People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University of 8 May 1945-Guelma-

Faculty of Mathematics, Computer Science and Science of Matter

Department of Computer Science

Master Thesis

Specialty : Computer Science

Option : Computer systems

Theme

Détection et réparation automatique des données de schéma flexible

Jury members :

Dr. ZEDADRA AMINA (Chairman)

Dr. AGGOUNE AICHA (Supervisor)

Dr. HANNOUSSE ABDELHAKIM (Examiner)

Presented by :

Boukara Akram

June 2023

Remerciements

Je commence par exprimer ma gratitude envers Dieu le tout-puissant pour nous avoir donné la force et la patience nécessaires pour mener à bien ce travail malgré les obstacles rencontrés. Je tiens à exprimer mes sincères remerciements à mon superviseur de mémoire de fin d'études, Dr Aicha AGGOUNE, pour ses conseils précieux et son orientation tout au long du projet.

Je souhaite également exprimer ma gratitude à toutes les personnes qui ont contribué au succès de mon projet et qui m'ont aidé dans la rédaction de ce mémoire. Enfin, je suis honoré que tous les membres du jury aient accepté de juger ce travail et je les remercie de leur temps et de leur attention.

Dédicaces

À

mes parents, qui ont été ma source inépuisable de soutien, d'encouragement et de bienveillance tout au long de mon parcours académique. Votre amour inconditionnel et votre confiance en moi ont été les fondations solides sur lesquelles j'ai pu construire mes ambitions et atteindre mes objectifs. Je vous dédie ce mémoire en témoignage de ma gratitude éternelle.

RÉSUMÉ

Le sujet de cette étude est axé sur le contrôle de qualité des données NoSQL, en se concentrant spécifiquement sur les données flexible à savoir les documents JSON et les données orientées document de base de données NoSQL MongoDB. Pour y parvenir, nous avons proposé la méthode RQDB (Repair Quality DataBase) pour détecter et réparer les problèmes de la qualité de données flexibles comme la redondance et l'incomplétude.
RQDB a été validée par l'implémentation de notre outil et évaluée sur des données flexibles précitées. Les résultats obtenus sont intéressants.

Mots clés : Qualité de données, Schéma flexible, MongoDB, JSON, Détection des problèmes, Réparation, Vérification.

ABSTRACT

The topic of this study is focused on the control of data quality, specifically on flexible data such as JSON document and data oriented documented of NoSQL MongoDB database. To achieve this, we have proposed RQDB (Repair Quality DataBase) to detect and repair the data quality issues for example duplication, and incompleteness.

RQDB allows to : (1) detect data quality problem, (2) repair the data and (3) verify the quality.

RQDB has been validated by implementing our tool using the aforementioned flexible data. The obtained results are interesting.

Keywords : Data quality, Flexible schema, MongoDB, JSON, problem detection, Repairing, Verification.

TABLE DES MATIÈRES

TABLE DES FIGURES

INTRODUCTION GÉNÉRALE

L'évolution exponentielle de la quantité et de la variété des données générées par les entreprises et les applications modernes a mis en évidence la nécessité de solutions de gestion de données plus flexibles et adaptées aux besoins changeants. Les bases de données traditionnelles basées sur le modèle relationnel ont été conçues pour manipuler des données structurées avec des schémas fixes, ce qui limite leur capacité à gérer efficacement des données non structurées et semi-structurées[30].

Les bases de données NoSQL se distinguent par leur capacité à stocker et à traiter des données non structurées, semi-structurées et structurées de manière flexible. Contrairement aux bases de données relationnelles, elles ne nécessitent pas de schéma prédéfini, ce qui permet une évolutivité horizontale, une répartition facile des données et une adaptation aux changements de structure des données. Les bases de données NoSQL sont conçues pour répondre aux exigences de performances, de disponibilité et d'évolutivité des applications modernes, telles que les applications Web à fort trafic, les applications mobiles et les systèmes de gestion de contenu[30].

Dans ce mémoire de fin d'étude, nous explorerons en détail les bases de données NoSQL et les autres déferants format JSON,BSON,Avro... en mettant l'accent sur leur utilisation pour la gestion des données de schéma flexible. Nous analyserons les principaux types de bases de données NoSQL, tels que les bases de données orientées

document, orientées colonne, orientées graphe et clé-valeur, en examinant leurs caractéristiques, leurs avantages et leurs limites. Nous discuterons également des différentes approches de modélisation de données pour les bases de données NoSQL, en mettant en évidence les bonnes pratiques et les considérations clés.

En conclusion, ce mémoire vise à approfondir notre compréhension beaucoup plus les données de schéma flexible comme NoSQL et JSON...

Pour atteindre cet objectif, notre travail se compose des éléments suivants :

- Étude des données NoSQL, plus précisément des données orientées documents de MongoDB.

- Revue des concepts liés à la qualité de données.

- Revue des techniques les plus récentes pour la gestion de la qualité de données NoSQL.

- Proposition d'une méthode pour le contrôle de qualité des données en fonction de trois dimensions de qualité : la consistance de la représentation, la complétude et la concision. Ces trois dimensions de qualité ont un impact sur d'autres dimensions de qualité, comme la cohérence, la facilité d'utilisation et la facilité de compréhension.

- Implémentation de l'outil RQDB (Repair Quality of Document oriented data-Base) pour valider et évaluer les performances de nos propositions.

INTRODUCTION GÉNÉRALE

Ce mémoire est organisé en 4 chapitres :

- Chapitre 01 : Les données de schémas flexibles

Dans ce chapitre, nous proposons une revue de l'état de l'art des schémas flexibles, en mettant en évidence les différents modèles de données et leurs distinctions. Nous conclurons ce chapitre en explorant les différentes familles de schémas flexibles, offrant ainsi une vue d'ensemble complète de ce domaine.

- Chapitre 02 : Qualité des données flexibles

Ce chapitre débute en fournissant une introduction générale sur les dimensions de qualité des données, mettant ensuite l'accent sur les particularités des données NoSQL. Il explore ensuite les différentes techniques disponibles pour soutenir le contrôle de qualité des données, tout en mentionnant quelques travaux récents dans le domaine du contrôle des données NoSQL.

- Chapitre 03 : nous abordons les problèmes de qualité de données spécifiques aux bases de données NoSQL orientées documents, ainsi que les documents JSON du web. Ensuite, nous exposons notre approche proposée, nommée RQDB (Repair Quality DataBase), pour le contrôle de qualité des données orientées documents.

- Chapitre 04 : Implémentation de l'outil RQDB Plus (Repair Quality dataBase Plus)

Au cours de ce chapitre, nous détaillons l'implémentation de notre outil RQDB Plus (Repair Quality dataBase Plus) qui vise à valider la méthode RQDB pour le contrôle de qualité des données NoSQL orientées document. Nous mettons en lumière les différentes étapes et fonctionnalités de cette implémentation afin de fournir une validation concrète de notre approche.

CHAPITRE 1

LES DONNÉES DE SCHÉMAS FLEXIBLES

1.1 Introduction

La flexibilité de schémas de données peut engendrer des problèmes de qualité des données et de difficulté de leur contrôle. Ainsi, de nombreux chercheurs se sont penchés sur l'étude des mesures pour évaluer la qualité des données stockées dans des bases de données NoSQL avec des schémas flexibles.

Dans ce chapitre , nous allons étudier les models de donnés et la différances entre eux . Nous terminerons le chapitre par les différentes familles des schémas flexible .

1.2 Schéma flexibles

Un schéma flexible de données signifie que la structure des données n'est pas prédéterminée et peut évoluer au fil du temps sans nécessiter de modifications complexes du schéma existant. Cela signifie que chaque document ou enregistrement peut avoir un nombre différent de champs et que les types de données de ces champs peuvent également varier.

Cela contraste avec les bases de données relationnelles, où les données sont organisées en tables avec des colonnes prédéfinies et des types de données fixes. Les bases de données NoSQL avec un schéma flexible sont souvent utilisées pour des applications qui nécessitent une grande flexibilité dans la structure des données et un temps d'exécution rapide, telles que les bases de données de Big Data et de gestion de données semi structurées [1].

1.3 Format de schémas flexibles

Il existe plusieurs formats de schémas flexibles, Ces différents formats de schémas flexibles offrent une grande flexibilité pour la représentation de données, ce qui les rend particulièrement utiles dans des domaines tels que le Big Data et l'IoT, où la structure des données peut être très variable. Voici quelques-uns des formats les plus courants :

1.3.1 Format JSON (JavaScript Object Notation)

JSON ou JavaScript Object Notation, est un format de données textuelles non structuré qui utilise des paires clé-valeur et des listes ordonnées pour représenter des données structurées. Bien que dérivé de JavaScript, JSON est pris en charge par la plupart des langages de programmation grâce à des bibliothèques dédiées ou une prise en charge native[2].

1.3.1.1 Fichiers JSON

Les fichiers contenant des données JSON ont une extension ".json" et sont des fichiers texte clairs, conformément au principe de JSON de rendre tout le code lisible. Par conséquent, ces fichiers peuvent être facilement ouverts et examinés [2].

1.3.1.2 Intérêt du JSON

Le format JSON présente plusieurs avantages, notamment[3] :

- Il est facile à lire et à écrire pour les humains.

- Il est facilement interprétable et utilisable par les langages de programmation.

-Il est plus adapté aux applications Web modernes, car il permet de transférer des données entre le serveur et le client de manière asynchrone et efficace

1.3.1.3 Inconvénient du JSON

Les principaux inconvénients de JSON sont[2] :

- Pas de validation de schéma intégré , Contrairement à XML, JSON ne dispose pas de validation de schéma intégrée pour vérifier la structure et la cohérence des données.

- Pas de support pour les commentaires.

- Il ne peut pas représenter directement les types de données binaires tels que les images, les fichiers audio ou vidéo, etc.

- Il ne prend pas en charge nativement la transformation de données, ce qui peut rendre le traitement de grandes quantités de données plus difficile.

- Contrairement à XML, il n'y a pas de standardisation pour les schémas JSON, ce qui peut rendre l'interopérabilité entre différentes applications plus difficile

1.3.2 Format XML (eXtensible Markup Language)

Le XML désigne un langage informatique Comme le langage HTML sous un format de données structurées utilisé pour stocker et échanger des données entre applications. Les données sont représentées sous forme de balises et peuvent être hiérarchisées pour refléter la structure des données. [4]

1.3.2.1 Fichiers XML

Un fichier XML est un document texte que vous pouvez enregistrer avec l'extension ".xml" . Vous pouvez écrire du XML de la même manière que d'autres fichiers texte.

1.3.2.2 Intérêt du XML

Le XML est de plus en plus utilisé car il permet de structurer les données de manière plus fiable que les fichiers binaires ou tabulaires. Cette structure facilite le traitement informatique, notamment sur Internet, les réseaux internes, les tableurs et les bases de données, tout en conservant un format texte lisible et éditable par l'être humain sans avoir besoin d'outils particuliers. [4]

1.3.2.3 Inconvénient du XML

par contre à l'intérêt du XML, voici quelleque inconvénient [5] :

- La structure du fichier XML peut devenir complexe et difficile à lire pour les personnes non familières avec le langage.

- La taille des fichiers XML peut être plus importante que les fichiers de données binaires.

- Le XML peut être moins efficace pour la transmission de données en temps réel en raison de la surcharge de traitement nécessaire pour convertir le texte en données exploitables.

- Le XML peut nécessiter l'utilisation de logiciels spécifiques pour être traité correctement, ce qui peut entraîner des coûts supplémentaires en termes de formation et d'acquisition de logiciels.

1.3.3 Format YAML (YAML Ain't Markup Language)

YAML est un langage de sérialisation des données qui est souvent utilisé pour coder des fichiers de configuration selon un format de données textuelles qui est facile

à lire et à écrire pour les humains. Les données sont représentées sous forme de paires clé-valeur, avec une syntaxe qui utilise l'indentation pour indiquer la structure des données.

1.3.3.1 Fichiers YAML

YAML, est une technologie de format de fichier utilisée pour les documents. Ces documents sont enregistrés sous forme de fichiers texte brut et portent l'extension ".yml" .

1.3.3.2 Intérêt du YAML

Le format YAML présente plusieurs avantages, notamment[6] :

- Sa flexibilité, qui permet aux développeurs de définir des données dans une syntaxe concise et facilement modifiable.

-Il est facile à comprendre pour les humains et est souvent utilisé pour configurer des applications et des systèmes.

- les fichiers YAML peuvent être ajoutés au système de contrôle du code source, tel que GitHub, afin que les modifications puissent être suivies et vérifiées.

1.3.3.3 Inconvénient du YAML

Voici quelques inconvénients potentiels du format YAML[7] :

- YAML est plus complexe que certains formats de données simples.

- Une simple erreur d'espacement dans l'indentation peut causer le dysfonctionnement du code.

- En raison de la nature de YAML comme format de texte brut, il peut être vulnérable aux attaques d'injection de code si les données sont traitées sans validation appropriée.

1.3.4 Format BSON (Binary JSON)

BSON (Binary JSON) est un format de données binaire basé sur JSON, qui offre des performances plus rapides pour les opérations de lecture et d'écriture. Ce dernier encode également des informations de type et de longueur, ce qui le rend plus facilement analysable par les machines. [8]

1.3.4.1 Intérêt du BSON

Voici les avantages du BSON [8] :

- BSON est plus compact que JSON, ce qui signifie que les données peuvent être stockées et transférées plus rapidement.

- Il prend en charge des types de données supplémentaires, tels que les dates et les binaire.

- Il est conçu pour être facilement analysé et traité par les machines, grâce à l'encodage des informations de type et de longueur.

1.3.4.2 Inconvénient du BSON

Voici les inconvénients du BSON [8] :

- BSON n'est pas facilement lisible par l'homme, ce qui peut rendre le débogage et le dépannage plus difficiles.

- BSON nécessite plus de ressources pour être traité, car il doit être converti en JSON ou en une autre représentation lisible par l'homme pour l'affichage et la manipulation.

- BSON n'est pas aussi largement pris en charge que JSON, ce qui peut limiter son utilisation dans certains environnements et bibliothèques.

1.3.5 Format Avro

Avro est un système open source de sérialisation de données sous format de données binaire qui utilise un schéma pour décrire la structure des données. Le schéma

peut être évolutif, ce qui signifie qu'il peut être mis à jour sans avoir besoin de réécrire les données existantes.

1.3.5.1 Fichiers Avro

L'extension de fichier Avro fait référence à un format de fichier de données binaire open source développé par Apache Software Foundation.L'extension de fichier pour Avro est ".avro".

1.3.5.2 L'intérêt du Avro

Voici quelques avantages de l'utilisation d'Avro : [9]

- Avro est plus efficace pour les opérations de traitement de données volumineuses à forte écriture.

- Avro prend également en charge le streaming de données, ce qui en fait un choix populaire pour les applications de traitement de flux de données en temps réel.

- Avro fournit un schéma qui permet de décrire la structure des données de manière indépendante du langage de programmation. Cela permet une interopérabilité entre différents systèmes et langages de programmation.

1.3.5.3 L'inconvénient du Avro

Voici quelle que inconvénients du Avro : [10]

- Pour lire/écrire des données, il est nécessaire d'avoir un schéma.

- La sérialisation peut potentiellement être plus lente.

- Les messages Avro ont besoin d'être accompagnés d'un schéma, ce qui peut augmenter la complexité et le coût de stockage des données.

1.4 Données NoSQL

Les données NoSQL (Not Only SQL) représentent un exemple de données de schéma flexible stockées dans des bases de données et gérées par un système de gestion de base de données NoSQL.Voila quekque defference entre SQL et NoSQL :

	SQL	NoSQL
Type	Relationnelle	Non-Relationnelle
Données	Données structurées	Données non-structurées
Schéma	Statique	Dynamique
Scalabilité	Verticale	Horizontale
Langage	Structured Query Language	Un-structured Query language
Flexibilité	Rigide	Flexible
Support	Support éditeurs	Support communauté Open-Source

FIGURE 1.1 – Tableau déférence entre SQL et NoSQL.

Quatre grandes familles se distinguent parmi celles-ci :

1.4.1 Bases de données clé-valeur

En termes de simplicité, le modèle clé-valeur est l'un des types de bases de données les plus simples. Il permet d'accéder aux données en utilisant une clé spécifique. La particularité de ce modèle réside dans sa capacité à stocker des données sans avoir à définir un schéma spécifique. Il utilise des opérations basiques telles que le stockage, la récupération et la suppression de données, qui sont organisées sous forme de paires clé/valeur. Ce type de base de données est extrêmement efficace pour les opérations de lecture et d'écriture, et est conçu pour s'adapter à une grande échelle tout en offrant des temps de réponse très rapides.

Les bases de données clé-valeur stockent les données dans un tableau de hachage où chaque clé est unique. La valeur associée à chaque clé peut prendre la forme d'un

JSON, d'un objet BLOB, d'une ligne de code ou d'autres types de données. Les opérations principales supportées par ces bases de données sont les suivantes :

- La commande PUT permet d'ajouter ou de mettre à jour une nouvelle paire clé-valeur.

- La commande GET permet de récupérer la valeur associée à une clé spécifique.

- La commande DELETE permet de supprimer une clé et sa valeur de la table.

Certains exemples de bases de données clé-valeur populaires sont Redis, Riak, Oracle NoSQL et Microsoft Azure Table Storage [1] [10].

1.4.2 Bases de données orientées colonnes

Les bases de données orientées colonnes sont une extension des bases de données clé-valeur, avec des colonnes et des caractéristiques empruntées aux bases de données relationnelles. Elles sont particulièrement adaptées aux scénarios où les ajouts de données sont fréquents plutôt que les mises à jour. Cependant, elles ne prennent pas en charge les relations directes entre les données, ce qui nécessite la création de structures en réseau pour représenter les connexions entre les données, ce qui peut ne pas être la solution la plus efficace. Ces bases de données sont souvent utilisées pour le stockage de grandes quantités de données avec une haute disponibilité. La plupart des systèmes orientés colonnes prennent également en charge les données géospatiales, mais se concentrent principalement sur les requêtes simples nécessitant des réponses rapides. Des exemples de bases de données orientées colonnes comprennent HBase, Cassandra, Big Table, DynamoDB et Accumulo, qui sont utilisées par des géants tels qu'Amazon, Google et Facebook [10].

1.4.3 Bases de données orientées document

Les bases de données orientées documents sont conçues pour gérer efficacement de grandes quantités de données. Chaque document est associé à une clé et peut

contenir plusieurs paires clé-valeur imbriquées. Les documents peuvent être importés dans des formats standard tels que XML (Extensible Markup Language), JSON (JavaScript Object Notation) ou BSON (Binary JSON). Par rapport aux bases de données clé-valeur, les bases de données orientées documents permettent de rechercher le contenu des documents, et non seulement leur clé. La présence de documents imbriqués facilite les recherches avancées et ne nécessite pas de schémas prédéfinis.

Les bases de données orientées documents offrent la possibilité de décrire un document avec un grand nombre de valeurs et de prendre en charge un schéma flexible. Elles sont capables de stocker une grande quantité de données sous différents formats. La relation entre les documents peut être représentée par l'intégration de documents imbriqués (Embedded documents) ou par l'utilisation de références. Ainsi, les requêtes portant sur les documents d'une même collection peuvent être traitées très rapidement. L'utilisation de documents XML peut améliorer les bases de données orientées documents en offrant des fonctionnalités complémentaires telles que XQuery, XPath et XPointer, ainsi que le concept de relation qui n'existait pas auparavant. CouchDB et MongoDB sont principalement utilisés comme solutions basées sur le concept de bases de données orientées documents, et Twitter en est un utilisateur notable [10].

1.4.4 Base de données orientées graphe

Les bases de données orientées graphe sont un type de base de données NoSQL qui utilise la théorie des graphes pour stocker, mapper et interroger les relations entre les données. Elles sont composées de nœuds et de relations, où chaque nœud représente une entité et chaque relation représente une connexion entre les nœuds.

Ces bases de données gagnent en popularité dans le domaine de l'analyse des interconnexions. Elles sont capables de gérer des données relationnelles et sont particulièrement adaptées aux données en réseau telles que les routes et les réseaux sociaux.

Cependant, contrairement aux autres types de bases de données NoSQL, les bases de données orientées graphe ne sont pas optimales pour les grandes quantités de données qui ne sont pas suffisamment connectées. Oracle Graph, Neo4j, Arango et OrientDB sont des exemples de systèmes de bases de données orientées graphe qui sont utilisés par des entreprises telles que Walmart et eBay [10].

1.5 Conclusion

Dans cette partie, Nous avons présenté les données de schémas flexibles, l'extension, les intérêts et les limites du chaque format de document flexible.

Dans le prochain chapitre, nous aborderons le concept de qualité des données dans les données flexibles, ainsi que ses différentes dimensions et quelques métriques de contrôle qualité. Nous présenterons également les travaux les plus récents sur le contrôle qualité des flexibles.

CHAPITRE 2

QUALITÉ DES DONNÉES FLEXIBLES

2.1 Introduction :

Les organisations sont confrontées à des enjeux majeurs en termes de performance d'analyse de données et de rentabilité, notamment en ce qui concerne la qualité des données. En effet, des données de qualité permettent aux entreprises d'améliorer leurs performances opérationnelles, de satisfaire leur clientèle et d'être plus compétitives en adaptant rapidement leur stratégie d'entreprise.

Dans ce chapitre, nous découvrons les dimensions générales de la qualité des données, en se concentrant particulièrement sur les données NoSQL.

En suite, nous présentons les techniques d'aides au contrôle de données ainsi que les travaux récents sur le contrôle de qualité des données NoSQL y sont également présentés[28].

2.2 Qu'est-ce que la qualité des données ?

Le concept de qualité des données (QD ou data quality en anglais) est largement connu dans la communauté des bases de données et est un domaine de recherche actif

depuis de nombreuses années [17]. Selon [32], la qualité des données n'est pas facile à définir et ses définitions varient selon le domaine d'application des données.

La définition de la qualité des données donnée par Sidi et al[25] correspond à la mesure de l'adéquation entre l'utilisation des données et les besoins des utilisateurs ainsi que leur niveau de satisfaction.

Devillers et ses collaborateurs [19] ont défini la qualité de données comme étant la conformité entre les caractéristiques des données et les besoins explicites et/ou implicites d'un utilisateur dans un domaine d'application donné.

Juran et al [24] ont défini la qualité comme "l'aptitude à l'emploi" (ou « Fitness for use »), c'est-à-dire la capacité d'un produit ou d'un service à répondre aux besoins des membres de société dans des conditions opérationnelles spécifiées. La qualité des données, appelée également Fitness to use, fait référence à cette même notion d'aptitude à l'emploi en ce qui concerne les données et leur utilisation.

La définition de Joseph M. Juran, qui est cité dans le livre de référence [24], a obtenu un accord officiel entre les organismes de normalisation tels que l'ISO et les organisations internationales telles que l'IEEE.

La norme ISO/CEI 25012 (ISO/CEI, 2008) identifie alors, la qualité de données selon deux points de vue [22] :

Qualité des données inhérentes : Les caractéristiques de qualité des données ont la capacité intrinsèque de répondre aux besoins de données, qu'ils soient explicites ou implicites.

Qualité des données dépendante du système : fait référence à la mesure dans laquelle la qualité des données est maintenue et atteinte grâce à un système d'information. Cela dépend du contexte technologique spécifique dans lequel les données sont utilisées.

2.3 Dimensions d'évaluation de qualité de données

Il existe plusieurs caractéristiques ou dimensions qui peuvent être utilisées pour décrire la qualité des données, telles que la précision, la performance, la cohérence, la fiabilité, la complétude, l'actualité, etc. Les classifications de ces dimensions varient selon les auteurs et chaque type de classification est accompagné de ses propres définitions pour chaque dimension. Wang et Strong ont été les premiers à établir une classification des dimensions basée sur l'utilisateur en 1996 [34], qui comprenait 20 dimensions réparties en quatre catégories : intrinsèques, contextuelles, représentationnelles et d'accessibilité. Cette classification a été établie en se basant sur différentes synthèses et comparaisons entre les classifications des dimensions dans la littérature. [12, 13, 16]

2.3.1 Dimensions intrinsèques

Les dimensions intrinsèques de la qualité des données font référence aux caractéristiques objectives et inhérentes des données, telles que leur crédibilité, leur exactitude, leur objectivité et leur réputation[18].

Crédibilité (Believability) : Les données extraites sont considérées comme valides lorsqu'elles répondent à l'objectif d'utilisation prévu. La crédibilité est étroitement liée à la validité, car les données doivent être collectées selon les règles de gestion et les paramètres commerciaux définis, et elles doivent être présentées dans le bon format[18].

Exactitude (Accuracy) : L'exactitude ou la précision des données représente le degré de correspondance avec les vraies valeurs des items, aussi connues comme les valeurs correctes et précises. La précision peut être de deux types : la précision syntaxique, qui concerne la mesure dans laquelle les données correspondent aux valeurs correctes et précises, et la précision sémantique, qui concerne le degré d'exactitude des données par rapport aux valeurs du monde réel. On peut calculer l'exactitude en divisant le nombre d'éléments ou d'enregistrements exacts par le nombre total

d'éléments ou d'enregistrements. Par exemple, si un registre de la population locale contient 932 904 numéros de téléphone et que 813 942 d'entre eux ont été confirmés comme exacts, alors la précision serait de 87,25% (813 942/932 904 * 100)[18].

Objectivité (Objectivity) : La mesure dans laquelle les données d'un dataset sont créées de manière objective détermine l'objectivité des données. Cette caractéristique est étroitement liée à la crédibilité, et elle peut même renforcer cette dernière.

Réputation (Reputation) : Les données sont considérées comme répondant aux attentes des utilisateurs si ces derniers les trouvent utiles pour leur utilisation prévue. La réputation d'un ensemble de données est déterminée par l'opinion des utilisateurs sur les données en raison de leur expérience passée. Cette réputation est étroitement liée à la crédibilité des données.

2.3.2 Dimensions contextuelles

Ces dimensions sont liées aux données elles-mêmes par rapport au contexte de leur utilisation. Elles comprennent la valeur ajoutée, la pertinence, l'opportunité, la complétude et la quantité de données [29].

Valeur ajoutée (added value) : Cela signifie que la qualité des données est évaluée en fonction de leur utilité et de l'avantage qu'elles offrent lorsqu'elles sont utilisées[29].

Pertinence (Relevancy) : La pertinence des données fait référence à leur utilité pour l'entreprise ou l'utilisateur. La collecte de données qui ne répondent pas aux attentes ou aux exigences peut entraîner une perte de temps et d'argent[29]. **Métrique de pertinence :** On peut mesurer la pertinence des données qualitativement, en fonction des exigences de l'utilisateur. Une méthode courante consiste à utiliser une scorecard, un outil qui permet de prendre en compte toutes les dimensions de la performance, au-delà des seules mesures financières[29].

Opportunité (Timeliness) : La disponibilité des données en tout temps est un facteur clé pour leur utilisation. Pour s'assurer que les données sont facilement accessibles, elles doivent être mises à jour en temps réel. L'opportunité des données est donc étroitement liée à leur actualité (currency), qui reflète la mesure dans laquelle les informations sont à jour par rapport au monde qu'elles représentent et si elles sont correctes malgré les éventuels changements liés au temps[29].
Métrique : On peut mesurer l'actualité des données en fonction de leur fréquence de mise à jour prévue et de la vérification de leur actualisation, qui peut nécessiter des processus automatisés et manuels. Des règles peuvent être établies pour déterminer la "durée de vie" d'une valeur de données avant qu'elle ne doive être vérifiée et mise à jour.

Complétude (Completeness) : Est-ce que les informations sont complètes ? Les champs à renseigner le sont-ils ? D'autres champs utiles pourraient-ils être ajoutés ? La complétude est une dimension couramment utilisée pour évaluer la qualité des données, et elle mesure en général la présence des données.
Métrique : Plusieurs métriques sont utilisées pour évaluer la complétude, par exemple en calculant le nombre de valeurs manquantes pour un attribut. Il existe plusieurs approches pour traiter les données manquantes. L'approche la plus simple consiste à supprimer les données ayant des valeurs manquantes de l'ensemble de données. Une autre approche consiste à imputer les données manquantes en leur attribuant des valeurs de remplacement.

Quantité (Quantity) : Il est nécessaire d'avoir une quantité adéquate de données en fonction du contexte de la tâche à accomplir. Cette quantité peut être déterminée en fonction du nombre d'enregistrements et de variables requis pour le contexte en question.

2.3.3 Dimensions représentationnelles

Les dimensions représentationnelles comprennent des aspects liés au format des données (représentation concise et cohérente) et à la signification des données (interprétabilité et facilité de compréhension)[21].

Interprétabilité (Interpretability) : La notion d'interprétabilité est liée à la clarté et à la description des concepts de données. Elle renvoie à la mesure dans laquelle les données sont bien présentées et définies, de sorte qu'elles puissent être facilement comprises et interprétées[21].

Facilité de compréhension (ease to understanding) : La clarté et la compréhension des données dans un ensemble de données sont mesurées par leur capacité à être facilement interprétées, sans ambigüité et compréhensibles[21].

Consistance représentationnelle (Representational consistency) : La cohérence des données se réfère à leur présentation constante dans un format compatible avec les données antérieures. Elle permet de détecter les éventuelles divergences de données identiques présentes dans plusieurs sources, qui, si elles existent, indiquent que les données ne peuvent pas être considérées comme fiables et nécessitent une investigation[21].

Métrique : La cohérence est une mesure de la conformité aux contraintes d'intégrité dans les bases de données relationnelles. Elle est souvent représentée comme le pourcentage d'éléments ou d'enregistrements considérés comme cohérents. Par exemple, la date de naissance dans un registre de population doit être stockée dans le format "AAAA-MM-JJ" (année-mois-jour).

Représentation concise (Concise representation) : La concision des données se mesure à la fois par leur présentation brève mais complète et pertinente, sans être écrasante, et par l'absence de redondances dans l'ensemble de données[21].

Métrique : Il est possible de détecter et de traiter les données redondantes ou dupliquées (duplicate data) en éliminant les doublons, en particulier dans le cas de Big Data.

2.3.4 Dimensions d'accessibilité

Elles concernent la facilité d'accès aux données du système avec toute sécurité[27].

Accessibilité : L'accessibilité se réfère à la disponibilité des données et décrit à quel point elles sont facilement accessibles pour les utilisateurs. Contrairement aux autres dimensions de la qualité des données, il n'y a pas de mesure quantitative suggérée pour mesurer l'accessibilité. Cette dimension doit être évaluée qualitativement[27].

Sécurité : La sécurité des données est étroitement liée à leur accessibilité et à la manière dont elles sont protégées. Elle mesure le niveau de protection des données, les droits d'accès qui leur sont accordés, les politiques de stockage qui leur sont appliquées ainsi que les contraintes de sécurité mises en place pour assurer leur protection[27].

Traçabilité (Tracability) : La mesure dans laquelle les données sont bien documentées, vérifiables et facilement attribuables à une source[27].

Rentabilité (Cost-effectiveness) : La mesure de la faisabilité économique de la collecte de données appropriées[27].

Facilité d'utilisation (Ease of operation) : La facilité de gestion des données fait référence à la mesure dans laquelle les données sont facilement manipulées et traitées (par exemple, mises à jour, déplacées, agrégées, reproduites, etc.).

Variété de données et de sources de données (Variety of data and data sources) : La disponibilité des données à partir de diverses sources est mesurée en fonction de leur accessibilité à partir de multiples origines de données distinctes.

Flexibilité (Flexibility) : La mesure dans laquelle les données sont extensibles, adaptables et facilement applicables à d'autres besoins.

En raison de la flexibilité de schéma et de la structuration des données dans les bases de données NoSQL, les problèmes de qualité de données sont principalement liés au niveau du schéma, notamment à la consistance de la représentation des données orientées documents dans MongoDB. Quant aux valeurs de données, l'accent est mis sur le contrôle de la complétude et de la concision. Ces deux dimensions sont

respectivement liées aux problèmes de valeurs manquantes et de redondance de données .

2.4 Travaux récents sur le contrôle de qualité de données flexibles

Plusieurs travaux et frameworks ont été développés pour la gestion et le contrôle de données relationnelles, ainsi que pour les données liées du Web ouvert. [23, 31]

Nous présentons par ordre chronologique quelque travaux d'évaluation de la qualité de données NoSQL, que nous jugeons intéressant dans le contexte de notre travail :

2.4.1 Contrôle qualité pour les Big Data basées sur NoSQL

Dans leur étude intitulée A Systematic Literature Review and Future Research Directions", Wahyudin et al [33]. (2021) ont effectué une revue systématique de la littérature pour analyser les travaux sur le contrôle de qualité des données NoSQL publiés entre 2015 et 2020. Les auteurs ont mis en évidence les différents aspects de la qualité des données NoSQL, les méthodes et les outils proposés pour assurer la qualité des données, ainsi que les défis actuels dans ce domaine. Ils ont également proposé des pistes de recherche futures pour améliorer le contrôle de qualité des données NoSQL.

2.4.2 Framework de contrôle qualité pour les bases de données MongoDB

Dhanapal et al [20]. (2020) ont proposé un framework de contrôle de qualité pour les bases de données MongoDB, basé sur les dimensions de qualité de données telles que la cohérence, la complétude et la concision. Leur framework repose sur des règles de qualité pour chaque dimension, et inclut un processus de vérification de la qualité

des données. Cette approche peut aider les organisations à améliorer la qualité des données et la fiabilité des analyses basées sur les données MongoDB.

2.4.3 Contrôle de la qualité des données dans une base de données NoSQL orientée documents

Aljohani et al [15]. (2020) ont proposé une approche pour améliorer la qualité des données dans les bases de données NoSQL orientées documents en se concentrant sur la détection et la correction des problèmes de qualité de données. Cette approche repose sur des techniques de validation de schéma qui permettent de détecter les anomalies dans les données en comparant le schéma attendu avec le schéma réel des documents stockés dans la base de données. Les auteurs ont testé leur approche sur des bases de données MongoDB et ont montré son effcacité pour améliorer la qualité des données en identifiant les erreurs de schéma et en proposant des corrections.

2.4.4 Contrôle de qualité des données évolutif et efficace pour les magasins de données NoSQL

La méthode proposée par Zhang et al [35]. (2019) pour le contrôle de qualité de données dans les magasins de données NoSQL est basée sur l'apprentissage automatique et l'analyse de corrélation, et utilise une approche de classification et de corrélation pour détecter les erreurs de qualité de données. Cette méthode a été évaluée expérimentalement et a montré des performances améliorées en termes de précision et d'efficacité.

2.4.5 Un Framework pour le contrôle qualité des Big Data stockées dans MongoDB

Le travail de Silva et al [26]. (2018) propose un cadre pour le contrôle de qualité des données stockées dans des bases de données MongoDB. Ce cadre est basé sur

des indicateurs de qualité de données tels que la validité, l'intégrité et la cohérence. Le but principal de ce cadre est d'identifier et de corriger les erreurs de qualité des données dans les bases de données MongoDB afin d'améliorer les performances des systèmes d'information basés sur le big data. Pour ce faire, le cadre fournit des critères et des métriques pour évaluer la qualité des données stockées dans MongoDB. Les critères incluent la vérification de la validité des données, l'évaluation de l'intégrité des données et la mesure de la cohérence des données. Les métriques associées à chaque critère permettent de quantifier et de comparer la qualité des données stockées dans différentes bases de données MongoDB. La mise en place de ce cadre peut aider les organisations à améliorer la qualité de leurs données et à optimiser la performance de leurs systèmes d'information.

2.5 Conclusion

Nous avons exposé les définitions de la qualité des données, les différentes dimensions de qualité accompagnées de métriques de contrôle. En outre, nous avons exposé les études récentes portant sur le contrôle de qualité de données NoSQL.

Notre intérêt se porte particulièrement sur les aspects de qualité liés à la structuration des données dans les systèmes de données orientées document, tels que MongoDB. Le prochain chapitre exposera en détail notre proposition à cet égard.

CHAPITRE 3

LA MÉTHODE RQDB POUR LE CONTRÔLE DE QUALITÉ DE DONNÉES FLEXIBLES : ETUDE DE CAS DE DONNÉES NOSQL ORIENTÉES DOCUMENTS

3.1 Introduction

Les bases de données NoSQL orientées documents présentent un défi complexe en matière de garantie de la qualité des données. Cela s'explique principalement par la flexibilité du schéma, l'absence de contraintes d'intégrité, le volume important de données et l'hétérogénéité des données. Les entreprises du domaine informatique ont rencontré de nombreux problèmes de qualité lorsqu'elles ont cherché à travailler avec des données NoSQL.

Après avoir réalisé une revue de l'état de l'art sur les données de schéma flexible et les travaux récents sur la gestion de la qualité des données, ce chapitre se concentre sur l'étude des problèmes de qualité des données suivants : la duplication des données, l'incomplétude des données et la normalisation . Ensuite, nous présenterons notre proposition pour le contrôle de qualité des données orientées documents, qui repose sur une méthode de calcul de la fréquence des données appelée méthode RQDB

(Reapair Quality DataBase).

3.2 Contributions

Notre objectif est de proposer une méthode, appelée RQDB (Reapair Quality DataBase), pour contrôler et améliorer la qualité des données NoSQL orientées documents. Cette méthode vise à détecter et corriger les problèmes de qualité en se basant sur les éléments les plus fréquemment utilisés.

Nous avons défini la qualité des données orientées documents en nous appuyant sur quatre dimensions principales : la consistance représentationnelle, la représentation concise, la complétude et la normalisation . Ces dimensions ont un impact sur d'autres aspects de la qualité des données, tels que la précision, la concision, la cohérence et la facilité de compréhension.Notre méthode RQDB permet de détecter et réparer les problèmes suivants :

— L'incomplétude dans un document d'une collection de données NoSQL orientées documents se réfère à la présence de valeurs non déclarer ou bien valeur manquants. La méthode RQDB propose trois approches pour gérer ce problème :l'imputation par le mot NULL par defaut, la suppression de clé null carrément depuis son document ou bien laisse le document à l'origine. La détection de l'incomplétude permet d'assurer l'intégrité et la qualité des données en prenant des mesures appropriées pour combler les lacunes.

— La redondance des données dans les bases de données NoSQL orientées documents est identifiée lorsque la fréquence d'apparition d'une donnée est supérieure à 1. La méthode RQDB détecte ces redondances en analysant la fréquence d'apparition des données. Pour remédier à ce problème, la réparation consiste simplement à supprimer les documents redondants, assurant ainsi l'intégrité et la cohérence des données. Cette étape de contrôle de qualité permet d'optimiser l'utilisation de l'espace de stockage et de garantir des résultats précis lors des opérations de recherche et d'analyse.

— La dénormalisation des documents : la méthode RQDB vise à uniformiser la structure des documents en appliquant des transformations spécifiques. Cela inclut la conversion de toutes les clés en chaînes de caractères pour assurer une cohérence et une compatibilité entre les documents. De plus, les indentations sont standardisées en utilisant la bibliothèque JSON, facilitant ainsi la manipulation et l'analyse ultérieure des données. Cette étape de normalisation contribue à améliorer la qualité et l'interopérabilité des documents dans un environnement NoSQL orienté document.

3.3 Les problèmes de qualité de données NoSQL :

- L'incomplétude dans un document :

L'incomplétude des données peut résulter d'un manque d'information ou d'une valeur précise sur certains champs ou attributs des documents. Lorsqu'une donnée est incomplète, cela signifie que les valeurs de certains attributs sont inconnues ou manquantes. L'information générée ou acquise peut être considérée comme incomplète lorsqu'une ou plusieurs valeurs d'attributs sont absentes ou inconnues. Il est essentiel de détecter et de traiter ces données incomplètes afin de garantir l'intégrité et la fiabilité de nos données.La figure suivante illustre un exemple de document avec données incomplètes :

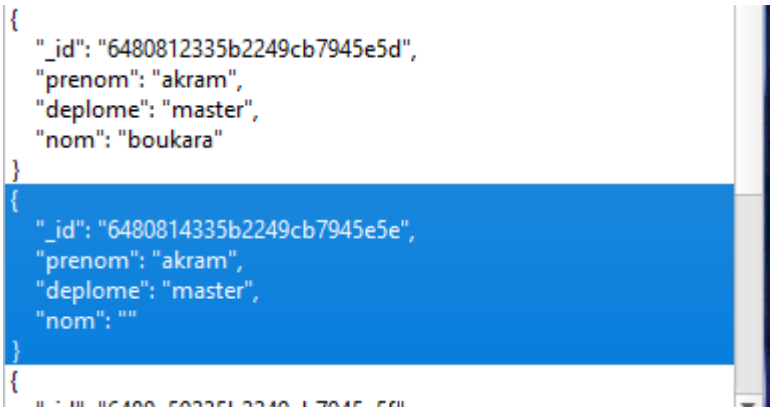

FIGURE 3.1 – Exemple de l'incomplétude de données d'un document

- La redondance des données :

La redondance des données est un problème courant dans les bases de données NoSQL orientées document. Bien qu'il ne puisse pas y avoir de duplication au niveau de l'ID, il peut y avoir une duplication au niveau d'autres attributs du document.

Cette redondance compromet la qualité de la base de données, car elle peut entraîner des incohérences et des erreurs. Il est donc essentiel de détecter et de résoudre ces problèmes de redondance pour assurer la qualité des données dans notre base de données.

Voici un exemple de collection avec des documents dupliquées :

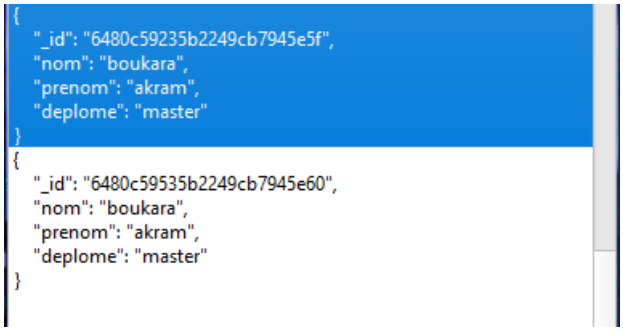

FIGURE 3.2 – Exemple de documents dupliquées.

- La normalisation des documents :

La différence entre des documents normalisés et des documents non normalisés réside dans la structure et l'organisation des données. Dans le cas des documents normalisés, les attributs sont standardisés et cohérents d'un document à l'autre, ce qui facilite la recherche et l'analyse des données.

En revanche, les documents non normalisés peuvent présenter des variations dans leur structure, avec des attributs qui diffèrent d'un document à l'autre, rendant la manipulation des données plus complexe.

La normalisation des documents vise à éliminer ces variations en appliquant des transformations spécifiques pour harmoniser la structure des documents, ce qui améliore la qualité et la cohérence des données dans un environnement de base de données orienté document.

Voici deux exemple des documents normalisé et non normalisé :

```
> db.test.insertOne({"name":"paul","age":35})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5956ab5fc598c62d7f03e789")
}
> db.test.find()
{ "_id" : ObjectId("5956aa4ec598c62d7f03e787"), "name" : "stephane" }
{ "_id" : ObjectId("5956aa80c598c62d7f03e788"), "name" : "stephane" }
{ "_id" : 0, "name" : "pierre" }
{ "_id" : ObjectId("5956ab5fc598c62d7f03e789"), "name" : "paul", "age" : 35 }
```

FIGURE 3.3 – Exemple de documents non normalisé.

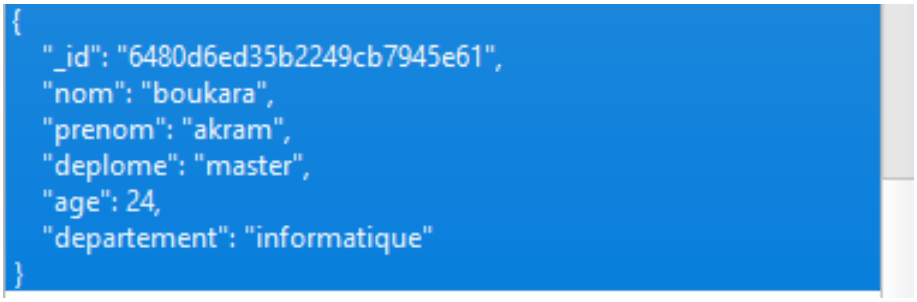

FIGURE 3.4 – Exemple de documents normalisé.

3.4 Présentation de la méthode proposée "RQDB" :

La méthode de contrôle de qualité de de données orientées documents RQDB (Repair Quality DataBase) repose sur les étapes suivantes :

3.4.1 Détection de qualité de données :

On distingue les types de détection de qualité suivants :

- Redondance : En parcourant les documents de la collection, RQDB identifie la redondance en se basant sur la fréquence d'apparition des documents (à l'exception de l'ID du document qui est toujours unique) et affiche le teaux des documents redondants. Si des duplications sont détectées dans la collection, RQDB procède à l'étape de réparation.

- L'incomplétude : RQDB parcourt la liste des documents de la collection, et cherche les documents avec les attributs qui ont des valeurs manquantes et donne le choix au utilisateur de faire gérer ces problème :l'imputation par le mot NULL par defaut, la suppression de clé null carrément depuis son document ou bien laisse le document à l'origine.

- Normalisation : RQDB Normalise la structure des documents en convertissant toutes les clés en chaînes de caractères et en uniformisant les indentations à l'aide de la bibliothèque json.

3.4.2 Réparation de données :

Réparation Documents redondants : La réparation du problème de documents redondant consiste à :

- Choisir une collection parmis certain base de données
- Lancer la réparation du contrôle de qualité
- Récupérer un document sans l'ID
- Parcourir l'ensemble des documents et vérifier s'il existe un document redondant et affiche le pourcentage des documents redondants dans une barre de pourcentage .
- Afficher le nombre totale des documents redondants dans la collecton séléctionée.
- Afficher la liste des documents de la collection selectioné (si l'utilisateur veut).
- C'est à l'utilisateur de décider la suppression ou non des documents.

Réparation de l'incomplétude : La réparation du problème de l'incomplétude consiste aprés la Choisir du collection permet certain base de données et Lancer la réparation du contrôle de qualité à :

- Parcourir l'ensemble des documents et vérifier s'il existe un attribut avec une valeur manquante et affiche le pourcentage des attributs avec des valeurs manquantes dans une barre de pourcentage.

- C'est à l'utilisateur de décider la suppression ou non des attributs comportant des valeurs manquantes ou de remplacer tous les valeurs manquates des attributs par le mot NULL par defaut .

3.5 Conclusion :

Dans ce chapitre, nous avons examiné plusieurs problèmes de qualité des données dans les bases de données MongoDB orientées document. Nous avons analysé en détail chaque problème, y compris la redondance des données et l'incomplétude. Pour résoudre ces problèmes, nous avons proposé une méthode novatrice appelée RQDB (Repair Quality DataBase) qui permet de détecter et de réparer les données de manière efficace. Nous avons également développé un outil pratique, RQDB Plus (Repair Quality DataBase Plus), pour mettre en œuvre notre méthode et évaluer ses performances. Les résultats obtenus ont démontré l'efficacité de notre approche pour améliorer la qualité des données dans les bases de données MongoDB orientées document.

CHAPITRE 4

IMPLÉMENTATION DE L'OUTIL RQDB+

4.1 Introduction :

Dans ce chapitre, nous aborderons l'implémentation de l'outil RQDB Plus (Repair Quality DataBase Plus) afin de valider la méthode RQDB pour le contrôle de qualité des bases de données NoSQL orientées document. Nous commencerons par illustrerons l'utilisation de l'outil RQDB Plus à travers quelques exemples concrets. Ces exemples permettront de mettre en évidence l'efficacité de notre méthode dans la détection et la réparation des problèmes de qualité des données dans les bases de données NoSQL orientées document.

4.2 Environnement de développement :

4.2.1 Caractéristiques de la machine :

La méthode RQDB que nous avons développée est implémentée en utilisant Python 3.10.4. Nous avons choisi ce langage en raison de sa simplicité d'utilisation, de sa capacité à manipuler et à traiter de grandes quantités de données, ainsi que de sa vaste communauté d'utilisateurs. Python offre des fonctionnalités avancées de calcul

et de gestion des données, ce qui en fait un choix approprié pour notre méthode de contrôle de qualité des données.

Notre approche repose sur une intégration directe avec les bases de données MongoDB. Toutes les expériences ont été menées sur une machine HP ayant les caractéristiques suivantes :

- Système d'exploitation : Windows 10 .
- Processeur (CPU) : i3 2.4 GHz*2 .
- RAM : 4 GB .
- Type du système : 64 bits.

4.2.2 Logiciels et langages utilisés :

Python :

Python est un langage de programmation interprété, orienté objet et de haut niveau, apprécié pour sa syntaxe simple [12]. Il offre un support pour les modules et les paquets, ce qui favorise la modularité et la réutilisation du code. Python dispose également d'une vaste collection de bibliothèques, notamment dans les domaines de l'apprentissage automatique et de l'intelligence artificielle. Dans notre étude, nous avons utilisé la version 3.10.4 de Python.

MongoDB :

Le modèle orienté documents de MongoDB offre une solution flexible et évolutive pour répondre aux exigences complexes, quel que soit le niveau de dimensionnement. Sa simplicité et sa facilité d'utilisation en font un choix apprécié par les développeurs [13].

PyQT5 :

Python offre une multitude d'options pour le développement d'applications graphiques, dont PyQt5 est une composante essentielle. PyQt5 est une boîte à outils d'interface graphique multiplateforme, offrant des liens Python pour Qt v5. Cette bibliothèque facilite grandement la création d'applications de bureau interactives grâce à

sa simplicité d'utilisation et à ses outils intégrés [14].

Visuel Studio Code :

C'est un IDE offre des fonctionnalités avancées telles que la saisie de code intelligente, la détection d'erreurs en temps réel et les corrections rapides. Il est utilisé pour développer l'outil QoDB en utilisant le langage de programmation Python [11].

4.3 Description de l'outil RQDB Plus :

Nous avons développé l'outil RQDB Plus, qui permet de contrôler la qualité des données dans une base de données NoSQL orientée documents selon les étapes de notre méthode RQDB présentée dans le chapitre précédent.

La figure suivante présente l'interface principale de RQDB Plus :

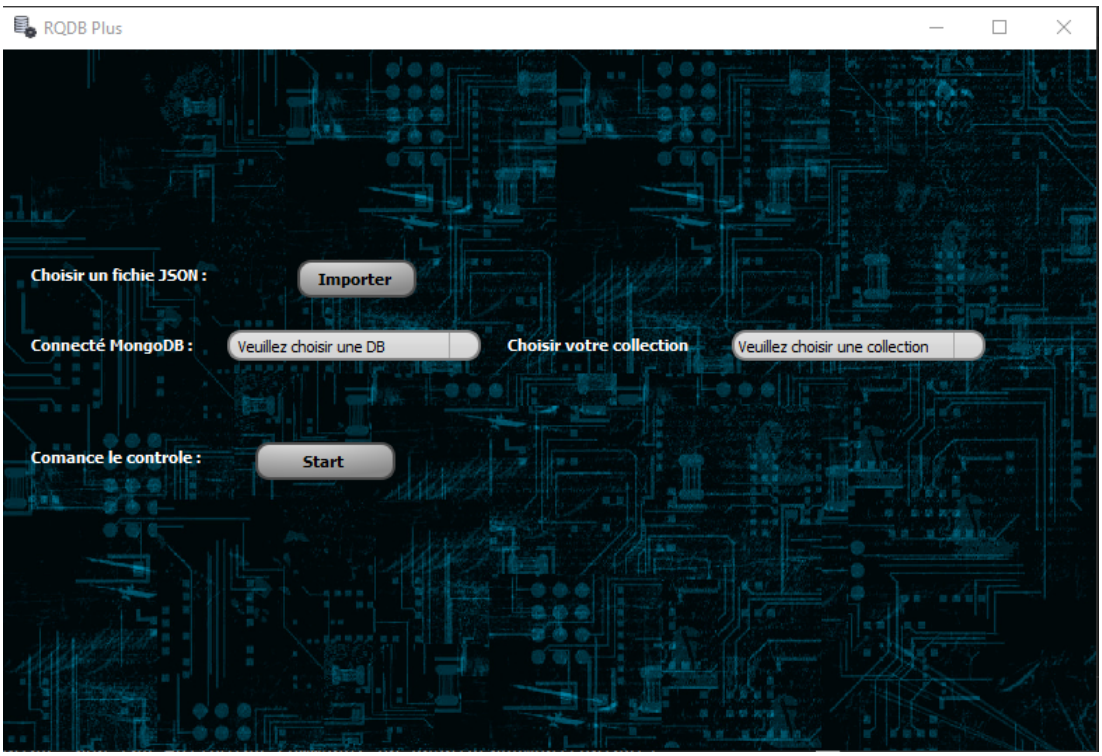

FIGURE 4.1 – L'interface principale de l'outil RQDB Plus.

- Cette illustration représente la séquence d'étapes du processus de validation des données selon la méthode RQDB . Elle débute par l'importation du base de donnés

que l'utilisateur veut la réparer et se termine par réparation du cette base de données. Cette représentation visuelle met en évidence la progression séquentielle du processus de contrôle de qualité des données

Importer une base de données à MongoDB : Cette etape permet d'importer des données JSON sélectionnées à partir d'un fichier dans une base de données MongoDB spécifique et une collection donnée.

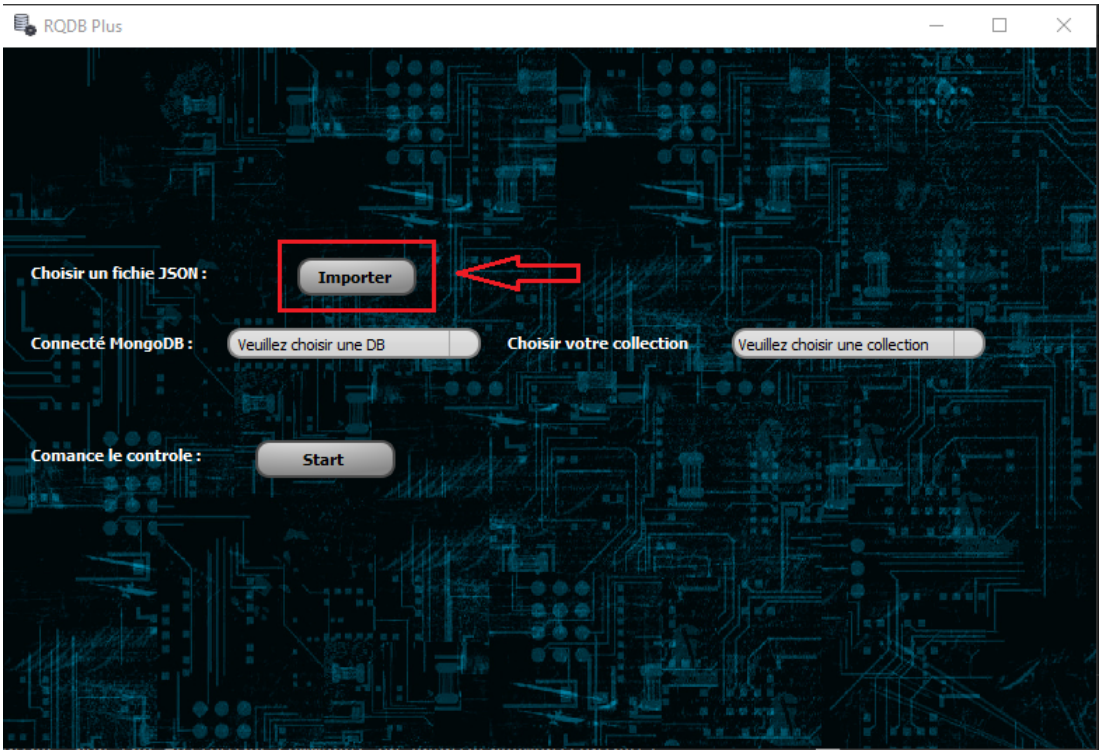

FIGURE 4.2 – Le bouton de l'importation .

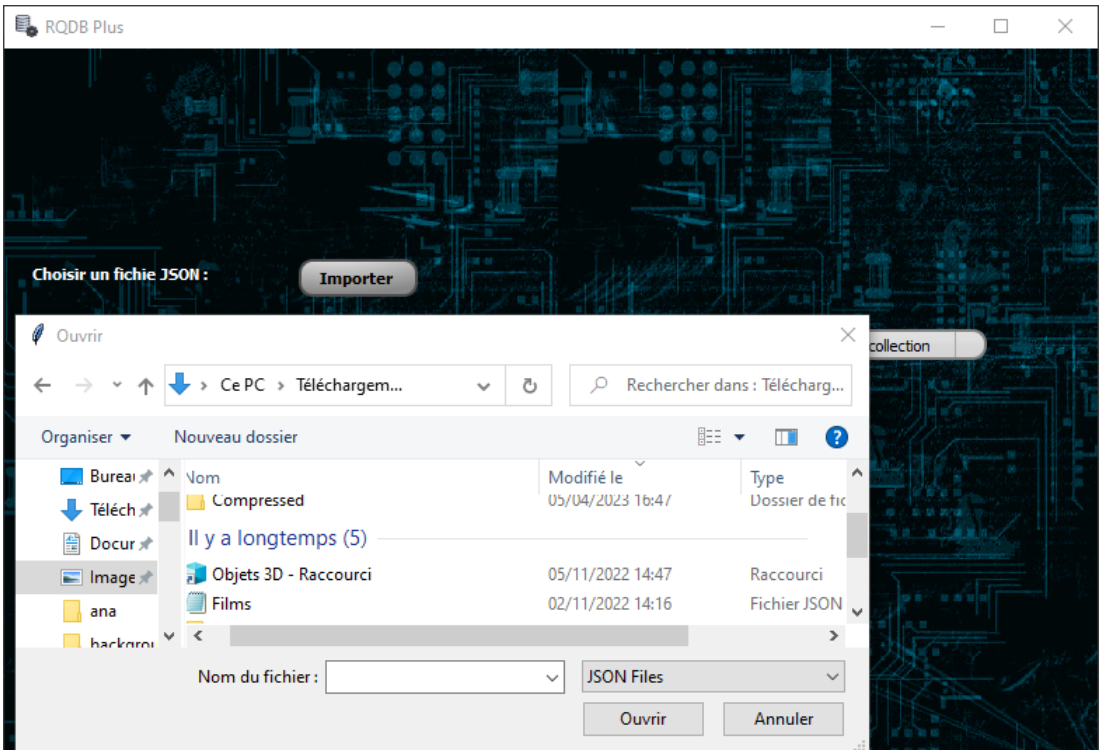

FIGURE 4.3 – Fenetre pour choisir un fichier JSON .

Se connecter à MongoDB : Pour utiliser notre outil de contrôle de la qualité, la première étape sert à se connecter à MongoDB. Ensuite sélectionner la base de données et la collection qu'on veut la contrôler.

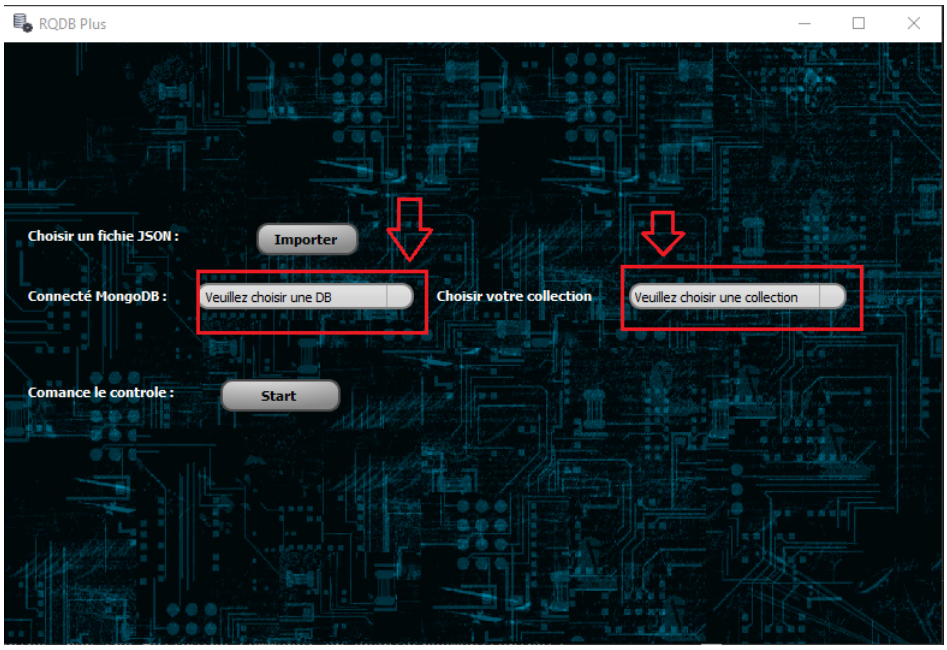

FIGURE 4.4 – Se connecter à MongoDB.

Après avoir se connecter à MongoDB et choisir la base de données et la collection, maintenant le tour de commencer le contrôle de qualité des données.

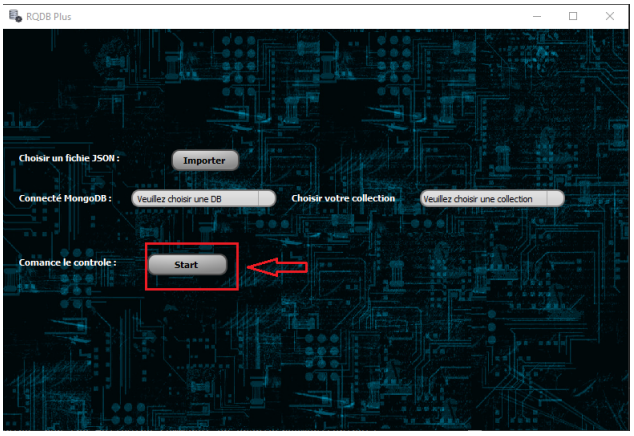

FIGURE 4.5 – Le bouton de démarrage de processus de vérification et réparation de la qualité de données.

Après avoir commencer le contrôle de qualité des données par cliquer sur le bouton start qui effectue des opérations sur la collection séléctionée. Elle normalise les documents,identifie les doublons, calcule les statistiques sur les valeurs manquantes, et affiche les résultats.

De plus, il permet de visualiser les informations clés telles que le nombre de doublons et le pourcentage de valeurs manquantes par rapport tous les documents de la collection.

La figure suivante présente l'interface du résultat aprés le contrôle de qualité :

FIGURE 4.6 – Résultats de controle de la qualité.

Après avoir retourner les résultats de controle, il s'avère nécessaire d'afficher la base de donnée avant la réparation.

La figure suivante présente l'affichage de la base de donnée après le controle et avant la réparation.

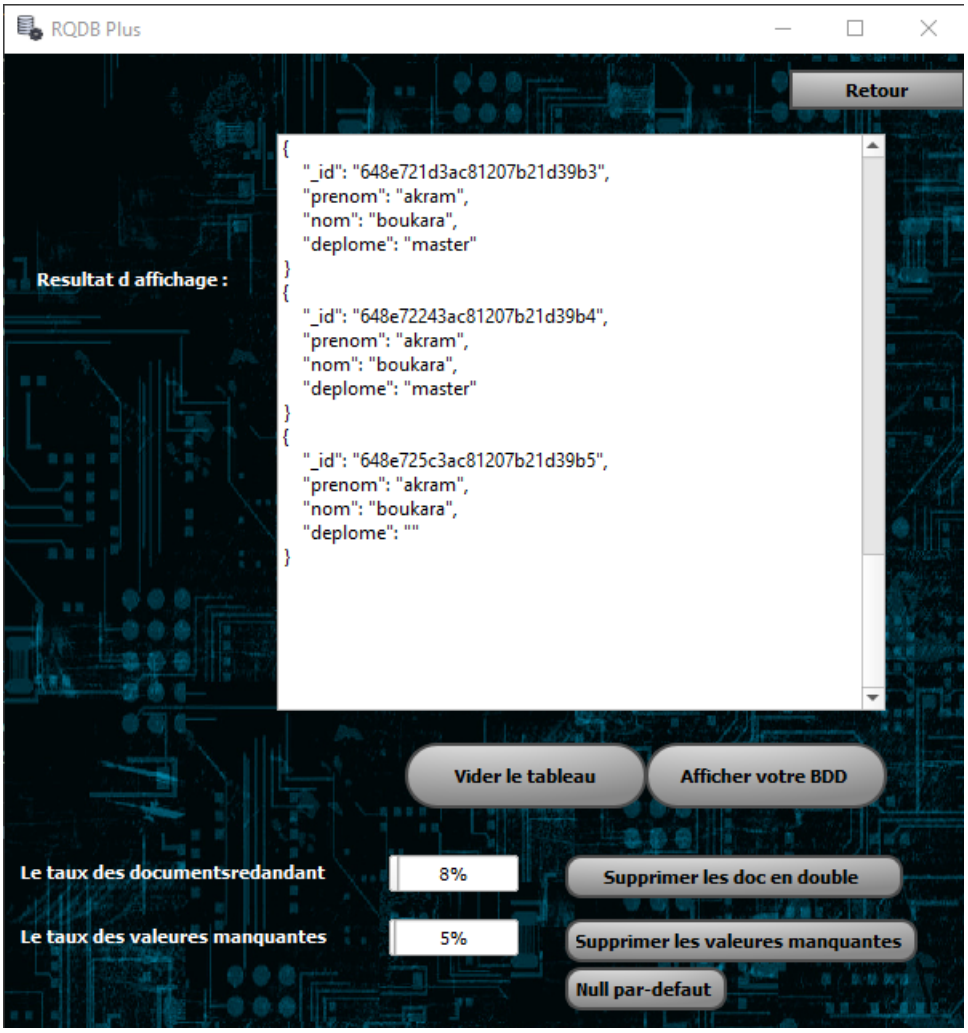

FIGURE 4.7 – L'affichage de la base de donnée avant la réparation.

L'outil RQDB Plus offre le choix à l'utilisateur concernant la réparation du la suppression des documents redondants.

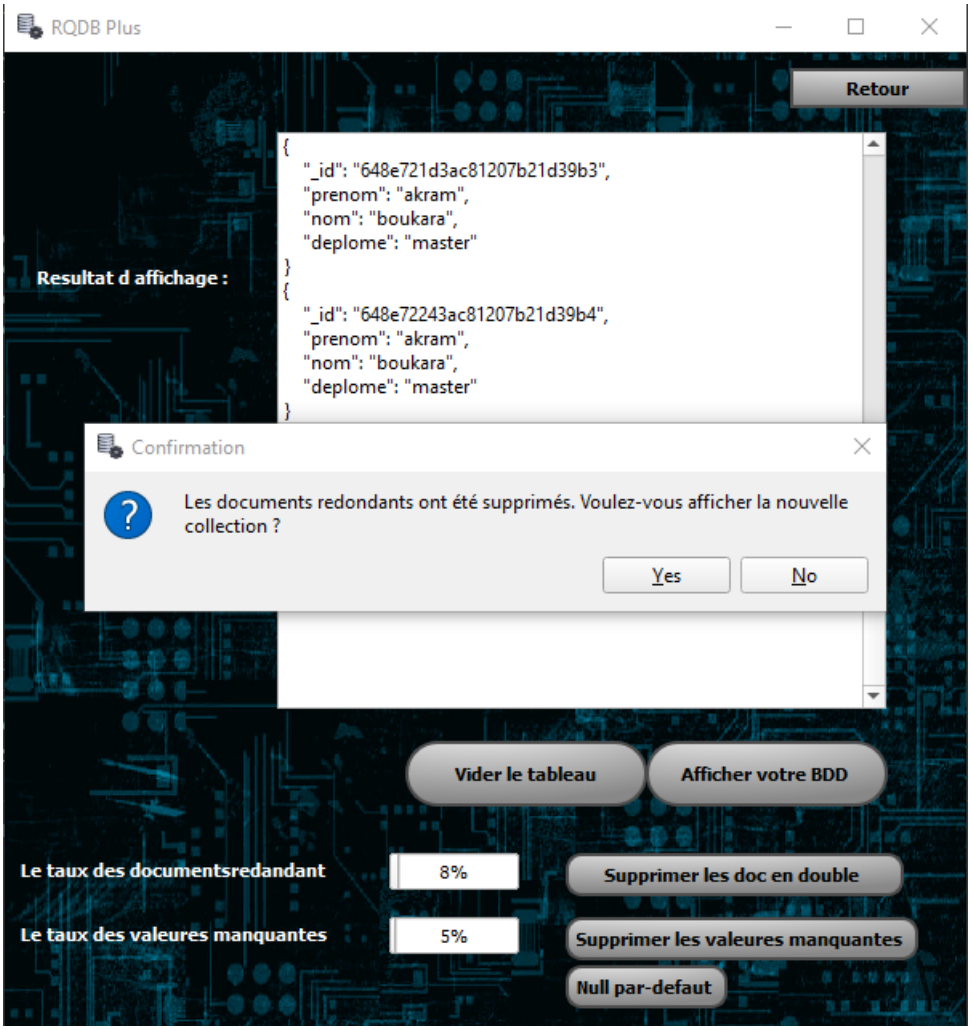

FIGURE 4.8 – La suppression des documents redondants.

La figure suivante présente les résultats après la suppression des documents redondants :

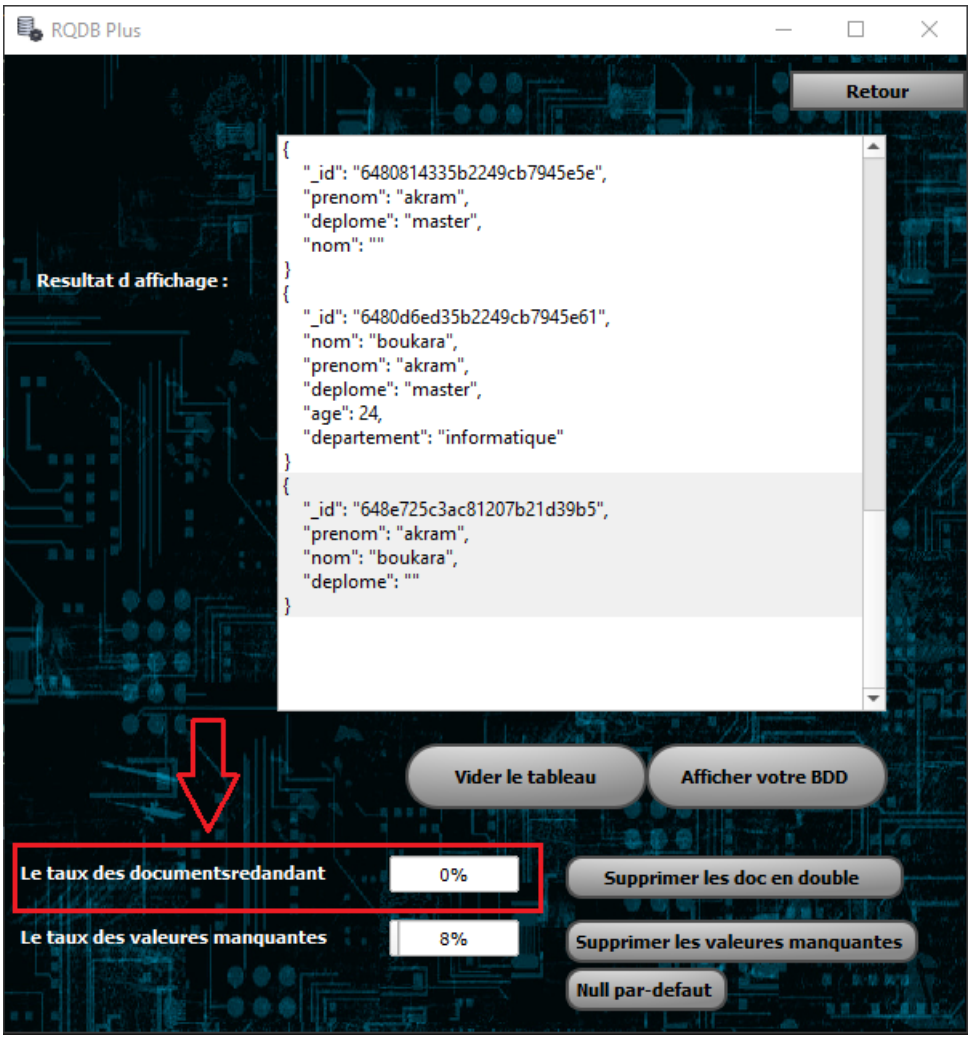

FIGURE 4.9 – Résultats après la suppression des documents redondants.

RQDB Plus offre aussi le choix à l'utilisateur concernant la réparation la suppression des attributs comportant des valeurs manquantes ou de remplacer toutes les valeurs manquantes des attributs par le mot NULL par defaut .

La figure suivante présente les résultats dans le cas où l'utilisateur veut remplacer tous les valeurs manquates des attributs par le mot NULL par defaut :

FIGURE 4.10 – Résultats de réparation par l'imputation par défault de la valeur NULL.

La figure suivante présente les résultats dans le cas où l'utilisateur veut supprimer tous les attributs comportant des valeurs manquantes :

FIGURE 4.11 – Résultats après la suppression des attributs comportant des valeurs manquantes.

4.4 Conclusion :

Afin de résoudre les problèmes de qualité de données dans les bases de données NoSQL orientées documents, nous avons développé l'outil RQDB Plus (Repair Quality DataBase Plus). Cette solution a été spécifiquement développée pour détecter et

réparer quatre problèmes majeurs : la normalisation, la concision (duplication) et l'incomplétude.

Lors de l'utilisation de l'outil RQDB, nous avons pu constater ses performances exceptionnelles en termes de contrôle de la qualité des données orientées documents. Les résultats obtenus ont démontré son efficacité dans la détection et la résolution des problèmes de qualité. Grâce à RQDB, il est possible d'améliorer considérablement la qualité et l'intégrité des données dans les bases de données NoSQL orientées documents .

CONCLUSION GÉNÉRALE

L'objectif principal de ce travail de mémoire est d'assurer le contrôle de qualité des données NoSQL orientées documents en abordant trois dimensions essentielles : la consistance représentationnelle, la complétude et la concision. Ces dimensions de qualité ont un impact direct sur d'autres aspects tels que la cohérence, la facilité d'utilisation et la compréhension des données. Afin de relever ce défi, nous avons développé une approche novatrice appelée RQDB (Repair Quality DataBase) qui vise à détecter et corriger plusieurs problèmes courants liés à la qualité des données NoSQL orientées documents, notamment la normalisation, la duplication des données et l'incomplétude des données. Grâce à notre méthode RQDB, nous contribuons à améliorer la qualité globale des données dans ce contexte spécifique .

Dans le cadre de cette étude, nous avons réalisé le développement de l'outil RQDB Plus (Repair Quality dataBase Plus) afin de valider et évaluer les performances de la méthode RQDB.

À la lumière de ce travail, plusieurs perspectives d'amélioration ont été identifiées pour des travaux futurs. Celles-ci incluent :

- La possibilité de contrôler les types de variables dans les documents d'une collection en utilisant le principe de RQDB.

- L'exploration des techniques d'apprentissage automatique pour gérer les valeurs aberrantes.

- La résolution de la duplication intra-document (duplication de champs au sein du même document) et le traitement sémantique des données textuelles.

Ces axes de développement offrent des opportunités intéressantes pour améliorer davantage la qualité des données dans les bases de données NoSQL orientées documents.

BIBLIOGRAPHIE

[1] https://mariadb.com/fr/database-topics/semi-structured-data/.

[2] https://www.lemondeinformatique.fr/actualites/lire-focus-sur-json-le-format-star-des-echanges-de-donnees-76951.html.

[3] https://www.oracle.com/ca-fr/database/what-is-json/.

[4] ttps://www.tetraedre.com/advanced/xml/.

[5] https://www.etudier.com/dissertations/Avantages-Et-Inconv.

[6] https://www.redhat.com/fr/topics/automation/what-is-yaml.

[7] https://www.csestack.org/advantages-disadvantages-yaml/.

[8] https://www.mongodb.com/basics/bson.

[9] https://www.snowflake.com/trending/avro-vs-parquet/.

[10] https://data-flair.training/blogs/avro-interview-questions/.

[11] BEKADDOUR Abderazak BECHLAGHEM Seyf-Allah, (2017), Etude comparative des performances des bases de données SQL et NoSQL MySQL vs MongoDB. Mémoire Master 2, Département informatique Université Abou Bakr Belkaid– Tlemcen.

[12] https://www.python.org/doc/essays/blurb/. (consulté le 05 Juin 2023).

[13] `https://fr.acervolima.com/python-introduction-a-pyqt5/`. (consulté le 05 Juin 2023).

[14] `https://fr.acervolima.com/python-introduction-a-pyqt5/`. (consulté le 05 Juin 2023).

[15] N.R. ALJOHANI, I.A. ZUALKERNAN et S.M. ALSHOMRANI. *Data Quality Control in Document-Oriented NoSQL Databases*. 2020.

[16] Cappiello C. Francalanci C. Maurino A. BATINI C. *Methodologies for data quality assessment and improvement*. 2009.

[17] Overview of data quality challenges in the context of BIG DATA. IEEE, 2015.

[18] Johnson D. (2022). Intrinsic Dimensions of Data Quality : A Comprehensive Review. Journal of Data Management 15(4)-78-95. BROWN C.

[19] R. DEVILLERS et al. *Towards spatial data quality information analysis tools for experts assessing the fitness for use of spatial data*. 2007.

[20] M. DHANAPAL, S. MUTHUKRISHNAN et K.C. SEKARAN. *A quality control framework for MongoDB databases*. 2020.

[21] Martinez L. (2021). Representational Dimensions of Data Quality : A Comprehensive Study. Data Quality Journal 8(2) 127-142. GARCIA M.

[22] Rodríguez M. Verdugo J. Caballero I. Piattini M. GUALO F. *Data quality certification using ISO/IEC 25012 : Industrial experiences*. 2021.

[23] I. F. ILYAS et X. CHU. *Trends in cleaning relational data : Consistency and deduplication*. 2015.

[24] J.M. JURAN, F.M.J. GRYNA et R.S. BINGHAM. *Quality Control Handbook*. 1974.

[25] F. SIDI et al. *Data quality : A survey of data quality dimensions*. IEEE, 2012.

[26] M.F. SILVA et al. *A Framework for Quality Control of Big Data Stored in MongoDB*. 2018.

[27] Johnson A. (2022). Accessibility SMITH J. et 245-262. Security of DATA : A COMPREHENSIVE ANALYSIS. DATA MANAGEMENT REVIEW 15(3).

[28] Johnson A. (2022). Enhancing Data Quality in Flexible Schema : A Review. Journal of Data Management 15(3)-45-62. SMITH J.

[29] Johnson E. (2022). Contextual Dimensions of Data Quality : A Comprehensive Analysis. International Journal of Data Science SMITH A. et 215-230. ANALYTICS 10(3).

[30] Improving data quality in the linked open data : a SURVEY.

[31] Improving data quality in the linked open data : a SURVEY. 2021.

[32] I. TALEB, M. A. SERHANI et R. DSSOULI. *Big data quality : A survey*. IEEE, 2018.

[33] D. WAHYUDIN et al. *Quality Control for NoSQL-based Big Data : A Systematic Literature Review and Future Research Directions*. 2021.

[34] R. Y. WANG et D. M. STRONG. *Beyond accuracy : What data quality means to data consumers*. 1996.

[35] Y. ZHANG et al. *Scalable and Efficient Data Quality Control for NoSQL Data Stores*. 2019.