



République Algérienne Démocratique Et Populaire  
Ministère De L'enseignement Supérieur Et De La Recherche  
Scientifique

**Université KASDI MERBAH Ouargla**

Faculté des Nouvelles Technologies de l'Information et la  
Communication

Département de l'Informatique et Technologies de l'Information

## **Mémoire de fin d'étude en vue d'obtention du diplôme de Master Académique**

Domaine : Mathématique et Informatique

Filière : Informatique

Spécialité : Informatique Industrielle

**Présenté par :** - Gana Abdelmounaïme  
- Bouhenika Ahmed

**Thème :**

---

**Déploiement d'une base de données NoSQL avec MongoDB**

---

Soutenu publiquement le: 18/06/2022.

**Devant le jury composé de:**

Mr.	Herrouz Abdelhakim	MAA	Président	UKM Ouargla
Mme.	Benkherourou Chafika	MAA	Rapporteur	UKM Ouargla
Mr.	Mahjoub Mohamed Elbachir	MAA	Examineur	UKM Ouargla

**Année universitaire: 2021/2022**

# Remerciement

*Nous rendons grâce à Dieu qui nous a aidés à faire ce travail. Nous sommes profondément reconnaissants à nos parents pour leurs encouragements, leur soutien et les sacrifices qu'ils ont endurés.*

*Nous remercions notre Encadreur, Benkherourou Chafika, pour les efforts qu'elle a déployés pour nous aider, nous donner des conseils, nous encourager et corriger nos erreurs.*

*Nous remercions les membres du jury d'avoir accepté d'examiner nos travaux.*

*Nous remercions également tous les membres du corps professoral du Département de l'Informatique et Technologies de l'Information qui ont contribué à notre formation universitaire.*

*Enfin, nous tenons à remercier tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail. Vous trouverez ici notre sincère gratitude.*

# *Dédicace*

*Je dédie cet humble acte à mes parents la source de générosité et de patience tout au long de mon parcours scolaire. Dieu me les garde.*

*A mes frères et sœurs, ma famille et mes proches, chacun en son nom, qui m'ont toujours montré le bon chemin et qui m'ont toujours aidé.*

*Aux personnes qui m'ont accompagné pendant mes études universitaires, en particulier Nassim Benmohammed, mon mentor et source d'inspiration dans le domaine, à mes chers amis pour leurs encouragements et leur soutien constants, à tous ceux qui ont éclairé mon esprit avec des connaissances, même avec une seule lettre, à toutes les étapes de ma vie.*

*Mes sincères remerciements et ma gratitude à vous.*

*Gana Abdelmounaïme*

# *Dédicace*

*Je dédie cet ouvrage*

*A ma maman qui m'a soutenu et encouragé durant ces années d'études. Qu'elle trouve ici le témoignage de ma profonde reconnaissance.*

*A mes frères, mes grands-parents et Ceux qui ont partagé avec moi tous les moments d'émotion lors de la réalisation de ce travail. Ils m'ont chaleureusement supporté et encouragé tout au long de mon parcours.*

*A ma famille, mes proches et à ceux qui me donnent de l'amour et de la vivacité.*

*A tous mes amis qui m'ont toujours encouragé, et à qui je souhaite plus de succès.*

*A tous ceux que j'aime*

*Bouhenika Ahmed*

# Table Des Matières

<i>Remerciement</i> .....	I
<b>Table Des Matières</b> .....	IV
<b>Liste Des Figures</b> .....	VII
<b>Liste Des Tableaux</b> .....	VIII
<b>Liste Des Abréviations</b> .....	IX
<i>Résumé</i> .....	X
<b>Introduction Générale</b> .....	2
1.1 Introduction .....	2
1.2 Contexte .....	2
1.3 Problématique.....	3
1.4 Motivations.....	3
1.5 Structure du mémoire .....	3
<b>Chapitre 1: Généralités sur les bases de données.</b> .....	6
1. Introduction:.....	6
2. Les bases de données.....	6
2.1 Définition:.....	6
2.2 Le système de gestion de bases de données: .....	7
2.3 Modèles de bases de données: .....	9
3. La base de données relationnelle.....	9
3.1 Définition d'une base de données relationnelle:.....	9
3.2 Les SGBD relationnels: .....	10
3.2.1 Des objectifs fondamentaux d'un SGBDR : .....	11
3.3 Le modèle relationnel: .....	11

3.4 Les langages de manipulation de données relationnels:.....	12
3.4.1 Algèbre relationnelle ou langages algébriques .....	12
3.4.2 Le calcul relationnel:.....	12
3.5 Le langage SQL (Structured Query Language) .....	12
3.6 Propriétés "ACID": .....	13
4. Les limites des bases de données relationnelles:.....	14
5. Le futur des bases de données relationnelles .....	15
6. Conclusion.....	16
<b>Chapitre 2: Les bases de données NoSQL.....</b>	<b>18</b>
1. Introduction: .....	18
2. Big Data: .....	18
2.1 Définition:.....	18
2.2 L'importance du "Big Data": .....	20
2.2.1 Améliorer le fonctionnement des entreprises .....	20
2.2.2 Améliorer les résultats des opérations marketing .....	21
2.2.3 Recherche scientifique et amélioration des résultats médicaux.....	21
2.2.4 Compagnies pétrolières.....	21
3. NoSQL .....	21
3.1 Histoire: .....	21
3.2 Définition:.....	22
3.3 Fondements des systèmes NoSQL: .....	22
3.4 Fonctionnalités de NoSQL: .....	23
3.4.1 Scalabilité.....	23
3.4.2 Le théorème de CAP:.....	24
3.4.3 Les propriétés de BASE:.....	25
3.5 Caractéristiques générales des BD NoSQL: .....	25
3.6 Types de bases de données NoSQL:.....	26

---

## Table Des Matières

---

3.6.1 Base de données orientée clé / valeur: .....	26
3.6.2 Base de données orientée colonne: .....	27
3.6.3 Base de données orientée document: .....	28
3.6.4 Base de données orientée graphe: .....	29
4. SQL contre NoSQL: .....	30
5. Passage des bases de données relationnelles aux bases de données NoSQL.....	31
6. Conclusion.....	32
<b>Chapitre 3: Implémentation de la base de données par MongoDB.....</b>	<b>34</b>
1. Introduction .....	34
2. Pourquoi MongoDB? .....	34
3. MongoDB :.....	35
3.1 Commandes MongoDB de base .....	36
4. Différence entre SGBDR et MongoDB : .....	37
5. ENVIRONNEMENT DE TRAVAIL .....	38
5.1 Ressources matérielles :.....	38
5.1.1 Configuration requise: .....	38
5.2 Ressources logicielles:.....	38
5.3 Installations et configurations.....	39
6. Conception et implémentation: .....	39
6.1 Utilisateurs :.....	40
6.2 Liste de produits : .....	42
6.3 Commandes et des paiements :.....	43
7. DISCUSSION: .....	44
8. Conclusion.....	45
<b>Conclusion générale.....</b>	<b>47</b>
<b>Références Bibliographique .....</b>	<b>49</b>

# Liste Des Figures

Figure 1 Le mécanisme d'un SGBD. ....	8
Figure 2 Les sous-systèmes d'un SGBD. [8].....	8
Figure 3 Un exemple de base de données relationnelle. [7] .....	10
Figure 4. Les concepts fondamentaux d'une BDR [8].....	11
Figure 5. Une exemple de requête SQL.....	13
Figure 6. Les 3V du Big Data [25]. ....	19
Figure 7. Scalabilité horizontale et verticale. ....	24
Figure 8 Le théorème de CAP. ....	25
Figure 9. Les bases de données clé-valeur.....	27
Figure 10 Les bases de données orientées colonnes.....	28
Figure 11. Les bases de données orientées document. ....	29
Figure 12. Les bases de données orientées graphe. ....	30
Figure 13 Exemple d'une base de données de documents dans MongoDB.....	36
Figure 14 Digramme de classe pour la boutique en ligne de l'entreprise "4ALL" .....	40
Figure 15 Exemple de document pour les clients .....	41
Figure 16 Exemple de document pour un administrateur.....	42
Figure 17 Exemple de document produit.....	42
Figure 18 Exemple de document pour un paiement .....	43
Figure 19 Exemple de document d'un processus de commande.....	44



# Liste Des Tableaux

Tableau 1 Bases de données NoSQL Versus bases de données SQL.....	30
Tableau2 Commandes de base de MongoDB .....	37
Tableau 3 Différence entre SGBDR et MongoDB .....	37
Tableau 4. Ressources matérielles .....	38
Tableau 5 Ressources logicielles .....	38

# Liste Des Abréviations

**BD** : Base De Données

**LDD** : Langage de Définition de Données

**LMD** : Langage de Manipulation de Données

**BDR** : Base De Données Relationnelles.

**SQL** : Structured Query Language.

**SGBD** : Système de Gestion de Base de Données.

**SGBDR** : Système de Gestion de Base de Données Relationnelles.

**RDBMS** : Relational database management system.

**ACID**: Atomicity, Consistency, Isolation, Durability.

**NoSQL**: Not only SQL, Non Seulement Sql.

**CAP**: Consistency, Availability, Partition tolerance.

**BASE**: Basically Available, Soft-state, Eventual consistency.

**CRUD**: Create, Read, Update, Delete.

**JSON** : JavaScript Object Notation.

**BSON** : Binary JSON.

**XML** : Extensible Markup Language

## Résumé

Dans le domaine informatique, une base de données est un ensemble d'informations organisées de manière à pouvoir être facilement accessibles, modifiées et gérées. Les bases de données sont utilisées par les organisations pour stocker, récupérer et gérer des informations. Les bases de données ont connu un grand développement, notamment dans les années soixante du siècle dernier, avec l'émergence des bases de données relationnelles, qui ont été les pionnières pendant un certain temps. Dans le cadre ce mémoire, nous avons mis en évidence l'incapacité des bases de données relationnelles à faire face à l'émergence du concept de Big Data, qui a contraint les développeurs à adopter de nouvelles solutions, notamment les développeurs des grandes entreprises. Nous avons proposé une solution de base de données NoSQL qui a montré des résultats satisfaisants dans notre exemple de déploiement de base de données de boutique en ligne par MongoDB, grâce à sa grande évolutivité horizontale et ses fortes performances en raison de ne pas coller à une architecture statique. En conclusion, nous avons souligné que NoSQL peut ne pas être la meilleure option dans certains cas, car il présente des avantages et des inconvénients, Par rapport à les bases de données relationnelles, de sorte qu'une solution de base de données relationnelle ou une solution commune peut être prise.

**Mot clé:** NoSQL, MongoDB, Big data, Base de données relationnelle, SQL.

## Abstract

In the computer field, a database is a set of information organized in such a way that it can be easily accessed, modified and managed. Databases are used by organizations to store, retrieve and manage information. Databases have experienced a great development, especially in the sixties of the last century, with the emergence of relational databases, which were the pioneers for some time. As part of this thesis, we have highlighted the inability of relational databases to cope with the emergence of the concept of Big Data, which has forced developers to adopt new solutions, especially developers in large companies. We came up with a NoSQL database solution that showed good results in our online store database deployment example by MongoDB, thanks to its high horizontal scalability and strong performance due to not sticking to a static architecture. In conclusion, we pointed out that NoSQL may not be the best option in some cases, as it has advantages and disadvantages, compared to relational databases, so a relational database solution or a common solution can be taken.

**Keyword:** NoSQL, MongoDB, Big data, Relational database, SQL.

## ملخص

في مجال الكمبيوتر، قاعدة البيانات هي مجموعة من المعلومات المنظمة بطريقة يمكن الوصول إليها وتعديلها وإدارتها بسهولة. تستخدم المنظمات قواعد البيانات لتخزين المعلومات واسترجاعها وإدارتها. شهدت قواعد البيانات تطوراً كبيراً، خاصة في ستينيات القرن الماضي، مع ظهور قواعد البيانات العلائقية، التي كانت رائدة لبعض الوقت. كجزء من هذه الأطروحة، قمنا بتسليط الضوء على عدم قدرة قواعد البيانات العلائقية على التعامل مع ظهور مفهوم البيانات الضخمة، مما أجبر المطورين على تبني حلول جديدة، وخاصة المطورين في الشركات الكبيرة. لقد توصلنا إلى حل قاعدة بيانات NoSQL الذي أظهر نتائج جيدة في مثال نشر قاعدة بيانات المتجر عبر الإنترنت بواسطة MongoDB، وذلك بفضل قابلية التوسع الأفقي العالية والأداء القوي بسبب عدم الالتزام ببنية ثابتة. في الختام، أشرنا إلى أن NoSQL قد لا يكون الخيار الأفضل في بعض الحالات، حيث أن له مزايا وعيوب، مقارنة بقواعد البيانات العلائقية، لذلك يمكن اتخاذ حل قاعدة بيانات علائقية أو حل مشترك. الكلمات المفتاحية: NoSQL، MongoDB، البيانات الضخمة، قاعدة البيانات العلائقية، SQL.

---

# **Introduction Générale**

---

# Introduction Générale

## 1.1 Introduction

Les bases de données sont considérées comme l'un des domaines les plus importants de notre vie quotidienne, en particulier dans le domaine de la technologie informatique, car elles nous permettent de stocker d'énormes quantités de données qui ont été collectées de manière organisée et d'y accéder facilement par l'utilisateur autorisé. Les bases de données sont importantes pour la croissance des entreprises de plusieurs manières, telles que Facebook et Google, car elles permettent aux entreprises de prendre des décisions commerciales éclairées, de stocker et de récupérer efficacement des informations pertinentes, de faciliter l'analyse des données et de collecter et stocker des données client critiques à partir de diverses applications. La plupart des gens qui mentionnent les bases de données ne pensent qu'au modèle traditionnel qui comprend des tables composées de lignes et de colonnes. Alors que les systèmes de gestion de bases de données relationnelles dominent toujours la part du lion des données sur Internet, les modèles de données alternatifs sont devenus de plus en plus populaires ces dernières années. Les développeurs et les grandes entreprises en général cherchant des solutions aux limites du modèle relationnel. Chacun à ses avantages, ses inconvénients et ses cas d'utilisation uniques, et NoSQL peut être une solution à cette limitation grâce à ses nombreux avantages, nous y reviendrons dans ce mémoire, en soulignant ses avantages en plus d'apporter un éclairage sur MongoDB.

## 1.2 Contexte

Le contexte de la modélisation dans n'importe quel domaine est un contour de la limite du modèle. Où notre travail sera sur les bases de données, ce qui signifie que le contexte sur lequel nous travaillons est un modèle de base de données. Le modèle peut être dans une variété de domaines, domaine médical, domaine éducatif, domaine économique... Nous fournirons une solution de base de données de documents MongoDB pour résoudre notre problème de base de données d'entreprise axé sur le commerce électronique "4ALL". Nous parlerons de ce modèle en détail plus tard.

### **1.3 Problématique**

Les bases de données traditionnelles font face à de nombreux problèmes allant de la difficulté à traiter l'énorme quantité de ces données en termes de recherche et de classification en plus de leur faiblesse à s'adapter aux données non structurées. A partir de là, il est devenu nécessaire d'utiliser d'autres modèles pour stocker ces données et ne pas les limiter à une base de données relationnelle.

De ce point de vue, nous posons notre problème : Peut-on considérer NoSQL comme la meilleure solution ? Quels sont les avantages et les inconvénients ?

### **1.4 Motivations**

Lorsque la plupart des gens pensent au domaine des bases de données, ils sous-estiment souvent son importance par rapport à d'autres domaines tels que l'intelligence artificielle et la sécurité de l'information et pensent que c'est un domaine limité et sous-développé, mais au contraire, c'est une science très importante, et la plupart des sciences y sont étroitement liées. Les développeurs et les grandes entreprises en général cherchent à étudier et à rechercher dans ce domaine pour trouver de nouvelles voies et solutions pour s'adapter au développement technologique et à l'augmentation du volume de données. C'est pour cela nous sommes poussés à l'étudier et à mettre en évidence les bases de données non relationnelles en particulier.

### **1.5 Structure du mémoire**

Outre ce que nous avons suggéré dans le présent chapitre d'introduction générale, ce mémoire est organisé comme suit :

- Dans le premier chapitre, nous commencerons par définir les bases de données en général. Ensuite, nous parlerons de l'un des types les plus connus, les bases de données relationnelles, et aborderons certains de ses concepts de base. Ensuite, nous soulignerons ses limites et les problèmes auxquels il est confronté. Enfin, nous mentionnerons la solution proposée pour faire face à ce problème.

- Dans le deuxième chapitre, nous commencerons par définir les bases de données NoSQL en plus d'expliquer le concept de Big Data. Ensuite, nous parlerons du NoSQL et mentionnerons ses bases, ses caractéristiques et ses types. Ensuite, nous présentons une comparaison avec les bases de données traditionnelles, et enfin, nous discuterons de la façon de passer des bases de données traditionnelles aux bases de données NoSQL et mentionnerons notre suggestion de MongoDB.
- Dans le chapitre 3, nous commencerons par présenter les raisons du choix de MongoDB et souligner certains de ses concepts. Ensuite, nous ferons une comparaison entre celui-ci et SGBDR. Puis, après avoir expliqué le cadre de développement, nous présenterons les résultats de tous les modèles proposés et fournirons quelques exemples. Enfin, nous passerons en revue le pour et le contre et discuterons des résultats obtenus.

# **Chapitre 1**

---

**Généralités sur les bases de données.**

---



## **Chapitre 1: Généralités sur les bases de données.**

### **1. Introduction:**

Le terme bases de données est apparu bien avant l'existence des ordinateurs. Une fois que la race humaine a commencé à compter, à écrire et à trouver des moyens de mesurer et de stocker des biens, le besoin de collecter des informations sur leurs avoirs s'est également accru.

Jusqu'aux années 1960, l'organisation classique des données créait de nombreuses complexités jusqu'à ce que l'idée de distinguer et de traiter les données émerge en 1965. Vers la fin des années 1960, on commence à voir émerger le premier SGBD (Système de Gestion de Base de Données) : Réseau et systèmes hiérarchiques. Ce n'est qu'en 1970 qu'apparaît la deuxième génération de SGBD : les systèmes relationnels [1]. Ces développements technologiques ont résolu les problèmes liés au travail avec des fichiers et présentent les avantages suivants :

- Standardisation de la saisie et standardisation du traitement (tous les résultats de la consultation sous forme de listes et de tableaux)
- Validation instantanée des données.
- Échange de données entre plusieurs processus, ce qui signifie moins de redondance de données.

Au cours de ce chapitre, nous clarifierons ce que sont les bases de données et leurs concepts les plus importants, en soulignant leurs types, afin de montrer les complexités et les solutions proposées.

## **2. Les bases de données**

### **2.1 Définition:**

En informatique, une base de données (Abr. : « BD » ou « BDD ») est un lot d'informations stockées dans un dispositif informatique. Les technologies existantes permettent d'organiser et de structurer la base de données de manière à pouvoir facilement manipuler le contenu et stocker efficacement de très grandes quantités d'informations [2].

L'organisation logique des données se fait selon un modèle de données. La structure physique des fichiers comporte des index destinés à accélérer les opérations de recherche et de tri. L'organisation logique est indépendante de la structure physique. Une modification de l'organisation logique ne provoque pas de modification de la structure physique et vice-versa [4].

La motivation fondamentale de l'organisation des données est de permettre d'effectuer des inférences (requêtes) sur celles-ci. Le modèle de données relationnel est aujourd'hui le plus utilisé parce qu'il est formellement démontré que ce type de représentation permet de résoudre toutes requêtes, contrairement aux modèles hiérarchiques et réseau [4].

## **2.2 Le système de gestion de bases de données:**

Le logiciel qui manipule les bases de données est appelé système de gestion de base de données (SGBD). C'est un ensemble de logiciels qui manipulent le contenu des bases de données. Il permet d'organiser, de contrôler, de consulter et de modifier la base de données. Les opérations sont parfois formulées dans un langage de requête tel que SQL – le plus connu et employé pour les modèles relationnels [4].

Les mécanismes du système de gestion de base de données visent à assurer la cohérence, la confidentialité et la pérennité du contenu des bases de données. Le logiciel refusera qu'un usager modifie ou supprime une information s'il n'y a pas été préalablement autorisé ; il refusera qu'un usager ajoute une information si celle-ci existe dans la base de données et fait l'objet d'une règle d'unicité; il refusera également de stocker une information qui n'est pas conforme aux règles de cohérence telles que les règles d'intégrité référentielle dans les bases de données relationnelles [7].

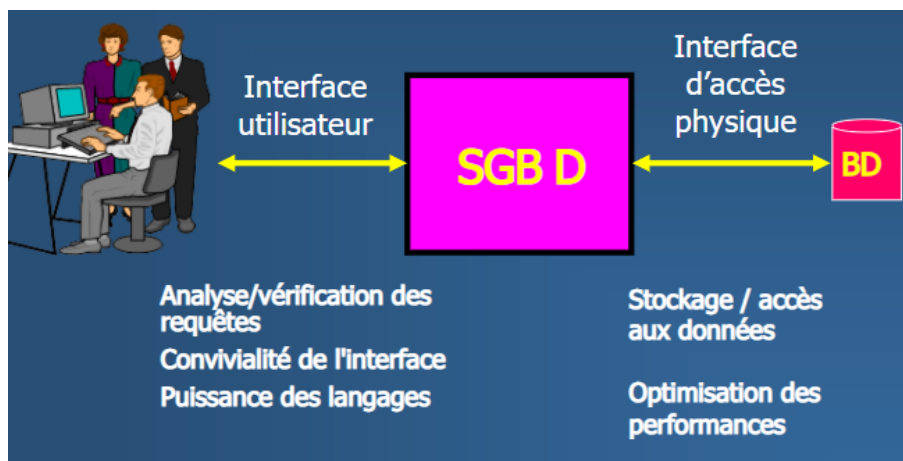


Figure 1 Le mécanisme d'un SGBD.

Le SGBD est divisé en trois sous-systèmes :

- **Système de gestion de fichiers** : Il permet de stocker des informations sur un support physique [9].
- **SGBD interne** : gère l'agencement des informations.
- **SGBD externe** : représente l'interface avec l'utilisateur.

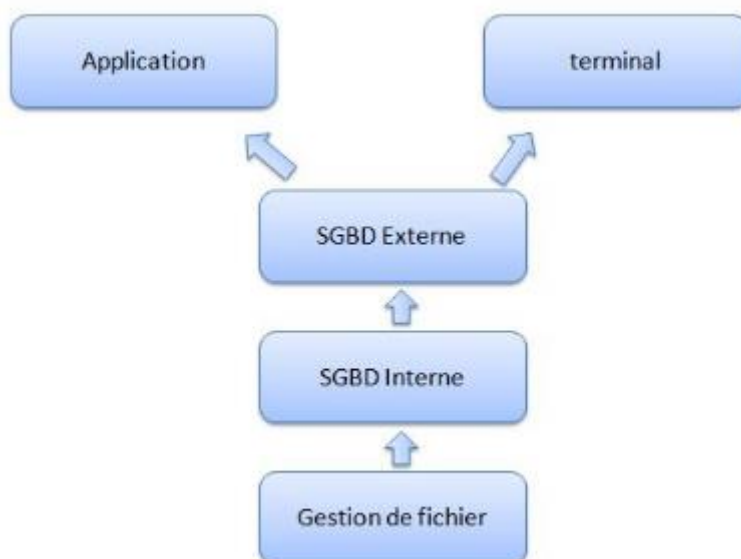


Figure 2 Les sous-systèmes d'un SGBD. [8]

Le système de gestion de base de données adapte automatiquement les index lors de chaque changement effectué sur une base de données et chaque opération est inscrite dans un journal contenu dans la base de données, ce qui permet d'annuler ou

de terminer l'opération même en cas de crash informatique et ainsi garantir la cohérence du contenu de la base de données.

IBM DB2, Oracle Database, MySQL et Microsoft SQL Server sont les systèmes de gestion de base de données les plus populaires.

### **2.3 Modèles de bases de données:**

Un modèle de base de données illustre la structure logique d'une base de données, y compris les relations et les contraintes qui déterminent comment les données peuvent être stockées et accessibles. Les modèles de base de données individuels sont conçus en fonction des règles et concepts du modèle de données plus général adopté par les concepteurs. La plupart des modèles de données peuvent être représentés par un diagramme de base de données [6].

Parmi les types de modèles de bases de données, nous mentionnons :

- Modèle de base de données hiérarchique
- Modèle relationnel
- Modèle réseau
- Le modèle relationnel-objet
- Modèle entité-association
- Modèle de base de données orientée objet
- Modèle document
- Modèle entité-attribut-valeur
- Schéma en étoile

## **3. La base de données relationnelle**

### **3.1 Définition d'une base de données relationnelle:**

Une base de données relationnelle est une collection de données organisées sous la forme de tables définies de façon formelle, à partir desquelles les données sont accessibles. Les bases de données relationnelles sont basées sur le modèle relationnel. Dans une base de données relationnelle, chaque ligne de la table est un enregistrement avec un identifiant unique appelé clé. Les colonnes de la table contiennent les attributs des données, et chaque enregistrement a généralement une

valeur pour chaque attribut, ce qui facilite l'établissement des relations entre les points de données [10].

En plus d'être relativement faciles à créer et pratiques pour y accéder, les bases de données relationnelles ont également l'avantage d'être faciles à étendre. Une fois la base de données créée, une nouvelle donnée peut être ajoutée sans qu'il ne soit nécessaire de modifier les applications en place.

Lorsqu'on crée une base de données relationnelle, on peut définir le domaine des valeurs dans une colonne et contraindre ce qui s'applique aux valeurs des données. La définition d'une base de données relationnelle donne une table de métadonnées or une description formelle des tables, colonnes, domaines et contraintes.

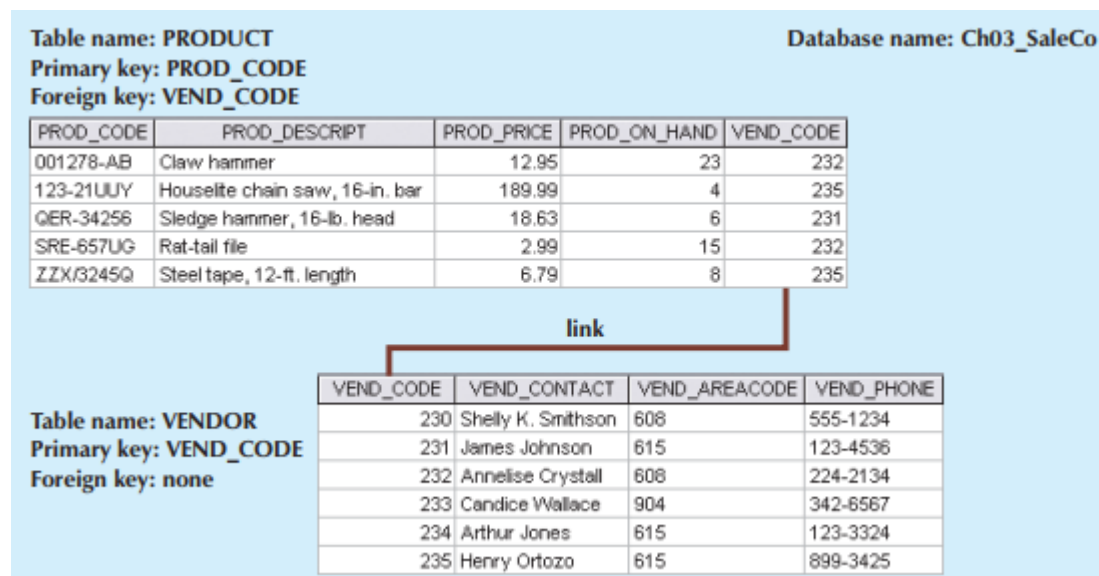


Figure 3 Un exemple de base de données relationnelle. [7]

### 3.2 Les SGBD relationnels:

Un système de gestion de BD relationnelles (SGBDR) est un logiciel standard qui repose sur les principes du modèle relationnel et mettent au premier plan les relations entre les données. Celles-ci sont organisées en tables à deux dimensions [10]. Un SGBDR propose les trois principales fonctions suivantes :

- La définition des données sous forme de relations.
- La manipulation des données par un langage déclaratif.
- L'administration des données.

3.2.1 Des objectifs fondamentaux d'un SGBDR :

- Organisation structurée des données dans des tables interconnectées.
- Implémentation d'un langage relationnel ensembliste permettant à l'utilisateur de décrire aisément les interrogations qu'il souhaite effectuer sur les données.
- Indépendance des données vis-à-vis des programmes.
- Gestion des opérations concurrentes pour permettre un accès multi-utilisateur sans conflit.
- Gestion de l'intégrité des données, de leur protection contre les pannes et les accès illicites.

3.3 Le modèle relationnel:

Le modèle le plus courant, le modèle relationnel a été introduit pour la première fois par Edgar Frank Codd au centre de recherche d'IBM en 1970 et attira immédiatement un intérêt considérable en raison de sa simplicité et de ses fondations mathématique [10]. Le modèle relationnel est un modèle abstrait utilisé pour organiser les données au sein d'une base de données [1].

Les éléments les plus fondamentaux du modèle relationnel sont les relations, que les utilisateurs et les SGBDR modernes reconnaissent comme des tables. Une relation est un ensemble de tuples, ou lignes dans une table. Chaque tuple partageant un ensemble d'attributs, ou colonnes :

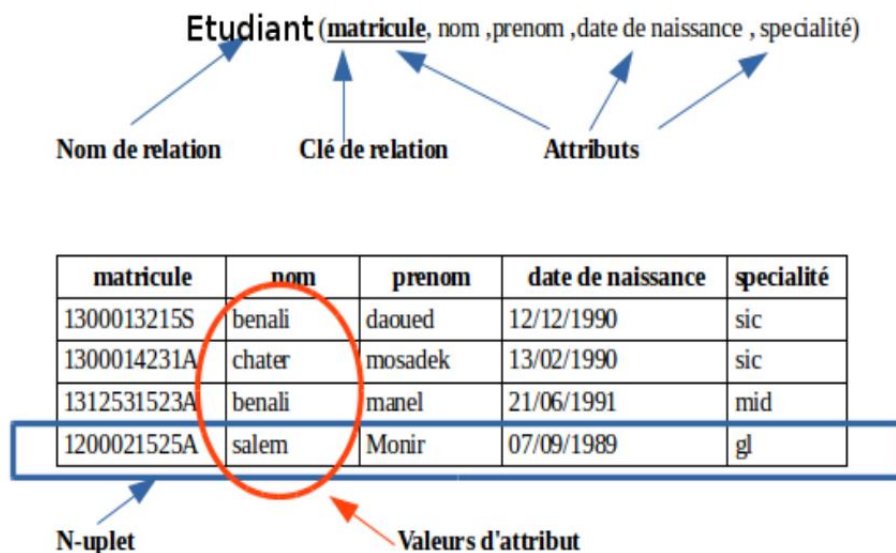


Figure 4. Les concepts fondamentaux d'une BDR [8].

### 3.4 Les langages de manipulation de données relationnels:

Le modèle relationnel a été à l'origine proposé avec deux langages de manipulation de données (LMD), l'algèbre relationnelle et le calcul relationnel, qui ont fixé l'ensemble des fonctions que tout LMD relationnel doit offrir.

SQL est le langage de manipulation de données relationnelles le plus populaire, de sorte que les bases de données relationnelles sont généralement écrites en SQL (Structured Query Language).

#### 3.4.1 Algèbre relationnelle ou langages algébriques

L'algèbre relationnelle est le langage interne d'un SGBD relationnel. Ce langage est constitué d'un ensemble d'opérations permettant de manipuler des relations (tables et colonnes). Son principe est de créer de nouvelles tables à partir de tables existantes, et ces nouvelles tables deviennent des objets utilisables immédiatement [7].

Cette algèbre se compose de facteurs pour manipuler les relations. Ces opérateurs sont regroupés en deux familles : les opérateurs ensemblistes et les opérateurs relationnels.

#### 3.4.2 Le calcul relationnel:

Le calcul des tuples est constitué des langages issus du calcul des prédicats de la logique du premier ordre. Deux adaptations de ce calcul au modèle relationnel ont été proposées [2], qui toutes deux ont conduit à des langages utilisateurs :

- Le calcul de tuples: construits à partir de la logique des prédicats, et dans lesquels les variables manipulées sont des tuples.
- Le calcul de domaine : également construit à partir de la logique des prédicats, mais en faisant varier les variables sur les domaines des relations.

### 3.5 Le langage SQL (Structured Query Language)

Le langage SQL (Structured Query Language) SQL est un langage informatique déclaratif non procédural utilisé pour exploiter des bases de données. Il permet de façon générale la définition, la manipulation et le contrôle de sécurité de données.

Dans la pratique, le langage SQL est utilisé pour créer des tables, ajouter des enregistrements sous forme de lignes, interroger une base de données, la mettre à jour, ou encore gérer les droits d'utilisateurs de cette base de données. Il est bien supporté par la très grande majorité des systèmes de gestion de base de données (SGBD). Créé au début des années 1970 par Donald D. Chamberlin et Raymond F. Boyce [11].

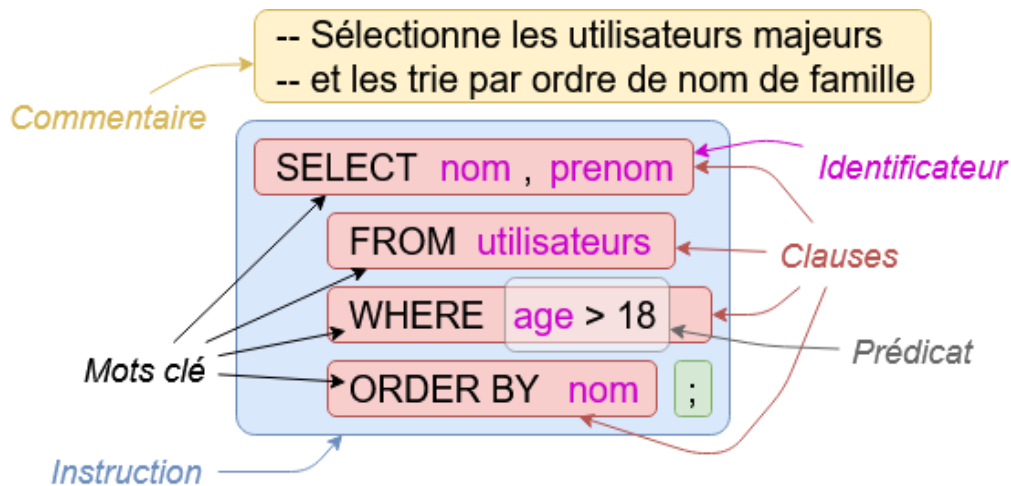


Figure 5. Une exemple de requête SQL.

### 3.6 Propriétés "ACID":

ACID définit les règles des transactions pour garantir la cohérence de la BD. C'est indispensable pour gérer les requêtes provenant de différentes applications [12].

**Atomicité** : cela signifie que les mises à jour de la base de données doivent être atomiques, c'est-à-dire, lorsqu'une transaction est effectuée, elle doit être exécutée entièrement ou pas du tout.

**Cohérence** : Avant et après l'exécution d'une transaction, les données d'une base doivent toujours être dans un état cohérent. La cohérence est établie par les règles fonctionnelles.

**Isolation** : les changements intermédiaires, apportés à la base par la transaction en cours, ne sont pas vus par les autres transactions exécutées en parallèle depuis d'autres tâches avant la validation.

**Durabilité** : Une transaction confirmée le demeure définitivement, peu importe les différents imprévus (panne d'électricité, panne d'ordinateur etc.).



#### 4. Les limites des bases de données relationnelles:

Les bases de données relationnelles sont les plus utilisées pour la gestion d'applications classiques. Cependant, il est important de garder à l'esprit que les bases de données relationnelles peuvent parfois être lentes et non évolutives notamment avec l'émergence de domaines applicatifs non traditionnels tels que la CAO (Conception Assistée par Ordinateur), le génie logiciel, et applications géographiques manipulant de très grandes quantités de données.

Parmi les problèmes les plus importants, nous mentionnons:

##### ❖ Types de données

Les types de données disponibles de modèle relationnel sont limités et non extensibles et le modèle relationnel est parfois peu adapté au stockage et à l'interrogation de certains types de données (données hiérarchiques, faiblement structurées, semi-structurées).

##### ❖ Langage de manipulation

Un autre inconvénient est lié au fait que le langage SQL n'est pas un langage complet. Le développement d'une application autour d'une base relationnelle nécessite l'utilisation d'autres langages (Java, C,...) et pose le problème de concordance des types entre le SGBD et le langage de développement.

##### ❖ Problème lié à l'application des propriétés "ACID"

Les propriétés ACID entraînent de sérieux surcoûts en latence, accès disques, temps CPU (verrous, journalisation, etc.) [15].

##### ❖ Surcharge sémantique

Le modèle relationnel s'appuie sur un seul concept (la relation) pour modéliser à la fois les entités et les associations entre ces entités [14].

**❖ Scalabilité limitée**

Atteinte par les groupes comme Yahoo, Google, Facebook, etc. Cette limite est la plus gênante pour les entreprises. Les SGBD Relationnels ne sont pas adaptés aux environnements distribués requis par les volumes gigantesques de données et par les trafics tout aussi gigantesques générés par ces sites [3].

**❖ Incompatibilités de types (“impedance mismatch”)**

Elles apparaissent lors de l’utilisation de SQL dans un langage hôte (pour la représentation des nombres, par exemple). Des conversions de types sont nécessaires, et peuvent s’avérer coûteuses en temps et parfois difficiles à mettre en œuvre [3].

**❖ Pas de schéma de base de données hiérarchique**

Les bases de données relationnelles n’offrent pas la possibilité d’implémenter des schémas de base de données avec des classes hiérarchiquement structurées. Des concepts tels que les entités subordonnées qui héritent de propriétés d’entités supérieures ne peuvent pas être implémentés avec elles.

**❖ Segmentation des données**

Le principe de base des systèmes de bases de données relationnelles, qui consiste à diviser l’information en tables séparées (normalisation), conduit inévitablement à une segmentation des données. Tout ce qui est connexes n’est pas nécessairement stocké ensemble. Cette conception de base de données entraîne des requêtes complexes sur plusieurs tables au niveau de l’application. Le nombre plus élevé de segments interrogés qui en résulte a généralement un impact négatif sur la performance [17].

**5. Le futur des bases de données relationnelles**

À l’heure du Big Data, les bases de données relationnelles ne sont plus adaptées car le traitement et le contrôle de grandes quantités de données deviennent un processus complexe pour un SGBDR. Pour prendre en charge les immenses

volumes de données, les stocker et les analyser, il est impératif de s'en remettre à de nouvelles solutions.

Partant de ce problème, la plupart des études se sont orientées vers l'invention d'une base de données caractérisée par :

- Une base de données “non relationnelle”.
- Permettant le stockage de données structurées, non structurées, semi-structurées ou polymorphique.
- Les jointures ne sont nécessaires, et le scaling est facilité.
- Traitement plus rapidement et plus efficace des données volumineuses (Big Data).

La base de données non relationnelle, également connue sous le nom de bases de données NoSQL "Not Only SQL (pas seulement SQL)", est apparue comme une solution pour gérer une grande base de données, garantissant la disponibilité, l'évolutivité et l'endurance des partitions.

## **6. Conclusion**

Les bases de données occupent aujourd'hui une place essentielle dans les technologies de l'information, notamment dans la gestion. Le modèle relationnel a fait ses preuves et s'est généralisé sur le marché grâce à son efficacité et sa pérennité depuis plus de 40 ans. Cependant, face aux évolutions informatiques de la dernière décennie et notamment aux volumes importants de données échangées, les systèmes RDBMS traditionnels ont montré des faiblesses très importantes. De ce fait, face à ces limitations, de nouveaux systèmes de gestion de données dits « NoSQL » viennent d'émerger en introduisant des Alternatives au modèle relationnel.

## **Chapitre 2**

---

### **Les bases de données NoSQL.**

---

## Chapitre 2: Les bases de données NoSQL.

### 1. Introduction:

Les exigences en matière de calcul et de stockage des applications telles que les analyses Big Data, la Business Intelligence et les réseaux sociaux nous ont amenés à passer de la base de données SQL à la base de données NOSQL.

Dans ce chapitre, nous suivrons le concept de Big Data, en plus d'expliquer les concepts de base des bases de données NoSQL et de leurs types, de souligner les différences les plus importantes entre elles et les bases de données relationnelles, et enfin de souligner la nature de leur relation avec des exemples.

### 2. Big Data:

#### 2.1 Définition:

Le terme «Big Data » ou "grosses données" est apparu il y'a quelques années pour expliquer un domaine issu d'une révolution dans la manière de traitement des données [23].

Big Data est un terme qui décrit des grands volumes de données à haute vitesse, complexes et variables qui requièrent des techniques et des technologies avancées pour permettre la capture, le stockage, la distribution, la gestion et l'analyse de l'information [16].

#### Exemples concrets de Big Data :

- WhatsApp compte plus d'un milliard d'utilisateurs, et plus de 42 milliards de messages et environ 1,6 milliard de photos sont échangés quotidiennement [19].
- Facebook gère plus de 50 milliards de photos de ses utilisateurs [19].
- Google gère environ 100 milliards de recherches par mois [19].

De nombreux responsables informatiques et autorités du secteur tendent à définir le Big Data selon trois grandes caractéristiques : Volume, Vitesse et Variété, soit les trois « V ».

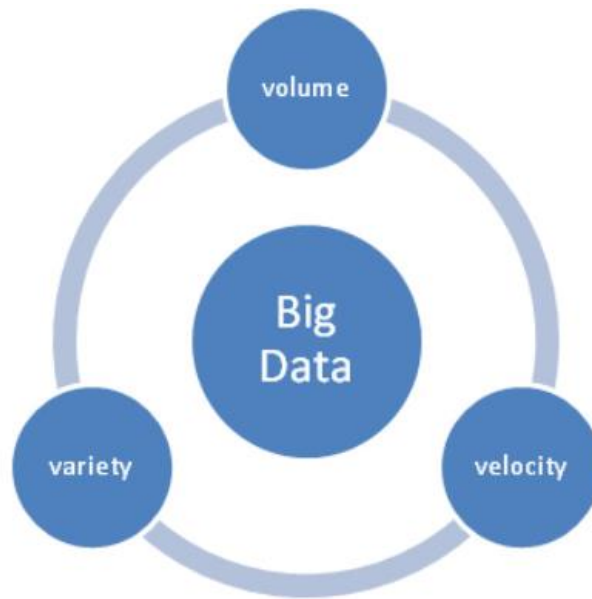


Figure 6. Les 3V du Big Data[25].

#### - Volume

Le volume décrit la quantité de données générées par des entreprises ou des personnes. Le Big Data est généralement associé à cette caractéristique. Les entreprises, tous secteurs d'activité confondus, devront trouver des moyens pour gérer le volume de données en constante augmentation qui est créé quotidiennement [24].

#### - Vitesse (rapidité) "Velocity"

La vitesse décrit la fréquence à laquelle les données sont générées, capturées, visualisées et partagées. Du fait des évolutions technologiques récentes, les consommateurs génèrent plus de données dans des temps beaucoup plus courts. [24]

#### - Variété "variety"

La prolifération de types de données provenant de sources comme les médias sociaux, les interactions Machine to Machine et les terminaux mobiles, crée une très grande diversité au-delà des données transactionnelles traditionnelles. Les données ne s'inscrivent plus dans des structures nettes [24].

Les nouveaux types de données incluent contenus, données géo spatiales, données de mesures, données mobiles, données issues de recherches, données de flux, données issues des médias sociaux, données texte et données issues du Web.

❖ A ce concept, on peut ajouter 4 nouveaux paramètres:

### - Véracité

Avoir beaucoup de données dans des volumes différents à grande vitesse est sans valeur si ces données sont incorrectes. Des données incorrectes peuvent causer beaucoup de problèmes aux organisations et aux consommateurs. Par conséquent, les organisations doivent s'assurer que les données sont correctes et que les analyses effectuées sur les données sont correctes. Surtout dans la prise de décision automatisée, où aucun humain n'est plus impliqué [25].

### - Variabilité

Le BigData est extrêmement variable. On définit la variabilité comme la "variance de sens dans le lexique". Pour effectuer des analyses de sentiment appropriées, les algorithmes doivent être capables de comprendre le contexte et de déchiffrer la signification exacte d'un mot dans ce contexte. C'est toujours très difficile [25].

### - Visualisation

C'est la partie difficile des données volumineuses. Faire en sorte que toutes ces données soient compréhensibles d'une manière facile à comprendre et à lire. Avec les bonnes analyses et visualisations, les données brutes peuvent être utilisées pour des données brutes, sinon les données brutes restent essentiellement inutiles [25].

### - Valeur

Toutes ces données disponibles créeront beaucoup de valeur pour les organisations, les sociétés et les consommateurs. Le Big Data signifie que les grandes entreprises et tous les secteurs tireront profit des données volumineuses [25].

## 2.2 L'importance du "Big Data":

### 2.2.1 Améliorer le fonctionnement des entreprises

Les entreprises utilisent le Big Data dans leurs systèmes pour améliorer leurs opérations, fournir un meilleur service client, créer des campagnes marketing ciblées et prendre d'autres mesures susceptibles d'augmenter les revenus et les bénéfices. Les entreprises qui utilisent efficacement le Big Data ont un avantage concurrentiel sur

celles qui ne le font pas, car elles sont capables de prendre des décisions commerciales plus rapides et plus précises [23].

### **2.2.2 Améliorer les résultats des opérations marketing**

Le Big Data fournit des informations plus claires lors de l'évaluation du comportement et des intérêts des clients. Les entreprises peuvent l'utiliser pour améliorer le marketing, la publicité et les promotions afin de cibler les clients en fonction de leurs envies et de leur comportement de consommation [23].

### **2.2.3 Recherche scientifique et amélioration des résultats médicaux**

Les chercheurs médicaux utilisent également les "Big Data" pour identifier les signes de maladie et les facteurs de risque. Les médecins les utilisent pour aider à diagnostiquer les maladies et les conditions médicales chez les patients. En outre, une gamme de données provenant des dossiers de santé électroniques, des sites de médias sociaux, d'Internet et d'autres sources fournit aux organisations de soins de santé et aux agences gouvernementales des informations toujours à jour sur les menaces ou les épidémies de maladies infectieuses [23].

### **2.2.4 Compagnies pétrolières**

Les mégadonnées dans le secteur de l'énergie aident les sociétés pétrolières et gazières à identifier les sites de forage potentiels et à surveiller les opérations des pipelines [23].

## **3. NoSQL**

### **3.1 Histoire:**

Carlo Strozzi a utilisé le terme NoSQL en 1998 pour nommer sa base de données relationnelle légère et open source qui n'exposaient pas l'interface SQL standard. Eric Evans, un employé de Rackspace, a réintroduit le terme NoSQL lorsque Johan Oskarsson de Last.fm a voulu organiser un événement pour discuter des bases de données distribuées open source. Le nom tentait d'étiqueter l'émergence d'un nombre croissant de magasins de données distribués non relationnels qui, souvent, n'essayaient pas de fournir les garanties ACID (atomicité, cohérence, isolation, durabilité), qui sont les attributs clés des systèmes de bases de données relationnelles classiques, tels que comme Sybase, IBM DB2, MySQL, Microsoft SQL Server, Postgres SQL, Oracle RDBMS, Informix et Oracle RDB [1].



### 3.2 Définition:

NoSQL est un type de système de gestion de base de données (SGBD) de nouvelle génération. Les bases de données NoSQL ont des schémas flexibles pour créer des applications modernes avec de grandes quantités de données et des charges élevées.

Souvent, les bases de données NoSQL sont classées en fonction de la manière dont elles stockent les données et relèvent de catégories telles que les magasins clé-valeur, les implémentations BigTable, les bases de données de magasins de documents et les bases de données de graphes. Les systèmes de base de données NoSQL se sont développés aux côtés de grandes sociétés Internet, telles que Google, Amazon, Microsoft et Meta (Facebook), qui avaient des défis très différents dans le traitement des données que les solutions RDBMS traditionnelles ne pouvaient pas gérer [5]. Les bases de données NoSQL sont souvent hautement optimisées pour les opérations de récupération et d'ajout et offrent souvent peu de fonctionnalités au-delà du stockage des enregistrements. La flexibilité réduite du temps d'exécution, par rapport aux systèmes SQL complets, est compensée par des gains significatifs d'évolutivité et de performances pour certains modèles de données [28].

Certaines bases de données NoSQL populaires: MongoDB, Apache Cassandra, amazon DynamoDB, Apache CouchDB, Apache HBase, Redis, OrientDB, Neo4j...

### 3.3 Fondements des systèmes NoSQL:

- Le « Sharding » : un partitionnement des données sur plusieurs serveurs un ensemble de techniques qui permet de répartir les données sur plusieurs machines pour assurer la scalabilité de l'architecture.
- Le « Consistent hashing » : un partitionnement des données sur plusieurs serveurs eux-mêmes partitionnés sur un segment, mécanisme de partitionnement (horizontal) dans lequel les objets données sont stockés sur des nœuds-serveurs différents en utilisant la même fonction de hachage à la fois pour le hachage des objets et le hachage des nœuds.
- Le « MapReduce » : MapReduce s'agit d'une approche de traitement de l'information distribuée qui prend une liste en entrée et en produit, une en retour. MapReduce a été défini en 2004 dans un article rédigé par Google. Le principe est simple: pour distribuer un traitement, Google a imaginé une

opération en deux étapes, tout d'abord, une attribution des opérations sur chaque machine (Map), suivie après traitement d'un rassemblement des résultats (Reduce).

- Le « MVCC » : pour « Contrôle de Concurrence Multi-Version », est un mécanisme permettant d'assurer le contrôle de concurrence, méthode de contrôle de concurrence couramment utilisée par les SGBD pour gérer des accès simultanés à la base de données avec mises à jour.
- Le « Vector-Clock » : ou horloges vectorielles permet des mises à jours concurrentes en datant les données par des vecteurs d'horloge Les ensembles de données répartis sur nœuds peuvent être lus et modifiés sur chaque nœud et aucune cohérence stricte n'est assurée par des protocoles de transactions distribuées.
- Le « Hadoop » : est une plateforme open source de la fondation Apache, ayant une capacité de gérer des données volumineuses, qui sont structurées et non structurées. Elle est conçue pour trouver une solution aux problèmes liés à la volumétrie et la variété des données en les traitants sur des différents serveurs simultanément. Cette architecture va offrir une puissance, et un stockage importants, les données vont être par la suite répliquées et réparties sur les machines du cluster, grâce à un système de réplication de façon à garantir une très haute disponibilité des données en cas de défaillance d'un ou de plusieurs serveurs.

### 3.4 Fonctionnalités de NoSQL:

#### 3.4.1 Scalabilité

Scalabilité est l'aptitude d'un système à conserver, maintenir son niveau de performance par augmentation des ressources matérielles. On distingue deux types de scalabilité :

**Scalabilité horizontale ou externe:** Le principe de la scalabilité horizontale consiste à simplement rajouter des serveurs en parallèle, c'est la raison pour laquelle on parle de croissance externe. On part d'un serveur basique et on rajoute de nouveaux serveurs identiques afin de répondre à l'augmentation de la charge [27].

**Scalabilité verticale ou interne:** Le principe de la scalabilité verticale consiste à modifier les ressources d'un seul serveur, comme par exemple le remplacement du

CPU par un modèle plus puissant ou par l'augmentation de la mémoire RAM. La croissance interne consiste au fait d'évoluer une machine dans le temps [27].

Le but des systèmes NoSQL est de renforcer la scalabilité horizontale, les SGBD NoSQL sont basés sur des systèmes innovants permettant la scalabilité horizontale consiste à simplement rajouter des serveurs dont la plupart d'entre eux sont Open Source.



Figure 7. Scalabilité horizontale et verticale.

### 3.4.2 Le théorème de CAP:

Le théorème de CAP est l'acronyme de « Coherence », « Availability » et « Partition tolérance ». Seuls deux des trois contraintes peuvent être respectés en même temps [2].

Pour que les SGBDs NoSQL puissent garantir une meilleure scalabilité horizontale, il faut respecter le principe de tolérance au partitionnement, ce qui exige l'abandon soit de la cohérence, soit de la haute disponibilité.

- « **Cohérence** » (**Cohérence**) : Tous les clients du système voient les mêmes données au même instant.
- « **Availability** » (**Haute disponibilité**) : Un système est dit disponible si toute requête reçue par un nœud retourne un résultat. Bien évidemment le nœud en question ne doit en aucun cas être victime de défaillance.
- « **Partition tolerance** » (**Tolérance à la partition**) : Un système est dit tolérant à la partition s'il continue à répondre aux requêtes de manière correcte même en cas de panne.

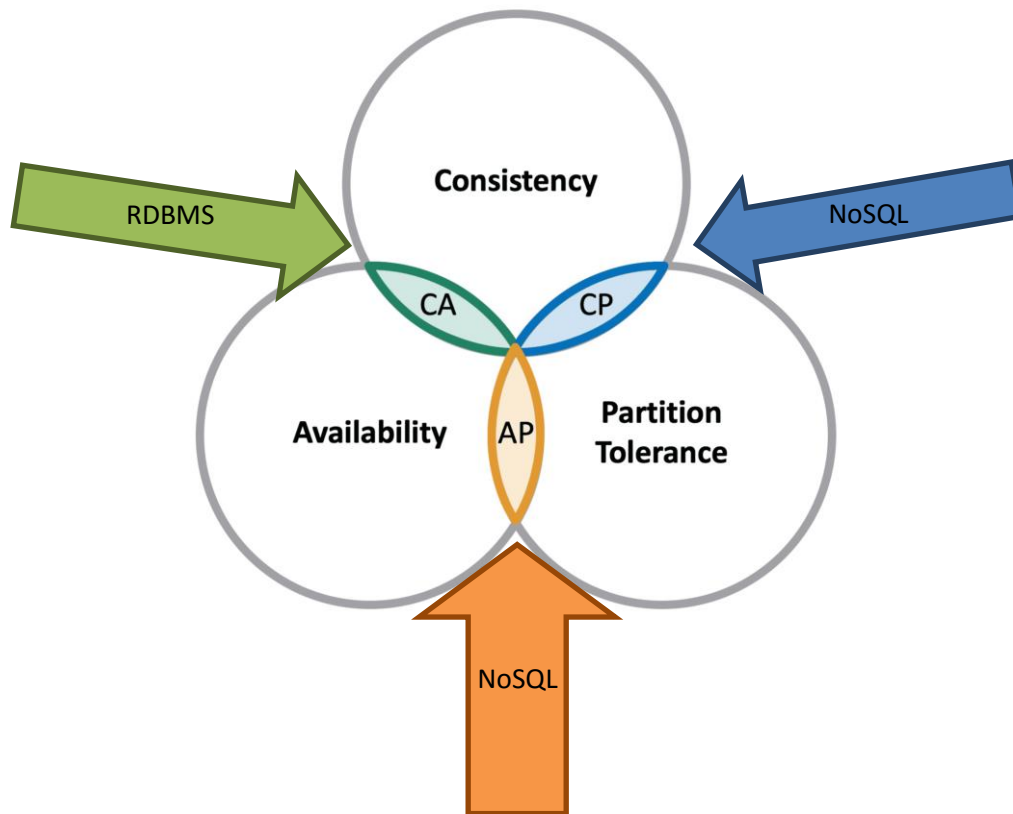


Figure 8 Le théorème de CAP.

### 3.4.3 Les propriétés de BASE:

Les SGBD NoSQL, selon le théorème CAP, privilégient la disponibilité ainsi que la tolérance à la partition plutôt que la cohérence, répondent aux propriétés de BASE. [26] Les caractéristiques de BASE sont fondées sur les limites que montrent les SGBD relationnelles :

- **Basically Available (Disponibilité basique)** : même en cas de désynchronisation ou de panne d'un des nœuds du cluster, le système reste disponible selon le théorème CAP.
- **Soft-state (Cohérence légère)** : indique que l'état du système peut changer à mesure que le temps passe, et ce sans action utilisateur.
- **Eventual consistency (Cohérence à terme)** : spécifie que le système sera consistant à mesure que le temps passe, à condition qu'il ne reçoive pas une action utilisateur entre temps.

### 3.5 Caractéristiques générales des BD NoSQL:

- ✓ Adoptent une représentation non relationnelle de données.

- ✓ Apportent une plus grande performance dans le contexte des applications Web avec des volumétries de données exponentielle.
- ✓ Utilisent une très forte distribution de ces données et des traitements associés sur de nombreux serveurs.
- ✓ Données de structures complexes ou imbriquées.
- ✓ Données distribuées : partitionnement horizontal des données sur plusieurs nœuds (serveurs) généralement par utilisation d'algorithmes MapReduce et/ou du Spark.
- ✓ Réplication des données sur plusieurs nœuds.
- ✓ Mode d'utilisation : Peu d'écritures, beaucoup de lectures [26].
- ✓ Elles privilégient souvent la disponibilité et le partitionnement à la cohérence (théorème CAP)
- ✓ Leurs langages de requête sont souvent plus complexes, mais plus flexibles que le SQL (Java, PHP)
- ✓ Leurs schémas de stockage sont dynamiques
- ✓ Elles ont une scalabilité verticale et horizontale (partitionnement des données et partage des calculs)

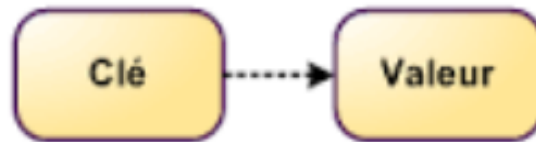
### 3.6 Types de bases de données NoSQL:

Les bases de données NoSQL sont principalement classées en quatre types: orientée clé / valeur, orientée colonne, orientée graphe et orientée document. Chaque catégorie a ses attributs et ses limites uniques. Aucune des bases de données spécifiées ci-dessus n'est meilleure pour résoudre tous les problèmes. Les utilisateurs doivent sélectionner la base de données en fonction de leurs besoins en produits [28].

#### 3.6.1 Base de données orientée clé / valeur:

Ces bases de données sont idéales pour accéder aux données par le biais d'une clé. La spécificité est que les données peuvent être stockées sans définir de schéma spécifique. Ces bases de données sont très efficaces pour la lecture et l'écriture, et conçues pour s'adapter massivement et proposer un temps de réaction extrêmement rapide. Les bases de données clé-valeurs sont les bases de données NoSQL les plus simples. Elles sont constituées d'une clé unique et d'un pointeur renvoyant à un certain type de données, créant des paires clé-valeur regroupées en une table de hachage. Ce type de base de données est utile pour des données analytiques. Elle ne possède aucune capacité de base de données, donc pour cela, l'utilisateur devra

uniquement se baser sur l'application. Les requêtes sont limitées à la valeur de la clé, ce qui permet d'offrir une grande vitesse de réponse, mais ce qui complique les mises à jour. Les références dans cette catégorie de bases de données sont Redis, Riak, Oracle NoSQL et Microsoft Azure Table Storage [28].



*Figure 9. Les bases de données clé-valeur.*

### **3.6.2 Base de données orientée colonne:**

Ce type de modèle de BD est une extension des bases de données clé-valeurs, avec des colonnes et des caractéristiques des bases de données relationnelles. Elles sont peu adaptées aux mises à jour fréquentes et préfèrent les ajouts. Les relations ne sont pas supportées, ce qui implique la création d'une structure en réseau pour les données connectées, ce qui n'est pas la solution la plus efficace. Elles sont utilisées quand une grande quantité de données doit être stockée et qu'une grande disponibilité est demandée. La plupart des systèmes de type orienté colonne peuvent supporter les données géospatiales, mais surtout avec des données intégrées et des requêtes simples avec une demande de réponse rapide HBase, Cassandra, Big Table, DynamoDB et Accumulo sont des exemples de bases de données orientées colonnes qui sont utilisées par Amazon, Google et Facebook [28].

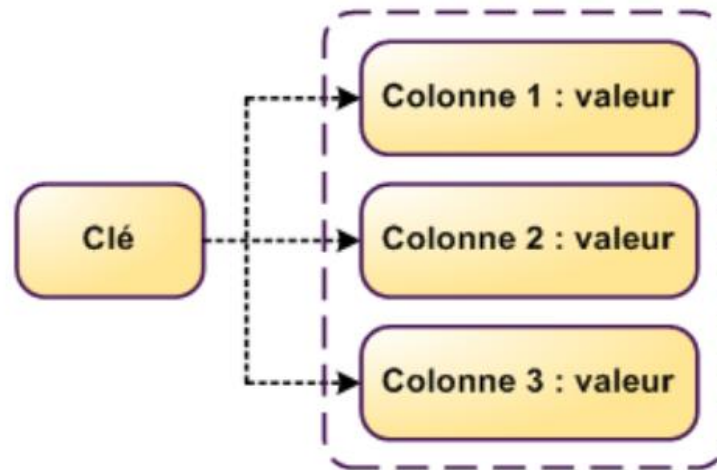


Figure 10 Les bases de données orientées colonnes.

### 3.6.3 Base de données orientée document:

Les bases de données orientées documents consistent à gérer une grande quantité de données. Une clé est assignée à un document qui peut lui-même posséder plusieurs couples clé-document intégrés. Un document peut être importé sous différents formats standards comme XML (Extensible Markup Language), JSON (JavaScript Option Notation) ou BSON (Binary JSON). La différence aux bases de données clé-valeurs est que les recherches peuvent être effectuées sur le contenu des documents et pas uniquement leur clé. La présence des documents intégrés permet d'effectuer des recherches avancées et ne requiert pas de schémas prédéfinis [28]. Les BD orientées documents permettent à un document d'être décrit par un grand nombre de valeurs et peuvent supporter un schéma flexible. Elles peuvent accueillir une grande quantité de données sous des formats fortement différents. La relation entre les documents est représentée par l'intégration de ceux-ci, il n'existe pas de relations extérieures. Un document est indépendant d'un autre qui ne lui est pas intégré. Ainsi, pour des requêtes qui interrogent les documents d'une même collection (ensemble de document), la réponse sera très rapide. L'utilisation de document XML peut améliorer les BD orientées documents en offrant des fonctionnalités complémentaires (XQuery, Xpath, XPointer) et en offrant le principe de relation qui n'existait pas sans le XML. On retrouve principalement CouchDB et MongoDB comme solutions basées sur le concept de base documentaire et Twitter est un d'utilisateur.

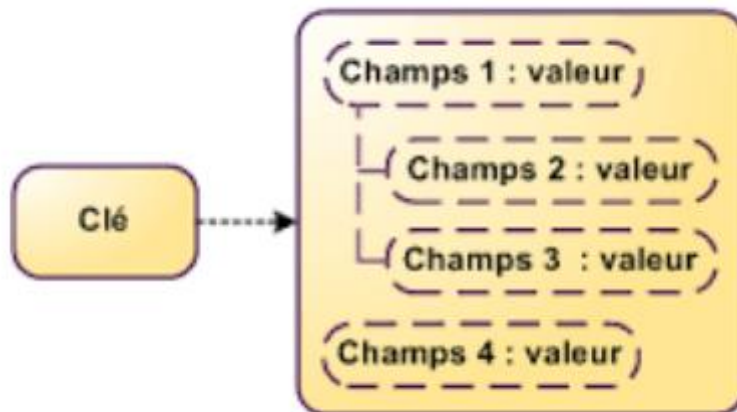


Figure 11. Les bases de données orientées document.

#### 3.6.4 Base de données orientée graphe:

La base de données orientée graphe est un type de base de données NoSQL utilisant la théorie des graphes pour stocker, cartographier et effectuer des requêtes sur les relations entre les données. Les bases de données orientées graphes sont constituées de nœuds et de bords. Chaque nœud représente une entité, et chaque bord représente une connexion entre les nœuds. Les bases de données graphes gagnent en popularité dans le domaine des analyses d'interconnexions. Ce modèle est capable de gérer des données relationnelles. Les BD graphes sont utiles pour les données constituées en réseau comme les routes, les réseaux sociaux [28].

Contrairement aux autres types de bases de données NoSQL, les BD orientées graphes ne sont pas optimales pour les grandes quantités de données si celles-ci ne sont pas assez connectées.

Oracle Graph, Neo4J, Arango et OrientDB sont des exemples de ce modèle. Elles sont utilisées par Walmart et Ebay.



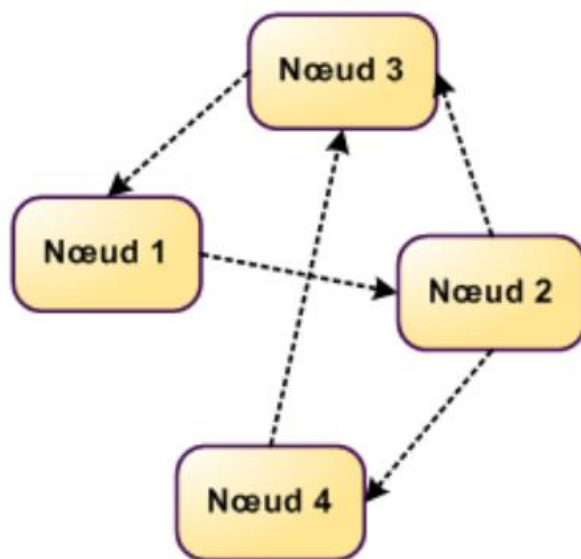


Figure 12. Les bases de données orientées graphe.

#### 4. SQL contre NoSQL:

Dans le tableau suivant, les différences les plus importantes entre les bases de données relationnelles et les bases de données non relationnelles :

Base de données NoSQL	Base de données SQL
Système de base de données non relationnelle ou distribuée.	Système de gestion des bases de données relationnelles (SGBDR)
schéma dynamique	schéma fixe ou statique
Données non structurées ou semi-structurées	Données structurées (tableaux)
Scalabilité horizontale et pas trop horizontale	Scalabilité vertical
Requêtes limitées	Requêtes flexibles (jointures)
Scalable Lecture /écriture	Scalable lecture
CAP (consistency, availability, partition tolerance)	Propriété ACID
SQL non requis	SQL requis

Tableau 1 Bases de données NoSQL Versus bases de données SQL

## 5. Passage des bases de données relationnelles aux bases de données NoSQL

Dans notre cas, nous étudions une entreprise active dans le marketing et la publicité. Au début de l'entreprise, il était facile pour RDBMS de traiter ses données, mais avec sa croissance explosive, il est devenu difficile pour RDBMS d'accomplir la tâche.

Voici le rôle des bases de données NoSQL qui sont plus efficaces pour travailler dans notre cas.

- De ce contexte, la problématique survenue : "Faut-il repenser la base de données à partir de zéro, ou pouvons-nous faire la migration ?"

S'il existe des bases de données relationnelles qui stockent du contenu, des documents, des fichiers ou qui contiennent des données non structurées, il faudrait penser à déplacer ces bases de données vers des bases de données NoSQL. Notre choix sera basé sur les avantages en termes de coûts, les performances, l'évolutivité, la pérennité des modifications, la réduction des travaux de conversion et une prise en charge étendue de l'analyse.

Apache Cassandra et MongoDB sont les bases de données NoSQL les plus utilisées. Cependant, il existe un nombre limité d'outils de migration de SGBDR vers MongoDB ou de SGBDR vers Cassandra disponibles sur le marché. Cette migration peut être réalisée à l'aide d'outils ETL ou de moteurs de transformation personnalisés.

Il y a des considérations importantes à garder à l'esprit lors de la migration de RDBMS vers NoSQL. La migration peut être difficile. Il a donc souvent besoin de l'aide d'experts.

En fin de compte, refaire à nouveau les bases de données peut être la dernière solution, mais parfois cela peut être la meilleure solution malgré ses inconvénients, notamment le fait qu'elle peut être très coûteuse et ne prend pas peu de temps.

## **6. Conclusion**

Nous avons discuté de différents aspects du SGBD NoSQL et nous avons également discuté du SGBDR pour trouver les différences avec le stockage NoSQL. Dans ce chapitre, nous avons mis en évidence le problème de l'incapacité des bases de données relationnelles à répondre aux besoins de l'entreprise commerciale, ce qui nous a poussé à proposer une solution pour les bases de données non relationnelles. Comme le SGBD NoSQL a différents modèles en fonction des cas d'utilisation cibles, il a également des fonctionnalités différentes. Cette technologie NoSQL a été principalement développée pour le traitement de données non structurées et volumineuses (big data).

## **Chapitre 3**

---

# **Implémentation de la base de données par MongoDB.**

---

## **Chapitre 3: Implémentation de la base de données par MongoDB.**

### **1. Introduction**

SGBDR peut faire face à de nombreux problèmes avec l'émergence du terme Big Data et nous avons été confrontés au problème de l'incapacité des bases de données relationnelles à répondre aux besoins de l'entreprise commerciale, ce qui nous a poussé à proposer une solution pour les bases de données non relationnelles.

Ce chapitre présente la solution de base de données non relationnelle MongoDB, qui appartient à la famille des bases de données orientées document, en indiquant les étapes pour l'installer, ainsi que ses exigences et comment l'utiliser.

### **2. Pourquoi MongoDB?**

MongoDB présente plusieurs avantages majeurs. Tout d'abord, cette base de données NoSQL orientée document se révèle très flexible et adaptée aux cas d'usage concrets d'une entreprise.

Les requêtes " Ad hoc " permettent par ailleurs de retrouver des champs spécifiques au sein des documents. Notons aussi la possibilité de créer des index pour améliorer la performance des recherches. N'importe quel champ peut être indexé [21].

MongoDB est un excellent choix pour la mise en place de projets Web qui se basent sur une importante masse de données déstructurées.

MongoDB prend en charge les transactions ACID (atomicité, cohérence, isolation et durabilité) et son offre Cloud est conforme à la norme PCI DSS (PaymentCardIndustry Data Security Standard). C'est l'une des meilleures options NoSQL pour le commerce électronique [30].

MongoDB enregistre également les documents dans un format de type JSON, ce qui signifie qu'il est très simple de convertir les requêtes et les résultats dans un format que le code frontal comprend. Cela supprime le besoin d'écrire des mappers de relation d'objet (ORM) [21].

Un autre avantage est la possibilité de créer des ” replica sets “ constitués de deux instances MongoDB ou plus. Chaque membre peut jouer le rôle de réplique secondaire ou primaire à tout moment [21].

Le travail orienté documents est prédestiné à de nombreux types de documents, notamment s'il ne suit pas de schéma prédéfini. Ces documents variés doivent être stockés et exploités avec rapidité [18].

Le système de bases de données permet, de plus, un échelonnage horizontal, presque illimité, qui peut être réparti sur plusieurs serveurs sans compromettre le fonctionnement [20].

Pour assurer la sécurité et la disponibilité des données sur le long terme, des copies peuvent être effectuées facilement avec MongoDB, et mises à disposition sur plusieurs serveurs [18].

Enfin, MongoDB est utile lors de la création d'applications sur tous les fronts commerciaux, en particulier les applications mobiles avec une évolutivité sans fin. La façon dont les documents sont stockés facilite l'accès et le partage des informations générées localement sur les réseaux. Cela le rend idéal pour développer des applications de données simple largeur [30].

### **3. MongoDB :**

C'est l'entreprise 10gen qui est à l'origine du développement de MongoDB (de l'anglais humongous = « gigantesque, énorme ») en 2007. C'est une base de données open source, centrée sur les documents. MongoDB a été mis sur le marché en l'espace de deux ans seulement et il aura fallu peu de temps pour que celui-ci devienne une des bases de données NoSQL les plus populaires.

MongoDB a été codé en langage de programmation C++ et enregistre les données en format BSON (Binary JSON), basé sur le format de JSON (JavaScript Object Notation)[5]. Ainsi, différents types de données JavaScript sont supportés et c'est pourquoi MongoDB est le choix optimal pour les plateformes Node.js. Les bases de données MongoDB contiennent un ou plusieurs ensembles de données qui gèrent différents documents avec de nombreux champs de données divers et variés [17].

La connexion au serveur de la base de données de MongoDB peut être effectuée de différentes manières. MongoDB est compatible avec presque tous les systèmes d'exploitation de Linux, Windows et Mac.

Un document représente l'unité basique de MongoDB, composé d'un ensemble ordonné de paires clé-valeur [30].

Une collection est un ensemble de documents équivalent à une table SQL. Les documents d'une même collection peuvent avoir des formes très différentes [5].

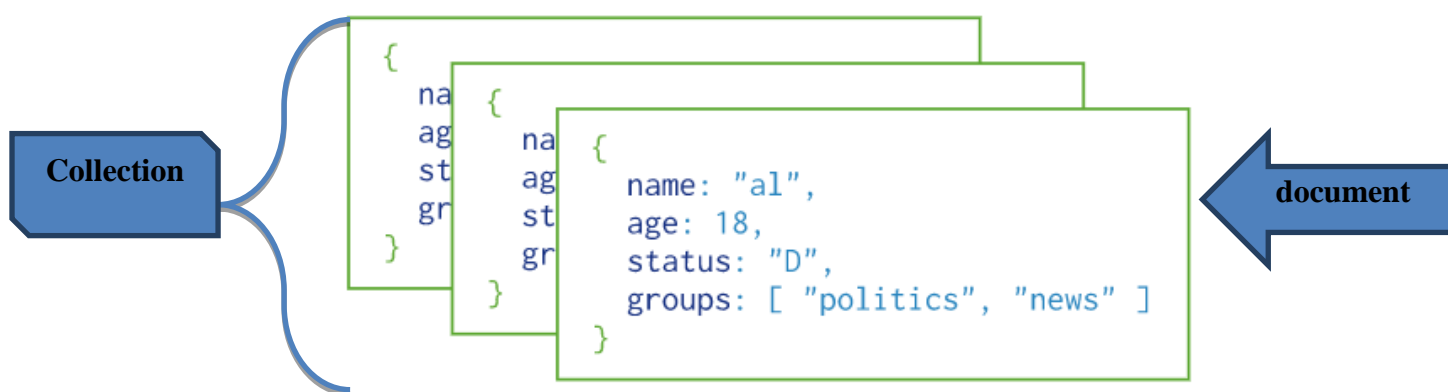


Figure 13 Exemple d'une base de données de documents dans MongoDB

### 3.1 Commandes MongoDB de base

db. help ()	obtenir une liste de commandes
show dbs	affiche une liste de toutes les bases de données sur le serveur
use myTestDB	créer une nouvelle base de données "myTestDB"
Db	connaître votre base de données sélectionnée actuelle
db. dropDatabase ()	déposez la base de données actuellement sélectionnée
db.createcollection("clients")	créer une nouvelle collection " clients "
show collections	imprimer une liste de toutes les collections créées
db. clients.drop ()	déposez la collection " clients "
db.clients.insert ({nom: 'Ahmed', adresse:'Touggourt'})	insérer un document dans la collection " clients "

db. clients.find ()	lister les documents de la collection " clients "
db.clients.update({'nom' : 'Ahmed'},{\$set: {'nom' : 'mohammed'}})	mettre à jour le document dans la collection " clients "
db. clients. save({" id": new ObjectId("15122aas2656as464s644"), nome: "Ahmed", adresse: "Touggourt"});	enregistrer le document dans la collection " clients "
db. clients.remove({'name': 'Ahmed'})	supprimer le document dans la collection " clients "
db. getUsers () ;	répertorier tous les utilisateurs de la base de données actuelle
show roles	liste de tous les rôles

Tableau2 Commandes de base de MongoDB

#### 4. Différence entre SGBDR et MongoDB :

Le tableau suivant montre les différences les plus importantes entre les systèmes de gestion de base de données traditionnelles et la base de données MongoDB.

SGBDR	MongoDB
Base de données relationnelle.	Base de données orientée document et non relationnelle.
Basé sur la ligne	Basé sur le document
Basé sur la colonne	Basé sur le champ
Basé sur une table	Basé sur une collection et une paire clé-valeur
A un schéma prédéfini et n'est pas bon pour le stockage de données hiérarchique	A un schéma dynamique et idéal pour le stockage de données hiérarchique
Le langage de requête est un langage de requête structuré	Le langage de requête est javascript
Il prend en charge JOIN	Il ne prend pas en charge JOIN

Tableau 3 Différence entre SGBDR et MongoDB



## 5. ENVIRONNEMENT DE TRAVAIL

Dans cette section, nous allons définir notre environnement de travail, à la fois logiciel et matériel, pour créer une base de données NoSQL (MongoDB) pour une entreprise active dans le domaine du e-commerce.

### 5.1 Ressources matérielles :

<b>Microprocesseur</b>	Intel® Core™ i3-6006u 2.00GHz
<b>RAM</b>	4.00 Go
<b>Disque dur</b>	500 Go
<b>Carte graphique</b>	Intel(R) HD Graphics520
<b>Système d'exploitation</b>	Windows 10 Pro 64x

Tableau 4. Ressources matérielles

#### 5.1.1 Configuration requise:

Chaque hôte doit répondre aux exigences suivantes:

- ✓ Au moins 10 Go d'espace disque libre plus tout l'espace nécessaire pour stocker vos données MongoDB.
- ✓ Au moins 4 Go de RAM.
- ✓ Si vous utilisez des instances AWS EC2, vous devez utiliser au minimum une instance m3.medium.
- ✓ MongoDB doit être installé uniquement sur les architectures 64 bits.

### 5.2 Ressources logicielles:

le programme	Version
<b>MongoDB Community Server</b>	5.0.6
<b>MongoDB Compass</b>	1.30.1
<b>MongoDB Shell</b>	1.3.1

Tableau 5 Ressources logicielles

**MongoDB Community Server** : Notre base de données NoSQL (distribuée).

**MongoDB Compass**: Un outil interactif pour interroger, optimiser et analyser graphiquement les données MongoDB.

**MongoDB Shell**: Se connecter à MongoDB pour travailler avec vos données et configurer votre base de données vous permet d'avoir une interface de commande.

### 5.3 Installations et configurations

Nous commençons par le téléchargement du le serveur MongoDB à l'adresse :

<https://www.mongodb.com/try/download/community>

- Télécharger le MSI (Microsoft System Installer). Les instructions d'installation sont ici:
- <https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/>
- Il suffit de double cliquer sur l'exécutable et le serveur s'installe.
- Cette installation inclue MongoDB Compass qui est une interface graphique par laquelle nous pouvons exploiter notre serveur.
- Nous pouvons également naviguer dans l'interface de commande MongoDB Shell.

## 6. Conception et implémentation:

Le commerce électronique capte une part de marché croissante des acheteurs, et en plus des exigences d'exploitation d'un magasin physique telles que le suivi des actifs, les vitrines de commerce électronique sont confrontées à des défis supplémentaires tels que la livraison du produit au client, l'utilisation d'un processeur de paiement et la construction relations clients.

Toutes ces tâches nécessitent une infrastructure spécialisée, qui à son tour interagit avec la base de données. Une grande partie de l'efficacité et de l'efficacité de la vitrine dépend des décisions prise concernant la structure et le stockage des données. Dans cette partie, nous allons explorer les avantages et les inconvénients d'une base de données NoSQL pour l'application de commerce électronique et utiliser le magasin de documents le plus populaire MongoDB dans notre exemple de base de données d'entreprise appelée "4ALL".

- **Présentation de l'entreprise**

4all est une entreprise fictive active dans le domaine du commerce électronique qui vend divers produits aux particuliers, notamment électroniques, avec la fonction de livraison à l'adresse, elle s'appuie fortement sur son site Web. On peut parcourir les produits et également s'inscrire sur le site pour commander, acheter et payer. Et il y a

un responsable sur le site qui a le pouvoir de gérer les demandes et les produits des clients.

Au début de l'entreprise et pendant un certain temps, ses données fonctionnaient selon le modèle relationnel et la base de données relationnelle selon diagramme de classe suivant :

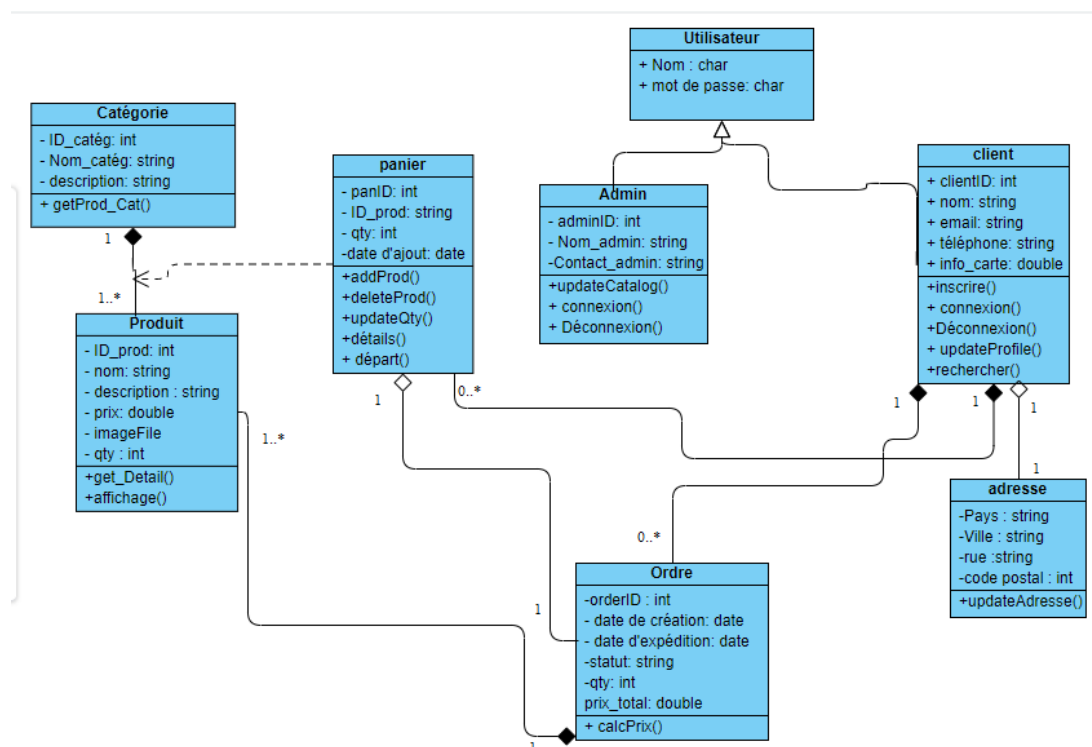


Figure 14 Diagramme de classe pour la boutique en ligne de l'entreprise "4ALL"

Maintenant que le modèle relationnel a atteint ses limites, c'est à notre tour d'essayer de concevoir une solution non relationnelle dans MongoDB, où nous allons diviser la base de données en quatre axes ou processus principaux :

- Gestion des utilisateurs
- Gestion des produits
- Gestion des commandes et des paiements

### 6.1 Utilisateurs :

Nous devons réfléchir à la façon de concevoir un formulaire pour les utilisateurs. Où cela peut être un client ou un administrateur, et c'est ce que nous avons adopté.

Nous souhaitons également stocker des informations qui nous permettent d'établir une relation continue avec les clients, comme la possibilité d'afficher des éléments qui les concernent. Le client doit disposer d'une pièce d'identité spéciale en plus de ses informations de base et ajouter des informations sur la carte bancaire en anticipation du paiement. Si les produits sont physiques, chaque client doit avoir des coordonnées, une adresse, et même l'administrateur doit avoir des informations de contact.

Nous donnons ci-dessous des profils d'utilisateurs simples pour les clients qui achèteront dans le magasin :

```
{
  "_id": {
    "$oid": "624ca515ba2cb2e381701b46"
  },
  "nom": "bouchafa",
  "prenom": "kamel",
  "password": "oran2022",
  "telephone": 656531355,
  "email": "adm120@gmail.com",
  "adresse": {
    "pays": "ALG",
    "rue": "",
    "ville": "oran",
    "code postal": 31012
  },
  "cart_info": {
    "type": "visaCard",
    "nom": "bouchafa kamel",
    "matricule": "2250-1662-1478-1179"
  },
  "nom_util": "bouchafa2022",
  "type_util": "client"
}
```

**client 1**

```
{
  "_id": {
    "$oid": "624ca515ba2cb2e381701b44"
  },
  "nom": "gana",
  "prenom": "abdelmonaim",
  "password": "123123",
  "email": "abdelmonaimgana@gmail.com",
  "adresse": {
    "pays": "ALG",
    "ville": "blidat amor",
    "code postal": 30005
  },
  "cart_info": {
    "type": "VisaCard",
    "nom": "monaimgana",
    "matricule": "4050-1462-1478-1479"
  },
  "nom_util": "mmgana",
  "type_util": "client"
}
```

**client 2**

```
{
  "_id": {
    "$oid": "624ca515ba2cb2e381701b45"
  },
  "nom": "bouhnik",
  "prenom": "ahmed",
  "password": "2014ff",
  "type_util": "client",
  "telephone": 661516489,
  "email": "ramibibo36@gmail.com",
  "adresse": {
    "pays": "ALG",
    "rue": "aid ben sehraoui",
    "ville": "ouargla",
    "code postal": 30013
  },
  "cart_info": {
    "type": "MasterCard",
    "nom": "ahmedbouhenika",
    "matricule": "1545-4646-6454-4467"
  },
  "nom_util": "ahmedb"
}
```

**client 3**

*Figure 15 Exemple de document pour les clients*

Voici également un profil utilisateur simple pour l'administrateur du magasin :

```

{
  "_id": {
    "$oid": "629490bfcf5792fdaa55ea33"
  },
  "contact": {
    "email": "imad2022@hotmail.com",
    "telephone": "789451269"
  },
  "nom": "imad",
  "nom_util": "imad2022",
  "password": "212466",
  "prenom": "chacha",
  "type_util": "Admin"
}

```

Figure 16 Exemple de document pour un administrateur

## 6.2 Liste de produits :

Afin de vendre des produits, nous devons enregistrer ce que nous avons afin que les clients puissent parcourir avant de s'engager dans un achat. L'essentiel pour un document produit est le prix et l'identifiant unique. Il est également courant d'inclure une description, une quantité et une image...En plus de la catégorie où nous le considérons comme un objet, la figure suivante montre un modèle pour les produits :

```

{
  "_id": {
    "$oid": "629491e5cf5792fdaa55ea36"
  },
  "nom": "samsung galaxy note 8",
  "description": "Le téléphone intelligent de Samsung 4G est disponible en plusieurs couleurs.",
  "detail": {
    "poids": "190g",
    "color": "noir,blanch,blue",
    "RAM": "6",
    "Processeur": "helio t120",
    "Rom": "128"
  },
  "qty": 40,
  "Image": "images/phone.jpg",
  "prix": "65000 dz",
  "category": {
    "nom": "Téléphones et communications",
    "description": "Téléphones portables, téléphones fixes, téléphones intelligents, accessoires de téléphonie."
  }
}

```

**produit 1**

```

{
  "_id": {
    "$oid": "649491e5cf5792fdaa55ea36"
  },
  "nom": "HP note book G6",
  "description": "L'ordinateur portable de HP est disponible en plusieurs couleurs et propose d'excellentes fonctionnalités pour les joueurs et les jeux électroniques.",
  "detail": {
    "poids": "2.5 kg",
    "color": "noir,blanch, gris",
    "RAM": "16",
    "Processeur": "intel i7 8eme gene",
    "Rom": "512 ssd"
  },
  "qty": 15,
  "Image": "images/laptop.jpg",
  "prix": "95000 dz",
  "category": {
    "nom": "informatique et sécurité",
    "description": "Ordinateur portable, ordinateur de bureau, ordinateur personnel, accessoires et accessoires informatiques."
  }
}

```

**produit 2**

Figure 17 Exemple de document produit.

### 6.3 Commandes et des paiements :

Nous traitons d'abord l'étape de paiement car nous en avons besoin avant la confirmation et l'enregistrement de la commande, et la modélisation du paiement est l'endroit le plus important pour être sûr. C'est également le premier endroit où nous devons référencer une autre table, où nous avons besoin des informations sur le client.

Ici, un exemple de modèle de paiement :

```
_id: ObjectId('6294c4aecf5792fdaa55ea46')
client_id: "624ca515ba2cb2e381701b44"
status: "payé"
type: "Carte de crédit"
  ✓ cart_info: Object
    type: "VisaCard"
    matrucile: "4050-1462-1478-1479"
```

*Figure 18. Exemple de document pour un paiement*

Les commandes sont un autre modèle de données pour suivre le comportement des clients. Les clients peuvent supprimer et modifier des commandes et demander des remboursements.

Cette étape de l'exemple rassemble toutes les parties mobiles de notre modèle : les produits que les clients achètent avec des paiements. Notre modèle inclura des éléments trouvés dans d'autres groupes, tels que les adresses de livraison et le panier.

Dans MongoDB, nous souhaitons renvoyer toutes les données pertinentes en une seule commande de lecture. Il est souvent préférable de faire une copie des informations pertinentes et de les stocker dans le jeu de commandes. Ainsi, lorsqu'un utilisateur vérifie une requête, on n'a pas besoin de lire plusieurs combinaisons pour afficher les informations souhaitées.

Ici, le résultat final de la modélisation d'une commande de produits en tant que panier est une série d'objets :

```
_id: ObjectId('6294cbd8cf5792fdaa55ea54')
client_id: "624ca515ba2cb2e381701b44"
paiement_id: "6294c4aecf5792fdaa55ea46"
paiement_status: "complété"
status: "En cours"
creation_date: 2022-07-26T11:30:00.000+00:00
▼ panier: Array
  ▼ 0: Object
    produit_id: "629491e5cf5792fdaa55ea36"
    qty: 1
    prix: "65000 DZA"
  ▼ 1: Object
    produit_id: "649491e5cf5792fdaa55ea36"
    qty: "1"
    prix: "95000 DZA"
▼ adresse: Object
  pays: "ALG"
  ville: "blidat amor"
  code postal: 30005
  prix_total: "160000 DZA"
```

Figure 19 Exemple de document d'un processus de commande

### 7. DISCUSSION:

Après avoir jeté un œil aux fonctionnalités de MongoDB, nous allons résumer dans cette section les principaux résultats que nous avons obtenus :

- MongoDB stocke la plupart des données dans la RAM. Il permet une performance plus rapide lors de l'exécution des requêtes.
- Ses attributs permettent une augmentation de la disponibilité des données. Il est également facile d'accéder aux documents grâce à l'indexation.
- MongoDB propose une syntaxe de requête simple et beaucoup plus facile à saisir. Il fournit un langage de requête expressif que les utilisateurs trouvent utile pendant le développement.
- L'installation, la configuration et l'exécution de MongoDB sont simples et rapides. Il propose des frameworks JavaScript modernes.
- Le schéma de MongoDB n'est pas prédéfini. Cela signifie qu'il a une architecture schématique dynamique qui fonctionne avec des données et un stockage non structurés.

- MongoDB utilise le sharding tout en gérant de grands ensembles de données. Le sharding est le processus de division des données d'un grand ensemble et de leur distribution sur plusieurs serveurs.
- Contrairement aux bases de données SQL qui utilisent l'évolutivité verticale, le sharding permet à MongoDB d'utiliser l'évolutivité horizontale.
- Une requête ad hoc est une requête non standard. Il est généré pour obtenir des informations si et quand cela est nécessaire.

Quelques défauts que je nous avons constatés :

- Joindre des documents dans MongoDB peut être une tâche très fastidieuse et quelque peu compliquée, et certainement pas normalisée.
- Si l'indexation est effectuée de manière incorrecte ou s'il y a des lacunes, MongoDB fonctionnera à une vitesse très lente et la correction des erreurs dans les index prendra du temps.
- MongoDB autorise une taille de document limitée à seulement 16 Mo.
- MongoDB nécessite une grande quantité de stockage en raison du manque de fonctionnalité de jointure, ce qui entraîne une redondance des données.
- Certains de ces problèmes rendent la tâche moins productive et peuvent gêner certains utilisateurs.

## 8. Conclusion

Dans ce chapitre, nous avons présenté MongoDB et les principales raisons de son choix, et MongoDB est une base de données NoSQL exceptionnellement compétitive/complémentaire qui peut être attractive pour la plupart des besoins e-commerce, puis nous avons appris à connaître ses fondements et ses différences avec SQL, puis nous avons déménagé du côté applicatif qui décrit l'environnement de travail et comment le configurer puis la modélisation et le développement et enfin nous avons mis en évidence Sur les points positifs et négatifs que nous avons rencontrés.



---

## **Conclusion générale**

---

### **Conclusion générale**

Dans ce travail, nous proposons d'implémenter une base de données NoSQL de type de document "MongoDB" dans notre exemple d'une boutique en ligne pour une entreprise de e-commerce en profitant de ses avantages. Le résultat du travail met en lumière les caractéristiques des bases de données non relationnelles (MongoDB) dans le traitement de grandes données non structurées et semi-structurées, combien nous avons bénéficié dans notre étude de la structure différente du modèle NoSQL, en plus de la flexibilité et évolutivité qui a fourni des performances rapides. Mais comme nous avons trouvé des avantages, il y avait des inconvénients, tels que la difficulté de joindre des documents et la grande taille, et la répétition qui entraîne la difficulté de recherche et de récupération.

En conclusion, ce travail nous a montré l'efficacité des bases de données NoSQL "MongoDB" pour le traitement de données volumineuses et non structurées pour notre cas, mais les bases de données relationnelles peuvent être plus efficaces dans le traitement de données structurées, et dans ces circonstances il peut être possible de trouver un compromis, par exemple, la coexistence de la création de bases de données hybrides qui les rassemblent ou les utilisent séparément selon les besoins. La meilleure solution peut être de développer une base de données qui fournit toutes leurs fonctionnalités, et c'est ce à quoi aspirent les développeurs.

---

## **Références Bibliographique**

---

### Références Bibliographique

- [1] Kristi L. Berg, Tom Seymour, Richa Goel. History Of Databases. International Journal of Management & Information Systems. Décembre 2012.
- [2] Olivier Losson. Introduction aux Systèmes de Gestion de Bases de Données Relationnelles. Cours Master Sciences et Technologies. Université Lille1.2016, 2017.
- [3] Robin Bloor. L'échec des Bases de Données Relationnelles, L'essor de la Technologie Objet et le Recours aux Bases de Données Hybrides. Baroudi Bloor International Inc., 2003, 2004.
- [4] Base de données - Définition et Explications <https://www.techno-science.net/glossaire-definition/Base-de-donnees-page-3.html>(Consulté le 09/02/2022).
- [5] Philippe Rigaux. Cours en ligne de Bases de données documentaires et distribuées. Version Septembre 2021. Disponible sur : <http://b3d.bdpedia.fr/>
- [6] Qu'est-ce qu'un modèle de base de données ? <https://www.lucidchart.com/pages/fr/quest-ce-quun-modele-de-base-de-donnees>(Consulté le 09/02/2022).
- [7] Carlos Coronel, Steven Morris, and Peter Rob. Database Systems: Design, Implementation, and Management, 9eme Edition. 2011 Cengage Learning.
- [8] BEKADDOUR Abderazak, BECHLAGHEM Seyf-Allah. Etude comparative des performances des bases de données SQL et NoSQL MySQL vs MongoDB. Mémoire master informatique. Université Abou BakrBelkaid– Tlemcen, 2017.
- [9] Matteo Di Maglie. Adoption d'une solution NoSQL dans l'entreprise. PhD thesis, Haute école de gestion de Genève, 2012.
- [10] Base de données relationnelle <https://www.lemagit.fr/definition/Base-de-donnees-relationnelle>(Consulté le 14/02/2022).

- [11] SQL (Structured Query Language) : définition, traduction et acteurs.  
<https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203603-sql-structured-query-language-definition-traduction-et-acteurs/>(Consulté le 15/02/2022).
- [12] Adriano Girolamo PIAZZA, NOSQL, Travail de Bachelor réalisé en vue de l'obtention du Bachelor HES, Haute École de Gestion de Genève, 9 octobre 2013.
- [13] Pierre Senellart. Limites des systèmes classiques de gestion de bases de données. Télécom ParisTech. le Oct 18, 2021
- [14] Base de données relationnelle  
<https://www.ionos.fr/digitalguide/hebergement/aspects-techniques/base-de-donnees-relationnelle/>(Consulté le 15/02/2022).
- [15] Limites des SGBDR pour le Big Data, NoSQL et NewSQL.  
<https://www.createursdemondes.fr/programmation/limites-des-sgbdr-pour-le-big-data-nosql-et-newsql/>(Consulté le 15/02/2022).
- [16] Google Cloud. <https://cloud.google.com/what-is-big-data> (Consulté le 17/02/2022).
- [17] MongoDB : présentation et comparaison avec MySQL.  
<https://www.ionos.fr/digitalguide/sites-internet/developpement-web/mongodb-presentation-et-comparaison-avec-mysql/>(Consulté le 20/03/2022).
- [18] What is MongoDB? Introduction, Architecture, Features & Example.  
<https://www.guru99.com/what-is-mongodb.html#6>(Consulté le 22/03/2022).
- [19] Mckinsey&Company: Big Data: The next frontier for innovation, competition and productivity. May 2011
- [20] Cassandra vs MongoDB: A Battle of the NoSQL DBs.  
<https://blog.panoply.io/cassandra-vs-mongodb>(Consulté le 23/03/2022).
- [21] MongoDB Features. <https://www.mongodb.com/what-is-mongodb/features>  
(Consulté le 25/03/2022).
- [22] Customer success stories. <https://www.mongodb.com/who-uses-mongodb>(Consulté le 25/03/2022).

- [23] أو البيانات الضخمة؟ "Big Data" ماذا يعني مصطلح [23]  
<https://www.almayadeen.net/news/misc/1489746/%D9%85%D8%A7%D8%B0%D8%A7-%D9%8A%D8%B9%D9%86%D9%8A-%D9%85%D8%B5%D8%B7%D9%84%D8%AD-big-data-%D8%A3%D9%88-%D8%A7%D9%84%D8%A8%D9%8A%D8%A7%D9%86%D8%A7%D8%AA-%D8%A7%D9%84%D8%B6%D8%AE%D9%85%D8%A9>(Consulté le 15/03/2022).
- [24] MongoDB. <https://www.mongodb.com/big-data-explained>(Consulté le 30/03/2022).
- [25] Madjid BOUZOURENE .Mise en place d'un modèle de cohérence orienté Data-centric dans un environnement cloud. Mémoire master informatique. Université Mouloud Mammeri de Tizi-Ouzou, 2020.
- [26] KOUEDI Emmanuel, Approche de migration d'une base de données relationnelle vers une base de données NoSQL orientée colonne, Mémoire master informatique, UNIVERSITE DE YAOUNDE I, mai 2012.
- [27] SGBDR et répartition de charge : « scalabilité.  
[https://blog.developpez.com/sqlpro/p10383/langage-sqlnorme/sgbdr\\_et\\_repartition\\_de\\_charge\\_scalabilite](https://blog.developpez.com/sqlpro/p10383/langage-sqlnorme/sgbdr_et_repartition_de_charge_scalabilite) (Consulté le 16/03/2022).
- [28] Bernard ESPINASSE, Introduction aux systèmes NoSQL (Not Only SQL), Ecole Polytechnique Universitaire de Marseille, 19p, Avril 2013
- [29] Aurélien Foucret, NosqlSmile open source ,2012.
- [30] Thierry Hamon. Entrepôts de données NoSQL. Université Paris 13. 2020/2021