RÉPUBLIQUE DU SÉNÉGAL

Un peuple - un but - une foi

¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR, DE LA RECHERCHE ET DE L'INNOVATON

¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤

L'excellence ma référence

¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤

Mention : Management Informatisé des organisations (MI0)

Département : Économie Gestion

UFR : Science Économique Sociale

¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤

SUJET

Stockage des Big Data : Etude approfondie sur les bases de données NoSQL

¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤

Présenté

FAMA TOURE

Sous la direction

Dr Edouard Ngor SARR

Enseignant-Chercheur en Informatique à l'UASZ

Avec l'accompagnement Technique

SERIGNE MOR TOURE

Année Académique : 2023-2024

Remerciements

Tout d'abord, je rends grâce au bon Dieu.

Je tiens à exprimer ma reconnaissance à mon Directeur de Projet, Docteur Edouard Ngor SARR, pour m'avoir confié ce projet et pour son encadrement, sa patience, ses conseils éclairés et sa disponibilité tout au long de cette étude. Son expertise et ses suggestions ont été d'une grande aide dans l'élaboration de ce travail.

Je remercie également l'ensemble du corps professoral de la filière Management Informatisé des Organisations (MIO) de l'Université Assane SECK de Ziguinchor pour la qualité de leur enseignement.

Ainsi mes remerciements vont à mon aîné Monsieur Sérigne Mor TOURE pour son soutien, son expertise, et sa participation à la réussite de ce travail.

Enfin, un immense merci à ma famille et à mes amis pour leur soutien moral et leur encouragement constant durant cette période et à vous tous, qui de près ou de loin, avez contribué à la réussite de ce travail.

Vous avez toute ma gratitude.

Table des matières

Liste des Figures

Liste des Tableaux

Sigles et Abréviations

- **ACID :** **A**tomicity, **C**onsistency, **I**solation, **D**urability **(A**tomicité, **C**ohérence, **I**solation, **D**urabilité)
- **AP :** Disponibility + Partition Tolerance (Disponibilité et Tolérance au Partitionnement)
- **API :** Application Programming Interface (Interface de Programmation d'Applications)
- **BSON :** Binary JSON (JSON Binaire)
- **CA :** Consistency + Disponibility (Cohérence et Disponibilité)
- **CAP :** **C**onsistency, **A**vailability and **P**artition tolerance (**C**ohérence, **D**isponibilité et **R**ésistance au partitionnement)
- **CP :** Consistency + Partition Tolerance (Cohérence et Tolérance au Partitionnement)
- **HDFS :** Hadoop Distributed File System (Système de Fichiers Distribué Hadoop)
- **HIPAA :** Health Insurance Portability and Accountability Act
- **IoT :** Internet of Things (Internet des Objets)
- **JSON :** JavaScript Object Notation (Notation Objet JavaScript)
- **JWT :** JSON Web Token (Jeton Web JSON)
- **LDAP :** Lightweight Directory Access Protocol (Protocole Léger d'Accès aux Annuaires)
- **NoSQL :** Not Only SQL (Pas Seulement SQL)
- **OAuth :** Open Authorization (Autorisation Ouverte)
- **OLAP :** Online Analytical Processing **(**Traitement Analytique en Ligne)
- **RBAC :** Role-Based Access Control (Contrôle d'Accès Basé sur les Rôles)
- **RGPD :** Règlement Générale sur la Protection des Données
- **SGBDR :** Système de Gestion de Bases de Données Relationnelles
- **SQL :** Structured Query Language (Langage de Requêtes Structuré)
- **XML :** Extensible Markup Language (Langage de Balises Extensible)

Introduction générale

Dans un monde de plus en plus dominé par la transformation numérique, les données sont devenues une ressource précieuse, souvent qualifiée de "nouveau pétrole". L'avènement des Big Data, caractérisé par un volume, une vitesse et une variété sans précédent, a bouleversé les paradigmes traditionnels de stockage et de gestion des données. Face à ces défis, les bases de données relationnelles classiques (SGBDR), bien que robustes et éprouvées, montrent leurs limites dans des environnements nécessitant une flexibilité et une scalabilité accrues.

C'est dans ce contexte que les bases de données NoSQL se sont imposées comme une solution incontournable. Conçues pour gérer efficacement des données massives et hétérogènes, elles offrent des architectures non conventionnelles, capables de répondre aux exigences des applications modernes, telles que le traitement en temps réel, l'analyse prédictive et la personnalisation des services.

Ce projet vise à explorer en profondeur les caractéristiques, avantages et inconvénients des bases NoSQL par rapport aux SGBDR. À travers une analyse structurée, il s'agit de comprendre pourquoi et comment les bases NoSQL complètent, voire remplacent, les bases relationnelles dans de nombreux cas d'usage.

Objectifs du Projet :

- Fournir une compréhension globale des Big Data, en expliquant leurs dimensions (les 5V), leurs enjeux et leurs domaines d'utilisation ;
- Présenter les bases NoSQL en détaillant leurs types, leur fonctionnement, ainsi que leurs avantages et limites ;
- Comparer les bases NoSQL et les bases relationnelles (SGBDR) à travers des critères tels que la cohérence, la scalabilité, la gestion des transactions et la sécurité ;
- Mettre en perspective les évolutions futures dans le domaine du stockage des données, tout en soulignant les tendances émergentes.

En plaçant le focus sur les bases NoSQL, ce projet se veut une contribution précieuse à la compréhension des transformations technologiques actuelles, tout en fournissant des recommandations pour relever les défis liés à la gestion des Big Data. Ce travail est également une opportunité permettant de mieux appréhender les choix technologiques stratégiques pour les organisations souhaitant exploiter le plein potentiel des données.

Chapitre 1 : Généralité sur les Big Data

1. Définitions

Littéralement, ces deux termes **'Big'** et **'Data'** signifient mégadonnées, grosses données ou encore données massives. Ils désignent un ensemble très volumineux de données qu'aucun outil classique de gestion de données de l'information ne peut vraiment travailler. Ce sont les informations provenant de partout : messages que nous nous envoyons sur les médias sociaux, vidéos que nous nous publions, capteurs utilisés pour collecter les informations climatiques, signaux GPS, enregistrements transactionnels d'achats en ligne et bien d'autres encore [1]. Ils s'agissent d'un ensemble de technologies et d'outils permettant à une organisation de collecter, stocker et analyser rapidement de larges quantités de données hétérogènes afin d'en extraire des informations pertinentes et permettre à tout le monde d'accéder en temps réel à des bases de données géantes [2].

- **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 [4] ;
- Facebook gère plus de 50 milliards de photos de ses utilisateurs [4] ;
- Google gère environ 100 milliards de recherches par mois [4].

2. Les 5V

Les **5V** du Big Data sont des caractéristiques qui définissent les données massives. Ces cinq dimensions sont essentielles pour comprendre la complexité et les défis associés à la gestion des données dans des volumes énormes. Voici un résumé des **5V** et un exemple pour chacun :

a. Volume (Volume) :

Si le volume n'est en aucun cas le seul élément qui rend le Big Data « Big », il fait certainement partie des principaux. Pour gérer et exploiter pleinement le Big Data, des algorithmes avancés et l'analytique pilotée par l'intelligence artificielle sont nécessaires. Mais avant tout cela, il faut un moyen sûr et fiable de stocker, d'organiser et d'extraire les nombreux téraoctets de données à la disposition des grandes entreprises.

- **L'exemple de Facebook :** Facebook compte plus d'utilisateurs que la Chine n'a d'habitants. Chacun de ces utilisateurs y stocke de nombreuses photos. Facebook stocke ainsi environ 250 milliards d'images.

b. La Véracité (Veracity) :

La véracité désigne la qualité des données. Les données peuvent être inexactes, incomplètes ou bruitées, et leur gestion devient complexe lorsqu'il faut garantir leur fiabilité et leur précision. La véracité est cruciale pour les décisions basées sur les données.

- **Exemple** : Les données de santé collectées par des dispositifs médicaux peuvent contenir des erreurs, des défauts de mesure ou des incohérences qui doivent être corrigés avant d'être analysés.

c. Valeur (Value) :

Sans aucun doute, les résultats de l'analyse du Big Data sont souvent fascinants et inattendus. Mais pour les entreprises, l'analytique du Big Data doit fournir des insights capables d'aider les entreprises à gagner en compétitivité et en résilience, et à mieux servir leurs clients. Les technologies modernes du Big Data offrent la possibilité de collecter et d'extraire des données susceptibles de procurer un avantage mesurable à la fois en matière de résultats nets et de résilience opérationnelle.

- **Exemple** : Les données des clients collectées par une entreprise, telles que les historiques d'achat, les comportements en ligne, etc., qui peuvent être utilisées pour personnaliser les offres et augmenter les ventes, créant ainsi une valeur commerciale.

d. La Vélocité (Velocity) :

Dernière dimension, tout aussi importante que les précédentes, la vélocité traduit la capacité à produire rapidement les données et à les transformer en temps utile pour leurs utilisateurs. L'exercice, déjà difficile dans un contexte "classique", prend toute sa valeur lorsqu'il doit être appliqué à d'immenses volumes de données de toutes sortes.

- **Exemple :** Google traite en moyenne plus de "40 000 requêtes de recherche par seconde", ce qui représente environ 3,5 milliards de recherches par jour.

e. La Variété (Variety) :

En ce qui concerne le Big Data, nous devons non seulement gérer des données structurées, mais également des données semi-structurées et surtout non structurées. Aujourd'hui, les informations utiles proviennent aussi d'autres sources : documents, courrier électronique, réseaux sociaux... et prennent de nouvelles formes : texte, image, vidéo... L'analyse des "big data" concerne aussi ces données, pour lesquelles les moyens disponibles sont encore en émergence.

- **Exemple :** Les messages électroniques : Actuellement on estime à environ un million de messages électroniques émis par jour. Aucun de ces messages ne sera exactement comme un autre. Chacun se composera de l'adresse électronique de l'expéditeur, d'un destinataire et d'un horodatage.

Chaque message comportera un texte et éventuellement des pièces jointes qui ne sont généralement pas structurés et très variés à savoir **:** Les photos, vidéos, enregistrements audio, messages électroniques, documents, livres, présentations et bien d'autre encore.

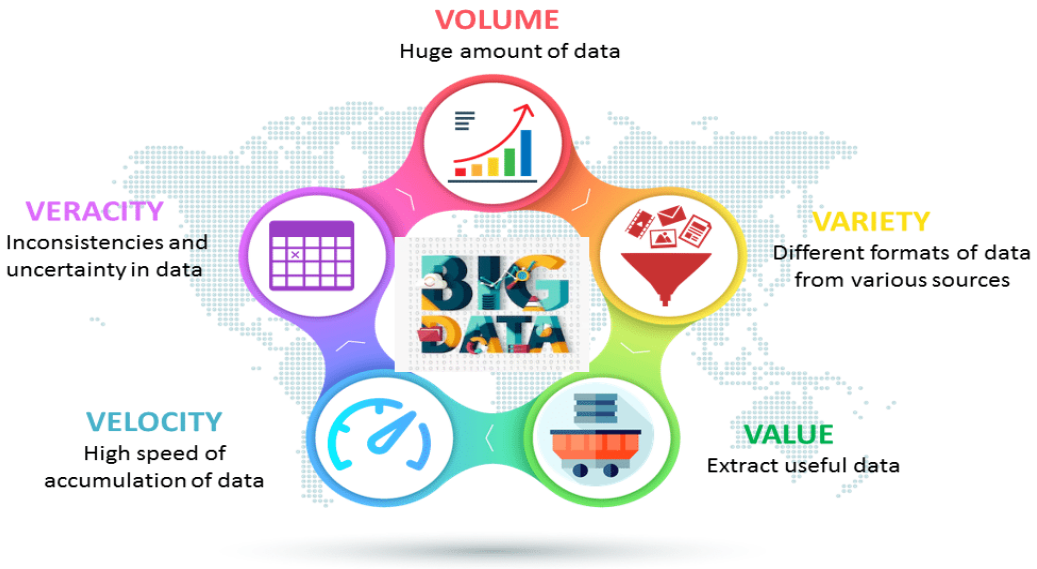

Figure 1: Schéma des 5V du Big Data

Ces **5V** sont interconnectés et les entreprises doivent réussir à gérer ces défis pour exploiter efficacement les données massives et en tirer des avantages concrets.

3. Enjeux

Les enjeux liés aux Big Data sont nombreux et touchent plusieurs domaines, que ce soit technique, économique, éthique ou social. Voici les principaux enjeux :

a. Enjeu de la gestion du volume et de la complexité des données

Les Big Data se caractérisent par des volumes de données énormes générés à une vitesse rapide et provenant de sources diverses. La collecte, le stockage, le traitement et l'analyse de ces données massives posent des défis techniques et nécessitent des infrastructures avancées (comme les solutions de stockage distribuées et les systèmes de traitement parallèles).

b. Enjeu de la qualité des données

La diversité des données (structurées, semi-structurées et non structurées) exige un travail de nettoyage et de transformation pour assurer leur qualité. La présence de données incomplètes, redondantes ou incorrectes peut entraîner des analyses erronées et des décisions inadaptées. Assurer une bonne qualité des données est donc essentiel pour en tirer de la valeur [3].

c. Enjeu de la sécurité et de la confidentialité

Les Big Data contiennent souvent des informations sensibles, notamment des données personnelles. Leur gestion soulève des questions de sécurité pour éviter les fuites et les cyberattaques, ainsi que de confidentialité pour protéger la vie privée des individus. Cela devient un défi pour les

entreprises, qui doivent se conformer aux réglementations (comme le RGPD en Europe) tout en exploitant les données à des fins commerciales.

d. Enjeu de l'analyse en temps réel

De nombreuses applications Big Data, comme celles dans les domaines de la finance ou de la santé, nécessitent un traitement et une analyse en temps réel. Répondre à ce besoin exige des infrastructures capables de traiter rapidement des flux de données continus, ce qui peut être coûteux et technologiquement complexe.

e. Enjeu de l'éthique et de la responsabilité

Les Big Data posent aussi des enjeux éthiques, notamment en ce qui concerne la vie privée, la transparence dans l'utilisation des données, et l'impact des analyses automatisées sur les individus (comme les biais dans les algorithmes d'IA). Les entreprises doivent être responsables dans leur manière d'exploiter les données, en veillant à éviter les discriminations et à respecter les droits des utilisateurs.

En résumé, les Big Data représentent une opportunité considérable pour les entreprises et les organisations, mais leur exploitation comporte des défis complexes. Gérer ces enjeux de manière efficace est essentiel pour tirer pleinement parti du potentiel des Big Data tout en respectant les réglementations et en assurant une utilisation responsable et éthique des données.

4. Domaines d'utilisations

Les Big Data sont utilisées dans de nombreux domaines en raison de leur capacité à traiter et analyser d'énormes quantités de données. Voici quelques domaines clés où les Big Data apportent une valeur significative :

a. Marketing et Publicité

Les entreprises utilisent les Big Data pour mieux comprendre les comportements des consommateurs, personnaliser les campagnes marketing et améliorer l'expérience client. Les analyses prédictives permettent de cibler les publicités en fonction des préférences et des habitudes d'achat, augmentant ainsi l'efficacité des campagnes.

b. Santé et Médecine

Les Big Data sont appliquées dans la recherche médicale pour analyser des données de santé à grande échelle, permettant de détecter des tendances, prédire des épidémies et améliorer les diagnostics. Elles facilitent aussi la médecine personnalisée, en adaptant les traitements aux profils spécifiques des patients, et améliorent la gestion des ressources médicales.

c. Finance et Banque

Dans la finance, les Big Data permettent de détecter les fraudes en temps réel, d'analyser les risques financiers et de comprendre les comportements des clients. Les institutions bancaires utilisent

les données pour optimiser la gestion des portefeuilles, prévoir les fluctuations des marchés, et personnaliser les services financiers.

d. Médias et Divertissement

Les plateformes de streaming et les réseaux sociaux utilisent les Big Data pour analyser les préférences des utilisateurs, personnaliser le contenu recommandé, et anticiper les tendances de consommation. Cela permet une meilleure fidélisation des clients en leur proposant des contenus adaptés à leurs goûts.

e. Sécurité et Défense

Dans le domaine de la sécurité, les Big Data aident à détecter des comportements suspects et à prévenir les menaces potentielles. Les données issues de la vidéosurveillance, des réseaux sociaux, et d'autres sources sont analysées pour identifier des anomalies. Les gouvernements et agences de défense utilisent aussi les Big Data pour le renseignement et la protection des infrastructures critiques.

f. Smart Cities (Villes Intelligentes)

Les Big Data sont au cœur des projets de villes intelligentes, où elles permettent de gérer les infrastructures urbaines, de fluidifier le trafic, de réduire la consommation d'énergie et de surveiller la qualité de l'air. Les capteurs et les appareils **IoT** collectent des données en temps réel pour optimiser les services urbains et améliorer la qualité de vie des citoyens.

g. Éducation

Dans l'éducation, les Big Data sont utilisées pour personnaliser l'apprentissage, améliorer les curriculums, et analyser les performances des élèves. Elles aident les établissements à identifier les besoins des étudiants et à adapter les méthodes d'enseignement en fonction des données sur l'apprentissage, améliorant ainsi les résultats académiques.

En résumé, les Big Data sont un outil puissant pour la prise de décision dans de nombreux domaines, permettant une analyse approfondie et la personnalisation des services. Elles contribuent ainsi à améliorer l'efficacité, l'expérience utilisateur, et l'innovation dans divers secteurs.

Chapitre 2 : Généralité sur les NoSQL

1. Définitions

Le terme **NoSQL « Not Only SQL »** est pour la première fois proposé par Carlo Strozzi en **1998**, lors de la présentation de son **SGBDR** open source. Il l'a appelé ainsi à cause de l'absence d'interface **SQL** pour interagir avec les bases de données. Le NoSQL regroupe de nombreuses bases de données, récentes pour la plupart, qui se caractérisent par une logique de représentation de données non relationnelles et qui n'offrent donc pas une interface de requêtes en SQL [3].

Ces bases de données stockent les données dans un format différent. Toutefois, les bases de données NoSQL peuvent être interrogées à l'aide d'**API** en langage idiomatique, de langages déclaratifs et de langages de requête par exemple, ce qui explique pourquoi elles sont également considérées comme des bases de données « pas seulement SQL ».

Les bases de données NoSQL sont largement utilisées dans les applications Web et le big data en temps réel, car elles présentent le principal avantage de proposer une évolutivité élevée et une haute disponibilité. Les bases de données NoSQL sont généralement préférées par les développeurs, car elles se prêtent naturellement à un paradigme de développement agile en s'adaptant rapidement à l'évolution des exigences. Les bases de données NoSQL permettent de stocker les données de manière plus intuitive et plus facile à comprendre, ou plus proche de la façon dont elles sont utilisées par les applications, avec moins de transformations requises lors du stockage ou de l'extraction à l'aide d'API de type NoSQL. De plus, les bases de données NoSQL peuvent tirer pleinement parti du cloud pour éviter tout temps d'inactivité.

2. Règle CAP

La règle **CAP** acronyme de **« C**onsistency, **A**vailability and **P**artition tolerance **»,** qui désigne en français : **« C**ohérence, **D**isponibilité et **R**ésistance au partitionnement **»** a été inventé par Éric Brewer en **2000**. Il s'agit d'une réflexion générale sur la conception de systèmes distribués, ce qui intéresse spécialement le monde du NoSQL pour les besoins de calcul distribué [3].

En **2002**, Seth GILBERT et Nancy LYCH affirment qu'un système d'information à calcul distribué ne peut satisfaire que deux, parmi les trois contraintes suivantes [3] :

- **C**ohérence (**C**onsistency) La cohérence signifie que tous les clients voient les mêmes données en même temps, quel que soit le nœud auquel ils se connectent. Pour cela, chaque fois que des données sont écrites sur un nœud, elles doivent être instantanément transférées ou répliquées vers tous les autres nœuds du système pour que l'écriture soit considérée comme « réussie » [3].

- **D**isponibilité (**A**vailability) La disponibilité signifie qu'un client qui effectue une demande de données obtient une réponse, même si un ou plusieurs nœuds sont en panne. Autre façon de l'exprimer : tous les nœuds actifs du système réparti renvoient une réponse valide à toute demande, sans exception [3]**.**
- **R**ésistance au partitionnement (**P**artition tolerance) Un partitionnement est une rupture de communication au sein d'un système réparti, une perte ou un retard temporaire de connexion entre deux nœuds. La tolérance au partitionnement signifie que la grappe (cluster) doit continuer à fonctionner quel que soit le nombre d'interruptions entre les nœuds du système [3]

Afin de créer une architecture distribuée correcte, on est amené à choisir deux de ces propriétés qu'on vient de citer, laissant ainsi trois designs possibles comme illustrer dans le schéma ci-dessous.

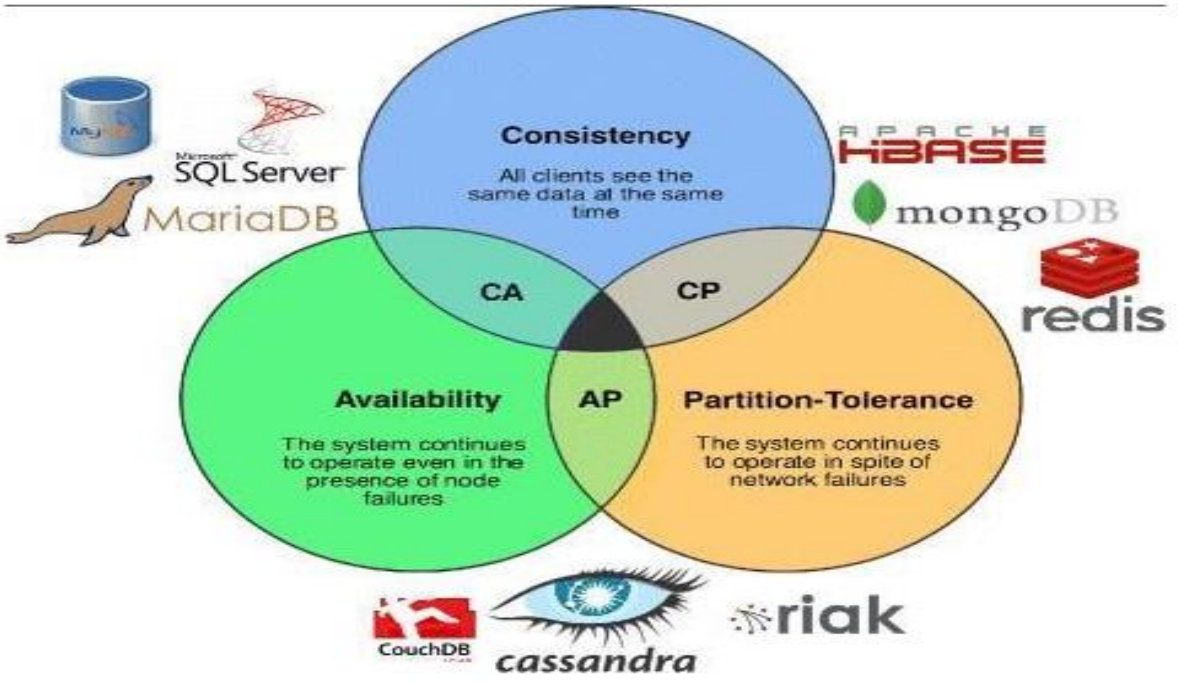

Figure 2: Schéma des 5V du Big Data [3]

3. Types de NoSQL

Les bases de données NoSQL se classent en quatre principaux types, chacun étant conçu pour des usages spécifiques et présentant des caractéristiques adaptées à différents besoins. Voici une présentation des quatre types principaux de NoSQL et des exemples pour chacun.

a. NoSQL orientés Clés-valeurs

i. Définition

Les bases de données clé-valeur stockent les données sous la forme de paires « clé-valeur ». Chaque clé est unique et associée à une valeur, qui peut être de tout type (chaîne, nombre, tableau, etc.). Ce type de base de données est très performant pour les opérations de recherche simple par clé et est particulièrement adapté aux systèmes nécessitant une mise en cache rapide.

ii. Exemples

- **Redis :** Utilisé pour la mise en cache, la gestion de sessions, et les files d'attente en temps réel.
- **Memcached :** Fréquemment utilisé pour réduire la charge sur les bases de données en cachant les données fréquemment demandées.

Cas d'utilisation : Applications nécessitant des accès rapides aux données, comme les réseaux sociaux pour stocker les sessions utilisateur ou les informations temporaires.

Figure 3: Exemple de NoSQL orienté Clé-Valeur [3]

b. NoSQL orientés Documents

i. Définition

Les bases orientées documents stockent les données sous forme de documents, généralement au format **JSON**, **BSON**, ou **XML**. Chaque document est autoportant et contient toutes les informations nécessaires, ce qui facilite la représentation de structures de données complexes. Ce type de base est adapté aux données semi-structurées ou non structurées.

ii. Exemples

- **MongoDB :** Populaire pour les applications Web, il permet de stocker des documents flexibles et scalables horizontalement.

- **Couchbase :** Permet une gestion des documents avec un support pour des opérations en temps réel.

Cas d'utilisation : Applications nécessitant une grande flexibilité dans les types de données, comme les systèmes de gestion de contenu, les catalogues de produits, et les applications analytiques.

Figure 4 : Exemple de NoSQL orienté document [6]

c. NoSQL orientés graphes

i. Définition

Les bases de données orientées graphes sont conçues pour stocker et gérer des relations complexes entre les données. Elles utilisent des structures de nœuds, d'arêtes et de propriétés pour représenter et stocker les données, ce qui les rend idéales pour les données très interconnectées.

ii. Exemples

- **Neo4j :** Très utilisé pour les applications de recommandation, les analyses de réseau, et la détection de fraudes.
- **Amazon Neptune :** Un service cloud de base de données orientée graphes pour les applications sociales, les moteurs de recommandations, et la gestion des relations d'entreprise.

Cas d'utilisation : Applications nécessitant des connexions complexes et des analyses de réseaux, comme les réseaux sociaux, la gestion des relations entre entités, et les systèmes de recommandations.

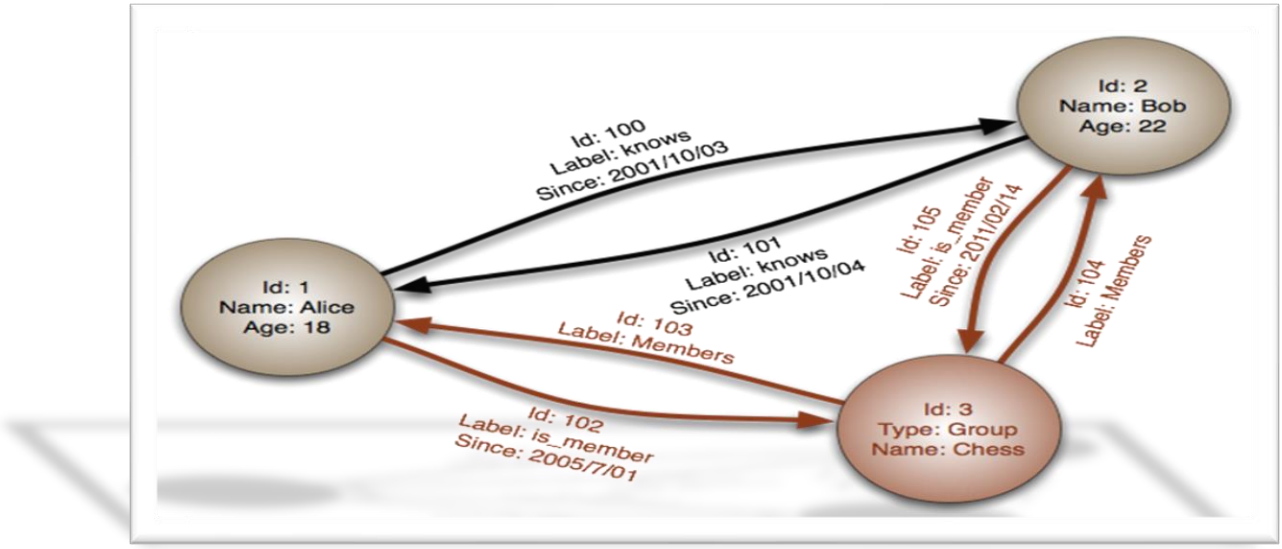

Figure 5 : Exemple de NoSQL orienté Graphe [4]

d. NoSQL orientés Colonnes

i. Définition

Dans les bases de données orientées colonnes, les données sont stockées en colonnes au lieu de lignes, ce qui optimise les opérations de lecture et d'écriture pour les traitements analytiques. Ce modèle est particulièrement efficace pour les analyses de grandes quantités de données.

ii. Exemples

- **Apache Cassandra :** Conçu pour gérer de grandes quantités de données distribuées sur plusieurs serveurs, il est utilisé pour les analyses en temps réel et la gestion de gros volumes de données.
- **HBase :** Basé sur le système de fichiers HDFS de Hadoop, il est utilisé pour les applications nécessitant une haute performance en lecture et écriture.

Cas d'utilisation : Idéal pour les applications analytiques en temps réel, comme les analyses de logs, le suivi d'activité, et les applications de traitement de données massives.

Product ID	Name	Price 1	Price 2	Price 3
1	Liquide vaisselle	**date**: 01/02/2020 **price**: 2.42€	**date**: 12/05/2020 **price**: 2.48€	
2	Shampooing	**date**: 08/05/2020 **price**: 1.56€	**date**: 12/09/2020 **price**: 1.12€	**date**: 19/09/2020 **price**: 1.56€
3	Fromage blanc	**date**: 12/05/2020 **price**: 2.02€		

Figure 6 : Exemple de NoSQL orienté Colonne [3]

4. Avantages et inconvénients

a. Avantages

La rapidité et l'ampleur sans précédent de l'interaction numérique et de la consommation de données constatées au cours des vingt dernières années ont nécessité aux entreprises d'adopter une approche plus moderne et fluide du stockage et de la consommation des données. Alors que les utilisateurs du monde entier exigent un flux ininterrompu de contenus et de fonctions, il n'est pas étonnant que les bases de données aient dû s'adapter rapidement. Dans cette optique, voici quelques-unes des principales raisons pour lesquelles les développeurs choisissent des bases de données NoSQL/non relationnelles :

- **Flexibilité**

Avec les bases de données SQL, les données sont stockées dans une structure bien plus rigide et prédéfinie. Toutefois, dans les bases de données NoSQL, les données peuvent être stockées de façon plus libre sans ces schémas rigides. Cette conception permet d'innover et de développer rapidement des applications. Les développeurs peuvent se concentrer sur la création de systèmes pour mieux servir leurs clients sans se soucier des schémas. Les bases de données NoSQL peuvent facilement gérer tous les formats de données, tels que des données structurées, semi-structurées et non structurées, dans un seul et même répertoire.

- **Évolutivité**

Au lieu d'évoluer en ajoutant d'autres serveurs, les bases de données NoSQL peuvent évoluer en utilisant du matériel de base. Il est donc possible de prendre en charge une augmentation du trafic afin de répondre à la demande sans aucun temps d'arrêt. En évoluant, les bases de données NoSQL peuvent devenir plus volumineuses et plus puissantes, c'est pourquoi elles sont devenues l'option privilégiée pour l'évolution des ensembles de données.

- **De hautes performances**

L'architecture évolutive d'une base de données NoSQL peut être particulièrement utile lorsque le volume de données ou le trafic augmente. Comme le montre le graphique ci-dessous, cette architecture garantit des temps de réponse rapides et prévisibles en quelques millisecondes. Les bases de données NoSQL peuvent également ingérer des données et les livrer rapidement et de manière fiable, c'est pourquoi les bases de données NoSQL sont utilisées dans des applications qui collectent des téraoctets de données chaque jour, tout en nécessitant une expérience utilisateur hautement interactive. Dans le graphique ci-dessous, nous affichons un taux entrant de **300 lectures** par seconde (ligne bleue) avec une latence du **95^{e}** percentile dans la plage des 3 à 4 ms et un taux entrant de **150** écritures par seconde (ligne verte) pour une latence du **95^{e}** percentile dans la plage des **4 à 5 ms**.

- **Disponibilité**

Les bases de données NoSQL répliquent automatiquement les données sur plusieurs serveurs, data centers ou ressources cloud. Dès lors, la latence pour les utilisateurs est réduite, quelle que soit leur localisation. Cette fonctionnalité permet également de réduire le fardeau que représente la gestion de base de données et permet aux équipes informatiques de se concentrer sur d'autres tâches plus importantes.

- **Hautement fonctionnel**

Les bases de données NoSQL sont conçues pour créer des répertoires de données distribués permettant le stockage de données extrêmement volumineuses. Le NoSQL constitue donc le choix idéal pour les big data, les applications Web en temps réel, les relations client, l'e-commerce, les jeux en ligne, l'Internet des objets, les réseaux sociaux et les applications de publicité en ligne.

b. Inconvénients

Les bases de données NoSQL présentent plusieurs inconvénients malgré leurs avantages pour certaines applications. Voici les principaux :

- **Absence de Cohérence Forte**

La majorité des bases de données NoSQL optent pour une cohérence éventuelle (surtout dans les systèmes distribués), ce qui signifie que les données peuvent ne pas être immédiatement synchronisées sur tous les nœuds.

Cela peut poser problème pour les applications nécessitant une cohérence stricte en temps réel, comme les systèmes financiers.

- **Faiblesse des Transactions ACID**

Contrairement aux SGBDR, qui respectent le modèle ACID (Atomicité, Cohérence, Isolation, Durabilité), les bases NoSQL sont souvent limitées dans la gestion des transactions complexes.

Par exemple, il peut être difficile d'assurer des transactions multi-documents ou multi-collections dans des bases orientées documents (comme MongoDB), ce qui peut entraîner des incohérences en cas d'échec partiel de la transaction.

- **Problèmes de Sécurité et de Conformité**

Les bases de données NoSQL n'ont pas toujours le même niveau de sécurité que les SGBDR, en particulier pour les systèmes distribués.

Le support de la sécurité, tel que le contrôle d'accès détaillé ou le chiffrement intégré, est parfois moins robuste. Cela peut rendre ces bases moins adaptées aux environnements réglementés nécessitant une conformité stricte (ex. : GDPR, HIPAA).

- **Manque de Standardisation**

Contrairement aux bases relationnelles qui utilisent toutes le SQL comme langage de requête standard, les bases NoSQL ont chacune leurs propres API et méthodes d'interrogation.

Cela signifie qu'il est plus difficile pour les développeurs de passer d'une technologie NoSQL à une autre, car il n'y a pas de langage ou de modèle universel.

- **Difficulté de Gestion de Relations Complexes**

Les bases de données NoSQL ne sont pas idéales pour gérer des relations complexes entre les données, comme les jointures.

Bien que certaines bases NoSQL (comme les bases orientées graphes) gèrent les relations de manière optimisée, les bases orientées documents et clé-valeur, par exemple, nécessitent souvent de dupliquer des données pour gérer les relations, ce qui peut entraîner des incohérences.

- **Scalabilité Horizontale Complexe**

Bien que les bases NoSQL soient conçues pour la scalabilité horizontale, cela exige des compétences techniques pour gérer correctement les répartitions de données, les partitions et les réplications entre les nœuds.

La gestion de la scalabilité horizontale peut devenir complexe et nécessite des outils et des configurations spécifiques pour éviter les goulets d'étranglement ou les pertes de données.

- **Maintenance et Administration Complexes**

Les systèmes NoSQL nécessitent souvent des configurations et des ajustements spécifiques, surtout pour les bases de données distribuées.

Comparées aux SGBDR, qui sont bien documentées et largement adoptées, les bases NoSQL peuvent nécessiter des compétences plus spécialisées en administration et en optimisation, ce qui augmente les coûts de maintenance.

- **Limites pour les Applications OLAP**

Les bases de données NoSQL ne sont pas idéales pour les applications **OLAP** (Online Analytical Processing) ou les traitements analytiques complexes, en raison du manque de support pour les requêtes complexes, les jointures, et les agrégations.

Par conséquent, elles sont souvent moins performantes pour les analyses approfondies, comparé aux SGBDR qui gèrent mieux les calculs analytiques.

- **Coût de Migration Elevé**

Passer d'une base de données relationnelle à NoSQL peut être coûteux et complexe en raison des différences fondamentales dans les structures de données et les modèles de transactions.

La migration nécessite souvent une refonte des applications et une reconfiguration des données pour s'adapter au modèle non-relationnel.

En résumé, les bases de données NoSQL sont puissantes et flexibles, particulièrement adaptées aux applications nécessitant une grande scalabilité et un traitement rapide de gros volumes de données non structurées. Cependant, leurs inconvénients en termes de cohérence, de sécurité, de standardisation, et de complexité de maintenance les rendent moins adaptées aux applications transactionnelles critiques, analytiques, et réglementées.

Chapitre 3 : NoSQL Vs SGBDR

1. ACID vs CAP

Les **SGBDR** suivent le modèle **ACID** (**A**tomicité, **C**ohérence, **I**solation, **D**urabilité), garantissant la fiabilité des transactions. Cela signifie que les transactions sont :

- **A**tomiques : toutes les opérations réussissent ou échouent ensemble ;
  - ✓ **Exemple :** Lors d'un transfert bancaire, si un compte est débité mais l'autre ne peut être crédité, tout est annulé.
- **C**ohérentes : Les données doivent toujours respecter les contraintes (intégrité référentielle, clés primaires/secondaires). L'intégrité des données est maintenue ;
  - ✓ **Exemple :** Impossible d'insérer un enregistrement avec un identifiant dupliqué.
- **I**solées : les transactions concurrentes n'interfèrent pas entre elles. Les données intermédiaires d'une transaction ne sont pas visibles par d'autres. ;
  - ✓ **Exemple** : Une commande en cours de traitement n'est pas visible tant qu'elle n'est pas validée.
- **D**urables : les modifications sont permanentes une fois validées.
  - ✓ **Exemple :** Une coupure d'électricité ne compromet pas une transaction validée.

Les bases de données **NoSQL**, en revanche, s'alignent souvent sur le théorème **CAP** (**C**ohérence, **D**isponibilité, **T**olérance au Partitionnement), qui stipule qu'un système distribué ne peut garantir simultanément les trois propriétés. Les bases NoSQL choisissent généralement deux parmi ces trois caractéristiques :

- **C**ohérence : toutes les copies des données sont synchronisées.
- **D**isponibilité : chaque demande reçoit une réponse, même si certaines données ne sont pas totalement à jour.
- **T**olérance au Partitionnement : le système continue de fonctionner malgré les pannes réseau.
  - ✓ Par exemple, **MongoDB** et **Cassandra** privilégient souvent la disponibilité et la tolérance au partitionnement, ce qui les rend adaptés aux systèmes distribués, mais au détriment de la cohérence stricte. Les SGBDR, quant à eux, maintiennent la cohérence et suivent les règles ACID, ce qui les rend préférables pour les transactions critiques.

Limite du CAP : Un système distribué peut maximiser aux plus deux propriétés simultanément.

- **CA** (**C**ohérence + **D**isponibilité) : Impossible si des partitions réseau existent.
- **CP** (**C**ohérence + **T**olérance aux partitions) : Réponse lente pendant une panne réseau.
- **AP** (**D**isponibilité + **T**olérance aux partitions) : Cohérence éventuelle seulement [5]

Différences fondamentales :

ACID (SGBDR)	CAP (NoSQL)
Transactions fiables	Flexibilité et performance
Cohérence stricte	Cohérence éventuelle
Centralisé	Distribution native
Adapté aux systèmes critiques	Optimisé pour les systèmes scalables

Tableau 1 : Différence Fondamentale entre ACID Vs CAP

2. Comparaison sur le stockage

SGBDR (Relationnel) utilise un modèle tabulaire où les données sont stockées sous forme de tables avec des lignes et colonnes, ce qui convient aux données structurées.

Le stockage est rigide en raison du schéma fixe, ce qui peut rendre les modifications de structure plus complexes.

Convient aux applications nécessitant des données fortement structurées et une intégrité référentielle élevée.

NoSQL (Non-relationnel) offre des modèles de stockage flexibles (clé-valeur, document, colonne, et graphe), qui permettent une meilleure adaptation aux données non structurées ou semi-structurées. Le schéma est souvent dynamique, permettant d'ajouter des champs sans besoin de restructurer la base de données. Idéal pour les applications nécessitant une scalabilité horizontale, où les données sont massives et hétérogènes [5].

Aspect	SGBDR	NoSQL
Modèle de stockage	Relationnel (tables, colonnes)	Clé-valeur, document, colonnes larges, graphes
Schéma	Schéma fixe	Schéma flexible ou sans schéma
Évolutivité	Verticale (plus de puissance machine)	Horizontale (ajout de serveurs)
Cas d'utilisation	Données structurées (transactions financières)	Données semi-structurées ou non structurées (Big Data, IoT)

Tableau 2 : Comparaison (SGBDR vs NoSQL) sur le stockage

3. Comparaison sur la gestion de la cohérence des données

SGBDR suit strictement les règles ACID, assurant une cohérence forte des données. Les transactions dans les SGBDR respectent l'intégrité référentielle, empêchant les incohérences grâce aux relations entre les tables (clés primaires, clés étrangères).

Adapté pour les applications où la cohérence des données est cruciale, comme les systèmes financiers. En revanche, les systèmes **NoSQL**, en particulier ceux suivant le théorème **CAP**, optent souvent pour une cohérence éventuelle plutôt qu'une cohérence forte. Cela signifie que les mises à jour sont propagées de manière asynchrone, et tous les nœuds finiront par avoir les mêmes données. La cohérence éventuelle est acceptable pour les applications où la disponibilité et la vitesse de traitement sont plus importantes que la cohérence immédiate, **par exemple** : réseaux sociaux et le E-commerce [5].

Critères	SGBDR	NoSQL
Cohérence stricte	Toujours garantie (transactions ACID)	Souvent éventuelle (BASE)
Modèle de cohérence	Modèle de verrouillage ou transactions sérialisées	Modèle éventuel pour haute disponibilité
Compromis	Performance réduite pour garantir la cohérence stricte	Favorise performance et disponibilité

Tableau 3 : Comparaison (SGBDR vs NoSQL) sur la gestion de la cohérence des données

4. Comparaison sur la disponibilité des données

Les SGBDR sont généralement conçus pour fonctionner sur un seul serveur, ce qui peut limiter leur disponibilité en cas de panne du serveur.

Il est possible d'implémenter des solutions de haute disponibilité, mais elles nécessitent souvent des architectures complexes comme les clusters de réplication.

La disponibilité peut être limitée dans les applications nécessitant des temps de réponse très faibles et des volumes de données massifs. Alors que les bases de données NoSQL sont conçues pour une scalabilité horizontale et sont réparties sur plusieurs nœuds, ce qui améliore leur disponibilité.

En cas de panne d'un nœud, les autres nœuds continuent de répondre aux requêtes, garantissant ainsi une meilleure disponibilité. Bien adapté aux applications en temps réel nécessitant une haute disponibilité, comme les plateformes de streaming et les systèmes IoT [5].

Critère	SGBDR	NoSQL
Disponibilité garantie	Dépend du serveur principal	Répartition sur plusieurs nœuds
Gestion des pannes	Impact significatif	Haute disponibilité grâce à la réplication
Adapter pour	Applications monolithiques	Systèmes distribués et scalables

Tableau 4 : Comparaison (SGBDR vs NoSQL) sur la disponibilité des données

5. Comparaison sur la gestion de la sécurité

SGBDR offre des mécanismes de sécurité robustes comme le contrôle d'accès basé sur les rôles (RBAC), le chiffrement des données, et les journaux de transactions.

Les SGBDR sont souvent préférés pour les applications nécessitant une sécurité élevée en raison de leurs normes de sécurité bien établies et de leur compatibilité avec les régulations comme GDPR.

Idéal pour les systèmes financiers, gouvernementaux, et les entreprises ayant des exigences strictes en matière de conformité. Bien que les bases de données NoSQL évoluent en termes de sécurité, elles peuvent être moins robustes que les SGBDR dans ce domaine, particulièrement pour les systèmes distribués. La sécurité dépend largement de l'implémentation du fournisseur NoSQL et des configurations personnalisées. Convient pour des applications nécessitant une sécurité modérée, mais peut nécessiter des configurations supplémentaires pour se conformer aux normes de sécurité élevées [5].

Critère	SGBDR	NoSQL
Contrôles d'accès	Granulaires (rôles, permissions)	Souvent limités ou personnalisés
Chiffrement des données	Support natif	Dépend de l'implémentation
Authentification	Intégration forte (LDAP, Kerberos)	Varie selon la base (OAuth, JWT)
Audit et logs	Support standard	Variable selon la solution

Tableau 5 : Comparaison (SGBDR vs NoSQL) sur la gestion de la sécurité

On peut en conclure que les SGBDR et NoSQL répondent à des besoins différents :

- **SGBDR** : Idéal pour les systèmes critiques nécessitant une cohérence stricte, une sécurité avancée et une gestion fiable des transactions.
- **NoSQL** : Privilégié pour les environnements modernes et distribués nécessitant une haute disponibilité, une flexibilité structurelle, et la capacité de gérer de grands volumes de données non structurées [5].

Conclusion & Perspectives

1. Conclusion

Les Big Data représentent une révolution majeure dans la manière dont les données sont collectées, analysées et exploitées. Cette étude approfondie sur le stockage des Big Data et les bases de données NoSQL a permis de mieux comprendre les défis et opportunités liés à ces technologies. Les bases NoSQL, en tant qu'alternative aux bases de données relationnelles traditionnelles, offrent une flexibilité inédite pour traiter des données massives, variées et en constante évolution.

Les bases de données relationnelles (SGBDR), grâce à leur structuration rigoureuse et leur conformité au modèle ACID, ont été pendant des décennies la référence pour la gestion des données. Cependant, l'ère des Big Data a révélé leurs limitations face à des besoins modernes, notamment la gestion de données non structurées, la scalabilité horizontale, et les performances en temps réel. Les bases NoSQL se sont imposées comme une solution innovante, capable de relever ces défis grâce à leur diversité, leur architecture flexible et leur compatibilité avec des systèmes distribués.

Les bases NoSQL se déclinent en plusieurs types, chacun adapté à des cas d'usage spécifiques :

- Clé-valeur : Optimisées pour des recherches rapides et des données simples ;
- Documents : Idéales pour des données semi-structurées, telles que des profils utilisateur ou des fichiers JSON ;
- Graphes : Adaptées à la modélisation des relations complexes, comme dans les réseaux sociaux ou les systèmes de recommandation ;
- Colonnes : Performantes dans les analyses massives et les requêtes agrégées sur de larges ensembles de données.

Malgré leurs atouts, les bases NoSQL ne sont pas exemptes de limites. Le manque de standardisation, la complexité de leur gestion de la cohérence des données dans des systèmes distribués (notamment en raison du théorème CAP), et la faible prise en charge native des transactions complexes sont des défis importants à relever. Ces inconvénients, bien que contraignants, n'entament pas leur potentiel d'évolution ni leur complémentarité avec les bases relationnelles.

Cette complémentarité est au cœur de la transformation numérique actuelle. Alors que les bases relationnelles excellent dans les environnements où la cohérence stricte et les relations complexes sont nécessaires, les bases NoSQL brillent dans des domaines nécessitant rapidité, flexibilité et scalabilité. Ainsi, loin d'être en opposition, ces deux approches se renforcent mutuellement, créant un écosystème technologique complet et robuste.

2. Perspectives

À l'avenir, plusieurs pistes de développement et d'amélioration peuvent être envisagées pour exploiter davantage le potentiel des bases de données dans le contexte des Big Data :

- Hybridation des systèmes : Une approche combinant les avantages des bases relationnelles et NoSQL pourrait émerger, offrant à la fois scalabilité et support des transactions complexes ;
- Optimisation de la cohérence : Les bases NoSQL peuvent évoluer pour mieux répondre aux exigences des systèmes critiques (banques, santé) en améliorant la gestion des transactions tout en maintenant leur performance ;
- Intégration de l'intelligence artificielle : Les bases de données pourraient intégrer des mécanismes d'IA pour optimiser le stockage, la recherche et l'analyse des données ;
- Sécurité renforcée : Les bases NoSQL devront adopter des mécanismes de sécurité avancés pour répondre aux défis croissants liés à la protection des données sensibles ;
- Adoption étendue dans les secteurs traditionnels : Bien que les NoSQL soient populaires dans les secteurs technologiques, leur adoption dans des industries traditionnelles (manufacture, agriculture) est une perspective prometteuse.

En conclusion, l'essor des Big Data et des bases NoSQL marque une révolution dans la manière dont nous stockons, analysons et exploitons les données. Ce domaine en constante évolution continuera d'apporter des solutions innovantes pour répondre aux défis des prochaines décennies

Références

1. Deghmani, F. (2023). Introduction au BIG DATA : Concepts et Technologies. *Revue de l'Information Scientifique et Technique*, *27*(1), 25-35. https://www.asjp.cerist.dz/index.php/en/downArticle/134/27/1/220365
2. Gana, A., & Bouhenika, A. *Déploiement d'une base de données NoSQL avec MongoDB* (Doctoral dissertation, UNIVERSITY OF KASDI MERBAH OUARGLA). https://dspace.univ-ouargla.dz/jspui/bitstream/123456789/31109/1/Gana-Bouhenika.pdf
3. HEMAIZIA, B. (2022). Contrôle de qualité de données NoSQL. https://dspace.univ-guelma.dz/jspui/bitstream/123456789/13181/1/HEMAIZIA_BASSEM_F5.pdf
4. JUGANARU, M. (2022). Bases de données NoSQL. https://ai4africa.github.io/DSS22/resource/nosql.pdf
5. AMARA, M. W. (2015). Etude comparative des bases de données NoSQL (Doctoral dissertation).