

UNIVERSITE MOHAMED BOUDIAF - M'SILA

**FACULTE DES MATHÉMATIQUES ET
DE L'INFORMATIQUE**

DEPARTEMENT D'INFORMATIQUE



MEMOIRE de fin d'étude

Présenté pour l'obtention du diplôme de MASTER

Domaine : Mathématiques et Informatique

Filière : Informatique

Spécialité : Informatique Décisionnel et Optimisation

Par : DAKHOUCHE AHMED

SUJET

Optimisation des Bases de Données

Cas : SGBD Oracle

Soutenu publiquement le : 13 / 07 / 2019 devant le jury composé de :

M. AMRI Said

Université de M'sila

Président

M. KHETTAF Abdelouahab

Université de M'sila

Rapporteur

M. BOUCETTA Mohamed

Université de M'sila

Examineur

Promotion : 2018 /2019



Dédicace

*Toutes les lettres ne sauraient trouver les mots qu'il faut
Tous les mots ne sauraient exprimer
* La gratitude * L'amour * Le respect **

Je dédie cette mémoire... ?

A mes chers parents

*Pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et
leurs prières tout au long de mes études,*

A mes chères sœurs

Pour leurs encouragements permanents, et leur soutien moral,

A mes chers frères

Pour leur appui et leur encouragement,

A toute ma famille

Pour leur soutien tout au long de mon parcours universitaire,

*Que ce travail soit l'accomplissement de vos vœux tant allégués, et le
fruit de votre soutien infaillible,
Merci d'être toujours là pour moi.*

AHMED



Remerciements

Pour commencer je veux adresser mes remerciements à mon directeur de mémoire, Khettaf. A, pour sa grande disponibilité et ses encouragements tout au long de la réalisation et la rédaction de ce mémoire.

Je remercie également tous les professeurs d'université de Mohamed Boudiaf

J'adresse mes plus sincères remerciements à ma famille : Mes parents, mes sœurs et mes frères, tous mes proches et amis, qui m'ont accompagné, aidé, soutenu et encouragé tout au long de la réalisation de ce mémoire.

Enfin, Je voudrais remercier aussi toutes les personnes qui ont participé de près ou de loin à mes recherches et à l'élaboration de ce mémoire.

TABLE DES MATIERES

INTRODUCTION GENERALE	- 1 -
CHAPITRE 01 : ETAT DE L'ART	
1. Introduction.....	- 4 -
2. Base de données.....	- 4 -
2.1. C'est quoi une base de données ?.....	- 4 -
2.2. Pour quoi Une Base de données ?.....	- 4 -
3. Système de gestion de base de données	- 5 -
4. Le modèle relationnel	- 7 -
5. SQL (Structured Query Language).....	- 7 -
6. SGBD Oracle.....	- 8 -
6.1. Les fonctionnalités d'Oracle	- 8 -
6.2. Les composants d'Oracle	- 8 -
6.3. Architecture d'un serveur Oracle.....	- 9 -
6.3.1. Structure d'une instance Oracle.....	- 10 -
6.3.1.1. Structures Mémoire Oracle.....	- 10 -
6.3.1.2. Processus Oracle.....	- 11 -
6.3.2. Structure de la base de données Oracle.....	- 12 -
6.3.2.1. Structure physique.....	- 12 -
6.3.2.2. Structure logique.....	- 13 -
7. Conclusion.....	- 14 -
CHAPITRE 02 : OPTIMISATION DES BASES DE DONNEES SOUS ORACLE	
1. Introduction.....	- 16 -
2. Présentation de l'optimisation.....	- 16 -
2.1. Optimisation au niveau : modèle de données.....	- 16 -
2.1.1. Présentation du modèle relationnel	- 16 -
2.1.2. Normalisation.....	- 17 -
2.1.2.1. Dépendance fonctionnelle :	- 18 -
2.1.2.2. La première forme normale (1NF).....	- 19 -
2.1.2.3. La Deuxième forme normale (2NF)	- 19 -
2.1.2.4. La troisième forme normale (3NF).....	- 19 -

2.1.2.5.	La forme normale de Boyce Codd (BCNF)	- 19 -
2.1.2.6.	Autres formes de normalisation	- 19 -
2.1.2.7.	Comment normaliser une table	- 20 -
2.1.3.	Dénormalisation	- 21 -
2.1.3.1.	Quand utiliser la dénormalisation ?	- 22 -
2.1.3.2.	Le partitionnement de table.....	- 22 -
2.2.	Optimisations au niveau : requêtes.....	- 23 -
2.2.1.	Statistiques sur les données	- 23 -
2.2.1.1.	Optimiseur basé sur le coût CBO	- 24 -
2.2.1.2.	Comment estimer le coût d'un plan d'exécution ?.....	- 24 -
2.2.2.	Indexes	- 24 -
2.2.2.1.	B-Tree	- 24 -
2.2.2.2.	Bitmap.....	- 25 -
2.2.3.	Réécriture des requêtes	- 26 -
2.2.4.	Hints.....	- 26 -
2.2.4.1.	Hints des approches d'optimisation pour l'objectif.....	- 26 -
2.2.4.2.	Hints de chemin d'accès.....	- 26 -
2.2.4.3.	Hints de transformation de requête	- 26 -
2.2.4.4.	Les hints de jointure.....	- 27 -
2.2.4.5.	Les hints de parallélisme	- 27 -
2.2.5.	SQL avancé.....	- 27 -
2.3.	Optimisations au niveau : infrastructure.....	- 27 -
2.3.1.	Optimisation de l'exécution du SGBDR.....	- 27 -
2.3.1.1.	Ajustement de la mémoire utilisable	- 28 -
2.3.1.2.	Répartition des fichiers	- 28 -
2.3.2.	Optimisation matérielle.....	- 28 -
3.	Conclusion.....	- 28 -

CHAPITRE 03 : EXPERIMENTATION ET ANALYSE

1.	Introduction	- 31 -
2.	La base de données de test	- 31 -
3.	Environnement du travail	- 33 -
4.	Eléments d'analyse et diagnostic	- 34 -

4.1.	Les critères de mesure	- 34 -
4.2.	Les outils d'optimisation	- 34 -
4.2.1.	Le plan d'exécution.....	- 34 -
4.2.2.	Analyse les SQL Trace avec TKPROF.....	- 35 -
4.2.3.	Moniteur de ressources Windows.....	- 36 -
4.2.4.	Analyseur de performance de la base données : SolarWinds	- 38 -
5.	Résultats expérimentaux	- 39 -
5.1.	Résultats des différentes structures	- 39 -
5.1.1.	Requêtes à exécuter.....	- 40 -
5.1.2.	Présentation des résultats obtenus	- 40 -
5.1.3.	Fragmentation Horizontale	- 41 -
	Présentation des résultats obtenus	- 41 -
5.1.4.	Fragmentation Verticale	- 42 -
	Présentation des résultats obtenus	- 42 -
5.1.5.	Analyse des résultats et Diagnostic.....	- 42 -
5.2.	Résultats des différentes requêtes	- 43 -
5.2.1.	Statistiques sur les données	- 43 -
5.2.2.	Indexes & Hints.....	- 44 -
5.2.3.	Réécriture des requêtes	- 45 -
5.2.4.	Analyse des résultats et Diagnostic.....	- 45 -
5.3.	Résultats des différentes infrastructures matérielles	- 46 -
5.3.1.	Requête de test.....	- 46 -
5.3.2.	Présentation des résultats obtenus	- 46 -
5.3.3.	Analyse et diagnostique des résultats	- 49 -
6.	Conclusion.....	- 50 -
	CONCLUSION GENERALE	- 51 -
	BIBLIOGRAPHIE	- 52 -
	Annexe A1 : Installation du SGBD ORACLE	- 55 -
	Annexe A2 : Création de la base de données et les tables	- 60 -
	Annexe A3 : Requêtes utilisés dans la partie d'expérimentation	- 65 -

LISTE DES FIGURES

CHAPITRE 01 : ETAT DE L'ART

Figure 1. 1 Niveaux de représentation des données	- 5 -
Figure 1. 2 Modèles de base de données	- 6 -
Figure 1. 3 Exemple de modèle relationnel	- 7 -
Figure 1. 4 Architecture d'un serveur Oracle	- 9 -
Figure 1. 5 Structures Mémoire Oracle	- 11 -
Figure 1. 6 Principe de stockage - Oracle.....	- 13 -

CHAPITRE 02 : OPTIMISATION DES BASES DE DONNEES SOUS ORACLE

Figure 2. 1 Processus de normalisation	- 18 -
Figure 2. 2 Exemple de La dénormalisation.....	- 22 -
Figure 2. 3 Exemple d'un index B*Tree	- 25 -
Figure 2. 4 Exemple d'un Index bitmap.....	- 25 -

CHAPITRE 03 : EXPERIMENTATION ET ANALYSE

Figure 3. 1 Le MPD de la deuxième base de test	- 32 -
Figure 3. 2 Le MPD de la première structure	- 33 -
Figure 3. 3 Le MPD de la deuxième structure.....	- 33 -
Figure 3. 4 Plan d'exécution avec SQL*Plus.....	- 35 -
Figure 3. 5 Création du fichier de trace	- 35 -
Figure 3. 6 Rapport généré par le TKPROF	- 36 -
Figure 3. 7 Moniteur de ressources (CPU)	- 37 -
Figure 3. 8 Moniteur de ressources (RAM).....	- 37 -
Figure 3. 9 Moniteur de ressources (E/S Disque).....	- 38 -
Figure 3. 10 Diagrammes d'utilisation CPU par SolarWinds	- 38 -
Figure 3. 11 Diagrammes d'utilisation Mémoire d'instance	- 39 -
Figure 3. 12 Diagrammes d'utilisation (E/S) Disque	- 39 -
Figure 3. 13 Résultats du Requête 1 sur les deux Structures.....	- 40 -
Figure 3. 14 Résultats du Requête 2 sur les deux Structures.....	- 41 -
Figure 3. 15 Résultats obtenus par la fragmentation horizontale	- 41 -
Figure 3. 16 Résultats obtenus par la fragmentation verticale.....	- 42 -
Figure 3. 17 Résultats (Avec/Sans) l'utilisation des Statistiques	- 44 -
Figure 3. 18 Résultats de l'utilisation des indexes & Hints.....	- 44 -
Figure 3. 19 Résultats obtenus par Réécriture des requêtes	- 45 -
Figure 3. 20 Résultats obtenus par les différentes SE	- 46 -
Figure 3. 21 Résultats obtenus par les différentes capacités de RAM.....	- 47 -
Figure 3. 22 Code java de la création du script SQL.....	- 48 -
Figure 3. 23 Résultats obtenus par CPU - Intel i3 @ 4 Core -	- 48 -
Figure 3. 24 Résultats obtenus par CPU - Intel i3 @ 1 Core -	- 49 -

ANNEXE A1 : INSTALLATION DU SGBD ORACLE

Figure A1. 1 Installation de la base de données- Etape 1	- 55 -
Figure A1. 2 Installation de la base de données- Etape 2	- 56 -
Figure A1. 3 Installation de la base de données- Etape 3	- 56 -
Figure A1. 4 Installation de la base de données- Etape 4	- 57 -
Figure A1. 5 Installation de la base de données- Etape 5	- 57 -
Figure A1. 6 Installation de la base de données- Etape 6	- 58 -
Figure A1. 7 Installation de la base de données- Etape 7	- 58 -
Figure A1. 8 Installation de la base de données- Etape 8	- 59 -

ANNEXE A2 : CREATION DE LA BASE DE DONNEES ET LES TABLES

Figure A2. 1 Accueil de l'Assistant Configuration de base de données.....	- 60 -
Figure A2. 2 Les options de l'Assistant Configuration de base de données	- 61 -
Figure A2. 3 Fin de la création de base de données	- 61 -
Figure A2. 4 connexion à la base de données avec SQL PLUS	- 62 -
Figure A2. 5 Script de la base BIGEMP	- 62 -
Figure A2. 6 Script de la base BIGEMP1	- 63 -
Figure A2. 7 Script de la base BigCommande	- 63 -

LISTE DES TABLEAUX

CHAPITRE 02 : OPTIMISATION DES BASES DE DONNEES SOUS ORACLE

<i>Tableau 2. 1 Table non normalisée</i>	<i>- 20 -</i>
<i>Tableau 2. 2 Table en première forme normale.....</i>	<i>- 20 -</i>
<i>Tableau 2. 3 Table en deuxième forme normale</i>	<i>- 21 -</i>
<i>Tableau 2. 4 Table en deuxième forme normale</i>	<i>- 21 -</i>

CHAPITRE 03 : EXPERIMENTATION ET ANALYSE

<i>Tableau 3. 1 Taille de chaque table de la base de test BIGEMP.....</i>	<i>- 32 -</i>
<i>Tableau 3. 2 Taille de chaque table de la deuxième base de test</i>	<i>- 32 -</i>
<i>Tableau 3. 3 Résultats obtenus par les structure</i>	<i>- 40 -</i>
<i>Tableau 3. 4 Résultats obtenus par la fragmentation horizontale</i>	<i>- 41 -</i>
<i>Tableau 3. 5 Résultats obtenus par la fragmentation verticale</i>	<i>- 42 -</i>
<i>Tableau 3. 6 Résultats obtenus par l'utilisation des statistiques</i>	<i>- 43 -</i>
<i>Tableau 3. 7 Résultats obtenus par l'utilisation des indexes & Hints</i>	<i>- 44 -</i>
<i>Tableau 3. 8 Résultats obtenus par Réécriture des requêtes</i>	<i>- 45 -</i>
<i>Tableau 3. 9 Résultats obtenus par les différentes SE</i>	<i>- 46 -</i>
<i>Tableau 3. 10 Résultats obtenus par les différentes capacités de RAM.....</i>	<i>- 47 -</i>
<i>Tableau 3. 11 Résultats obtenus par des différentes infrastructures matérielles.....</i>	<i>- 48 -</i>

LISTE DES ABREVIATIONS

SGBD : Système de Gestion de Base de Données.

SQL : Structured Query Language.

SGBDR : Système de Gestion de Base de Données Relationnelles.

DBMS : DataBase Management System.

MPD : Modèle Physique de Données.

LDD : Langage de Définition de Données.

LMD : Langage de Manipulation de Monnées.

DCL : Data Control Language.

SGA : System Global Area.

PGA : Program Global Area.

JVM : Java Virtual Machine.

UPI : User Programme Interface.

BDD : Base De Données.

CBO : Cost Based Optimizer.

RAM : Random Access Memory.

CPU : Central Processing Unit.

E/S : Entrées/Sorties Disque.

DBA : Data Base Administrator.

MLD : Modèle Logique de Données

INTRODUCTION GENERALE

Les systèmes de gestion de bases de données, qui permettent de stocker des données de grande masse, les administrer et les sécuriser, sont ainsi devenues omniprésentes en entreprise, avec des usages de plus en plus variés, parce que les données font partie des éléments vitaux de l'entreprise.

Les données sont en augmentation constante ce qui poussent les entreprises à chercher des solutions pour sa base de données afin de gérer ce volume de données.

La question qui s'impose est « Quels sont les solutions permettant de traiter ces énormes quantités de données avec moindre coût et avec les meilleures performances possibles ? »

L'optimisation des bases de données est un sujet très large, qui permet de résoudre les problèmes de performance de la base de données, Ceci est fait par des dizaines de mécanismes soit avant la conception de la base de données ou bien après, c'est à dire disponible sur le Système de Gestion des Base de Données.

Afin de mieux comprendre l'optimisation de la base de données sous oracle, il est important d'identifier les aspects d'amélioration qui sont divisés en trois axes principaux :

- Optimisations au niveau : modèle de données.
- Optimisations au niveau : requêtes.
- Optimisations au niveau : infrastructure.

Dans chacun de ces axes, il y a des nombreux mécanismes qui permettent de mieux gérer la base de données, et minimiser les effets indésirables influençant négativement sur les performances de cette base.

Dans ce mémoire nous allons présenter les principaux mécanismes d'optimisation disponibles sur les trois axes, après ça nous allons faire des tests sur une base de données en utilisant ces mécanismes et le contraire pour déduire la différence entre les deux cas et pour mettre en évidence l'importance d'utiliser les mécanismes d'optimisation.

On a divisé notre mémoire en trois (03) chapitres.

Le premier chapitre nous donnons un état de l'art sur les bases de données, le modèle relationnel et le Système de Gestion des Base de Données le plus populaire et le plus performant du marché ORACLE.

Le deuxième chapitre présente les mécanismes d'optimisation de base des données sous oracle, qui permettent de garantir un bon temps de réponse et une réduction du coût de traitement et d'obtenir les meilleures performances.

Dans le dernier chapitre nous allons valider l'importance d'utiliser ces mécanismes d'optimisation sous Oracle et vérifier et analyser son impact avec des tests fait sur une base de données.

En fin, notre mémoire s'achève par une conclusion générale et quelques perspectives.

CHAPITRE 01
ETAT DE L'ART

1. Introduction

Aujourd’hui, les entreprises doivent gérer d’énormes volumes de données et cela en garantissant l’intégrité, la sécurité et la précision des données, Oracle est le meilleur choix puisqu’actuellement, c’est le SGBD le plus populaire et le plus performant du marché [22]. C’est un moteur de base de données très solide qui permet de gérer de grand volume de données.

Pour mieux appréhender une base de données et la maîtriser, il est prépondérant d’avoir une idée sur les éléments nécessaires à son fonctionnement.

Ce chapitre est d’abord une occasion d’étudier et de repérer quelques notions élémentaires sur les base de données et le SGBD oracle qui seront utiles tout au long de ce mémoire.

Dans notre étude on utilisera Oracle 11g Professionnel, Car il offre un environnement de travail plus vaste.

2. Base de données

2.1. C’est quoi une base de données ?

Les bases de données sont apparues à la fin des années 60, à une époque où la nécessité d’un système de gestion de l’information souple se faisait ressentir.

On appelle base de données une collection de grandes quantités d’informations stockées dans des fichiers et accessibles à la demande pour plusieurs utilisateurs et des besoins divers, afin d’en faciliter l’exploitation (ajout, mise à jour, recherche de données). [15]

Ces données représentent des informations servant aux activités et au management d’une entreprise.

2.2. Pour quoi Une Base de données ?

Les bases de données ont pris aujourd’hui une place essentielle dans l’informatique, plus particulièrement en gestion. Une base de données permet de mettre des données à la disposition d’utilisateurs pour une consultation, une saisie ou bien une mise à jour, tout en s’assurant des droits accordés à ces derniers. Cela est d’autant plus utile que les données informatiques sont de plus en plus nombreuses.

L’avantage majeur de l’utilisation de bases de données est la possibilité de pouvoir être accédées par plusieurs utilisateurs simultanément. [12] [7]

Pour gérer et contrôler la base de données, un système de gestion de base de données (SGBD) reste obligatoire et nécessaire.

3. Système de gestion de base de données

Le SGBD (Système de Gestion de Bases de Données) ou en anglais DBMS (Data Base Management System) est un système intégré pour la gestion unifiée des bases de données [2] une autre définition, SGBD est un logiciel intermédiaire entre l'utilisateur et la base et qui est chargé de réaliser toutes les opérations s'effectuant sur la base de données. [15]

Il existe sur le marché plusieurs SGBD parmi lesquels on peut citer : My SQL server, Microsoft SQL Server, Sybase, Oracle ...etc.

Des objectifs principaux ont été fixés aux SGBD sont les suivants :

- Indépendance physique et logique.
- Accès efficace aux données.
- Sécurité des données.
- Partage des données.
- Administration centralisée des données.
- Résistance aux pannes.
- Manipulation des données.

Un objectif majeur des SGBD est d'assurer une abstraction des données stockées sur disques pour simplifier la vision des utilisateurs. Pour cela, trois niveaux de description de données ont été distingués par le groupe ANSI/SPARC sont :

- **Le niveau externe** : il définit les vues des utilisateurs.
- **Le niveau conceptuel** : Il définit l'arrangement des informations au sein de la base de données.
- **Le niveau interne ou physique** : il définit la façon selon laquelle sont stockées les données et les méthodes pour y accéder

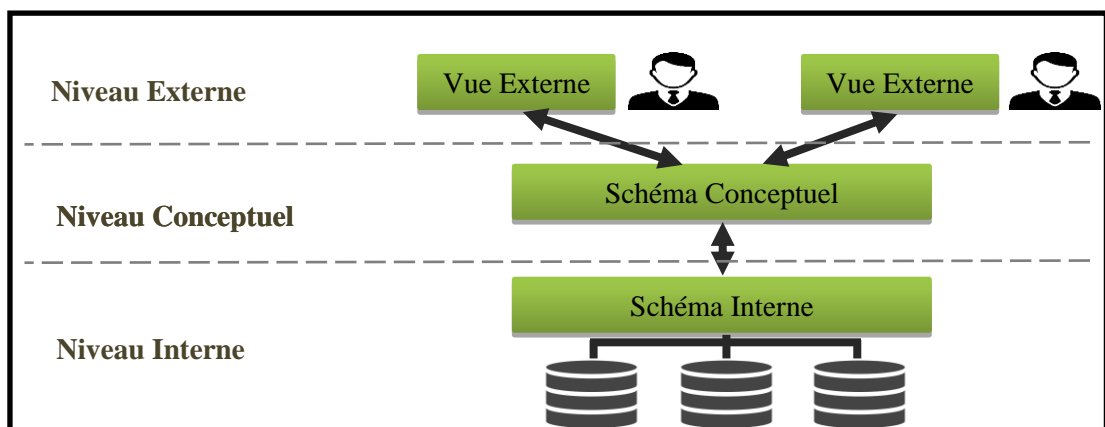


Figure 1. 1 Niveaux de représentation des données

Le SGBD doit être capable de faire des transformations entre chaque niveau, de manière à transformer une requête exprimée au niveau externe en requête du niveau conceptuel, enfin en requête de niveau physique. [20]

Il existe différents modèles de base de données, différenciés selon la représentation des données qu'elle contient (voir figure 1.2) :

- **Le modèle hiérarchique** : les données sont classées hiérarchiquement, selon une arborescence descendante. Ce modèle utilise des pointeurs entre les différents enregistrements. Il s'agit du premier modèle de SGBD.
- **Le modèle réseau** : comme le modèle hiérarchique, ce modèle utilise des pointeurs vers des enregistrements. Toutefois la structure n'est plus forcément arborescente dans le sens descendant.
- **Le modèle relationnel** : les données sont enregistrées dans des tableaux à deux dimensions. La manipulation de ces données se fait selon la théorie mathématique des relations.
- **Le modèle objet** : les données sont stockées sous forme d'objets, c'est-à-dire de structures appelées *classes* présentant des données membres. Les champs sont des instances de ces classes. [7]

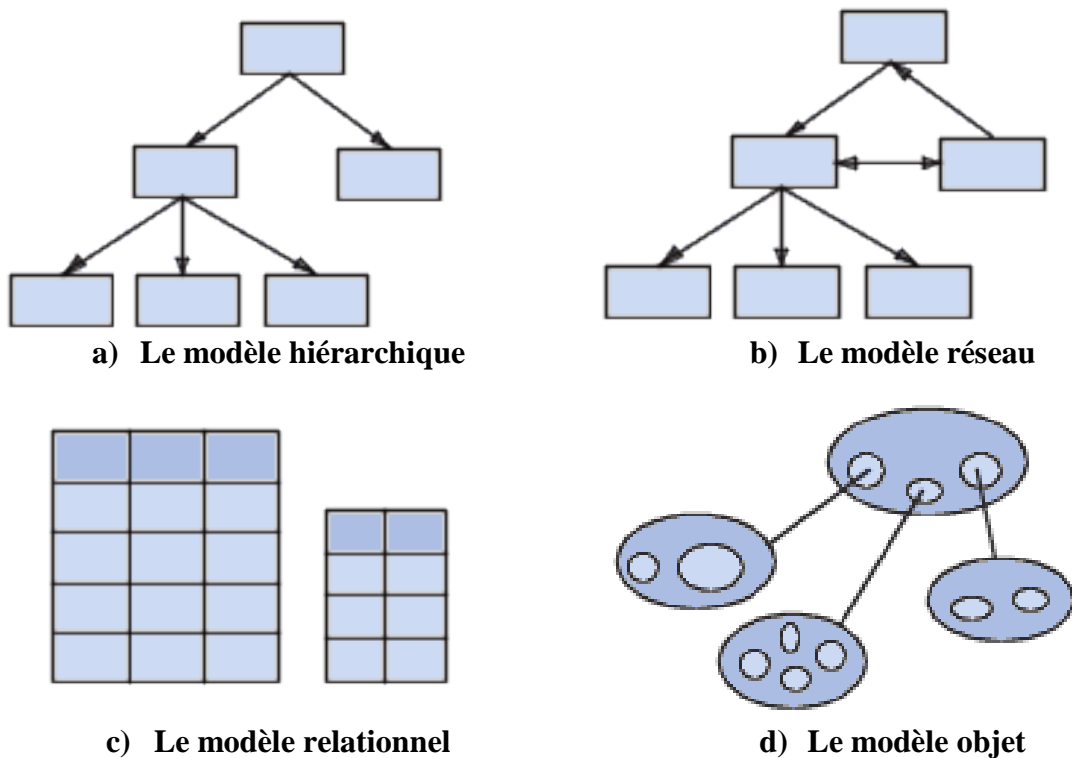


Figure 1. 2 Modèles de base de données [7]

Parmi ces modèles de base de données, il nous a intéressé le modèle relationnel, parce que la représentation des données dans ce modèle s’applique à la base de données de notre étude.

4. Le modèle relationnel

Le modèle relationnel est basé sur une organisation des données sous forme de tables. La manipulation des données se fait selon le concept mathématique des relations de la théorie des ensembles, c'est-à-dire l'algèbre relationnelle.

La modélisation relationnelle permet de représenter les relations à l'aide des tables constituées de lignes et de colonnes (deux dimensions) dont chaque colonne a un identificateur qui représente un domaine. Une ligne du tableau représente donc une entité et chacune des cases représente un de ses attributs. [7]

Le modèle relationnel permet de représenter les données que l'on va gérer de manière simple et souple. Il facilite le traitement et l'extraction des données à travers des langages spéciaux. L'exemple le plus populaire est SQL (Structured Query Language).

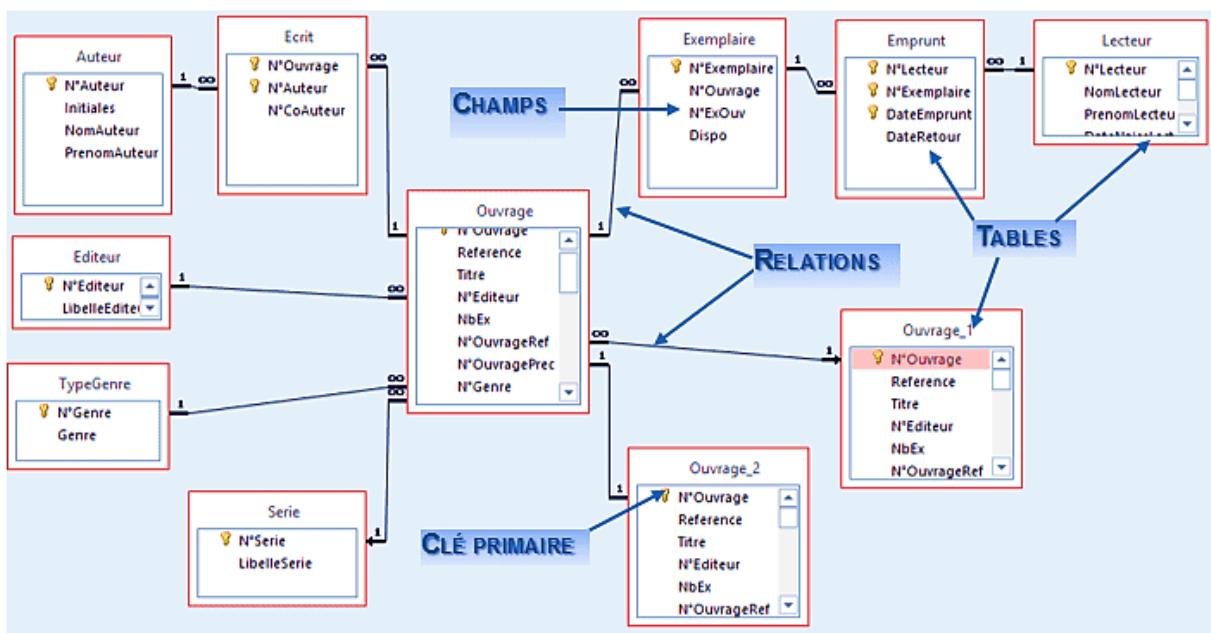


Figure 1. 3 Exemple de modèle relationnel [23]

5. SQL (Structured Query Language)

Est un langage déclaratif créé en 1970 par IBM, Il est destiné à la manipulation des bases de données au sein des SGBD et plus particulièrement des SGBDR.

En principe c’est un langage qui permet aux utilisateurs de traduire leurs besoins aux machines. [19]

SQL est un langage de définition de données (LDD), c'est-à-dire qu'il permet de créer des tables dans une base de données relationnelle.

SQL est un Langage de manipulation de données : SQL est un langage de manipulation de données (LMD), cela signifie qu'il permet de sélectionner, insérer, modifier ou supprimer des données.

SQL est un Langage de protections d'accès : Il est possible avec SQL de définir des permissions au niveau des utilisateurs d'une base de données. On parle de DCL (Data Control Language). [7]

SQL est le langage le plus couramment utilisé pour les bases de données relationnelles. Il est utilisé dans les système de gestion de bases de données relationnelles telles que :

- Oracle (Oracle Corporation).
- MySQL.
- SQL Server et Access (Microsoft Corporation) ... etc.

Nous utiliserons SQL pour créer des tables et manipuler des données dans ces tables.

6. SGBD Oracle

Oracle est un système de gestion de bases de données édité par la société (Oracle Corporation - www.oracle.com), leader mondial des bases de données. [7]

6.1. Les fonctionnalités d'Oracle

Oracle est un SGBD permettant d'assurer :

- 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.
- La gestion des accès concurrents.

6.2. Les composants d'Oracle

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. [21]

6.3. Architecture d’un serveur Oracle

Un système de gestion de base de données relationnelles (SGBDR) est un système qui va permettre de gérer le contrôle d’accès à une base de données relationnelles. Les SGBDR peuvent être soit du type client-serveur, soit de type fichiers partagés mais les plus utilisés à nos jours sont les SGBDR de type client-serveur. [8]

Un serveur Oracle est un système qui permet de gérer les bases de données et qui offre un moyen de gestion des données, il est constitué d’une instance et d’une base de données.

Un serveur Oracle = Instance Oracle + Une base de données

Une Session est une connexion spécifique entre un utilisateur et un serveur Oracle. On peut ouvrir nombreuses sessions concurrentes d'un même utilisateur ou de plusieurs qui peuvent s'exécuter sur le serveur Oracle. [21] [10]

La figure suivante illustre l'architecture du serveur Oracle constituée des structures de stockage, des processus, et des structures mémoire.

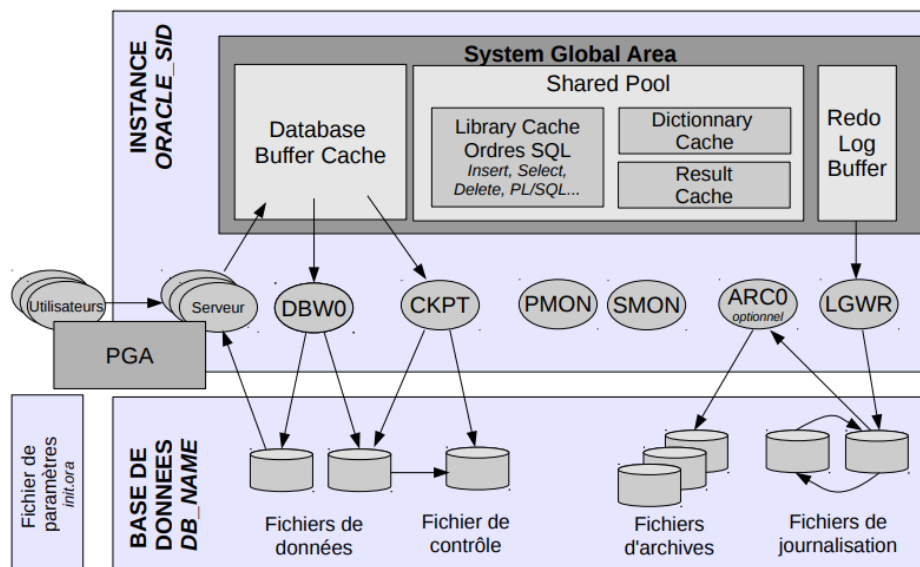


Figure 1. 4 Architecture d’un serveur Oracle [10]

6.3.1. Structure d'une instance Oracle

Tous les systèmes sont similaires en termes de principe de travail, Le principe de fonctionnement d’instance est presque le même pour le système d’exploitation, elle est considérée comme le canal et l’interface de communication entre la base de données et l’utilisateur.

L’instance Oracle permet d’accéder à la base de données Oracle et ne permet l’ouverture qu’une et une seule base de données.

Lorsqu'une instance est démarrée, Oracle Database :

- Alloue une zone de mémoire appelée SGA (System Global Area).
- Démarre les processus en arrière-plan.

L’instance est indispensable au bon fonctionnement d'une base de données Oracle. Sans instance il ne vous sera pas possible d'accéder à la base de données. [20]

6.3.1.1. Structures Mémoire Oracle

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

a) Mémoire SGA (System Global Area) :

La SGA est une zone de mémoire (lecture / écriture) qui est utilisée par la base de données pour partager les informations entre les processus, c’est à dire tous les processus peuvent lire des informations dans l'instance SGA, Aussi plusieurs processus écrivent sur la SGA pendant le fonctionnement de la base de données.

La SGA est un groupe de structures de mémoire partagée appelées composants SGA. Les composants SGA les plus importants sont les suivants :

- **Database Buffer Cache** : Est la zone de mémoire dans laquelle sont stockées des copies des blocs de données lus à partir de fichiers de données.
- **Redo Log Buffer** : Est un espace mémoire qui stocke les informations nécessaires pour reconstruire ou rétablir les modifications apportées à la base de données.
- **Shared Pool** : Met en cache différents types de données.
- **Large Pool** : Est une zone de mémoire fonctionne de la même manière que le Shared Pool, mais pour les allocations de mémoire supérieures à celles appropriées pour le Shared Pool.

- **Java Pool** : Est une espace de mémoire qui stocke tous les codes et données Java spécifiques à la session dans la machine virtuelle Java (JVM).
- **Streams Pool** : Stocke les messages de la file d'attente mise en mémoire tampon et fournit de la mémoire pour les processus de capture et les processus d'application Oracle Streams. Streams Pool est utilisé uniquement par Oracle Streams.

b) Mémoire PGA (Program Global Area):

La PGA est une mémoire spécifique à un processus en arrière-plan ou bien processus serveur, elle est allouée lors du démarrage du processus et désallouée lors de l'arrêt du processus, pour cela on ne peut pas allouer la PGA dans la SGA.

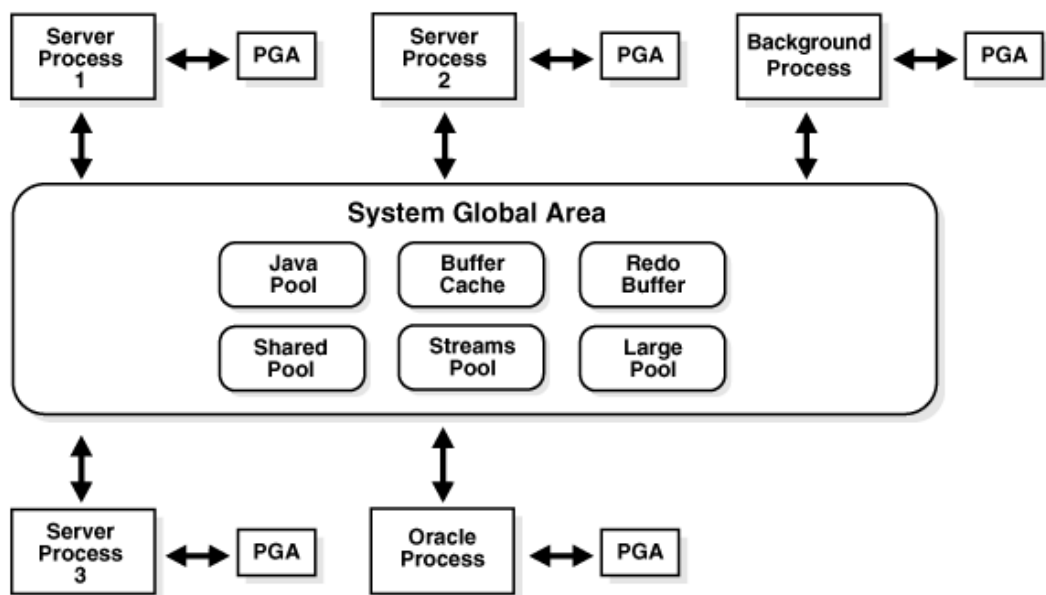


Figure 1. 5 Structures Mémoire Oracle [16]

6.3.1.2. Processus Oracle

Chaque processus d'une instance de base de données effectue un travail spécifique.

Les processus dans Oracle sont classifiés comme suit :

a) Les processus utilisateurs :

Généralement, le travail des processus utilisateurs est d'exécuter le code d'une application ou d'un outil Oracle (SQL*PLUS, ...).

b) Les Processus Serveurs :

Son rôle principal est de construire un canal de communication entre les processus utilisateurs et l'instance. [16]

c) Les Processus arrière-plan :

Les processus en arrière-plan sont créés automatiquement au démarrage d'une instance de base de données pour la maintenance requise et pour exploiter la base de données.

Il y a plusieurs processus d’arrière-plan, nous les citons :

- **SMON** : est un processus de maintenance qui va servir à corriger les problèmes de l'instance et à vérifier la synchronisation des données.
- **PMON** : est un processus surtout dédié aux processus des utilisateurs.
- **DBWn** : Le processus DBWn va être dédié à l'écriture du Database Buffer Cache dans les fichiers de données de la base de données et laisser assez de place disponible pour l'écriture des données dans le buffer.
- **CKPT** : C’est un processus parmi les processus de maintenance, il va servir à mettre à jour les en-têtes des fichiers de données.
- **ARCn** : Ce processus permet de copier un fichier REDO LOG FILE à un autre emplacement. [20]

6.3.2. Structure de la base de données Oracle

La structure globale de stockage d’une base de données oracle est composée d’une structure visible à l’OS et d’autre invisible. La structure visible correspond à la structure physique et la structure invisible correspond à la structure logique de la base de données.

6.3.2.1. Structure physique

La base de données Oracle se compose de plusieurs fichiers, ils sont organisés comme suit:

- **Les fichiers de contrôles** : Ce sont des fichiers qui contiennent les données sur la base de données elle-même. Sans ces fichiers, il est impossible d’ouvrir les fichiers de données pour accéder aux données de la base de données.
- **Les fichiers de données** : Ce sont des fichiers qui contiennent tous les informations stockées dans la base de données.
- **Les fichiers de journalisation** : Ce sont des fichiers qui vont servir pour reconstruire de la base de données en cas de crashe de celle-ci, en effet elles contiennent les données relatives aux changements apportés à la base de données. Si la base de données crashe et ne perd pas de fichier de données, alors l’instance peut recouvrir la base de données avec les informations que contiennent ces fichiers.

Il existe d’autres fichiers qui ne font pas partir officiellement de la base de données mais sont important pour le fonctionnement de la base de données il s’agit de :

- **Le fichier de paramètre** : Utilisé pour définir comment l’instance sera configurée à son démarrage.
- **Le fichier de mot de passe** : Permet aux utilisateurs de se connecter à distance à la base de données et d’effectuer des tâches d’administrations.
- **Les fichiers d’archivage** : Ces fichiers contiennent l’historique en cours du redo générée par l’instance, elles permettent le recouvrement de la base de données. L’utilisation de ces fichiers avec le backup de la base de données permet de recouvrer les fichiers de données perdues. [8]

6.3.2.2. Structure logique

Oracle Database alloue un espace logique pour toutes les données de la base de données. Les unités logiques d'allocation d'espace de base de données sont les blocks de données, les extents, les segments et les Tablespaces. [16]

Un Tablespace est un espace logique dans lequel sont stockés les objets de base de données (table, vue, index, procédure, etc.). Il ne peut appartenir qu’à une et une seule base de données à un certain moment et il est composé d’un ou plusieurs fichiers de données.

Les objets de base de données sont stockés dans des tablespaces sous forme de segment. Chaque segment contient un ou plusieurs extents.

Un extent est composé de blocks de données contigus, ce qui signifie que chaque extent ne peut exister que dans un data file. [8]

Les blocks de données sont de petite unité d’entrées/sorties logique de la base de données [11].

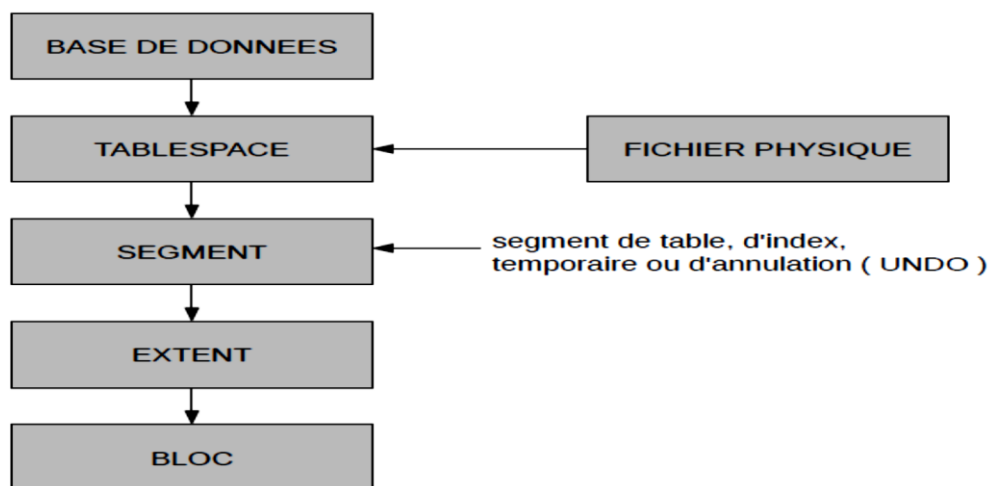


Figure 1. 6 Principe de stockage - Oracle [9]

7. Conclusion

Dans ce chapitre, nous avons donné un aperçu général sur les concepts de base de la base de données, nous avons défini la base de données, ensuite, nous avons parlé de système de gestion de base de données, SQL et le modèle relationnel.

Comme nous avons vu une idée sur le leader historique du marché des bases de données relationnelles, à savoir Oracle dans sa version 11g Professionnel.

D’après les principes de base nécessaires à la compréhension du fonctionnement du SGBD Oracle qu'on a vus dans ce chapitre, cela nous va faciliter de mettre en œuvre le SGBD oracle afin d’appliquer les techniques d’optimisation des bases de données sous oracle que nous en discuterons dans le chapitre suivant.

CHAPITRE : 02

**OPTIMISATION DES BASES DE DONNEES SOUS
ORACLE**

1. Introduction

Parmi les problèmes rencontrés par le chercheur et l'ingénieur dans le SGBD oracle, on trouve ceux qui concernent l'optimisation des bases de données.

L'optimisation peut se réaliser dans différents domaines, toujours avec le même objectif : améliorer le fonctionnement de quelque chose au moyen d'une gestion performante des ressources.

Dans le cadre des bases de données, L'optimisation des requêtes, consiste à améliorer le temps de réponse dans la gestion des bases de données. Ce qui concerne le langage des requêtes SQL, l'optimisation vise à simplifier les requêtes les plus complexes pouvant être plus longues à résoudre. [5]

Dans ce chapitre nous présentons les optimisations possibles des bases de données dans le SGBD oracle.

2. Présentation de l'optimisation

Il y a trois axes majeurs pour optimiser les performances d'une base de données :

- Optimisations au niveau : modèle de données.
- Optimisations au niveau : des requêtes.
- Optimisations au niveau : d'infrastructure.

2.1. Optimisation au niveau : modèle de données

Avant la conception de la base de données. Il faut faire attention lors du choix de la représentation des données mieux adaptée pour obtenir les meilleures performances.

Il est également nécessaire d'accorder une attention particulière aux performances lors de la conception de ces tables, aussi le nombre d'utilisateurs accédant à la base de données doit également être pris en compte. Parce que l'augmentation du nombre d'utilisateurs pouvant accéder à la base de données laisse un impact clair sur les performances de la base.

2.1.1. Présentation du modèle relationnel

Les bases de données relationnelles que nous étudions sont fondées sur le modèle relationnel inventé par Edgar Frank Codd en 1970 et largement adopté depuis cette époque. Un des grands intérêts du modèle relationnel est qu'il répartit l'ensemble des données dans différentes relations et établit des associations entre ces relations en utilisant leur clé primaire. Ce principe permet d'éviter la redondance d'information et donc d'avoir des données plus consistantes.

Le modèle relationnel a été conçu pour être utilisé avec une algèbre relationnelle qui permet d'effectuer des opérations sur les entités. Les principales sont : [13]

- **La sélection** : Filtre les tuples dont les attributs répondent à un prédicat.
- **La jointure** : Relie deux entités par leur association.
- **La projection** : Sélectionner seulement certains attributs d'une entité.

Bien évidemment, Pour faire des schémas de bases de données plus performants, il est important d'utiliser les bonnes pratiques de modélisation et de conception de base de données. Pour ce faire, Il existe une multitude d'astuces et de recommandation pour optimiser les performances d'une base de données, Les principales sont citées comme suit :

- **Structure des tables** : Le choix des types de données et la maîtrise de la qualité des données représentent, lorsqu'on ne les met pas en œuvre, plus de 70% des problèmes de performance, Par Exemple : Ne pas utiliser un BIGINT pour stocker un nombre ne dépassant pas les 10 chiffres, si un TINYINT suffit amplement.
- **Indexation** : Utiliser des indexes sur les colonnes qui en ont besoin, L'utilisation d'index est indispensable pour conserver de bonnes performances lors de la lecture des données.
- **Données** : Les données sont bien entendues importantes. Il convient tout de même de les maîtriser car le poids d'une base de données influe sur les performances, Si pour ça il faut éviter les données redondantes.

En principe, toutes les informations d'une base de données relationnelle peuvent également être stockées dans une table globale complète. Cela aurait l'avantage d'éviter la nécessité de lier les tables de la base de données, ainsi que la syntaxe complexe associée aux requêtes sur plusieurs tables. C'est cependant ce qui fait la force de la base de données et son modèle relationnel. La division de l'information en plusieurs tables sert à réduire les entrées doubles (appelées anomalies) et s'appelle la normalisation. [4]

Nous devons appliquer la normalisation pour assurer la bonne conception du modèle conceptuel de données, et éviter la redondance de ses données.

2.1.2. Normalisation

La normalisation correspond au processus d'organisation des données dans une base de données. Ce processus comprend la création de tables et l'établissement de relations entre celles-ci conformément à des règles conçues à la fois pour protéger les données et pour rendre la base de données plus flexible grâce à l'élimination de la redondance et des dépendances incohérentes.

Les données redondantes entraînent un gaspillage d'espace disque et créent des problèmes de maintenance. [1]

Nous ne pouvons pas parler de la normalisation sans mentionner la dépendance fonctionnelle, à cause de son rôle important pour diminuer la redondance.

2.1.2.1. Dépendance fonctionnelle :

Soit A et B deux attributs, B dépend fonctionnellement de A, si étant donné une valeur de A, il lui correspond une unique valeur de B (quel que soit l'extension).

Exemple : Soit la relation R suivante : R (cours, heure, jour, salle).

- (Salle, jour, heure) → cours

Parce que dans une salle à un moment donné ne peut avoir lieu qu'un cours.

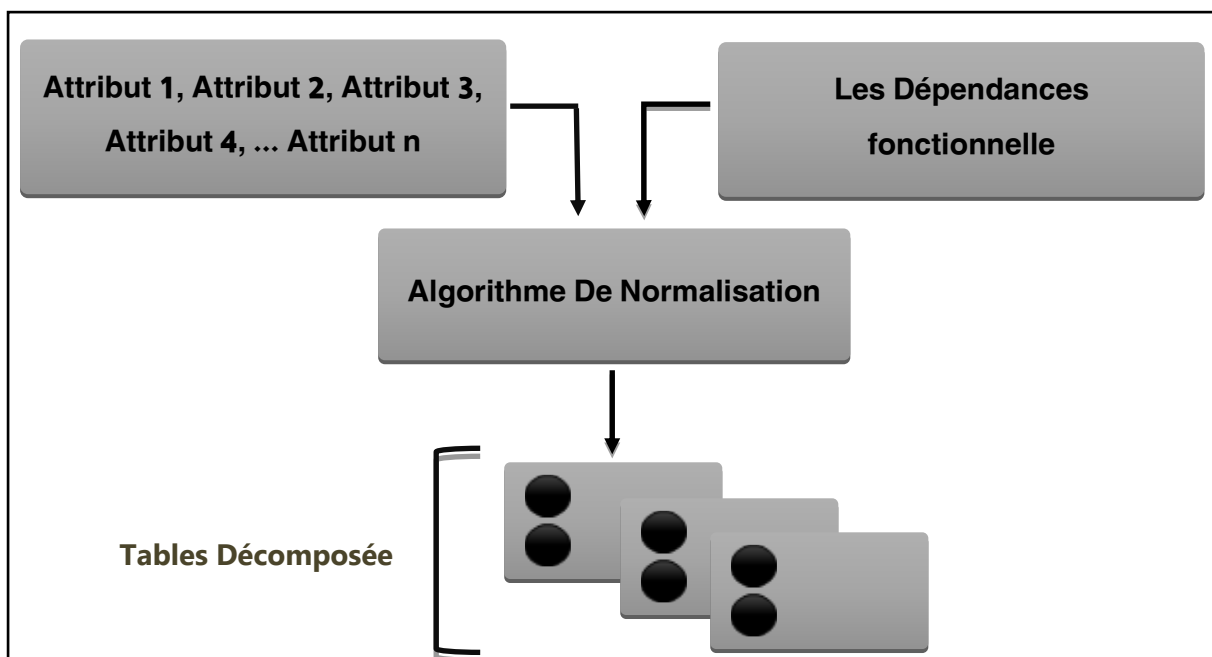


Figure 2. 1 Processus de normalisation

Le degré de normalisation peut être déterminé à base de formes normales prédéfinies. Les formes normales courantes pour les tables de bases de données relationnelles sont :

- La première forme normale (1NF).
- La deuxième forme normale (2NF).
- La troisième forme normale (3NF).
- La forme normale de Boyce Codd (BCNF).
- Autres formes de normalisation.

2.1.2.2. La première forme normale (1NF)

Pour qu'une entité soit en première forme normale (1NF), elle doit :

- Avoir une clé primaire.
- Être constituée de valeurs atomiques, ne pas contenir d'attributs ou d'ensembles d'attributs qui soient des collections de valeurs.

2.1.2.3. La Deuxième forme normale (2NF)

Pour qu'une entité soit en deuxième forme normale (2NF), il faut :

- Qu'elle soit en première forme normale (1NF).
- Que tous les attributs ne faisant pas partie de ses clés dépendent des clés candidates complètes et non pas seulement d'une partie d'entre elles.

2.1.2.4. La troisième forme normale (3NF)

Pour qu'une entité soit en troisième forme normale (3NF), il faut :

- Qu'elle soit en deuxième forme normale (2NF).
- Que tous les attributs ne faisant pas partie de ses clés dépendent directement des clés candidates.

2.1.2.5. La forme normale de Boyce Codd (BCNF)

Pour qu'une entité soit en forme normale de Boyce Codd (BCNF), il faut :

- Qu'elle soit en troisième forme normale (3NF).
- Que tous les attributs ne faisant pas partie de ses clés dépendent exclusivement des clés candidates.

2.1.2.6. Autres formes de normalisation

Il existe d'autres formes normales sont :

- La quatrième forme normale (4NF).
- La cinquième forme normale (5NF).
- La sixième forme normale (6NF).

Bien qu'elles ressemblent presque au troisième forme normale (3NF) et à la forme normale de Boyce Codd, mais elles sont un peu plus spécifiques et la forme normale de Boyce Codd est considérée comme tout à fait suffisante. [13]

2.1.2.7. Comment normaliser une table

Les étapes suivantes démontrent le processus de normalisation d’une table d’étudiants fictive.

- Table non normalisée :

N° d’étudiant	Conseiller	Salle-Conseiller	Cours1	Cours2	Cours3
1022	Aymen	412	101-07	143-01	159-02
4123	Rabeh	216	201-01	211-02	214-01

Tableau 2. 1 Table non normalisée

- Première forme normale : aucune répétition de groupe

Les tables ne doivent avoir que deux dimensions. Un étudiant ayant plusieurs cours, trois cours doivent être répertoriés dans une table distincte. Les champs Cours1, Cours2 et Cours3 des enregistrements ci-dessus sont l’indication d’un problème conceptuel.

Créez une autre table en première forme normale en éliminant le groupe répété (N° de cours), comme ci-dessous :

N° d’étudiant	Conseiller	Salle-Conseiller	N° de cours
1022	Aymen	412	101-07
1022	Aymen	412	143-01
1022	Aymen	412	159-02
4123	Rabeh	216	201-01
4123	Rabeh	216	211-02
4123	Rabeh	216	214-01

Tableau 2. 2 Table en première forme normale

- Deuxième forme normale : éliminer les données redondantes

Notez les valeurs multiples N° de cours pour chaque valeur N° d’étudiant dans la table ci-dessus. N° de cours ne dépend pas fonctionnellement de N° d’étudiant (clé primaire) ; cette relation n’est donc pas en deuxième forme normale.

Les deux tables suivantes démontrent la deuxième forme normale :

Étudiants :

N° d'étudiant	Conseiller	Salle- Conseiller
1022	Aymen	412
4123	Rabeh	216

Inscription :

N° d'étudiant	N° de cours
1022	101-07
1022	143-01
1022	159-02
4123	201-01
4123	211-02
4123	214-01

Tableau 2. 3 Table en deuxième forme normale

- Troisième forme normale : éliminer les données qui ne dépendent pas de la clé

Dans le dernier exemple, Salle-Conseiller dépend de manière fonctionnelle de l'attribut Conseiller. La solution consiste à déplacer cet attribut de la table Étudiants à la table Faculté, tel qu'illustré ci-dessous : [1]

Étudiants :

N° d'étudiant	Conseiller
1022	Aymen
4123	Rabeh

Faculté :

N° d'étudiant	Salle	Département
1022	412	42
4123	216	42

Tableau 2. 4 Table en deuxième forme normale

2.1.3. Dénormalisation

Si la normalisation conduit à la fragmentation des données dans plusieurs tables, la dénormalisation c'est absolument le contraire, dans ce processus on regroupe plusieurs tables liées par des références, en une seule table, c'est-à-dire, ce processus c'est un ensemble des règles appliquées au niveau du modèle logique des données (MLD).

L'objectif de la dénormalisation est d'améliorer les performances de la BDD en recherche sur les tables considérées, en implémentant les jointures plutôt qu'en les calculant.

2.1.3.1. Quand utiliser la dénormalisation ?

En dit qu'un schéma doit être dénormalisé lorsque les performances de certaines recherches sont insuffisantes et que cette insuffisance à cause des jointures, dans ce cas, il faut éliminer les jointures et regrouper les tables de ce schéma en une seule table.

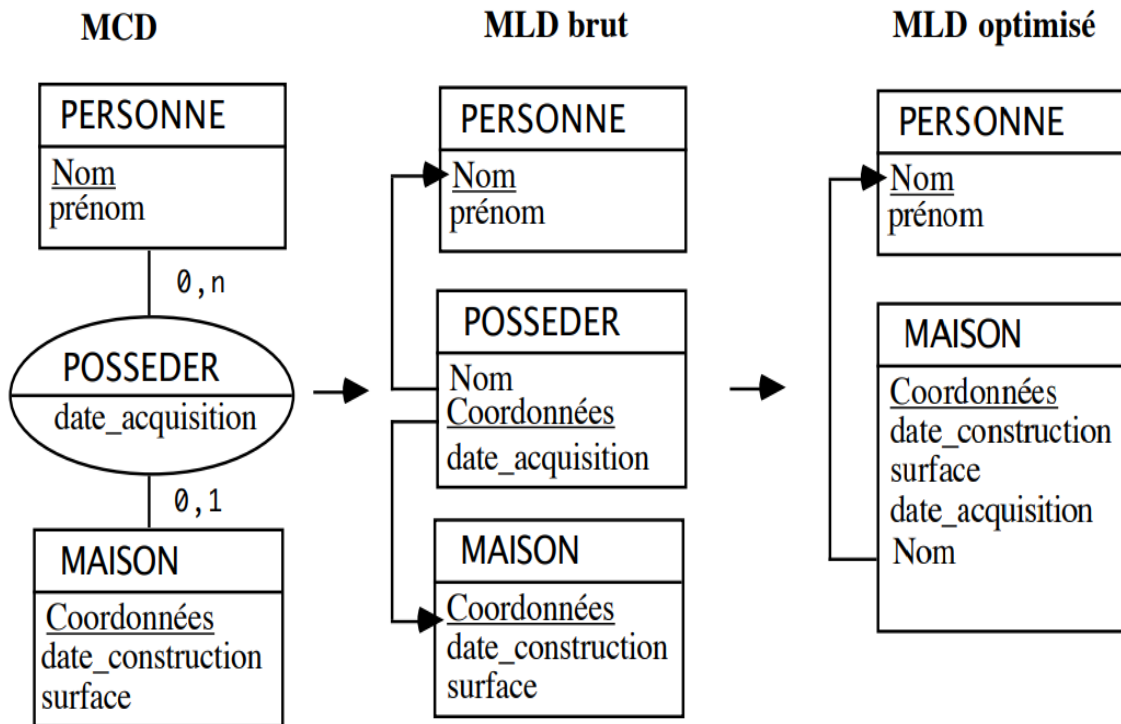


Figure 2. 2 Exemple de La dénormalisation [6]

La dénormalisation ouvre la porte aux redondances dans la base de données, cela conduit à des tables très volumineuse, à ce titre, elle doit être utilisée à bon escient et des moyens doivent être mis en œuvre pour contrôler la redondance créée.

Parmi les moyens qui aident à diminuer l'impact de la redondance sur les performances, nous allons présenter le partitionnement de table :

2.1.3.2. Le partitionnement de table

Le partitionnement d'une table consiste à découper cette table afin qu'elle soit moins volumineuse, permettant ainsi d'optimiser certains traitements sur cette table.

a) Partitionnement vertical

C'est une technique qui permet de découper une table en plusieurs tables, chacune ne possédant qu'une partie des attributs de la table initiale.

b) Partitionnement horizontal

C'est une technique qui permet de découper une table en plusieurs tables, chacune ne possédant qu'une partie des enregistrements de la table initiale. [19][6]

2.2. Optimisations au niveau : requêtes

Une requête est une traduction des ordres d'utilisateurs et ce qu'il veut extraire de la base de données, pour cela, elle a une grande importance. Sans elle, la base de données devient une pile d'informations inutile, c'est pour ça, il faut les manipuler avec beaucoup de prudence.

Il existe beaucoup des mécanismes aidant le SGBD à traiter les requêtes avec les meilleures performances possibles et nous allons les présenter dans ce qui suit :

2.2.1. Statistiques sur les données

Les statistiques Oracle sont en fait la récupération de diverses informations concernant la volumétrie des tables, la distribution des différentes valeurs des champs indexés, cet ensemble d'informations générera via un algorithme propre à Oracle. [4]

Les statistiques sont des informations sur l'inclus de la base, elles sont principalement pour le fonctionnement de l'optimiseur basé sur le coût CBO (Cost Based Optimizer), pour cela le moteur d'optimisation a besoin de connaître le contenu de BDD et pour collecter ces informations on utilise le package DBMS_STATS, qui fournit à l'optimiseur des informations plus détaillées.

Ce package permet de gérer les statistiques avec les opérations suivantes :

- ❖ Collecte des statistiques (Gathering Optimizer Statistics) :
 - Pour une table on a la procédure GATHER_TABLE_STATS.
 - Pour un index on a la procédure GATHER_INDEX_STATS.
 - Pour un schéma on a la procédure GATHER_SCHEMA_STATS.
 - Pour toute la base on a la procédure GATHER_DATABASE_STATS.
- ❖ Effacement de statistiques.
- ❖ Transfert de statistiques (export/import).
- ❖ Restauration d'anciennes statistiques ... et d'autres.

Exemple de collecte des statistiques du table CLIENTS :

```
exec dbms_stats.gather_table_stats(null, 'clients');
```

2.2.1.1. Optimiseur basé sur le coût CBO

Le CBO est un optimiseur qui est basé sur l'estimation des coûts d'exécution des opérations des plans d'exécution. Pour une requête donnée, le SGBDR établit plusieurs plans d'exécution possibles, le CBO estime pour chacun d'eux le coût d'exécution et choisit le moins élevé. Le rôle de l'optimiseur est de trouver le plan d'exécution le plus performant pour exécuter une requête.

2.2.1.2. Comment estimer le coût d'un plan d'exécution ?

Le SGBDR évalue le coût en ressources utilisées pour exécuter ce plan. Ces ressources sont : [13] [18]

- Le temps CPU.
- Le nombre d'E/S disque dur.
- La quantité de mémoire vive (RAM) nécessaire.

2.2.2. Indexes

Un index est une structure séparée dans la base de données qui sont utilisés pour accélérer les recherches. Créer un index ne modifie pas les données de la table, cela crée une nouvelle structure de données faisant référence à la table. En fait, un index de base de données ressemble très fortement à l'index d'un livre, les indexes effectuées un accès plus optimisé aux données au lieu de parcours séquentiel des enregistrements. [14]

Les indexes. Sous Oracle, ils peuvent être soit de type B-Tree soit de type bitmap.

2.2.2.1. B-Tree

L'index B-Tree est le plus commun type des indexes de base de données, leur principe est de diviser les valeurs en listes et ces listes sont réparties sur différents champs et chaque valeur pointe sur une valeur ou bien une liste du champ qui suit, on appelle le premier liste la racine et les autres appelé les feuilles. Ainsi, ils permettent de trouver très rapidement une valeur précise en parcourant l'arbre de la racine vers les feuilles plutôt de parcourir tous les enregistrements de la table ligne par ligne.

On prend l'exemple indiqué dans la Figure 2.3, Pour obtenir une valeur dans cette structure arborescente, on a indexé la colonne ID et on va chercher sur la valeur 1015.

Le SGBD va procéder de la manière suivante :

- 1- Recherche ligne par ligne et compare notre valeur avec les valeurs dans le bloc racine, si la valeur est inférieure on passe à la ligne suivante sinon on choisit la valeur précédente ce qui pointe au premier bloc branche.

- 2- Dans le bloc branche, il répète la même manière, ce qui le conduit à un bloc feuille.
- 3- Dans le bloc feuille, il recherche la valeur. S’il n’existe pas, il s’arrête et renvoie que la valeur recherchée n’existe pas dans la table. S’il existe, la valeur dans ce bloc a un pointeur qui permet d’obtenir l’enregistrement correspondant.

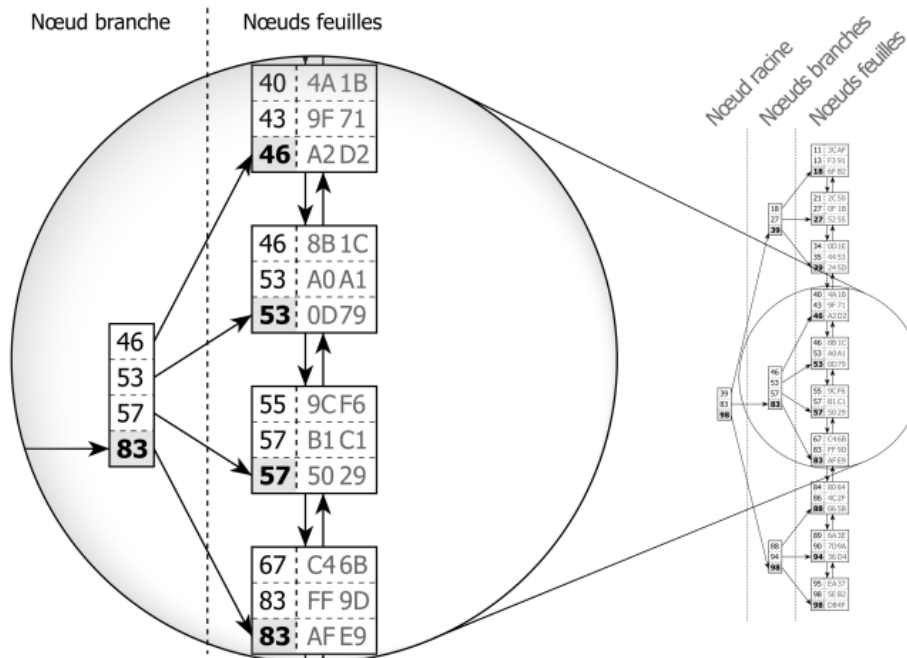


Figure 2. 3 Exemple d'un index B-Tree [14]

2.2.2.2. Bitmap

L’index bitmap est adapté à la manipulation de colonnes avec peu de valeurs distinctes. L’indexation est effectuée en collectant les valeurs distinctes d’une colonne et en mettant « 1 » si la ligne contient cette valeur ou « 0 » si elle ne contient pas. Pour plus de simplification, (Voir Figure 2.4). [11] [18]

N° ligne	EMPNO	ENAME	JOB	COMPTABLE	VENDEUR	PRESIDENT	MANAGER	INGENIEUR
1	14347	DUPOND	COMPTABLE	1	0	0	0	0
2	14484	MICHEL	VENDEUR	0	1	0	0	0
3	14621	LEROY	VENDEUR	0	1	0	0	0
4	14758	DURAND	MANAGER	0	0	0	1	0
5	14895	MEUNIER	VENDEUR	0	1	0	0	0
6	15032	MARTIN	MANAGER	0	0	0	1	0
7	15169	DUPONT	MANAGER	0	0	0	1	0
8	15306	LECLERC	INGENIEUR	0	0	0	0	1
9	15443	NEUVILLE	PRESIDENT	0	0	1	0	0
10	15580	DUJARDIN	VENDEUR	0	1	0	0	0
11	15717	BERTHIER	COMPTABLE	1	0	0	0	0
12	15854	GUY	COMPTABLE	1	0	0	0	0
13	15991	RENAUD	INGENIEUR	0	0	0	0	1
14	16128	LAVAU	COMPTABLE	1	0	0	0	0

Figure 2. 4 Exemple d'un Index bitmap [13]

2.2.3. Réécriture des requêtes

La réécriture des requêtes est un mécanisme parmi les mécanismes d'optimisation qui, de manière générale, améliore les performances très significativement. D'une autre manière, la façon d'écrire votre requête a un impact très significatif.

L'exécution d'une requête SQL suit les étapes suivantes :

- Analyse syntaxique : Vérification de la correction syntaxique.
- Contrôle : Vérification des droits d'accès.
- Optimisation : Génération de plans d'exécution et choix du meilleur.
- Exécution : Compilation et exécution du plan choisi.

L'étape optimisation est une tâche de l'optimiseur CBO. Cette étape consiste à modifier la requête qu'on a déjà exécuté en une requête équivalente afin de la redonner plus performante. Mais il existe des cas que le SGBD ne peut pas le faire ces changements tout seul, pour cela l'utilisation des écritures les plus performantes permettra d'améliorer les résultats. [19][9]

2.2.4. Hints

Les hints sont des règles ou bien des conseils que l'on donne à l'optimiseur (CBO) pour obliger ce dernier à suivre un plan d'exécution.

La syntaxe suivante est utilisée pour les hints :

```
Select /*+ hint */ nom from . . .
```

Les hints relèvent des classifications générales suivantes :

2.2.4.1. Hints des approches d'optimisation pour l'objectif

Les Hints de cette section permettent de choisir entre les approches d'optimisation basées sur les coûts et sur les règles. Pour l'objectif du meilleur temps de réponse.

Par exemple : ALL_ROWS, FIRST_ROWS(n), CHOOSE, RULE.

2.2.4.2. Hints de chemin d'accès

Chaque hint dans ce terrain spécifier un chemin d'accès à l'optimiseur s'il existe déjà.

Par exemple : FULL, INDEX, NO_INDEX.

2.2.4.3. Hints de transformation de requête

Les hints de transformation de requête ont pour but de transformer la requête avec certaine manière ou bien d'interdire la transformation à l'optimiseur.

Par exemple : REWRITE, MERGE ... etc.

2.2.4.4. Les hints de jointure

Les Hints de cette section permettent de manipuler les jointures qui relient les tables. On utilise ces hints si on connaît le nombre d'enregistrement définis dans chaque table que l'optimiseur ne connaît pas.

Par exemple : ORDERED, USE_MERGE, LEADING ... etc.

2.2.4.5. Les hints de parallélisme

Les hints de cette section permettent de paralléliser les opérations et contrôler les serveurs de parallélisations.

Par exemple : PARALLEL, PARALLEL_INDEX ... etc. [17] [18]

2.2.5. SQL avancé

Ces fonctions avancées permettent de transformer des requêtes très compliquées aux requêtes très simples pour améliorer les performances, il y a beaucoup de fonctions mais dans ce mémoire nous citons ce qui suit : [13]

- **Les Grouping Sets** : permettent d'utiliser les fonctions d'agrégats sur plusieurs groupes différents dans la même requête.
- **Rollup Group By** : L'opérateur ROLLUP GROUP BY permet de générer les sous-totaux de GROUP BY multi-colonnes dans une requête unique.
- **Cube Group By** : L'opérateur CUBE GROUP BY permet de générer les sous-totaux de toutes les dimensions des GROUP BY multi-colonnes dans une requête unique.

2.3. Optimisations au niveau : infrastructure

L'infrastructure en informatique désigne les équipements matériel et logiciel, à ce titre, nous allons présenter l'amélioration du coût matériel et logiciel, car les performances du matériel informatique influent directement sur la performance de la base des données.

2.3.1. Optimisation de l'exécution du SGBDR

Cette tâche est plutôt dans le périmètre du gestionnaire de la base de données, c'est-à-dire, pour créer un BDD dans le SGBD oracle, vous devez passer par plusieurs étapes, parmi ces étapes il existe une étape qui permet aux utilisateurs de spécifier certains paramètres du serveur d'oracle, y compris la mémoire « SGA et PGA » et autres, Il faut les manipuler avec beaucoup de prudence parce qu'ils peuvent avoir de lourdes conséquences.

2.3.1.1. Ajustement de la mémoire utilisable

La plupart des SGBDR intègrent un paramètre permettant de configurer la quantité maximale de mémoire qu'ils peuvent utiliser. Il faut veiller à ce que ces paramètres n'étouffent pas le serveur (il ne faut pas changer ces zones), mais il faut que le SGBDR utilise au mieux la mémoire disponible sur le serveur.

Généralement, sous Oracle, ces paramètres sont automatiquement configurés en fonction de la mémoire disponible au moment de la création de la base de données, Par défaut, Oracle 11g consomme une certaine quantité de mémoire pour son serveur, estimée à 40% de la mémoire vive (RAM) totale de l'ordinateur.

2.3.1.2. Répartition des fichiers

Répartir les fichiers de données, de contrôle et de journalisation, sur des disques différents améliore généralement les performances en lecture et en écriture. Pour les fichiers de contrôle Oracle, cette pratique est recommandée pour des raisons de fiabilité.

Répartir les données et les index sur des tablespaces qui sont sur des disques différents est une bonne pratique assez répandue.

2.3.2. Optimisation matérielle

Un des moyens d'amélioration des performances consiste à améliorer les périphériques utilisés pour faire tourner le SGBDR, Cette solution devrait être envisagée seulement lorsque toutes les autres ont été épuisées, Parce que les solutions que nous avons vues précédemment, vous permettent de vous passer complètement de ce moyen. [13]

Nous allons étudier les impacts de l'amélioration des éléments principales.

- Processeur (CPU)
- La mémoire vive (RAM)
- Disques durs
- Le réseau

3. Conclusion

Nous avons présenté dans ce chapitre les différents mécanismes d'optimisation de base de données sous oracle aux différents niveaux tels que : la normalisation aux niveaux structurelle de la base de données qui permet de réduire les redondances, les indexes qui permettent de réduire le temps d'exécution des requêtes en offrant un accès rapide aux données et Optimisation d'infrastructure qui permettent d'améliorer les performances du matériel pour le bon fonctionnement du SGBD.

Dans le chapitre suivant nous allons appliquer ces mécanismes et voire la différence avec et sans utilisation de l'optimisation.

CHAPITRE : 03
EXPERIMENTATION ET ANALYSE

1. Introduction

Les bases de données sont devenues indispensables dans la gestion des entreprises. Au fil du temps, nous remarquons que le monde évolue constamment et que les requêtes deviennent plus complexes et très consommatrices de ressources et peuvent ralentir la base de données pendant un petit bout de temps.

Dans ce chapitre nous allons appliquer quelques techniques d'optimisation que nous avons vu dans le chapitre précédent, pour cela, on a créé un exemplaire de base de données qui contient des tables du grand taille en utilisant d'Oracle 11g professionnel, pour valider ces mécanismes.

2. La base de données de test

Pour faire les tests, on a utilisé deux bases du livre « Optimisation des bases de données Mise en œuvre sous oracle » qui est dans la bibliographie sur le numéro 13 et disponible sur le site de l'auteur à l'adresse <http://www.altidev.com/livres.php>, ces bases définis comme suit :

- **La première base de test** : Cette base est basée sur le schéma « SCOTT » prédéfini sous Oracle, ce schéma contient très peu d'enregistrements mais nous l'avons répété 100 000 fois pour obtenir des tables de grande taille.

BIGEMP (EMPNO, ENAME, JOB, MGR, MONTANT, HIREDATE, SAL, COMM, DEPNO, DNAME, LOC)

EMPNO : ID d'employé.

ENAME : Nom de l'employé.

JOB : Profession.

HIREDATE : Date d'embauche.

SAL : Salaire.

DEPNO : ID de département.

DNAME : Nom du département.

LOC : La localisation du département.

MONTANT : Le montant d'augmentation de chaque employé.

MGR : numéro d'employé responsable direct.

COMM : commission.

La table	Le nombre d'enregistrements
Bigemp	1 400 014
Bigdept	400 004
Augmentation	1 400 014

Tableau 3. 1 Taille de chaque table de la base de test BIGEMP

- **La deuxième base de test :** Cette base est composée de quatre tables avec très grand d'enregistrements, elle est présentée dans le diagramme ci-dessous :

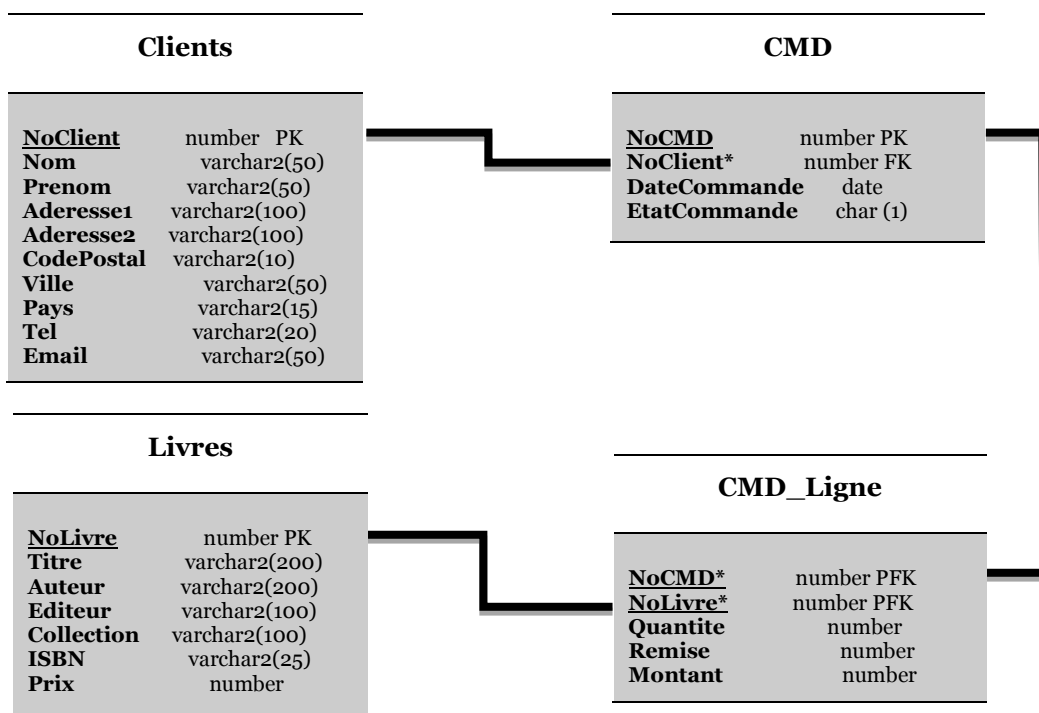


Figure 3. 1 Le MPD de la deuxième base de test

La table	Le nombre d'enregistrements
Cmd_lignes	2 606 477
Cmd	1 000 000
Clients	45 000
Livres	2 973

Tableau 3. 2 Taille de chaque table de la deuxième base de test

En ce qui concerne les tests de la matériel et les requêtes, nous allons utiliser la deuxième base de test qu'on a déjà présenté précédemment, cependant, pour les tests du modèle de données, nous allons utiliser la base BIGEMP mais avec des structures différentes :

- **La première structure « BIGEMP1 »** : Cette base est dénormalisée c'est-à-dire tous les attributs dans une seule table.

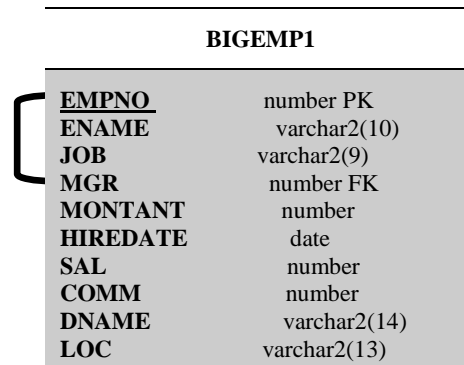


Figure 3. 2 Le MPD de la première structure

- **La deuxième structure « BIGEMP »** : Dans cette base on a utilisé les formes de la normalisation pour former ces différentes tables.

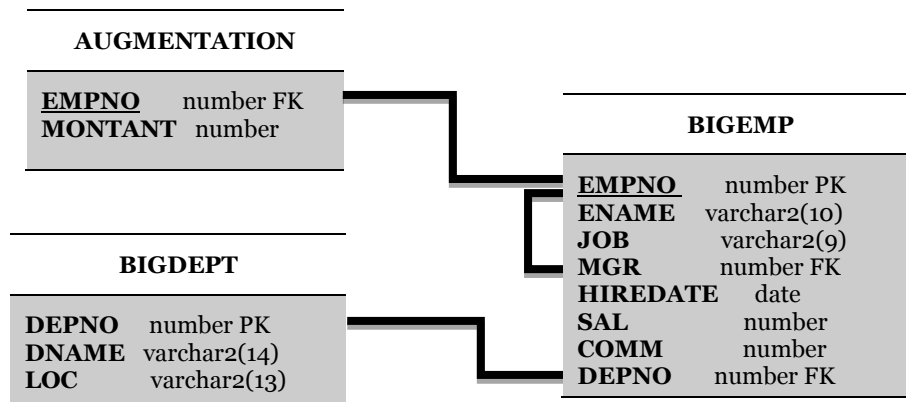


Figure 3. 3 Le MPD de la deuxième structure

3. Environnement du travail

Pour commencer n'importe quel travail, on doit recourir à un environnement qui nous aide à la réussite de ce travail, en ce qui concerne le périmètre de notre travail (Les Rôles du DBA) qui se compose en deux axes principaux sont :

- L'installation d'Oracle database 11g Professionnel (Voir Annexe A1).
- La création de la base de données avec le remplissage de ses tables (Voir Annexe A2).

Avant de commencer les tests, il faut d'abord connaître les critères de mesure et le plus important sont les méthodes et les outils permettant de valider la différence entre le mode normal et après l'optimisation.

4. Eléments d'analyse et diagnostic

Dans cette section, nous verrons les critères de mesure et les outils de diagnostic qui nous permettent de juger les résultats obtenus.

4.1. Les critères de mesure

La mesure c'est une méthode de description des informations en utilisant des chiffres. Généralement en physique, pour suivre l'évolution de n'importe quel phénomène, il doit d'abord d'identifier les facteurs aidant à observer ce développement.

Comme l'a dit William Deming, "On ne peut améliorer que ce que l'on mesure !" [13]. Les plus importants critères utilisés sont :

- Le temps nécessaire pour la réponse.
- La consommation du mémoire vive RAM.
- La consommation CPU.
- Le nombre d'Entrées /Sorties disque.

4.2. Les outils d'optimisation

Dans notre vie quotidienne, on doit connaître les mesures, mais on ne peut pas les connaître sans utiliser des outils spéciaux, Oracle fournit des utilitaires d'aide à l'optimisation de requêtes SQL et dans cette section on va présenter les méthodes qu'on a appliqués.

4.2.1. Le plan d'exécution

C'est une table qui contient des informations sur le plan que l'optimiseur CBO doit l'exécuter pour une requête donnée, c'est à dire la liste des opérations qui vont être exécutées et l'estimation du coût de chaque opération.

Sous Oracle on peut consulter le plan d'exécution par différentes méthodes :

- Par la commande SQL : **Explain plan for < La requête >**

Après ça afficher le plan par la requête suivante :

```
Select * from table(dbms_xplan.display);
```

- Par l'activation de la trace automatique : **SQL> Set autotrace on.**

```
Plan d'exécution
-----
Plan hash value: 41783786
-----
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	48	2603 (2)	00:00:32
1	NESTED LOOPS		1	48	2603 (2)	00:00:32
* 2	TABLE ACCESS FULL	BIGEMP	1	45	2603 (2)	00:00:32
* 3	INDEX UNIQUE SCAN	PK_DEPT	1	3	0 (0)	00:00:01

Figure 3. 4 Plan d'exécution avec SQL*Plus

4.2.2. Analyse les SQL Trace avec TKPROF

TKPROF c'est un outil d'Oracle qui permet de générer depuis un fichier de trace (.trc) un rapport (fichier texte lisible) qui contient des informations sur les temps d'exécution des requêtes SQL, le nombre de blocs recherchés en mémoire et sur disque ... etc.

Pour générer un rapport avec TKPROF, il faut suivre les étapes suivantes :

- **1^{er} Etape** : Il faut activer la trace de la session avec la commande :
`SQL> alter session set sql_trace = true;`
- **2^{ème} Etape** : exécuter la requête.
- **3^{ème} Etape** : Il faut désactiver la trace de la session avec la commande :
`SQL> alter session set sql_trace = false;`
- **4^{ème} Etape** : Exécuter le TKPROF depuis le CMD Windows avec le nom du fichier de trace et l'emplacement où vous souhaitez placer le rapport comme suit :
`> TKPROF <fichier.trc> C:\Users\Administrateur\Desktop\rapport.txt`

```
SQL> alter session set sql_trace = true;
Session modifiée.
SQL> select *
  2  from emp;
SQL> alter session set sql_trace = false;
Session modifiée.
SQL>
```

Figure 3. 5 Création du fichier de trace

```

TKPROF: Release 11.2.0.2.0 - Development on Jeu. Mai 23 03:13:12 2019

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.

Trace file: oacle_ora_5820.trc
Sort options: default

*****
count  = number of times OCI procedure was executed
cpu    = cpu time in seconds executing
elapsed = elapsed time in seconds executing
disk   = number of physical reads of buffers from disk
query  = number of buffers gotten for consistent read
current = number of buffers gotten in current mode (usually for update)
rows   = number of rows processed by the fetch or execute call
*****
select *
from emp

call      count      cpu    elapsed      disk    query    current    rows
-----
Parse     1          0.00    0.05         0         2         0         0
Execute   1          0.00    0.00         0         0         0         0
Fetch     2          0.00    0.00         0         4         0        14
-----
total     4          0.00    0.05         0         6         0        14

Misses in library cache during parse: 1
Optimizer mode: ALL_ROWS
Parsing user id: 5
Number of plan statistics captured: 1

Rows (1st) Rows (avg) Rows (max)  Row Source Operation
-----
      14         14         14  TABLE ACCESS FULL EMP (cr=4 pr=0 pw=0 time=23 us cost=2 size=1218 card=14)

```

Figure 3. 6 Rapport généré par le TKPROF

Le rapport du TKPROF très intéressant, même s’il contienne des informations très précieuses, il affiche également tous les plans d’exécution possibles proposés par l’optimiseur pour améliorer les performances.

4.2.3. Moniteur de ressources Windows

Le moniteur de ressources est une application très utile et propre à Windows, nous allons utiliser les informations fournit par cette application afin d’indiquer l’utilisation CPU, Mémoire, disque ... etc. et cela pour surveiller les performances du serveur.

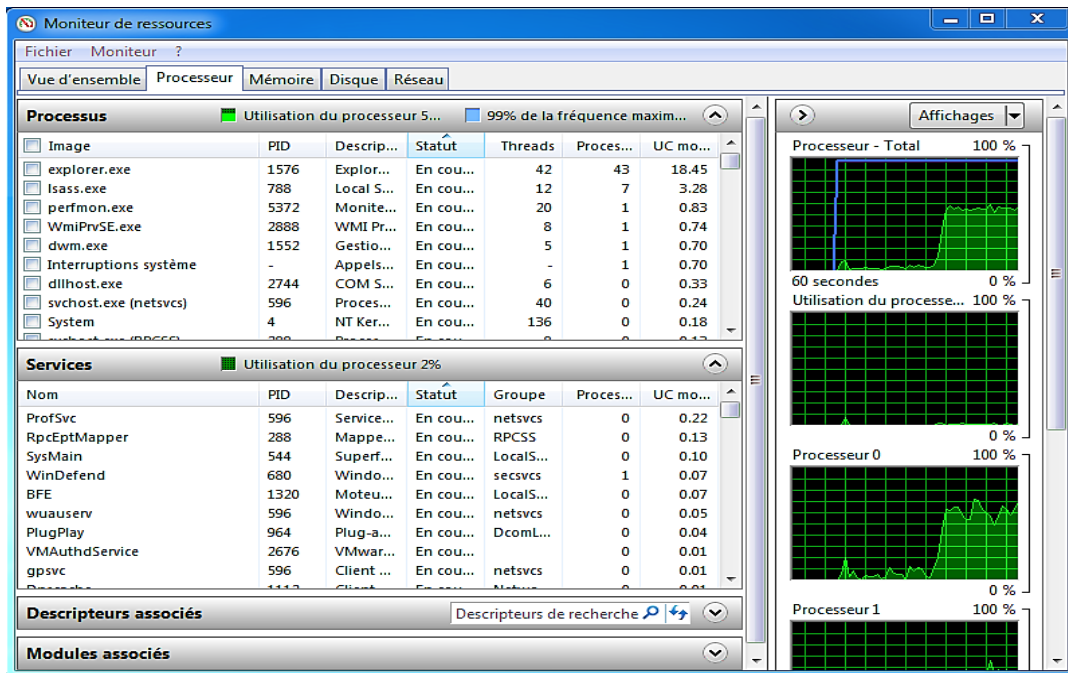


Figure 3. 7 Moniteur de ressources (CPU)

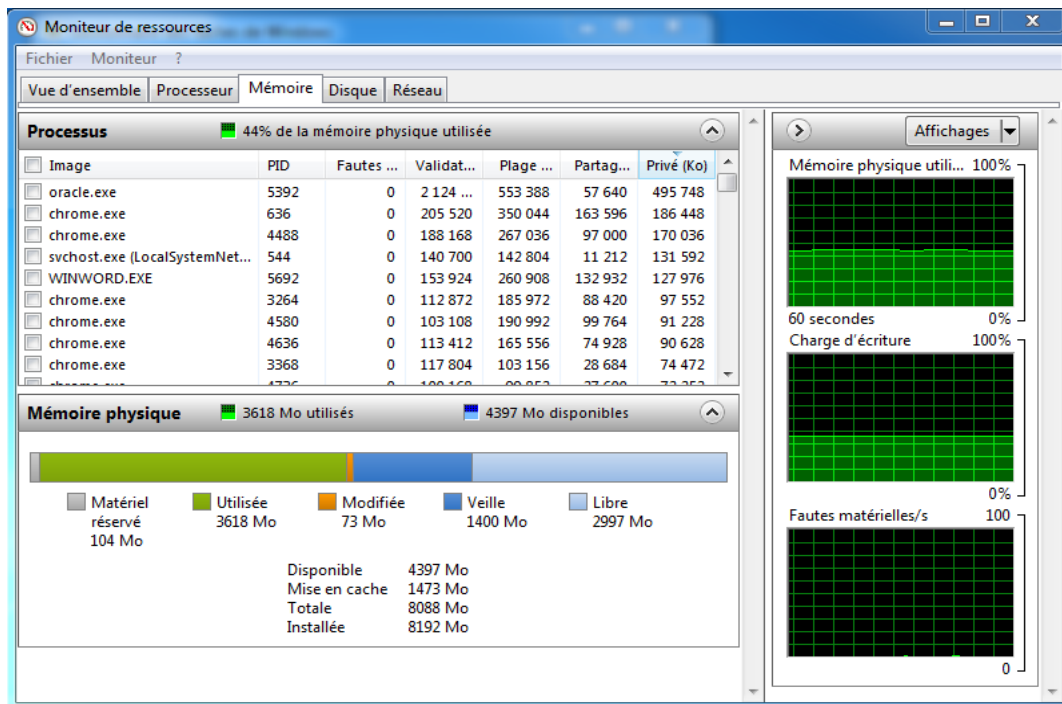


Figure 3. 8 Moniteur de ressources (RAM)

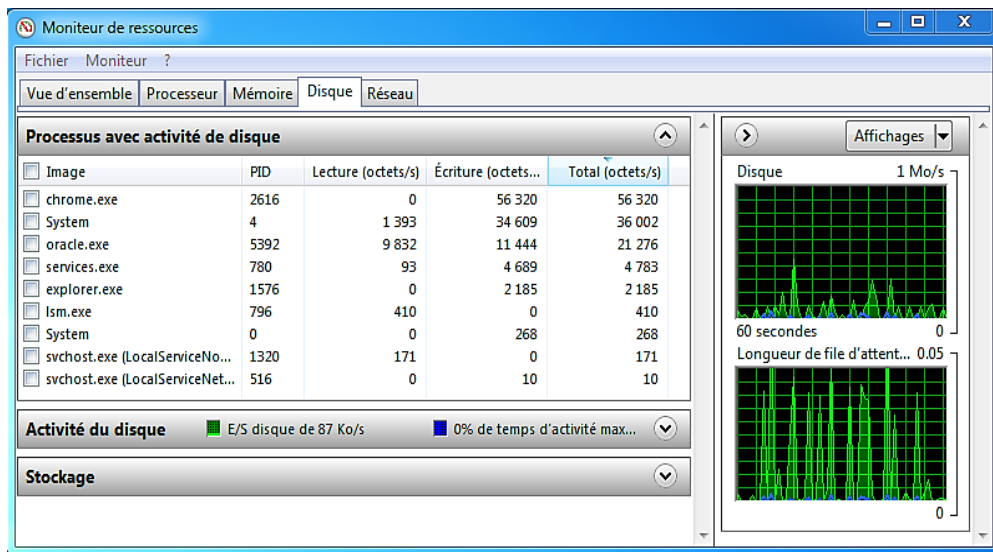


Figure 3. 9 Moniteur de ressources (E/S Disque)

4.2.4. Analyseur de performance de la base données : SolarWinds

C'est un logiciel développé par la société American SolarWinds qui fournit des informations détaillées pour aider les développeurs à surveiller la performance et la consommation du ressources (CPU, RAM, Disque ... etc.) par l'instance de la base de données.

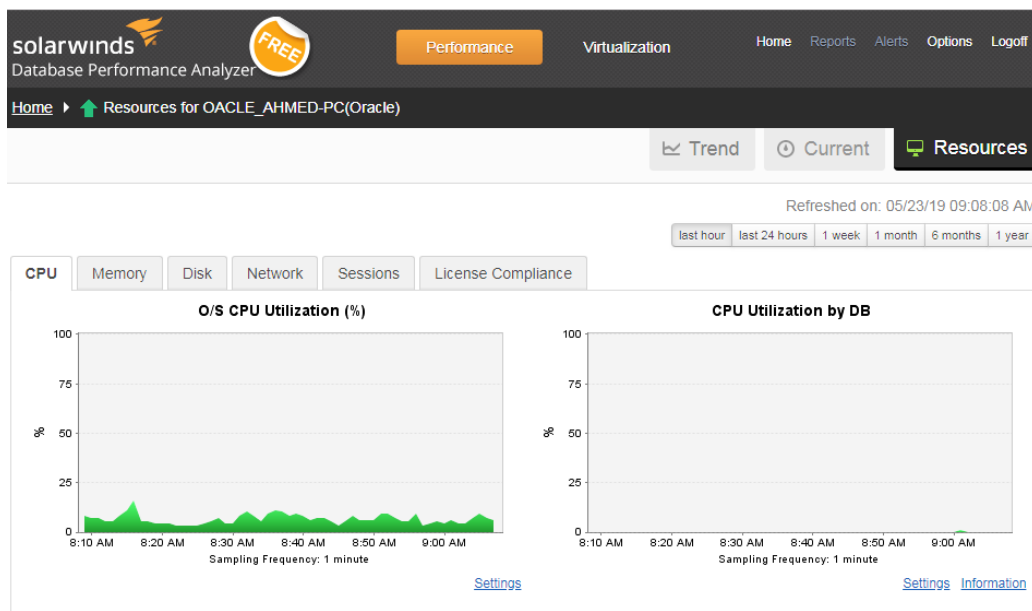


Figure 3. 10 Diagrammes d'utilisation CPU par SolarWinds

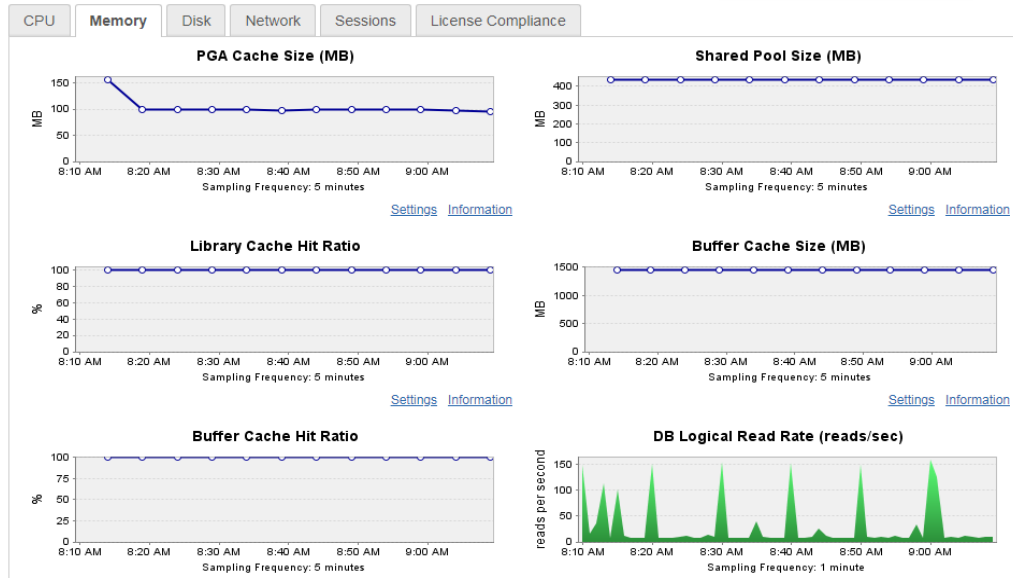


Figure 3. 11 Diagrammes d'utilisation Mémoire d'instance

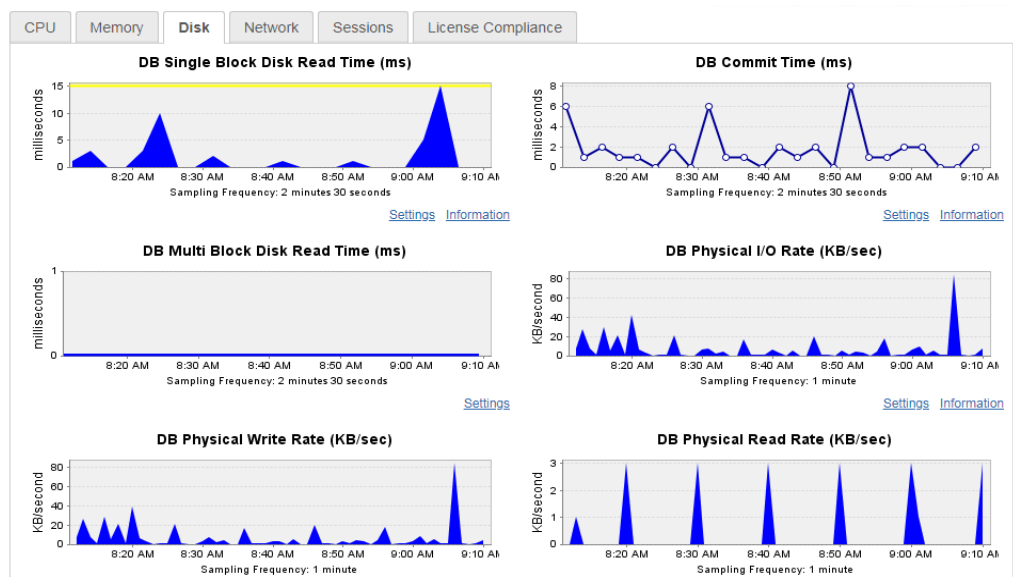


Figure 3. 12 Diagrammes d'utilisation (E/S) Disque

5. Résultats expérimentaux

Dans cette partie, nous allons appliquer les mécanismes d'optimisation citées au chapitre précédent et analyser les résultats de notre série des tests en utilisant les moyens cités dans ce chapitre.

5.1. Résultats des différentes structures

Comme nous avons déjà mentionné au début de cette chapitre, en ce qui concerne les tests de cette partie, nous avons généré deux structures différentes pour la même base de données. (Voir Figure 3.2, Figure 3.3)

5.1.1. Requêtes à exécuter

Requête 1 : sélectionner tous les attributs depuis les 3 tables si le montant supérieure à 1500.

Requête 2 : sélectionner juste les attributs qui nous intéresse depuis la table BIGEMP & AUGMENTATION si le montant supérieure à 1500 (Voir Annexe A3).

5.1.2. Présentation des résultats obtenus

	Requête 1		Requête 2	
	Structure 1	Structure 2	Structure 1	Structure 2
Temps Écoulé (s)	95	26.69	9.67	8.40
Coût CPU+(I/O)	95846	48259	43278	26650
RAM (Mo)	408.77	222	139	72

Tableau 3. 3 Résultats obtenus par les structure

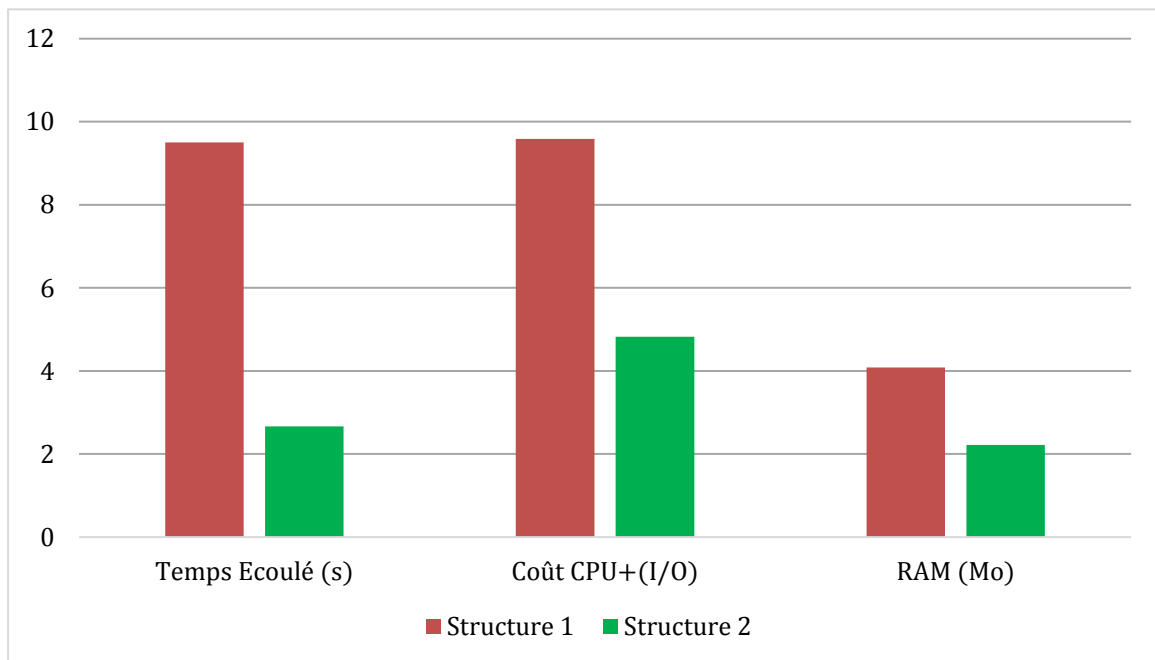


Figure 3. 13 Résultats du Requête 1 sur les deux Structures

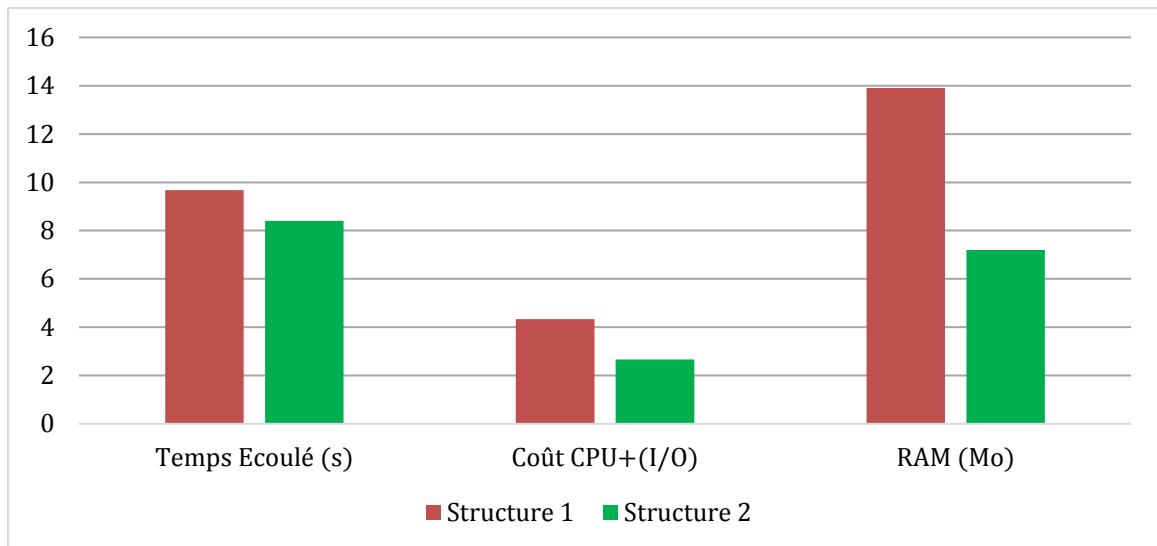


Figure 3. 14 Résultats du Requête 2 sur les deux Structures

5.1.3. Fragmentation Horizontale

Présentation des résultats obtenus

	Structure 2	
	Fragmentation Horizontale	Non-fragmenté
Temps Ecoulé (s)	1,21	3,08
Coût CPU+(I/O)	1574	14788
RAM (Mo)	20	20

Tableau 3. 4 Résultats obtenus par la fragmentation horizontale

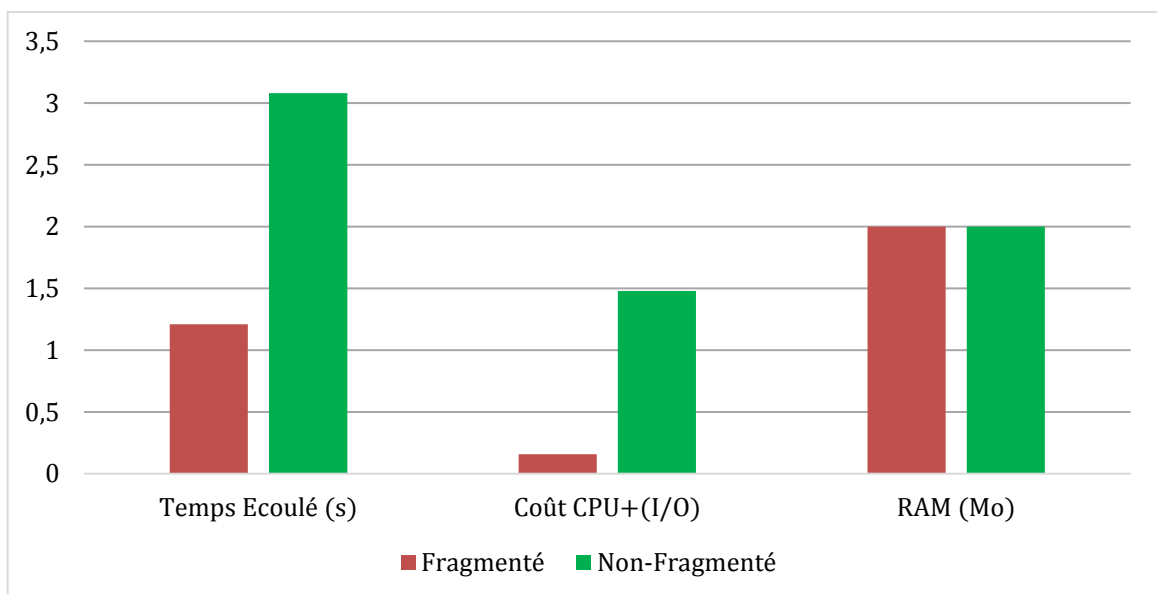


Figure 3. 15 Résultats obtenus par la fragmentation horizontale

5.1.4. Fragmentation Verticale

Présentation des résultats obtenus

	Structure 2	
	Fragmentation Verticale	Non-fragmenté
Temps Ecoulé (s)	0,89	1,09
Coût CPU+(I/O)	2620	14786
RAM (Mo)	0.8105	0.8105

Tableau 3. 5 Résultats obtenus par la fragmentation verticale

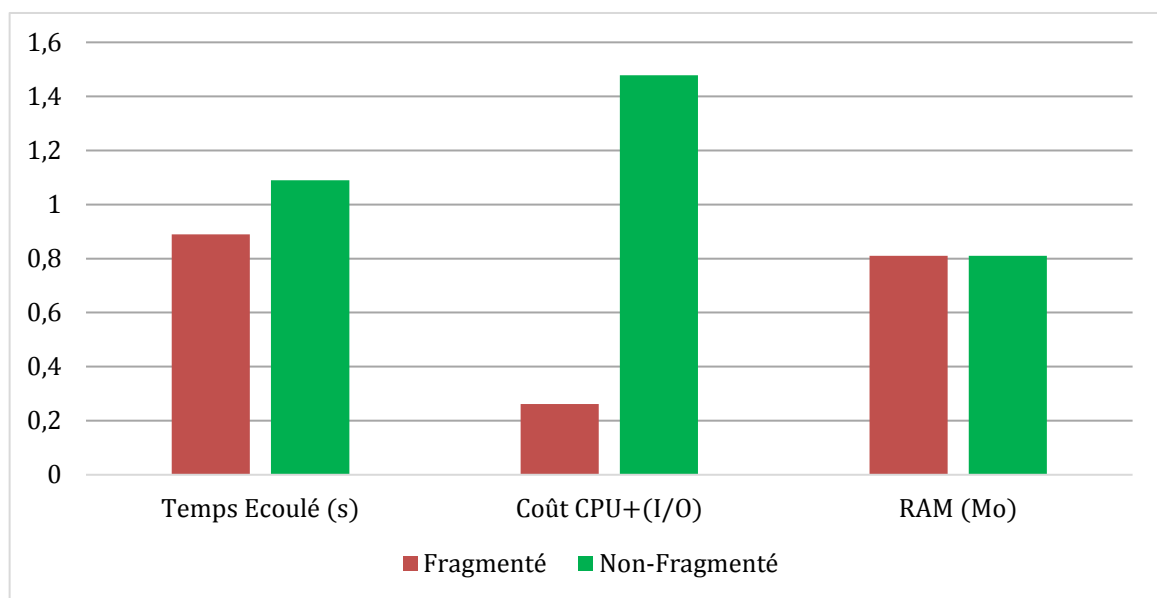


Figure 3. 16 Résultats obtenus par la fragmentation verticale

5.1.5. Analyse des résultats et Diagnostic

Selon les résultats obtenus, on trouve que la structure dénormalisée est mieux par rapport à la structure Normalisée. Malgré que la normalisation résolve plusieurs problèmes, surtout les redondances, mais la mal utilisation de ses règles dans la division du tables cause des catastrophes, avec le temps, les tables seront nombreuses du côté des enregistrements et cela rend l'utilisation des jointures entre les tables difficile et cause des problèmes de performance.

Quant au dénormalisation, premièrement, il facilite l'écriture des requêtes et pour cela, on trouve que certains mettent toutes les informations dans une seule table. Aussi, la rapidité d'extraire les données depuis la base parce qu'il n'a pas besoins d'utiliser les jointures.

Le dénormalisation a aussi des points faibles surtout les redondances non voulues, qu'on a déjà mentionnées, qui influence négativement sur l'espace occupée par la base de données sur le disque dur, et à ce moment, commence les problèmes.

Alors, la base BIGEMP1 est plus optimisé que la base BIGEMP, mais, il y a une autre technique pour améliorer la base BIGEMP1 c'est le partitionnement de grande table a des petites tables pour faciliter la recherche. Au lieu de parcourir complètement tous les éléments de la grande table, la recherche se limite à une table plus petite, comme le montrent les résultats du Tableau 3.4.

Enfin, nous pouvons dire qu'une compréhension complète de tous les aspects de la conception de la base de données, ainsi qu'une vision future, sont des facteurs clés qui aident à choisir le meilleur pour construire une base plus cohérente.

5.2. Résultats des différentes requêtes

Pour être plus clair, cette section est le plus vaste par rapport les deux autres sections, celle-ci est basée sur ce qu'on a vu dans le chapitre précédent.

Comme d'habitude, on va effectuer de nombreux tests dans cette section, principalement, ces tests sont basés sur l'exécution de la même requête mais avec différentes façons d'écriture et basée sur l'utilisation des méthodes qu'on a vue précédemment.

5.2.1. Statistiques sur les données

Requêtes à exécuter : sélectionner juste les attributs qui nous intéresse depuis les tables de la deuxième base du test (LIVRE, CMD_LIGNE, CMD, CLIENT) si le montant inférieure à 1000 et Ordonner par les pays (Voir Annexe A3).

Présentation des résultats obtenus

	Avec Statistiques	Sans Statistiques
Temps Ecoulé (s)	66.70	70.57
Coût CPU+(I/O)	153164	366807
RAM (Mo)	879.25	2513.67

Tableau 3. 6 Résultats obtenus par l'utilisation des statistiques

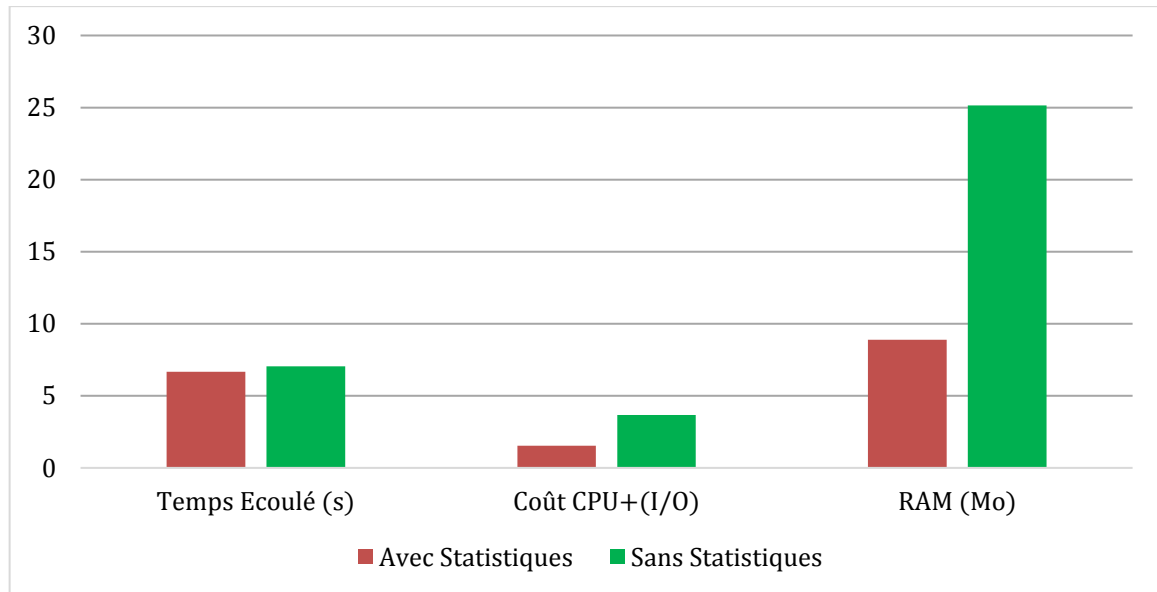


Figure 3. 17 Résultats (Avec/Sans) l'utilisation des Statistiques

5.2.2. Indexes & Hints

Requêtes à exécuter : sélectionner tous les attributs triés par le nom de la table BIGEMP1 si le nom égale « JONES » le salaire inférieur à 6000 (Voir Annexe A3).

Présentation des résultats obtenus

	Avant	Après
Temps Ecoulé (s)	2.66	1.56
Coût CPU+(I/O)	7416	833
RAM (Mo)	24	24

Tableau 3. 7 Résultats obtenus par l'utilisation des indexes & Hints

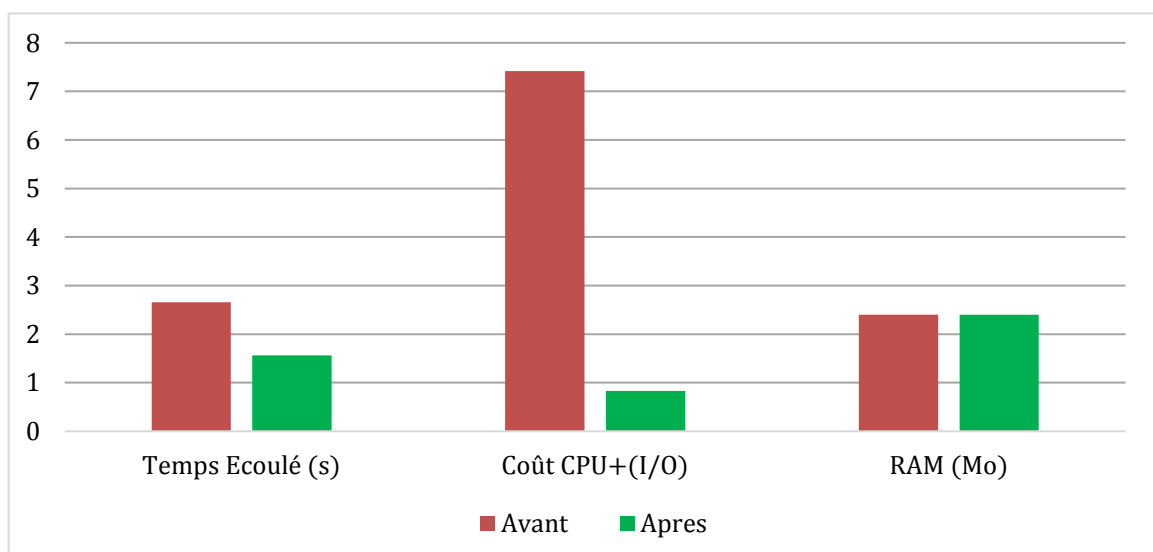


Figure 3. 18 Résultats de l'utilisation des indexes & Hints

5.2.3. Réécriture des requêtes

Requête : Extraire les données de la table livres si le montant d'achat supérieur à 100 et inférieur à 9000 avec différentes façons d'écritures suivantes (Voir Annexe A3) :

Facon 1 : Avec l'utilisation des jointures.

Facon 2 : Avec l'utilisation du sous-requête (IN).

Présentation des résultats obtenus

	Facon 1	Facon 2
Temps Ecoulé (s)	1,49	2,2
Coût CPU+(I/O)	6868	6868
RAM (Mo)	14,04	14,04

Tableau 3. 8 Résultats obtenus par Réécriture des requêtes

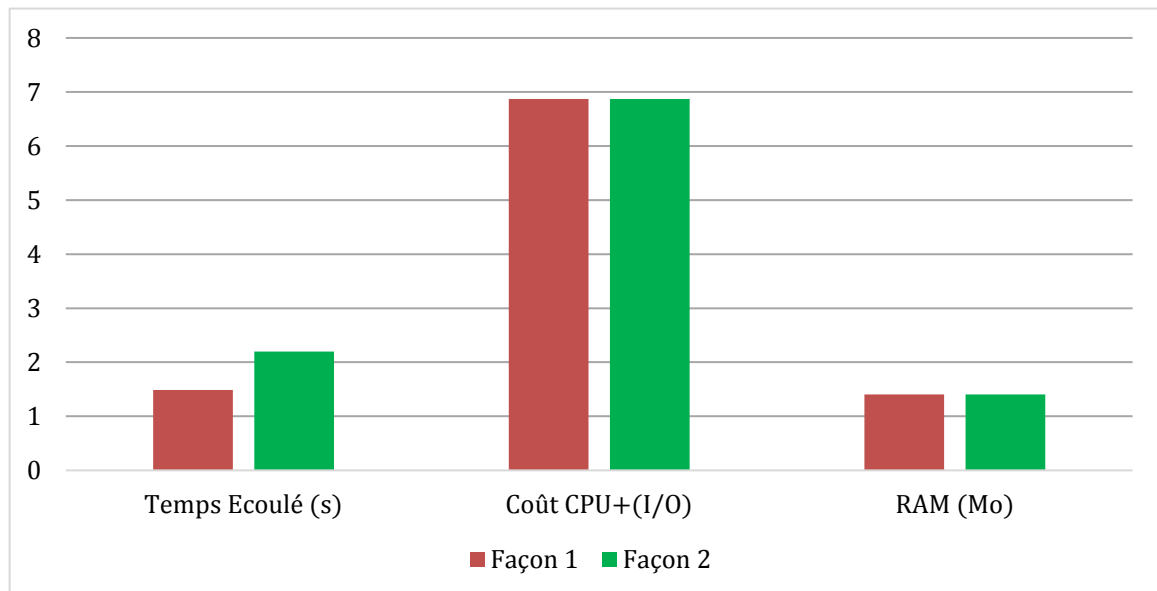


Figure 3. 19 Résultats obtenus par Réécriture des requêtes

5.2.4. Analyse des résultats et Diagnostic

Les tests de cette section étaient construits sur la base d'examiner tous les mécanismes afin de savoir l'importance de chacune.

En conséquence, on trouve une importance impressionnée quand on utilise et quand on n'utilise pas ces mécanismes qu'on doit utiliser pour construire une base de données, de collecter des statistiques sur les tableaux de la base, et cela afin d'atteindre un seul but "la meilleure performance".

5.3. Résultats des différentes infrastructures matérielles

Pour réaliser les tests de cette partie, nous avons créé différentes infrastructures matérielles à l'aide du VMware Workstation avec les caractéristiques suivantes :

- **PC1** : Processeur CPU : Intel i3 @ 4 Core 1.70 GHz.
Mémoire installée (RAM) : 3 Go.
Système d'exploitation : Windows 7 64 bits.
- **PC2** : Processeur CPU : Intel i3 @ 4 Core 1.70 GHz.
Mémoire installée (RAM) : 3 Go.
Système d'exploitation : Windows 7 32 bits.

5.3.1. Requête de test

Pour les tests de cette section on a sélectionné depuis les tables de la deuxième base du test (LIVRE, CMD_LIGNE, CMD, CLIENT) juste les attributs qui nous intéressent si le montant inférieure à 1000 et Ordonner par les pays (Voir Annexe A3).

5.3.2. Présentation des résultats obtenus

	Différents Système d'Exploitation	
	PC 1 - Windows 64bits -	PC 2 - Windows 32bits -
Temps Ecoulé (s)	46.31	47.90
Coût CPU+(I/O)	51290	51294
RAM (Mo)	164.75	164.75

Tableau 3. 9 Résultats obtenus par les différentes SE

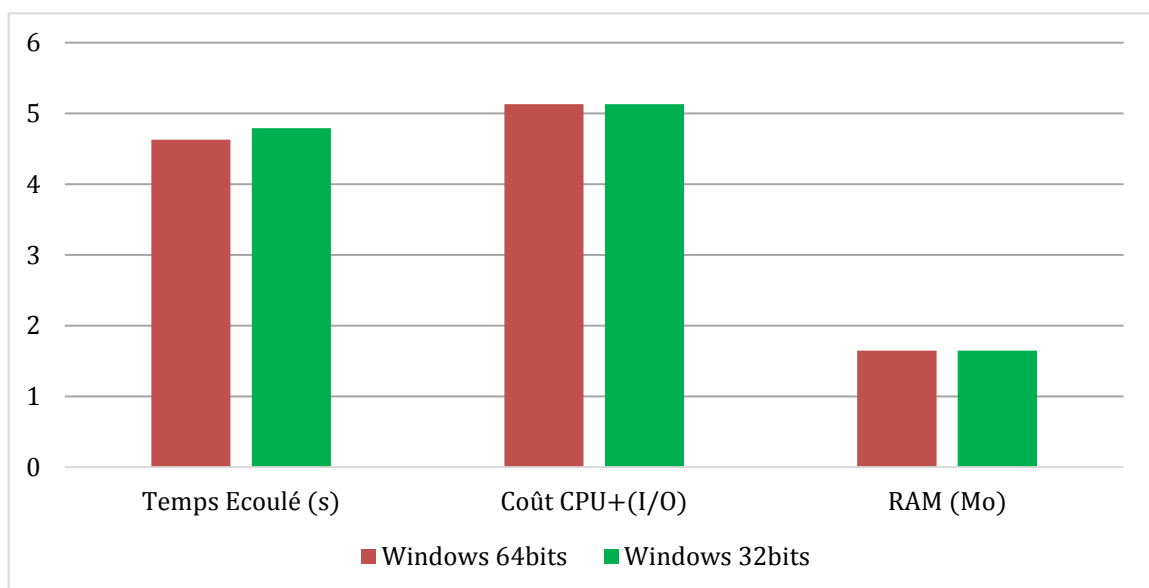


Figure 3. 20 Résultats obtenus par les différentes SE

	Différents Capacité de RAM			
	RAM 1Go SGA 252 Mo	RAM 2Go SGA 504 Mo	RAM 4Go SGA 1004 Mo	RAM 6Go SGA 1520 Mo
Temps Ecoulé (s)	87	72.61	54.88	54.2
Coût CPU+(I/O)	403615	303615	303615	303615
RAM (Mo)	2913.36	2913.36	2913.36	2913.36
Nbr de Lecture sur Le Disque	50478	44084	33295	33295

Tableau 3. 10 Résultats obtenus par les différentes capacités de RAM

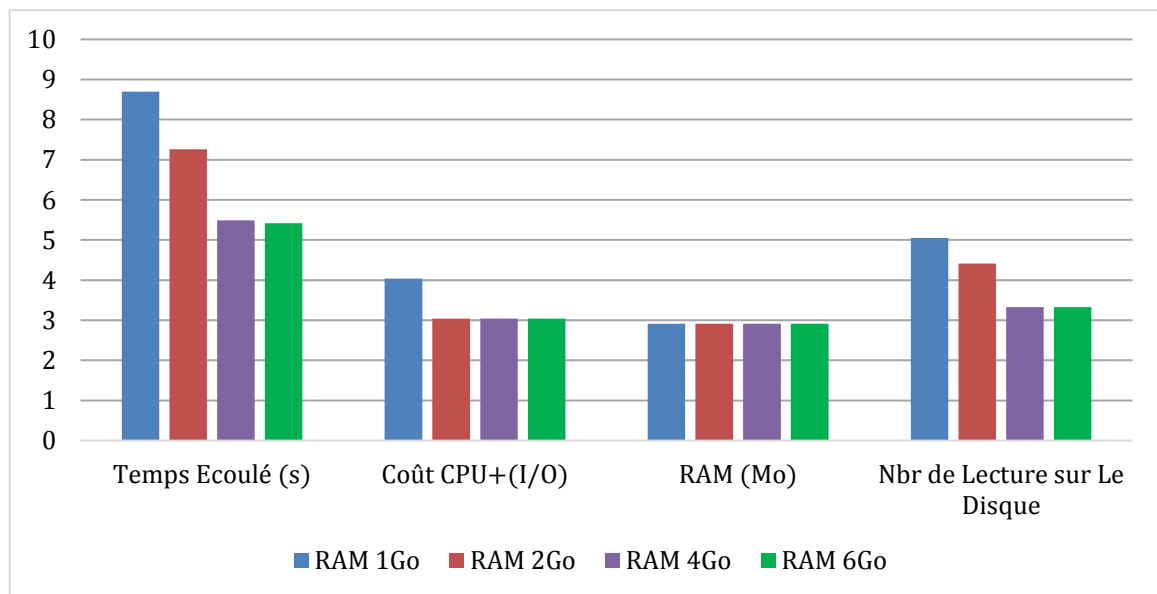


Figure 3. 21 Résultats obtenus par les différentes capacités de RAM

Le test que nous allons maintenant présenter a pour but de mettre en évidence le nombre de noyaux de processeur, et pour ce test nous avons créé un nouveau table AMD contient 25 millions de ligne à l'aide d'utilisation d'un code java. (Voir Figure 3.22)

Voilà la requête de test :

```
Select count (*) From
(Select amd_id, Sum(montant*montant) From amd Group by amd_id);
```

```

int x=0;
String content;
        content = "set feedback off\n" +
"set define off\n" +
"prompt Creating AMD...\n" +
"create table amd\n" +
"(\n" +
" amd_id NUMBER(10) not null,\n" +
" ENAME VARCHAR2(20),\n" +
" MONTANT NUMBER(10)\n" +
"); \n";
        bw.write(content);
for (int i = 1; i < 25000000; i++) {
content = "insert into amd (amd_id, ENAME, MONTANT)\n" +
"values ("+(i)+", 'SMITH'+i+'', "+(50000+i)+"); \n";
if(i%1000 == 0){
        x++;
content = "commit;\n" +
"prompt "+(x*1000)+" records loaded \n";
        bw.write(content);
        content = "set feedback on\n" +
"set define on\n" +
"prompt Done.";
        bw.write(content);
        bw.close();
}
}
    
```

Figure 3. 22 Code java de la création du script SQL

	CPU - Intel i3 @ 4 Core -		CPU - Intel i3 @ 1 Core -	
	Parallel	Non-parallel	Parallel	Non-parallel
Temps Ecoulé (s)	63.55	97.59	117.20	117.42
Coût CPU+(I/O)	32934	119259	32934	119259
RAM (Mo)	4008	1336	4008	1336

Tableau 3. 11 Résultats obtenus par des différentes infrastructures matérielles

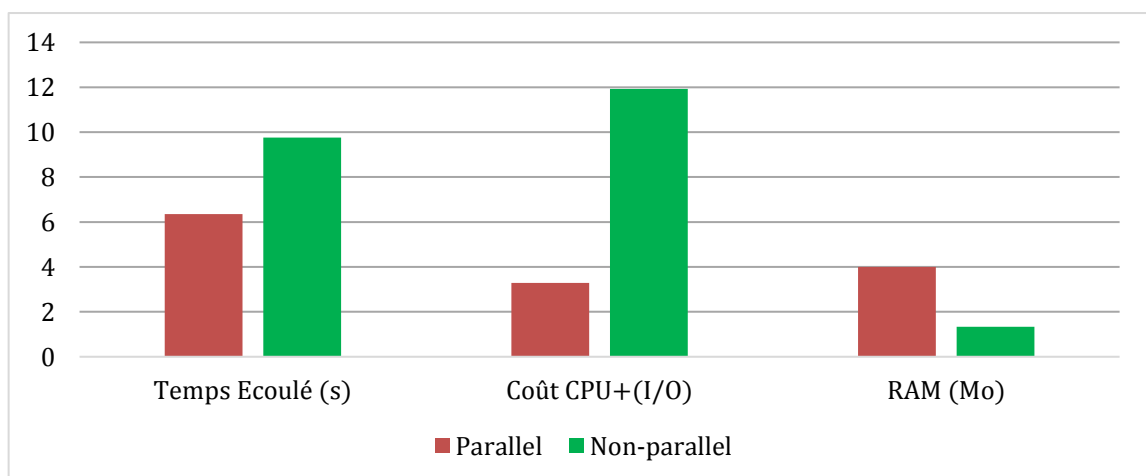


Figure 3. 23 Résultats obtenus par CPU - Intel i3 @ 4 Core -

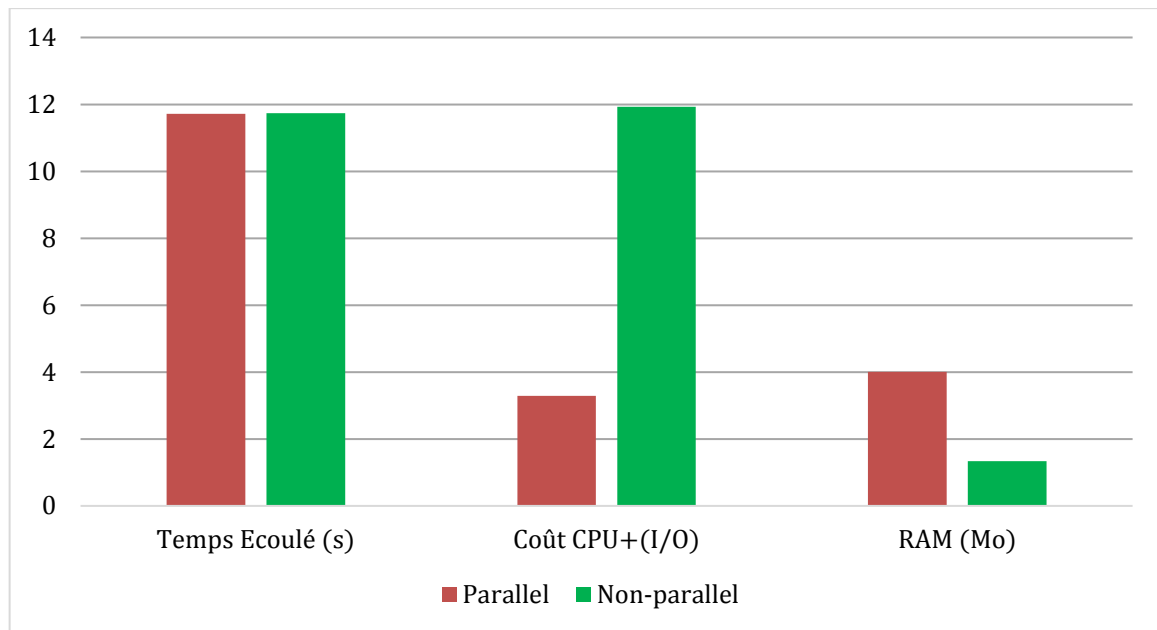


Figure 3. 24 Résultats obtenus par CPU - Intel i3 @ 1 Core -

5.3.3. Analyse et diagnostique des résultats

Les ordinateurs connaissent une amélioration significative de leur efficacité ce qui influence directement sur les performances des programmes sur lesquelles ils fonctionnent, pour cela, la question qui se pose : Quel est le facteur clé qui contribue à améliorer les performances ?

Ça nous amène à plusieurs possibilités qui peuvent jouer le rôle d'un facteur clé, premièrement la cause peut être le processeur (CPU) car ce dernier se considère plus fort relativement à l'augmentation de ces noyaux, une autre possibilité, c'est que la RAM a une relation dans cette amélioration, si on a une grande capacité de RAM, ça veut dire le volume d'information à traiter sera plus grand, peut être aussi les systèmes d'exploitation, on doit savoir que les systèmes d'exploitations 64bits soient plus performants que les systèmes d'exploitations 32bits.

Pour répondre à cette question les résultats des ordinateurs doivent être comparés les uns aux autres.

Lorsque on compare les résultats du Tableau 3.8 par rapport aux systèmes d'exploitation, il n'y a pas de différence impressionnante entre 64bits et 32bits, autre fois les systèmes 32bits ne sont pas recommandés car ils ne peuvent pas utiliser un RAM supérieur de 3 Go.

Les résultats du Tableau 3.9 montre que l'augmentation de la capacité du RAM n'est pas nécessaire pour améliorer les performances et le temps de réponse, Cependant la RAM est liée

directement à la mémoire d'instance d'Oracle (SGA & PGA), pour que celle-ci soit capable d'assurer de bonne performance, elle doit être de capacité suffisante.

Les résultats du Tableau 3.10 confirme la notion connue qui déclare que l'améliorations du processeur (CPU) a toujours un impact positif sur les performances, d'une part l'augmentation de la vitesse CPU influence positivement au temps de réponse, ce qui concerne le nombre de noyaux n'a pas d'effet sauf s'il existe des exécutions en parallèles.

6. Conclusion

À la fin de ce chapitre, nous pouvons souligner l'importance d'utilisation des mécanismes d'optimisation à travers nos tests.

En effet, nos résultats expérimentaux obtenus à tous les niveaux nous montrent la différence entre l'utilisation et la non-utilisation des mécanismes d'optimisation que nous avons vu dans le chapitre précédent dont le but est d'améliorer les performances et diminuer les coûts de consommation des composants serveur, en commençant par la conception de la base et nous avons également abordé les mécanismes d'amélioration des requêtes , enfin nous avons conclu ce chapitre par l'amélioration des matériels.

CONCLUSION GENERALE

Ce mémoire avait pour ambition de mettre en évidence la nécessité d'utiliser les outils d'optimisations des bases de données, après, notre travail que nous avons mené autour du sujet de ce mémoire nous pouvons sans hésitation affirmer que ce dernier est une partie intégrante de la conception de bases de données et de la rédaction de requêtes.

L'accroissement continu de la volumétrie des données et la complexité des requêtes rend difficile leur traitement, et afin de remédier au problème d'énorme quantité des données il est de plus en plus important de se reposer sur d'autres solutions et outils entre autres les mécanismes d'optimisation des bases de données qui permettent le progrès de performances des requêtes en offrant un accès rapide aux données.

Il a fallu dans un premier temps définir les notions principales sur les bases de données et également donner un aperçu général sur l'environnement de travail Oracle, pour un objectif consistait à exploiter les solutions dans trois grands axes d'optimisation, à savoir le schéma conceptuel, l'infrastructure matérielle et les requêtes, qui permettent d'optimiser les bases de données sous oracle et évolué un accès rapide aux enregistrements de la base.

Avant de conclure, on peut affirmer que les mécanismes d'optimisation de données sous oracle ont montré son impact positif sur le temps d'exécution des requêtes ainsi que la réduction du coût de traitement du côté matériels.

En conclusion, on peut assurer qu'une base de données performante commence par une bonne conception dans un environnement matériel adéquat en plus de faisant les bons choix et en maîtrisant les outils d'administration à savoir le SGBD oracle.

Ce travail de mémoire est basé sur des tests à des bases de données locales, mais dans cette nouvelle perspective et pour le but de compléter cette étude, il serait pertinent de procéder à une étude sur une base de données réparties sur des serveurs distincts avec de nombreux utilisateurs qui pouvant accéder à cette base.

BIBLIOGRAPHIE

- [1] Aide et support Microsoft, Principes de base de la normalisation des bases de données, 30 août 2017, Consulté sur <https://support.microsoft.com/fr-ca/help/283878/description-of-the-database-normalization-basics>.
- [2] A. MEIER, Introduction pratique aux bases de données relationnelles (Deuxième édition), Springer-Verlag France, (2002-2006).
- [3] Article [En Ligne], Le modèle de bases de données relationnelles, Publié le 09 Mai 2019, Consulté sur <https://www.ionos.fr/digitalguide/hebergement/aspects-techniques/base-de-donnees-relationnelle/>.
- [4] Article [En Ligne], Les statistiques Oracle, Disponible sur <http://www.aide-oracle.net/2009/04/les-statistiques-oracle.html>.
- [5] Article [En Ligne], Définitions d'optimisation, Publié le 12 Sept 2012, Consulté sur <http://lesdefinitions.fr/optimisation>.
- [6] B. ESPINASSE, Optimisation et performance dans les bases de données relationnelles, Support du Cours Universitaire de Marseille, Mars 2013.
- [7] Comment Ça Marche - Communauté informatique, <https://web.maths.unsw.edu.au/~lafaye/CCM/>, Consulté en avril 2019.
- [8] Ch. TEDAJIO SUKI, Architecture Oracle Database, Publié le 08/07/2018, Consulté sur <https://www.supinfo.com/articles/single/7113-architecture-oracle-database>.
- [9] D. VODISLAV, Bases de données avancées -Évaluation et optimisation des requêtes, Cours BDA Master Informatique Université de Cergy-Pontoise.
- [10] E. JOURDAIN, Oracle 11g - ADMINISTRATION, Support du Cours.
- [11] F. DE MARCHI, Administration des bases de données sous Oracle, Université de Lyon.
- [12] G. GARDARIN, Bases de données, EDITIONS EYROLLES, Paris-France 2003.
- [13] L. Navarro, Optimisation base de données, Pearson Education France 2010.
- [14] M. WINAND, SQL : Au cœur des performances (Edition Français), Vienne Autriche, 03 Mars 2013.
- [15] N. TOUNSI, INTRODUCTION AUX BASES DE DONNÉES, Cours à l'usage des premières années informatique.

- [16] ORACLE Help Center, Database Concepts, Consulté sur https://docs.oracle.com/cd/E11882_01/server.112/e40540/indexiot.htm#.
- [17] OraDev, <http://www.oradev.com/hints.html>, Consulté en avril 2019.
- [18] Sam R. Alapati, D. Kuhn, et B. Padfeild, Oracle Database 11g Performance Tuning Recipes, Apress ®.
- [19] S. CROZAT, Introduction aux bases de données relationnelles, 30 janvier 2018.
- [20] Site Developpez.Com, <https://www.developpez.com/>, Consulté en Mars 2019.
- [21] Site Comment Ça Marche CCM, <https://www.commentcamarche.net/>, Consulté en Mars 2019.
- [22] Site Official d'ORACLE, <https://www.oracle.com/fr/database/pourquoi-utiliser-base-de-donnees.html>, Consulté en Mars 2019.
- [23] Site informatique-bureautique.com, Formation Access 2013, développer une application, Consulté sur <https://www.informatique-bureautique.com>.

ANNEXES

ANNEXE A1 : INSTALLATION DU SGBD ORACLE

Le SGBD oracle fourni par Oracle Corporation. C'est l'un des SGBD les plus populaire au niveau mondial.

Sous Windows, l'installation d'oracle database doit se faire par l'administrateur du système et exécuté du fichier **setup.exe** d'oracle database 11g, par exemple dans mon PC le chemin du ce dernier et comme suit : « **D:\database\setup.exe** » pour démarrer oracle Universal installer et après suivez ces étapes :

- **Configuration des mises à jour de sécurité** : Dans cette étape Si vous souhaitez recevoir des mises à jour de sécurité, entrez votre email et vous recevrez toutes les dernières mises à jour.

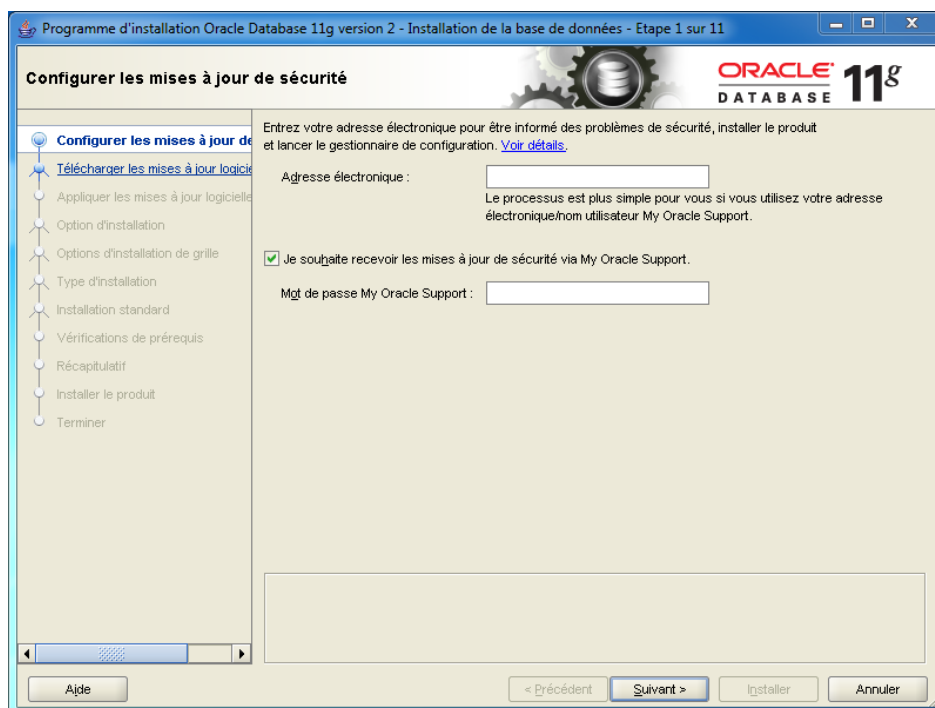


Figure A1. 1 Installation de la base de données- Etape 1

- **Télécharger les mises à jour logicielles** : cette étape permet de télécharger les dernières mises à jour.

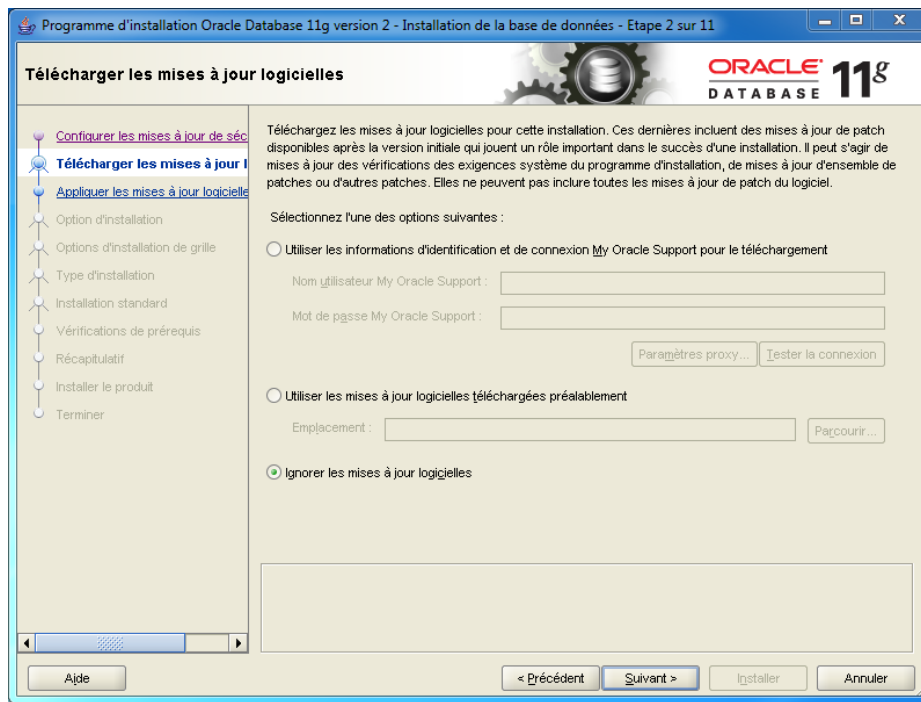


Figure A1. 2 Installation de la base de données- Etape 2

- **Option d'installation :** cette étape contient trois choix : 1) créer et configurer une base de données, 2) installer le logiciel de la base de données uniquement, 3) mettre à niveau une base de données, Choisissez ce que vous voulez et aller au l'étape suivante.

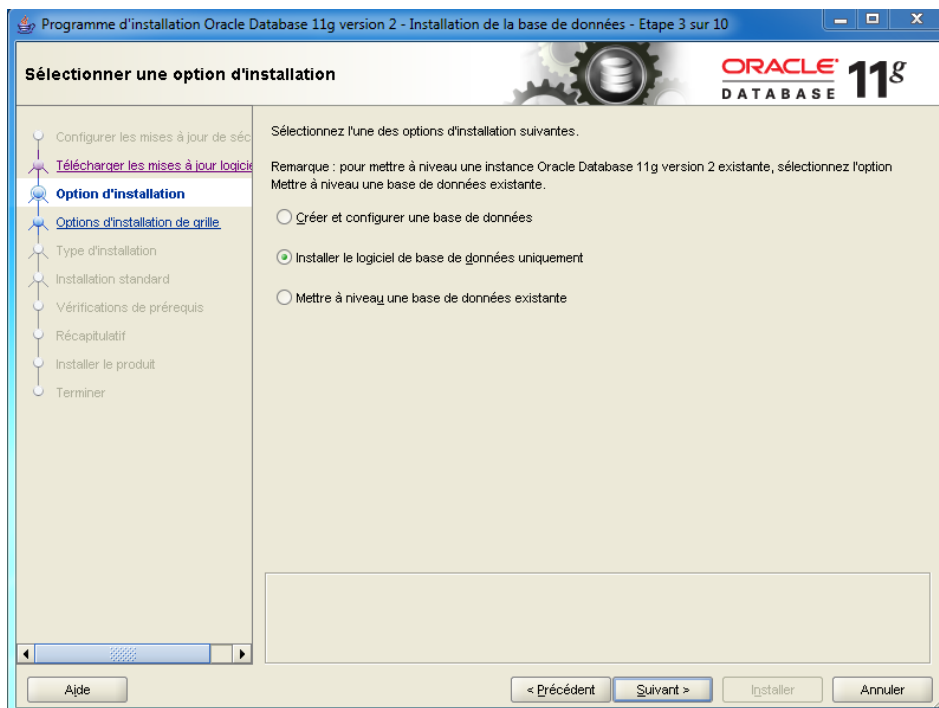


Figure A1. 3 Installation de la base de données- Etape 3

- **Option d'installation de grille** : Dans cette étape sélectionner le type d'installation de la base de données parmi les choix afficher sur l'écran.

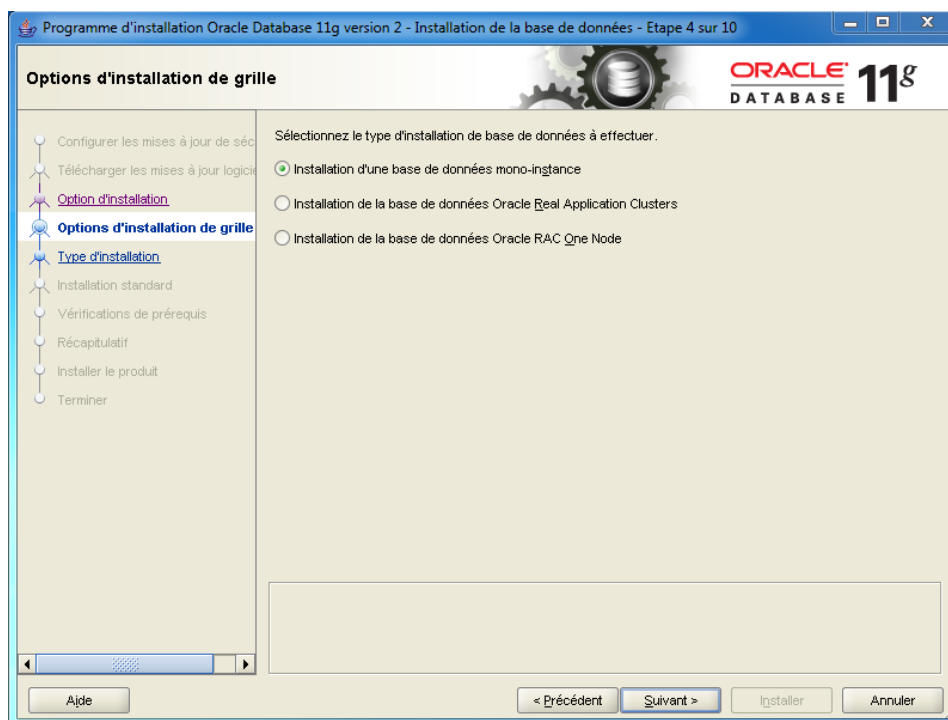


Figure A1. 4 Installation de la base de données- Etape 4

- **Langues du produit** : Dans cette étape choisissez la langue que vous pouvez facilement comprendre parmi des dizaines des langues disponibles.

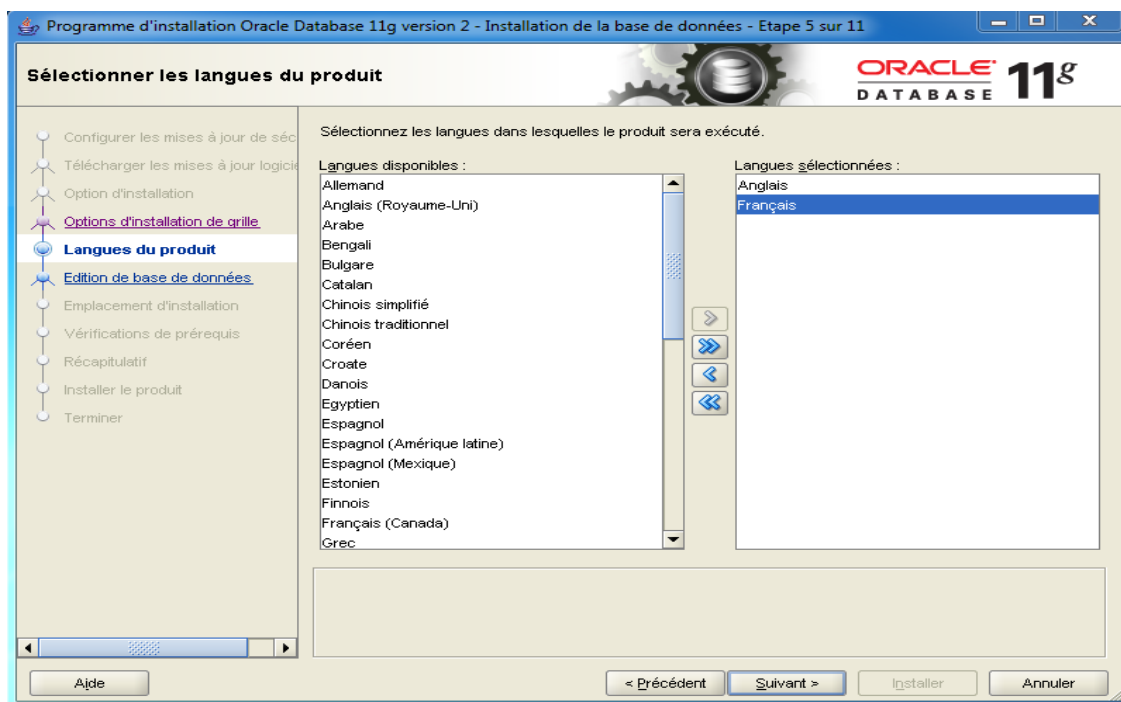


Figure A1. 5 Installation de la base de données- Etape 5

- **Edition de base de données :** parmi les quatre choix disponibles nous avons installé l'édition entreprise dans ce projet car elle offrant un environnement de travail très vaste.

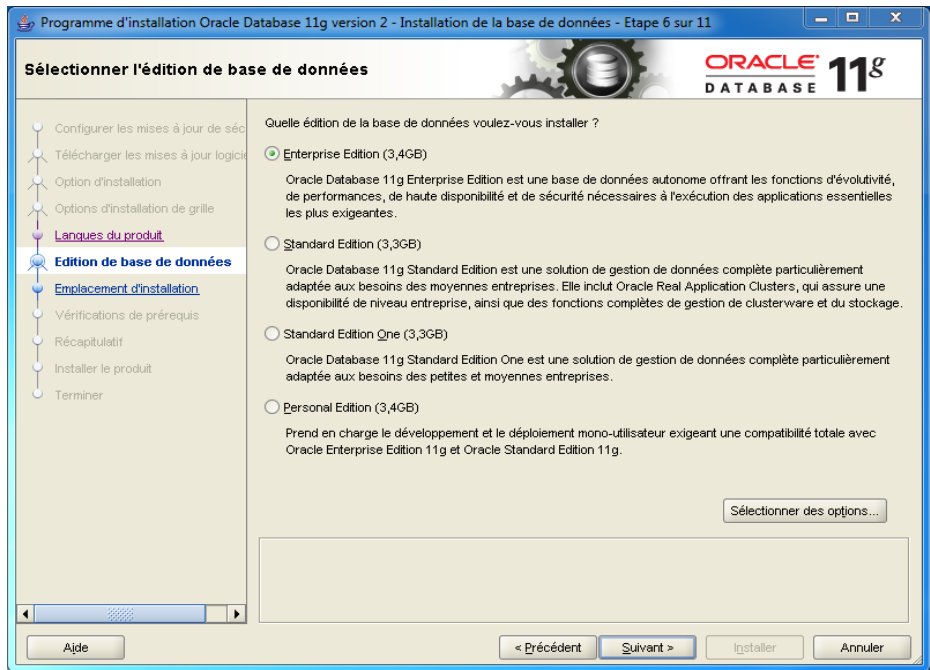


Figure A1. 6 Installation de la base de données- Etape 6

- **Emplacement de l'installation :** cette étape vous permet de choisir l'emplacement de l'installation du logiciel.

Par défaut : D:\app\UTILISATEUR\product\11.2.0\dbhome_2

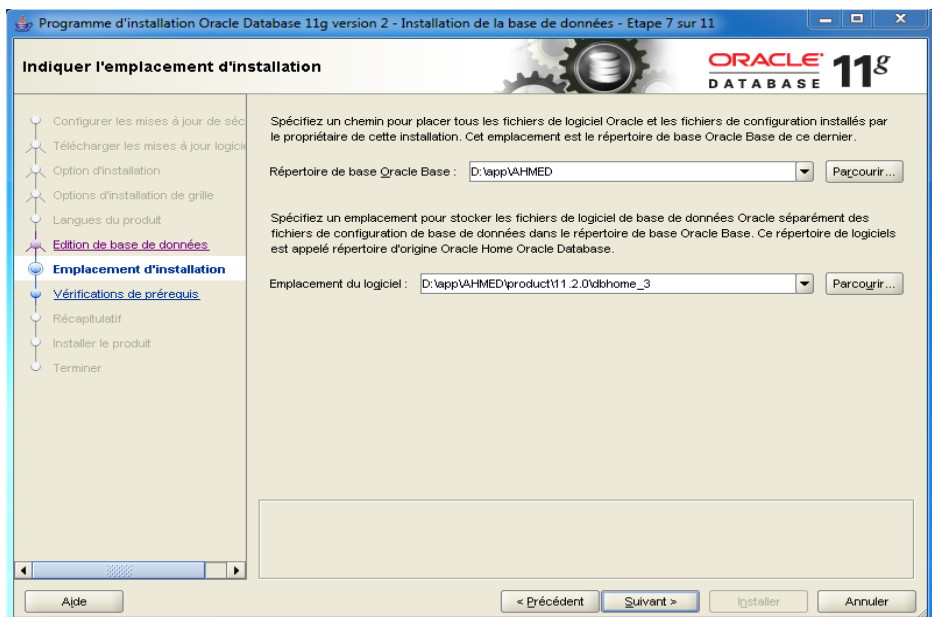


Figure A1. 7 Installation de la base de données- Etape 7

- **Vérification de prérequis** : cette étape vérifie les exigences principales de l'installation du logiciel à savoir l'espace libre sur le disque et la mémoire physique ... etc.

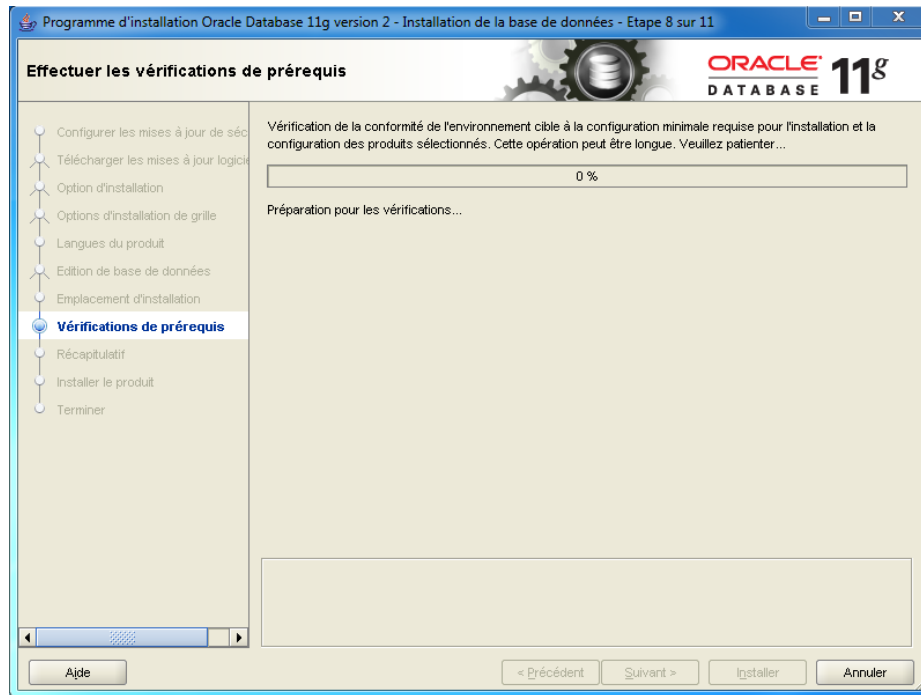


Figure A1. 8 Installation de la base de données- Etape 8

- **Installation du produit** : Dans cette étape, vous devez attendre la fin du téléchargement du logiciel à 100%, puis bienvenue à Oracle 11g.

ANNEXE A2 : CREATION DE LA BASE DE DONNEES ET LES TABLES

Dans cette section nous allons voir comment créer une base de données et remplir ses tables sous Oracle.

A2.1. La base de données

Pour créer une nouvelle base de données il faut accéder à l'Assistant Configuration de base de données sur le chemin suivant :

Démarrer >> Programmes >> Oracle - OraDb11g home1 >> Outils de Configuration et de migration >> Assistant Configuration de base de données.

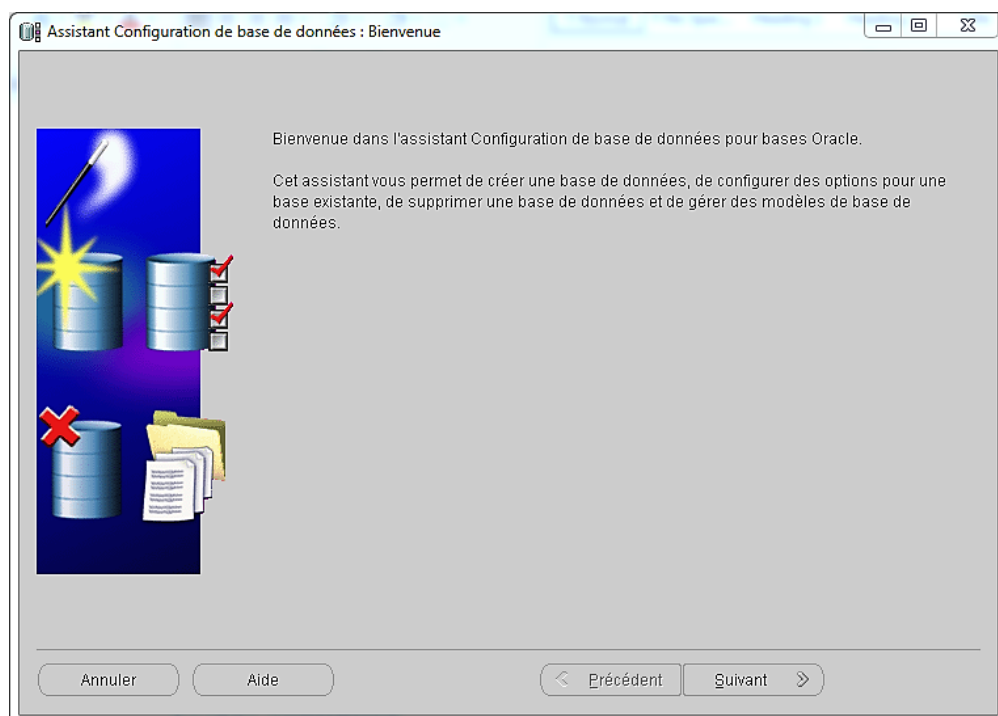


Figure A2. 1 Accueil de l'Assistant Configuration de base de données

Juste après une clique dans l'étape suivante l'Assistant Configuration de base de données affiche les options comme suit :

- **Créer une base de données** : pour créer une nouvelle base de données.
- **Configurer les options de base de données** : pour configurer et modifier les paramètres d'une base de données existante.
- **Supprimer une base de données** : pour supprimer base de données existante.
- **Gérer les modèles** : pour ajouter ou bien supprimer un modèle de base de données.

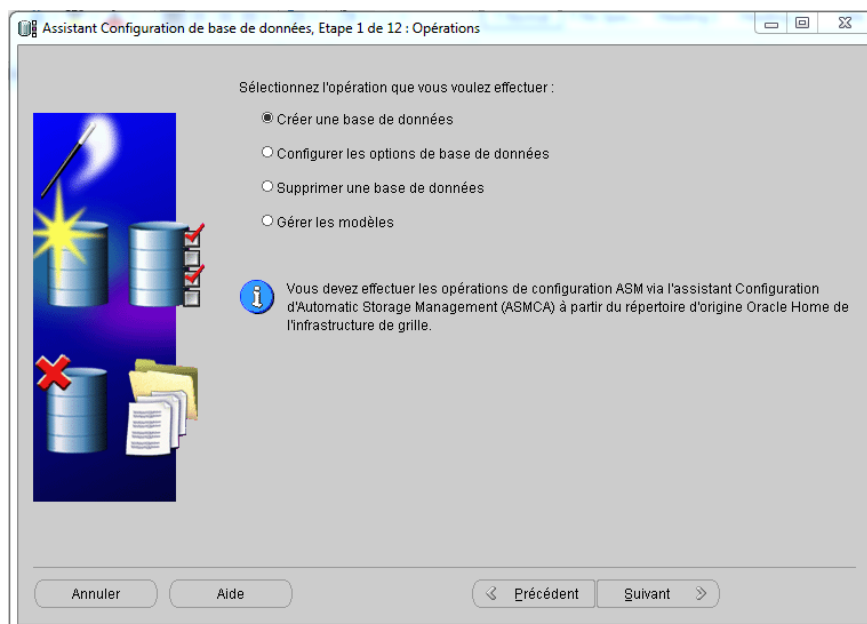


Figure A2. 2 Les options de l'Assistant Configuration de base de données

Et bien sûr parmi ces options nous allons choisir la création d'une nouvelle base Oracle 11g qui passe par 12 étapes. Nous laissons les valeurs par défaut durant toutes les étapes sauf sur la 4^{ème} étape décocher l'option entreprise manager.

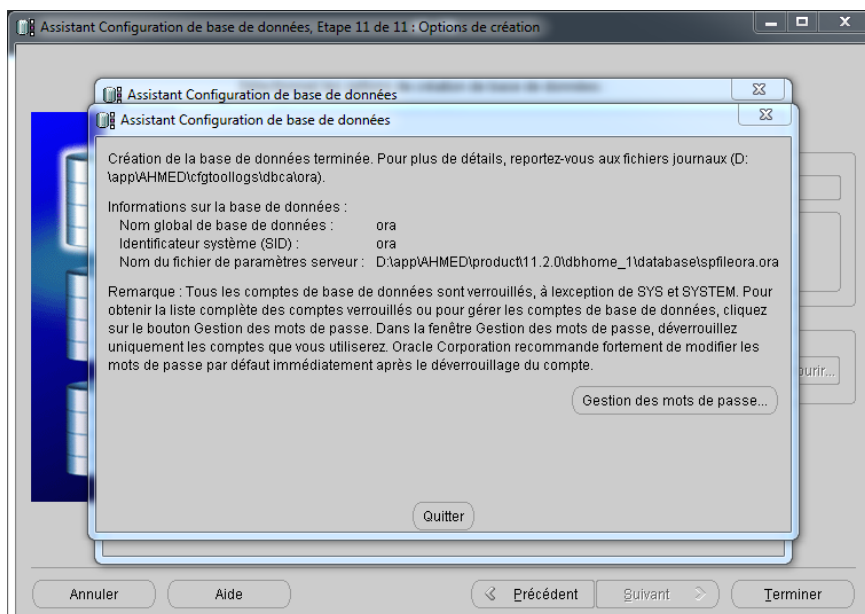


Figure A2. 3 Fin de la création de base de données

II.2. Les tables

Au fur et à mesure de l'augmentation des enregistrements dans la base de données, des problèmes commencent à apparaître, et pour voir une différence significative des améliorations, il faut utiliser des bases de données avec des tables de grande taille.

Avant de créer les tables il est obligé de connecter à la base des données par l'utilisateur «SYSTEM» et le mot de passe avec l'outil **SQL PLUS** dans le menu démarrer, il est accessible sur le chemin suivant :

Démarrer >> Programmes >> Oracle – OraDb11g home1 >> Développement d'application >> SQL Plus

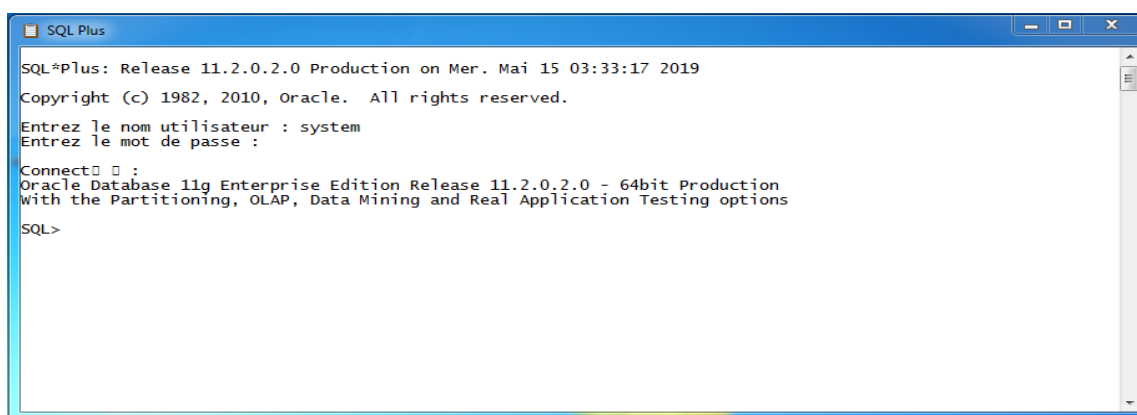


Figure A2. 4 connexion à la base de données avec SQL PLUS

Après la connexion nous allons maintenant à l'étape de création des tables et remplis les enregistrements dans ce dernier, pour cela, nous avons utilisé les scripts suivants :

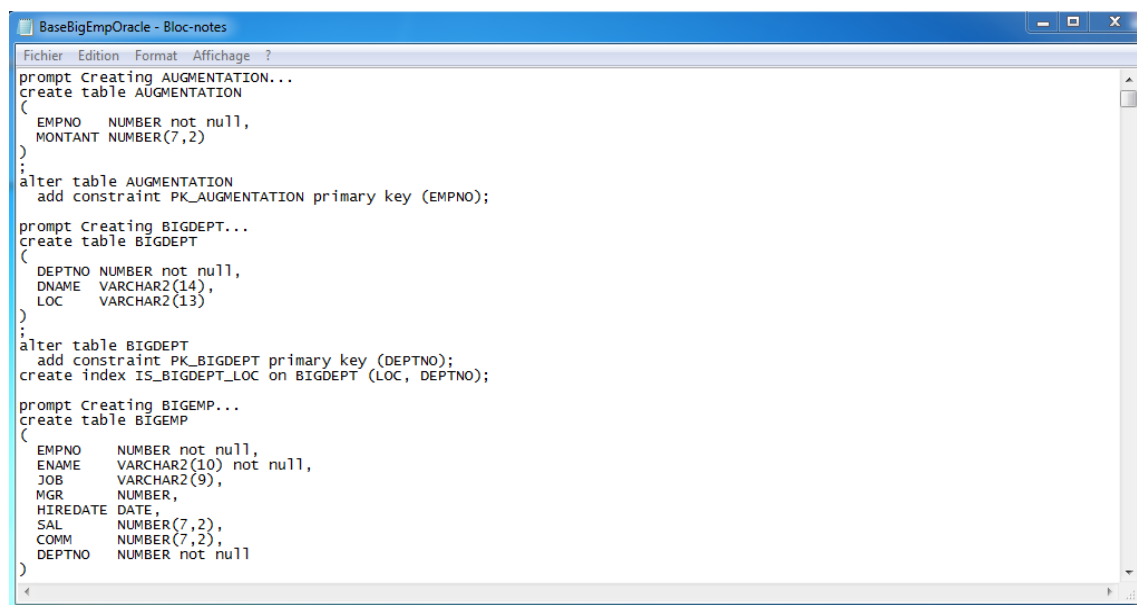


Figure A2. 5 Script de la base BIGEMP

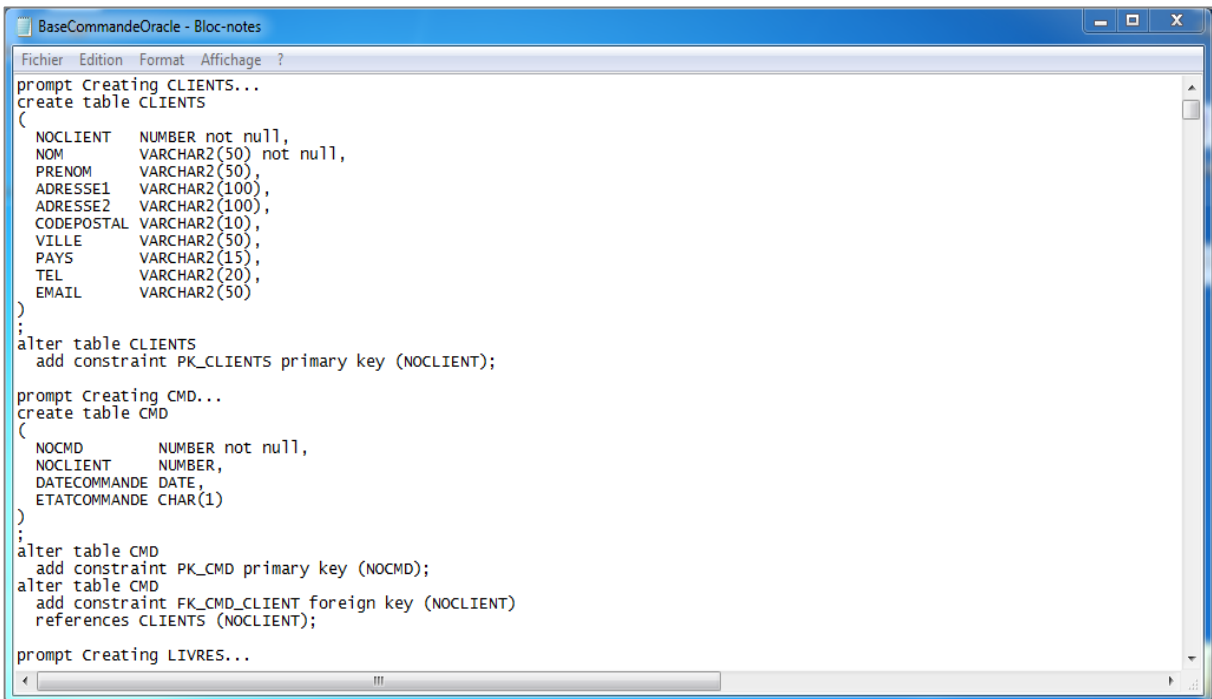
```

prompt Creating BIGEMP1...
create table BIGEMP1
(
  EMPNO    NUMBER not null,
  ENAME    VARCHAR2(10) not null,
  JOB      VARCHAR2(9),
  MGR      NUMBER,
  MONTANT  NUMBER(7,2),
  HIREDATE DATE,
  SAL      NUMBER(7,2),
  COMM     NUMBER(7,2),
  DEPTNO   NUMBER not null,
  DNAME    VARCHAR2(14),
  LOC      VARCHAR2(13)
)
;
alter table BIGEMP1
  add constraint PK_BIGEMP1 primary key (EMPNO);
alter table BIGEMP1
  add constraint FK_BIGEMP1_EMP foreign key (MGR)
  references BIGEMP1 (EMPNO);
alter table BIGEMP1
  add check (ename=UPPER(ename));
alter table BIGEMP1
  add check (SAL > 500 );

prompt Disabling foreign key constraints for BIGEMP1...
alter table BIGEMP1 disable constraint FK_BIGEMP1_EMP;
prompt Loading BIGEMP1...
insert into BIGEMP1 (EMPNO, ENAME, JOB, MGR, MONTANT, HIREDATE, SAL, COMM, DEPTNO, DNAME, LOC)
values (7369, 'SMITH', 'CLERK', 7902, 1480, to_date('17-12-1980', 'dd-mm-yyyy'), 800, null, 20, 'OPERATIONS', 'BOSTON');
insert into BIGEMP1 (EMPNO, ENAME, JOB, MGR, MONTANT, HIREDATE, SAL, COMM, DEPTNO, DNAME, LOC)
values (7499, 'ALLEN', 'SALESMAN', 7698, 3080, to_date('20-02-1981', 'dd-mm-yyyy'), 1600, 300, 30, 'ACCOUNTING', 'NEW YORK');
insert into BIGEMP1 (EMPNO, ENAME, JOB, MGR, MONTANT, HIREDATE, SAL, COMM, DEPTNO, DNAME, LOC)
values (7521, 'WARD', 'SALESMAN', 7698, 2450, to_date('22-02-1981', 'dd-mm-yyyy'), 1250, 500, 30, 'RESEARCH', 'DALLAS');
insert into BIGEMP1 (EMPNO, ENAME, JOB, MGR, MONTANT, HIREDATE, SAL, COMM, DEPTNO, DNAME, LOC)
values (7566, 'JONES', 'MANAGER', 7839, 3000, to_date('02-04-1981', 'dd-mm-yyyy'), 2975, null, 2020, 'SALES', 'CHICAGO');

```

Figure A2. 6 Script de la base BIGEMP1



```

BaseCommandeOracle - Bloc-notes
Fichier Edition Format Affichage ?
prompt Creating CLIENTS...
create table CLIENTS
(
  NOCLIENT NUMBER not null,
  NOM       VARCHAR2(50) not null,
  PRENOM   VARCHAR2(50),
  ADRESSE1 VARCHAR2(100),
  ADRESSE2 VARCHAR2(100),
  CODEPOSTAL VARCHAR2(10),
  VILLE    VARCHAR2(50),
  PAYS     VARCHAR2(15),
  TEL      VARCHAR2(20),
  EMAIL    VARCHAR2(50)
)
;
alter table CLIENTS
  add constraint PK_CLIENTS primary key (NOCLIENT);

prompt Creating CMD...
create table CMD
(
  NOCMD      NUMBER not null,
  NOCLIENT  NUMBER,
  DATECOMMANDE DATE,
  ETATCOMMANDE CHAR(1)
)
;
alter table CMD
  add constraint PK_CMD primary key (NOCMD);
alter table CMD
  add constraint FK_CMD_CLIENT foreign key (NOCLIENT)
  references CLIENTS (NOCLIENT);

prompt Creating LIVRES...

```

Figure A2. 7 Script de la base BigCommande

Pour exécuter ces scripts tape sur le **SQL PLUS** un arobase avec le chemin où se trouve le fichier SQL, Comme suit :

@ le chemin\le nom du fichier.sql

Exemple :

SQL> @ D:\data\ BaseBigEmpOracle.sql

ANNEXE A3 : REQUETES UTILISES DANS LA PARTIE D'EXPERIMENTATION

Dans ce terrain nous allons citer les requêtes exécuter pour obtenir les résultats expérimentaux dans le chapitre 3.

A3.1. Requêtes pour : différentes structures

A3.1.1. Structure 1

Requête 1 :

```
Select * From augmentation a, bigemp b, bigdept d  
Where a.empno = b.empno And b.deptno = d.deptno  
And montant > 1500 order by ename ;
```

Requête 2 :

```
Select ename, job, sal From augmentation a, bigemp b, bigdept d  
Where a.empno = b.empno And b.deptno = d.deptno  
And montant > 1500 order by ename ;
```

A3.1.2. Structure 2

Requête 1 : **Select** * **From** bigemp1 **Where** montant > 1500 **order by** ename ;

Requête 2 : **Select** ename, job, sal **From** bigemp1 **Where** montant > 1500
order by ename ;

A3.1.3. Fragmentation horizontale

```
Select * From dal Where Montant < 1500 ;
```

A3.1.4. Fragmentation verticale

```
Select * From tabl Where ename = 'FORD' And loc = 'BOSTON' ;
```

A3.2. Requêtes pour : différentes requêtes

A3.2.1 Statistiques sur les données

```
Select titre, prix, quantite, nom, prenom, pays  
From livres l, cmd_lignes cl, clients c, cmd  
Where l.nolivres = cl.nolivres And cmd.nocmd = cl.nocmd And c.noclient  
= cmd.noclient And montant < 1000 Order by pays;
```

A3.2.2. Indexe & Hints

- Avant

```
Select * From bigemp1 Where ename = 'JONES' And montant < 6000  
order by ename ;
```

- Apres

```
Select /*+ index_ffs (bigempl, is_ename) */ * From bigempl Where  
ename = 'JONES' And montant < 6000 order by ename ;
```

A3.2.3. Réécriture des requêtes

- In

```
Select sum (Montant) From cmd_lignes  
Where nolivre IN (Select nolivre From livres Where auteur = 'King  
Stephen') ;
```

- Jointure

```
Select sum (Montant) From livres l, cmd_lignes cl  
Where l.nolivre = cl.nolivre And auteur = 'King Stephen' ;
```

A3.3. Les requêtes utiliser pour les résultats différentes infrastructures matérielles

Requête à exécuter :

```
Select titre, prix, quantite, nom, prenom, pays  
From livres l, cmd_lignes cl, clients c, cmd  
Where l.nolivre = cl.nolivre And cmd.nocmd = cl.nocmd And c.noclient  
= cmd.noclient And montant <1000 Order by pays;
```

ملخص

إن تطور المؤسسات، بالإضافة إلى سرعة التحويلات الرقمية، يدفع قاعدة البيانات لتسجيل كميات هائلة من المعلومات مع الاستمرار في الزيادة بمرور الوقت، مما يعرضها لخطر تباطؤ الأداء. ولحل هاته المشكلة، يجب على المطور استغلال آليات تحسين البيانات على مختلف المحاور الرئيسية من أجل تحسين أداء أوامر SQL. وعملنا هذا يهدف الى تسليط الضوء على اهمية هاته الآليات، بدءا من تعريف بعض المفاهيم الأساسية حول قاعدة البيانات بالإضافة إلى تقديم لمحة عامة عن نظام إدارة قواعد البيانات أوراكل. بعد ذلك شرحنا أساليب تحسين قاعدة البيانات التي تقلل من وقت الاستجابة وكذلك تكلفة المعالجة. لننتهي بدراسة تجريبية تظهر قيمة آليات التحسين على قاعدة بيانات علانقية في نظام إدارة قواعد البيانات أوراكل 11g .

الكلمات المفتاحية: قاعدة البيانات ، الأداء ، آليات التحسين ، نظام ادارة قواعد البيانات أوراكل 11g

Abstract

Enterprise development, in addition to the speed of digital transformation, drives the database to record huge amounts of information while continuing to increase over time, putting them at risk of slower performance. To solve this problem, the developer must exploit the optimization mechanisms of data on the different major axes, in order to improve the performance of the queries SQL. In our work, we define some criteria notions about database in addition we present a general overview of the Oracle Database Management System. After that we have explained the optimization techniques of database, which reduces the query response time as well as the cost of processing. We end with an experimental study and comparative tests under the oracle DBMS 11g on a relational database that show the value of optimizing databases.

Keywords: database, performances, optimization mechanisms, Oracle 11g DBMS.

Résumé

Le développement des entreprises, en plus la rapidité de la transformation digitale, conduit la base de données pour enregistrer d'énorme quantités des informations tout en continuant à augmenter avec le temps, ce qui les expose à un risque de ralentissement des performances. Pour résoudre ce problème, le développeur doit exploiter les mécanismes d'optimisation de base des données sur les différent axes majeurs, afin d'améliorer les performances d'exécution des requêtes SQL. Dans notre travail, nous définissons quelques notions critères relatif à la base de données et un aperçu général sur le Système de Gestion des Base de Données Oracle. Nous présentons les techniques d'optimisation de base des données, qui permet de réduire le temps de réponse des requêtes ainsi que le coût de traitement. Nous terminons par une étude expérimentale et des tests comparatifs sous le SGBD oracle 11g sur une base de données relationnelle qui montrent l'intérêt de l'optimisation des bases de données.

Mots clés : la base de données, les performances, mécanismes d'optimisation, SGBD oracle 11g.