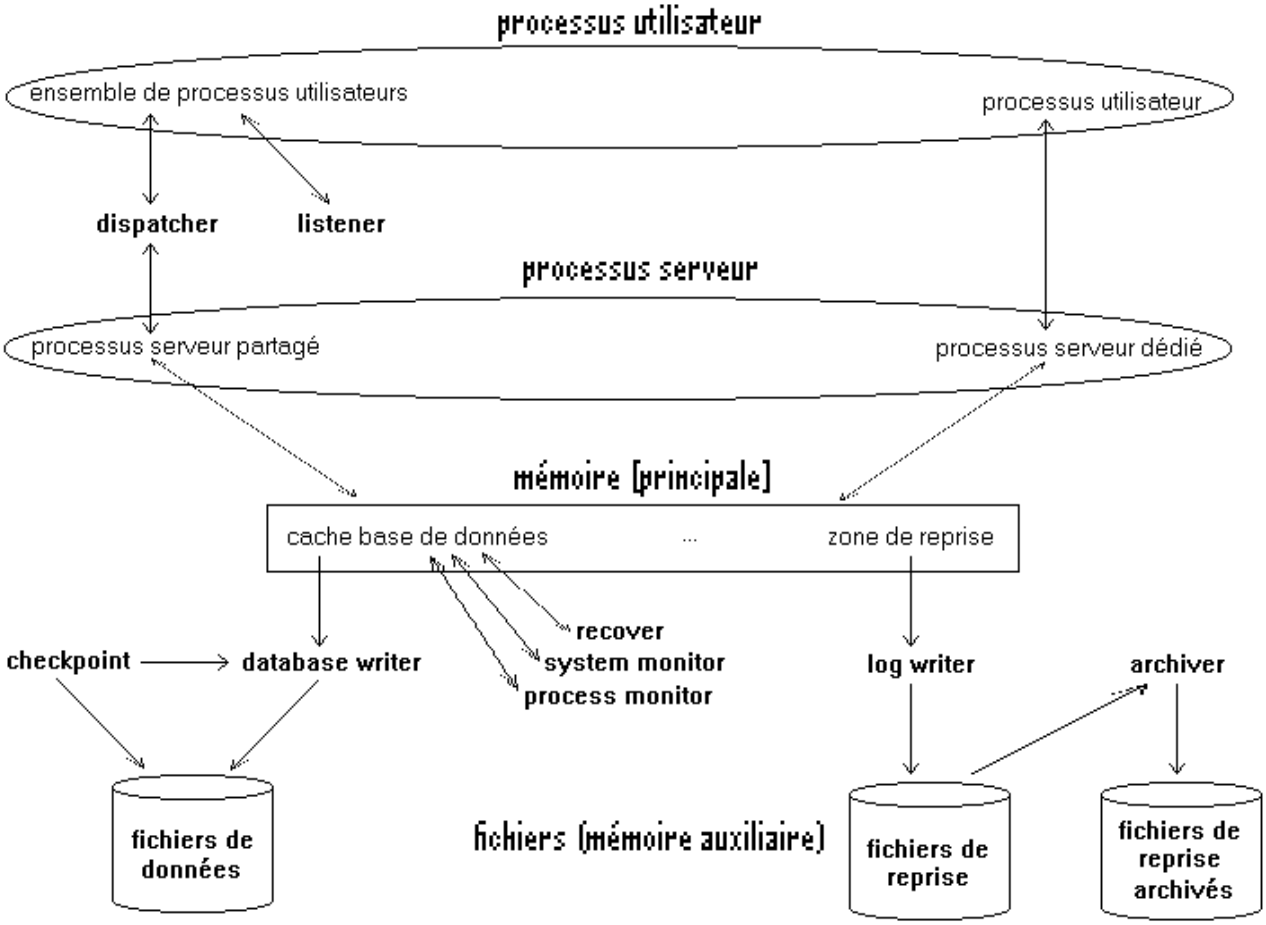
  Le dictionnaire des données est organisé comme une base de données relationnelle de sorte que ses tables et vues sont accessibles en SQL, en lecture pour certains utilisateurs, en lecture et mise à jour par le noyau via l'utilisateur `SYS` ; ces vues sont réparties en quatre classes (la liste complète des tables, vues et synonymes s'obtient en consultant la vue `DICTIONARY`) :
    - ✓ vues relatives aux objets d'un utilisateur (`USER_…`) : trace de connexion/déconnexion de l'utilisateur (`USER_AUDIT_SESSION`), tables, vues, synonymes et séquences de l'utilisateur (`USER_CATALOG`), objets de l'utilisateur (`USER_OBJECTS`), tables créées par l'utilisateur (`USER_TABLES`), colonnes des tables, vues, clusters crées par l'utilisateur (`USER_TAB_COLUMNS`), espaces de tables accessibles par l'utilisateur (`USER_TABLESPACES`), informations sur l'utilisateur (`USER_USERS`), privilèges systèmes attribués à l'utilisateur (`USER_SYS_GRANTS`), autorisations sur les tables et vues appartenant à l'utilisateur ou données ou reçues (`USER_TAB_GRANTS`), etc.
    - ✓ vues relatives aux objets accessibles à un utilisateur (`ALL_…`) : les mêmes !
    - ✓ vues relatives aux administrateurs (`DBA_…`) : les mêmes, description des fichiers de données (`DBA_DATA_FILES`), extensions libres dans les espaces de tables (`DBA_FREE_SPACE`), etc.
    - ✓ vues relatives au suivi des performances disponibles aux administrateurs (`V$…` ou `V_$…`) : objets actuellement verrouillés (`V$ACCESS`), description de la base de données ou des fichiers de données ou des fichiers de reprise à partir du fichier de contrôle (`V$DATABASE` ou `V$DATAFILE` ou `V$LOGFILE`), valeurs de paramètres d'initialisation (`V$PARAMETER`), processus actifs (`V$PROCESS`), mémoire (`V$SGA`), transactions en cours (`V$TRANSACTION`), session courante (`V$SESSION`), etc.
- SQL : interface entre le noyau et les différents outils Oracle ; chaque commande (interprétée) subit une vérification syntaxique et sémantique, est décomposée en opérations élémentaires et soumise au noyau pour exécution, est récupérée et transmise à l'application ou l'outil en ayant fait la demande
- PL/SQL : extension procédurale du langage SQL
- outils de développement d'applications : SQL*Plus, SQL*Forms, SQL*ReportWriter, SQL*Menu, Pro* (Pro*C, Pro*COBOL, SQL*Webserver, etc.)
- outils de communication SQL*Star : SQL*Net, SQL*Connect
- outils de génie logiciel : CASE*Dictionary, CASE*Designer, CASE*Generator
- outils d'administration
  - SQL*DBA : toute administration d'Oracle, en mode menu ou en mode ligne
  - Import/Export : échange de données entre différentes bases Oracle ; c'est également un outil de défragmentation
  - SQL*Loader : importer un fichier dans une base de données Oracle
- outils d'aide à la décision : ORA 1-2-3, SQL*QMX, Easy*SQL, SQL*Calc, Oracle*Mail

2. Architecture interne

L'architecture interne d'Oracle est composée de fichiers, de processus (programmes en cours d'exécution) et de mémoire. Une instance Oracle (désignée par le SID, *System Identifier*) correspond à un ensemble de fichiers, une zone mémoire allouée et des exécutables assurant le fonctionnement d'une base de données. Chaque instance est complètement indépendante et il est possible de faire fonctionner simultanément plusieurs instances.

2.1. Principe général

2.2. Les fichiers



Oracle respecte la norme ANSI/X3/SPARC représentant une base de données en trois niveaux (externe, conceptuel, interne respectivement les vues et la gestion des utilisateurs et de leurs droits d'accès, la structure logique, la structure physique).

2.2.1. La structure physique

La structure physique est composée d'un ensemble de fichiers constituant le support physique de stockage des données et concerne :

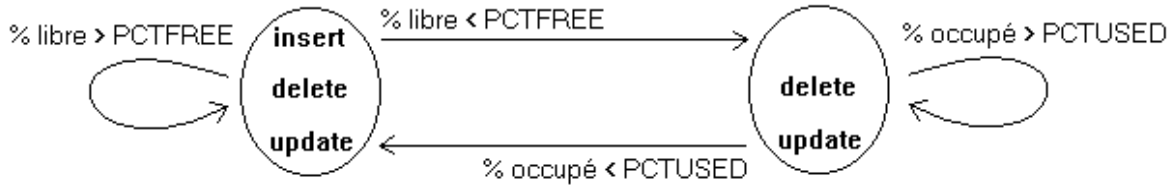
- le fichier d'initialisation `init`*`SID`*`.ora` fixant les paramètres d'initialisation
- les fichiers du réseau : `listener.ora` sur le serveur et `Tnsnames.ora` sur chaque client
- les fichiers de données (*data files*), qui peuvent être connus en interrogeant la vue `V$DATAFILE` du dictionnaire des données, contenant le dictionnaire des données et les objets créés par les utilisateurs (i.e. la base de données : tables, contraintes d'intégrité, index, déclencheurs et procédures stockées, privilèges, etc.)
- les fichiers de reprise (*log files*), qui peuvent être connus en interrogeant la vue `V$LOGFILE` du dictionnaire des données, contenant les mises à jour les plus récentes de la base de données et permettant (suite à une panne matérielle ou logicielle) de remettre la base de données dans un état cohérent
  - gérés circulairement
  - possibilité d'archivage avant réutilisation (fichiers de reprise archivés)
  - (*mirroring*) possibilité de les dupliquer (fichiers de reprise multiples)
- les fichiers de contrôle (*control files*), qui peuvent être connus en interrogeant la vue `V$CONTROLFILE` du dictionnaire des données, contenant des informations sur la structure physique d'une base de données (nom de la base de données, estampille de la création de la base, noms et localisations des fichiers de données et de reprise, numéro de séquence courant du journal)
  - (*mirroring*) possibilité de les dupliquer

2.2.2. La structure logique

La structure logique est définie par la hiérarchie suivante :

- base de données : ensemble d'objets de schéma répartis dans les espaces de tables
  - inaccessible si la base n'est pas chargée, même si l'instance (ensemble de la mémoire allouée et des processus, utilisateurs et SGBD, actifs) est démarrée
  - accessible uniquement aux administrateurs si la base est chargée (et associée à une instance) mais non ouverte
  - accessible à tous les utilisateurs (dont les administrateurs) si la base est ouverte
- objets de schéma (*schéma objects*) : objets logiques (tables, vues, clusters, index, séquences, procédures stockées, fonctions, paquetages, déclencheurs)
  - chaque objet logique appartient à un seul espace de tables

- espace de tables : regroupement d'objets logiques sous la forme d'un ensemble de fichiers
  - au moins `SYSTEM` pour le dictionnaire des données,
    il est conseillé d'en avoir au moins un second pour stocker les objets de la base de données séparément du dictionnaire des données
  - désactivable (excepté `SYSTEM`) pour rendre un espace de tables inaccessible aux utilisateurs
- fichier : ensemble de segments
- segment : ensemble d'extensions
  - données : stockage des tables utilisateurs, systèmes et groupées (clusters)
  - index : stockage des données d'index
  - annulation (*rollback*) : enregistrement des actions des transactions (identification du bloc modifié et des valeurs avant mise à jour pour chaque *rollback entry*)
  - temporaire : associé aux requêtes nécessitant un espace disque supplémentaire (création d'index, sélection avec tri, opérateur ensembliste)
  - amorçage (*bootstrap*) : définition du dictionnaire des données chargée lors de l'ouverture de la base de données (situé dans l'espace de tables `SYSTEM`)
- extension (*extent*) : suite de blocs (logiques) contigus
  - extension initiale et supplémentaires (suivante et pourcentage d'accroissement à partir de la troisième extension)
- bloc (logique) : la plus petite unité logique d'entrée/sortie gérée par Oracle
  - la taille d'un bloc logique est un multiple de la taille d'un bloc physique géré par le système d'exploitation



`PCTFREE` (10% par défaut) : pourcentage d'espace du bloc devant rester libre (pour les modifications)
Si la valeur de `PCTFREE` est faible, on gagne en espace (peu de blocs) mais on perd en performance (données fragmentées).
La valeur de `PCTFREE` doit être supérieure à 10% si et seulement si la mise à jour d'une ligne entraîne souvent une augmentation de sa taille.

`PCTUSED` (40% par défaut) : pourcentage d'espace du bloc autorisant ou non l'insertion dans ce bloc

Format d'un bloc : entête (adresse du bloc, type de segment auquel il appartient), liste des tables (informations sur les tables clustérisées), liste des lignes (adresse de chaque ligne, etc.), espace libre (pour la modification des lignes), données (sur une ou plusieurs lignes, ou sur une partie de ligne imposant un chaînage des blocs).

Format d'une ligne : entête de ligne (overhead, nombre de colonnes, identifiant de clé de cluster, adresse physique en cas de chaînage) et colonnes (longueur et valeur de chacune).

2.3. La mémoire

La mémoire est ainsi structurée :

- zones réservées au code de l'applicatif : code des programmes en cours d'exécution
  - noyau Oracle
  - outils Oracle (SQL*DBA, SQL*Plus, SQL*Forms, etc.) et programmes d'application faisant appel à Oracle
- zone globale système (*System Global Area* ou *Shared Global Area*) : zones tampons partagées contenant les données et les informations de contrôle relatives à une instance
  - cache base de données (ou buffer cache) : ensemble de zones tampons partagées (par tous les processus) contenant des copies des blocs de données (des tables, index, segments d'annulation, clusters) lus à partir des fichiers de données ; géré par une liste des tampons modifiés non encore écrits sur le disque et une liste des tampons récemment les moins utilisés (algorithme LRU i.e. *Least Recently Used*)
  - zone de reprise (*Log buffer*) concernant les mises à jour des données
  - pool partagé (*Shared buffer pool*), géré par un algorithme LRU
    - ✓ cache de la bibliothèque (zone SQL partagée)
    - ✓ cache du dictionnaire des données
    - ✓ données des sessions (pour la version multi-threads d'Oracle)
  - zone privée SQL et PL/SQL
- zone globale programme (*Program Global Area*) : informations de contrôle relatives à un processus

2.4. Les processus

Les types de processus sont les suivants :

- processus utilisateur (client) : tâches soumises par SQL*DBA, SQL*Plus, SQL*Forms, Pro*C, etc.
- processus SGBD (*RDBMS process*) : gestion des données

- processus serveur : processus associé à un ensemble de processus utilisateurs
  - ✓ analyse et exécution des requêtes SQL soumises par les applications
  - ✓ transfert des blocs de données du disque en mémoire
  - ✓ communication des résultats aux applications
- processus d'arrière plan (*background process*) : tâches du SGBD exécutées de façon asynchrone
  - ✓ `smon` (*system monitor*) : recouvrement d'instance lors de son démarrage
  - ✓ `dbwr` (*database writer*) : gestion du cache base de données et écriture des mises à jour apportées à ce cache dans les fichiers de données
  - ✓ `lgwr` (*log writer*) : gestion de la zone de reprise et écriture des tampons de reprise dans le fichier de reprise
  - ✓ `pmon` (*process monitor*) : recouvrement des processus utilisateurs
  - ✓ `reco` (*recover*) : recouvrement en cas d'échec d'une transaction répartie
  - ✓ `arch` (*archiver*) : copie des fichiers de reprise pleins
  - ✓ `ckpt` (*checkpoint*) : écriture de toutes les données (du cache base de données) modifiées
  - ✓ `oralsn` (*listener*) : aiguille les connexions utilisateurs vers un processus dispatcher (cas d'une configuration multi-thread server)
  - ✓ `Dnnn` (*dispatcher*) : partage les processus serveurs pour les processus utilisateurs
  - ✓ `lck0`, …, `lck9` (*lock*) : gèrent le verrouillage inter-instances (cas d'un serveur parallèle)

Gestion multi-processus :

- architecture combinée : un processus utilisateur pour chaque utilisateur (remplissant aussi les fonctions des processus serveurs)
- architecture à serveur dédié : un processus utilisateur et un processus serveur pour chaque utilisateur
- architecture à serveur partagé : un processus serveur peut être associé à plusieurs processus utilisateurs

3. Création et démarrage de la base de données

3.1. Création de la base de données

- Préambule
  - Être administrateur de la base de données ayant le plus haut niveau de privilèges
  - Évaluer l'espace disque nécessaire (et en disposer)
  - Prévoir les moyens de gestion de la sécurité (fichiers de reprise, archivage, sauvegarde et restauration, etc.)
- Sauvegarde (commande du système d'exploitation) des bases existantes (fichiers d'initialisation, du réseau, de données, de reprise, de contrôle)
- Création et modification du(des) fichier(s) d'initialisation, et notamment les paramètres suivants :
  - `DB_DOMAIN` : nom logique du domaine dans un réseau
  - `DB_NAME` : nom local de la base de données
  - `CONTROL_FILES` : nom et localisation des fichiers de contrôle
  - `INIT_SQL_FILES` : fichier SQL construisant le dictionnaire de données et d'autres fichiers initiaux
    - ✓ `SQL.BSQ` : le premier script obligatoire permettant la création de l'espace de tables `SYSTEM` (avec le segment d'annulation `SYSTEM`) et des tables du dictionnaire des données (et le chargement de certaines d'entre elles)
    - ✓ `CATALOG.ORA` : le deuxième script obligatoire permettant la création des vues du dictionnaire des données (avec les synonymes publiques de ces vues et l'accès `PUBLIC` à ces synonymes)
    - ✓ éventuellement `AUDIT.SQL` pour l'obtention de vues sur les traces d'activité du système
    - ✓ etc.
- Arrêt et redémarrage d'Oracle
  - Arrêt de la base de données ouverte : `shutdown` (de SQL*DBA)
  - Se connecter avec le privilège `INTERNAL`
  - Démarrer une nouvelle instance (qui utilise les fichiers d'initialisation, `INIT_ORA` par défaut) sans ouvrir la base de données :
    `startup nomount pfile=◁fichier d'initialisation▷`
- Création de la nouvelle base de données
  La commande `create database ◁nom de base de données▷ controlfile reuse logfile ◁fichiers de reprise▷ datafile ◁fichiers de données▷` génère la création des fichiers de données, des fichiers de reprise, des fichiers de contrôle, de l'espace de tables `SYSTEM` et du segment d'annulation `SYSTEM`, du dictionnaire des données (tables, vues, synonymes), de deux utilisateurs (`SYS` et `SYSTEM`) ; la base de données est alors non seulement créée mais l'instance est démarrée et la base de données est chargée (montée) et ouverte.

3.2. Modification des caractéristiques de la base de données

La commande `alter database` permet l'ajout et la suppression de fichiers de reprise, la redénomination de fichiers de données ou de reprise dans les fichiers de contrôle (opération à faire aussi au niveau du système d'exploitation), l'ajout de fichiers de données (notamment en cas de problème avec un ancien fichier de données), etc.

3.3. Démarrage et fermeture de la base de données

3.3.1. Démarrage d'une base de données Oracle

Le démarrage d'une base de données Oracle consiste à mettre la base de données à la disposition des utilisateurs.

- Se connecter avec le privilège `INTERNAL`
  `connect INTERNAL`
  Démarrer une instance
  Le démarrage d'une instance consiste à préparer le contexte (espace mémoire et processus) pour les utilisateurs.
  `startup nomount`
- Chargement d'une base de données
  Le chargement d'une base de données consiste à associer une base de données à une instance démarrée.
  `alter database mount` (ou `startup mount` pour faire en même temps le démarrage de l'instance)
  La base de données est alors accessible aux administrateurs mais pas aux autres utilisateurs ; cela permet de renommer un fichier de données, ajouter ou supprimer ou renommer un fichier de reprise, activer ou désactiver l'option d'archivage des fichiers de reprise, effectuer un recouvrement (restauration suite à une panne) total de la base de données, etc.
- Ouverture d'une base de données
  L'ouverture d'une base de données consiste à rendre la base de données accessible à tous les utilisateurs.
  `alter database open` (ou `startup open` pour faire en même temps le démarrage de l'instance et le chargement de la base de données)
  Les fichiers de données et de reprise sont alors ouverts.

3.3.2. Fermeture d'une base de données Oracle

- Se connecter avec le privilège `INTERNAL`
  `connect INTERNAL`
- Fermeture d'une base de données
  Les données en mémoire sont enregistrées dans les fichiers de données et de reprise qui sont ensuite fermés.
- Déchargement d'une base de données
  Le déchargement d'une base de données consiste à dissocier une base de données d'une instance.
  Les fichiers de contrôle sont alors fermés.
- Arrêter une instance
  L'arrêt d'une instance consiste à libérer la mémoire et arrêter tous les processus.
  La commande `shutdown normal` refuse toute nouvelle connexion et attend que tous les utilisateurs se déconnectent ; quand il n'y a plus aucun utilisateur connecté, la base de données est fermée et déchargée et l'instance est arrêtée.

4. Transactions et accès concurrents

- Une transaction débute implicitement par le début d'une session et se termine soit à la fin d'une session soit par l'ordre SQL `commit` (qui valide toutes les opérations de mise à jour des données de la transaction) ou `rollback` (qui annule ces opérations).
- Il est possible de placer de points de repères intermédiaires par l'ordre `savepoint`.
- Les accès concurrents font intervenir les concepts d'intégrité, de concurrence, de consistance et mettent en œuvre la technique de verrouillage des données.
- Le granule de verrouillage est logique (table, ligne, etc.) ou physique (segment, fichier, page, etc.).
- Le verrouillage est implicite (géré par Oracle pour maintenir la cohérence des données) ou explicite (l'utilisateur contrôle lui-même les verrouillages, au niveau des transactions ou d'une instance).
  - Au démarrage d'une instance, par défaut (verrouillage implicite), le paramètre `SERIALIZABLE` vaut `FALSE` (`TRUE` signifierait un verrouillage systématique au niveau table) tandis que le paramètre `ROW_LOCKING` vaut `ALWAYS` (`INTENT` signifierait la non mise à jour simultanée des lignes d'une table par plusieurs transactions).
  - Les différents modes de verrouillage explicite des données sont les suivants :
    - ✓ aucun verrouillage (`V$LOCK.LMODE=1`)
      tous les autres verrous (RS RX S SRX X, décrits ci-après) sont autorisés
      `select`
    - ✓ lignes partagées RS (*row share*) (`V$LOCK.LMODE=2`)
      verrou sélectif des lignes en vue de leur mise à jour
      autres verrous autorisés : RS RX S SRX
      `select ... for update`
      `lock table ... in row share mode`
    - ✓ lignes exclusives RX (*row exclusive*) (`V$LOCK.LMODE=3`)
      verrou sélectif de lignes d'une table en vue de leur mise à jour ; les lignes ne sont cependant pas partagées avec d'autres transactions
      autres verrous autorisés : RS RX
      `insert`

update
delete
lock table ... in row exclusive mode

- ✓ table partagée S (*share*) (V$LOCK.LMODE=4)
  verrou sur une table ; les autres transactions peuvent effectuer des interrogations sur cette table ; si plusieurs transactions verrouillent la même table en mode table partagée, aucune ne peut effectuer de mise à jour
  autres verrous autorisés : RS S
  lock table ... in share mode
- ✓ partage exclusif de lignes SRX (*share row exclusive*) (V$LOCK.LMODE=5)
  verrou sur une table ; les autres transactions peuvent effectuer des interrogations sur cette table
  autres verrous autorisés : RS
  lock table ... in share exclusive mode
- ✓ table exclusive X (*exclusive*) (V$LOCK.LMODE=6)
  verrou sur une table ; les autres transactions peuvent effectuer des interrogations sur cette table ; la transaction ayant positionné ce verrou peut effectuer une écriture exclusive sur la table
  aucun autre verrou autorisé
  lock table ... in exclusive mode

- La fin d'une transaction lève les verrous qu'elle a posés.
- Les verrouillages du dictionnaire des données et internes (mémoire, etc.) sont transparents (gérés par Oracle).

5. Sécurité des données

La sécurité des données consiste à contrôler l'authenticité des utilisateurs lors de leur connexion, à définir les espaces de tables accessibles par défaut, à définir les limitations des ressources, à accorder les privilèges d'accès aux objets de la base de données.
Les commandes permettant de gérer les utilisateurs avec leurs privilèges sur les différents objets :

- profil : {create|alter|drop} profile ◁profil▷
- rôle : {create|alter|drop} role ◁rôle▷ pour sa création, set role ◁rôles▷ pour son activation
- privilège d'accès : privilège système (accès à la base de données et à la définition de ses objets) ou privilège objet (manipulation d'objets de la base de données)
  grant {◁privilèges système▷|◁rôles▷} to {◁utilisateurs▷|◁rôles▷} pour attribuer un privilège système
  revoke {◁privilèges système▷|◁rôles▷} from {◁utilisateurs▷|◁rôles▷} pour retirer un privilège système
  grant ◁privilèges objet▷ on ◁objet▷ to {◁utilisateurs▷|◁rôles▷} pour attribuer un privilège objet
  revoke ◁privilèges objet▷ on ◁objet▷ from {◁utilisateurs▷|◁rôles▷} pour retirer un privilège objet
- utilisateur : {create|alter|drop} user ◁utilisateur▷ (profil DEFAULT par défaut)

Deux comptes sont créés automatiquement lors de l'installation du logiciel : SYS (avec le mot de passe change_on_install) et SYSTEM (avec le mot de passe manager). Ces comptes ont des privilèges très élevés (même supérieurs à ceux des administrateurs de la base de données puisque par exemple SYS est le seul compte à avoir accès au dictionnaire de données) et ne doivent donc jamais être utilisés (exceptions faites de l'installation, de la désinstallation, et de la création du premier compte d'administration de la base de données).
Si vous installez les exemples fournis par Oracle, le compte SCOTT (avec le mot de passe tiger) est également créé.

6. Sauvegarde et restauration de données

Il s'agit de la planification et de la mise en place de procédures de sauvegarde et de restauration (suite à une panne quelconque) de la base de données.
La restauration peut se faire sans l'intervention de l'administrateur (reprise à chaud) ou avec son concours à partir d'une sauvegarde (reprise à froid).

6.1. Les différentes pannes

- erreur accidentelle : suppression d'un objet de la base (la solution consiste à renforcer la sécurité)
- panne sur une commande SQL : par manque de ressources, Oracle annule la transaction et retourne un code d'erreur
- panne d'un processus : qui s'arrête sans validation ou annulation de la transaction en cours.
  pmon détecte cette panne et se substitue au processus défaillant
- panne réseau : qui interrompt l'exécution d'une application.
  pmon détecte cette panne et se substitue au processus défaillant, ou reco si la transaction est répartie
- panne d'instance : suite à une panne matérielle ou logicielle, la base s'arrête dans un état incohérent puisque des mises à jour validées ne sont pas répercutées dans les fichiers de données (mais présentes dans les fichiers de reprise) ou encore des mises à jour non validées ont été transmises aux fichiers de données (mais leurs images avant sont présentes dans les

segments d'annulation).
Oracle effectue automatiquement le recouvrement : `smon` effectue tout d'abord un roll-forward des mises à jour présentes dans les fichiers de reprise et ensuite un rollback des mises à jour non validées dont les images avant sont présentes dans les segments d'annulation

- panne disque : qui, suite à une cause matérielle, empêche la lecture ou l'écriture d'un fichier.
  - pour un fichier de reprise : si un fichier miroir existe, Oracle continue sans interruption ; sinon Oracle s'arrête avec une perte de données
  - pour un fichier de contrôle : Oracle s'arrête (même si un fichier miroir existe)
  - pour un fichier des données : en lecture, Oracle signale l'erreur et continue ; en écriture, si le fichier de reprise en ligne plein est archivé, alors une erreur est retournée dans le fichier de trace et l'espace de tables est mis hors service, sinon `dbwr` s'arrête et l'instance se bloque

  Si l'option `NOARCHIVELOG` est activée, le recouvrement doit se faire en restaurant la dernière sauvegarde complète (toutes les données mises à jour depuis la date de cette dernière sauvegarde sont définitivement perdues).
  Si l'option `ARCHIVELOG` est activée, le recouvrement se fait en restaurant la base jusqu'à la dernière transaction consistante validée.

6.2. Les mécanismes mis en œuvre

6.2.1. Le journal

- Les fichiers de reprise (ou journal) possèdent la trace des enregistrements mis à jour dans la base de données, l'écriture dans les fichiers de reprise étant assurée par le processus `lgwr` à partir du tampon (*buffer*) de reprise de la mémoire (SGA).
  - Le fichier de reprise en ligne est le journal courant de l'instance.
  - Il y a toujours au moins deux fichiers de reprise de sorte que l'un d'entre eux est toujours disponible s'il y en a un qui est plein et en attente d'archivage.
  - Les fichiers de reprise sont gérés circulairement :
    - ✓ quand le fichier de reprise en ligne est plein, il est alors désactivé et le fichier de reprise suivant est alors activé : cet événement est appelé l'interrupteur journal
    - ✓ chaque fichier de reprise a un numéro de séquence journal qui s'incrémente chaque fois que l'interrupteur journal se produit ; de plus, le fichier de contrôle contient le numéro de séquence journal le plus récent
  - Un point de vérification (*checkpoint*) se produit lorsque `dbwr` enregistre tous les buffers modifiés de la SGA dans les fichiers de données ; `ckpt` fait toutes les écritures reflétant ce point de vérification.
  - Les fichiers de reprise archivés permettent le recouvrement de toutes les transactions validées sans perte d'information.
  - Archivage des fichiers de reprise pleins une fois inactifs :
    - ✓ automatiquement par le processus `arch` si le paramètre `LOG_ARCHIVE_START` est `TRUE` ou après avoir tapé la commande `alter system archive log start` (le format des noms des fichiers à archiver et leur destination sont disponibles respectivement dans les paramètres `LOG_ARCHIVE_FORMAT` et `LOG_ARCHIVE_DEST`)
    - ✓ par l'administrateur … ce qui est fortement déconseillé sauf si l'archivage automatique est suspendu suite à un problème quelconque
- Les segments d'annulation stockent les anciennes valeurs de données mises à jour pour permettre une annulation de transaction.
- Les fichiers de contrôle possèdent les statuts des structures physiques de la base de données ; ils sont modifiés à chaque mise à jour de la structure de la base.

6.2.2. La restauration

- Deux situations peuvent se produire :
  - des données sont mises à jour par des transactions mais ne sont pas écrites dans les fichiers de données au moment de la validation (de sorte que ces données ne sont présentes que dans les fichiers de reprise)
  - les fichiers de reprise contiennent des données non validées dues à des transactions annulées (les segments d'annulation disposent de cette information)
- Les étapes du recouvrement sont alors les suivantes :
  - récupérer la dernière sauvegarde ;
    la base est dans un état cohérent, mais a perdu les mises à jour les plus récentes.
  - (*roll-forward*) appliquer sur les fichiers de données l'image avant du journal i.e. effectuer sur les fichiers de données toutes les mises à jour présentes dans les fichiers de reprise (en ligne et archivés) et dans les segments d'annulation ; la base contient alors toutes les mises à jour, validées ou non.
  - (*rollback*) appliquer sur les fichiers de données l'image arrière à partir des segments d'annulation (qui annulent les transactions non validées) ;
    la base est dans un état cohérent et contient alors toutes les mises à jour les plus récentes ayant été validées.

6.3. Les sauvegardes

6.3.1. Les stratégies de sauvegardes

- Peut-on accepter de perdre des données en cas de panne ?
  - Si la réponse est non, les fichiers de reprises doivent être archivés (option `ARCHIVELOG` activée) et dupliqués (*mirroring*).
  - Si la réponse est oui, sauvegarde régulière (quotidienne, hebdomadaire, mensuelle, etc. selon la période de tolérance de perte des données respectivement d'un jour, d'une semaine, d'un mois, etc.) et les fichiers de reprises n'ont pas à être archivés (option `NOARCHIVELOG` activée).
- Quelle est la disponibilité nécessaire pour la base de données ?
  - Si la réponse est toujours, les fichiers de reprises doivent être archivés (option `ARCHIVELOG` activée).
  - Si la réponse est « peu disponible », les fichiers de reprises n'ont pas à être archivés (option `NOARCHIVELOG` activée).

6.3.2. Les différentes sauvegardes

6.3.2.1. Sauvegarde complète

- Sauvegarde sur un support magnétique de tous les fichiers de la base de données (de données, de reprise, de contrôle) --- ainsi que les fichiers d'initialisation et du réseau --- par une commande du système d'exploitation.
- Fait suite à une fermeture normale de la base de données.
- En mode `ARCHIVELOG`, recouvrement à partir de la sauvegarde la plus récente et utilisation des fichiers de reprise (en ligne et archivés) permettant de reprendre jusqu'au moment où s'est produite la panne.
- Procédure :
  `shutdown normal`
  sauvegarde de tous les fichiers (de données, de reprise, de contrôle) par une commande du système d'exploitation
  `startup`

6.3.2.2. Sauvegarde partielle

Utile qu'en mode `ARCHIVELOG`.

- Sauvegarde d'un espace de tables ou d'un fichier de données ou de contrôle.
- La base peut être ouverte ou fermée.
- Procédure dans le cas d'un fichier de données ou d'un espace de tables activé (*online*) :
  - identifier le fichier de données d'un espace de tables activé par la commande
    `select FILE_NAME from DBA_DATA_FILES where TABLESPACE_NAME='◁tablespace▷'`
  - marquer le début des opérations de sauvegarde de l'espace de tables activé : `alter tablespace`
  - sauvegarde des fichiers de données en ligne par une commande du système d'exploitation
  - marquer la fin des opérations de sauvegarde de l'espace de tables désactivé : `alter tablespace`

  En mode `ARCHIVELOG`, Oracle insère le dernier point de vérification (*checkpoint*) dans le fichier de données permettant, en cas de restauration, d'appliquer le recouvrement à partir de ce point.
- Procédure dans le cas d'un fichier de données ou d'un espace de tables désactivé (*offline*) :
  - identifier le fichier de données d'un espace de tables désactivé par la commande
    `select FILE_NAME from DBA_DATA_FILES where TABLESPACE_NAME='◁tablespace▷'`
  - désactiver l'espace de tables : `alter tablespace`
  - sauvegarde des fichiers de données en ligne par une commande du système d'exploitation
  - réactiver l'espace de tables : `alter tablespace`

  La sauvegarde d'un fichier de données inactivé ne pose pas de problème car il est dans un état cohérent.

6.3.3. Quelques conseils !

- Sauvegarder la base (entièrement) lors de sa création.
- Effectuer des sauvegardes partielles mettant à jour la sauvegarde de la base.
- Effectuer des sauvegardes des fichiers de données (notamment des espaces de tables les plus utilisés).
- Effectuer des sauvegardes des fichiers de contrôle à chaque mise à jour de la structure de la base de données.
- Rappelons que la sauvegarde du fichier de reprise est implicite en mode `ARCHIVELOG`.

7. L'optimisation

Il s'agit de fixer les objectifs à atteindre pour une meilleure utilisation des performances en adaptant Oracle aux besoins spécifiques (voire contradictoires) des applications. Par exemple, doit-on privilégier le temps de réponse ou de débit du résultat, ou encore doit-on considérer une requête ou une application ou tous les traitements concurrents, etc.

Il convient d'optimiser tout d'abord les requêtes, ensuite la mémoire et enfin les entrées/sorties et les contentions des processus.
Dans tous les cas, les performances de votre système peuvent être déplorables si la valeur du paramètre `DB_BLOCK_SIZE` (correspondant à l'unité de taille d'échange entre la mémoire principale et la mémoire auxiliaire) a mal été définie (et fixée une fois pour toutes) lors de la création de la base de données.

7.1. Optimisation d'une requête

Oracle dispose d'un optimiseur intégré ayant pour objectif de trouver le moyen le plus efficace pour exécuter une commande SQL.

7.1.1. Les phases d'exécution d'une requête

Les différentes phases d'exécution d'une requête sont l'analyse, l'exécution et la recherche. Toute requête suit le traitement suivant :

si l'analyse n'a pas déjà été effectuée
création d'un curseur
analyse (*parse*) de la requête
vérifications syntaxique, vérifications d'existence et des droits sur les objets
verrouillages
planification de l'exécution, chargement en mémoire, répartition éventuelle
si la requête est une requête d'interrogation
description des champs du résultat (notamment pour une requête interactive ou du SQL dynamique)
définition des variables en sortie devant accueillir le résultat
substitution des variables (*bind*) par les valeurs courantes pour les paramètres en entrée
exécution (*exec*) de la commande, avec verrouillage (suppression ou modification) ou sans verrouillage (insertion ou interrogation)
si la requête est une requête d'interrogation
recherche (*fetch*), une à une, des lignes du résultat, avec tri éventuel

7.1.2. Le plan d'exécution

Le plan d'exécution d'une requête consiste pour Oracle à trouver le meilleur chemin pour accéder aux données, en minimisant le nombre d'opérations d'entrée/sortie et en minimisant le temps de traitement, et en optant pour une méthode statique ou statistique.
La commande `explain` permet d'expliquer le plan d'exécution d'une requête.

7.1.3. L'accès aux données

L'accès aux données (sur disque) peut s'effectuer :

- directement dans la table relativement à l'adresse physique (identification du fichier, bloc dans le fichier et ligne dans le bloc) de la ligne (`ROWID`)
- par un parcours d'un cluster indexé (*cluster scan*) où les lignes ayant même valeur pour la clé du cluster sont stockées dans le même bloc
- par un parcours d'un cluster par hachage (*hash scan*) où les lignes ayant même valeur de hachage sont stockées dans le même bloc
- par un parcours d'un index (*index scan*) pour retrouver les lignes grâce à une ou plusieurs colonnes de l'index ; l'index peut être unique (le parcours de la table d'index retourne une seule valeur `ROWID`) ou multiple (le parcours de la table d'index retourne une ou plusieurs valeurs `ROWID`)
- par un parcours séquentiel de toutes les données directement recherchées dans la table (*full table scan*)

7.1.4. Les méthodes d'optimisation

7.1.4.1. La méthode statique

La méthode statique consiste à déterminer le plan d'exécution en fonction des différentes méthodes d'accès aux données disponibles en privilégiant celui de moindre coût compte-tenu de la classification suivante (en commençant par la plus efficace) :

- accès à une ligne par son adresse physique
- accès à une ligne par jointure clustérisée
- accès à une ligne par cluster par hachage avec clé unique
- accès à une ligne par clé unique
- jointure clustérisée (i.e. les deux tables référencées par la jointure sont stockées dans le même cluster)
- clé de cluster par hachage
- clé de cluster indexé

- index composé (conjonction sur toutes ses colonnes qui testent l'égalité)
- index unique
- recherche bornée sur des colonnes indexées
- recherche non bornée sur des colonnes indexées
- jointure avec tri et fusion
- fonctions d'agrégation minimum ou maximum de colonnes indexées
- tri sur une colonne indexée
- parcours complet de la table

7.1.4.2. La méthode statistique

La méthode statistique consiste à :
- commencer par générer les plans d'exécution en fonction des différents chemins d'accès aux données disponibles,
- estimer ensuite le coût (nombre d'entrées/sorties, temps CPU, mémoire nécessaire) de chaque plan d'exécution à partir des statistiques existantes sur les tables/colonnes/clusters/index concernés en considérant les critères de sélectivité (pourcentage de lignes ramenées) et de facteur de blocage (nombre de lectures multi-blocs dépendant du paramètre `DB_FILE_MULTIBLOCK_READ_COUNT`),
- choisir finalement l'un des plans d'exécution de coût minimal.

La commande `analyze` permet de générer des statistiques sur un objet. Par exemple, l'ordre `analyze index ◁index▷ compute statistics` lance une analyse complète (plutôt pour les index ou de petites tables) d'un index tandis que l'ordre `analyze table ◁table▷ estimate statistics sample 20 percent` lance une analyse partielle (plutôt pour des tables volumineuses) d'une table.

7.1.5. Les buts d'optimisation

7.1.5.1. Le meilleur temps de réponse

Le but d'optimisation consistant à obtenir le meilleur temps de réponse permet de réduire le temps d'obtention de la première ligne du résultat.
Cela concerne éventuellement un traitement interactif.
Par exemple, une jointure par imbrication de boucles (`NESTED LOOPS`) favorise le temps de réponse.

7.1.5.2. Le meilleur débit du résultat

Le but d'optimisation consistant à obtenir le meilleur débit du résultat permet de minimiser le temps global d'exécution (i.e. pour obtenir la totalité) du résultat.
Cela concerne assurément les traitements par lot.
Par exemple, une jointure par tri et fusion (`MERGE JOIN`) favorise le débit du résultat.

7.1.6. Paramétrage

Le paramètre de session `OPTIMIZER_GOAL` (modifiable par la commande `alter session`) indique la méthode et le but poursuivis :
- statique et temps de réponse (`FIRST_ROWS`) indépendamment de la disponibilité de valeurs statistiques
- statique et débit du résultat (`ALL_ROWS`) indépendamment de la disponibilité de valeurs statistiques
- statistique (`CHOOSE`) : en présence de valeurs statistiques relatives à au moins une table référencée dans la requête (le but d'optimisation est alors le meilleur débit du résultat) ; en l'absence de valeurs statistiques, la méthode utilisée est alors statique

Le paramètre d'initialisation `OPTIMIZER_MODE` indique la méthode utilisée :
- statique (`RULE`) indépendamment de la disponibilité de valeurs statistiques
- statistique (`COST`) en présence de valeurs statistiques relatives à au moins une table référencée dans la requête (le but d'optimisation est alors le meilleur débit du résultat) ; en l'absence de valeurs statistiques, la méthode utilisée est alors statique

Le paramètre `OPTIMIZER_GOAL` est prioritaire sur le paramètre `OPTIMIZER_MODE`.

7.1.7. Forcer les choix de l'optimiseur par programmation

Comme le programmeur a souvent une connaissance plus fine de l'architecture des données que l'optimiseur, il peut forcer le comportement d'Oracle pour exécuter une requête SQL en y insérant un mot-clé (*hint*), entouré par `/*+` et `*/`, pour le choix :
- de la méthode et du but : statique (`RULE`), statique et temps de réponse (`FIRST_ROWS`), statique et débit du résultat (`ALL_ROWS`)
- de la méthode d'accès : parcours séquentiel (`FULL`), directement avec l'adresse physique (`ROWID`), parcours d'un cluster indexé (`CLUSTER`), parcours d'un cluster par hachage (`HASH`), parcours d'un index (`INDEX`)

- de la jointure : dans l'ordre dans lequel les tables se présentent (ORDERED), par des boucles imbriquées (USE_NL), par un algorithme tri et fusion (USE_MERGE)

7.1.8. Modification de la requête par l'optimiseur

Oracle modifie la requête en une syntaxe équivalente lorsque cela est possible :

- en évaluant les expressions
  Par exemple, le prédicat `X=12/4 and X>Y and Z in ('un','deux')` devient `Y<3 and (Z='un' or Z='deux')`.
- en transformant une requête ayant un prédicat contenant un `or` en une requête `union all` si cela améliore l'exécution (i.e. si chaque composante de la condition génère un chemin d'accès basé sur un index ou si l'une des deux conditions impose le parcours complet de la table en l'absence d'index)
- en transformant une requête complexe en une jointure équivalente (pour une sous-requête par exemple)
- en prenant en compte les vues, soit en propageant la définition de la vue dans la requête, soit en propageant la définition de la requête dans la vue

7.2. Optimisation de la mémoire (centrale)

Il s'agit d'obtenir une bonne distribution de la mémoire centrale pour les différentes structures internes d'Oracle de sorte à maximiser les données accessibles en mémoire (i.e. minimiser les défauts mémoire).
Certains paramètres définissent la taille de la mémoire : du cache base de données (DB_BLOCK_BUFFERS), de la zone de reprise (LOG_BUFFER), du pool partagé (SHARED_POOL_SIZE).
La zone privée SQL et PL/SQL contient l'analyse et le plan d'exécution des commandes SQL partagées ; le paramètre `OPEN_CURSORS` définit le nombre maximum de zones par utilisateur.

7.2.1. Appels superflus à l'analyseur

Réduire le nombre d'appels superflus à l'analyseur en comparant le nombre de fois qu'une requête a été analysée relativement à son nombre d'exécutions.
Le diagnostic peut se faire à l'aide de la trace de l'exécution en positionnant le paramètre de session `SQL_TRACE`. Si le nombre d'appels à l'analyseur (champ `COUNT`) en phase d'analyse approche le nombre d'appels à l'analyseur en phase d'exécution, cela signifie que chaque exécution d'une commande fait appel à l'analyseur de l'optimiseur.
La solution dépend de l'outil de développement utilisé. Par exemple, pour un pré-compilateur, les paramètres `HOLD_CURSOR` et `RELEASE_CURSOR` permettent de contrôler les zones privées et les appels à l'analyseur.

7.2.2. Défauts de mémoire du cache de la bibliothèque

Réduire le nombre de défauts de la mémoire du cache de la bibliothèque intervenant lorsqu'il n'y a plus de place mémoire pour une nouvelle requête (en phase d'analyse comme en phase d'exécution) ; il y a alors ré-analyse et ré-allocation de zones partagées.
Le diagnostic peut se faire à l'aide de la vue `V$LIBRARYCACHE` ; ses principaux champs sont le nom de l'élément (NAMESPACE), le nombre de fois qu'un élément a été exécuté (PINS), le nombre de défauts mémoire lors de l'exécution (PINHITS), le nombre de défauts mémoire lors de l'exécution obligeant une ré-analyse ou le rechargement d'un objet système (RELOADS).
L'objectif est d'annuler la somme totale des `RELOADS` ou tout au moins d'avoir le rapport somme totale des `PINS` sur somme totale des `RELOADS` supérieur à 1%.
Les solutions consistent à augmenter la taille de la mémoire correspondante (SHARED_POOL_SIZE), à normaliser l'écriture des requêtes (afin d'obtenir des scripts de requêtes parfaitement comparables, caractère par caractère), à ne libérer la place mémoire occupée par une requête qu'après la fermeture de son curseur (en positionnant le paramètre `CURSOR_SPACE_FOR_TIME` à `TRUE`).

7.2.3. Défauts de mémoire du cache du dictionnaire des données

Réduire le nombre de défauts de la mémoire du cache du dictionnaire des données.
Le diagnostic peut se faire à l'aide de la vue `V$ROWCACHE` ; ses principaux champs sont le nombre total de demandes de données (GETS), le nombre de défauts mémoire (GETMISSES).

7.2.4. Défauts de mémoire du buffer cache

Réduire le nombre de défauts de la mémoire du buffer cache en comparant le nombre total de demandes de données relativement au nombre total de demandes de données ayant provoquées des accès disque.
Le diagnostic peut se faire à l'aide de la vue `V$SYSSTAT` ; ses principaux champs sont le nom (`NAME` : les valeurs `'db blok gets'` et `'consistent gets'` correspondent au nombre total de demandes formulées, la valeur `'physical reads'` correspond au nombre total de demandes ayant généré un accès disque) et sa valeur (VALUE).

La solution consiste à augmenter le nombre de buffers (DB_BLOCK_BUFFERS) si le rapport du nombre d'accès aux données sur le nombre de demandes de données est inférieur à 3/5.
Il faut définir la valeur du nombre de buffers en sachant que la taille mémoire allouée en cache pour les données et les index est égale à DB_BLOCK_BUFFERS * DB_BLOCK_SIZE.

7.2.5. Défauts de mémoire des fichiers de reprise

Augmenter la zone conservant les données à insérer dans les fichiers de reprise (LOG_BUFFER).

7.2.6. Non utilisation de PL/SQL

Réduire la taille du pool partagé (SHARED_POOL-SIZE) conservant les plans d'exécution des requêtes et les blocs PL/SQL, si le PL/SQL est peu utilisé.

7.2.7. Défauts de mémoire liés aux tris

Éviter l'utilisation du segment temporaire du disque survenant lorsque le volume de données à trier est trop grand (dépassant la taille définie par le paramètre SORT_AREA_SIZE).

7.3. Optimisation des entrées/sorties

7.3.1. Contentions des disques

Réduire les contentions des disques i.e. les accès simultanés au disque.
Le diagnostic peut se faire à l'aide de la vue V$FILESTAT ; ses principaux champs sont le nombre de lectures pour chaque fichier (PHYRDS), le nombre d'écritures pour chaque fichier (PHYWRTS).
Les solutions consistent à placer les fichiers de données et de reprise sur des disques différents, placer chaque groupe de fichiers de reprise sur des disques différents, éclater les tables en stockant les données dans des fichiers différents (d'un même espace de tables), séparer tables et index dans des espaces de tables différents.

7.3.2. Allocation d'espace dans les blocs

Améliorer l'allocation d'espace dans les blocs c'est-à-dire réduire les chaînages des lignes sur plusieurs blocs et éviter les migrations d'une ligne d'un bloc vers un autre bloc (lors de la modification de la ligne). En effet, lors de la création d'un enregistrement (table ou index), il est affecté dans un bloc mais suite à des modifications, s'il ne loge plus dans ce bloc, il y a un chaînage de l'enregistrement d'un bloc vers un autre bloc.
Le diagnostic peut se faire à l'aide de la commande analyze avec l'option list chained rows.
La solution préventive consiste à choisir judicieusement les valeurs de la clause de stockage (PCTFREE et PCTUSED) ; par exemple, on peut aller jusqu'à affecter zéro à PCTFREE si on sait que l'information est rarement modifiée. La solution curative consiste à reconstruire la table (ou l'index) morcelée en utilisant une table temporaire servant à stocker (sans chaînage) les lignes chaînées de la table initiale avant de les insérer de nouveau dans celle-ci.

7.3.3. Défragmentation du disque

Éviter d'avoir des tables et des index fragmentés. Cela se produit lorsque l'espace physique initial alloué est plein et requiert une extension (non forcément au voisinage de l'espace initial ou de l'extension précédente).
La solution préventive consiste à choisir judicieusement les valeurs de la clause de stockage (PCTFREE, INITIAL, NEXT et PCTINCREASE). La solution curative consiste à reconstruire la table ou l'index fragmenté en utilisant l'outil Import/Export (option COMPRESSE=YES pour avoir ensemble la table et l'index).

7.3.4. Allocation dynamique d'espace supplémentaire

Réduire les allocations dynamiques d'espace supplémentaire générant des appels récursifs (*recursive calls*) lors de défaut mémoire dans le cache du dictionnaire ou du déclenchement des triggers ou de l'exécution des commandes du langage de définition des données ou de l'exécution des commandes dans les procédures, fonctions, paquetages, blocs PL/SQL ou encore du renforcement des contraintes d'intégrité référentielles.
Le diagnostic peut se faire à l'aide de la vue V$SYSSTAT (valeur 'recursive calls' du champ NAME).
La solution consiste à déterminer et à choisir la taille maximale de l'objet à stocker.

7.4. Réduction des contentions des processus

Une contention des processus se produit lorsque des processus veulent accéder à la même ressource (segments d'annulation, processus des serveurs multi-threads, loquets de buffers de reprise).

7.4.1. Contentions sur les segments d'annulation

Réduire les contentions sur les segments d'annulation.

Le diagnostic peut se faire à l'aide de la vue V$WAITSTAT ; ses principaux champs sont la classe (CLASS avec comme principales valeurs 'system undo header', 'undo header', 'system undo block' et 'undo block'), le nombre (COUNT), le nom (NAME avec comme principales valeurs 'db block get' et 'consistent gets'), la valeur (VALUE).
La solution consiste à augmenter le nombre de segments d'annulation si le nombre d'attente d'une classe est supérieur à 1%.
Une règle simple pour définir la valeur du nombre de segments d'annulation : nombre d'utilisateurs / 4. De plus, lors de la création des segments d'annulation, il est préférable d'utiliser l'option OPTIMAL (afin d'avoir une taille optimale pour ces segments).

7.4.2. Taux d'occupation du processus dispatcher

Réduire le taux d'occupation du processus dispatcher.
Le diagnostic peut se faire à l'aide de la vue V$DISPATCHER (processus dispatcher) ; ses principaux champs sont le temps d'inoccupation des processus dispatcher (IDLE), le temps d'occupation des processus dispatcher (BUSY). On pourra aussi consulter la vue V$SHARED_SERVER (processus serveurs).
La solution consiste à augmenter le nombre de processus dispatcher (paramètres systèmes MTS_DISPATCHERS et MTS_SERVERS, paramètres MTS_MAX_DISPATCHERS et MTS_MAX_SERVERS) si le temps d'occupation est supérieur au temps d'inoccupation.

7.4.3. Temps d'attente du processus dispatcher ; processus partagés

Réduire le temps d'attente du processus dispatcher et les contentions des processus partagés.
Le diagnostic peut se faire à l'aide de la vue V$QUEUE ; ses principaux champs sont le temps total d'attente de toutes les réponses (WAIT), le nombre total de réponses (TOTALQ).
La solution consiste à augmenter le nombre de processus dispatcher si le temps moyen d'attente pour un protocole continue à s'accroître.

8. Outils : amélioration des performances, développement, administration

8.1. Trace de l'exécution d'une requête

8.1.1. Le paramètre SQL_TRACE

Le paramètre SQL_TRACE permet d'obtenir la trace de l'exécution d'une commande SQL dans un fichier contenant des informations sur les phases d'analyse, d'exécution et de recherche, sur le temps CPU et le temps écoulé, sur les lectures physiques et logiques, et sur le nombre de lignes traitées.
Le paramètre se positionne pour une session par la commande alter session set SQL_TRACE TRUE ou pour toutes les sessions par SQL_TRACE=TRUE dans le fichier d'initialisation.
Le paramètre USER_DUMP_DEST correspond au répertoire où sont générés les fichiers de traces d'exécution.
Il faut positionner le paramètre TIMED_STATISTICS à TRUE pour avoir des statistiques disponibles par SQL*DBA (menu Monitor, option Statistic).

8.1.2. L'utilitaire TKPROF

L'utilitaire tkprof ◁fichier de trace▷ ◁fichier en sortie▷ permet d'exploiter le fichier de traces d'exécution, globalement pour la session et dans le détail pour chaque phase d'exécution de la commande SQL : analyse (PRS), exécution (EXE) et recherche (FCH). Ces informations sont :

- (CNT) le nombre de fois que la commande a été analysée, exécutée et le nombre d'appels pour ramener le résultat (*count*)
- (CPU) le temps CPU (*cpu*)
- (ELA) le temps écoulé total (*elapsed*)
- (PHR) le nombre de blocs physiquement lus sur le disque (*physical read*)
- (CR) le nombre de fois qu'une ancienne version du buffer a été utilisé (*consistent read*)
- (CU) le nombre de fois que la valeur courante du buffer a été utilisé (*current*)
- (ROW) le nombre de lignes trouvées (*rows*)

Par exemple, le critère PRSCNT correspond au nombre de fois que la commande a été analysée ; de même, EXECR+EXECU est le nombre de lectures effectuées durant l'exécution (en modes consistant et courant) i.e. le nombre de blocs chargés en mémoire.
Pour tous les traitements transactionnels, les temps de réponse doivent être optimisés :

- EXECPU < 1 sec.
- PRSCPU < 1 sec.
- éviter l'utilisation de la table DUAL
- n'effectuer le parcours complet des tables que pour celles de petite taille
- choisir un sens pour les jointures

Pour tous les traitements, on vérifiera :

- si PRSCPU,PRSELA,PRSPHR sont importants relativement à EXECPU,EXEELA,EXEPHR,FCHCPU,FCHELA,FCHPHR alors
  diagnostic : beaucoup de temps est passé en phase d'analyse
  solution : augmenter la taille du dictionnaire des données
- si PRSCNT est trop important alors
  diagnostic : des curseurs fermés sont peut-être ré-analysés
- si EXEPHR+FCHPHR>1.1*(EXECR+EXECU+FCHCR+FCHCU) alors
  solution : augmenter la taille du cache base de données
- si FCHCNT≈2*FCHROW alors
  solution : éviter les curseurs implicites
- si une requête interactive est telle que PRSELA+EXEELA+FCHELA>250 (plus de 25 sec.) alors
  solution : écrire différemment ce traitement
- si EXECR est trop important et si FCHROW,PRSCU,EXECU,FCHCU sont beaucoup moins importants alors
  solution : créer des index sur les tables

8.1.3. La commande EXPLAIN

La commande explain plan set STATEMENT_ID='◁identifiant externe▷' for ◁requête d'interrogation▷ permet d'expliquer le plan d'exécution d'une requête d'interrogation.
Les résultats de l'explication sont placés dans la table PLAN_TABLE dont voici les principaux champs :

- STATEMENT_ID : identifiant externe donné par l'utilisateur de la commande à analyser
- ID : numéro d'une étape (identifiant interne)
- PARENT_ID : numéro d'une étape utilisant en entrée les sorties de l'étape de numéro ID
- POSITION : ordre des traitements au sein d'une même étape (ils ont même PARENT_ID)
- OPERATION : opération interne effectuée lors d'une étape
- OPTIONS : description supplémentaire éventuelle de l'opération

Un traitement classique (il s'agit d'une requête récursive : cf. start with et connect by prior) pour obtenir les résultats de l'explication :

```
-- saisie de l'identifiant externe affecté dans la variable ParamStatId
accept ParamStatId char prompt 'Indiquez la valeur du STATEMENT_ID : '
-- interrogation pour l'obtention des résultats de l'explication
select STATEMENT_ID , POSITION , ID , PARENT_ID , level ,
   lpad(' ',3*(level-1))||OPERATION||' : '||OPTIONS||' : '||OBJECT_NAME as SUITE
from PLAN_TABLE
start with ID=0 and STATEMENT_ID='&ParamStatId'
connect by prior ID=PARENT_ID and STATEMENT_ID='&ParamStatId';
```

8.2. Génération de statistiques destinées à l'optimiseur : la commande ANALYZE

La commande analyze {table ◁relation▷|index ◁index▷|cluster ◁cluster▷} {compute|estimate|delete} statistics permet de générer des statistiques utilisées par l'optimiseur pour la méthode statistique.
Les statistiques concernent les tables (nombre de lignes, nombre de blocs, nombre de blocs inutilisés, espace libre moyen, nombre de lignes chaînées, longueur moyenne des lignes), les index (niveau de l'index, nombre de blocs terminaux, nombre de clés distinctes, nombre moyen de blocs terminaux par clé, nombre moyen de blocs de données, valeur minimale de la clé, valeur maximale de la clé), les colonnes (nombre de valeurs distinctes).
Les statistiques peuvent être calculées pour chaque valeur (compute), estimées grâce au dictionnaire des données (estimate) ou supprimées (delete).

8.3. Outils de développement

8.3.1. Utilisation des tableaux

Les tableaux peuvent permettre de diminuer le nombre d'appels qu'une application fait à Oracle, notamment pour la phase de recherche.
La taille du tableau (paramètre ARRAYSIZE) conditionne le nombre de lignes à ramener en une seule fois : plus la taille est grande, moins il y a d'appels (mais au delà d'une certaine taille, l'intérêt devient négligeable comparé à la place mémoire requise).

8.3.2. Programmation en PL/SQL

- Définir des blocs pour grouper des commandes exécutées en une seule fois afin de décharger le trafic sur le réseau.
- Utiliser des curseurs explicites pour éviter un fetch supplémentaire (vérifiant qu'il n'y a plus de lignes).

8.3.3. Quelques paquetages

- Procédures pour les transactions discrètes dont les mises à jour sur les données sont différées après la validation de la transaction.
- Paquetage pour des alerteurs permettant à une application ou une session de se mettre en attente d'un événement et d'y réagir lorsqu'il survient.
- Utilisations de « *pipes* nommés » permettant à deux sessions de la même instance de communiquer.
- Sorties à partir des procédures et des déclencheurs.

8.4. Outils d'administration

8.4.1. L'utilitaire SQL*DBA

L'utilitaire SQL*DBA permet d'effectuer toutes les opérations d'administration (excepté l'échange de données) :

- démarrer et arrêter une instance
- charger et décharger une base de données
- ouvrir et fermer une base de données
- piloter en temps réel le fonctionnement d'Oracle
- sauvegarder et restaurer des données ou journaux
- exécuter toute commande SQL et PL/SQL

SQL*DBA peut être utilisé en mode menu (interface semi-graphique : `sqldba`) ou en mode ligne (saisie des commandes : `sqldba LMODE=Y`). Il faut ensuite se connecter avec l'utilisateur interne (commande `connect INTERNAL`).
Les commandes disponibles en mode menu sont :

- Session : connexion (`connect`) ou déconnexion (`disconnect`), exécution d'une commande du système d'exploitation (`host`), quitter temporairement SQL*DBA, quitter (`exit`)
- Instance : démarrage (`startup`), arrêt (`shutdown`), chargement, ouverture, point de vérification, changement de fichier de reprise, etc.
- Espace de tables : création ou suppression, activation ou désactivation, ajout de fichier, re-dénomination de fichier, modification des paramètres de stockage
- Segment d'annulation (*rbsegs*) : création ou suppression, activation ou désactivation, modification des paramètres de stockage
- Fichier de reprise (*logs*) : ajout ou suppression d'un groupe, ajout ou suppression ou re-dénomination d'un fichier d'un groupe, activation ou désactivation d'un groupe, démarrage ou arrêt de l'archivage automatique, début de l'archivage manuel, affichage des informations (`archive log`)
- Sauvegarde (*backup*) : début ou fin de sauvegarde d'un espace de tables actif, restauration d'une base de données ou d'un espace de tables ou d'un fichier de données (`recover`)
- Sécurité (*security*) : création ou modification ou suppression d'un utilisateur ou d'un rôle ou d'un profil, définition de ressources, attribution ou retrait de privilèges
- « monitoring » (*monitor*) : informations sur les processus (`monitor PROCESS`), statistiques sur les sessions (`monitor SESSIONS`), liste des tables utilisées (`monitor TABLE`), informations sur les loquets (*latch*) i.e. verrous internes utilisés par Oracle pour protéger des structures partagées de la mémoire (`monitor LATCHES`), informations sur les verrous (`monitor LOCKS`), activités de lecture/écriture sur chaque fichier (`monitor FILE`), pourcentage d'activité d'entrées/sorties des différents processus (`monitor I/O`), état des segments d'annulation (`monitor ROLLBACK`), statistiques générales sur l'instance (`monitor STATISTICS`)

Les autres commandes disponibles en mode ligne sont : `describe`, `execute` ◁`commande PL/SQL`▷, `print` ◁`variable`▷, `remark`, `set`, `show`, `spool`, `variable`.

8.4.2. L'utilitaire Import/Export

L'utilitaire Import/Export permet l'échange de données entre différentes bases Oracle. C'est un outil de migration, de défragmentation, et de transfert de données d'un environnement vers un autre.

8.4.2.1. L'importation

L'utilitaire Import (`imp`) permet l'importation de données à partir d'une base de données Oracle ; on parle de « restauration logique » à partir d'un fichier de commandes SQL de mises à jour de données.

8.4.2.2. L'exportation

L'utilitaire Export (`exp`) permet l'exportation de données à partir d'une base de données Oracle ; on parle de « copie logique » générant dans un fichier des commandes SQL de mises à jour de données.
Les objets concernés sont : les structures des tables, les données des tables, les privilèges, les vues, les clusters, les synonymes, les séquences, les contraintes d'intégrité.
Trois modes d'exportation sont possibles : mode table, mode utilisateur ou mode base de données.

Le mode d'exportation base de données peut concerner tous les objets y compris ceux appartenant à l'utilisateur `SYS` (*entire database*), tous les objets exceptés ceux de `SYS` (*full database*), incrémental i.e. concerner tous les objets y compris ceux de `SYS` et cela depuis la dernière exportation en mode incrémental ou cumulatif ou complet (*incremental*), cumulatif i.e. concerner tous les objets y compris ceux de `SYS` et cela depuis la dernière exportation en mode cumulatif ou complet (*cumulative*), complet i.e. concerner tous les objets y compris ceux de `SYS` et en remettant à zéro les compteurs pour les modes incrémental et cumulatif (*complete*).
Une bonne stratégie consiste à faire une première exportation en mode complet, régulièrement des exportations en mode incrémental, de temps en temps des exportations en mode cumulatif.

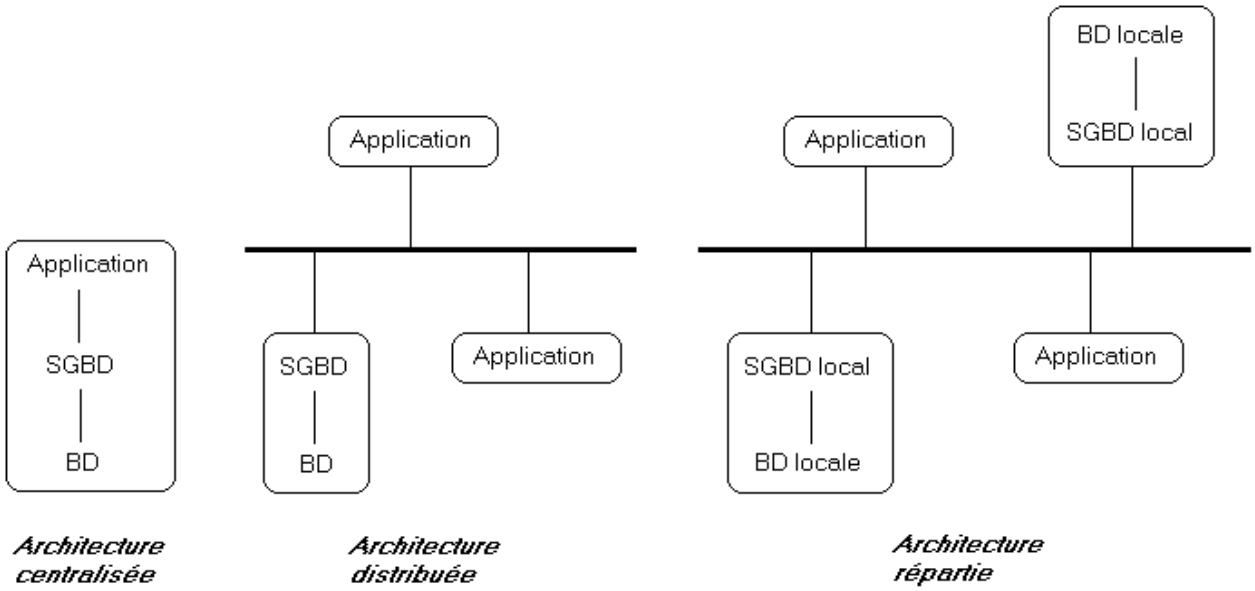
8.4.3. L'utilitaire SQL*Loader

L'utilitaire SQL*Loader (`sqlload`) permet de charger dans une base de données Oracle des données provenant de fichiers externes (générés par des logiciels autres qu'Oracle).
SQL*Loader utilise un fichier de contrôle et des fichiers de données, charge les données, et génère un fichier pour les enregistrements erronés, un fichier pour les enregistrements rejetés et un fichier journal.
Le fichier de contrôle contient la description des données à charger (noms et structures) et leur destination dans la base de données (liste des tables et colonnes où effectuer le chargement, correspondance entre les champs des fichiers de données et les colonnes). Il existe des utilitaires pour importer un fichier DBase, un fichier Lotus ou encore tout fichier ASCII.
Les fichiers de données sont au format binaire ou caractères, de longueur fixe ou variable.
Le fichier pour les enregistrements erronés correspond à des enregistrements n'ayant pas pu être interprétés.
Le fichier pour les enregistrements rejetés correspond à des enregistrements ne satisfaisant pas à une condition spécifiée dans le fichier de contrôle (clause `when`).
Le fichier journal est un compte-rendu de l'exécution de l'opération de chargement.

8.5. Outils pour un environnement réseau

Selon que les données et/ou les traitements sont situés sur un ordinateur central ou des ordinateurs distants, l'architecture diffère (cf. schéma ci-après) :

- architecture centralisée : les données et les traitements sont localisés sur un même ordinateur central
- architecture distribuée : les données sont localisées sur un ordinateur central tandis que les traitements sont localisés sur des ordinateurs distants
  On parle aussi d'architecture client/serveur.
  La solution proposée par Oracle est SQL*Net.
- architecture répartie : les données et les traitements sont localisés sur des ordinateurs distants
  On parle aussi de multi-bases de données.
  La solution proposée par Oracle est SQL*Star.



Oracle a une approche client/serveur de sorte que toute application ou outil Oracle nécessitant un accès à la base de données envoie une requête à la machine gérant la base de données ; cela autorise notamment la répartition des données (SQL*Net : connexion d'une application à un serveur telle que le client envoie sa requête sous forme de message que le serveur exécute et renvoie alors le résultat) et des traitements (SQL*Star = SQL*Net + noyau réparti pour gérer les requêtes multi-bases + SQL*Connect pour tout ce qui n'est pas Oracle).

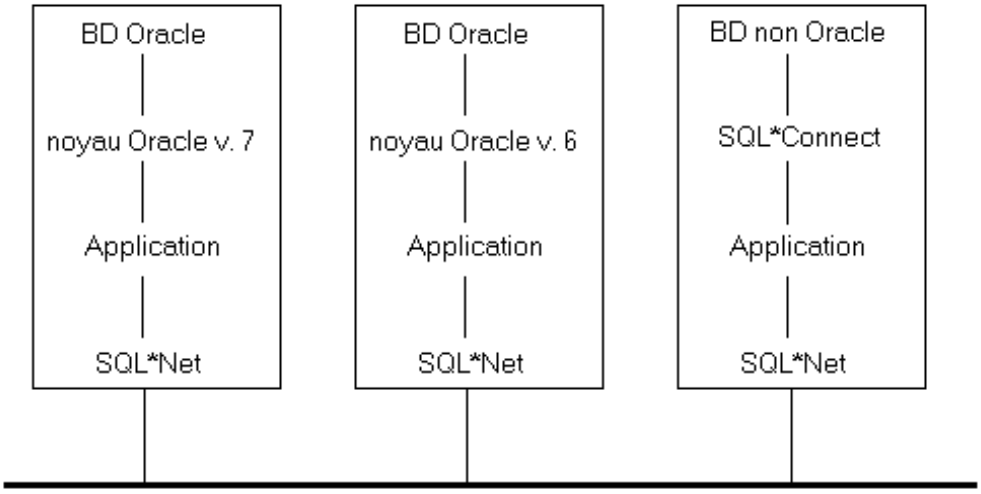
8.5.1. SQL*Net et l'architecture client/serveur

Dans cette architecture, le serveur est dédié à la gestion des données tandis que les applications sont les clients :

- le serveur, dès réception d'une commande, l'analyse, la traite et retourne le résultat au client,
- le client exécute l'application, transmet chaque requête au serveur et attend le résultat.

8.5.2. SQL*Star et les bases de données réparties

La base de données répartie est formée de l'union des bases de données locales gérées chacune par un SGBD local.
On parle de base de données répartie hétérogène si les SGBD locaux sont différents.
Oracle assure l'unicité de tous les objets : ◁`nom schéma`▷.◁`nom objet`▷@◁`nom base`▷.◁`adresse réseau`▷. Cette hiérarchisation des noms est telle que l'unicité est garantie à chaque niveau de la hiérarchie.
Un lien permet d'établir une liaison entre une base de données locale et une base de données distante par la commande `create database link` ◁`nom lien`▷ ; ce lien n'est disponible que pour les utilisateurs pouvant créer une session sur la base de données distante.
L'utilisation de procédures, synonymes, vues rend la localisation des données parfaitement transparente pour l'utilisateur et le programmeur.
Pour toute transaction répartie terminée, Oracle déclenche automatiquement le mécanisme de validation en deux phases (transparent pour l'utilisateur, le programmeur et l'administrateur) qui assure que la validation ou l'annulation de la transaction est faite dans toute les bases de données concernées (même en cas de panne car le traitement de la transaction est repris dès la restauration). La première phase de préparation consiste pour le site d'origine à demander à tous les autres sites concernés s'ils sont prêts tandis que la seconde phase de validation consiste pour le site d'origine à demander à tous les autres sites concernés, s'ils sont tous prêts, de valider la transaction. La table `DBA_2PC_PENDING` contient toutes les informations sur les transactions réparties en cours.
La réplication des données d'une base de données consiste à mettre à disposition d'autres bases de données une copie des données disponibles qu'en interrogation ; cette réplication peut être synchrone i.e. les copies sont mises à jour dès mise à jour de l'original ou asynchrone (*snapshot*) i.e. la mise à jour des copies se fait à intervalle régulier.
L'administration est autonome pour chaque SGBD local ; l'administration globale consiste à choisir la meilleure localisation des données, assurer l'unicité des noms, attribuer les privilèges lors de la création de liens, définir les contraintes d'intégrité globales, définir une politique de sauvegarde et de restauration.



2ème partie : Nouveautés des versions plus récentes d'Oracle

Cette partie présente les nouveautés des versions les plus récentes d'Oracle : 8, 9, 10 et 11.

9. Nouveautés des versions 8 et 8i

La version 8i s'est orientée vers Internet comparativement la version 8.

Oracle8i est, selon Oracle Corporation, « le SGBD Relationnel-Objet internet pour les systèmes classiques, transactionnels et décisionnels ». Ainsi, Oracle couvre les applications transactionnelles (OLTP, *online transaction processing*) et les entrepôts de données, et supporte les bases de données volumineuses (VLDB, *very large data bases*) et les applications critiques.

Les principales possibilités de la nouvelle version d'Oracle sont les suivantes :

- créer des applications client/serveur (les programmes sont sur les postes clients tandis que les données sont sur le serveur : la communication consiste uniquement à présenter une requête et à retourner les données),
- partitionner les applications (les applications sont réparties sur plusieurs machines mais accèdent toutes à la même base de données),
- répliquer les données (les informations sont disséminées dans plusieurs bases de données),
- gérer des solutions de type infocentre ou entrepôt de données (*data warehouse*) ou *data-marts* (rafraîchissement régulier du contenu, interrogation à des fins d'analyse, de consolidation, de décision),
- pour internet ou intranet (Oracle WebServer Manager configure les serveurs Web, Oracle Web Application Server sert des pages HTML stockées localement et peut exécuter des sous-programmes stockés dans la base de données, Oracle Web Publishing Assistant permet d'interroger une base de données et d'en publier le résultat dans des pages Web).

D'autres caractéristiques d'Oracle peuvent également être relevées.

- Système informatique à trois niveaux : le poste client (terminal graphique disposant d'un navigateur assurant la communication entre les applications et l'utilisateur) est connecté au serveur d'applications (qui gère la logique des traitements, les transactions, etc.) lui-même connecté au serveur de données (dédié à l'hébergement des bases de données). Oracle supporte Java à chacun de ces trois niveaux (client, serveur, médiateur).
- Utilisation indifféremment de Java ou de PL/SQL (par exemple, pour programmer un déclencheur) : une machine virtuelle Java a été intégrée dans Oracle.
- Oracle8i et Java : JDBC (Java Database Connectivity) ou SQLJ (SQL intégré dans du Java) ?
  JDBC (incorporé en standard dans Oracle) s'appuie soit sur ODBC, soit sur Net8, soit ni sur ODBC ni sur Net8.
  SQLJ (commercialisé séparément par Oracle, norme de trois entreprises : Oracle, IBM et SUN) nécessite une phase de pré-compilation des ordres SQL en code Java (utilisant une interface JDBC).
  Oracle Jdeveloper permet de développer en Java une application Oracle complète ; il est possible également d'utiliser les composants Enterprise JavaBeans et de s'appuyer sur le protocole de communication IIOP (norme de communication CORBA standard).
- Paradigme relationnel-objet : intégration de nouveaux types d'objets (types structurés munis de méthodes et de procédures stockées), gestion de données complexes i.e. de collections d'objets similaires (ensembles ordonnés de données de même type (*varrays, variable size arrays*), ensembles non ordonnés de données de même type (*nested tables*), etc.) bénéficiant des opérations de l'algèbre relationnelle étendue (utilisation indifféremment des identifiants d'objets ou des clés primaires, opérations de restructuration permettant notamment d'obtenir (*unnesting*) une relation de forme plate à partir d'une collection, etc.), gestion (avec *inter*Media) de données multimédia (texte, image, son, vidéo, coordonnées géographiques i.e. données spatiales) i.e. d'objets non structurés (*large objects*, LOBs). Il est possible d'effectuer des opérations de mises à jour et de gérer des déclencheurs pour ces données (tables et vues). Il est possible d'appeler des procédures externes depuis la base de données.
  Respect de la norme SQL3 standard pour la définition des types d'objets et les techniques de modélisation des objets (création/modification de types, génération/stockage des identifiants d'objets, création de références aux objets, etc.).
- Gestion des agrégats utilisant un mécanisme de stockage de plusieurs dimensions et un calcul automatique (de sous-totaux par exemple). Oracle Express Analyzer est un outil d'analyse multidimensionnelle (OLAP et ROLAP) intégrant les opérateurs `CUBE` et `ROLLUP` dans la clause SQL `GROUP BY`. Gestion des requêtes en étoile (*star queries*) i.e. des requêtes utilisant une (ou plusieurs) table(s) volumineuse(s) reliée(s) à de nombreuses tables de petites taille (*dimension tables*). Il est possible d'exécuter des opérations de mise à jour (insertion, modification, suppression) en parallèle, que les tables soient partitionnées ou non.
- Système de fichiers internet (*i*FS i.e. internet File System) permettant des manipulations faciles des données (recherche de texte par exemple).
- Fonction d'échantillonnage (*data mining*) pour les requêtes d'interrogation (clause `SAMPLE`).
- Désactivation/Activation des contraintes.

Enfin, citons quelques spécificités relevant de l'administration.

- Sécurité : gestion des ressources accédées par plusieurs utilisateurs selon un classement de priorité, contrôle d'accès granulaire, exécution d'un programme dans un contexte de sécurité paramétrable (par exemple, pour permettre à un client, selon la valeur de son code client, de consulter uniquement les factures le concernant), gestion d'utilisateurs à un niveau

intermédiaire (entre les utilisateurs du système et ceux ne pouvant pas franchir le pare-feu) dans le cadre d'environnement multi-tiers.
Oracle Secure Directory implémente SSL (Secure Socket Layer) v3 permettant l'authentification d'accès et la protection des données privées.

- Réplication des applications de front office : déploiement de masse facilité grâce à de nouveaux modèles de clichés (*snapshots*), possibilité de partitionner horizontalement ou verticalement les clichés afin de déployer la quantité minimale de données nécessaires à un site distant (de plus, les données sont localisées le plus proche possible des utilisateurs les manipulant).
- Partitionnement des tables et des index par hachage (répartition des données en partitions de tailles égales et uniformes sur différents périphériques) ou hybride (les données sont d'abord partionnées selon des plages de valeurs avant d'être hachées en plusieurs sous-partitions).
- Entrepôts de données (d*ata warehouse* ou *data mart*) : reconstruction automatique d'index en ligne (sans interruption des opérations de mise à jour), suivi d'opérations longues (construction d'index, sauvegarde de la base, etc.).
  Gestion de la base de données de secours : disponible constamment en lecture seule, fichiers de reprise (journaux de ré-exécution) lui sont automatiquement expédiés et appliqués.
- Optimiseur : copie des statistiques d'optimisation d'une base de données vers une autre, stockage des plans d'exécution en les rendant non modifiables.
- Transactions : modèle de publication/abonnement (transmission automatique des messages aux abonnés) utilisant des files d'attente et de messagerie afin de travailler en mode asynchrone avec d'autres systèmes (*Advanced Queuing*).
- Outils de diagnostics (via le service mondial d'assistance technique) pour l'analyse et le dépannage.
- Oracle Parallel Server : nouveau mécanisme de communication entre les instances (les blocs passent par l'interconnexion entre les nœuds et non par la lecture du disque), utilisation simplifiée (pour la localisation des erreurs, par des statistiques sur la coordination des travaux entre instances, etc.), équilibrage automatique de charge à la connexion, affectation possible d'un travail à un nœud spécifique.
- Espaces de tables transportables (i.e. copiables d'une base de données vers une autre) séparément.
- Nouveaux événements : arrêt/démarrage de la base de données, connexion/déconnexion utilisateur, erreur du serveur, création/modification/suppression d'une table.
- Gestion de plusieurs écrivains (`dbwriters`) sous la forme d'esclaves d'entrées/sorties (`i/o slaves`) ; aussi, le paramètre `DB_WRITERS` a pour nouveau nom `DB_WRITER_PROCESSES`.

Quelques uns des outils.

- Oracle Universal Installer (OUI) : outil graphique pour installer et mettre à jour les produits Oracle.
- Oracle Online Documentation : documentation d'Oracle en ligne.
- Oracle Enterprise Manager (OEM) : outil d'administration de tous les objets de la base de données.
- Oracle DataBase Assistant : aide à la création d'une base de données.
- Oracle Utilities : outils d'administration (Server Manager, Export/Import, SQL*Loader).
- Oracle WedDB : ensemble d'outils de développement (création, déploiement, administration) d'applications Web dynamiques.
- Oracle ConText Cartridge (extraction de texte, quel qu'en soit le format : HTML, Word, Excel, Acrobat, etc.), Oracle Image Cartridge, Oracle Spatial Cartridge : gestion des objets multimédia.
- Oracle Call Interface (OCI) : interface de programmation de bas niveau (proche d'Oracle).
- Oracle Migration Assistant for MS Access : outil de migration d'une base Access vers Oracle (l'application Access --- formulaires, états, macros, modules --- est conservée tandis que les tables Access sont attachées à des tables Oracle qui gère de plus les contraintes d'intégrité).

Quelques changements.

- L'outil SQL*DBA cède sa place à Server Manager et à Oracle Enterprise Manager.
- SQL*Net disparaît au profit de Net8.
- Les outils SQL*Forms, SQL*ReportWriter et SQL*Menu sont regroupés dans l'outil Developer/2000, avec la possibilité d'utiliser indifféremment les langages PL/SQL et/ou Java.
- Les outils de génie logiciel CASE*Dictionary, CASE*Designer et CASE*Generator sont quant à eux regroupés dans l'outil Designer/2000.
- Sont déclarés obsolètes : le compte `INTERNAL` (il faut utiliser les rôles `SYSOPER` et `SYSDBA`), l'instruction `SERIALIZABLE=TRUE`, l'équivalence entre la chaîne de caractères vide `''` et `NULL`, l'absence du privilège `select` avec au moins un privilège de mise à jour (`insert`, `delete`, `update`), etc.

Nous ne développerons par la suite que quelques unes des nouveautés (relevant uniquement de l'administration) : Oracle Enterprise Manager, Net8, les rôles `SYSOPER` et `SYSDBA`, Server Manager et l'architecture « cluster ».

Oracle Enterprise Manager

Oracle Enterprise Manager est une (unique pour la version 8, multiutilisateurs pour la version 8i) console graphique (et simple d'emploi) centralisée d'administration (et de supervision) de tous les objets de la base de données, disponible sous les différents systèmes d'exploitation, qui permet de lancer des utilitaires spécialisés (assistants dédiés).

Les agents de communication (agents intelligents) sont chargés de gérer les travaux et les événements sur la machine distante :exécution des travaux, surveillance des événements, réception des ordres de la console, transmission des informations à la console.
Il existe un mécanisme de découverte automatique des services d'une machine cible désignée.
Le référentiel (*repository*) est une base Oracle contenant toutes les informations de configuration et d'état des bases gérées (dans la version 8i, plusieurs consoles OEM partagent le référentiel commun).
Oracle Enterprise Manager se compose de quatre fenêtres :

- liste hiérarchique permettant d'administrer, de gérer les performances et de superviser les objets des différentes bases de données du réseau (bases, groupes, listeners, noms des serveurs, serveurs Web, etc.)
- image graphique permettant d'associer une base de données à un emplacement géographique
- système de travaux (*jobs*) : définition de travaux à accomplir sur des sites distants (transmission, historisation, visualisation des travaux en cours, contrôle de l'exécution et de la fin des travaux)
- gestionnaire des événements (*events*) : surveillance de certains points précis et possibilité d'être prévenu automatiquement en cas d'anomalie (voire même d'associer un travail correctif (*fixit job*) dès que l'événement se produit)

et de deux palettes d'outils :

- applications :
  - Oracle Backup Manager : gestion et automatisation des sauvegardes ; trois étapes sont nécessaires : préciser le lieu et la manière d'effectuer la sauvegarde (*channel*), configurer la sauvegarde à transmettre à l'agent intelligent sous forme d'un travail à réaliser, lancer l'assistant de sauvegarde
  - Oracle Data Manager : importation, exportation et chargement de tables (i.e. surcouche aux outils Import/Export et SQL*Loader)
  - Oracle Instance Manager : administration, démarrage, arrêt, visualisation des états des instances et réglage fin (*tuning*) (en modifiant certains paramètres d'initialisation)
  - Oracle Replication Manager : mise en œuvre, contrôle du fonctionnement et action (en cas de problème ou de conflit) de tous les aspects de la réplication
  - Oracle Schema Manager : accès et modification de toutes les caractéristiques du schéma (table, contrainte d'intégrité, index, déclencheur et procédure stockée, etc.)
  - Oracle Security Manager : accès à tous les paramètres de sécurité (mots de passe, privilèges, et les dépendances croisées entre comptes et objets)
  - Oracle SQL WorkSheet : environnement SQL interactif (beaucoup plus convivial que SQL*Plus !) pour visualiser les commandes, avec historisation des requêtes
  - Oracle Storage Manager : aspects physiques de la base de données (fichiers de données, fichiers de reprise, remplissage des espaces de tables, espace disponible, etc.)
  - Oracle Net8 Assistant : configuration du réseau (et création des alias pointant des bases de données cibles distantes)
  - Oracle Software Manager : télédistribution (déploiement) des logiciels
- modules optionnels (i.e. non livrés en standard) du Performance Pack pour l'analyse des performances et le réglage fin des bases :
  - Oracle Lock Manager : visualisation des verrous (au niveau des enregistrements et non des tables ou des blocs du disque) posés par les utilisateurs
  - Oracle Peformance Manager : tableau d'indicateurs de performance (nombre d'utilisateurs connectés, nombre d'utilisateurs actifs, etc.), caches (buffer, bibliothèque, dictionnaire des données, etc.), mémoire, etc.
  - Oracle Tablespace Manager : visualisation du contenu des espaces de tables, de la fragmentation des objets, de la place libre, etc. ; défragmentation avec l'outil Export/Import
  - Oracle Top Session : visualisation des sessions (machine, compte Oracle, compte du système d'exploitation, etc.), ordre SQL en cours (et passés), plan d'exécution des requêtes
  - Oracle Trace : trace les applications en y insérant des sondes
  - Oracle Export : analyse la base de données de production et donne un avis d'expert pour le réglage fin et l'amélioration des performances

Net8

Net8 est le logiciel médiateur (*middleware*) : il assure le lien, sur une machine (client ou serveur), entre l'application et le protocole réseau.
Sur le serveur, il s'agit d'un listener (logiciel qui écoute les demandes de connexions), commun à toutes les bases de données gérées par cette machine.
Sur chaque client, le fichier `Tnsnames.ora` définit un alias référençant l'instance administrée par le serveur. Cet alias précise le protocole réseau, la machine cible disposant de la base de données, l'identifiant de l'instance (SID) de la base de données cible, etc.

Les rôles SYSOPER et SYSDBA

Le compte `INTERNAL` permettant d'arrêter et de démarrer la base est déclaré obsolète (i.e. devrait disparaître dans les prochaines versions). Il convient maintenant d'utiliser les rôles `SYSOPER` et `SYSDBA`, attribués par `INTERNAL` ou `SYS` à des utilisateurs qui pourront à tout moment arrêter et démarrer la base. Ces utilisateurs, une fois connectés en tant que

`SYSOPER` (commande `connect ◁utilisateur▷/◁mot de passe▷ as sysoper`) n'auront cependant pas les privilèges des administrateurs de la base de données tandis que ceux connectés en tant que `SYSDBA` (commande `connect ◁utilisateur▷/◁mot de passe▷ as sysdba`) auront alors de surcroît les privilèges des administrateurs de la base de données.

Server Manager

Server Manager est une interface SQL interactive (mode texte) offrant quelques fonctionnalités supplémentaires par rapport à SQL*Plus : arrêt de la base (shutdown), démarrer la base (startup), se connecter avec le compte `INTERNAL`, etc.
Comparativement à SQL*DBA, Server Manager a perdu l'interface semi-graphique et le monitoring.

L'architecture « cluster »

L'architecture « cluster » (à ne pas confondre avec le terme cluster désignant un regroupement physique de tables ayant des colonnes communes) consiste à faire partager une base de données commune (sur un ou plusieurs disques) par des instances travaillant sur des machines différentes (elles-mêmes mono ou multiprocesseurs) ; c'est le seul cas où Oracle permet à un unique disque de données (i.e. une unique base) d'être partagé par plusieurs instances. Les instances disposent d'un mécanisme de synchronisation logicielle (pour garantir la cohérence des données).
Dans cette architecture matérielle où plusieurs serveurs (nœuds) indépendants sont reliés (par des canaux spéciaux à haut débit) pour fonctionner comme s'ils constituaient un seul serveur, les avantages sont nombreux : performance, fiabilité et disponibilité, évolutivité (par ajout de nœuds), administration unique des machines, etc. Les problèmes consistent par contre à gérer cette mise en commun de ressources, à vérifier le bon fonctionnement de l'ensemble, à répartir les charges, et à pallier aux pannes.
Oracle propose deux solutions : Oracle Fail Safe (en standard) garantissant la disponibilité et Oracle Parallel Server (en option) offrant disponibilité, performance et évolutivité.
Oracle Fail Safe : la base fonctionne sur un seul (des deux) serveurs à un instant donné et en cas de panne, l'instance est automatiquement relancée sur l'autre serveur (qui était inactif jusqu'à présent !).
Oracle Parallel Server : la base fonctionne simultanément sur l'ensemble des serveurs, l'ajout et l'arrêt de nœuds se fait dynamiquement, en cas de défaillance d'un nœud les autres nœuds assurent la continuité du service ; par contre, les applications doivent être modifiées pour préciser le nœud par défaut et le nœud auquel il faut se connecter en cas de défaillance du nœud par défaut.
Afin de paralléliser l'exécution d'un ordre SQL (lecture complète d'une table, tri, création d'index, chargement d'une table avec SQL*Loader, etc.), Oracle Parallel Query découpe l'ordre et le confie à plusieurs processeurs.

10. Nouveautés de la version 9i (et 8i)

Nouveautés SQL : nouvelles fonctions, fonctions analytiques, jointures, insertions multi-tables, fusions, stabilité du plan d'exécution.
Fonctionnalités de développement : extensions de types objet, améliorations du langage PL/SQL, Advanced Queuing, Java stocké, XML, référentiel XML DB.
Sécurité des données : contraintes d'intégrité, option *Fast Start*, blocs corrompus, destinations d'archivage multiples, Recovery Manager, LogMiner, Oracle Data Guard.
Sécurité des accès : gestion des contextes, bases de données virtuelles privées, Fine-Grained Auditing, Resource Manager, rôles applicatifs sécurisés.
Améliorations système : plafonds et nouveaux types, fichiers `init.ora` et `spfile`, multi-buffer pool, gestion de la mémoire, améliorations de l'optimiseur de requêtes, globalisation, Oracle Real Application Cluster.
Fonctionnalités d'administration : SQL*Plus pour l'administration, gestion des espaces de tables, réorganisation en ligne, reprises d'opérations, Oracle Managed Files, processus *job*, Export/Import des espaces de tables.
Gestion des gros volumes : LOBs, techniques de partitionnement des tables et index, tables temporaires, techniques d'indexation, tables organisées en index (IOT), Oracle Workspace Management, ETL (Extract, Transform, Load), vues matérialisées et dimensions, mode de réécriture des requêtes, contraintes sur vue.

11. Nouveautés de la version 10g

Cette version a introduit l'architecture Grille (i.e. *Grid*).

Outils : évolutions de SQL*Plus, export et import des données avec Data Pump.
Gestion du stockage : Automatic Storage Management (ASM), Tablespace SYSAUX, améliorations des espaces des tables, segments (partitionnement, sorted hash-clustered et stockage).
Exploitation : nouveaux outils d'installation et d'administration, ordonnanceur de travaux, nouveaux paramètres du système, Automatic Workload Repository (AWR), Diagnostic Monitor (ADDM), conseillers SQL, Resource management et consumer group, streams.
Haute disponibilité : Supplemental logging, Oracle Data Guard, Logminer, Real Application Cluster, redéfinition en ligne.
Optimisation : obsolescence de l'optimiseur sur les règles, collecte automatique des statistiques, gestion automatique de la SGA, amélioration du Wait Model, utilitaire TrcSess, SQL access advisor.

Sécurité : Flashback recovery area, Flashback database/table/query, Flashback row history et transaction history, RMAN (Fast incremental backup, incrémentaux sur les copies de fichiers, diverses evolutions), gestion des identités, Strong authentication (VPD, Oracle Label Security et Audit).
Gestion des gros volumes : déchargement de table externe, vues matérialisées et dimensions, nouvelles fonctions statistiques, CDC capture asynchrone.
Développement : amélioration du MERGE, jointure externe partitionnée, clause MODEL, requêtes insensibles à la casse, expressions régulières, améliorations du langage PL/SQL.

12. Nouveautés de la version 11g

Cette version est une évolution majeure qui contient plus de 400 nouvelles fonctions.
Oracle 11g supporte ainsi une multitude de nouveaux types d'objets qui permettent de gérer des données non structurées. Il s'agit, par exemple, d'images médicales, de présentations Powerpoint, de ressources multimédias (fichiers audio et vidéo) ou encore de données RFID auxquelles il associe des métadonnées spécifiques. Ces objets sont désormais gérés directement dans les bases via un système de fichiers très rapide (accès dit *fast file*).
Oracle annonce également un meilleur support des bases de données XML qui peuvent désormais être archivées et traitées au format binaire pour améliorer les performances.
Autre nouveauté, Oracle intègre des fonctions de compression qui divisent l'espace de stockage des données par trois (en revanche, les performances baissent lorsque la compression est activée).

La version Standard Edition

Cette version inclut Oracle Real Application Clusters.

Installation, configuration et administration

Installation rapide aussi bien pour une architecture « cluster » (dont *third-party clusterware*) que pour un environnement avec un seul serveur de données.
Préconfiguré pour être immédiatement utilisable et complet (gestion de la mémoire et de l'espace de stockage, sauvegarde et recouvrement automatiques, gestion des statistiques d'optimisation).
Enterprise Manager Database Control fournit une interface Web permettant de suivre l'état de la base de données et du cluster, d'administrer.

Données et applications complètes

Support de tous les types de données relationnels standards ainsi que XML (en stockage natif), texte, document, image, audio, vidéo, donnée géographique.
Accès via des interfaces standards : JDBC, SQLJ, ODBC .NET, OLE .NET et ODP .NET, SQL/XML et Xquery, WebDAV.
Les procédures stockées peuvent être écrites en Java, PL/SQL, .Net (avec CLR).
Possibilités de conception, statistiques, analyse à partir de n'importe quel environnement dit *SQL-based Business Intelligence*.

Développement

Oracle Application Express : unique environnement de développement en ligne pour construire une application (utilisant uniquement un navigateur Internet).
Oracle SQL Developer : outil de développement graphique gratuit permettant de parcourir les objets, écrire et lancer des ordres SQL, mettre au point des programmes PL/SQL.
Possibilité d'intégration dans Visual Studio grâce aux Oracle Data Access Components (pour des applications Oracle basées sur .NET et services Web).

Performance, Reliability, Security and Scalability

Idem Oracle Database Enterprise Edition (mêmes techniques de gestion de la concurrence) ! Lorsqu'un traitement tourne sur un cluster, il est automatiquement réparti sur les différentes machines disponibles (par Real Application Clusters), sans interruption (même si une machine du cluster devient indisponible).
Possibilité de données avec miroir, sauvegarde et recouvrement automatique.
Flashback Query : permet de voir et de récupérer des anciennes versions des données.
Les rôles et l'audit fournissent un grand contrôle d'accès afin de gérer la sécurité et la confidentialité des données.