

Comparatif entre 04 SGBDs
«Oracle, MS SQL Server, MySQL, MS Access»

Table des matières

Introduction générale	1
1 Généralités sur les SGBDs	3
1.1 Introduction	3
1.2 Définitions	4
1.2.1 Qu'es qu'un SGBD ?	4
1.2.2 Des acteurs bien installés	4
1.3 Classification des SGBDs	6
1.3.1 MICROSOFT	6
1.3.1.1 SQL Server	6
1.3.1.2 MS Access	6
1.3.2 Oracle Corporation	7
1.3.2.1 Oracle Database	7
1.3.2.2 Berkeley DB	7
1.3.2.3 HyperionEssBase	7
1.3.2.4 MySQL	7
1.3.3 Firebird SQL Foundation	8
1.3.3.1 Firebird	8
1.3.4 Sybase	8
1.3.4.1 SQL Anywhere	8
1.3.4.2 Adaptive Server Enterprise	9
1.3.5 Apache	9
1.3.5.1 ApacheDerby	9
1.4 Fil d'actualité	9
1.4.1 Qu'est-ce qui provoque le passage à NoSQL ?	9
1.4.2 Les types de base NoSQL	10
1.4.2.1 Paradigme clé / valeur	10
1.4.2.2 Bases documentaires	12
1.4.2.3 Bases orientées colonnes	13

1.4.2.4	Paradigme graphe	14
1.4.3	Panorama des solutions NoSQL existantes	15
1.5	Conclusion	16
2	Présentation des SGBDs étudiés	17
2.1	Introduction	17
2.2	SQL server	17
2.2.1	Définition	17
2.2.2	Avantages	17
2.2.3	Inconvénients	18
2.3	Oracle DB	18
2.3.1	Définition	18
2.3.2	Avantages	19
2.3.3	Inconvénients	19
2.4	MySQL	20
2.4.1	Définition	20
2.4.2	Avantages	20
2.4.3	Inconvénients	20
2.5	MS Access	21
2.5.1	Définition	21
2.5.2	Avantages	21
2.5.3	Inconvénients	21
2.6	Conclusion	22
3	Comparatif entre les 04 SGBDs (Oracle, SQL Server, MySQL, MS Access)	23
3.1	Introduction	23
3.2	Configuration du Hardware	23
3.3	Description de la base de données	24
3.3.1	Listes des tables de la base de données	24
3.4	Migration vers les autres SGBDs	25
3.4.1	Migration de la base de données de MS SQL Server vers MS Access	26
3.4.2	Migration de la base de données de MS SQL Server vers MySQL	30
3.4.3	Migration de SQL Server vers Oracle :	33
3.5	Comparatif	35
3.5.1	Outils d'administration	35
3.5.2	Prix	35
3.5.3	Capacité de stockage	35

3.5.4	Performance	35
3.5.4.1	Req1 lecture . Sélectionner tous les noms, prénoms, dates de naissances des propriétaires ayants des véhicules de marque "PEUGEOT".	36
3.5.4.2	Req2 lecture.Sélectionner tous les enregistrements de la table carte où la date d'établissement des cartes grises est limité entre le 02/01/2013 et 30/10/2013	37
3.5.4.3	Req3 jointure simple. Sélectionner noms, prénoms et date de naissance des propriétaires ayant des véhicules gagés : jointure entre la table carte et la table gage.	38
3.5.4.4	Req4 jointure complexe.les propriétaires ayants des véhicules sous opposition depuis le 02/01/2013 par ordre alphabétique des noms.	39
3.5.5	Résultats obtenus	40
3.6	Conclusion	41
	Conclusion générale	41

Table des figures

1.1	Quelques chiffres sur l'utilisation du Web	10
1.2	Solution NoSQL clé/valeur	11
1.3	Solution NoSQL base documentaire	12
1.4	Solution NoSQL Bases orientées colonnes	13
1.5	Solution NoSQL Bases orientées famille de colonnes	14
1.6	Solution NoSQL paradigme graphe	15
3.1	Migration de la BDD SQL Server vers MS Access étape 01	26
3.2	Migration de la BDD SQL Server vers MS Access étape 02	26
3.3	Migration de la BDD SQL Server vers MS Access étape 03	27
3.4	Migration de la BDD SQL Server vers MS Access étape 04	27
3.5	Migration de la BDD SQL Server vers MS Access étape 05 selection des tables contenant des données	28
3.6	Migration de la BDD SQL Server vers MS Access étape 06	28
3.7	Base de données des cartes grise sous MS Access	29
3.8	Migration de la BDD SQL Server vers MySQL étape 01	30
3.9	Migration de la BDD SQL Server vers MySQL étape 02	30
3.10	Migration de la BDD SQL Server vers MySQL étape 03	31
3.11	Migration de la BDD SQL Server vers MySQL étape 04	31
3.12	Migration de la BDD SQL Server vers MySQL étape 05	31
3.13	Migration de la BDD SQL Server vers MySQL étape 06	32
3.14	La base de données des cartes grises sous MySQL	32
3.15	Migration de la BDD SQL Server vers Oracle étape 01	33
3.16	Migration de la BDD SQL Server vers Oracle étape 02	33
3.17	Migration de la BDD SQL Server vers Oracle étape 03	34
3.18	Migration de la BDD SQL Server vers Oracle étape 04	34
3.19	Migration de la BDD SQL Server vers Oracle étape 05	34
3.20	Caractéristiques des SGBDs	40

Introduction générale

Contexte

Le cerveau humain se compose de milliards de neurones. Chacun de ces neurones forme à peu près 1000 connexions avec d'autres neurones, ce qui correspond donc à plusieurs milliards de connexions et qui implique une capacité de stockage plutôt importante. Le mystère de Dieu le tout puissant réside dans la façon de gérer ces informations et leurs relations gigantesques.

Vous dites-nous ne sommes pas dans la médecine mais en effet l'informatique se rapproche du système humain. Le serveur peut stocker des milliards d'informations qui composent une base de données. Afin de les mettre à la disposition de l'utilisateur plusieurs opérations doivent être effectuées, on parle des transactions (lecture, jointure...etc). Un système qui assure ces dernières dit le système de gestion de base de données qui est l'objet de notre thèse.

Problématique

Plusieurs SGBDs disponibles sur le marché, choisir un d'entre eux n'est pas une chose assez facile, on peut facilement dire : je veux travailler avec celui- là ou travailler avec l'autre mais le plus difficile est comment juger son choix ?

Notre objectif est de mettre en œuvre une comparaison réelle entre des différents SGBDs pour aider les utilisateurs à bien savoir choisir un SGBD qui leur convient.

Contribution

Pour atteindre cet objectif nous avons mené une étude comparative entre quatre systèmes de gestion de base de données en se basant sur les critères suivants :

- Outils d'administration.
- Capacité de stockage.

- Prix.
- Performance.

Plan du mémoire

Le présent manuscrit va être organisé comme suit :

- Dans le premier chapitre, nous allons présenter quelques concepts se rapportant aux SGBDs, puis nous allons effectuer une classification des différents SGBDs selon leurs catégories (relationnel ou non, open source ou non, Amateur...etc), nous concluons par la présentation du modèle NoSQL le fil d'actualité des SGBDs.
- Le second chapitre va être consacré aux SGBDs relationnels qui vont être opposé dans le dernier chapitre, nous citons les principes ainsi que les avantages et les inconvénients de chacun.
- Le troisième chapitre fera l'objet de notre comparatif, en commençant par présenter la configuration du matériel utilisé, ensuite une description de la base de données sur laquelle les différents tests vont être faits, ainsi que les critères de comparaison, nous terminons cette section par une lecture des résultats obtenus.
- Finalement, on achève notre étude par une conclusion générale et un ensemble de perspectives qui peuvent compléter notre projet de fin d'études.

Chapitre 1

Généralités sur les SGBDs

1.1 Introduction

Le monde d'aujourd'hui est d'une complexité croissante. En effet, le nombre d'habitants de notre planète augmente quotidiennement, et les systèmes qui assurent les services vis-à-vis de cette population deviennent de plus en plus critiques. Les établissements gouvernementaux, les banques, et toutes les administrations ont pour rôle la gestion des affaires publiques.

L'accroissement des services critiques exige des systèmes de gestion de base de données efficaces et robuste qui permettra de garantir à la fois un maximum de stockage de données et un minimum de temps de réponse aux requêtes interrogeant ces dernières.

Durant le passé, la manipulation des données n'était pas très pratique, faute de technologie. En générale, le stockage d'informations s'effectuait manuellement et la recherche était une tâche pénible qui prenait beaucoup de temps.

Par contre, de nos jours, et grâce à l'évolution de l'informatique une telle tâche est devenue plus aisée avec une rapidité apparente qui tire profit de la puissance de calcul offerte par l'informatique moderne.

1.2 Définitions

1.2.1 Qu'es qu'un SGBD ?

Une base de données est un ensemble d'informations, connexes de manière directe ou indirecte, enregistrées dans un dispositif informatique. Dans une base de données relationnelle, les informations sont stockées sous forme de groupe de valeurs : les enregistrements. Un ensemble d'enregistrements relatif à un sujet forme une relation et est stocké dans une table. La base de données comporte une ou plusieurs tables.

La fonction première d'un Système de Gestion de Base de Données (SGBD) est d'être un outil de stockage d'informations offrant des fonctions simples de manipulation de grands volumes de données. L'un des avantages de ces SGBDs est que l'interrogation de ces informations s'effectue d'une manière indépendante de l'architecture physique de stockage.

Les SGBDs garantissent la cohérence de ces données en cas de mise à jour simultanée par plusieurs utilisateurs. Les transactions assurent l'intégrité des données en cas d'opérations incorrectes réalisées par un programme ou un utilisateur. Les données stockées dans un SGBD sont dites persistantes, leur fiabilité et leur récupération en cas de panne matérielle ou logicielle doit être toujours possible. De plus, le SGBD doit assurer la confidentialité des données en cas d'accès malveillant ou accidentel.[Christophe CARPENA, 2006]

Un SGBDR (système de gestion de bases de données relationnelles) est un système (logiciel) qui permet de gérer une base de données relationnelle. Les SGBDRs peuvent être soit de type client/serveur (Oracle, SQL server, MYSQL, PostgreSQL, etc...) soit de type fichiers partagés (Access, SQL server CE, Paradox, DBase, etc...).

1.2.2 Des acteurs bien installés

Les technologies de bases de données relationnelles et transactionnelles, qu'on résumera ici par „technologies SQL,, , règnent en maîtres pour le stockage et la manipulation de données depuis plus de 20 ans. Cette situation de leadership technologique peut facilement être expliquée par plusieurs facteurs.

Tout d'abord, SQL est un langage standardisé. Bien que chaque fournisseur de base de données ait implémenté quelques variantes, il n'en demeure pas moins que ce socle commun est bien connu par les développeurs. Pour une entreprise, investir dans une technologie

SQL s'avère donc moins coûteux sur le plan de la formation que toute autre technologie. Cela explique sans doute l'échec des bases de données objets que l'on a vu fleurir autour des années 2000 et qui n'ont jamais véritablement percé, malgré une approche novatrice.

Ensuite, SQL embarque de nombreuses fonctionnalités importantes, adaptées aux besoins des entreprises, en particulier la gestion de l'intégrité des données et l'implémentation de transactions, indispensables pour un bon nombre d'applications de gestion.

Enfin, les outils permettant l'exploitation de ce type de bases de données sont matures. Que l'on se place au niveau des outils de développements (IDE graphiques, intégration dans les langages et les frameworks,...) ou d'exploitation (outils de sauvegarde, monitoring,...), les solutions existent et sont prêtes depuis déjà un certain temps. On peut donc considérer SQL comme un standard de fait pour toute problématique ayant trait au stockage et à la manipulation de données.

La plupart des moteurs de SGBDs relationnels sont transactionnels ce qui leur impose le respect des contraintes Atomicity, Consistency, Isolation, Durability, communément appelé par son acronyme **ACID** :

- **A** comme **A**tomicity : Cela signifie que les mises à jour de la base de données doivent être atomiques, c'est-à-dire qu'elles doivent être totalement réalisées ou pas du tout. Par exemple, sur 5000 lignes devant être modifiées au sein d'une même transaction, si la modification d'une seule échoue, alors la transaction entière doit être annulée. C'est primordial, car chaque ligne modifiée peut dépendre du contexte de modification d'une autre, et toute rupture de ce contexte pourrait engendrer une incohérence des données de la base.
- **C** comme **C**onsistency : Cela signifie que les modifications apportées à la base doivent être valides, en accord avec l'ensemble de la base et de ses contraintes d'intégrité. Si un changement enfreint l'intégrité des données, alors soit le système doit modifier les données dépendantes, comme dans le cas classique d'une suppression en cascade, soit la transaction doit être interdite.
- **I** comme **I**solation : Cela signifie que les transactions lancées au même moment ne doivent jamais interférer entre elles, ni même agir selon le fonctionnement de chacune. Par exemple, si une requête est lancée alors qu'une transaction est en cours, le résultat de celle-ci ne peut montrer que l'état original ou final d'une donnée, mais pas l'état intermédiaire. De fait, les transactions doivent s'enchaîner les unes à la

suite des autres, et non de manière concurrentielle.

- **D** comme **Durability** : Cela signifie que toutes les transactions sont lancées de manière définitive. Une base ne doit pas afficher le succès d'une transaction, pour ensuite remettre les données modifiées dans leur état initial. Pour ce faire, toute transaction est sauvegardée dans un fichier journal, de sorte que si un problème survient empêchant sa validation complète, la transaction pourra être correctement terminée lors de la disponibilité du système. [Aurélien FOUCRET, 2011]

1.3 Classification des SGBDs

Nous proposons ici une classification selon les éditeurs comme suit :[Fabien Celaia, 2003]

1.3.1 MICROSOFT

1.3.1.1 SQL Server



Version récente 2012, disponible sous Windows seulement, sous licence commerciale, autre contributeur SQL Pro.

1.3.1.2 MS Access



Version actuelle 2010, Disponibilité sous Windows avec une Licence commerciale, Autre contributeur Maxence Hubiche.



1.3.2 Oracle Corporation

1.3.2.1 Oracle Database

Version actuelle : 11.2.0.4 ou Oracle 12c, Disponibilité : Linux, Windows, Unix, MacOSX, sous Licence commerciale, gratuite dans sa version Express, et comme Autre contributeur OraFrance.

1.3.2.2 Berkeley DB



Version actuelle : 4.7 (Java en 3.3, XML en 2.4), Disponibilité : Linux, Windows, Unix, Licence : commerciale ou OpenSource, Librairies de base de données aisément intégrables à des programmes C, C++, Java, Perl, Python, Tcl, Smalltalk.

1.3.2.3 HyperionEssBase



Disponibilité sous Linux, Windows, Unix, licence : commerciale, Oracle HyperionEssbase est généralement considéré comme le leader des SGBD multidimensionnels, autrement dit (M)OLAP.

1.3.2.4 MySQL

version actuelle : 6.0.5, disponibilité sous Linux, Windows, MacOSX, Unix, BSD, OS2, open source.

version disponible :



- MySQL Community Server : licence GPL.
- MySQL Enterprise = MySQL Community Server + certifié sécurité et performance + licence d'entreprise.

1.3.3 Firebird SQL Foundation

1.3.3.1 Firebird



Version actuelle : 2.5, disponibilité : Linux, Windows, MacOSX, Solaris, HP-UX, BSD, licence : OpenSource (IBPL pour les modules de base + IDPL pour les modules récents), Issu d'Interbase 6.0 de Inprise (Borland), repris dans un projet opensource, géré par la Fondation FirebirdSQL. Réécrit depuis en C++.

1.3.4 Sybase

1.3.4.1 SQL Anywhere



Version actuelle :12, disponible sous Windows, PalmOS, WinCE, Novell Netware, Blackberry, Anciennement nommé SQL Anywhere, Adaptive Server Anywhere ou encore

Watcom SQL. Orientée environnements embarqués grâce à son module de synchronisation SQL Remote, cette base de données aux normes SQL2 convient pour des volumes ne dépassant généralement pas les 5Gb avec pas plus de 20 utilisateurs concurrents. Elle offre l'avantage d'une administration minimum et la possibilité d'utiliser le Transact-SQL (possible de passer sur Sybase ASE ou MS-SQL sans trop de problèmes en cas de montée en charge).

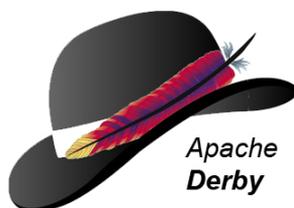
1.3.4.2 Adaptive Server Enterprise



disponible sous Linux, BSD, Windows, Unix, MacOSX, sa licence est commerciale, mais gratuite sous Linux dans sa version Express. Anciennement nommé Sybase SQL Server Père de MS-SQL Server, c'est un moteur SGBDR qui se comporte extrêmement bien en environnement OLTP ou mixte. En 2 mots : puissance et simplicité.

1.3.5 Apache

1.3.5.1 ApacheDerby



Version récente 10.10.1.1, disponible sous tous les systèmes comprenant une JVM, Open Source Apache 2, c'est un petit moteur SGBDR (2Mo) écrit en Java. Dès le rachat de Cloudscape par IBM, a été transférée en OpenSource à Apache sous le nom de Derby. IBM continue de la distribuer sous le nom de Cloudscape, Sun sous celui de SunJava DB.

1.4 Fil d'actualité

1.4.1 Qu'est-ce qui provoque le passage à NoSQL ?

L'utilisation de l'internet dans nos jours devienne indispensable, pour effectuer une simple recherche il nous faut des grands moteurs de recherche qui répond à nos besoins,

ces réponses ou bien dit ces données sont stockés quelque part sur le Wide World Web c'est-à-dire le stockage se fait sur des architectures distribuées où les systèmes de gestion de base de données relationnelles ne sont pas adaptés aux environnements distribués requis par les volumes gigantesques de données et par les trafics tout aussi gigantesques générés par ces opérateurs donc trois grandes tendances interdépendantes :BigData, Big User, et Cloud Computing conduisent à l'adoption de Technologie NoSQL.



FIGURE 1.1 – Quelques chiffres sur l'utilisation du Web

1.4.2 Les types de base NoSQL

1.4.2.1 Paradigme clé / valeur

Il s'agit de la catégorie de base de données la plus basique. Dans ce modèle chaque objet est identifié par une clé unique qui constitue la seule manière de le requêter.

La structure de l'objet est libre et le plus souvent laissé à la charge du développeur de l'application (XML, JSON,...), la base ne gérant généralement que des chaînes d'octets.

Dans ce modèle on ne dispose généralement que des quatre opérations Create Read Update Delete (CRUD) :

- **Create** : créer un nouvel objet avec sa clé : `create(key, value)`.
- **Read** : lit un objet à partir de sa clé : `read(key)`.
- **Update** : met à jour la valeur d'un objet à partir de sa clé : `update(key, value)`.
- **Delete** : supprime un objet à partir de sa clé : `delete(key)`.

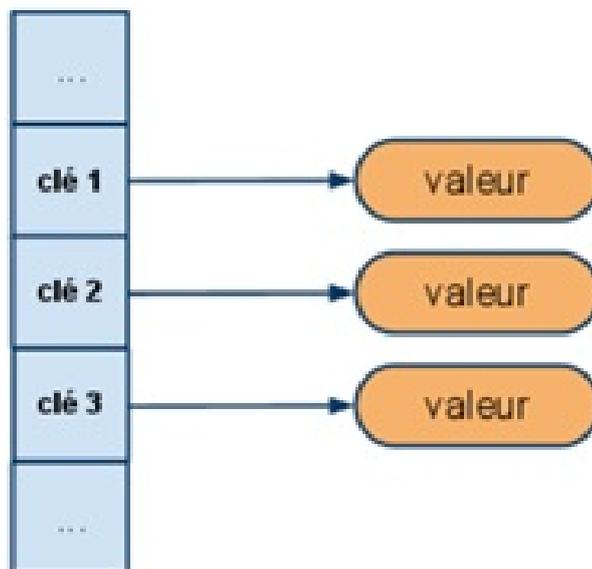


FIGURE 1.2 – Solution NoSQL clé/valeur

Les bases de ce type disposent pour la plupart d'une interface HTTP REST permettant de procéder très simplement à des requêtes, et ceci depuis n'importe quel langage de développement.

L'approche, volontairement très limitée de ces systèmes sur le plan fonctionnel, est radicale et leur permet d'afficher des performances exceptionnellement élevées en lecture et en écriture ainsi qu'une scalabilité horizontale considérable. Le besoin de scalabilité verticale est fortement réduit au niveau des bases par le caractère très simple des opérations effectuées.

Du fait de leur modèle de requêtes très simple et réduit aux opérations CRUD, ces systèmes sont principalement utilisés comme dépôt de données dès lors que les besoins en termes de requêtes sont très simples. On les retrouve très souvent comme système de stockage de cache ou de sessions distribuées, notamment là où l'intégrité relationnelle des données est non significative.[Aurélien FOUCRET, 2011]

1.4.2.2 Bases documentaires

Les bases de données documentaires sont constituées de collections de documents. Un document est composé de champs et des valeurs associées, ces dernières pouvant être requêtées.

Par ailleurs, les valeurs peuvent être, soit d'un type simple (entier, chaîne de caractère, date,...), soit elles-mêmes composées de plusieurs couples clé/valeur.

Bien que les documents soient structurés, ces bases sont dites : schemaless. A ce titre, il n'est pas nécessaire de définir au préalable les champs utilisés dans un document. Les documents peuvent être très hétérogènes au sein de la base.

Le stockage structuré des documents leur confère des fonctionnalités dont ne disposent pas les bases clés/valeurs simples dont la plus évidente est la capacité à effectuer des requêtes sur le contenu des objets.

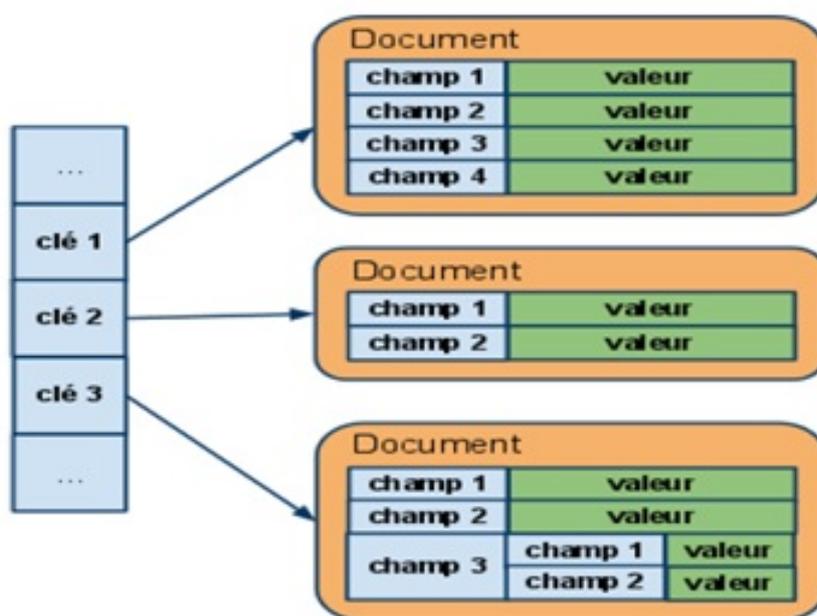


FIGURE 1.3 – Solution NoSQL base documentaire

On conserve généralement une interface d'accès HTTP REST permettant d'effectuer simplement des requêtes sur la base mais celle-ci est plus complexe que l'interface CRUD des bases clés/valeurs. Les formats de données JSON et XML sont le plus souvent utilisés

afin d'assurer le transfert des données sérialisées entre l'application et la base.

Du fait de leur conception plus simple que les bases de données relationnelles, ces bases conservent des performances élevées. Elles disposent par ailleurs de fonctionnalités très intéressantes en terme de flexibilité du modèle documentaire, ce que les bases relationnelles ne peuvent offrir. Cette dernière fonctionnalité leur confère une popularité importante, notamment dans le milieu du développement de CMS (Content Management System, outils de gestion de contenus) pour lesquels ce modèle est très adapté.

1.4.2.3 Bases orientées colonnes

Les bases orientées colonnes sont résolument les plus complexes à appréhender parmi la mouvance NoSQL. Bien que l'on obtienne au final un schéma relativement proche des bases documentaires, l'organisation sous-jacente des données permet à ces bases d'exceller dans les traitements d'analyse de données et dans les traitements massifs (notamment via des opérations de type MapReduce).

Bien que l'on trouve des variations en fonction de la base considérée, les concepts essentiels sont les suivants :

- Colonne : il s'agit de l'entité de base représentant un champ de donnée. Chaque colonne est définie par un couple clé / valeur. Une colonne contenant d'autres colonnes est nommée super-colonne.

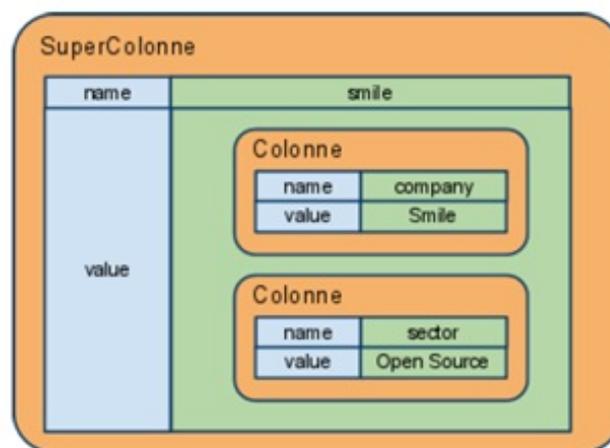


FIGURE 1.4 – Solution NoSQL Bases orientées colonnes

- Famille de colonnes : il s'agit d'un conteneur permettant de regrouper plusieurs colonnes ou super-colonnes. Les colonnes sont regroupées par ligne et chaque ligne

est identifiée par un identifiant unique. Elles sont généralement assimilées aux tables dans le modèle relationnel et sont identifiées par un nom unique.

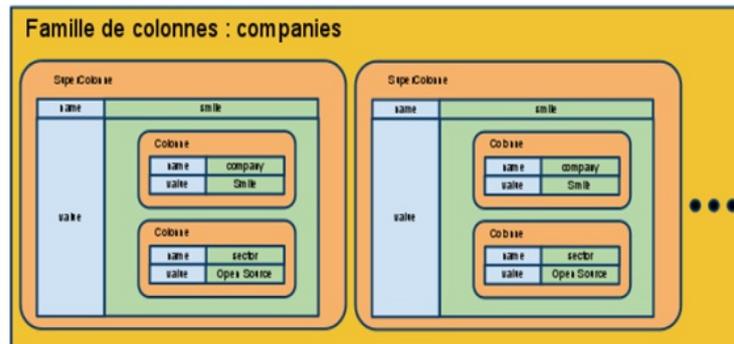


FIGURE 1.5 – Solution NoSQL Bases orientées famille de colonnes

Les super-colonnes situées dans les familles de colonnes sont souvent utilisées comme les lignes d'une table de jointure dans le modèle relationnel. Comparativement au modèle relationnel, les bases orientée colonnes offrent plus de flexibilité. Il est effet possible d'ajouter une colonne ou une super colonne à n'importe quelle ligne d'une famille de colonnes, colonnes ou super-colonne à tout instant.

1.4.2.4 Paradigme graphe

Ce paradigme est le moins connu de ceux de la mouvance NoSQL. Ce modèle s'appuie principalement sur deux concepts : d'une part l'utilisation d'un moteur de stockage pour les objets (qui se présentent sous la forme d'une base documentaire, chaque entité de cette base étant nommée nœud). D'autre part, à ce modèle, vient s'ajouter un mécanisme permettant de décrire les arcs (relations entre les objets), ceux-ci étant orientés et disposant de propriétés (nom, date,...).

Par essence ces bases de données sont nettement plus efficaces que le modèle relationnel, pour traiter les problématiques liées aux réseaux (cartographie, relations entre personnes,...). En effet, lorsqu'on utilise le modèle relationnel, cela nécessite un grand nombre d'opérations complexes pour obtenir des résultats.

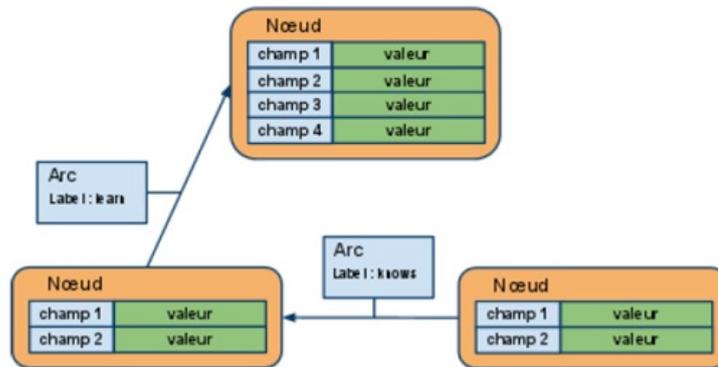


FIGURE 1.6 – Solution NoSQL paradigme graphe

1.4.3 Panorama des solutions NoSQL existantes

On dénombre actuellement une douzaine de solutions existantes, qu'elles soient de type :

- Base clé/valeur, comme Redis, Riak, Voldemort.
- Base documentaire, comme MongoDB, CouchDB, Terrastore.
- Base orientée colonne, comme Cassandra, Amazon SimpleDB, Google BigTable, HBase.
- Base orientée graphe, comme Neo4j, OrientDB.

Dans la mouvance NoSQL, les grands noms des bases de données propriétaires sont quasiment absents, les solutions citées ci-dessus étant soit open source, soit proposées en SaaS.

Les produits issus du monde SaaS et proposés sur ce mode uniquement, principalement portés par Google et Amazon, font partie intégrante de l'offre PaaS (Platform as a Service) mise à disposition des développeurs dans le cadre des frameworks de développements clouds.

Les produits open source constituent la majorité de l'offre NoSQL, et l'on assiste à une réelle montée de ces solutions dans ce secteur, souvent portée par des contributeurs importants comme Facebook, Twitter, LinkedIn, ... Un certain nombre de produits ont été développés pour les usages internes de ces grands acteurs de l'Internet, puis libérés dans des états d'avancement très importants, puisque déjà en production pour ces grands sites web. Notons que la fondation Apache joue ici encore un rôle primordial dans ces projets.

1.5 Conclusion

Ce chapitre illustre la disponibilité de nombreux SGBDs sur le marché les uns sont free ou payants et d'autre open source où sous licence commerciale. Et malgré les limites des systèmes de gestion de base de données relationnels et l'apparition du NoSQL (Not Only SQL), ils restent indispensables et toujours en progression, leur utilisation dépend des besoins des entreprises dans différents domaines tels que le e-commerce, les dossiers médicaux, les paiements, la gestion des ressources humaines ...etc.

La problématique est toujours posée et critique comment choisir le meilleur SGBD dans cette panoplie de SGBDs. C'est à cette problématique, qu'on va essayer de donner quelques éléments de réponse.

Chapitre 2

Présentation des SGBDs étudiés

2.1 Introduction

Dans le milieu professionnel, on retrouve principalement des SGBDR client/serveur même si les solutions en fichiers partagés ont eu leur heure de gloire et sont encore utilisées dans certaines applications. L'apparition de SGBDR client/serveur gratuits a fortement contribué à populariser ce modèle ces dernières années. Le modèle client/serveur nécessite généralement la présence d'un serveur, qui traite les requêtes transmises par le client et lui retourne le résultat.

Dans ce comparatif nous avons choisi 4 SGBDs : deux sont free, les deux autres sont payants, un open source et les trois autres non. Deux font partie des produits lourds et deux des légers.

2.2 SQL server

2.2.1 Définition

Produit par Microsoft, SQL Server est un système de gestion de bases de données relationnelles. Le stockage, la manipulation et l'analyse de ces données se font au sein de son moteur de bases de données. Ce service permet la réalisation de nombreuses applications, requêtes, et transactions, notamment grâce au langage T-SQL (Transact-SQL).[Fabien Celaia, 2003]

2.2.2 Avantages

Parmi les avantages de SQL Server nous citons :

- Administration aisée.
- Fonction d'audit évoluée.
- Indépendance entre les diverses bases, facilitant l'intégration de plusieurs applicatifs dans une même instance.
- Une des bases les plus performantes sous Windows en configuration par défaut.
- Optimiseur statistique enrichi à flux tendu.
- Langage T-SQL très convivial, intégration de CLR.
- Gestion de l'indexation textuelle.
- Niveau de SQL très près de la norme SQL et implémente presque toutes les possibilités de SQL.
- Services Web et Support XML.
- Ordonnanceur intégré.
- Compression des données et des sauvegardes.

2.2.3 Inconvénients

Malgré tous ses avantages, le SQL Server a des inconvénients :

- Distributions fortement liées au système d'exploitation.
- Jungle des versions, mais fonctionnalités cantonnées dans les éditions Enterprise, Developer et Standard.
- Mono-plateforme (MS Windows).
- Depuis la version 2005, plus de prise directe sur les tables système (remplacées par de vues système).
- Toujours pas de cluster (hormis en actif-passif, en se basant sur le cluster OS).
- Pas certifié SQLJ, pas d'intégration Java, orientation C#.
- Pas de contraintes d'unicité multi null.

2.3 Oracle DB

2.3.1 Définition

Oracle, édité par Oracle Corporation (qui édite également MySQL) est un SGBDR payant. Son coût élevé fait qu'il est principalement utilisé par des entreprises. Oracle gère très bien de grands volumes de données. Il est inutile d'acheter une licence oracle pour un projet de petite taille, car les performances ne seront pas déférentes de celles de MySQL ou d'un autre SGBDR. Par contre, pour des projets conséquents (plusieurs centaines de Go de données), Oracle sera bien plus performant. Par ailleurs, Oracle dispose d'un langage procédural très puissant (du moins plus puissant que le langage procédural de MySQL) :

le PL/SQL.

2.3.2 Avantages

Comme tout SGBD nous indiquons quelques avantages d'Oracle :

- Richesse fonctionnelle.
- Fonction d'audit évolué.
- Row Level Storage Security (RLSS) : permet de ne faire apparaître que certaines lignes des tables pour un utilisateur/une application donnée.
- Parallélisme, caches nommés, haute disponibilité.
- Procédures stockés en PL-Sql (langage propriétaire Oracle, orienté ADA) ou... en JAVA (depuis la version 8.1.7) ce qui peut s'avérer utile pour les équipes de développement.
- Assistants performants via Oracle Manager Server, possibilité de gérer en interne des tâches et des alarmes.
- Gestion centralisée de plusieurs instances.
- Concept unique de retour arrière (Flashback).
- Pérennité de l'éditeur : avec plus de 40
- Réglages fins : dans la mesure où l'on connaît suffisamment le moteur, presque tout est paramétrable.
- Accès aux données système via des vues, bien plus aisément manipulable que des procédures stockées.
- Services Web, support XML
- Ordonnanceur intégré
- Compression des données et des sauvegardes
- Support technique Orion extrêmement riche et fourni.

2.3.3 Inconvénients

Les inconvénients d'Oracle sont les suivants :

- Prix élevé, tant au point de vue des licences que des composants matériels (RAM, CPU) à fournir pour de bonnes performances.
- Administration complexe, liée à la richesse fonctionnelle.
- Fort demandeur de ressources, ce qui n'arrange rien au point précité, Oracle est bien plus gourmand en ressource mémoire que ses concurrents, ce qui implique un investissement matériel non négligeable.
- Méta-modèle propriétaire, loin de la norme.
- Gestion des verrous mortels mal conçue (suppression d'une commande bloquante sans rollback).

- Une quantité des bugs proportionnels à la richesse fonctionnelle, surtout sur les dernières versions.
- Gestion erratique des rôles et privilèges (pas possible de donner des droits sur des schémas particuliers sans passer par leurs objets, désactivation des rôles lors d'exécution de packages...)
- Pas de type auto-incrément déclaratif : les séquences ne peuvent être déclarativement dédiées à une table spécifique (risque de mélange)
- Nombreuses failles de sécurités liées à l'architecture elle-même.

2.4 MySQL

2.4.1 Définition

MySQL est un SGBDR qui utilise le langage SQL, et fait parti des plus utilisés. Sa popularité est due en grande partie au fait qu'il s'agit d'un logiciel Open Source, ce qui signifie que son code source est librement disponible et que quiconque qui en ressent l'envie et/ou le besoin peut modifier MySQL pour l'améliorer ou l'adapter à ses besoins. Une version gratuite de MySQL est par conséquent disponible. À noter qu'une version commerciale payante existe également [Christophe CARPENA, 2006].

2.4.2 Avantages

Les avantages de MySQL sont comme suit :

- Solution très courante en hébergement public.
- Très bonne intégration dans l'environnement Apache/PHP
- OpenSource, bien que les critères de licence soient de plus en plus difficiles à supporter.
- Version cluster depuis la version 4.
- ordonnanceur et partitionnement dès la version 5.1.
- Facilité de déploiement et de prise en main.
- Plusieurs moteurs de stockage adaptés aux différentes problématiques, configurable au niveau table.
- Multi plate-forme Linux, Windows, OSX, Unix...ect.

2.4.3 Inconvénients

- Ne supporte qu'une faible partie des standards SQL-92.
- Support incomplet des triggers et procédures stockées.
- Assez peu de richesse fonctionnelle.

- Manque de robustesse avec de fortes volumétries.
- Pas d'héritage de table.
- Pas de vue matérialisée.
- Cluster par clonage de base : impact prépondérant sur la volumétrie.

2.5 MS Access

2.5.1 Définition

MS Access ou Microsoft Access est un logiciel édité par Microsoft (comme son nom l'indique...), par conséquent, c'est un logiciel payant qui ne fonctionne que sous Windows. Il n'est pas du tout adapté pour gérer un grand volume de données et a beaucoup moins de fonctionnalités que les autres SGBDR. Son avantage principal est l'interface graphique intuitive qui vient avec le logiciel.

2.5.2 Avantages

- Très puissant et très ludique, il apporte un grand nombre d'outils pour réaliser des outils de reporting de données. Possibilité de s'en servir comme interface sur une base SQLServer ou connecté à une (des) liste(s)WSS.
- Il contient une grande série d'outils de conversion de données, pour récupérer ou exporter depuis presque n'importe quel format vers n'importe quel format.
- Il contient une grande série d'outils de conversion de données, pour récupérer ou exporter depuis presque n'importe quel format vers n'importe quel format.
- Les macros permettent à des néophytes de se lancer dans une forme allégée de l'automatisation.
- Quantité d'assistants dirigeant l'utilisateur vers une première solution.
- Forte intégration à la Suite Microsoft Office/VBA, déjà fortement répandue en entreprise.
- Possibilité de développer des applications Runtime évitant ainsi l'achat massif de licences Access sur numéraire.

2.5.3 Inconvénients

- Le moteur JET étant un moteur «Fichier », il est gourmand en ressources réseau et ne convient pas pour les applications distantes.
- Le système de verrouillage des enregistrements peut induire des messages d'avertissements si deux utilisateurs cherchent à écrire simultanément dans une même page de données. La consultation est parcontre multi-utilisateur.

- Mono-plateforme (MS Windows).
- N'implémente pas complètement les normes SQL. Certaines options permettent néanmoins de passer du SQL-Access à une écriture conforme à la norme ANSI92, mais elles sont peu connues et peu utilisées.
- La «plaie »du DBA en entreprise : son utilisation ouverte aux débutants risque de laisser fleurir une kyrielle d'applications/de base de données sans contrôle au sein de l'entreprise.

2.6 Conclusion

Chacun des SGBDs cités dans ce chapitre a des points forts et des points faibles par rapport à l'autre, mais choisir un parmi les autres se voit toujours comme une vraie problématique, même pour les experts du domaine.

Pour apporter quelques éléments de réponse à cette question, le chapitre suivant abordera un comparatif entre les quatre SGBDs qui fixe comme objectif d'assister les utilisateurs dans leurs choix, avant d'investir et commettre l'irréparable.

Chapitre 3

Comparatif entre les 04 SGBDs (Oracle, SQL Server, MySQL, MS Access)

3.1 Introduction

Notre but dans ce projet de fin d'étude est de donner des éléments de réponse aux utilisateurs pour choisir le SGBD convenable.

Dans ce dernier chapitre, nous allons présenter l'étude comparative suivie, nous commençons par une présentation du matériel utilisé, une description de la base de données sur laquelle on fera nos différents tests, avec une explication des étapes de migration de la base données vers les différents SGBDs, puis une présentation des différents critères de comparaison, ainsi que les résultats obtenus.

3.2 Configuration du Hardware

La configuration du PC sur lequel le comparatif a été effectué est comme suit :

- Type : Laptop.
- Système d'exploitation : Windows 7 Edition Familial Premium ;
- Fabricant : ASUSTeK Computer Inc.
- Modèle : ASSUS Notebook Series
- Evaluation : 4.5 Indice de performance Windows.
- Processeur : Intel (R) Pentium(R) CPU B950 @ 2.10GHz 2.10GHz
- Mémoire installée (RAM) : 4.00 Go

- Type du système : système d'exploitation 64 bits.

3.3 Description de la base de données

Notre travail a été effectué sur la base de données des cartes grises de la wilaya de Tlemcen backup du 30/10/2013, faite sous MS SQL Server 2005, comportant 25 tables en totalité.

3.3.1 Listes des tables de la base de données

Nous citons ci-dessous une listes des tables de la base de données :

1. Table Carrosserie : contient 04 champs et 147 enregistrements, Carrosseries des véhicules.
2. Table Carte : contient 68 champs et 251568 enregistrements, c'est la table comportant le maximum d'informations concernant les véhicules et leurs propriétaires.
3. Table Commune : contient 04 champs et 53 enregistrements, Les 53 communes de la wilaya de Tlemcen.
4. Table Fonction : contient 04 champs et 41 enregistrements, Les fonctions des propriétaires.
5. Table GAGE : contient 04 champs et 1099 enregistrements, Les informations sur les gages des véhicules.
6. Table Genre : contient 03 champs et 27 enregistrements, Une liste des genres des véhicules.
7. Table HISTO : contient 53 champs et 413247 enregistrements, L'historique des propriétaires des véhicules.
8. Table Marque : contient 04 champs et 1050 enregistrements, Les marques des véhicules.
9. Table Modif_ave : contient 17 champs et 27028 enregistrements, La modification des caractéristiques des véhicules.
10. Table Modif_Std : contient 24 champs et 25398 enregistrements, La modification des informations des propriétaires.
11. Table Nuldisp : contient 02 champs et 611 enregistrements, Les numéros d'immatriculation de chaque année par catégorie.
12. Table NumSerAnnule : contient 07 champs et 6777 enregistrements, Les numéros de série des cartes grise annulées.

13. Table OPP_MAT : contient 07 champs et 4214 enregistrements, Oppositions sur les véhicules par matricule.
14. Table OPP_TS : contient 09 champs et 827 enregistrements, Oppositions sur les véhicules par type et numéro de série.
15. Table Permis : contient 27 champs et 82 enregistrements, Permissions des utilisateurs.
16. Table Poste : contient 02 champs et 908 enregistrements, Les postes de saisie autorisée pour chaque utilisateur.
17. Table PVmine : contient 03 champs et 6499 enregistrements, Les informations des PV de mine.
18. Table Quittance : contient 03 champs et 10 enregistrements, Les montants des quittances.
19. Table SUP_CARTE : contient 34 champs et 7024 enregistrements, Liste des cartes grise supprimées.
20. Table SUP_DUP : contient 06 champs et 590 enregistrements, Liste des cartes grise duplicatas supprimées.
21. Table SUP_FC : contient 09 champs et 7783 enregistrements, Liste des fiches de contrôles supprimées.
22. Table SUP_OM : contient 06 champs et 1165 enregistrements, Les oppositions par matricule annulées.
23. Table SUP_OTS : contient 07 champs et 13 enregistrements, Les oppositions par type et numéro série annulées.
24. Table SUP_VG : contient 07 champs et 7963 enregistrements, Liste des gages annulés.
25. Table WILAYA : contient 04 champs et 394 enregistrements, Liste des wilayas plus leurs dairas.

3.4 Migration vers les autres SGBDs

Notre projet a pour sujet la comparaison entre les SGBDs choisis (MS SQL Server, Oracle, MySQL et MS Access) comme il est mentionné au pare avant, alors on a fait une migration de la même BDD vers les 03 autres SGBDs, pour déterminer les performances de chacun de ces SGBDs en tenant compte du type du hardware utilisé et les requêtes choisies pour l'interrogation de la BDD, pour aboutir à la fin à des résultats d'évaluation concrets.

3.4.1 Migration de la base de données de MS SQL Server vers MS Access

Nous avons fait une importation des données à partir de SQL Server vers MS Access à l'aide de l'outil de l'importation dans MS Access.

Les étapes à suivre sont comme suit :

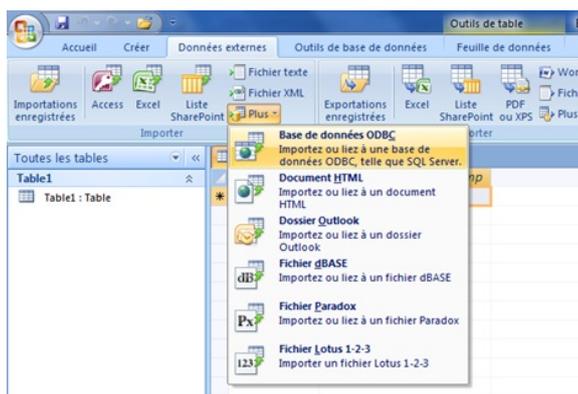


FIGURE 3.1 – Migration de la BDD SQL Server vers MS Access étape 01

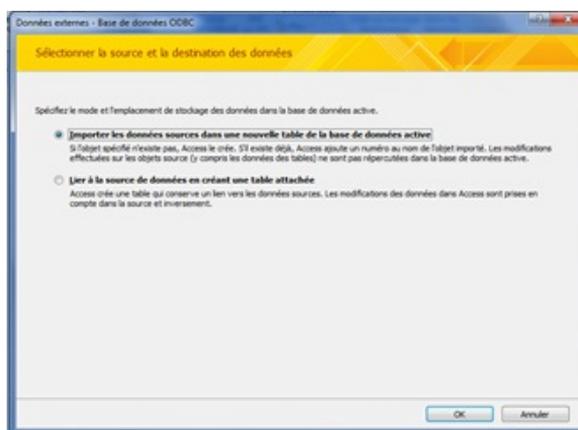


FIGURE 3.2 – Migration de la BDD SQL Server vers MS Access étape 02

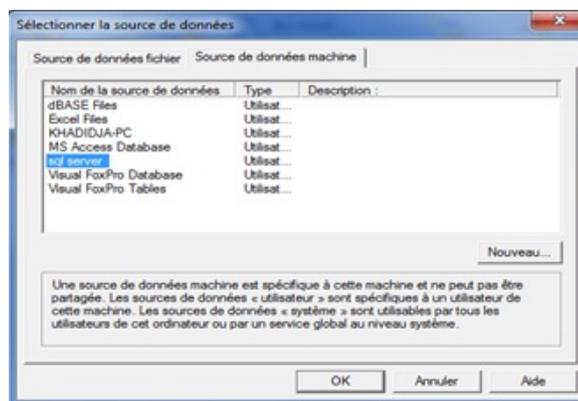


FIGURE 3.3 – Migration de la BDD SQL Server vers MS Access étape 03

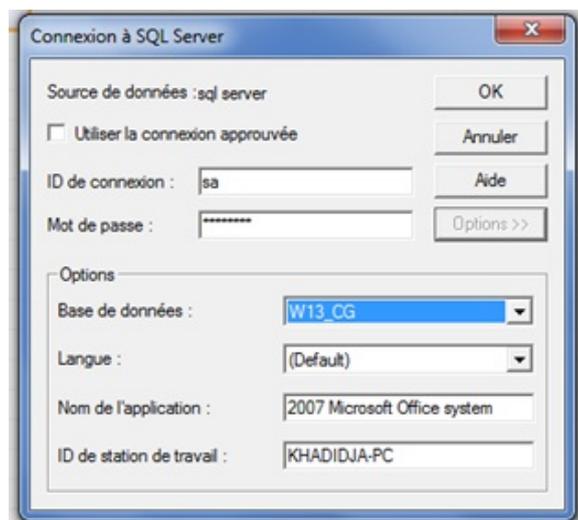


FIGURE 3.4 – Migration de la BDD SQL Server vers MS Access étape 04

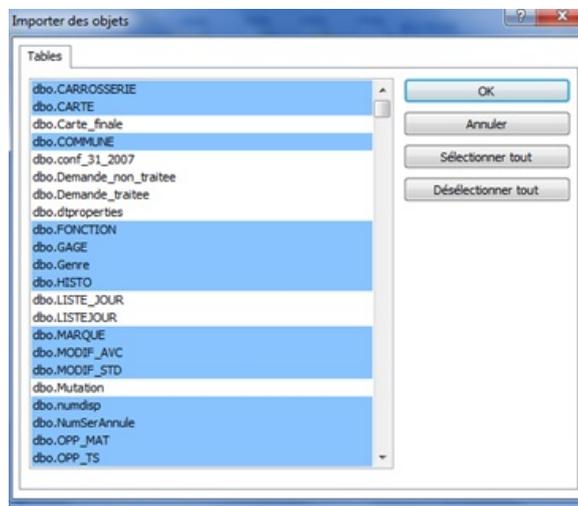


FIGURE 3.5 – Migration de la BDD SQL Server vers MS Access étape 05 selection des tables contenant des données

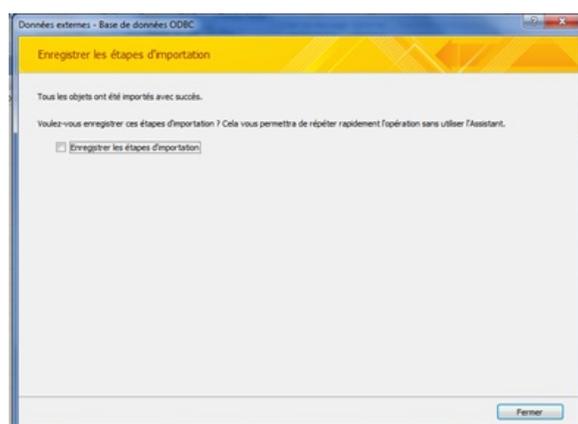


FIGURE 3.6 – Migration de la BDD SQL Server vers MS Access étape 06

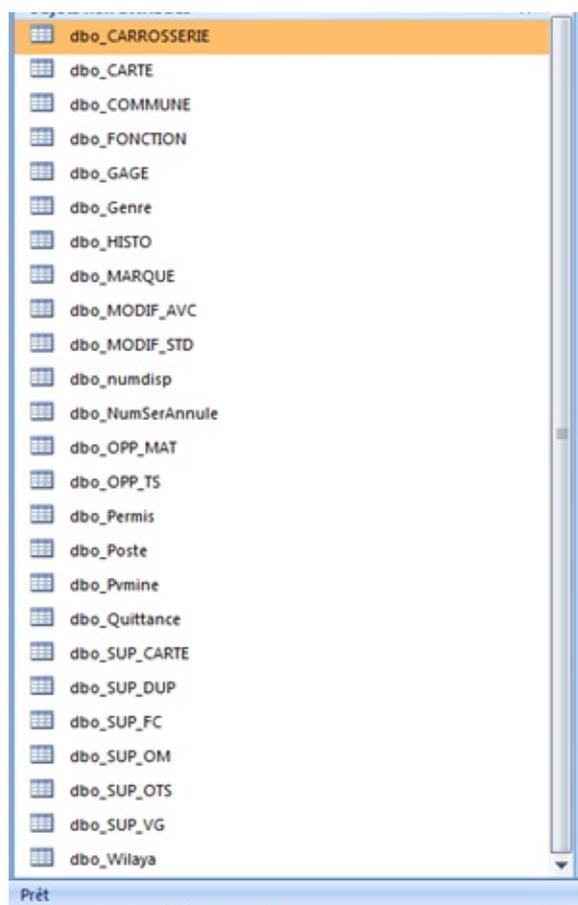


FIGURE 3.7 – Base de données des cartes grise sous MS Access

3.4.2 Migration de la base de données de MS SQL Server vers MySQL

La Migration a été faite avec l’outil Migration Wizard fournis par le MySQL sous les étapes suivantes :

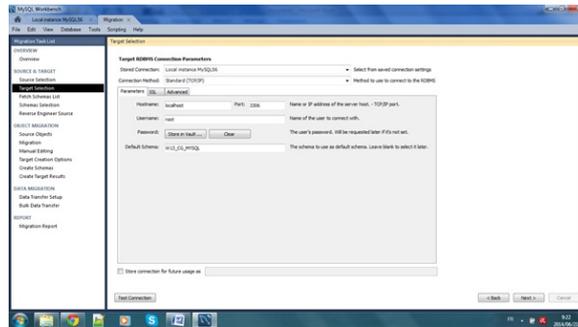


FIGURE 3.8 – Migration de la BDD SQL Server vers MySQL étape 01

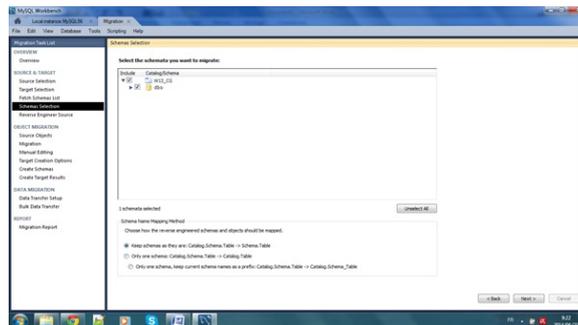


FIGURE 3.9 – Migration de la BDD SQL Server vers MySQL étape 02

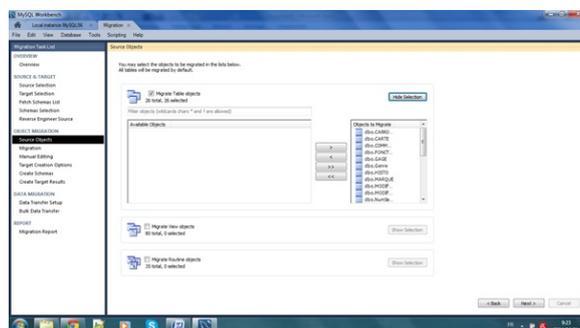


FIGURE 3.10 – Migration de la BDD SQL Server vers MySQL étape 03

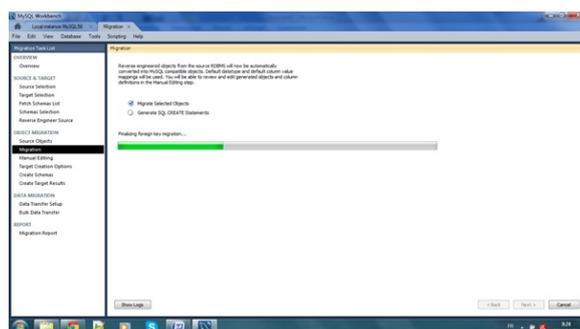


FIGURE 3.11 – Migration de la BDD SQL Server vers MySQL étape 04

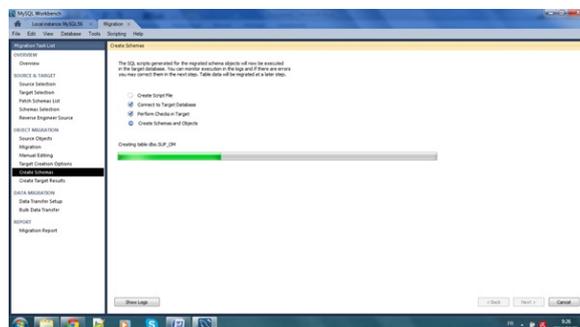


FIGURE 3.12 – Migration de la BDD SQL Server vers MySQL étape 05

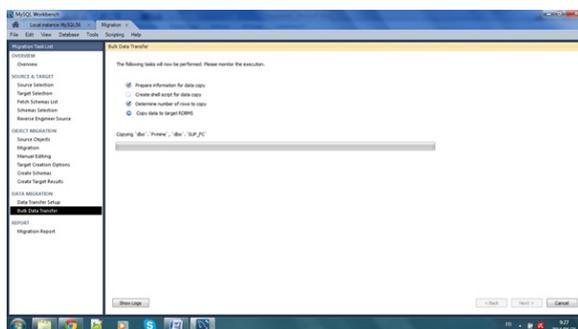


FIGURE 3.13 – Migration de la BDD SQL Server vers MySQL étape 06

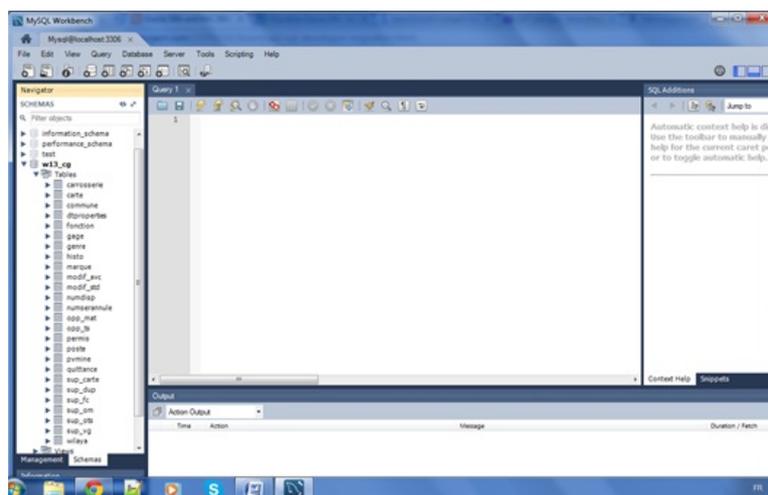


FIGURE 3.14 – La base de données des cartes grises sous MySQL

3.4.3 Migration de SQL Server vers Oracle :

La Migration a été faite avec un utilitaire ESF Database Migration Toolkit sous les étapes suivantes :



FIGURE 3.15 – Migration de la BDD SQL Server vers Oracle étape 01

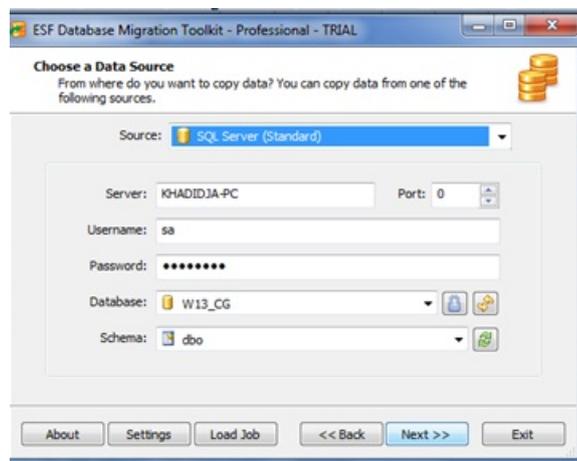


FIGURE 3.16 – Migration de la BDD SQL Server vers Oracle étape 02



FIGURE 3.17 – Migration de la BDD SQL Server vers Oracle étape 03

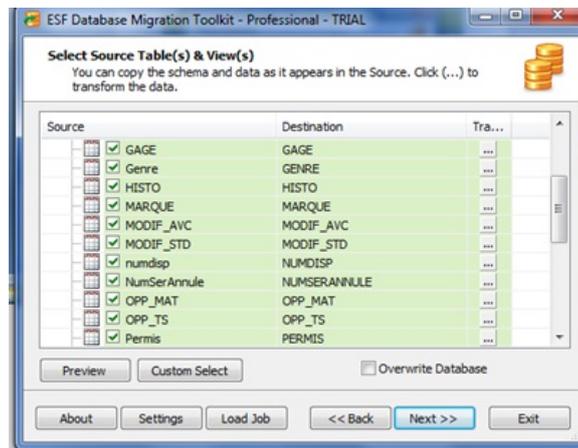


FIGURE 3.18 – Migration de la BDD SQL Server vers Oracle étape 04

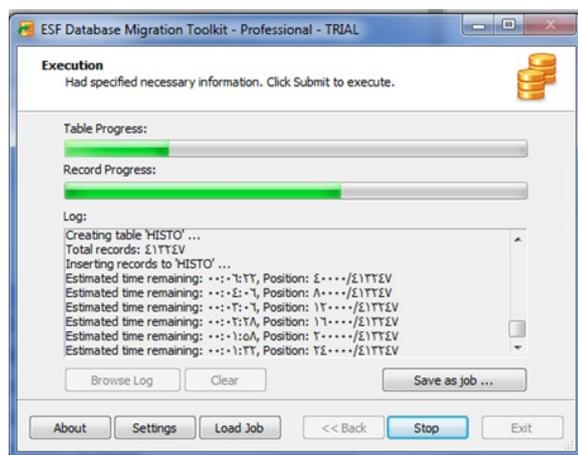


FIGURE 3.19 – Migration de la BDD SQL Server vers Oracle étape 05

3.5 Comparatif

3.5.1 Outils d'administration

SGBDs	Oracle	SQL Srver	MySQL	MS Access
Outils d'administration	Puissants mais la console d'administration n'est pas assez simple	Le plus simple à utiliser mais ne fonctionne que sous NT	Facile à utiliser sous plusieurs plates-formes	Simple à utiliser mais ne fonctionne que sous Windows

3.5.2 Prix

SGBDs	Oracle	SQL Srver	MySQL	MS Access
Prix	Payant, chère par rapport aux petites entreprises environ 95000\$	les versions payantes environ 12692\$	Gratuit et open source	Environ 31.26\$

3.5.3 Capacité de stockage

SGBDs	Oracle	SQL Srver	MySQL	MS Access
Capacité de stockage	Un stockage supérieur à 16 To	524 Po	entre 2 Go et 4 Go tous dépend du système d'exploitation	2 Go moins l'espace nécessaire aux objets système

3.5.4 Performance

La comparaison s'est fait à travers un ensemble de requêtes simples et autres massives pour avoir les différents temps de réponse de chaque SGBD.

3.5.4.1 Req1 lecture . Sélectionner tous les noms, prénoms, dates de naissances des propriétaires ayants des véhicules de marque "PEUGEOT".

SGBDs	Oracle	SQL Svrer	MySQL	MS Access
Syntaxe	Select nom,prenom,daten from carte where marque="& marque"; PEUGEOT	Select nom,prenom,daten from carte where marque="PEUGEOT"	select nom,prenom,daten from carte where marque = "PEU- GEOT"	SELECT nom,prenom,daten FROM dbo_CARTE where marque="PEUGEOT"
Nbr d'enre- gistrements retournés	18326	18326	18326	18326
Temps de ré- ponse	2 sec	2 sec	3 sec	3 sec

3.5.4.2 Req2 lecture.Sélectionner tous les enregistrements de la table carte où la date d'établissement des cartes grises est limité entre le 02/01/2013 et 30/10/2013

SGBDs	Oracle	SQL Srvr	MySQL	MS Access
Syntaxe	Select * from CARTE where dateetab >"02.01.2013 "and dateetab <"30.10.2013 "	Select * from carte WHERE (dateetab BETWEEN CONVERT (DATE-TIME, "2013-01-02 00 :00 :00", 102) AND CONVERT (DATE-TIME, "2013-10-30 00 :00 :00", 102))	select * from carte where dateetab between "2013-01-02"and "2013-10-30"	SELECT dbo_CARTE.* FROM dbo_CARTE WHERE (((dbo_CARTE. dateetab)>#1/1/2013# And (dbo_CARTE. dateetab)<#10/31/2013#))
Nbr d'enregistrements retournés	51484	51484	51484	51484
Temps de réponse	2 sec	8sec	3.47sec	3sec

3.5.4.3 Req3 jointure simple. Sélectionner noms, prénoms et date de naissance des propriétaires ayant des véhicules gagés : jointure entre la table carte et la table gage.

SGBDs	Oracle	SQL Svrer	MySQL	MS Access
Syntaxe	<pre>SELECT carte.matricule, carte.nom, carte.prenom, carte.daten, carte.dateetab, gage.matricule FROM gage WHERE gage.matricule= carte.matricule;</pre>	<pre>SELECT matricule, nom, prenom, da- ten, dateetab FROM CARTE WHERE (matricule IN (SE- LECT Matricule FROM GAGE))</pre>	<pre>SELECT matricule, prenom, da- ten, dateetab FROM CARTE WHERE (matricule IN (SE- LECT Matricule FROM GAGE))</pre>	<pre>SELECT matricule, nom, prenom, dateetab FROM dbo_CARTE WHERE (matricule IN (SELECT Matricule FROM dbo_GAGE));</pre>
Nbr d'enregistrements retournés	10701	10701	10701	10701
Temps de réponse	2 sec	1sec	15sec	2.56sec

3.5.4.4 Req4 jointure complexe.les propriétaires ayant des véhicules sous opposition depuis le 02/01/2013 par ordre alphabétique des noms.

SGBDs	Oracle	SQL Svrer	MySQL	MS Access
Syntaxe	<pre>SELECT carte.matricule, carte.nom, carte.prenom, carte.daten, OPP_MAT.matricule, OPP_MAT.date FROM CARTE, OPP_MAT WHERE (OPP_MAT.matricule = carte.Matricule and OPP_MAT.date >"30.10.2013 "OR- DER BY nom ;</pre>	<pre>SELECT matricule, nom, prenom, da- ten FROM CARTE WHERE (matri- cule IN (SELECT matricule FROM OPP_MAT WHERE (date >"2013-01-02") ORDER BY nom</pre>	<pre>SELECT matricule, nom, prenom, daten FROM dbo_CARTE WHERE (matri- cule IN (SELECT matricule FROM dbo_OPP_MAT WHERE (date >#02 / 01 / 2013#))) ORDER BY nom ;</pre>	
Nbr d'enre- gistrements retournés	234	234	234	234
Temps de ré- ponse	0.40 sec	0.45 sec	0.15 sec	0.62 sec

3.5.5 Résultats obtenus

Le comparatif réalisé nous a permis de tirer quelques résultats sur l'utilisation de ces 04 SGBDs commençant par :

- **Oracle** : Connu par le leader des SGBDs c'est le SGBD adéquat aux grandes entreprises malgré son prix élevé mais sa capacité de stockage et ces outils de sécurité de données sont importants, en fait en théorie le stockage d'Oracle est illimité mais en pratique il peut être supérieur à 16 To mais utiliser Oracle nécessite des grandes compétences à l'installation et à l'administration.
- **SQL Server** : c'est un SGBD très puissant, robuste avec une interface ergonomique et facile à utiliser, sa capacité de stockage des données est aussi importante mais un grand problème avec SQL Server est dédié aux entreprises utilisant le Système d'exploitation Windows seulement.
- **MySQL** : c'est le plus populaire au monde avec sa vitesse supérieure, sa fiabilité et facilité d'utilisation, son point fort est d'être gratuit et peut être utilisé même par les débutants au domaine.
- **MS Access** : c'est le SGBD convenable pour les petites applications, sa capacité de stockage est limitée mais sa rapidité est très remarquable.

Nous récapitulons les résultats obtenus dans l'histogramme ci-dessous :

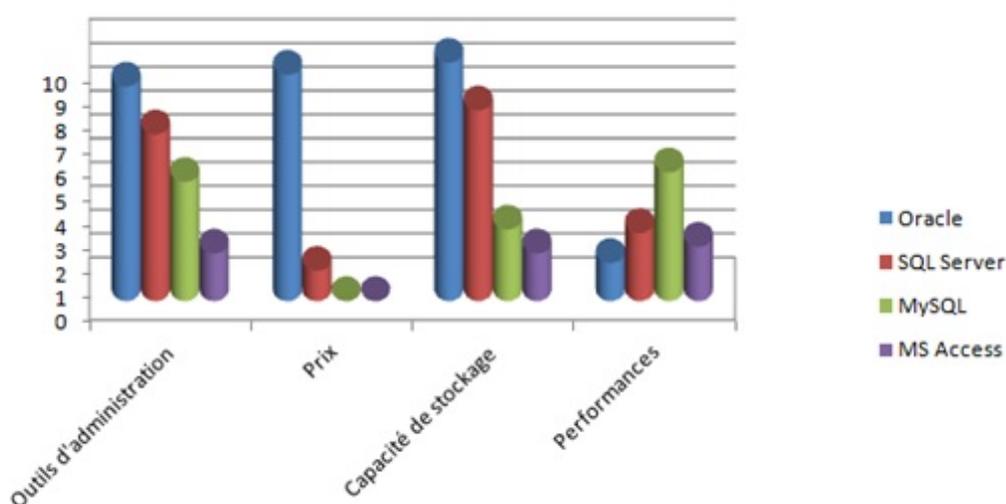


FIGURE 3.20 – Caractéristiques des SGBDs

3.6 Conclusion

Le but de notre projet de fin d'étude est de réaliser un comparatif concret sur les SGBDs parmi les plus connus dans le monde informatique, en se basant sur des critères bien définis qui restent des facteurs de témoignage sur les performances de chacun entre eux. Il faut dire aussi que le travail n'était pas facile, notamment la migration de la base de données et l'installation de certains SGBD citons l'oracle à titre d'exemple. En fin, on ne peut pas dire qu'un SGBD est meilleur que l'autre, le choix dépend des besoins et des moyens de chaque utilisateur (entreprise).

Conclusion générale

Dans le cadre de notre projet de fin d'étude, l'objectif ciblé était donner des indicateurs aux utilisateurs dans les différents environnements matériel-logiciel, grandes-petites entreprises, amateurs-professionnels, privés-étatiques, qui vont leur servir comme éléments de bases dans leurs cahiers de charge.

L'étude comparative développée s'est appuyée sur les différents critères suivants :

- Les outils d'administration.
- Les prix des SGBDs payants.
- Capacité de stockage.
- Et la performance de chacun des SGBDs choisis.

Les résultats obtenus ont été très convaincants et déterminants pour assister les décideurs d'établissements : Directeurs, informaticiens, financiers et enseignants, pour d'éventuelles prises de décisions qui sont jugées primordiales pour l'avenir de n'importe quelle entreprise.

Enfin, comme dans tout travail des perspectives auront toujours lieu à accomplir dont nous citons une liste non exhaustive des dites améliorations susceptibles d'être réalisées ultérieurement à ce projet :

- Utiliser un vrai serveur avec une configuration puissante et un système d'exploitation dédié aux serveurs tel que Windows Server 2008 ou d'autre dans le but de minimiser les temps de réponse des SGBDs aux différentes requêtes quelque sois simples ou complexes.
- Développer une application qui interroge les SGBDs depuis les postes clients et qui calcule automatiquement les temps de réponse.
- Enfin nous pensons à travailler dans le future sur le NoSQL dans le but de réaliser une étude comparative entre ces différents paradigmes et développer une application Web approuvant les résultats issu de cette comparaison.

Bibliographie

[Christophe CARPENA, 2006] Christophe CARPENA. Comparaison de SGBD libres : langage, triggers, administration. Examen probatoire du cycle ingénieur C.N.A.M. CONSERVATOIRE NATIONAL DES ARTS ET METIERS CENTRE REGIONAL DE MONTPELLIER. Pub (2006).

[Aurélien FOUCRET, 2011] Aurélien FOUCRET. NoSQL une nouvelle approche du stockage et de la manipulation des données. Article scientifique .(2011)

[Fabien Celaia, 2003] Faibien Celaia. Quel SGBD choisir?. Articles scientifique Publié le : 01.03.2003. Mise à jour 22.09.2010

[Didier Donsez, 2002] Didier Donsez, Panorama des bases de données. Université Joseph Fourier (Grenoble 1).(2002)

[Christian Soutou, 2005] Christian Soutou. Apprendre SQL avec MySQL. Edition EYRILLES (2005).

Site Web

[W1] <http://www.microsoft.com/france/serveurcloud/sql/>

[W2] <http://www.oracle.com/fr/index.html>