

UNIVERSITE KASDI MERBAH OUARGLA

Faculté des Nouvelles Technologies de l'Information et de la Communication

Département d'Informatique et Technologie de l'information



Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique

Domaine : Mathématique et Informatique

Filière : Informatique

Spécialité : Informatique Industrielle

Présenté par : DJEBALI Hadjer

KENOUZE Ilyas

Thème :

Un Outil d'administration du Système de Gestion de Base de données Oracle (SGBD Oracle)

Soutenu publiquement

Juin 2014

Devant le jury :

M. BELOUAAR Houcine

Président

UKM OUARGLA

M. ZGA Adel

Examineur

UKM OUARGLA

M. DJEDIAI Hmida

Encadreur

UKM OUARGLA

Année Universitaire: 2013 / 2014

Remerciements

Nous remercions Allah le tout puissant, qui nous a donné la force et la patience pour l'accomplissement de ce travail.

Nous tenons à exprimer nos remerciements et notre profonde gratitude à notre encadreur Monsieur

M.DJEDIAI Hmida

pour son encadrement, son suivi et ces conseils tout au long de cette période.

Nous tenons aussi à remercier

Monsieur BELLOUAAAR Houcine président du jury,

et Monsieur ZGA Adel membre du jury

pour leur précieux temps accordé à l'étude de notre mémoire.

Bien entendu, Nous tenons surtout à remercier

nos parents

pour leurs sacrifices et leur patience,

tout au long de leurs vies.

Que toute personne ayant œuvré de près ou de loin à la réalisation de ce projet par une quelconque forme de contribution, trouve ici le témoignage de notre plus profonde reconnaissance.

Dédicaces

Je dédie ce mémoire :

*A mes adorables parents Amel et Mohammed Amine qui m'ont
toujours encouragé durant mes années d'étude et mes grands-parents mima, oumi
hayat et baba.*

*A ma sœur Rim et mes frères Imad, Abdelkader, Yahya et mes à qui je souhaite
beaucoup de réussite dans leurs études.*

A tout ma famille Djebale et Karoui.

A qui je souhaite beaucoup de bonheur et de réussite.

*A mes amies Karima, Samira, Fatiha, Fatima, Nour, Nacira, Khadidja, Amel, Lyna,
Halima, Asma, Maroua, Baraa, Djemaila, Imane, Nedjah, Kheira, Sara, Hana et à
toutes les personnes qui ont une place spéciale dans mon cœur et ma vie.*

A mon binôme dans mémoire. Ilyas et ma à souhaite beaucoup réussite dans son vie.

Et tous les étudiants 2eme mastre industriel

Merci

**DJEBALI
Hadjer**



Dédicace

*A mon cher père et ma chère mère
A mes chers Frère et mes chères sœurs
A tous mes oncles et tantes
A tous mes proches
A tous mes amis
A tous ce que j'aime
Ce mémoire leur est dédié.*

**KENOUZE
ILYAS**

Résumé

Le système de gestion de base de données c'est ensemble d'opérations dans une base de données et la multi-base de données ainsi que ces modes d'opérations.

La base de données Oracle a un système spéciale c'est le SGBDR et ce dernier à un Outil d'administration, mais l'utilisation de ces outils ce défère.

A la fin, on peut dire que cette application aura plus de chance dans le futur proche dans la base de données Oracle.

Mots clés :

Oracle, système SGBD,SGBDR, SQL, outil, base de données, outil d'administration.

Abstract

The basic management system of data it is together of operations in a database and multi-bases it data like these modes of operations.

The Oracle database has a system special it is the SGBDR and the latter with a Tool of administration, but the use of these tools this submits.

To the end, one can say that this application will have more chance in the near future in the Oracle database.

Key words:

Oracle, system DBMS, SGBDR, SQL, tool, database, tool of administration.

TABLE DES MATIERES

Introduction Générale	1
Chapitre I : Système de gestion de base de données (SGBD)	4
1. Introduction :	4
2. Système de gestion de bases données relationnelle (SGBDR):	4
3. Les caractéristiques d'un SGBD:	5
4. Différencier les SGBDR :	5
5. les systèmes de gestion bases de données :	6
5.1. ORACLE :	6
5.2. ACCESS Microsoft :	7
5.3. MYSQL :	9
5.4. SQL Server :	11
6. conclusion :	13
Chapitre II : Oracle	15
1. Historique :	15
2. Introduction :	15
3. Architecture Oracle :	16
3.1. Architecture Interne :	16
3.2. Architecture fonctionnelle :	20
4. Démarrer & Arrêter une base de données :	24
5. Gestion de l'instance et SPFILE :	27
5.1. Créer le fichier du paramètre SPFILE :	27
5.2. Exporter un fichier de paramètres serveur SPFILE :	28
5.3. Modifier des paramètres de l'instance ou du SPFILE :	28
5.4. Vues du dictionnaire de données:	29
6. la sécurité la base de données :	30
6.1. Le fichier de contrôle :	30
6.2. Protection des fichiers de Redo Log.....	30
7. Conclusion :	32
Chapitre III : Les Outils D'administration Oracle	34
1. Introduction :	34
2. Les outils administrateurs Oracle :	34
2.1. Oracle SQL Developer :	35
2.2. TOAD de Quest:	37

2.3. EMS SQL Manager for Oracle:	39
2.4. Oracle Enterprise Manager :.....	42
3. Conclusion :.....	44
Chapitre IV : Réalisation	46
1. Historique du langage JAVA :	46
2. Introduction :	47
2.1. Les différentes versions de java :	48
2.2. Caractéristiques et usage du langage Java :	48
2.3. Le cycle de développement :	49
2.4. Les avantages Java :.....	49
3. NetBeans :.....	49
4. l'interface JDBC :.....	50
5. Création, Modification et suppression :	53
6. Select, modification et supprime de la table :	54
7. L'interface Graphique de Notre application :.....	54
8. Conclusion:.....	60
Conclusion général	62
BIBLIOGRAPHIE	63

List de Figure

Figure II. 1 : Chronologie des versions d'Oracle	15
Figure II. 2 : Architecture Internet Oracle	17
Figure II. 3: RAC= 2instance face à une baie de disque	18
Figure II. 4 : environnement les outils d'Oracle	21
Figure II. 5 : démarrage et arrête d'une base de données	24
Figure III. 1 : Interface Oracle SQL Developer	35
Figure III. 2 : Interface Toad pour Oracle.....	37
Figure III. 3 : interface EMS SQL Manager for Oracle	41
Figure III. 4 : l'architecture OEM.....	43
Figure IV 1 : le cycle de développement	49
Figure IV 2 : l'interface de l'IDE Netbeans 7.4.....	50
Figure IV 3 : Architecture d'une application Java-JDBC-SGBD	51
Figure IV 4 : JDBC API standard	51
Figure IV 5 : JDBC API d'Oracle	51
Figure IV 6 : Programme JDBC	52
Figure IV 7 : Interface connexion	54
Figure IV 8 : Interface application	54
Figure IV 9 : interface crée index	55
Figure IV 10 : Interface Create table	55
Figure IV 11 : Interface create table space	56
Figure IV 12 : Interface Create Schema	56
Figure IV 13 : Interface Create Profile.....	56
Figure IV 14 : Interface create Role	57
Figure IV 15 : Interface create User	57
Figure IV 16 : Interface Create database link.....	57
Figure IV 17 : Interface drop table	58
Figure IV 18 : Interface drop tablespace	58
Figure IV 19 : Interface drop profile	58
Figure IV 20 : Interface drop rôle.....	59
Figure IV 21 : Interface drop User	59
Figure IV 22 : Interface drop databaselink.....	59
Figure IV 23 : Interface alter databaselink	59

List des tableaux

Tableau 1 : les requête pour Création et Modification et Suppression	53
Tableau 2 : les requete pour select et modification et suppression de la table	54

Introduction

Générale

Introduction Générale

Le développement actuel des applications informatiques de plus en plus tournés vers le transactionnel met en évidence une tendance assez générale des entreprises et des administrations à créer des bases de données et à utiliser des systèmes de gestion de bases de données (SGBD). Cette tendance se vérifie aussi bien dans le domaine traditionnel de l'informatique de gestion que dans les domaines plus nouveaux de l'information technique, économique, juridique ou scientifique pour lesquels on voit se multiplier des banques d'informations (appelées aussi banques de données) qui devraient connaître un usage croissant avec l'extension de la télématique et la pénétration de l'informatique dans les activités quotidiennes professionnelles ou domestiques. Dans le domaine de l'informatique de gestion, l'approche bases de données conduit à remettre en cause la manière de penser le système d'information et la façon de le construire.

Ils ont conduit à la définition de nouveaux modèles de données mieux adaptés à la construction de représentations pertinentes et de nouveaux langages facilitant la manipulation de ces représentations. Ces travaux ont contribué aussi à développer de nouvelles familles d'outils, Un grand nombre de systèmes de gestion de bases de données tels que l'Oracle, MySQL et Access Microsoft ...etc. Cela devrait permettre aux utilisateurs de construire et d'exploiter des systèmes de données tout simplement pour le domaine d'activité.

Mais la plupart des entreprises utilisent SGBD Oracle parce que les meilleurs systèmes de gestion de base de données et toutes les bases de données Oracle sont rétro-compatibles. Cela permet aux entreprises de mettre à niveau leurs systèmes sans une refonte complète de son système de base de données. Ceci fournit des mises à jour efficaces et à faible coût. En outre, de nouvelles versions de bases de données Oracle offrent de nouvelles fonctionnalités tout en conservant les caractéristiques populaires à partir des versions plus anciennes. Cela garantit que leur produit est basé sur la fonction du client plutôt que ce qui est rentable pour Oracle.

Les outils se retrouvent sur toutes les plateformes quelle que soit la version. Dans notre mémoire nous avons créé l'outil administration pour gérer la création,

Introduction Générale

les suppressions et la modification : (Index, Tablespace, DatabaseLink, Table, Profile, etc)

Notre mémoire est organisé de la façon suivante :

Le premier chapitre présente les concepts du Système de gestion de base de données SGBD.

Le second chapitre présente quelques notions sur le SGBD Oracle où sont décrites les différentes architectures.

Le troisième chapitre propose les outils d'administration Oracle.

Le quatrième chapitre est consacré à la contribution. Nous présentons dans cette partie le langage Java et son éditeur Netbeans IDE 7.4 et la version Oracle : (oracle Database 11g Express Edition), ainsi que notre application Outil d'Administration Oracle.

La conclusion et les perspectives de ce travail seront présentées à la fin du mémoire

Chapitre I :

System de Gestion De base De Données (SGBD)

Chapitre I : Système de gestion de base de données (SGBD)

Chapitre I : Système de gestion de base de données (SGBD)

1. Introduction :

Les Bases de données sont actuellement le cœur du système d'information, une base donnée est un ensemble de données qui ont été stockées sur un support informatique et organisées et structurées de manière à pouvoir facilement consulter et modifier leur contenu.

Une base de données seule ne suffit donc pas, il est nécessaire d'avoir également :

- Un système permettant de gérer cette base
- Un langage pour transmettre des instructions à la base de données (par l'intermédiaire du système de gestion).

Les systèmes de gestion de bases de données (data base mangement system) est un système qui permet de gérer une base de données qui partage les informations stockées (description, consultation, adjonction, modification, suppression et autorisation).

Dans ce chapitre nous allons étudier les différents SGBD. [1]

2. Système de gestion de bases données relationnelle (SGBDR):

Un SGBDR est un SGBD qui implémente la théorie relationnelle. MySQL à titre d'exemple implémente la théorie relationnelle, les données sont contenues dans ce qu'on appelle des relations, qui sont représentées sous forme de tables. Une relation est composée de deux parties, l'en-tête et le corps. [1]

Chapitre I : Système de gestion de base de données (SGBD)

3. Les caractéristiques d'un SGBD:

L'architecture à trois niveaux définie par le standard ANSI/SPARC permet d'avoir une indépendance entre les données et les traitements. D'une manière générale un SGBD doit avoir les caractéristiques suivantes:

- **Indépendance physique:** Le niveau physique peut être modifié indépendamment du niveau conceptuel. Cela signifie que tous les aspects matériels de la base de données n'apparaissent pas pour l'utilisateur, il s'agit simplement d'une structure transparente de représentation des informations
- **Manipulabilité:** des personnes ne connaissant pas la base de données doivent être capables de décrire leurs requêtes sans faire référence à des éléments techniques de la base de données
- **Rapidité des accès:** le système doit pouvoir fournir les réponses aux requêtes le plus rapidement possibles, cela implique des algorithmes de recherche rapides
- **Administration centralisée:** le SGBD doit permettre à l'administrateur de pouvoir manipuler les données, insérer des éléments, vérifier son intégrité de façon centralisée
- **Limitation de la redondance:** le SGBD doit pouvoir éviter dans la mesure du possible des informations redondantes, afin d'éviter d'une part un gaspillage d'espace mémoire mais aussi des erreurs
- **Vérification de l'intégrité:** les données doivent être cohérentes entre elles, de plus lorsque des éléments font références à d'autres, ces derniers doivent être présents
- **Partageabilité des données:** le SGBD doit permettre l'accès simultané à la base de données par plusieurs utilisateurs
- **Sécurité des données:** Le SGBD doit présenter des mécanismes permettant de gérer les droits d'accès aux données selon les utilisateurs [2]

4. Différencier les SGBDR :

Tous les SGBDR présentent à peu près les mêmes fonctionnalités. Ils se distinguent par : [2]

- leur coût,
- le volume de données qu'ils sont capables de gérer,
- le nombre d'utilisateurs qui peuvent interroger la base simultanément,
- la facilité avec laquelle ils s'interfaçent avec les autres logiciels d'application

Chapitre I : Système de gestion de base de données (SGBD)

5. les systèmes de gestion bases de données :

5.1. ORACLE :

5.1.1. Définition :

Oracle est un système de gestion de base de données relationnel (SGBDR) fourni par Oracle Corporation. Il a été développé par Lawrence Ellison, accompagné d'autres personnes telles que Bob Miner et Ed Oates. [Site 2]

5.1.2. les versions d'Oracle :

Oracle se décline en plusieurs versions [Site 1]

- Oracle Server Standard : une version comprenant les outils les plus courants de la solution Oracle. Il ne s'agit pas pour autant d'une version bridée.
- Oracle Server Enterprise Edition.

5.1.3. les fonctionnalités d'Oracle :

Oracle est un SGBD permettant d'assurer : [Site 1]

- La définition et la manipulation des données,
- La cohérence des données,
- La confidentialité des données,
- L'intégrité des données,
- La sauvegarde et la restauration des données, et
- La gestion des accès concurrents.

5.1.4. les outils d'administrateur d'oracle :

Il existe plusieurs Outils d'administration Oracle par en suite :

- Oracle Manager (SQL*DBA),
- NetWork Manager,
- Oracle Enterprise Manager,
- Import/Export : un outil permettant d'échanger des données entre deux bases Oracle, et
- Toad.

Chapitre I : Système de gestion de base de données (SGBD)

5.1.5. Les outils de développement d'Oracle :

Oracle propose également de nombreux outils de développement permettant d'automatiser la création d'applications s'interfaçant avec la base de données. Ces outils de développement sont :
[Site 1]

- Oracle Designer ,
- Oracle Developer,
- SQL*Plus : une interface interactive permettant d'envoyer des requêtes SQL et PL/SQL à la base de données. SQL*Plus permet notamment de paramétrer l'environnement de travail (formatage des résultats, longueur d'une ligne, nombre de lignes par page, ...) , et
- Oracle Developer : il s'agit d'une suite de produits destinés à la conception et à la création d'applications

5.2. ACCESS Microsoft :

5.2.1. Définition :

Microsoft Access (officiellement Microsoft Office Access) est un SGBD relationnel édité par Microsoft. Il fait partie de la suite bureautique MS Office Pro.

MS Access est composé de plusieurs programmes : le moteur de base de données Microsoft Jet, un éditeur graphique, une interface de type Query by Exemple pour manipuler les bases de données, et le langage de programmation Visual Basic for Applications.[Site 5]

5.2.2. Spécificités d'Access

Pour gérer les données, Access utilise cinq parties distinctes: [Site 6]

- Les tables reprennent tous les enregistrements avec leurs champs. A ce stade, les données ne sont pas traitées.
- Les requêtes reprennent les traitements sur les tables: filtrage et classement suivant les différents champs mais aussi l'affichage ou non des champs calculés à partir des autres champs.
- Les Formulaires permettent d'afficher à l'écran les fiches à partir des tables ou des requêtes, soit sous forme individuelles, soit sous forme de listes.

Chapitre I : Système de gestion de base de données (SGBD)

- Les états sont utilisés pour l'impression, de nouveau en récupérant les données à partir des tables ou des requêtes. Différentes possibilités de mises en page sont paramétrables. L'impression n'est pas directe comme dans un tableur.
- Les macros reprennent des programmations spécifiques sous forme de commandes préprogrammées ou programmées en langage VBA.

5.2.3. Les avantages d'Access : [Site 6]

- Pouvoir questionner une base de données,
- Manipuler les données pour les éditer dans l'ordre et dans la position que l'on désire
- Pouvoir faire des calculs
- Faire des mises à jour des fichiers
- Relations avec les autres programmes bureautiques (publipostage avec Word, liaisons vers des feuilles de calculs pour des calculs plus complexes)
- Possibilité d'automatiser sa base de données par la création de menu constituant un véritable panneau de contrôle pour l'utilisateur
- Possibilité de verrouiller certaines parties de l'application en définissant des autorisations
- L'abondance d'assistants.

5.2.4. les Inconvénients d'Access : [Site 6]

Si on est simple utilisateur, il n'y en a pas ; si on souhaite développer dessus, il faut alors considérer deux cas de figure :

1. Je désire développer une application : je dois alors maîtriser la conceptualisation des données
2. Je désire travailler simplement sur une base de données : formation obligatoire.

Chapitre I : Système de gestion de base de données (SGBD)

5.3. MYSQL :

5.3.1. Définition :

MySQL est un système de gestion de bases de données relationnelles. Le SQL dans “MySQL” signifie “Structured Query Language” : le langage standard pour les traitements de bases de données.

MySQL est Open Source. Open Source signifie qu’il est possible à chacun d’utiliser et de modifier le logiciel. Tout le monde peut le télécharger sur Internet et l’utiliser sans payer aucun droit. Toute personne en ayant la volonté peut étudier et modifier le code source pour l’adapter à ses besoins propres. Toutefois, si vous devez intégrer MySQL dans une application commerciale, vous devez vous procurer une licence auprès de MySQL AB. [Site 3]

5.3.2. Historique :

La première version de MySQL est apparue le 23 mai 1995. Il a d’abord été créé pour un usage personnel à partir de mSQL en s’appuyant sur le langage de bas niveau ISAM qu’ils trouvaient trop lent et trop rigide. Ils ont créé une nouvelle interface SQL en gardant la même API que mSQL.

MySQL est passé en licence GPL à partir de la version 3.23.19 (juin 2000)

- Version 4.0 : première version en octobre 2001, stable depuis mars 2003
- Version 4.1 : première version en avril 2003, stable depuis octobre 2004
- Version 5.0 : première version en décembre 2003, stable depuis octobre 2005
- Version 5.1 : première version en novembre 2005, Release Candidate distribuée depuis septembre 2007
- Version 5.2 : distribuée en avant-première (ajout du nouveau moteur de stockage Falcon) en février 2007, cette ligne a ensuite été renommée 6.0
- Version 5.5 : Version stable et depuis octobre 2010
- Version 5.6 : Version stable et depuis février 2013
- Version 6.0 : première version alpha en avril 2007, abandonnée depuis le rachat de MySQL par oracle en décembre 2010

En 2003, MySQL AB et SAP AG concluent un accord. Le résultat s’appellera MaxDB. Il est issu de l’intégration du système de SAP (SAP DB) dans MySQL.

Chapitre I : Système de gestion de base de données (SGBD)

Le 20 avril 2009, Oracle Corporation annonce racheter Sun Microsystems pour 7,4 milliards de dollars, créant la crainte de voir MySQL disparaître ou du moins ne plus être développé au profit d'Oracle, le SGBD d'Oracle Corporation. MySQL est cependant loin d'avoir toutes les fonctionnalités d'Oracle, et pourrait donc être vu au contraire par la société comme un produit gratuit d'appel banalisant l'usage de SQL et préparant donc ses futures ventes. [Site 3]

5.3.3. Outils de gestion, d'administration et de conception pour MySQL :[Site 3]

- phpMyAdmin, outil de gestion de bases de données MySQL écrit en PHP
- Adminer, un autre outil de gestion lui aussi écrit en PHP
- MySQL administrator, logiciel d'administration de bases de données
- MySQL Workbench (versions MySQL > 4) et anciennement DBDesigner (versions 3 et 4), système de conception de base de données sous forme graphique qui intègre tous les niveaux de conception, de modification, de création et de maintenance sur la structure d'une base de données
- MySQL-Front, outil d'administration de base de données MySQL
- SQL Buddy

Chapitre I : Système de gestion de base de données (SGBD)

5.3.4. les avantages : [3]

- Open Source.
- Fiable et relativement performant.
- Riche fonctionnellement.
- Simple d'utilisation et d'administration.
- Héritage de tables.

5.3.5. les inconvénients [3]

- Supporte les bases de moyenne importance.
- Sauvegardes peu évoluées.
- Gestion des permissions limitées.

5.4. SQL Server :

5.4.1. Définition :

Microsoft SQL Server est un système de gestion de bases de données relationnelles édité et commercialisé par Microsoft depuis 1994.

Microsoft SQL Server est une plate-forme de données d'entreprise permettant de gérer et stocker dans des bases de données tout type d'information : [Site 4]

- Données structurées : données relationnelles par exemple
- Données non structurées : documents, images, ...

5.4.2. Fonctionnalités principales de Microsoft SQL Server [Site 04]

- Gestion de bases de données relationnelles
- Gestion et déploiement centralisé de plusieurs instances et applications depuis un seul point de contrôle
- Optimisation de stockage des bases de données volumineuses (tables et indexes partitionnées, compression de données, ...)
- Prise en charge des données géographiques
- Gestion de la haute disponibilité
- Ordonnanceur intégré (SQL Agent)
- Service de notification
- Gestion de la réplication
- Prise en charge de la virtualisation

Chapitre I : Système de gestion de base de données (SGBD)

5.4.3. Dans quels cas utiliser Microsoft SQL Server ? [Site 4]

- Mise en place d'applications de gestion
- Mise en place d'applications décisionnelles
- Mise en place de Data Warehouses complexes et optimisés (de dix à quelques centaines de To de volume)

5.4.4. Les Outils d'administration SQL Server : [Site 4]

- SQL Server Management Studio (SSMS)
- SQL Server Management Studio Express Edition (SSMSE)
- SQL Server Business Intelligence Development Studio
- Entreprise manager
- Query Analyzer (Analyseur de requête)
- Profiler (générateur de profil)

5.4.5. Avantages de Microsoft SQL Server : [Site 4]

- SQL Server intègre par défaut des outils de gestion, d'administration et de développement de bases de données.
- Déploiement par un setup, mise en œuvre et administration par des interfaces graphiques intuitives.
- Programmabilité
- Gestion avancée de la sécurité en offrant deux modes d'authentification (Authentification Windows et Authentification SQL Server)
- Prise en compte des spécificités des projets décisionnels (Parallélisations de Datawarehouses, ...)
- Coût relativement moins cher par rapport aux autres SGBD du marché

Chapitre I : Système de gestion de base de données (SGBD)

6. conclusion :

on s'appuie sur un système de gestion de base de données relationnelle (SGBDR) qui est un logiciel pour organiser les éléments d'information (fichiers, textes, images) au sein d'une structure en tables liées entre-elles. [2]

Ce SGBDR se chargera d'assurer l'intégrité des données

Leur contenu est accessible par des applications clients via des langages de requêtes, exécutés via des interfaces : ODBC, JDBC. Mais le langage le plus performants c'est l'ORACLE car : [2]

- Supporte les très grosses bases.
- Pérennité (40% de part de marché).
- Interface utilisateur très riche.
- Procédures en PL/SQL ou java (un + pour les équipes de développement) .
- Flashback query (retour à une daSte).
- Système de vues.
- Services web.

Chapitre II :

Oracle

Chapitre II : Oracle

1. Historique :

La société Oracle Corporation a été créée en 1977 par Lawrence Ellison, Bob Miner, et Ed Oates. Elle s'appelle alors Relational Software Incorporated (RSI) et commercialise un Système de Gestion de Bases de données relationnelles (SGBDR ou RDBMS pour Relational Database Management System) nommé Oracle.

En 1979, le premier prototype (RDBMS - RSI1) intégrant la séparation des espaces d'adressage entre les programmes utilisateurs et le noyau Oracle est commercialisé. Cette version est entièrement développée en langage assembleur. La seconde version (RDBMS - RSI2) est un portage de l'application sur d'autres plates-formes. [Site 1]

1979 Oracle 2	Première version commerciale écrite en C/assembleur pour Digital – pas de mode transactionnel.
1983 Oracle 3	Réécrit en C - verrous.
1984 Oracle 4	Portage sur IBM/VM, MVS, PC – transaction (lecture consistante).
1986 Oracle 5	Architecture client-serveur avec SQL*Net – version pour Apple.
1988 Oracle 6	Verrouillage niveau ligne – sauvegarde/restauration – AGL – PL/SQL.
1991 Oracle 6.1	<i>Parallel Server</i> sur DEC.
1992 Oracle 7	Contraintes référentielles – procédures cataloguées – déclencheurs – version Windows en 1995.
1994	Serveur de données vidéo.
1995	Connexions sur le Web.
1997 Oracle 8	Objet-relationnel – partitionnement – LOB – Java.
1998 Oracle 8i	<i>i</i> comme Internet, SQLJ – Linux – XML.
2001 Oracle9i	Services Web – serveur d'applications – architectures sans fil.
2004 Oracle10g	<i>g</i> comme <i>Grid computing</i> (ressources en <i>clusters</i>).
2007 Oracle11g	Auto-configuration.

Figure II. 1 : Chronologie des versions d'Oracle

2. Introduction :

Oracle est un SGBD (système de gestion de bases de données) édité par la société du même nom (Oracle Corporation - <http://www.oracle.com>), leader mondial des bases de données.

3. Architecture Oracle :

3.1. Architecture Interne :

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

- **Une instance** est constituée : [6]
 - D'une zone de mémoire partagée appelée System Global Area (SGA)
 - D'un ensemble de processus d'arrière-plan ayant chacun un rôle bien précis
 - D'un ensemble de processus serveur chargés de traiter les requêtes des utilisateurs.
- **La base de données** est l'ensemble des fichiers qui permettent de gérer les données de la base.

Une base de données est constituée de : [6]

- Un fichier de contrôle, contenant les informations sur tous les autres fichiers de la base (nom, emplacement, taille).
- Fichiers de REDO LOG, contenant l'activité des sessions connectées à la base. Ce sont des journaux de transactions de la base. Ils sont organisés en groupe possédant le même nombre de membres.
- Et éventuellement, de fichiers de Redo Log archivés contenant les archives d'anciens fichiers de Redo Log.
- D'un ou plusieurs fichiers de données qui contiennent les données des tables de la base.

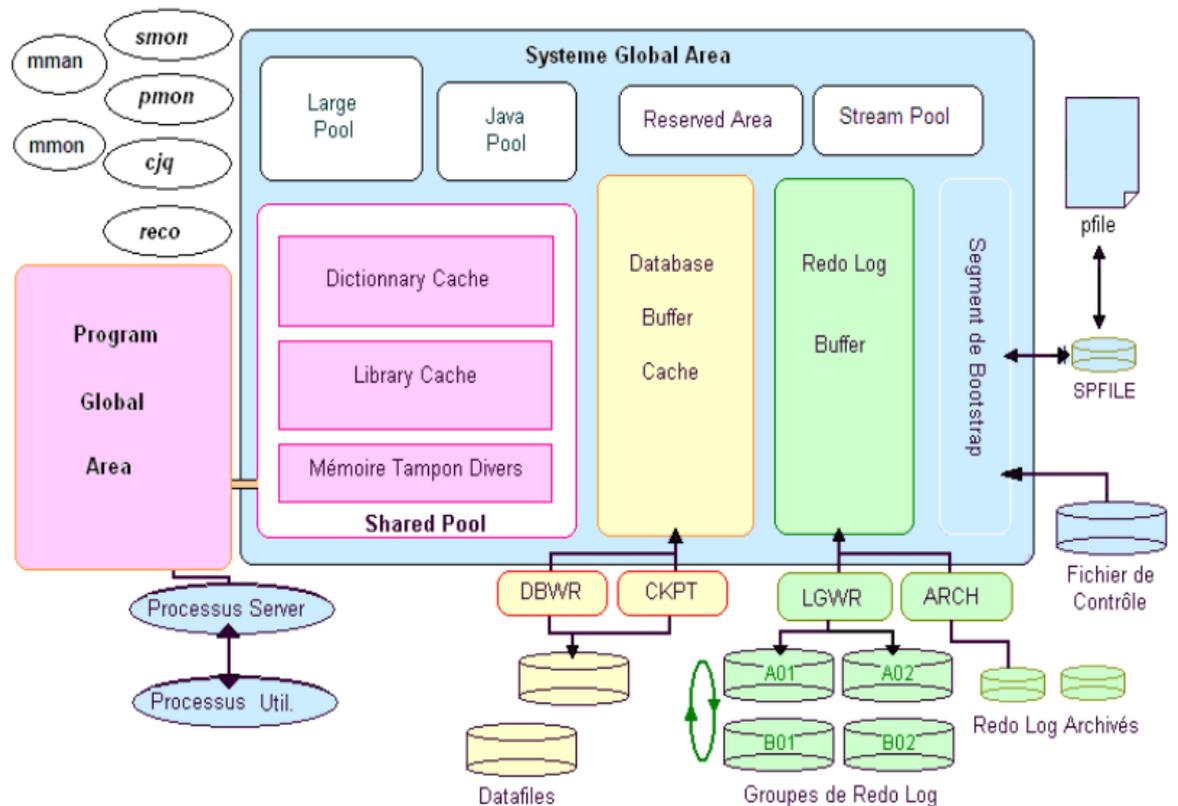


Figure II. 2 : Architecture Interne Oracle [6]

Une instance est l'ensemble des processus d'arrière-plan (background process) et de zones mémoire qui est allouées au démarrage de la base de données, pour permettre l'exploitation des données. [6]

Une instance ne peut ouvrir qu'une seule base de données à la fois et dans la grande majorité des cas, une base de données est ouverte par une seule instance. [6]

Néanmoins, moyennant la mise en œuvre de l'option RAC (Oracle Real Application Clusters), permettant d'utiliser Oracle sur des serveurs en cluster, une base de données peut être ouverte par plusieurs instances situées sur des nœuds distincts d'un cluster de serveurs cette option est intéressante pour la haute disponibilité mais elle est relativement complexe à mettre en œuvre. [6]

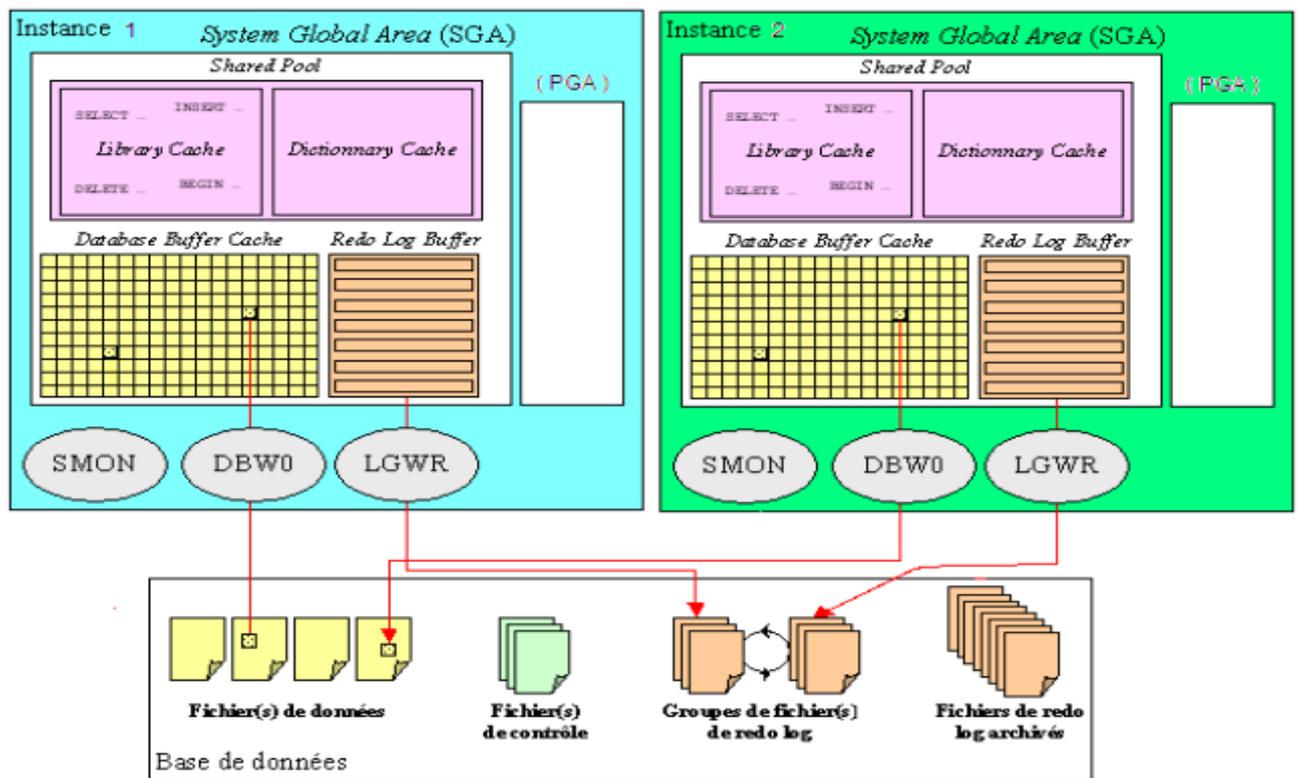


Figure II. 3: RAC= instance face à une baie de disque [6]

3.1.1. 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 c.-à-d. 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) :** Mémoire privée des différents processus distribuée au moment de la connexion d'un client.[5]

Pour un processus serveur, la PGA contient : [6]

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

3.1.2. 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 [5]

- 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 les fichiers de données

- LGWR (log writer) : gestion de la zone de reprise et écriture des tampons de reprise dans le fichier d
- 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'un con multithread server)
- DNNS (dispatcher) : partage les processus serveurs pour les processus utilisateurs
- LCK0, ..., LCK9 (LOCK) : gèrent le verrouillage inter-instances (cas d'un serveur parallèle).

Gestion multiprocessus : [5]

- architecture combinée : un processus utilisateur pour chaque utilisateur (remplissant aussi les fonctions des 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.

3.2. 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.[5]

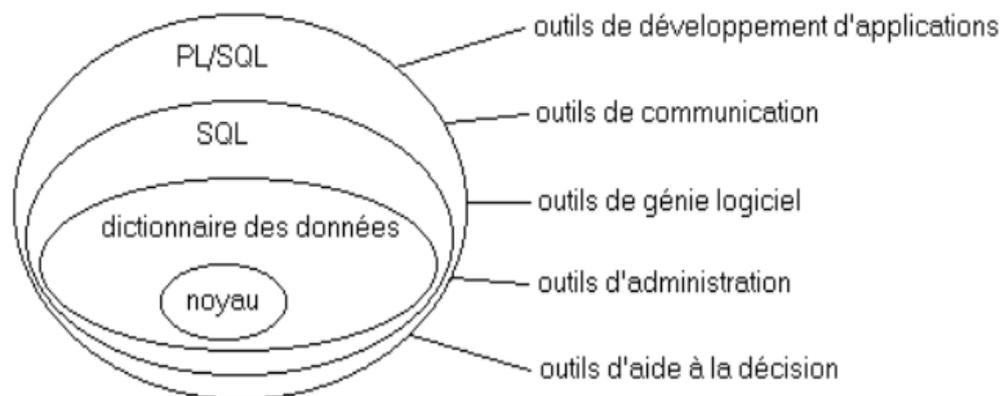


Figure II. 4 : environnement les outils d'Oracle [5]

Description :

Noyau : communication avec la base de données, connexion à d'autres noyaux (cas d'une base répartie) [5]

- 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 (c.-à-d. structure centralisée contenant la description de tous les objets gérés par Oracle) [5]

Objets de la base : [5]

- 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 maîtresse 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 : redénomination 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 [5]

- 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) [5]

- **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. [5]

- **PL/SQL** : extension procédurale du langage SQL [5]
- **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

4. Démarrer & Arrêter une base de données :

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

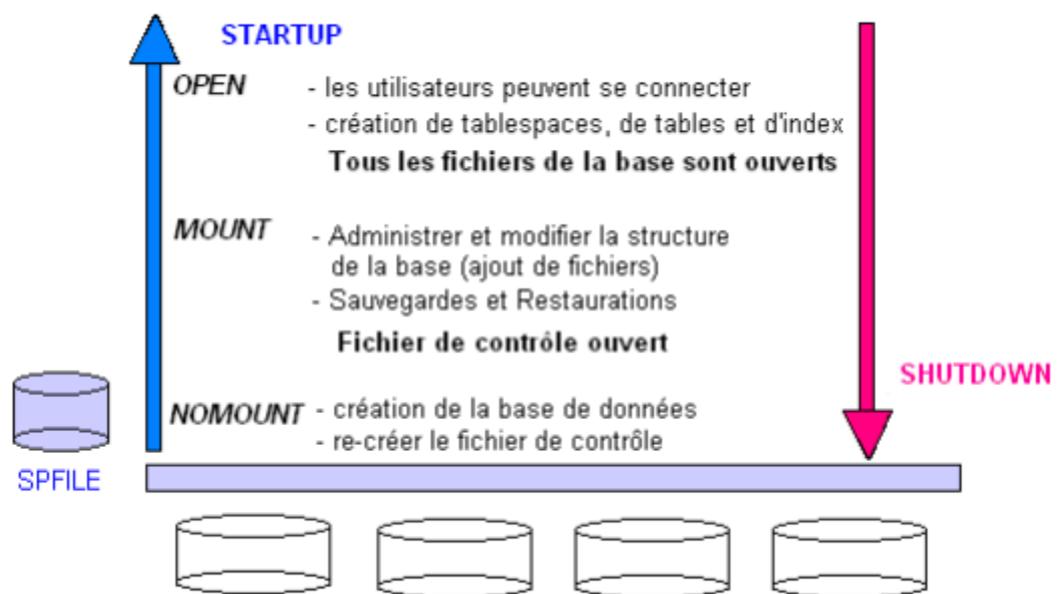


Figure II. 5 : démarrage et arrête d'une base de données [6]

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

Il y a trois étapes dans le processus de démarrage :

1. Démarrage de l'instance
2. Montage de la base
3. Ouverture de la base

Un fichier de paramètres SPFILE est lu lors du démarrage de l'instance. Il permet de configurer les paramètres de l'instance. [6]

```
SQL> startup

Instance ORACLE lancé.

Total System Global Area 135338868 bytes

Fixed Size          453492 bytes
Variable Size       117440512 bytes
Database Buffers    16777216 bytes
```

De même, il y a trois étapes dans le processus d'arrêt : [6]

1. Fermeture de la base
2. Démontage de la base
3. Arrêt de l'instance

```
SQL> shutdown immediate

Database closed.

Database dismounted.

ORACLE instance shut down.
```

4.1. Démarrer la base de données :

La commande STARTUP permet de démarrer une instance et de lui associer une base de données avec le niveau de disponibilité souhaité.[6]

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

- **NOMOUNT | MOUNT | OPEN** : niveau de disponibilité souhaité
- **Nom_base** : nom de la base à monter ou à ouvrir
- **RESTRICT** : restreint l'accès à la base aux utilisateurs ayant le privilège RESTRICTED SESSION
- **PFILE** : nom du fichier de paramètres à utiliser. [6]

4.2. Fermeture d'une base de données Oracle :

La commande SHUTDOWN permet d'arrêter l'instance et la base de données.

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

- **NORMAL** : Oracle attend que tous les utilisateurs soient déconnectés (pas de nouvelle connexion autorisée) puis ferme proprement la base.
- **IMMEDIATE** : Oracle déconnecte tous les utilisateurs (en effectuant un ROLLBACK des éventuelles transactions en cours) puis ferme proprement la base.
- **TRANSACTIONNAL** : Oracle attend que toutes les transactions en cours se terminent avant de déconnecter les utilisateurs (pas de nouvelle transaction autorisée) puis ferme et démonte proprement la base.
- **ABORT** : Oracle déconnecte tous les utilisateurs (sans effectuer de ROLLBACK des éventuelles transactions en cours) puis ferme brutalement la base ; une restauration de l'instance sera nécessaire lors du prochain démarrage.[6]

4.3. Vues du dictionnaire de données : [6]

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

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

La vue dynamique V\$MEMORY_TARGET_ADVICE :

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

Cette vue contient les colonnes : [6]

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

5. Gestion de l'instance et SPFILE :

Au démarrage, l'instance lit un fichier de paramètres binaire SPFILE qui contient des paramètres d'initialisation. Ce fichier est géré par le DBA.[6]

Il s'agit d'un référentiel centralisé des paramètres d'initialisation de l'instance au démarrage de la base de données en binaire qui permet d'effectuer des modifications de paramètres pendant le fonctionnement de l'instance (sans avoir besoin d'arrêter la base de données. [6]

Règles concernant l'écriture des paramètres :

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

Ces paramètres sont pris en compte directement en mémoire (paramètres dynamiques) ou uniquement dans le SPFILE (paramètres statiques). Dans dernier ce cas il faut arrêter puis redémarrer la base de données pour que la modification soit prise en compte par l'instance.[6]

5.1. Créer le fichier du paramètre SPFILE :

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

```
CREATE SPFILE [= 'nom_pfile' ] FROM PFILE [= 'nom_spfile' ]. [6]
```

5.2. Exporter un fichier de paramètres serveur SPFILE :

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

```
CREATE PFILE [= 'nom_pfile' ] FROM SPFILE [= 'nom_spfile' ] ;[6]
```

5.3. Modifier des paramètres de l'instance ou du SPFILE :

L'ordre SQL ALTER SYSTEM permet de modifier dynamiquement la valeur des paramètres d'initialisation [6]

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

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

5.4. Vues du dictionnaire de données:

Plusieurs vues du dictionnaire permettent de visualiser les paramètres : [6]

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

La vue dynamique V\$SYSTEM PARAMETER

Cette vue dynamique de performances, permet de la valeur des paramètres de l'instance. [6]

Cette vue contient les colonnes :

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

6. la sécurité la base de données :

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

Cette sécurité est assurée par la mise en œuvre d'une protection des fichiers sensibles de la base de données : [6]

- Fichiers de contrôle
- Fichiers de redo log

6.1. Le fichier de contrôle :

Lorsqu'une instance est lancée pour ouvrir une base de données, le fichier de contrôle est le premier fichier ouvert, il permet ensuite à l'instance de localiser et d'ouvrir les autres fichiers de la base de données. Si on perd le fichier de control, la base de données reste à l'état NOMOUNT, et ne pourra pas s'ouvrir.[6]

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

Un fichier de contrôle contient les informations suivantes :

- Le nom et l'identifiant de la base de données
- Le nom et l'emplacement des fichiers de données et des fichiers Redo Log
- Le nom des tablespaces
- La date et l'heure de création de la base de données
- Le numéro de séquence du journal courant
- Des informations relatives au point de synchronisation
- L'historique du journal
- Les informations de sauvegarde de l'utilitaire Recovery Manager. [6]

6.2. Protection des fichiers de Redo Log

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

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

Ils sont utilisés pour la restauration de la base après un arrêt anormal de celle-ci. Ils peuvent être réappliqués à une sauvegarde de fichier de données, pour rejouer toutes les modifications survenues entre la sauvegarde et un incident ayant endommagé le fichier (c'est la restauration de média). [6]

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

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

Mettre au minimum deux ou trois membres par groupe. [6]

7. Conclusion :

Outre la base de données, la solution Oracle est un véritable environnement de travail constitué de nombreux logiciels permettant notamment une administration graphique d'Oracle, de s'interfacer avec des produits divers et d'assistants de création de bases de données et de configuration de celles-ci.

On peut classer les outils d'Oracle selon diverses catégories :

- Les outils d'administration
- Les outils de développement
- Les outils de communication
- Les outils de génie logiciel
- Les outils d'aide à la décision

Chapitre III.

Les Outils D'administration

Oracle

Chapitre III : Les Outils D'administration Oracle

1. Introduction :

D.B.A. = Data Base Administrator = Administrateur de bases de données. Il est responsable du bon fonctionnement des bases de données de l'entreprise (bases de développement, test et production).

Les taches et les Rôles d'un administrateur De bases de données : [Site 9]

- Installer le logiciel, faire les mises à jour (patches, changement de version du noyau mais aussi des autres produits),
- Créer les bases de données physiques et gérer l'espace physique,
- Gérer les utilisateurs et leurs droits d'accès,
- Valider les schémas de données (cohérence, non redondance, optimisation)
- Assurer la sécurité de la base (sauvegarde, restauration, confidentialité d'accès),
- Gérer les ressources systèmes et optimiser les performances
- Faire les transferts de données de et vers d'autres systèmes.

Oracle est fourni avec de nombreux outils permettant de simplifier l'administration de la base de données. Parmi ces outils, les plus connus sont : [Site 9]

- Oracle Manager (SQL*DBA)
- NetWork Manager
- Oracle Enterprise Manager
- Import/Export : un outil permettant d'échanger des données entre deux bases Oracle
- Oracle SQL Developer
- TOAD de Quest© For Oracle.

2. Les outils administrateurs Oracle :

2.1. Oracle SQL Developer :

2.1.1. Introduction :

Oracle SQL Developer est une application graphique permettant d'exécuter des requêtes ou scripts SQL, de gérer les objets d'une base de données (table, vue, etc.), développer et mettre au point des programmes PL / SQL. [7]

La fenêtre principale d'Oracle SQL Developer à l'allure suivante :

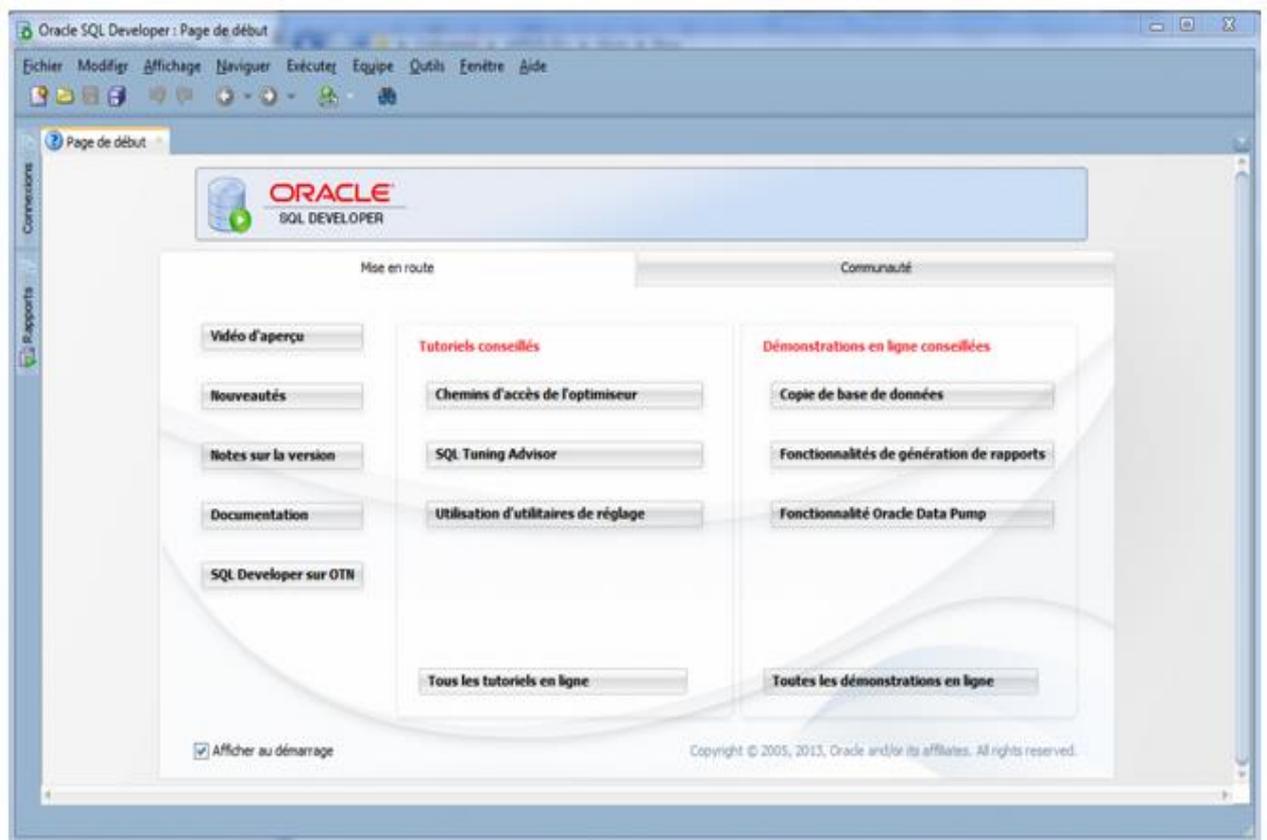


Figure III. 1 : Interface Oracle SQL Developer

2.1.2. Historique :

La version 1.0 du logiciel est lancée en mars 2006, bien que des versions antérieures existent sous le nom de Raptor.

SQL Developer évolue vers sa version 1.5 en avril 2008 (prise en charge des contrôles de versions de type CVS et SVN), sa version 2.1 en décembre 2009 (test unitaire de PL/SQL et visualiseur de modélisation de données), sa version 3.0 en mars 2011 (nouveau constructeur de requêtes, fonctionnalités pour DBA) et sa version 4.0 en décembre 2013 (nouveaux outils de reporting).[Site 10]

2.1.3. Fonctionnalité :

Oracle SQL Developer permet le développement de A à Z d'applications en PL/SQL, la mise à disposition de feuilles de travail pour exécuter les requêtes et les scripts, une console pour l'administration de bases de données (DBA), un interface pour la génération de rapports (reporting), un solutionne complète de conception du modèle de données et un interface de migration permettant de migrer les bases de données d'éditeurs tiers vers Oracle.[Site 10]

Oracle SQL Developer supporte les produits Oracle ainsi que des plugins qui permettent de se connecter à des bases de données non Oracle. Oracle SQL Developer fonctionne avec IBM DB2, Microsoft Access, Microsoft SQL Server, MySQL, Sybase Adaptive Server, et les bases de données Teradata. [Site 10]

2.2. TOAD de Quest:

2.2.1. Introduction :

Toad (Tool for Oracle Application Developers) est un logiciel de la société Quest Software qui permet de consulter et d'administrer une base de données. Il est utilisé en particulier par les développeurs Oracle et les administrateurs de bases de données.

[Site 11]

La fenêtre principale Toad à l'allure suivante :

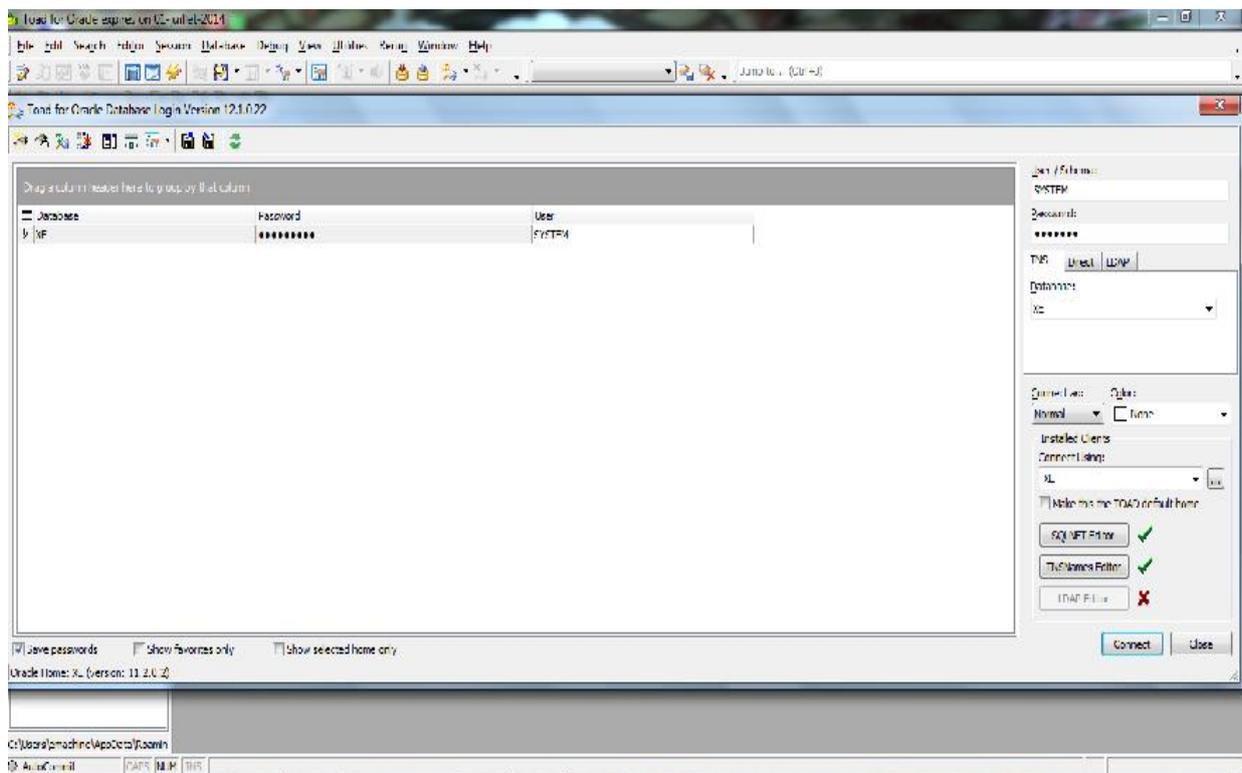


Figure III. 2 : Interface Toad pour Oracle

2.2.2. Version de Quest Toad pour Oracle :

Toad est disponible dans les versions suivantes:

Toad for Oracle : fournit les outils fondamentaux et basiques aux professionnels d'Oracle pour créer et exécuter des requêtes, ainsi que de créer et gérer les objets de bases de données. Cette édition permet de disposer des outils de mise en forme du code et d'une interface améliorée, et du support pour SQL Plus. Il est possible d'intégrer des fonctionnalités optionnelles de TOAD for Oracle.[Site 12]

Toad for Oracle Professional : (inclut les fonctions de Toad for Oracle auxquels s'ajoutent le Debugger intégré, la base de connaissances PL/SQL, l'analyse dynamique et statistique du code).

Développer du code PL/SQL est un processus fastidieux et nécessitant une très grande précision. Les composants de Toad for Oracle Professional intègrent un éditeur de procédure unique, permettant de comparer les procédures stockées à un ensemble de règles ; des recommandations correctives détaillées sont ensuite proposées.

Toad for Oracle Xpert : (inclut les fonctions Toad for Oracle Professional ainsi qu'une solution d'optimisation SQL et PL/SQL appelé CodeXpert). Toad for Oracle Xpert simplifie les tâches en identifiant les causes potentielles de baisse de performance. Le composant SQL Optimisation réduit le temps d'optimisation pour les développeurs, administrateurs de bases de données et les analystes.

ToadDevelopment Suite for Oracle : fournit un ensemble d'outils intégré garantissant que le code produit est de qualité, optimisé et écrit pour une meilleure performance – indépendamment du niveau de compétences de l'utilisateur .your database.[Site 12]

2.2.3. Les techniques d'administration d'une base avec Toad : [Site 12]

- Vérification de l'état de santé de la base
- Réglage de certains paramètres
- Le monitoring UNIX : surveillance des process
- Gestion de l'espace, les tablespaces
- Estimation des tailles de tables et d'index
- Exportation et importation
- Les informations de session, identifier et tuer une session
- Les utilitaires de REDO log
- Comparaison et synchronisation de deux schémas
- Les utilisateurs et les droits
- Les statistiques d'utilisation de la base

2.2.4. L'Avantage :

Automatise les tâches courantes et réduit les coûts de développement et d'administration

- Contribue à réduire le coût total de possession (TCO) de l'environnement Oracle et à accélérer les cycles de développement, en identifiant les problèmes, avant qu'ils n'affectent les utilisateurs finaux
- Réduit des risques d'indisponibilité et de dégradation des performances, et renforce la capacité à respecter les engagements de niveau de service (SLA)
- Permet de s'assurer que les nouvelles applications répondent aux objectifs métiers et affichent les performances attendues en production.

[Site 13]

2.2.5. À propos de Quest Software :

Afin d'aider ses clients à faire face à la complexité croissante de leur système d'information et à concilier gain de productivité, adaptabilité et sécurité, Quest Software fournit des solutions logicielles pour simplifier l'administration, optimiser les performances et assurer la disponibilité de leurs applications, bases de données et infrastructures. Pour plus d'informations www.quest.com [Site 12]

2.3. EMS SQL Manager for Oracle:

EMS SQL Manager for Oracle est un outil à hautes performances de développement et d'administration de bases de données Oracle. SQL Manager for Oracle fonctionne avec toutes les versions d'Oracle à partir de la 8.1.7 et supporte toutes les nouvelles fonctionnalités de serveur Oracle ainsi que les types de données, y compris les tables compressées, les tables OLAP, les tables en lecture seule et lecture/écriture; les index invisibles, les triggers composés, les triggers avec la clause FOLLOWS et d'autres. Le programme fournit beaucoup d'outils Oracle puissants tels que le débogueur de code PL/SQL; l'assistant de sauvegarde/restauration de base de données, le concepteur visuel de bases de données pour créer les bases de données Oracle en quelques clics; le constructeur visuel de requêtes pour créer les requêtes SQL Oracle complexes, les statistiques de base de données pour surveiller les performances d'Oracle, le gestionnaire des droits pour la gestion efficace de la

sécurité d'Oracle, et beaucoup d'autres fonctionnalités utiles pour l'administration efficace d'Oracle. SQL Manager for Oracle dispose d'une interface utilisateur graphique et d'un système des assistants avec la description détaillée dont l'usage est simple même pour les novices. [Site 14]

2.3.1. Caractéristiques clés : [Site 14]

- Support des dernières versions d'Oracle
- Gestion des bases de données et navigation rapide
- Outils puissants qui rendent votre travail avec le serveur Oracle plus facile que possible
- Assistants conviviaux pour exécuter les tâches administratives d'Oracle
- Gestion facile de tous les objets Oracle (y compris les opérations de création/modification/ suppression)
- Gestion de la sécurité
- Outils avancés de manipulation de données
- Outils excellents visuels et outils texte pour la construction des requêtes
- Possibilités impressionnantes d'importer et d'exporter les données
- Concepteur visuel de bases de données
- Concepteur de rapports comprenant l'assistant de construction de rapports facile à utiliser
- Connexion via port local au travers du tunnel SSH
- Support de données Unicode
- Nouvelle interface utilisateur graphique de pointe
- Autres fonctionnalités.

2.3.2. L'Avantage : [Site 14]

- Un an de maintenance GRATUIT est déjà inclut
- Mises à jour mineures et mises à jour majeures gratuites pendant toute la période de maintenance
- Support technique GRATUIT pendant toute la période de maintenance
- Renouvellement de maintenance à un prix très avantageux - à partir de 20% par an
- Remises sur le montant de commande avec l'achat de deux copies du produit ou plus

- Remises sur les ventes croisées avec l'achat de produits apparentés
- Remboursement garanti sous 30 jours.

2.3.3. Captures d'écran

Nouvelle interface utilisateur graphique ultra moderne Support des options visuelles Office11 et Windows XP

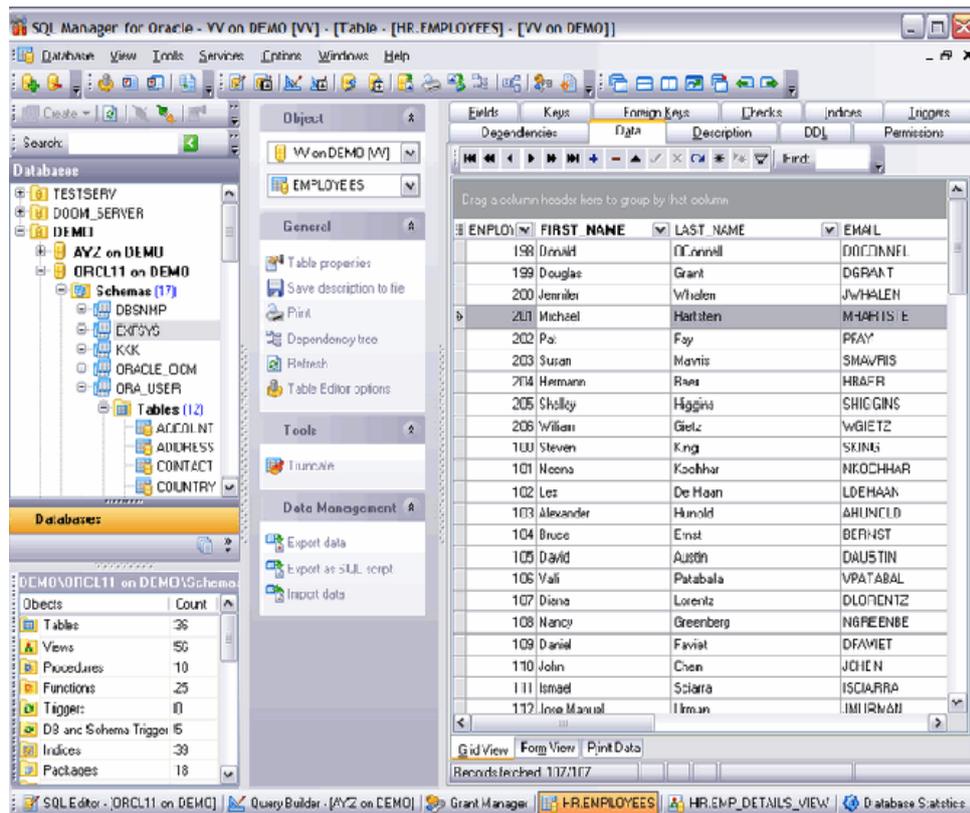


Figure III. 3 : interface EMS SQL Manager for Oracle. [Site 14]

2.3.4. Versions Disponibles

- **EMS SQL Manager for Oracle** : EMS SQL Manager for Oracle est un outil graphique puissant de développement et d'administration de bases de données Oracle. L'interface conviviale et les fonctionnalités puissantes rendent votre travail avec le serveur Oracle plus facile que possible.
- **EMS SQL Manager for Oracle Freeware** : EMS SQL Manager for Oracle Freeware (EMS SQL Manager Lite for Oracle) est un excellent outil graphique gratuit d'administration des bases de données Oracle. Il offre l'ensemble d'outils et de fonctionnalités

nécessaires pour les utilisateurs novices d'Oracle qui ont besoin uniquement des fonctionnalités de base. En tant qu'application gratuite SQL Manager Freeware dispose de certaines restrictions : la taille maximum de bases de données enregistrées ne doit pas dépasser 4Gb, ainsi que beaucoup d'autres restrictions. [Site 14]

2.4. Oracle Enterprise Manager :

2.4.1. Introduction :

Oracle Enterprise Manager est un ensemble d'outils qui utilisent une interface graphique et qui simplifient la gestion des différents objets de la base de données. Il permet de centraliser l'administration de plusieurs bases de données installées sur des serveurs différents implantés dans des environnements d'exploitation différents (Unix, Windows...).

Le produit Oracle Enterprise Manager permet la création d'un serveur Oracle Management Server pour gérer des bases de données et administrer des travaux, des événements et des groupes. Toutefois il n'est pas nécessaire pour pouvoir gérer plusieurs bases de données à l'aide d'un même outil.[Site 15]

2.4.2. L'architecture d'OEM :

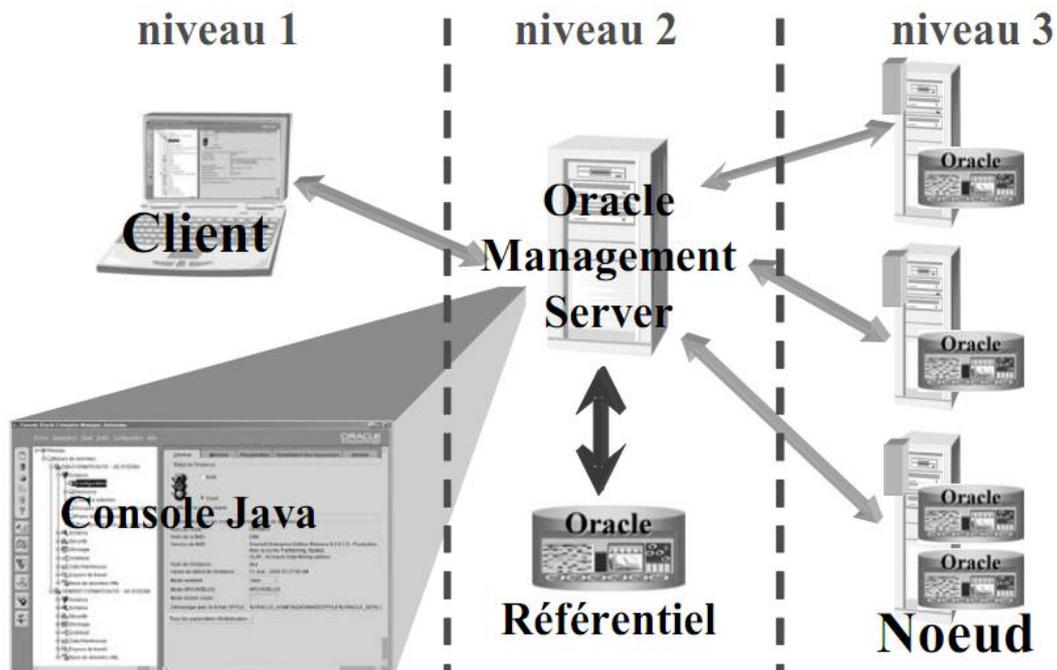


Figure III. 4 : l'architecture OEM

OEM (Oracle Enterprise Manager) dispose d'une architecture sur trois niveaux :

- **Premier niveau** : le client, est doté d'une console Java, d'outils intégrés ou simplement d'un navigateur Web.
- **Second niveau** : le serveur Management Server ou le groupe de serveurs Management Server défini offre des fonctions d'intelligence centralisée et de contrôle distribué entre les clients et les cibles.
- **Troisième niveau** : avec différentes cibles, telles que les bases de données, les nœuds ou d'autres services gérés ; Intelligent Agent est installé sur chaque nœud, en surveille les services pour les événements inscrits (occurrences d'erreur potentielle), et exécute les travaux que la console envoie via les serveurs Oracle Management Server. [Site 15]

3. Conclusion :

Les outils de base de données servant à créer une base de données et à valoriser au maximum vos données. Il existe différents outils de base de données pour créer des formulaires, trier des données, rechercher des enregistrements, concevoir des rapports, etc.

Nous étudions les autres administrations Oracle mais dans le dernier chapitre 04 nous créons nouveau programme pour l'outil administration.

Chapitre IV.

Réalisation

Chapitre IV : Réalisation

Dans le présent chapitre nous allons présenter notre solution qui consiste à créer une application d'administration Oracle. Cette solution est implémentée en langage JAVA en utilisant l'éditeur NetBeans.

1. Historique du langage JAVA :

On peut faire remonter la naissance de Java à 1991. A cette époque, des ingénieurs de chez SUN ont cherché à concevoir un langage applicable à de petits appareils électriques (on parle de code embarqué). Pour ce faire, ils se sont fondés sur une syntaxe très proche de celle de C++, en reprenant le concept de machine virtuelle déjà exploité auparavant par le Pascal UCSD. L'idée consistait à traduire d'abord un programme source, non pas directement en langage machine, mais dans un pseudo langage universel, disposant des fonctionnalités communes à toutes les machines. Ce code intermédiaire, dont on dit qu'il est formé de byte codes¹, se trouve ainsi compact et portable sur n'importe quelle machine ; il suffit simplement que cette dernière dispose d'un programme approprié (on parle alors de machine virtuelle) permettant de l'interpréter dans le langage de la machine concernée.

En fait, ce projet de langage pour code embarqué n'a pas abouti en tant que tel. Mais ces concepts ont été repris en 1995 dans la réalisation du logiciel Hot Java, un navigateur Web écrit par SUN en Java, et capable d'exécuter des applets écrits précisément en byte codes.

Les autres navigateurs Web ont suivi, ce qui a contribué à l'essor du langage qui a beaucoup évolué depuis cette date, sous forme de versions successives : 1.01 et 1.02 en 1996, 1.1 en 98 et 1.2 (finalement rebaptisée Java 2) en 1999, 1.3 en 2000, 1.4 en 2002, 1.5 en 2004 (toujours appelées Java 2). [8]

2. Introduction :

Langage conçu par les ingénieurs de « SUN Microsystem » (société rachetée par Oracle en avril 2009)

Définition de SUN : "Java est un langage simple, orienté objet, distribué, robuste, sûr, indépendant des architectures matérielles, portable, de haute performance, multithread et dynamique" [9]

- Simple
- Inspiré du C++, Fortran, Lisp, Small talk
- Pas de pointeur; pas de surcharge d'opérateurs; pas d'héritage multiple
- Présence d'un "garbagecollector"
- Orienté objet
- La programmation objet modélise des objets ayant un état (ensemble de variables) et des méthodes (fonctions) qui leur sont propres. L'unité de base en Java est la classe. Un des intérêts de Java est de disposer de nombreuses classes déjà faites. Un objet crée à partir d'une classe est une instance.
- Distribué
- Les fonctions d'accès au réseau et les protocoles internet les plus courants sont intégrés.
- Robuste
- Typage des données très strict
- Pas de pointeur
- Sûr
- Java n'est pas compilé à destination d'un processeur particulier mais en « byte code » qui pourra être ensuite interprété sur une machine virtuelle Java (JVM = Java Virtual Machine). Le "byte code" généré est vérifié par les interpréteurs java avant exécution.
- Un débordement de tableau déclenchera automatiquement une exception.
- L'absence d'arithmétique de pointeur évite les malversations.
- Portable
- Les types de données sont indépendants de la plate-forme (par exemple les types numériques sont définis indépendamment du type de plate-forme sur laquelle le byte code sera interprétée).

- Haute performance
- Java est un langage pseudo interprété
- Techniques de "Just in Time" (JIT) améliorent ces performances
- Multi thread
- Une application peut être décomposée en unités d'exécution fonctionnant simultanément
- Dynamique
- Les classes Java peuvent être modifiées sans avoir à modifier le programme qui les utilise.

2.1. Les différentes versions de java [9]

- ✓ Java 1.0 (23 janvier 1996)
 - 8 Packages
 - 212 Classes et Interfaces
 - 1545 Méthodes
- ✓ Java 1.1 (19 février 1997)
 - 23 Packages
 - 504 Classes et Interfaces
 - 3 851 Méthodes
- ✓ Java 1.2 (Java 2) (9 décembre 1998)
 - 60 Packages
 - 1 781 Classes et Interfaces
 - 15 060 Méthodes
- ✓ Et bien plus dans les versions 1.3 (8 mai 2000), 1.4 (6 février 2002), 1.5 (tiger, 30 septembre 2004), 1.6 (mustang, 11 décembre 2006)
- ✓ Dernière version: 1.7, le 28 juillet 2011 (Open Source et sous l'ère Oracle).
- ✓ Prochaine version: 1.8, le 18 mars 2014.

2.2. Caractéristiques et usage du langage Java : [Site 16]

- Indépendance de la plateforme portabilité
- Fait pour le réseau: applets, servlets, libraries réseau
- beaucoup de librairies (packages)
- Strictement typé ("typed")
- Fait par Sun (mais plusieurs autres implémentations).

2.3. Le cycle de développement :

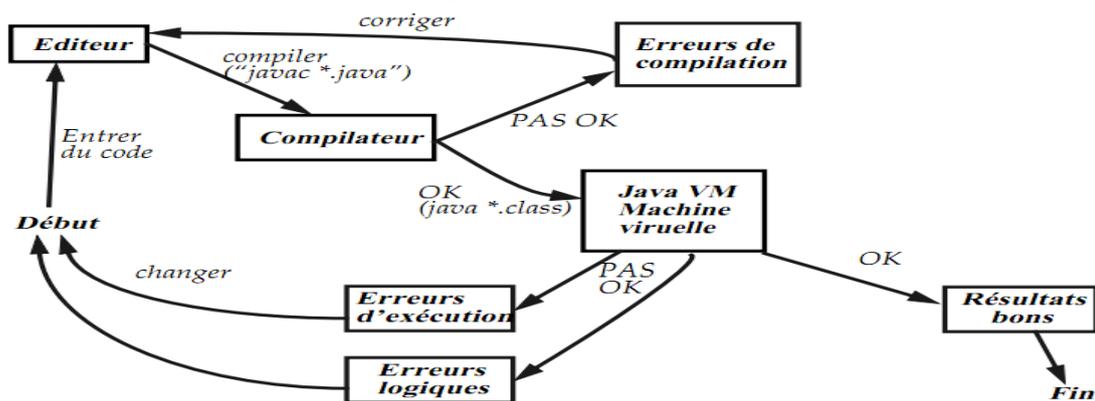


Figure IV 1 : le cycle de développement [Site 16]

2.4. Les avantages Java :

- Orienté objet, Interprété et portable, Multithread,
- Fortement et statiquement typé, à éditions de liens
- dynamique, Simplifié, Sécurisé
- Gérant la mémoire :
 - ✓ La libération de mémoire est effectuée automatiquement par le Ramasse-miette (Garbage Collector).
 - ✓ Pas de pointeurs
- Facile à apprendre et à programmer (Un temps de développement réduit d'un tiers par rapport à du C++ pour les développeurs expérimentés)
- Une documentation abondante sur le WEB. [10]

3. NetBeans :

NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL et GPLv2 (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une

machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java.

NetBeans constitue par ailleurs une plate-forme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plate-forme.

L'IDE Netbeans s'enrichit à l'aide de plugins.

Il comprend toutes les caractéristiques d'un IDE moderne (coloration syntaxique, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages web, etc). [Site 17]

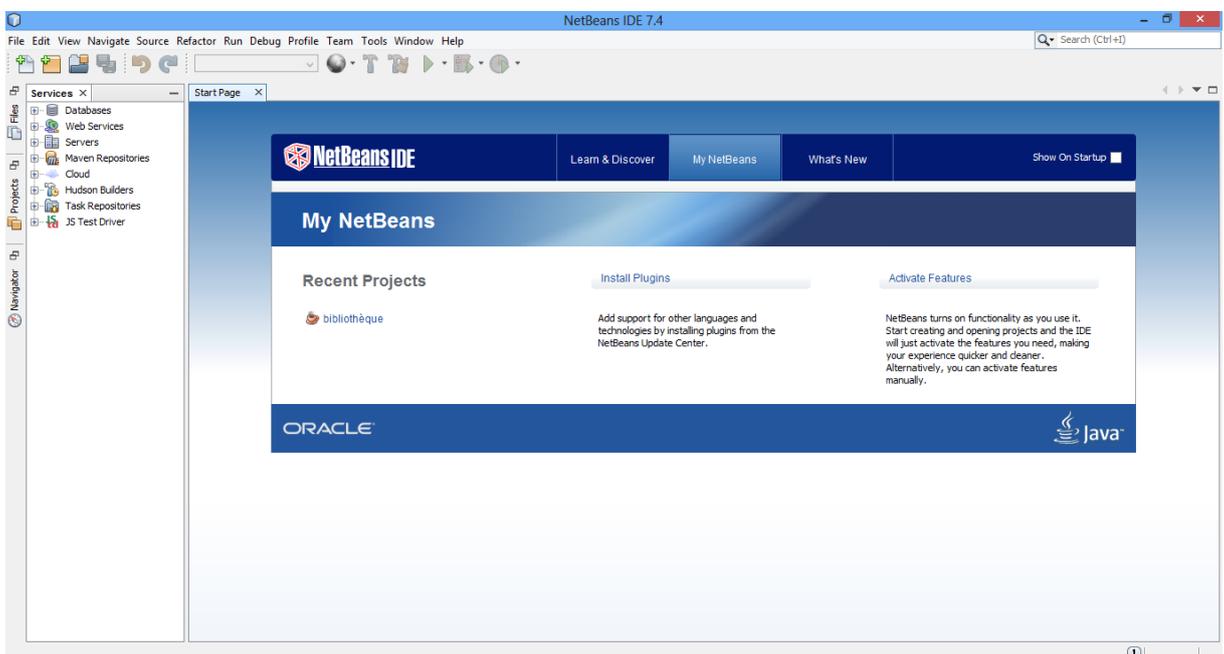


Figure IV 2 : l'interface de l'IDE Netbeans 7.4

4. l'interface JDBC :

JDBC (Java Data Base Connectivity) est une API (Application Programme Interface) Java (ensemble de classes et d'interfaces défini par SUN et les acteurs du domaine des BD) permettant d'accéder aux bases de données à l'aide du langage Java via des requêtes SQL. Cette API permet d'atteindre de manière quasi-transparente des bases Sybase, Oracle, Informix, ... avec le même programme Java JDBC. [Site 18]

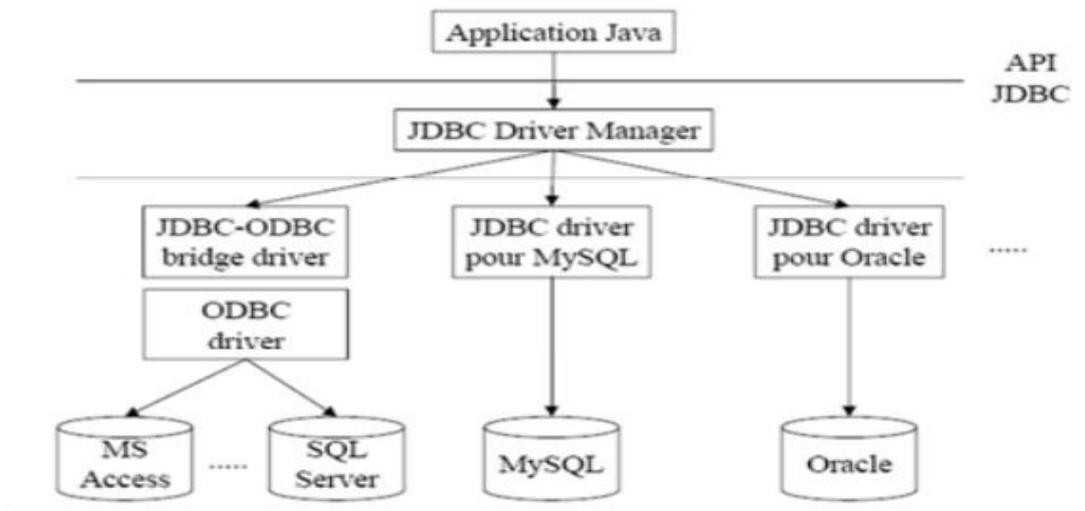


Figure IV 3 : Architecture d’une application Java-JDBC-SGBD [4]

Classe/interface	Description
java.sql.Driver	Pilotes JDBC pour les connexions aux sources de données SQL.
java.sql.Connection	
java.sql.Statement	Construction d’ordres SQL.
java.sql.PreparedStatement	
java.sql.CallableStatement	
java.sql.ResultSet	Gestion des résultats des requêtes SQL.
java.sql.DriverManager	Gestion des pilotes de connexion.
java.sql.SQLException	Gestion des erreurs SQL.
java.sql.DatabaseMetaData	Gestion des méta-informations (description de la base de données, des tables...).
java.sql.ResultSetMetaData	
java.sql.SavePoint	Gestion des transactions et sous-transactions.

Figure IV 4 : JDBC API standard [4]

Classe/interface	Description
oracle.sql.OracleDriver	Connexions aux bases de données (pilotes JDBC OCI et léger).
oracle.sql.OracleConnection	
oracle.sql.OracleStatement	Construction d’ordres SQL.
oracle.sql.OraclePreparedStatement	
oracle.sql.OracleCallableStatement	
oracle.sql.OracleResultSet	Gestion des résultats des requêtes SQL.
oracle.sql.OracleDriverManager	Gestion des pilotes de connexion.
oracle.sql.OracleSQLException	Gestion des erreurs SQL.
oracle.sql.OracleSavePoint	Gestion des transactions et des sous-transactions.

Figure IV 5 : JDBC API d’Oracle [4]

Structure d'un programme :

La structure d'un programme Java utilisant JDBC pour Oracle comprend successivement les phases :

- D'importation de paquets
- De chargement d'un pilote
- De création d'une ou plusieurs connexions
- De création d'un ou de plusieurs états
- D'émission d'instructions SQL sur ces états
- De fermeture des objets créés.

Le code suivant (JDBCTest.java) décrit la syntaxe du plus simple programme JDBC. Nous inscrivons toutes les phases dans un même bloc mais elles peuvent se trouver dans différents blocs ou dans plusieurs méthodes de diverses classes. [4]

Code Java	Commentaires
<pre>import java.sql.*; import oracle.jdbc.driver.*;</pre>	Importation de paquets.
<pre>class JDBCTest { public static void main (String args []) throws SQLException {try{</pre>	Classe ayant une méthode main.
<pre> DriverManager.registerDriver (new oracle.jdbc.driver.OracleDriver()); Connection conn = DriverManager.getConnection ("jdbc:oracle:thin:@CAMPAROLS:1521:BDSoutou", "scott", "tiger");</pre>	Chargement d'un pilote JDBC Oracle et création d'une connexion.
<pre> Statement stmt = conn.createStatement(); ... //Ordres SQL, voir plus loin ...</pre>	Création d'un état de connexion et exécutions d'instructions SQL.
<pre> } catch(SQLException ex){ System.err.println("Erreur : "+ex);} } }</pre>	Gestion des erreurs.

Figure IV 6 : Programme JDBC [4]

5. Création, Modification et suppression :

	Création	Modification	Supprime
Table	CREATE TABLE [schéma.]nomTable (colonne1 type1 [DEFAULT valeur1] [NOT NULL] [, colonne2 type2 [DEFAULT valeur2] [NOT NULL]] [CONSTRAINT nomContrainte1 typeContrainte1]...);	ALTER TABLE [schéma.]nomTable ADD [CONSTRAINT nom Contrainte] type Contrainte;	DROP TABLE nom_table;
tablespaces	CREATE [UNDO] TABLESPACE tablespace_name DATAFILE Datafile_OptionsStorage_Option s;	ALTER TABLESPACE fin RENAME TO new ;	DROP TABLESPACE;
Schéma	CREATE SCHEMA AUTHORIZATION schema_name [create_table_statement] [create_view_statement] [grant_statement];	ALTER SCHEMA schema_name { RENAME TO new_name / OWNER TO new_owner }	DROP SCHEMA mes_affaires CASCADE;
Profile	CREATE PROFILE nomProfil LIMIT {Paramètre Ressource Paramètre Mot de Passe} [Paramètre Ressource Paramètre Mot de Passe]...;	ALTER PROFILE nomProfil LIMIT {Paramètre Ressource Paramètre Mot de Passe} [Paramètre Ressource Paramètre Mot de Passe]...;	DROP PROFILE nom_Profil [CASCADE] ;
Rôle	CREATE ROLE role_name [NOT IDENTIFIED IDENTIFIED {BY password USING [schéma.] package EXTERNALLY GLOBALLY } ;	ALTER ROLE nomRôle [NOT IDENTIFIED IDENTIFIED {BY motdePasse USING [schéma.]paquetage EXTERNALLY GLOBALLY }] ;	DROP ROLE rôle
User	Create username user identified by password;	ALTER USER user_name IDENTIFIED BY new_password;	DROP USER utilisateur [CASCADE];
Databaselink	Create public database link mylink connect to remote_username identified by mypassword using 'tns_service_name';	ALTER SESSION CLOSE DATABASE LINK <link_name>;	DROP DATABASE LINK <link_name>;

Tableau 1 : les requête pour Création et Modification et Suppression

6. Select, modification et supprime de la table :

Select	Modification	Supprime
SELECT * FROM nom_table ;	ALTER TABLE nom_tablewhere nom_colone="nom" ;	DROP TABLE nom_tablewhere nom_colone="nom" ;

Tableau 2 : les requête pour select et modification et suppression de la table

7. L'interface Graphique de Notre application :

L'interface principale de notre application est simple, il s'agit d'un masque de saisie qui comporte des champs à remplir par l'utilisateur ces champs sont le username, le mots de passe, le nom de la base des données, le numéro de port et l'adresse de serveur.



Figure IV 7 : Interface de connexion

Quand l'utilisateur lance la connexion, il obtient une autre Fenêtre comme le montre la figure suivante :



Figure IV 8 : Interface d'application

Les côtes de création dons notre application :

1- création d'Index :

La figure suivante consiste sur la création de l'index, et pour crée cet index on doit attribuer à un nom et un type (Standar , Unique, Bitmap).

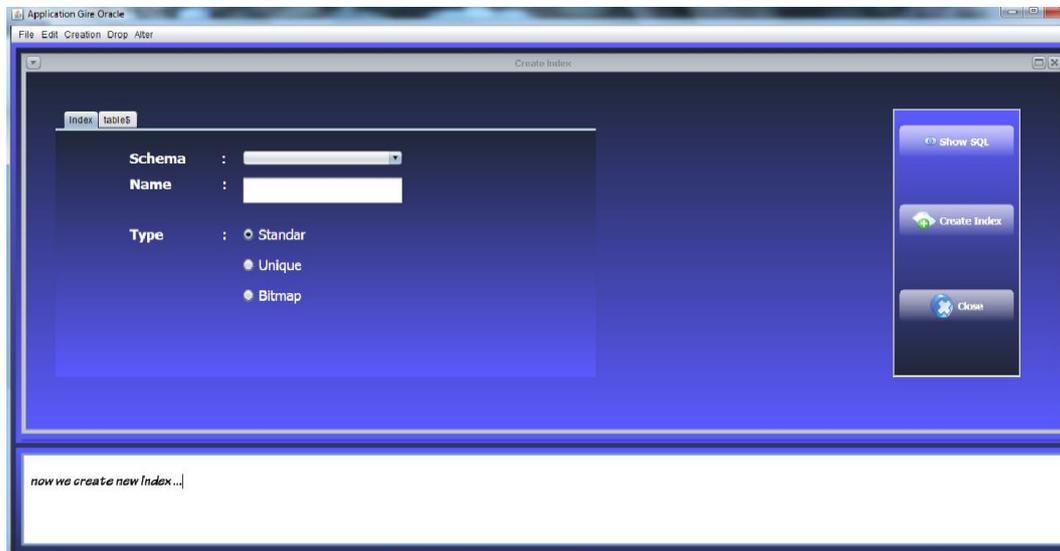


Figure IV 9 : Interface de création de l'index

2- création Table :

La figure si dessous présente la création d'une nouvelle table.

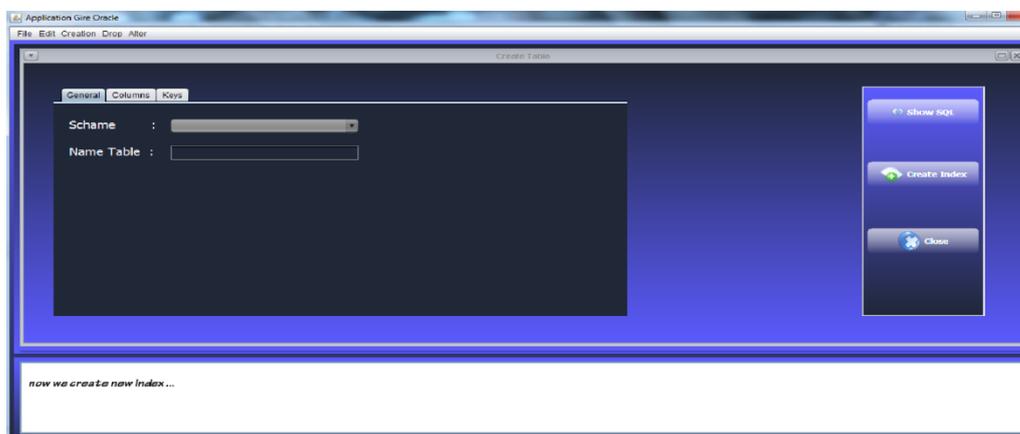


Figure IV 10 : Interface de création de table

3- création Tablespace :

et pour crée les tablespaces il faut utiliser l’instruction create tablespace dans la bar de menus ou la commande (Ctrl+Alt+T).

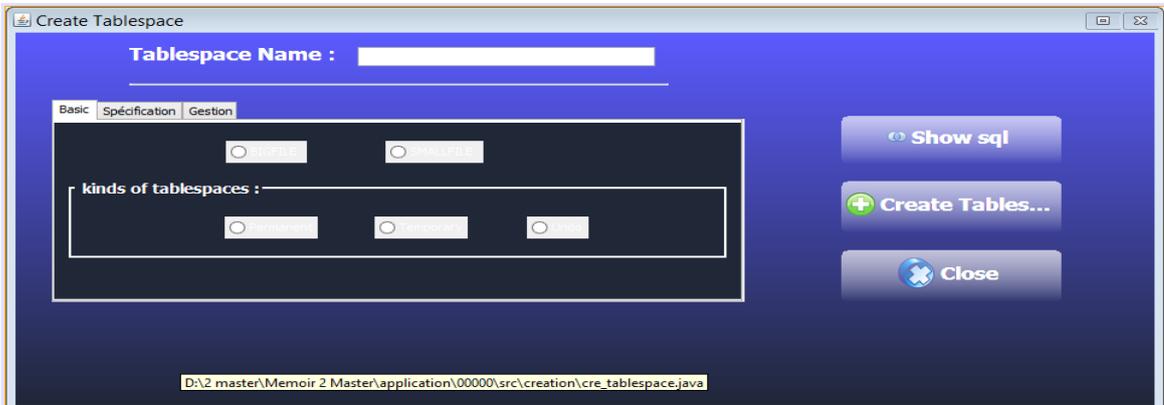


Figure IV 11 : Interface de création de espace de tables

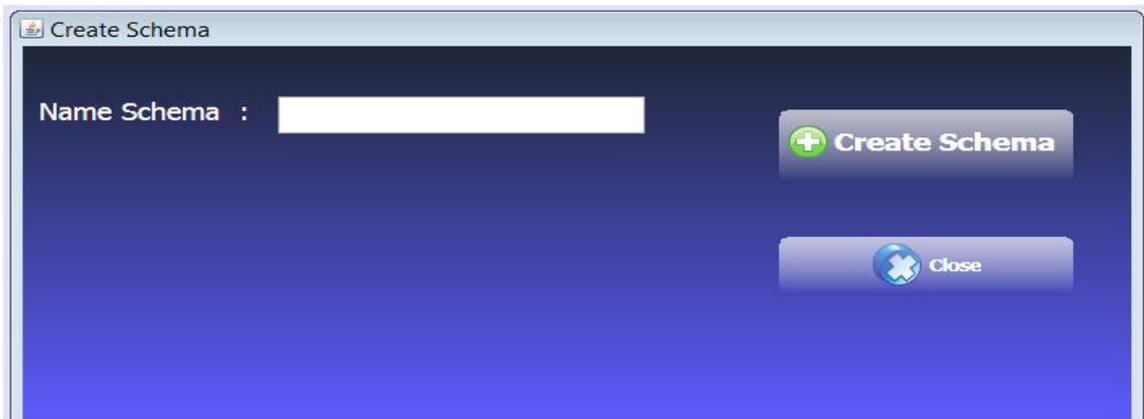


Figure IV 12 : Interface de création de schéma

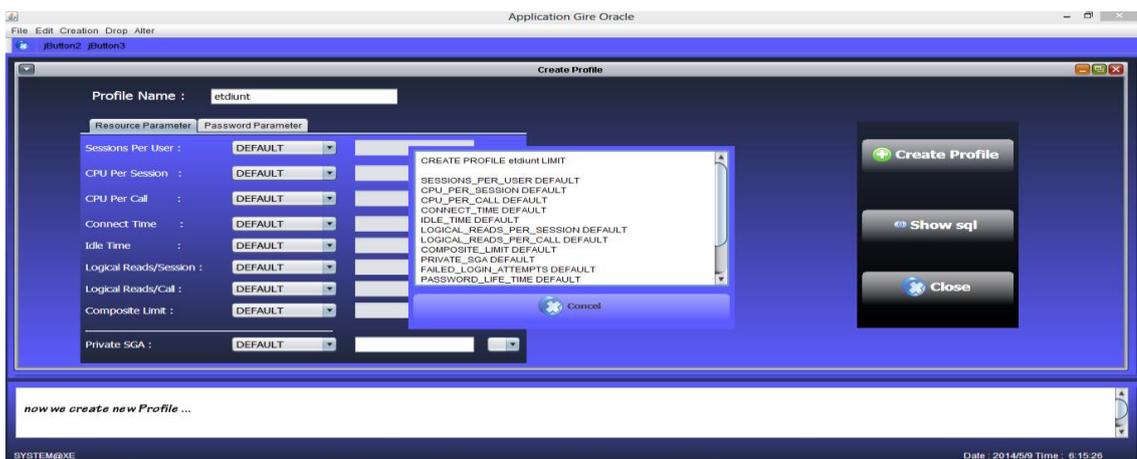


Figure IV 13 : Interface de création de profile

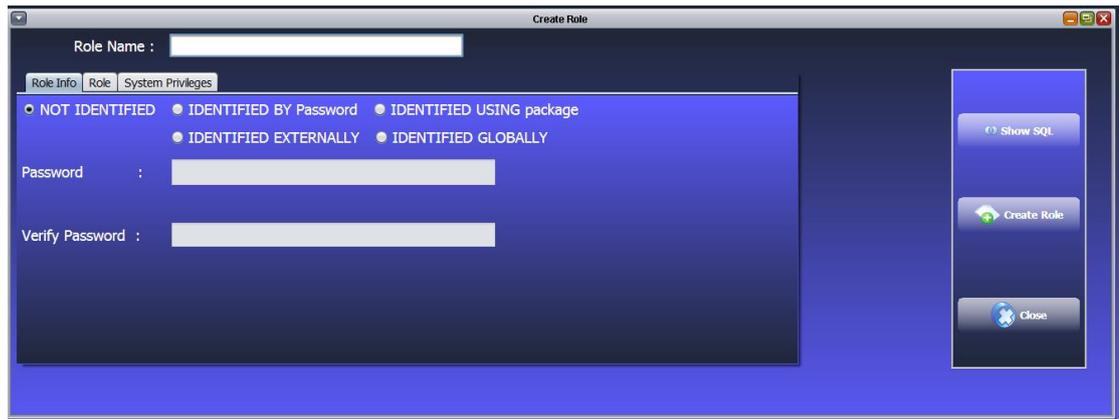


Figure IV 14 : Interface de création de Rôle

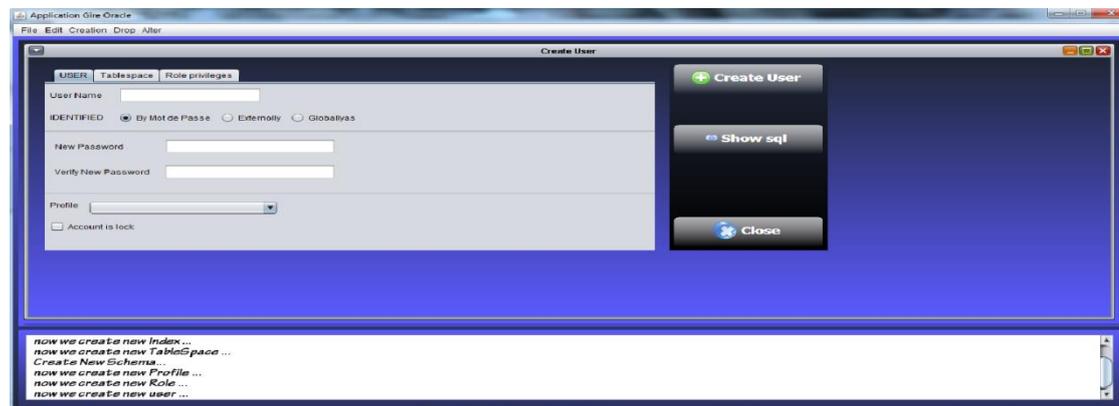


Figure IV 15 : Interface de création d'utilisateur

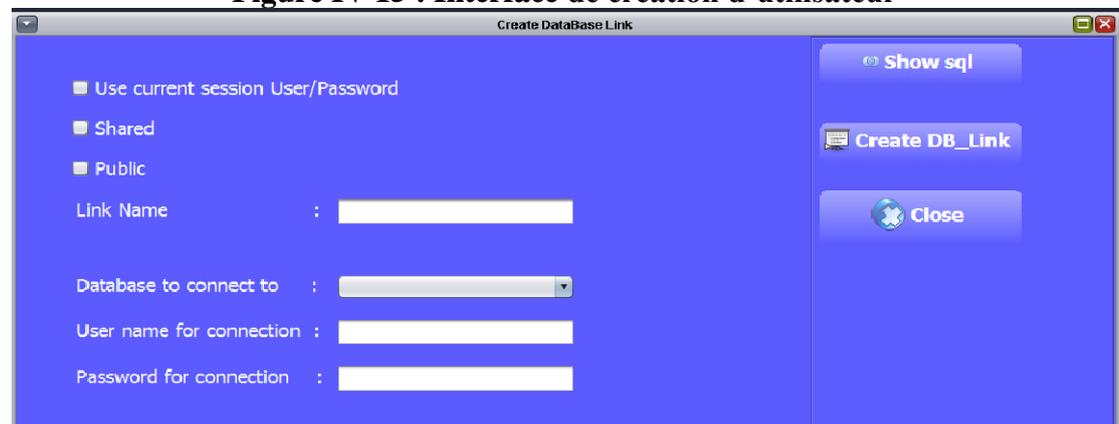


Figure IV 16 : Interface de création de lien de base de données

Figure suivant suppression les tableaux

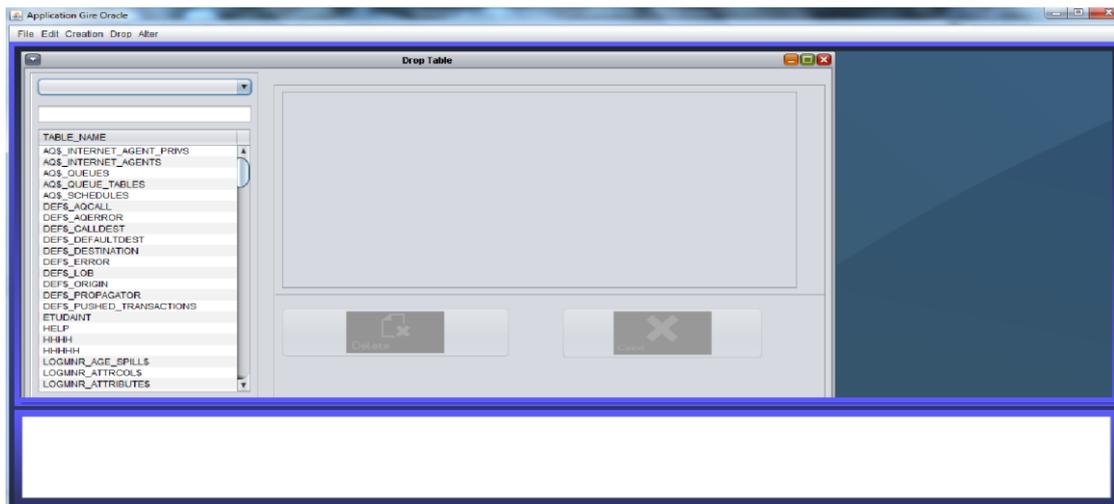


Figure IV 17 : Interface de suppression d'une table

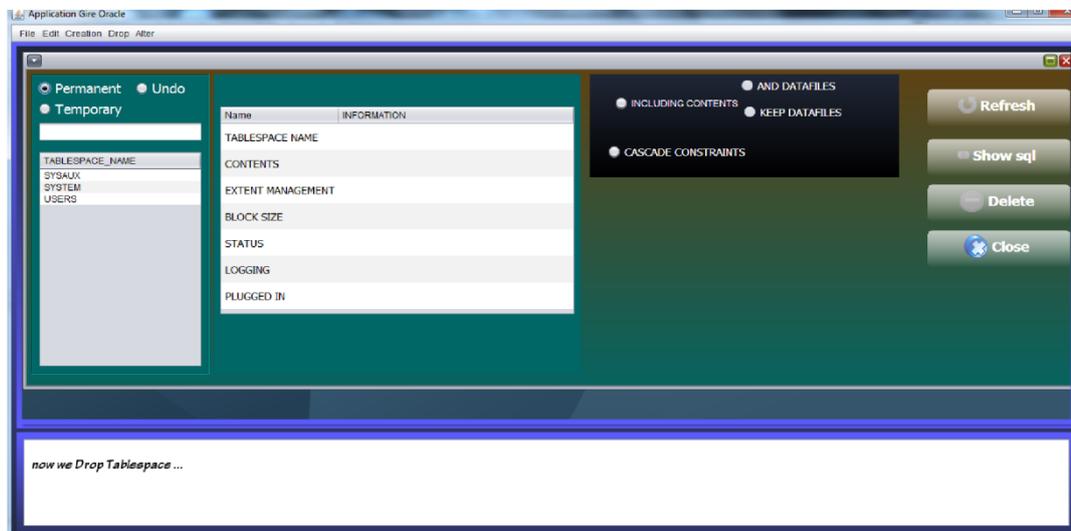


Figure IV 18 : Interface de suppression d'une espace table



Figure IV 19 : Interface de suppression d'une Profile



Figure IV 20 : Interface de suppression d'une Rôle

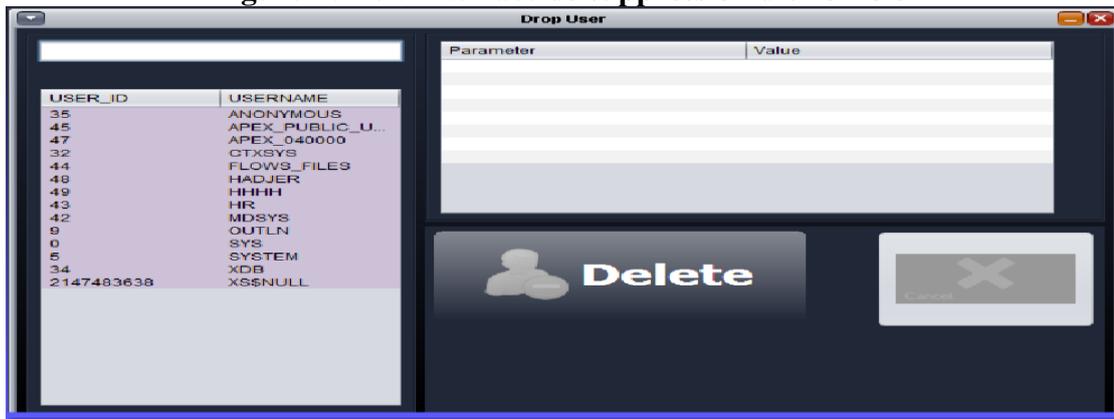


Figure IV 21 : Interface de suppression d'un Utilisateur

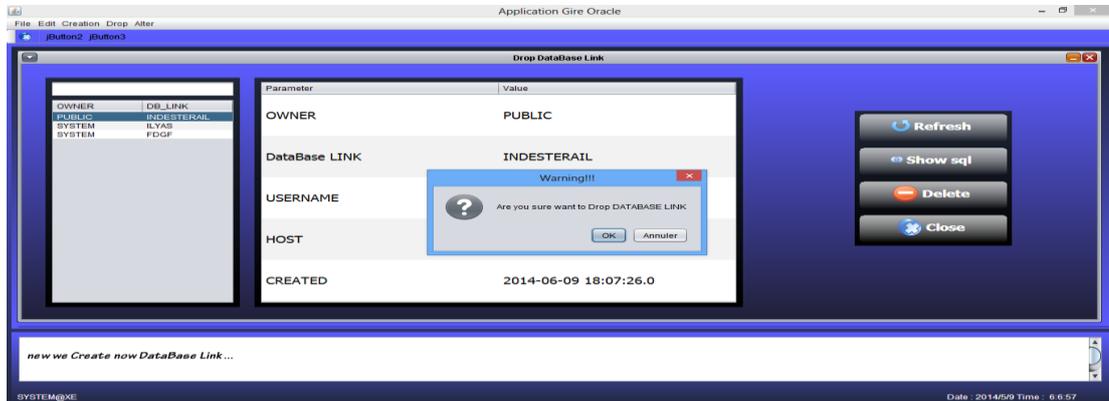


Figure IV 22 : Interface de suppression d'un lien de base de données



Figure IV 23 : Interface de modification d'un lien de base de données

8. Conclusion:

Dans ce chapitre, nous avons présenté les outils nécessaires pour développer notre application ainsi que l'application elle-même, et pour faciliter l'utilisation de notre outil d'administration d'ORACLE (oracle pole) nous avons expliqué en détails toutes ces fonctionnalités.

Conclusion

Générale

Conclusion général

A la fin de mémoire, nous avons étudié les différences entre les systèmes SGBD lorsque nous définissons SGBD, SGBDR, caractéristique (Indépendance physique, Manipulabilité, Rapidité des accès, Administration centralisée, Limitation de la redondance, Vérification de l'intégrité, Partageable des données, Sécurité des données) et entrée définition SGBD «Oracle, SQL Server,etc. »

Identifié Oracle, historique, les architectures Oracle « Interne est fonctionnelle », outils administration « Toad, SQL Developer Oracle,.. » et étude les caractéristiques de l'Oracle.

Et dernier partie à étudié, c'est le langage Java Historique et Introduction java et définition NetBeans et comme connecte avec base de données Oracle.

Et la dernière étape consiste sur notre outil. Nous avons obtenu la création, la modification et la suppression (tablespace, index, table,). Notre application consiste sur les éléments suivants:

- L'efficacité de l'interface.
- Affichage des requêtes.
- La facilité de création. exp : Table, User...etc.
- La rapidité de connexion entre l'interface et l'Oracle.

A la fin, on peut dire que cette application aura plus de chance dans le temps qui va venir sur la base de données Oracle.

BIBLIOGRAPHIE

- [1] Administration vos Bases de données avec MYSQL
- [2] BASE de DONNEES SYSTEME de GESTION de BASE de DONNEES MySQL / SQL
- [3] les systèmes de gestion de bases de données (SGBD)
- [4] SQL pour Oracle 3^e édition, Christian Soutou, 2008
- [5] Oracle : administration (cours), Olivier Guibert, Département Informatique de l'Institut Universitaire de Technologie de l'Université Bordeaux 1
- [6] Oracle 11g Exploitation , Clotilde Attouche , 06/05/2010
- [7] Oracle 11 g administration , Olivier HEURTEL , Copyring – Editions ENI – 12/2008
- [8] programmer Java , © Groupe Eyrolles, 2002
- [9] JAVA programmation objet, Patrick Ducrot / <http://www.ducrot.org/java/PolycopieJAVA.pdf>
- [10] Les concepts de Java , B. Tatibouët

Les sites web :

- [Site 1] <http://www.commentcamarche.net/contents/702-oracle-introduction-au-sgbd-oracle> /Avril 2014/ 25 Avril 2014/
- [Site 2] <http://www.techno-science.net/?onglet=glossaire&definition=7708> /25 Avril 2014/
- [Site 3] <http://fr.wikipedia.org/wiki/MySQL#Historique> 28 Avril 2014
- [Site 4] http://www.ouestdecision.fr/contenu/les_editeurs/fiches/fiche-produit-sqlserver.htm 2011/29 Avril 2014
- [Site 5] http://fr.wikipedia.org/wiki/Microsoft_Access / 29 Avril 2014
- [Site 6] <http://forum.cultureco.com/bts-ag-e6-projet-de-developpement-de-la-pme/18648-avantages-et-inconvenients-daccess.html> 29 Avril
- [Site 7] <http://www.ybet.be/access/> 29 Avril
- [Site 8] http://www.memoireonline.com/04/10/3347/m_Mise-en-place-dun-systeme-dinformation1.html 30 Avril
- [Site 9] http://www.didier.deglise.free.fr/dba/introduction/intro_main.htm 30 Avril

Référence

[Site10] http://fr.wikipedia.org/wiki/Oracle_SQL_Developer 30 Avril

[Site 11] http://fr.wikipedia.org/wiki/Toad_%28logiciel%29 30 Avril

[Site 12] www.quest.com/quest_site_assets/pdf/dsd-toad4oracle-f-fr.pdf 30 Avril

[Site 13] www.quest.com/quest_site_assets/pdf/toad_for_mysql_mai_2007.pdf 30 Avril

[Site 14] <http://www.sqlmanager.net/fr/products/oracle/manager> 30 Avril

[Site 15] www.eyrolles.com/Chapitres/9782212117479/module6_1_Bizoi.pdf /30 avril

[Site 16] <http://tecfa.unige.ch/guides/tie/pdf/files/java-intro.pdf> 2 juin

[Site 17] <http://doc.ubuntu-fr.org/netbeans> 02/juin

[Site18] <http://cedric.cnam.fr/~farinone/IAGL/JDBC.pdf> 2 juin Adresse