MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE ABDELHAMID IBN BADIS - MOSTAGANEM

Faculté des Sciences Exactes et d'Informatique

Département de Mathématiques et informatique

Filière : Informatique

MEMOIRE DE FIN D'ETUDES

Pour l'Obtention du Diplôme de Master en Informatique

Option : **Ingénierie des Systèmes d'Information**

Présenté par :

✓ **HAMMADI Mahmoud**

✓ **MENAD Abderrahmane**

THEME :

Modélisation et Implémentation d'une Base de Données Décisionnelle NoSQL Orientée Graphe Etude de cas Réseau de Distribution des Produits Alimentaires

Soutenu le :

Devant le jury composé de :

M.HABIB ZAHMANI M. MCA Université de Mostaganem Président

M.KHIAT S. MCB Université de Mostaganem Examinateur

M.ABDALLAH BENSALLOUA C. MCB Université de Mostaganem Encadreur

Année Universitaire 2021-2022

Résumé

Les solutions NoSQL ont été créées pour répondre à de nombreux problèmes rencontrés lors de stockage des ensembles de données très volumineuses, non structurées et la nécessité de minimiser les frais de gestion de ces données pour garantir le bon fonctionnement dans les différents entreprises et organisations. Dans ce mémoire nous avons implémenter une base de données NoSQL orientée graphe pour le stockage, la visualisation, l'analyse et la comparaison, permettant aux utilisateurs de générer des rapports et des graphiques. Pour cette fin, nous proposerons le processus d'implémentation d'un entrepôt de données multidimensionnel avec un modèle NoSQL orienté graphe. Des règles de transformation de modèle de données conceptuel multidimensionnel en modèles logiques orientés graphes seront définis et appliqué pour créer le modèle dimensionnel graphe qui sera instancié avec des données relatives aux réseaux de distribution des produits alimentaires en Algérie sur **Neo4j** puis nous explorons, analysons et créons des rapports et des graphiques d'aide à la décision en exploitant les outils de visualisation installés avec **Neo4j**.

Mots-clés: **NoSQL**, Entrepôt de Données Multidimensionnel, Base de Donnée orientée Graphe, **LPG**, Base de Donnée orientée Graphe native, Modèle Dimensionnel de Graphe, **Neo4j**.

Abstract

NoSQL solutions were created to address many problems encountered when storing very large, unstructured data sets and the need to minimize the cost of managing this data to ensure proper functioning in different companies and organizations. In this thesis we have implemented a graph-oriented NoSQL database for storage, visualization, analysis and comparison, allowing users to generate reports and graphs. For this purpose, we will propose the process of implementing a multidimensional data warehouse with a graph-oriented NoSQL model. Rules for transforming multidimensional conceptual data model into graph-oriented logic models will be defined and applied to create the graph-dimensional model that will be instantiated with data relating to the food distribution networks in Algeria on Neo4j then we explore, analyse and create reports and graphs to help the decision by exploiting the visualization tools installed with Neo4j.

Keywords: **NoSQL**, Multidimensional Data Warehouse, Graph Database, **LPG**, Native Graph Database, Graph Dimensional Model, **Neo4j**.

ملخص

تم إنشاء حلول NoSQL لمعالجة العديد من المشكلات التي تمت مواجهتها عند تخزين مجموعات بيانات كبيرة جدًا وغير مهيكلة والحاجة إلى تقليل تكلفة إدارة هذه البيانات لضمان حسن سير العمل في مختلف الشركات والمؤسسات. في هذه المذكرة قمنا بتنفيذ قاعدة بيانات NoSQL رسومية للتخزين والتصور والتحليل والمقارنة ، مما يسمح للمستخدمين بإنشاء تقارير ورسوم بيانية. لهذا الغرض ، سوف نقترح عملية تنفيذ مستودع بيانات متعدد الأبعاد مع نموذج NoSQL رسومي. سيتم تحديد وتطبيق قواعد تحويل نموذج البيانات المفاهيمية متعددة الأبعاد إلى نموذج منطقي رسومي لإنشاء نموذج الأبعاد الرسومي الذي سيتم إنشاء مثيل له باستخدام بيانات متعلقة بشبكة توزيع المواد الغذائية في الجزائر على Neo4j ثم نقوم باستكشاف وتحليل وإنشاء تقارير ورسوم بيانية للمساعدة في اتخاذ القرار من خلال استغلال أدوات العرض المثبتة مع Neo4j.

كلمات مفتاحية: NoSQL، مستودع بيانات متعدد الأبعاد، نموذج متعدد الأبعاد،قاعدة بيانات رسومية، **LPG**، قاعدة بيانات رسومية أصلية،نموذج الأبعاد الرسومي،**Neo4**.

Dédicaces

Je dédie ce travaille à ma famille, mes parents et mes collègues de travail.

HAMMADI Mahmoud

Remerciements

Tout d'abord, nous remercions "Allah" Tout-Puissant de nous avoir accordé le succès dans ce travail.

Nous remercions nos familles, nos collègues de travail pour leurs soutient et leurs encouragements. Nous remercions Mr. ABDALLAH BENSALLOUA Charef pour son encadrement et son assistance durant la réalisation de notre projet de fin d'étude. Nous remercions également les membres du jury pour l'honneur qu'ils nous ont fait d'évaluer ce travail.

Nous tenons à remercier l'ensemble d'enseignants département de mathématiques informatique qui nous beaucoup orienté, conseillé pour l'aboutissement de ce travail et qui fait des efforts pour nous instruire.

En fin, nous formulons nos remerciements à toutes les personnes, qui nous on aidées à l'élaboration de ce mémoire.

Liste des figures

Liste des abréviations

Abréviation	Expression Complète	Page
SQL	Structured Query Language	1
NoSQL	Not Only SQL	1
DbaaS	Database as a Service	2
SGBDR	Système de Gestion de Bases de Données Relationnel	3
ACID	Atomicity, Consistency, Isolation, Durability	3
API	Application Programming Interface	3
ADO	ActiveX Data Object	3
CPU	Central Processing Unit	3
RAM	Random Access Memory	3
SSD	Solid State Drive	3
BASE	Basicaly Availlable, Soft state, Eventual consistency	4
CAP	Consistency, Availability, Partition tolerance	4
Redis	REmote Dictionary Server	5
HBase	Hadoop Database	5
GDHA	Geographically Dispersed High Availability	5
BI	Business Intelligence	10
XML	eXtensible Markup Language	13
JSON	JavaScript Object Notation	13
BSON	Binary JSON	13
R : W	Read Write	15
IoT	Internet of Things	16
E/S	Entrée/Sortie	19
HTTP	Hypertext Transfer Protocol	19
HDFS	Hadoop Distributed File System	19
AQL	ArangoDB Query Language	21
REST	Representational State Transfer	21
OSX	Operating System X	21
MVRB	Multi Value Red Black	22

RDF	Resource Description Framework	27
OLTP	Online Transaction Processing	27
OODBMS	Object Oriented Database Management System	30
LPG	Label Property Graph	30
AL	Adjacency List	31
DEX	DataEXchanger	32
IP	Internet Protocol	38
ATM	Asynchronous Transfer Mode	38
GDPR	General Data Protection Requirements	43
IA	Intelligence Artificiel	47
ONU	Organisation des Nations Unies	53
OAIC	Office Algérien Interprofessionnel des Céréales	58
GIPLAIT	Groupe Industriel des Productions Laitières	59
ORLAC	Office Régional de Lait Centre	59
OROLAIT	Office Régional Ouest de Lait	59
ORELAIT	Office Régional Est de Lait	59
ONAB	Office National des Aliments de Bétail	59
ORAC	Office Régional Avicole Centre	59
ORAVIO	Office Régional Avicole Ouest	59
ORAVIE	Office Régional Avicole Est	59
ONCV	Office National de Commercialisation du Vin	59
ONILEV	Office National Interprofessionnel des Légumes et des Viande	59
CNRC	Centre National du Registre de Commerce	61
UGCAA	Union Générale des Commerçants et Artisans Algériens	61
GDM	Graph Dimentional Model	65

Table des matières

Introduction Générale

En premier temps les données sont représentées par la théorie relationnelle des données proposée par le Docteur E. F. Codd en 1970 apparaissait dans les applications commerciales de base de données relationnelle. Dans le modèle relationnel les données sont gérées par un système de gestion de bases de données relationnelle et stockées sur des tables, ces tables sont manipulées par le langage SQL.

Au fil du temps et en raison de la taille croissante des bases de données des grandes entreprises, l'incapacité de ce modèle à répondre aux besoins de ces entreprises a commencé à apparaître. Les limites pratiques et théoriques, liées à l'usage des bases de données relationnelles face aux nouveaux besoins des systèmes d'information, justifient et motivent la migration des SGBD relationnels vers de nouveaux modèles de bases de données dites NoSQL.

Dans ce travail nous effectuons une étude bibliographique sur les bases de données NoSQL en général et les bases de données NoSQL orientées graphes en particulier puis nous décrirons le cas d'étude et enfin nous proposant et nous implémentons une base de données NoSQL orientée graphe.

Une base de données NoSQL orientée graphe correspond à un système de stockage capable de fournir une adjacence entre éléments voisins : chaque voisin d'une entité est accessible grâce à un pointeur physique. C'est une base de données orientée objet adaptée à l'exploitation des structures de données de type graphe ou dérivée, comme des arbres.

Dans le 1er chapitre nous faisons une étude comparative du SQL et NoSQL ainsi qu'un panorama des bases de données NoSQL le plus utilisés, dans le 2eme chapitre nous décrirons les bases de données NoSQL orientées graphes. Dans le 3eme chapitre nous décrirons le cas d'étude réseau de distribution des produits alimentaires. Dans le 4eme chapitre nous modélisons et nous implémentons une base de données orientée graphe pour le cas d'étude et on termine par une conclusion générale.

Chapitre 1

Présentation des Systèmes NoSQL

1.1 Introduction

Les solutions NoSQL n'ont pas été créées aux mêmes fins que SQL. Alors que SQL est principalement dédié aux données structurées et gestion des transactions, les solutions NoSQL ont été créées pour résoudre les problèmes de stockage d'ensembles de données massifs non structurées, le Figure 1.1 montre l'évolution du volume de données structurées et non structurées [1].

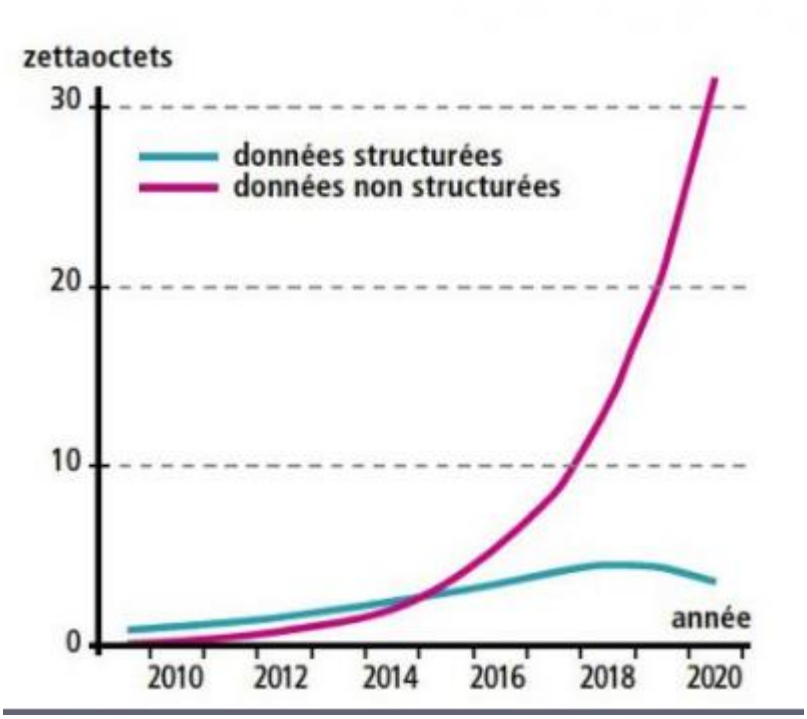

Figure 1.1 – Evolution du volume de données structurées et non structurées [1]

NoSQL a été développé depuis longtemps avant son adoption complète. Le terme NoSQL a été utilisé pour la première fois en 1998 pour les bases de données relationnelles qui n'utilisent pas SQL et utilisé par la suite en 2009 dans une conférence sur les bases de données non relationnelles à San Francisco. En effet, la prolifération de cloud, en particulier les bases de données en tant que service (DbaaS) et le besoin urgent de bases de données rapides, scalable et moins chère pour gérer le Big Data a favorisé la diffusion des bases de données NoSQL [2].

Une autre motivation est la nécessité de stocker les données dans une structure plus simple pour prendre en charge les principes orientés objet tout en évitant le mappage objet-relationnel. Donc NoSQL était une réponse pour remplir les exigences d'applications simples qui ne sont pas complexes. Une autre tendance est la prolifération des technologies Web et Cloud Computing nécessitant une faible surcharge d'administration et haute scalabilité. Il y a aussi le mouvement dans les langages de programmation et les Frameworks de développement qui ont essayé de cacher la complexité du SQL et des bases de données relationnelles pour proposer des technologies plus flexibles et plus pratiques (par exemple : Ruby, Java, API persistante, ADO.net, etc.) [2].

1.2 Etude comparative SQL et NoSQL

1.2.1 Scalabilité, performances et flexibilité

Le SGBDR rencontre trois problèmes principaux en traitant du Big Data et de certaines applications Web : Scalabilité des données, Performances des serveurs uniques, Conception de schéma rigide [2].

Les bases de données SQL sont scalables verticalement. En effet, pour gérer la charge croissante, les utilisateurs doivent augmenter la capacité et les performances à l'intérieur d'un seul serveur. Ceci est réalisé en augmentant par exemple la capacité du CPU, de la RAM, ou le SSD du serveur de base de données dédié. Cependant, le partage de plusieurs tables sur de grands clusters ou grilles est coûteux et complexe [2].

Au contraire, les bases de données NoSQL sont scalables horizontalement. Ainsi, pour gérer de gros volumes de données, les utilisateurs n'ont qu'à ajouter des serveurs à l'infrastructure de base de données NoSQL. Par conséquent, la scalabilité du système est plus facile et moins chère à réaliser en utilisant NoSQL [2].

D'autre part, les bases de données relationnelles nécessitent un modèle de données prédéfinie et données structurées. Ils offrent des fonctionnalités avancées pour gérer, mettre à jour et interroger les données à l'aide de SQL. Il y a divers avantages tels que la préservation de l'intégrité, de la cohérence et la fiabilité des données et des transactions. Il n'y a pas de doute que les bases de données relationnelles assurent plus de fiabilité en comparaison aux bases de données NoSQL. Les bases de données SQL préservent la fiabilité et l'intégrité des données et les transactions en respectant les propriétés ACID [2].

Les principes ACID :

- Atomicité (**A**tomicity) : chaque tâche d'une transaction réussit ou la transaction entière est annulée,
- Cohérence (**C**onsistency) : une transaction conserve un état valide pour la base de données avant et après son achèvement et ne peut pas laisser la base de données dans un état incohérent,
- Isolation (**I**solation) : une transaction non encore validée ne doit pas interférer avec une autre transaction et doit rester isolé,
- Durabilité (**D**urability) : les transactions validées persistent dans la base de données et peuvent être récupérées dans le cas de défaillance de la base de données.[3]

Cependant, il est difficile de garantir les propriétés ACID dans le cas d'énormes ensembles de données en croissance. C'est pourquoi les bases de données NoSQL comptez plutôt sur les principes BASE (Basically Avaliable, Soft state, Eventually consistency).

Les principes BASE :

- Fondamentalement disponible (**B**asicaly **A**vaillable) : la garantie de disponibilité du système en cas de panne,
- Etat mou (**S**oft State) : l'état des données peut changer sans interactions d'application en raison de cohérence éventuelle,
- Cohérence éventuelle (**E**ventually consistency) : le système sera finalement cohérent après la saisie de l'application. Les données seront répliquées sur différents nœuds et atteindront éventuellement un état cohérent. Mais la cohérence n'est pas garantie au niveau de la transaction.[3]

Ainsi, ils offrent une architecture pour gérer non seulement les données structurées mais aussi non structurées et les données semi-structurées. Les utilisateurs peuvent facilement effectuer des pousses de code fréquent et itérations rapides.

Les propriétés ACID et BASE sont dérivés du théorème CAP (Consistency, Availability, Partition tolerance).

Théorème CAP (voir Figure 1.2**) :**

- Système cohérent et disponible (**C**onsistent and **A**vailable) : si l'application nécessite une cohérence élevée et disponibilité sans tolérance de partition, un système CA est un bon choix. La plupart des SGBDR traditionnels sont des systèmes CA. Une base de données orientée graphe telle que Neo4j est également un système CA [3].

- Système cohérent et tolérant aux partitions (**C**onsistent and **P**artition) : si l'application nécessite un cohérence et tolérance de partition, un système CP est un bon choix. Les systèmes CP ne sont pas capables de garantir la disponibilité car le système renvoie une erreur jusqu'à ce que l'état partitionné soit résolu. Redis (K : V), MongoDB (Doc Store) et HBase (Col Oriented) sont des exemples [3].

- Système disponible et tolérant aux partitions (**A**vailable and **P**artition Tolerant System) : si l'application nécessite une haute disponibilité et la tolérance de partition, un système AP est un bon choix. Les systèmes AP ne sont pas en mesure de garantir la cohérence car les écritures/mises à jour peuvent être effectuées de chaque côté de la partition. De tels systèmes fournissent généralement GDHA où les données sont répliquées de manière bidirectionnelle sur deux centres de données et les deux sont dans Configuration active-active, c'est-à-dire que l'application peut écrire/lire vers/depuis l'un ou l'autre centre de données. Riak (K : V), Couchbase (Doc Store) et Cassandra (Col Oriented) en sont des exemples [3].

Les principes BASE sont plus flexibles que ceux d'ACID tandis que les propriétés ACID assurent plus la cohérence et la fiabilité des transactions. Cependant, ces deux qualités sont atteintes comme du coût de la performance et des investissements importants. Ainsi, en fonction du cas d'utilisation et des besoins métiers, les utilisateurs doivent analyser leurs besoins en termes de flexibilité et performance. Ils peuvent choisir soit la base de données relationnelle pour garantir la cohérence via les propriétés ACID ou les bases de données NoSQL lorsque la flexibilité et la performance sont privilégiées pour gérer de grands ensembles de données et gérer plusieurs serveurs dans un cluster, même si la flexibilité signifie moins d'intégrité.

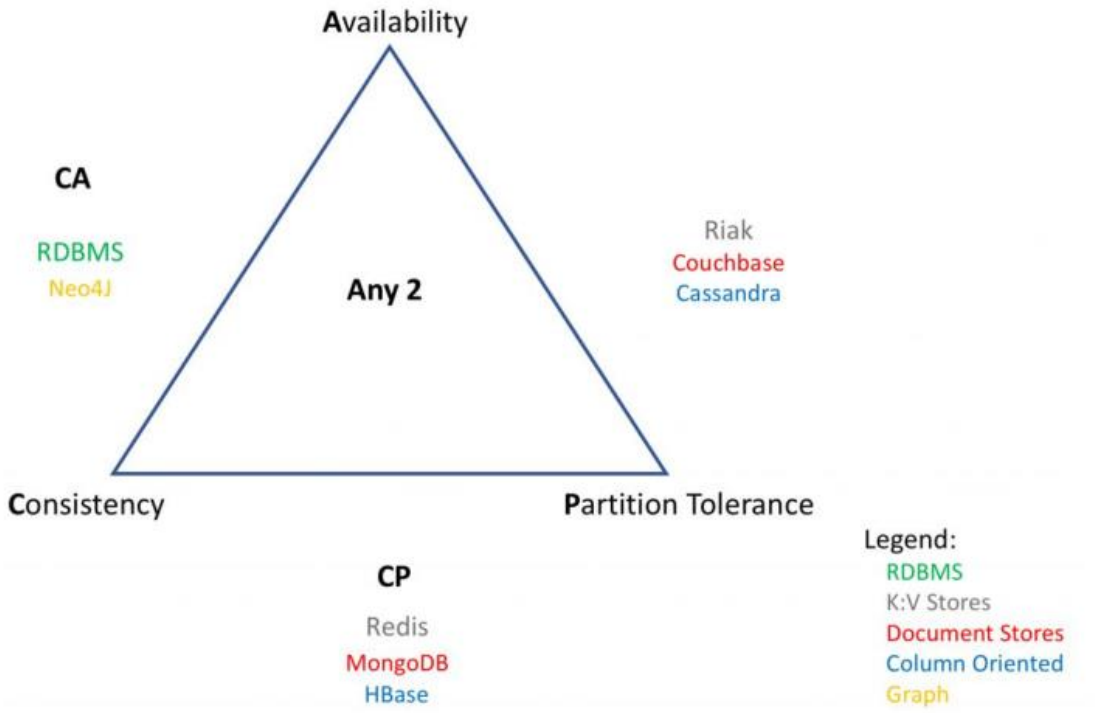

Figure 1.2- Théorème CAP [3]

En plus de tout cela, les bases de données relationnelles manquent d'efficacité lorsqu'il s'agit de Big Data. En effet, les performances des bases de données relationnelles ont tendance à diminuer à mesure que le volume de données augmente surtout lorsqu'il s'agit de données semi-structurées en grand entrepôts. De plus, ils nécessitent des investissements importants lorsqu'il faut augmenter la scalabilité (par exemple, l'ajout de serveurs pour stocker et traiter de grands ensembles de données nécessite l'achat des licences additionnelles). De plus, le besoin croissant de véritable analyse temporelle de grands volumes de données hétérogènes évolutifs ajoute un autre niveau de difficulté. En plus les modèles de stockage des lignes de SGBDR sont moins rapides que les magasins de colonnes (par exemple, le traitement statistique est lent dans le SGBDR). Quelques recherches proposer des mises à niveau du SGBDR pour faire face à ces problèmes. Le SGBDR devrait intégrer le stockage sur baie et être étendu pour inclure des matrices et des bibliothèques mathématiques. Autres, experts soutenir la promesse de bases de données NoSQL et bases de données sans schéma (par exemple, des graphes ou des bases de données orientées objet) [2].

Contrairement au SGBDR, les bases de données NoSQL ont été adaptées et amélioré pour fournir la scalabilité, les performances et la flexibilité nécessaires pour les cas d'utilisation du Big Data. Par exemple, des milliards de données peuvent être injectées par jour dans la colonne-store Hyper table de Zvent, tandis que google est capable de traiter

20 pétaoctets de données stockées dans BigTable via MapReduce. De plus, ils sont basés sur du matériel et des technologies plus abordables que les bases de données relationnelles. NoSQL sont même privilégiés pour certaines applications simples où le stockage et le traitement des données ne nécessitent pas des fonctionnalités avancées du SGBDR ni pour garantir l'intégrité les données comme pour les transactions bancaires. Ainsi, NoSQL permet d'éviter la complexité inutile des bases de données relationnelles. Par exemple, les sites de médias sociaux et les grandes applications Web ne sont pas nécessairement besoin de transactions de confiance et de propriétés ACID (par exemple, mettre à jour le statut Facebook ou les commentaires des Tweets). Zéro données perte, zéro interruption de service ne sont pas cruciales dans ces cas. De plus, la mise en œuvre des propriétés ACID du SGBDR peut être cher par rapport à l'utilité des médias sociaux [2].

Pour toutes les raisons évoquées, les bases de données relationnelles ne sont pas adaptées à l'environnement Cloud. En fait, les SGBDR ont un nombre limité de scalabilité et reposent sur les propriétés ACID. Ainsi, ils ne peuvent prendre en charge de très grands ensembles de données semi-structurés et non structurés. Au contraire, les bases de données NoSQL offrent une meilleure disponibilité, scalabilité, performances et flexibilité. Ils peuvent gérer tous types de données (structurées, semi-structurées et non structurées) [2].

De plus, la gestion du changement est très difficile à gérer dans les bases de données relationnelles. Les utilisateurs doivent définir le schéma de la base des données avant l'injection de données. De plus, tout changement de schéma ou les tables de la base de données doivent être étudiés attentivement. Autre, de tels changements peuvent entraîner une panne de service, réduire les performances ou peut nécessiter un entretien et des investissements supplémentaires pour adapter les modules applicatifs [2].

Au contraire, les bases de données NoSQL permettent un changement facilitant la gestion. En fait, il n'est pas nécessaire de préciser à l'avance le schéma rigide de base de données. Cela donne aux utilisateurs la flexibilité de stocker les données sans schéma prédéfini. De plus, il est possible modifier à tout moment le modèle de données sans affecter le système ou les performances des applications. Ainsi, les utilisateurs doivent choisir le modèle de données et base de données appropriés en fonction de leur cas utilisation [2].

1.2.2 Interrogation et analyse

Les utilisateurs de bases de données relationnelles lancent des requêtes en utilisant les normes de langage de requête structuré (SQL). Cependant, il n'y a pas de norme pour interroger les bases de données NoSQL. En effet, chaque base de données NoSQL à sa façon

unique de gérer, d'extraire et d'interroger des données. Par conséquent, les data scientists sont confrontés au défi de comprendre le langage de requête de chaque base de données NoSQL [2].

D'autre part, les bases de données SQL sont puissantes pour gérer les requêtes complexes via une interface standardisée, alors que les bases de données NoSQL manquent de performances lorsqu'elles traitent des requêtes complexes. Les jointures sont difficiles à réaliser dans les bases de données NoSQL, mais NoSQL est plus adéquat pour gérer les calculs parallèles et les équations mathématiques dans de grands ensembles de données distribués [2].

1.2.3 Sécurité

Alors que les bases de données NoSQL offrent une meilleure scalabilité et flexibilité pour gérer le Big Data, ils ont de nombreux problèmes de sécurité que fournisseurs et chercheurs tentent de résoudre. En fait, la plupart des bases de données NoSQL ne sécurisent pas les communications client/serveur et ne fournissent pas d'authentification ni mécanismes d'audit. Cela compromet la sécurité des données. Pour garantir l'authentification, les utilisateurs doivent généralement ajouter des composants externes à l'infrastructure NoSQL [2].

De plus, si le chiffrement des données structurées est plus facile aux bases de données relationnelles, le chiffrement de très grosses sources de données non structurés est difficile à réaliser. Ainsi, la plupart de ces sources sont stockés au format clair dans NoSQL [2].

1.2.4 Partitionnement

Le partitionnement signifie partitionner de gros volumes de données entre serveurs et nœuds de données virtuels. Les bases de données NoSQL embrassent le partitionnement pour équilibrer la charge et assurer le stockage parallèle et le traitement. Ils offrent la possibilité précieuse d'ajouter ou supprimer les serveurs de la couche de données sans affecter la performance de l'application [2].

Au contraire, les SGBDR n'ont pas été créés à l'origine avec cet objectif. Au lieu de cela, la fonctionnalité de partitionnement a été ajoutée au SGBDR. Les tables sont partitionnées sur plusieurs serveurs. Le partage est basé sur le mappage entre les fragments et les nœuds de données qui contiennent ces fragments. La cartographie peut être soit dynamique ou statique. Un inconvénient du partitionnement est qu'il ne permet pas les jointures entre les fragments [2].

1.2.5 Réplication des données

Les moyens traditionnels de redondance des données se concentrent sur le miroir de données. Ils répliquent les données sur des baies cibles au centre de données ou sur un site distant. Cette méthode consomme beaucoup d'espace de stockage en particulier dans le cas d'ensembles de données volumineux dépassant les pétaoctets. En effet, il s'agit de frais généraux et coûteux pour l'organisation de stocker de gros flux de données ainsi que de gros archives de données par des moyens traditionnels. Pour éviter la perte de données, la plupart des bases de données NoSQL fournissent une réplication automatique des données pour la tolérance aux pannes. Ils répliquent les données sur les serveurs de cluster et même entre les centres de données. Ainsi, en utilisant les technologies Big Data et les bases de données NoSQL, les développeurs n'ont pas à s'inquiéter à la complexité de l'environnement de stockage hétérogène ni aux mécanismes de traitement parallèle. La réplication des données en temps réel permet d'assurer une haute disponibilité continue des données dans les deux environnements hétérogènes et homogènes. La réplication des données en temps réel est cruciale pour effectuer des rapports, une analyse interactive et pour assurer les transactions synchronisées. Il aide à extraire des informations précises, prend en charge la prise de décision rapide et optimise les ressources [2].

Une solution alternative à la réplication des données dans un environnement distribué est l'option basée sur un algorithme error-correcting appelé ErasureCoding qui est associé à la technique de stockage basée sur les objets. Par exemple, un objet de données (comme un document avec ses métadonnées) est divisé en segments. Chaque segment est encodé et découpé en tranches qui sont stockés sur différents serveurs. Ainsi, si certaines tranches ne sont plus accessibles en raison d'une panne de disque, l'organisation peut encore reconstruire les données d'origine. Cette solution réduit le coût, consomme moins de stockage et garantit la tolérance à pannes référentielles. Cependant, il n'est pas encore mûr [2].

1.2.6 Choix entre SQL et NoSQL

- Comprendre l'ajustement des propriétés ACID et BASE.

-Choisir NoSQL si on besoin :

1. Données semi-structurées ou non structurées / schéma flexible
2. Chemins d'accès et modèles de requête prédéfinis limités
3. Pas de requêtes complexes, de procédures stockées ou de vues
4. Transactions à grande vitesse

5. Grand volume de données (dans la plage de téraoctets) nécessitant une évolutivité rapide et bon marché

6. Nécessite un calcul et un stockage distribués

7. Aucun cas d'utilisation d'entrepôt de données, d'analyse ou de BI

- Choisir SQL si on besoin :

1. Données cohérentes/transactions ACID

2. Requêtes dynamiques complexes nécessitant des procédures stockées, ou vues

3. Possibilité de migrer vers une autre base de données sans modification significative de l'existante, les chemins d'accès ou la logique de l'application

4. Cas d'utilisation d'entrepôt de données, d'analyse ou de BI [3].

1.3 Entrepôt de données

1.3.1 Définition d'un entrepôt de données

Un entrepôt de données est une collection de données thématiques, intégrées, non volatiles, historiées et exclusivement destinées aux processus d'aide à la décision [4].

1.3.2 Les caractéristiques

L'Entrepôt de données n'est pas une simple copie des données de production mais il est également organisé et structuré.

- Orienté sujet : Les données des entrepôts sont organisées par sujet plutôt que par application
- Intégrées : Les données provenant des différentes sources sont intégrées avant d'être proposées à l'utilisation. L'intégration (mise en correspondance des formats, par exemple), permet d'avoir une cohérence de l'information.
- Non volatiles : A la différence des données opérationnelles, celles de l'entrepôt ne disparaissent pas et ne changent pas au fil des traitements, au fil du temps (Read-Only). Le rafraîchissement de l'entrepôt, consiste à ajouter de nouvelles données, sans modifier ou perdre celles qui existent.
- Historiées : Les données non volatiles sont aussi horodatées. On peut ainsi visualiser l'évolution dans le temps d'une valeur donnée. Le degré de détail de

l'archivage est bien entendu relatif à la nature des données. Toutes les données ne méritent pas d'être archivées. [4]

1.3.3 Modélisation multidimensionnelle d'un entrepôt de données

La construction d'un entrepôt de données est réalisée à partir de la modélisation multidimensionnelle. Les données de la modélisation multidimensionnelle sont organisées de manière à mettre en évidence le sujet de l'analyse (la table des faits) et les différentes perspectives de l'analyse (les tables des dimensions) qui symbolisent les différentes valeurs de l'activité analysée.

- **Le schéma en étoile**

Ce model tire son nom de sa configuration, en effet il se forme d'un objet central appelé table des faits et est relié à un ensemble d'objets de manière radiale, les tables de dimension.

Un fait est la plus petite information analysable. C'est une information qui contient les données observables (les faits) que l'on possède sur un sujet et que l'on veut étudier, selon divers axes d'analyse (les dimensions). Les "faits" dans un entrepôt de données, sont normalement numériques, puisque d'ordre quantitatif [5]. La table des faits contient les différentes mesures, possède une clé étrangère pour chacune des tables de dimension et elle est dénormalisée.

Une dimension est une "table" qui représente un axe d'analyse selon lequel on veut étudier des données observables (les faits) qui, soumises à une analyse multidimensionnelle, donnent aux utilisateurs des renseignements nécessaires à la prise de décision [5].

- **Schéma en flocon de neige**

Le schéma en flocon est dérivé du schéma en étoile, il se compose d'une relation centrale entourée des différentes tables de dimension qui sont normalisées. Avec ce schéma, les dimensions sont éclatées ou décomposées en sous hiérarchies.

- **Schéma en constellation**

Le schéma en constellation dérive du schéma en étoile, il représente plusieurs relations de faits partageant ou non des dimensions.

- **Schéma en cube**

Dans le modèle multidimensionnel, le cube est l'objet central, lequel est composé des éléments nommés cellules qui contiennent une ou plusieurs mesures. La reconnaissance de la cellule est faite à travers les axes, ou chaque axe correspond à une dimension.

1.4 Types de bases de données NoSQL

Les bases de données NoSQL peuvent être classées en utilisant différentes approches et divers critères. Certains experts les classent selon leur modèle de données et la plupart d'entre eux décrivent quatre grands types de bases de données NoSQL : bases de données orientés clé-valeur, bases de données orientés documents, bases de données orientées colonnes et bases de données orientés graphes. Dans les sections suivantes, nous présentons ces quatre grands types.

1.4.1 Bases de données orientés clé-valeur

Ce modèle est implémenté à l'aide d'une table de hachage où il y a une clé unique et un pointeur vers un élément de données particulier en créant une paire clé-valeur (voir Figure 1.3). La table de hachage convient aux recherches pour des valeurs simples ou complexes dans des ensembles de données extrêmement volumineux [2].

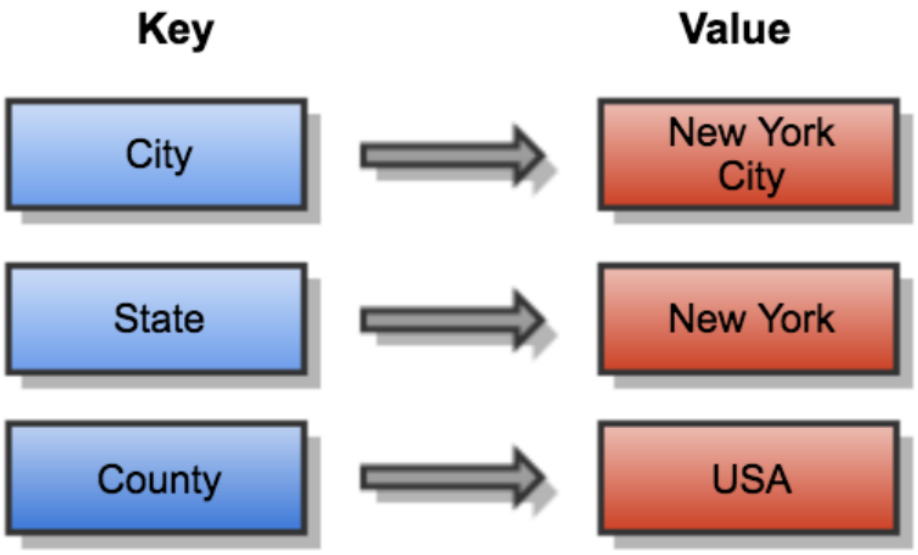

Figure 1.3 -Modèle orienté clé-valeur [3]

Les bases de données orientés clé-valeur peuvent gérer un très grand nombre d'enregistrements. Ils peuvent prendre en charge des volumes élevés de changements d'état

par seconde avec des millions d'utilisateurs simultanés via traitement et stockage distribués. Les bases de données orientés clé-valeur s'appuient sur leur redondance pour faire face à la perte de nœuds de stockage et pour protéger les applications. Ils sont très utiles à la fois pour stocker les résultats d'un algorithme analytique (tels que le nombre de phrases parmi les nombres de documents) et de produire ces résultats via rapports [2].

Cependant, les bases de données orienté clé-valeur héritent d'un inconvénient de bases de données NoSQL. Ils ne fournissent aucune sorte de capacités de la base de données traditionnel. Ainsi, pour assurer l'atomicité des transactions ou la cohérence de plusieurs transactions parallèles, les utilisateurs doivent s'appuyer plutôt sur l'application elle-même [2].

Un autre inconvénient est que les utilisateurs ne peuvent pas accéder aux données par valeur. En effet, il est impossible d'interroger un magasin de données clé-valeur pour extraire tous les enregistrements qui contiennent un ensemble particulier de valeurs. Le seul moyen d'interroger une base de données orienté clé-valeur est en spécifiant une requête par clé ou par plage de clés [2].

Exemples de cas d'utilisation :

- Gestion des sessions
- Profils, préférences & configurations
- Mise en cache des données
- Stockage de fichiers multimédias ou d'objets volumineux [3].

Nous choisissons la base de données orientée clés-valeurs si on besoin:

1. Schéma simple
2. Lecture/écriture à grande vitesse sans mises à jour fréquentes
3. Hautes performances et évolutivité
4. Pas de requêtes complexes impliquant plusieurs clés ou jointures [3].

1.4.2 Bases de données orientés documents

Les bases de données orientés documents ont été conçues pour gérer le stockage et la gestion des documents à grande échelle. Ce type de base de données attribue une valeur clé à chaque document. Les documents peuvent contenir plusieurs paires clé-valeur, ou paires clé-tableau, ou même les documents imbriqués (voir Figure 1.4). Les documents sont encodés dans un standard de format d'échange de données tel que XML, JSON ou BSON [2]

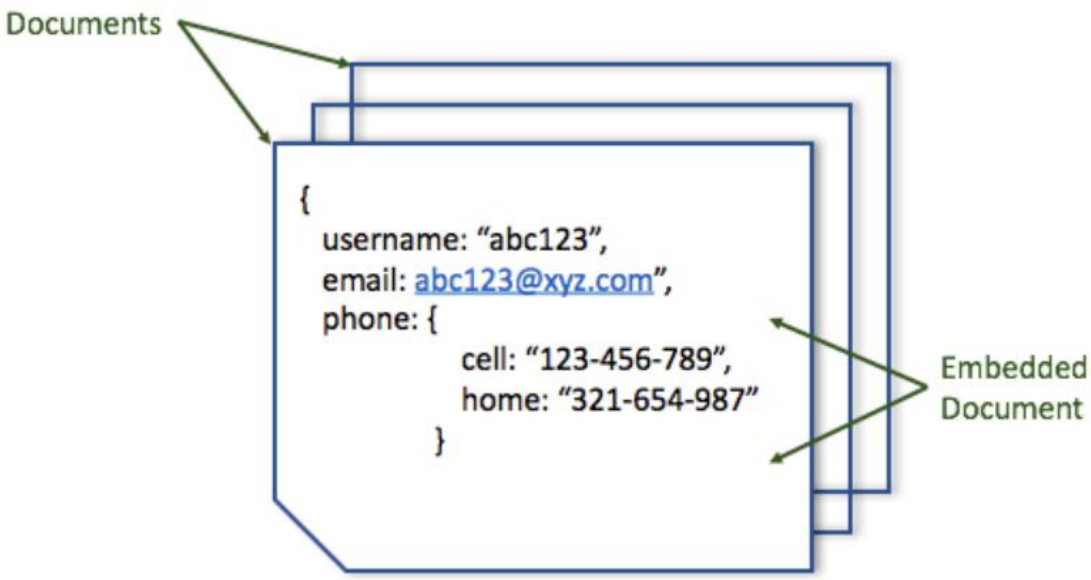

Figure 1.4-Modèle orienté documents [3]

Les bases de données orientés documents sont reconnues comme un outil puissant, flexible et agile pour stocker les BigData [2].

Les bases de données orientés documents sont différentes des magasins de clés-valeurs. En effet, tandis que les magasins de clés-valeurs permettent de rechercher des données uniquement par la valeur de la clé, les bases de données documentaires permettent aux utilisateurs de rechercher des données en fonction du contenu des documents. Ils peuvent interroger soit par des clés, des valeurs ou des exemples. En effet, les documents codés contiennent des objets de métadonnées, il est donc possible d'interroger des données en exemple. Cela donne aux bases de données de documents une grande flexibilité requis par certains cas d'utilisation. Pour lancer des requêtes, les utilisateurs peuvent soit s'appuyer sur une API de programmation ou un langage de requête [2].

Contrairement aux magasins de clés-valeurs simples, la valeur de la colonne dans les bases de données orienté documents contient des données semi-structurée et en particulier les paires d'attribut nom/valeur. Par ailleurs, les bases de données orientés documents prennent en charge un schéma flexible. En effet, ils permettent de stocker des centaines d'attributs dans une seule colonne d'un schéma de documents. Ainsi, les rangées peuvent recevoir divers montants et types d'attributs [2].

Exemples de cas d'utilisation :

- Systèmes de gestion de contenu
- Sites de commerce électronique
- Applications middleware qui utilisent JSON.[3]

On choisir la base de données orientée documents si on besoin :

1. Schéma flexible avec requêtes complexes
2. Formats de données JSON/BSON ou XML
3. Tirez parti d'index complexes (multi-clés, géospatiales, recherche plein texte, etc.)
4. Haute performance et rapport R : W équilibré.[3]

1.4.3 Bases de données orientées colonnes

Bases de données orientée colonnes ou à colonnes étendues (également appelées magasins d'enregistrement extensible) représentent une extension de l'architecture clé-valeur avec des colonnes (voir Figure 1.5). Ils ont été conçus pour traiter des données sur un pool d'infrastructures. Les bases de données à colonnes étendues reposent sur une approche hybride qui s'appuie sur les caractéristiques déclaratives des bases de données et divers schémas des magasins clé-valeur [2].

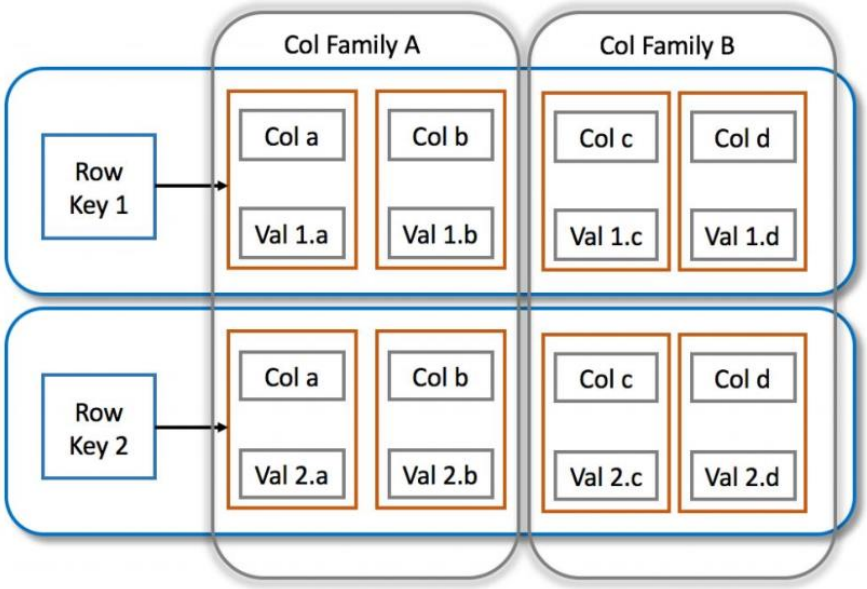

Figure 1.5 -Modèle orienté colonnes [3]

Exemples de cas d'utilisation :

- Données de séries chronologiques
- Applications IoT
- Journalisation et autres applications lourdes en écriture. [3]

On choisir la base de données orientée colonnes si on besoin :

1. Volume élevé de données
2. Vitesses d'écriture extrêmes avec des lectures relativement moins rapides
3. Extractions de données par colonnes à l'aide de clés de ligne
4. Pas de modèles de requête ad-hoc, d'indices complexes ou de niveau élevé d'agrégations.[3]

1.4.4 Bases de données orientées graphes

Les bases de données relationnelles et les bases de données NoSQL comme les bases de données orientées clés/valeurs ne sont pas efficaces lorsqu'ils traitent des données hautement connectées. Ils manquent des performances et de la flexibilité nécessaires pour traiter et interroger plusieurs relations à l'intérieur de grands ensembles de données [2].

Même si ce paradigme MapReduce associé à Hadoop framework offre une scalabilité, une tolérance aux pannes et un outil de programmation pour de grands ensembles de données, il a été prouvé que certaines bases de données NoSQL comme les bases orientées clés-valeurs ne sont pas toujours adaptées pour les données connectées et les très grands graphes. Au contraire, les bases de données orientées graphes sont adaptées pour stocker non seulement des informations sur les objets mais aussi toutes les relations qui existent entre eux (voir Figure 1.6). Ils reposent sur un modèle de graphe sans schéma afin de modéliser et représenter facilement les données connectées. Telle modèle comprend des sommets (par exemple, des objets ou des éléments représentés par nœuds) et des arêtes pour représenter les connexions entre les données [2].

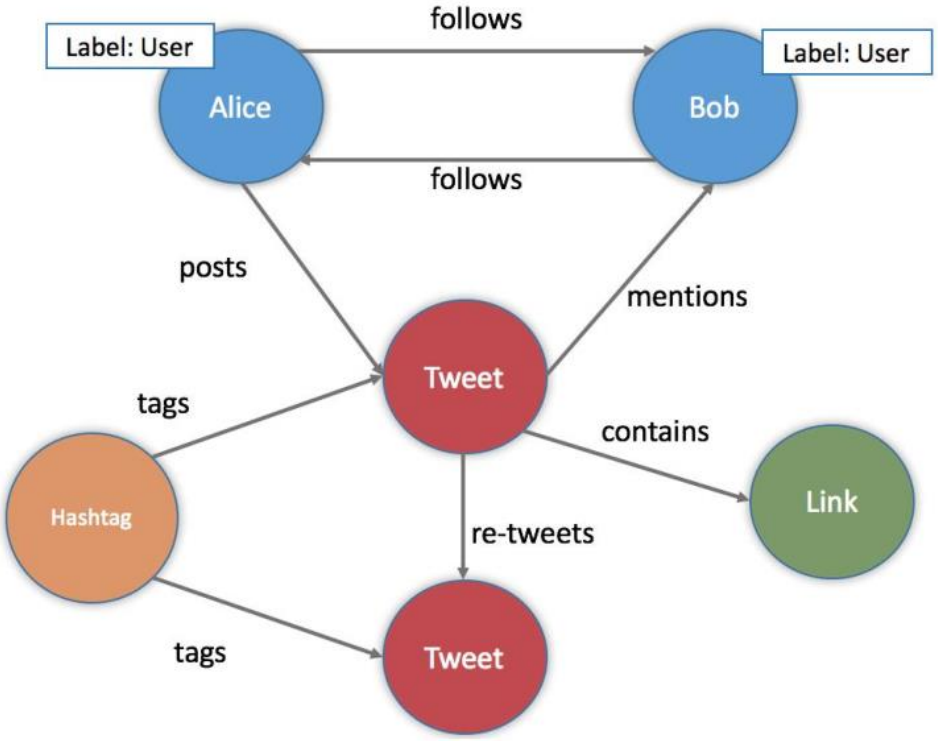

Figure 1.6 - Modèle orienté graphes [3]

Pour illustrer cela, un graphe peut se référer à un réseau professionnel comme dans Viadeo. Dans ce cas, les sommets représentent des professionnels tandis que les bords dirigés représentent des liens et relations entre ces professionnels. Chaque sommet est également initialisé avec une valeur. Il convient de mentionner que même si les bases de données orientées graphes enregistrent les relations, elles n'ont rien à voir avec des bases de données relationnelles [2].

Ils sont utiles pour stocker, accéder et analyser la force et la nature des relations entre deux ou plusieurs éléments (par exemple, à quel point la relation entre deux personnes est-elle étroite ?). Répondre à de telles questions permet de formuler de précieuses recommandations dans de nombreuses industries [2].

Les bases de données orientées graphes offrent pour de nombreux cas d'utilisation une performance améliorée (ils assurent une latence plus faible par rapport aux traitement par lots des agrégats), modèle de données flexible (moyen facile d’exprimer des relations et enrichir le graphe sous forme de données et les besoins métiers se précisent) et l'agilité (capacité à faire évoluer les applications de manière contrôlée alignée avec Agil et pratiques de développement de logiciels pilotés par les tests) . Contrairement à la plupart des classes de bases de données

NoSQL, les bases de données orientées graphes ne sont pas les meilleures solutions pour mettre à jour des ensembles de données ou pour de très grands volumes de données [2].

Exemples de cas d'utilisation :

- Services basés sur la localisation et la navigation
- Infrastructure réseau et informatique
- Détection de fraude
- Gestion des métadonnées. [3]

On choisir la base de données orientée graphes si on besoin :

1. Applications nécessitant une traversée entre les points de données
2. Possibilité de stocker les propriétés de chaque point de données ainsi que la relation entre eux
3. Requêtes complexes pour déterminer les relations entre les points de données
4. Besoin de détecter des modèles entre les points de données. [3]

1.5 Panorama des logiciels utilisés

1.5.1 Logiciels orientés clé-valeur

Redis est plus adapté aux applications critiques en temps car il repose sur un ensemble de données en mémoire pour des réponses rapides. **Redis** est approprié pour traiter rapidement le changement des données telles que la collecte de données en temps réel à partir de capteurs et les communications en temps réel. **Riak** est conçu pour les environnements hautement distribués comme le cloud. Il garantit une haute tolérance aux pannes mais avec moins de performances que **Redis** [2].

Voldemort convient aux ensembles de données très volumineux tels que les données géologiques et métadonnées des cartes. En effet, il peut gérer le stockage d'énormes volumes sans grand impact sur les performances. Des expériences ont montré que **Redis** s'adapte à l'augmentation des données définit les volumes mais ne s'adapte pas à l'augmentation du nombre de nœuds. Au contraire, **Voldemort** évolue lorsque le nombre des nœuds augmente mais ne s'adapte pas à la taille croissante d'ensembles de données. **Redis** assure une meilleure disponibilité des données par rapport à **Voldemort**. Cependant, les deux ont montré une réduction de disponibilité lorsqu'il s'agit de traiter des ensembles de données très volumineux.

Ils ont également été montrés que l'ajout de nœuds au système **Voldemort** contribue à améliorer sa disponibilité [2].

1.5.2 Logiciels orientés documents

CouchDB et **MongoDB** sont des bases de données orientées documents open source. Pour les deux, les données sont stockées dans des documents avec des enregistrements autonomes et aucune relation intrinsèque [2].

CouchDB stocke les données sur le disque en ajoutant uniquement des fichiers tandis que **MongoDB** stocke des données sur le moteur de stockage mappé en mémoire. Il utilise les fichiers mappés mémoire pour toutes les E/S de disque. En tant que format d'échange, **CouchDB** propose une API HTTP pour l'accès aux données et l'administration. **MongoDB** fournit à la place un protocole de fil à base de socket avec BSON [2].

Pour la tolérance aux pannes, **CouchDB** prend en charge à la fois la réplication maître/maître et maître/esclave. La réplication peut être affinée via des filtres de réplication. Au contraire, **MongoDB** gère la réplication en utilisant une forme de réplication maître/esclave asynchrone appelés répliques ensembles [2].

Pour conclure, les deux ont de nombreuses caractéristiques communes telles que réplication pour la tolérance aux pannes et le système de fichiers à mémoire volatile pour le stockage des données. Les deux reposent sur le paradigme MapReduce pour le traitement des données et ont également un bon soutien communautaire. Cependant, **CouchDB** n'est pas adapté aux données extrêmement modifiées. En effet, alors que **CouchDB** nécessite de configurer des requêtes, **MongoDB** est adapté aux requêtes dynamiques et garantit une meilleure performance sur les grandes bases de données [2].

Terrastore est une base documentaire distribuée sous la licence Apache 2.0 qui est assez intéressante dans la mesure où celle-ci est extensible (développement en Java) via un système d'événements. Le langage de requête est lui aussi extensible par ce même biais.

1.5.3 Logiciels orientés colonnes

HBase est un système de gestion de base de données NoSQL conçu pour fonctionner sur le HDFS. C'est un projet open source qui convient à la gestion de grands ensembles de données divers. Il est basé sur le modèle des données clé/valeur orientées colonne. **HBase** est conçu pour prendre en charge des taux de mise à jour de table élevés et pour évoluer horizontalement dans des clusters distribués. **HBase** fournit un hébergement structuré flexible

pour de très grandes tables dans un format de type BigTable. **HBase** fournit de nombreuses fonctionnalités telles que les requêtes temps réel, recherche en langage naturel, accès cohérent aux sources Big Data, scalabilité linéaire et modulaire, automatique et le partitionnement configurable des tables. C'est une base de données non relationnelle populaire qui est incluse dans de nombreuses solutions de BigData et des sites Web axés sur les données [2].

Cassandra est également une base de données NoSQL populaire pour les très grands ensembles de données. Il s'agit d'une base de données clé-valeur qui utilise le stockage orienté colonnes, partitionnement par plages de clés et stockage redondant. Il offre de la scalabilité, des performances de lecture/écriture, ainsi que de la résilience contre les nœuds « chauds » et les défaillances de nœuds. **Cassandra** permet de configurer les paramètres pour ajuster les préférences de compromis entre cohérence et disponibilité [2].

Apache Accumulo est une solution de stockage de colonnes distribué qui offre une scalabilité et des performances élevées. **Apache Accumulo** est basé sur la conception BigTable de Google et il est construit sur le sommet de Hadoop, Zookeeper et Thrift. Il permet un accès contrôle au niveau de la cellule sur la BigTable. Il permet également de modifier des paires clé/valeur à divers points de processus de gestion des données. Ceci est assuré par une programmation côté serveur [2].

BigTable est un système de stockage distribué pour la gestion des données structurées conçues pour s'adapter à une très grande taille : des pétaoctets de données sur des milliers de produits les serveurs. De nombreux projets chez Google stockent les données dans BigTable, y compris l'indexation Web, Google Earth et Google Finance.[6]

Amazon SimpleDBest un service Web permettant d'exécuter des requêtes sur des données structurées en temps réel. Ce service fonctionne en étroite collaboration avec Amazon Simple Storage Service (Amazon S3) et Amazon Elastic Compute Cloud (Amazon EC2), offrant collectivement la possibilité de stocker, traiter et interroger des ensembles de données dans le cloud. Ces services sont conçus pour rendre l'informatique à l'échelle du Web plus facile et plus rentable pour les développeurs.[7]

1.5.4 Logiciels orientés graphes

Neo4j est une base de données open source qui stocke des données connectées dans un format graphique plutôt que dans des tableaux (qui sont plus approprié pour les données agrégées). En effet, les données sont stockées dans de nœuds connectés les uns aux autres par des relations définies. Les deux, les nœuds et les relations ont leurs propriétés. Il est écrit

entièrement en Java et supporté par la Neotechnology. C'est un moteur de persistance Java basé sur disque et entièrement transactionnel avec quelques petits jars. Il offre une grande scalabilité permettant d'ajouter jusqu'à plusieurs milliards de données, haute disponibilité même si les données sont réparties sur de nombreuses machines, requêtes rapides et identification de chemin rapide à travers son cadre de traversée. De plus, les analystes de données peuvent utiliser son langage de requête lisible par l'homme adapté pour les modèles de graphes. **Neo4j** offre également une solution pratique et accès simple (par interface Rest ou une API Java orientée objet). Cela fournit également, comme les bases de données relationnelles, des propriétés ACID complètes pour des transactions fiables [2].

Neo4jSpatial est une bibliothèque des utilitaires pour Neo4j qui permet d'ajouter des index spatiaux aux données déjà localisées. Il est conçu pour faciliter les opérations spatiales sur les données (c'est-à-dire pour rechercher des données spécifiques par région ou dans un périmètre défini) [2].

ArangoDB est une base de données NoSQL distribuée, polyvalente et open source. Il prend en charge plusieurs modèles de données y compris les documents, les graphes et les clés-valeurs. Il convient pour des applications nécessitant un gain de place, des performances élevées avec des outils d'interrogation pratiques. En effet, il permet d'utiliser un langage SQL-like de requête ainsi que les extensions JavaScript et Ruby. En effet, AQL est conçu pour prendre en charge des requêtes complexes en particulier sur les modèles de données **ArangoDB**. Le stockage et la récupération des données sont basés sur des collections. AQL est considéré comme un langage déclaratif centré sur les résultats au lieu de la façon dont les résultats devraient être produits. Par ailleurs, AQL est indépendant du langage de programmation des clients. Ainsi, tous les clients utilisent le même langage et la même syntaxe. De plus, il offre l'option REST pour interroger des documents. Il permet un partitionnement vertical et horizontal (pour ajouter plus de puissance de calcul et pour partager des données sur de nombreux serveurs). Il fonctionne sur différentes plateformes comme Linux, Windows, OSX et même Raspberry Pi. Il est disponible sous licence Apache 2 [2].

OrientDB est un système de gestion de base de données NoSQL open source publiée sous la licence Apache 2. Il est écrit en Java pour qu'il puisse fonctionner sur Linux, Windows et n'importe quel système qui prend en charge Java. Il offre la flexibilité des documents ainsi qu'une bonne performance pour gérer les graphes distribués. En effet, **OrientDB** est une base de données documentaire qui se compose de ODocuments (possibilité d'ajouter et de

supprimer dynamiquement les propriétés). En même temps, il permet de gérer les relations comme dans les bases de données orientée graphes avec des connexions directes entre les enregistrements. Par conséquent, il émule la propriété de contiguïté sans index de documents. Il prend en charge plusieurs modes, y compris sans schéma, schéma-full et schéma-mixed. En outre, **OrientDB** prend en charge SQL comme langage de requête avec la possibilité de gérer les graphes de documents connectés. Il peut gérer les relations sans jointure SQL. Pour des insertions et des requêtes rapides, **OrientDB** est basé sur un nouvel algorithme d'indexation appelé MVRB-Tree. **OrientDB** prend en charge d'autres fonctionnalités telles qu'ACID pour des transactions fiables et un profilage de sécurité via les rôles et les utilisateurs [2].

Neo4j et **OrientDB** sont dédiés à prendre en charge le stockage de grands ensembles de données basé sur un modèle de graphes. Cela garantit la scalabilité et les performances lors de l'insertion ou de l'interrogation de données. La principale différence entre ces deux bases de données NoSQL réside dans le stockage central. En effet, alors qu'**OrientDB** est basé essentiellement sur les documents comme stockage principal (en plus d'une couche graphique qui prend en charge les graphes), **Neo4j** est basé sur les graphes comme stockage de base. Selon certaines expériences, les deux **Neo4j** et **OrientDB** démontrent des performances comparables lorsqu'il s'agit de petits graphes. Cependant, **Neo4j** semble être plus efficace qu'**OrientDB** lors de la gestion du stockage et des requêtes sur de grands graphes [2].

1.6 Conclusion

Dans ce chapitre nous avons présenté les bases de données NoSQL on les compare avec les systèmes SQL selon plusieurs critères comme la scalabilité, la flexibilité, la sécurité, le partitionnement et la duplication, nous avons également décrit les concepts d'entrepôt de données et la modélisation multidimensionnelle ensuite nous avons décrit les grands types de ces bases on les compare et les logiciels les plus utilisés pour leur gestion. Dans le prochain chapitre nous focalisons l'étude sur un type des bases de données NoSQL à savoir les bases orientées graphe.

Chapitre 2

Bases de données NoSQL orientées graphes

2.1 Introduction

Les bases de données orientées graphes sont adaptées pour stocker non seulement des informations sur les objets mais aussi toutes les relations qui existent entre eux (voir Figure 1.6). Ils reposent sur un modèle de graphe sans schéma afin de modéliser et représenter facilement les données connectées. Telle modèle comprend des sommets (par exemple, des objets ou des éléments représentés par nœuds) et des arêtes pour représenter les connexions entre les données.

2.2 Caractérisation d'une base de données NoSQL orientée graphe

Deux propriétés des bases de données orientées graphes sont utiles à comprendre lors de l'étude de technologies de base de données orientée graphe [8] :

2.2.1 Le stockage sous-jacent

Certaines bases de données de graphes utilisent un stockage de graphes natif, qui est optimisé et conçu pour stocker et gérer des graphes. Toutes les technologies de base de données orientée graphes n'utilisent pas nativement le stockage graphique. Certains sérialisent les données du graphe dans une base de données relationnelle, base de données orientée objet ou d'autres types de magasins NoSQL.

2.2.2 Le moteur de traitement

Certaines définitions de bases de données de graphes nécessitent une adjacence sans index, ce qui signifie que les nœuds connectés pointent physiquement les uns vers les autres dans la base de données. Ici, toute base de données qui, du point de vue de l'utilisateur se comporte comme une base de données orientée graphes est qualifiée de base de données orientée graphes. Nous reconnaissons, cependant, les avantages de performance significatifs de l'adjacence sans index, et donc utiliser le terme traitement de graphe natif en référence aux bases de données orientées graphes qui tirer parti de l'adjacence sans index.

Il est important de noter que le stockage de graphe natif et le traitement de graphe natif ne sont ni bons ni mauvais - ce sont simplement des compromis d'ingénieries classiques. L'avantage du stockage graphique natif est que son spécialement conçue pour les performances et l'évolutivité. L'avantage du stockage de graphe non natif, en revanche, est qu'il dépend essentiellement d'un backend non graphique mature (tel que MySQL) dont les caractéristiques de production sont bien comprises par les équipes d'exploitation.

Le traitement natif des graphes (adjacence sans index) améliore les performances de traversée, mais au détriment de la création de certaines requêtes qui n'utilisent pas des traversées difficiles ou gourmandes en mémoire.

2.3 Détails de structure

2.3.1 Modèles conceptuels

Outre l'adoption d'une approche spécifique de stockage et de traitement, une base de données orientée graphe doit également adopter un modèle de données spécifique [9].

- **Graphe simple (**voir Figure 2.7**)**

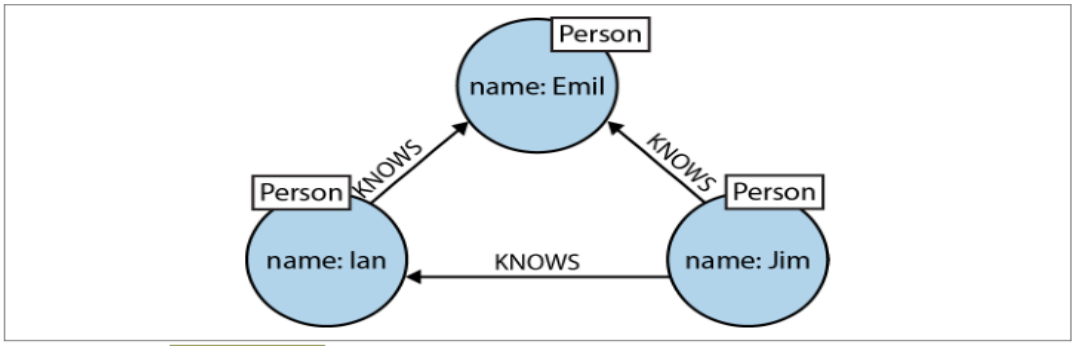

Figure 2.7 -Graphe simple [9]

- **Graphe de propriétés (**voir Figure 2.8**)**

Un graphe de propriété aux caractéristiques suivantes :

• Il contient des nœuds et des relations.

• Les nœuds contiennent des propriétés (paires clé-valeur).

• Les nœuds peuvent être étiquetés avec une ou plusieurs étiquettes.

• Les relations sont nommées et dirigées, et ont toujours un nœud de début et de fin.

• Les relations peuvent également contenir des propriétés.

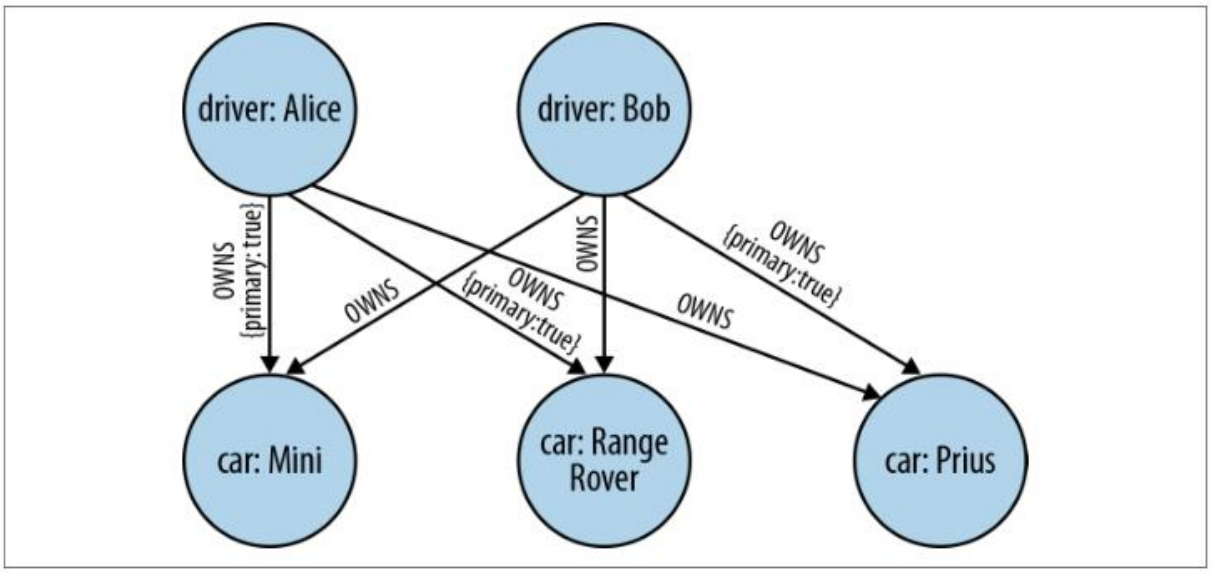

Figure 2.8 - Graphe de propriétés affiné sémantiquement [9]

- **Hypergraphe**

Un hypergraphe est un modèle de graphe généralisé dans lequel une relation (appelée hyper-edge) peut connecter un nombre quelconque de nœuds (voir Figure 2.9). Alors que le modèle de graphe de propriétés permet une relation pour n'avoir qu'un seul nœud de départ et un seul nœud de fin, le modèle hypergraphe autorise un nombre quelconque de nœuds à chaque extrémité d'une relation. Les hypergraphes peuvent être utiles où le domaine se compose principalement de relations plusieurs-à-plusieurs. Par exemple, sur la figure 2.7, nous voyons qu'Alice et Bob sont propriétaires de trois véhicules. Nous exprimons ça en utilisant une seul hyper-arête, alors que dans un graphe de propriétés, nous utiliserions six relations.

Bien qu'en théorie les hypergraphes produisent des modèles précis et riches en informations, en pratique, il nous est très facile de manquer certains détails lors de la modélisation.

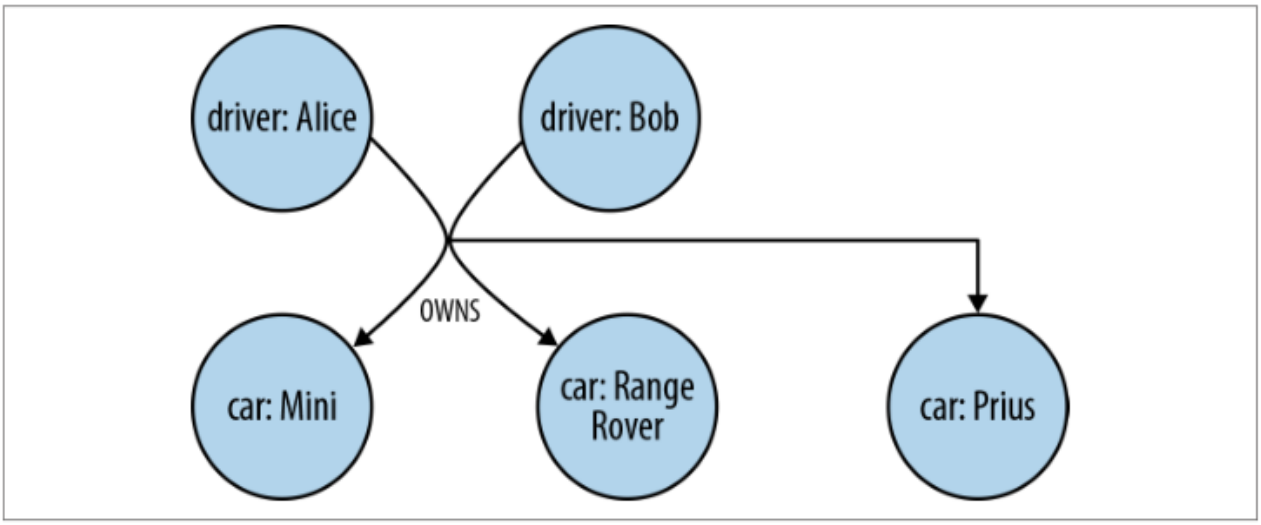

Figure 2.9 - un hypergraphe simple orienté [9]

Les hyper-arêtes étant multidimensionnelles, les hypergraphes comprennent un modèle plus général que les graphes de propriétés. Cela dit, les deux modèles sont isomorphes. Il est toujours possible de représenter les informations d'un hypergraphe sous la forme d'un graphe de propriétés (bien qu'en utilisant plus de relations et nœuds intermédiaires). Qu'il s'agisse d'un hypergraphe ou d'un graphe des propriétés qui vous convient le mieux dépendra de votre modélisation, état d'esprit et les types d'applications que vous créez. Pour la plupart des cas, les graphes de propriétés sont largement considérés comme ayant le meilleur équilibre entre pragmatisme et efficacité de modélisation, d'où leur popularité écrasante dans l'espace des bases de données de graphes. Pourtant, dans les situations où vous devez capturer la maintenance, qualifiant efficacement une relation avec une autre (par exemple, j'aime le fait que vous aimiez cette voiture), les hypergraphes nécessitent généralement moins de primitives que les graphes de propriétés.

- **Triples stores**

Les magasins triples proviennent du mouvement du Web sémantique, où les chercheurs s'intéressent à l'inférence de connaissances à grande échelle en ajoutant un balisage sémantique aux liens qui connecte des ressources Web. A ce jour, très peu de pages Web ont été balisées de manière utile, il est donc rare d'exécuter des requêtes à travers la couche sémantique. Au lieu de cela, la plupart l'effort dans le Web sémantique semble être investi dans la collecte de données utiles et d'informations sur les relations à partir du Web (ou d'autres sources de données plus banales, telles que les applications) et le déposer dans des magasins triple pour interrogation.

Un triple est une structure de données sujet-prédicat-objet. En utilisant des triples, nous pouvons capturer des faits, comme « Fred aime la crème glacée ». Individuellement, des triples simples sont sémantiquement pauvres, mais en masse, ils fournissent un ensemble de données riche à partir duquel on récolte des connaissances et on déduire des connexions. Les magasins triples fournissent généralement les capacités SPARQL de raisonner et de stocker les données RDF.

RDF, la lingua franca (une langue qui est adoptée comme langue commune entre des locuteurs dont les langues maternelles sont différentes) des triples stores et du Web sémantique, peut être sérialisé de plusieurs manières.

L'extrait suivant montre comment les triplets se réunissent pour former des liens de données, en utilisant le format RDF/XML :

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns="http://www.example.org/terms/">
        <rdf:Description rdf:about="http://www.example.org/ginger">
                <name>Ginger Rogers</name>
                <occupation>dancer</occupation>
                <partner rdf:resource="http://www.example.org/fred"/>
        </rdf:Description>
        <rdf:Description rdf:about="http://www.example.org/fred">
                <name>Fred Astaire</name>
                <occupation>dancer</occupation>
                <likes rdf:resource="http://www.example.org/ice-cream"/>
        </rdf:Description>
</rdf:RDF>
```

Les magasins triples entrent dans la catégorie générale des bases de données orientées graphes car ils traitent des données qui, une fois traitées, ont tendance à être liées logiquement. Ils ne sont pas cependant des bases de données de graphes « natives », car elles ne prennent pas en charge l'adjacence sans index, ni leurs moteurs de stockage optimisés pour stocker des graphes de propriétés. Les magasins triples stockent les triples en tant qu'artefacts indépendants, ce qui leur permet d'être mis à l'échelle horizontalement pour le stockage, mais les empêche de traverser rapidement les relations. Pour effectuer des requêtes graphiques, les magasins triples doivent créer des structures connectées à partir de faits indépendants, ce qui ajoute de la latence à chaque requête. Pour ces raisons, le point idéal pour un magasin triple est l'analyse, où la latence est une considération secondaire, plutôt que OLTP.

Bien que les bases de données de graphes soient conçues principalement pour les performances de traversé et l'exécution d'algorithmes de graphes, il est possible d'utiliser en tant que magasin de sauvegarde derrière un point de terminaison RDF/SPARQL. Par exemple, l'API Blueprints SAIL fournit une interface RDF pour plusieurs bases de données de graphes, dont Neo4j. En pratique, cela implique un niveau d'isomorphisme fonctionnel entre les bases de données orientées graphes et les magasins triples. Cependant, chaque type de magasin est adapté à un type différent décharge de travail, avec des bases de données de graphes optimisées pour les charges de travail graphiques et les traversées rapides.

2.3.2 Modèles de données non graphiques et schémas de stockage utilisés dans les bases de données orientées graphes

Outre les modèles de graphes conceptuels, les bases de données de graphes intègrent souvent différents types de schémas stockage et modèles de données qui ne ciblent pas spécifiquement les graphes mais sont utilisés dans divers systèmes pour modéliser et stocker des graphes. Ces modèles incluent des collections de paires clé-valeur, des documents et les tuples (utilisés dans différents types de magasins NoSQL), les relations et les tables (utilisées dans les bases de données relationnelles traditionnelles) et les objets (utilisés dans les bases de données orientées objet) [8].

- **Collection de paires clé-valeur**

Les magasins clé-valeur sont les magasins NoSQL les plus simples. Ici, les données sont stockées sous la forme d'une collection de paires clé-valeur, en mettant l'accent sur la haute performance et la haute recherche évolutives basées sur des clés. La forme exacte des clés et des valeurs dépend d'un système ou une application. Les clés peuvent être simples (par exemple, un URI ou un hachage) ou structurées. Les valeurs sont souvent codées sous forme de tableaux d'octets (c'est-à-dire que la structure des valeurs est généralement sans schéma). Cependant, un magasin clé-valeur peut également imposer une disposition de données supplémentaire, structurant les valeurs sans schéma.

En raison de la nature générale des magasins clé-valeur, il peut y avoir plusieurs façons de représenter un graphe comme une collection de clé-valeur. Il y a plusieurs exemples concrets de systèmes (Microsoft’s Graph Engine (Trinity), Hyper GraphDB). Par exemple, on peut utiliser des étiquettes de sommets comme clés et coder les voisinages des sommets comme valeurs.

- **Collecte de Documents**

Un document est une unité de stockage fondamentale dans une classe de bases de données NoSQL appelées magasins de documents. Ces documents sont stockés dans des collections. Plusieurs collections de documents constituent une base de données. Un document est encodé à l'aide d'une norme sélectionnée de format semi-structuré, par exemple JSON ou XML. Les magasins de documents étendent les magasins clé-valeur dans cela qu'un document peut être vu comme une valeur qui a un certain schéma flexible. Ce schéma est composé d'attributs, où chaque attribut a un nom avec une ou plusieurs valeurs. Une telle structure basée sur les documents avec des attributs permet différents types de valeurs, le stockage de paires clé-valeur et récursif (les valeurs d'attribut peuvent être des listes ou des dictionnaires clé-valeur).

Dans tous les magasins de documents étudiés (exemple OrientDB et ArangoDB), chaque sommet est stocké dans un document de sommet. La capacité des documents à stocker des paires clé-valeur est utilisée pour stocker les étiquettes de sommet et les propriétés dans le document de sommet correspondant. Les détails du stockage périphérique, cependant, sont dépendant du système : les arêtes peuvent être stockées dans le document correspondant au sommet source de chaque arrête, ou dans les documents des sommets de destination. Comme les documents n'imposent aucune restriction sur quelles paires clé-valeur peuvent être stockées, les sommets et les arêtes peuvent avoir des ensembles de propriétés différents.

- **Collection de tuples**

Les tuples sont une base de magasins NoSQL appelés magasins de tuples. Un magasin de tuples généralise un magasin RDF : les magasins RDF sont limités aux triplets (ou - dans certains cas - aux 4-tuples, également appelés quads) tandis que les magasins de tuples peuvent contenir des tuples d'une taille arbitraire. Ainsi, le nombre des éléments d'un tuple n'est pas fixe et peut varier, même au sein d'une même base de données. Chaque tuple a un identificateur qui peut aussi être un pointeur mémoire direct.

Une collection de tuples peut modéliser un graphe de différentes manières. Par exemple, un tuple de taille n peut stocker des pointeurs vers d'autres tuples qui contiennent des voisinages de sommets. La correspondance exacte entre ces tuples et ces données de graphe sont spécifiques à différentes bases de données.

- **Collection de tables**

Les tables sont la base des systèmes de gestion de bases de données relationnelles (SGBD). Les tables sont constituées de lignes et de colonnes. Chaque ligne représente un seul élément de donnée, par exemple une voiture. Une seule colonne définit généralement un certain attribut de données, par exemple le couleur d'une voiture. Certaines colonnes peuvent définir des identifiants uniques d'éléments de données, appelés clés primaires. Les clés primaires peuvent être utilisées pour implémenter des relations entre les données. Une relation un-à-un ou un-to-plusieurs peut être implémentée avec une seule colonne supplémentaire qui contient la copie d'une clé primaire de l'élément de données associé (cette copie de clé primaire est appelée clé étrangère). Une relation plusieurs à plusieurs peut être implémentée avec une table dédiée contenant les clés étrangères des éléments de données associés.

Pour modéliser un graphe comme une collection de tables, on peut implémenter des sommets et des arêtes sous forme de lignes dans deux tableaux séparés. Chaque sommet a une clé primaire unique qui constitue son identificateur. Les arrêtes peuvent être liés à leurs sommets source ou destination en se référant à leurs clés primaires (comme clés étrangères). Les graphes de propriété, ainsi que les prédicats RDF, peuvent être modélisés avec des colonnes supplémentaires (exemple Titan et JanusGraph).

- **Collection d'objets**

On peut également utiliser des collections d'objets dans les systèmes de gestion de base de données orientée objet (OODBMS) pour modéliser des graphes. Ici, les éléments de données et leurs relations sont implémentés sous forme d'objets liés à une certaine forme de pointeurs. Les détails des graphes de modélisation comme les objets dépendent fortement de conceptions spécifiques (exemple VelocityGraph).

2.3.3 Bases de données de graphes natives basées sur LPG

Les systèmes de base de données de graphes décrits dans les sections précédentes sont tous basés sur un backend de base de données qui n'a pas été construit à l'origine uniquement pour gérer des graphiques. Dans ce qui suit, nous décrivons les bases de données de graphes natives basé sur LPG : systèmes spécialement conçus pour maintenir et traiter les graphes [8].

- **Neo4j : pointeurs directs**

Neo4j est le système de base de données de graphes le plus populaire, selon différents classements de bases de données. Neo4j implémente le modèle LPG en utilisant une conception de stockage basée sur des enregistrements de taille fixe. Un sommet v est représenté par un enregistrement de sommet, qui stocke (1) les étiquettes de v, (2) un pointeur vers une liste chaînée des propriétés de v, (3) un pointeur vers la première arête adjacent à v, et (4) quelques drapeaux. Une arête e est représentée par un enregistrement d'arête, qui stocke (1) le type d'arête (une étiquette), (2) un pointeur vers une liste chaînée des propriétés de e, (3) un pointeur vers deux enregistrements de sommets qui représentent des sommets adjacents à e, (4) des pointeurs vers les AL des deux sommets adjacents, et (5) certains drapeaux. Chaque enregistrement de propriété peut stocker jusqu'à quatre propriétés, selon la taille de la valeur de la propriété. Les grandes valeurs (par exemple, les chaînes longues) sont stockées dans un magasin dynamique séparé. Le stockage des propriétés des enregistrements de sommets et d'arêtes extérieurs permettent à ces enregistrements d'être petits. De plus, si aucune propriété n'est accédée dans une requête, ils ne sont pas chargés du tout. L'AL d'un sommet est implémenté comme une liste doublement liée. Une arête est stockée une fois, mais fait partie de deux de ces listes chaînées (une liste pour chaque sommet adjacent). Ainsi, une arête a deux pointeurs vers les arêtes précédentes et deux pointeurs vers les arêtes suivantes. Le Figure 2.10 décrit la conception Neo4j, le Figure 2.11 montre les détails des enregistrements de sommets et d'arêtes.

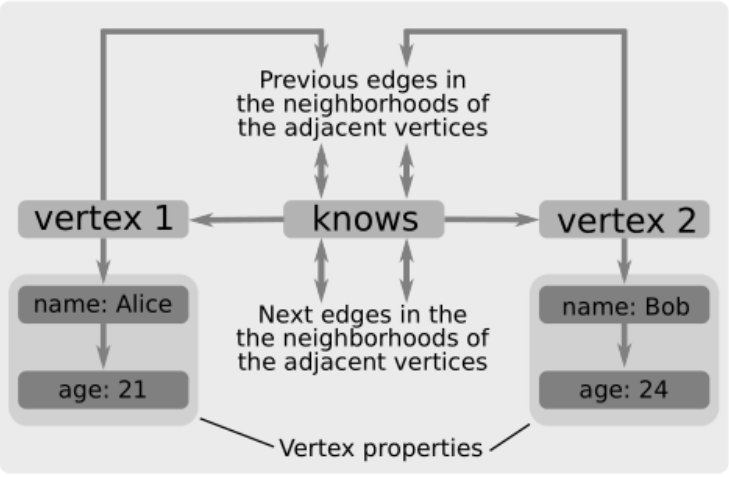

Figure 2.10 - Résumé de la structure Neo4j [8]

Un concept de base dans Neo4j utilise des pointeurs directs : un sommet stocke des pointeurs vers les emplacements de ses voisins. Ainsi, pour les requêtes de voisinage ou les

traversées, on n'a pas besoin d'index et peut à la place suivre des pointeurs directs (sauf pour les sommets racines dans les traversées). En conséquence, la complexité de la requête ne dépend pas de la taille du graphe mais elle dépend uniquement de la taille du sous-graphe visité.

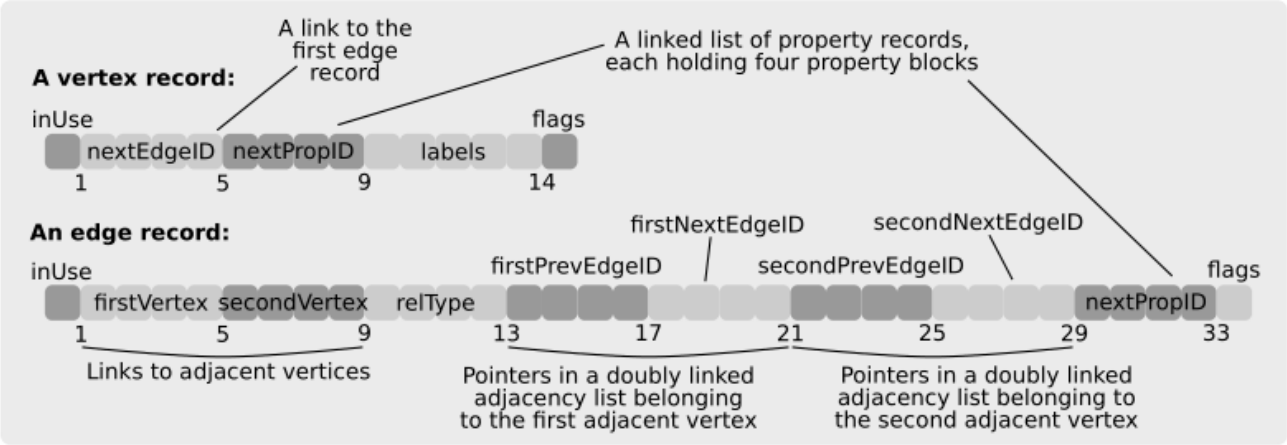

Figure 2.11 - Un enregistrement Neo4j [8]

- **Sparksee/DEX: Arbres B+ et Bitmaps**

Sparksee est un système de base de données orientée graphe qui a été anciennement connu sous le nom de DEX. Sparksee implémente le modèle LPG de la manière suivante. Les sommets et les arrêtes (les deux sont appelés objets) sont identifiés par des identifiants uniques. Pour chaque nom de propriété, il y a un arbre B+ associé qui mappe les identificateurs de sommet et d'arête aux valeurs de propriété respectives. L'inverse le mappage d'une valeur de propriété aux l'identificateur de sommet et d'arête est maintenu par un bitmap, où un bit défini sur 1 indique que l'identificateur correspondant à une valeur de propriété. Les étiquettes, les sommets et les arêtes sont mappés les uns aux autres de la même manière. De plus, pour chaque sommet, deux bitmaps sont stockés : un bitmap indique les arrêtes entrants, et un autre les arrête sortants. De plus, deux arbres B+ conservent les informations sur les sommets auxquels une arête est connectée (un arbre pour chaque direction d'arête). La figure 2.12 illustre des exemples de mappages.

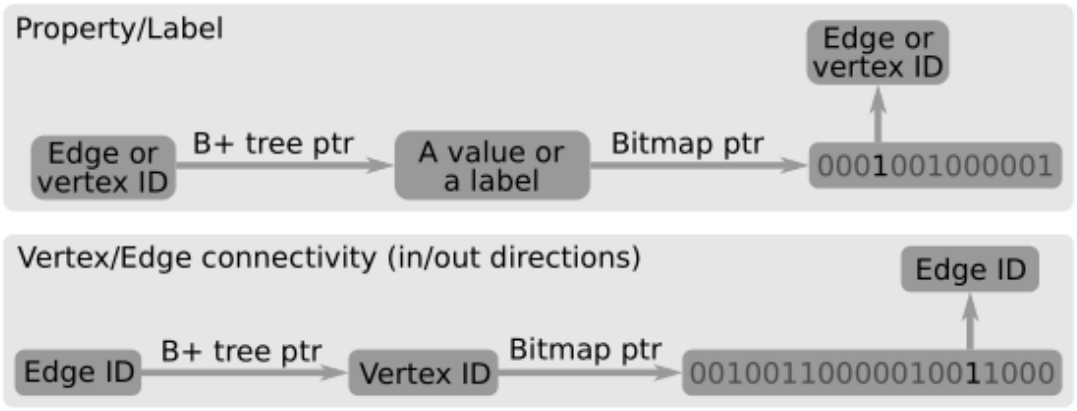

Figure 2.12 - Carte Sparksee pour les propriétés [8]

Sparksee est l'un des rares systèmes qui ne sont pas basés sur des enregistrements. Au lieu de cela, Sparksee utilise des cartes mis en œuvre sous forme d'arbres B+ et de bitmaps. L'utilisation de bitmaps permet à certaines opérations d'être exécutées en tant qu'opérations au niveau du bit. Par exemple, si l'on veut trouver tous les sommets avec certaines valeurs de propriétés telles que "âge" et "prénom", on peut simplement trouver deux bitmaps associés aux propriétés "âge" et "prénom", puis dérivez un troisième bitmap résultant de l'application d'une opération AND au niveau du bit sur les deux bitmaps d'entrée.

Les bitmaps non compressés pouvaient grossir de façon ingérable. Comme la plupart des graphes sont clairsemés, les bitmaps indexés par des sommets ou des arêtes contiennent principalement des zéros. Pour atténuer les grandes tailles de ces bitmaps clairsemés, ils sont découpés en clusters de 32 bits. Si un cluster contient un bit différent de zéro, il est stocké explicitement. Le bitmap est alors représenté par une collection de paires (cluster-id, bit-data). Ces paires sont stockées dans une structure arborescente triée pour permettre une recherche, une insertion et une suppression efficaces.

- **GBase : format de matrice d'adjacence clairsemée**

GBase est un système qui ne peut représenter que la structure d'un graphe orienté ; il ne stocke ni propriétés ni étiquettes. L'objectif de GBase est de maintenir une compression de la matrice d'adjacence d'un graphe de manière à pouvoir récupérer efficacement toutes les arêtes entrantes et sortantes d'un sommet sélectionné sans les frais généraux de stockage prohibitif de la matrice $O(n^2)$. Simultanément, l'utilisation de la matrice d'adjacence permet de vérifier en temps O (1) si deux sommets arbitraires sont connectés. Pour compresser la

matrice d'adjacence, GBase la coupe en K^2 blocs quadratiques (il y a K blocs le long de chaque ligne et colonne).

Ainsi, les requêtes qui récupèrent les voisins entrants et sortants de chaque sommet ne nécessitent que de récupérer K blocs. Le paramètre K peut être optimisé pour des bases de données spécifiques. Lorsque K devient plus petit, il faut récupérer plus de petits fichiers (en supposant un bloc est stocké dans un fichier). Si K grossit, il y a moins de fichiers mais ils deviennent plus gros et génèrent des frais généraux. D'autres optimisations peuvent être faites lorsque les blocs ne contiennent que des zéros ou des uns ; cela permet des taux de compression plus élevés.

2.3.4 Détails et optimisations de l'organisation des données

Par la suite, lors de l'examen des bases de données, on considère différents aspects de l'organisation des données et on fournit plus d'informations sur les types de backend de base de données de graphes fondamentaux [8].

- **Diviser les données en enregistrements**

Les bases de données graphiques organisent généralement les données en petites unités appelées enregistrements. Un enregistrement contient des informations sur une certaine entité unique (par exemple, une personne), ces informations sont organisées en champs logiques spécifiés (par exemple, un nom, un prénom, etc.). Un certain nombre d'enregistrements est souvent conservé dans un bloc contigu en mémoire ou sur disque pour améliorer la localité d'accès aux données. Les détails de l'organisation des données basée sur les enregistrements dépendent fortement d'un système spécifique. Par exemple, une base de données relationnelle peut traiter une ligne de table comme un enregistrement, les magasins de clés-valeurs conservent souvent une seule valeur dans un seul enregistrement, tandis que dans les magasins de documents, un seul document peut être un enregistrement. Surtout, certains systèmes autorisent des enregistrements de taille variable (par exemple, ArangoDB), d'autres n'autorisent que des enregistrements de taille fixe (par exemple, Neo4j). Enfin, nous observons que si certains systèmes (par exemple, certains magasins triples tels que CrayGraph Engine) ne mentionnent pas explicitement les enregistrements, les données pourraient toujours être implicitement organisées dans une manière basée sur les enregistrements. Dans les magasins triples, on associerait naturellement un triple à un disque.

Les bases de données de graphes utilisent souvent un ou plusieurs enregistrements par sommet (ces enregistrements sont parfois référencés en tant qu'enregistrements de sommet).

Neo4j utilise plusieurs enregistrements de taille fixe pour les sommets, tandis que les bases de données de documents utiliser un document par sommet (par exemple, ArangoDB).

Les arêtes sont parfois stockées dans le même enregistrement avec les sommets associés (source ou destination) (par exemple, Titan ou JanusGraph). Autrement, les arêtes sont stockées dans des enregistrements d'arêtes séparés (par exemple, ArangoDB).

- **Stockage de données dans des structures d'index**

Les bases de données graphiques utilisent couramment des index pour accélérer les requêtes. Désormais, les systèmes basés sur des backends non graphiques, par exemple des SGBDR ou des magasins de documents, s'appuyer sur l'infrastructure d'indexation existante présente dans de tels systèmes. Les bases de données graphiques natives emploient des structures d'index pour les voisinages de chaque sommet, souvent sous la forme de pointeurs directs.

En plus d'utiliser des structures d'index pour conserver les emplacements des données, certaines bases de données stockent également les données du graphique dans les index eux-mêmes. Dans de tels cas, l'index ne pointe pas vers une certaine donnée mais l'index lui-même contient les données souhaitées. Des exemples de systèmes avec une telle fonctionnalité sont Sparksee/DEX et moteur graphique Cray. Pour maintenir les index, le premier utilise des bitmaps et des arbres B+tandis que ce dernier utilise des tables de hachage.

- **Utilisation d'arêtes légers**

Certains systèmes (par exemple, OrientDB) autorisent les arêtes sans étiquettes ou propriétés à stocker en tant qu'arêtes légères. De telles arêtes sont stockées dans les enregistrements des correspondants sommets source et/ou destination. Ces arêtes légères sont représentées par l'identificateur de leur sommet de destination, ou par un pointeur vers ce sommet. Cela peut économiser de l'espace de stockage et accélérer la résolution des différentes requêtes de graphes telles que la vérification de la connectivité de deux sommets.

- **Lier des enregistrements avec des pointeurs directs**

Dans les systèmes basés sur des enregistrements, les sommets et les arêtes sont stockés dans les enregistrements. Pour permettre une résolution efficace des requêtes de connectivité (c'est-à-dire vérifier si deux sommets sont connectés), ces enregistrements doivent pointer vers d'autres enregistrements. Une option consiste à stocker des pointeurs directs (c'est-à-dire

les adresses de mémoire) aux enregistrements connectés respectifs. Par exemple, un enregistrement d'arrête peut stocker les pointeurs directs vers les enregistrements de sommets avec des sommets adjacents. Une autre option consiste à affecter chaque enregistrement un identifiant unique et utilisez ces identifiants au lieu de pointeurs directs pour faire référence à d'autres enregistrements. D'une part, cela nécessite une structure d'indexation supplémentaire pour trouver l'emplacement physique d'un enregistrement en fonction de son identifiant. En revanche, si l'emplacement physique change, il est généralement plus facile de mettre à jour la structure d'indexation au lieu de modifier tous les pointeurs directs associés.

Un système donné peut également utiliser des pointeurs directs pour éviter de maintenir une structure d'indexation dédiée supplémentaire pour parcourir le graphe. Notez qu'un index peut toujours être utilisé pour trouver un sommet ; en utilisant directement les pointeurs dans ce contexte signifie que seule la structure des données d'adjacence n'a pas d'index. L'utilisation de pointeurs directs peut accélérer les parcours de graphes, tant que les parcours d'index supplémentaires sont évités. Cependant, lorsque les données d'adjacence doivent être mises à jour, généralement un grand nombre de pointeurs doivent également être mis à jour, générant une surcharge supplémentaire.

2.4 Détail des cas d'utilisation

Les organisations du monde entier se tournent vers la technologie des graphes. Dans ce qui suit, nous décrirons quelques-unes des utilisations les plus populaires de la base de données orientée graphe.[10]

2.4.1 Services financiers

Quels que soient leurs efforts, les criminels financiers sont liés par des relations, qu'il s'agisse de relations avec d'autres criminels, de lieux, ou bien sûr, des comptes bancaires. La technologie graphique tire parti de ce fait pour ouvrir de nouvelles possibilités dans le monde des services financiers.

- **Blanchiment d'argent**

Le problème : Conceptuellement, le blanchiment d'argent est simple. L'argent sale circule pour le mélanger avec des fonds légitimes et ensuite transformé en actifs durables. C'est le genre de processus qui a été utilisé dans l'analyse des Panama Papers. Plus précisément, un transfert d'argent circulaire implique un criminel qui s'envoie d'importantes sommes d'argent obtenues frauduleusement à lui-même ou elle-même, mais le cache à travers une longue et complexe série de virements entre comptes ‘’normaux’’. Ces comptes

‘’normaux’’ sont en fait des comptes créés avec des identités synthétiques. Ils sont généralement partagés certaines informations similaires car elles sont générées à partir de identités volées (adresses e-mail, adresses, etc.) et c'est les informations liées qui font de l'analyse graphique un si bon ajustement pour les rendre révéler leurs origines frauduleuses.

La solution graphique : Pour simplifier la détection des fraudes, les utilisateurs peuvent créer un graphique à partir des transactions entre les entités ainsi que les entités qui partagent certaines informations, y compris les adresses email, mots de passe, adresses, et plus. Une fois qu'un graphique est créé, l'exécution d'une requête simple trouvera tous les clients avec des comptes qui ont des informations similaires, et révèlent quels comptes s'envoient de l'argent les uns aux autres.

- **Détecter les transfère de l'argent acquis illégalement**

Le problème : La fraude implique une personne, qui transfère des biens. Il peut s'agir de drogues, mais lorsqu'il s'agit de l'industrie financière, implique généralement de l'argent. La personne transfère de l'argent à son propre compte, et l'argent est ensuite transféré à un autre opérateur frauduleux qui se trouve généralement dans un autre pays. Traditionnellement, les modèles basés sur des règles créent des alertes et les comptes suspects sont signalés par les humains. L'apprentissage automatique est également utilisé pour prédire les décisions. Cependant, il est souvent difficile d'améliorer les modèles car les comptes eux-mêmes ont généralement des informations limitées.

La solution graphique : C'est là qu'interviennent les graphes. Avec la technologie graphique, les utilisateurs peuvent prendre les informations de transaction en tant que arêtes et générer plus de fonctionnalités des comptes en fonction des relations et des transactions environnantes. Pour exemple, en utilisant des scores de centralité basés sur des graphes, les utilisateurs peuvent déterminer à quel point certains comptes sont proches des comptes muletiers connus. De plus, ces faux comptes partagent souvent des informations similaires (telles que l'adresse ou les numéros de téléphone) parce que ces informations sont nécessaires à l'enregistrement des comptes et les criminels n'ont autant d'identités sur lesquelles puiser. En utilisant des requêtes basées sur des graphes, les utilisateurs peuvent découvrir rapidement les comptes avec des relations similaires ou les comptes impliqués avec des modèles comme la circulation et les signaler pour plus enquête. Grâce à cette méthode, la technologie des graphes peut améliorer les modèles d'apprentissage automatique formés pour découvrir les transfère de l'argent acquis illégalement.

- **Détection des fraudes en temps réel**

Le problème : Dans le monde d'aujourd'hui, les consommateurs exigent un accès instantané aux services et aux transferts d'argent, ce qui ouvre des opportunités aux criminels. Pour exemple, les applications de services de paiement essaient de livrer de l'argent aussi rapidement que possible aux utilisateurs valides tout en garantissant que l'argent n'est pas envoyé à des fins illicites ou en cachant le vrai récepteur en étant envoyé dans des itinéraires détournés. Cela nécessite une détection des fraudes en temps réel.

La solution graphique : Parce que les graphes permettent des réponses rapides aux demandes de renseignements et parce qu'ils élargissent l'accès aux données, ils sont devenus une technologie populaire dans le domaine de la détection des fraudes en temps réel. Lorsque vous étudiez des transactions à l'aide de la technologie de création de graphiques, ce ne sont pas seulement les transactions qui peuvent être modélisées dans des graphiques. Les graphiques sont très flexibles, ce qui signifie que des informations hétérogènes peuvent également être modélisées. Par exemple, les adresses IP client, ATM, la géolocalisation, les numéros de carte et les identifiants de compte peuvent tous devenir des sommets, et les connexions peuvent toutes devenir des arrêtes. Le graphe de propriété est souvent utilisé pour la détection de fraude, en particulier dans les analyses de l'emplacement des banques et des guichets automatiques, car les utilisateurs peuvent concevoir les règles pour détecter la fraude sur la base d'ensembles de données.

2.4.2 Fabrication

La fabrication est une question de relations et de dépendances, ce qui rend les technologies graphiques conviennent parfaitement pour découvrir plus d'informations dans un manière rapide.

- **Nomenclature**

Le problème : Une voiture a 30 000 pièces. Alors quel est l'impact du changement d'une pièce ? Quoi si vous changez quelques pièces à la fois ? Ce type d'analyse peut être très compliqué avec une voiture, où chaque pièce peut avoir potentiellement des milliers de dépendances. Les requêtes pour de telles analyses prenaient autrefois une beaucoup de temps en raison de toutes les jointures de table en plusieurs étapes nécessaires.

La solution graphique : En utilisant des requêtes graphiques, le temps de réponse peut être réduit à quelques secondes voire plus rapide, ce qui signifie que l'analyse interactive

en temps réel est désormais réaliste. Les graphiques prennent les relations que toutes les parties ont avec chacune autre, et les rend clairs, de sorte que tout défaut ou dépendance négative deviennent également clairs. En utilisant un graphique pour une facture d'analyse des matériaux, vous pouvez créer un modèle d'analyse des informations sur le produit et dépendances. Vous pouvez également ajouter des informations supplémentaires sur les produits, tels que les vendeurs, ingénieurs, fournisseurs, matériaux, âge des matériaux, etc. pour créer une variété de modèles. Cela vous permet d'explorer des composants avec confiance, la fiabilité des fournisseurs, les options des fournisseurs, et plus encore.

- **Traçabilité**

Le problème : La traçabilité est d'une grande importance dans le monde de la fabrication. Un constructeur automobile pourrait devoir émettre un rappel pour un modèle de voiture parce que ce modèle spécifique à un composant qui a été produit à partir d'une usine pendant un créneau horaire limité. L'entreprise doit retracer la causalité composante et ensuite trouver les voitures qui sont sur le marché ou sur la route de l'usine. Cela peut être très difficile. La plupart des entreprises ont une base de données de production qui gère le lot des informations sur le produit. Mais ils ont également une base de données de vente au détail distincte, et une base de données de vente distincte, et une base de données d'expédition distincte. Il est compliqué de découvrir toutes les informations pertinentes pour trouver les voitures avec le problème, où ils ont été expédiés et à qui ils ont été vendus.

La solution graphique : Sans technologies graphiques, les analystes doivent combiner toutes ces bases de données et exécutez une requête de traversée d'une voiture spécifique vers la base de données d'usine qui gère la ligne de production. Tout cela nécessite une modélisation de données complexe et de nombreuses jointures, à moins que l'entreprise ne dispose d'une base de données de graphes pour se connecter toutes les relations et algorithmes graphiques pour mettre en évidence les connexions et les informations pertinentes.

- **Gestion des données de référence**

Le problème : Dans de nombreuses usines de fabrication, l'équipe de conception peut utiliser un certain nom d'un composant. Le service de production peut utiliser un autre nom et d'autres départements peuvent également utiliser d'autres noms, le tout pour le même article. Lorsque des problèmes surviennent ou lorsque l'entreprise souhaite se renseigner plus sur certains cas d'utilisation et quels composants sont impliqués pour cet élément spécifique, tous

les noms différents et incohérents rendent difficile la correspondance avec le bon élément et la découverte des composants en question.

La solution graphique : Les graphiques RDF sont parfaits pour modéliser différents composants et utiliser les relations et les liens qu'ils entretiennent les uns avec les autres. Les graphes RDF utilisent tous ces informations pour créer une couche de métadonnées qui aide à déterminer si des noms différents indiquent le même élément, si les éléments sont liés, et même indiquer si différents éléments peuvent être utilisés de manière interchangeable en raison de leurs similitudes. Ceci est également utilisé dans le monde pharmaceutique, pour identifier différents produits chimiques, médicaments et noms génériques.

Sans graphe RDF, les applications intègrent généralement une logique pour aider à trouver les éléments corrects. Mais cette logique ne fonctionne pas toujours dans toutes les bases de données, car chacune a souvent des conventions de nommage différentes. Et si le DBA qui a créé cette logique d'application quitte l'entreprise, la logique se perd souvent.

L'utilisation d'un graphe RDF pour résumer les informations dans une couche de métadonnées supprime non seulement cette dépendance à la logique d'application, mais crée également des couches utiles supplémentaires. Les graphiques RDF ne vous dit pas seulement si des noms différents se réfèrent au même article manufacturé ; ils exposent également les relations et les dépendances de ces éléments avec d'autres éléments, ce qui facilite la recherche d'autres éléments connexes et découvrir des faits et des relations implicites.

En un mot, avec un graphique RDF, vous obtenez un moyen d'avoir des données auto descriptive qui capture le contenu et la sémantique d'une manière lisible et utilisable par machine. De plus, il n'est pas nécessaire de s'assurer que la logique d'application est maintenue jusqu'à date ; les applications s'améliorent automatiquement à mesure que le contenu et la qualité RDF aller mieux.

2.4.3 Gouvernement

De l'activité criminelle à la recherche des contacts, de nombreux problèmes liés au gouvernement peuvent être traité avec des technologies de graphes.

- **Fraude fiscale**

Le problème : La fraude fiscale est un problème croissant pour de nombreux gouvernements. Les gouvernements deviennent souvent plus courts de ressources tandis que les criminels grandissent plus inventif. Non seulement cela, mais la technologie moderne

présente de nouveaux défis pour les gouvernements moins agiles et fournit des moyens faciles de déplacer de l'argent à travers les frontières internationales, incitant ainsi encore plus les criminels.

Désormais, les criminels peuvent créer des sociétés fictives, puis faire passer ces sociétés en des entités légitimes. L'argent est acheminé via plusieurs comptes, dans les deux sens et partout, dans un chemin détourné et délibérément déroutant qui finit par se retrouver avec l'argent du gouvernement entre les mains des criminels.

La solution graphique : Démêler ces chemins complexes n'est pas une tâche facile, avec plusieurs couches de relations cachées au plus profond des données. Suivi du chemin à travers chaque couche de la relation est une tâche difficile, mais les bases de données graphiques peuvent aider à comprendre la structure des entités corporatives coquilles, fournir des outils de visualisation pour aider à l'enquête manuelle, découvrir les motifs en plusieurs sauts, et découvrez les chemins qui serpentent et conduisent finalement à une personne ou à une organisation corrompue.

Dans un autre cas d'utilisation de fraude fiscale, les technologies graphiques peuvent également révéler des propriétés cachées et des salaires que les gens essaient de cacher. Par exemple, un individu peut recevoir des salaires de plusieurs entreprises et essayer de se cacher certains d'entre eux. Ou il ou elle peut avoir d'autres actifs de placement qui n'étaient pas divulgué. Et lorsqu'il y a des revenus provenant de plusieurs sources, y compris des immeubles locatifs, des redevances, des partenariats, des successions et des fiducies, il peut être difficile de tout suivre et de s'assurer que les bons impôts sont payés, surtout lorsque plusieurs personnes sont impliquées dans la propriété des actifs.

Les technologies graphiques peuvent présenter ces actifs et les personnes impliquées pour rendre les relations entre eux et l'argent dû plus claires.

- **Enquête criminelle**

Le problème : Les bases de données graphiques révolutionnent l'analyse des activités criminelles. Quelques crimes se produisent sur une petite échelle opportuniste. Mais le type de crime que la police travaille ensemble pour suivre et éliminer a tendance à se produire à grande échelle avec de nombreuses personnes, gangs, entreprises et même des lieux interconnectés, ce qui signifie qu'il n'a pas tendance à se produire en silos.

La solution graphique : Mettre des données dans des graphiques fournit un moyen naturel et efficace d’identifier les réseaux criminels et rechercher des modèles. En appliquant

des algorithmes basés sur des graphiques comme le PageRank ou la centralité, il devient plus facile de rechercher des personnes vulnérables dans le graphique, de découvrir plus d'informations sur les lieux et même de rechercher des personnes importantes et des gangs criminels potentiels. Par exemple, en appliquant la centralité d'intermédiation, les utilisateurs peuvent trouver le « maillon le plus faible », c'est-à-dire le sommet sur lequel repose le graphe. Si vous supprimez ce sommet, le graphique entier peut s'effondrer, ce qui signifie que vous venez peut-être de trouver la cheville ouvrière d'un gang criminel.

- **Recherche de contacts**

Le problème : La recherche des contacts pathologiques est une activité essentielle dans le monde entier. Les gens contractent de nouvelles maladies hautement contagieuses et continuent de mener une vie normale, visitant des cinémas, des gymnases bondés, des mariages bondés et des pratiques de chorale propageant cette maladie partout où ils vont.

Lorsqu'une personne est diagnostiquée, la course pour retrouver tous ceux qui ont été en contact avec cette personne malade pour leur demander de se mettre en quarantaine devient une course contre le temps. Les traceurs de contact doivent pouvoir faire leur travail le plus rapidement possible pour arrêter la propagation de la maladie.

Les bases de données graphiques, qui mettent fortement l'accent sur les relations, sont idéales pour analyser les modèles de maladies. Les analystes peuvent saisir des informations sur les personnes qui ont été testées malades, les membres de la famille et les amis avec lesquels ils ont interagi, et les endroits qu'ils ont visités, pour localiser rapidement les hotspots et les connexions. De cette façon, les analystes peuvent travailler plus rapidement pour isoler les personnes malades et prévenir de nouvelles épidémies.

Il existe trois niveaux pour le suivi des contacts avec des graphes :

Premièrement, il est nécessaire de comprendre les relations des gens, les communautés, et les lieux qu'ils visitent que les graphiques peuvent rendre clairs s'ils sont fournis avec suffisamment de données mobiles.

Deuxièmement, les graphes doivent trouver la propagation possible, ce qui signifie examiner les liens potentiels entre les personnes susceptibles de propager la maladie. La personne a-t-elle voyagé en bus ? Pouvons-nous identifier tout le monde dans le bus ?

Troisièmement, les traceurs de contact doivent trouver des "super diffuseurs" et se précipiter pour isoler ces personnes en premier. Cela implique de trouver les personnes qui

ont des contacts larges et denses et qui sont susceptibles d'avoir des liens avec de nombreuses communautés différentes. Cela implique d'explorer les graphiques avec des notions de centralité et d'intermédiarité, pour trouver les personnes fortement connectées.

2.4.4 Règlement des données et la vie privée

À mesure que les données gagnent en valeur, les entreprises les collectent, les vendent et les utilisent plus activement. Dans le même temps, les lois, réglementations et normes relatives aux données ont également considérablement augmenté. Mais à mesure que les données continuent d'augmenter en volume, la gestion de ces données et la garantie de la confidentialité des données et des réglementations deviennent de plus en plus complexes.

- **Exigences générales en matière de protection des données**

Le problème : Partout dans le monde, les professionnels de la gestion des données sont toujours aux prises avec le problème de la gestion du GPDR. Comment peuvent-ils continuer à préserver la vie privée des personnes, répondre aux demandes d'accès aux données et répondre aux demandes d'oubli, entre autres ?

L'une des difficultés majeures réside dans la découverte de ce qui est stocké dans chaque base de données. Les données sont déplacées. Les données sont transformées. Les données peuvent être consommées par les utilisateurs et d'autres processus. Et il peut être extrêmement difficile de suivre et de retracer ce qui s'est passé avec toutes ces données.

Mais ce n'est pas le seul problème. Les données peuvent avoir été stockées à l'origine dans un tableau, mais des rapports ont ensuite été créés à partir des données. Les rapports contiennent des informations et ont également des règles d'accès. Si quelqu'un veut exercer son droit à l'oubli, suivre cette trace électronique de l'endroit où les données se trouvaient à l'origine, où elles ont été copiées et où elles ont été utilisées dans les tableaux et les rapports, tout cela devient extrêmement compliqué. Remplir les exigences de cet élément du GPDR est une tâche monumentale

La solution graphique : Le suivi de la lignée des données correspond parfaitement à un graphique. Les différentes étapes du cycle de vie des données peuvent être suivies et parcourues, sommet par sommet, en suivant les bords. Avec le graphique, il devient possible de suivre un chemin et de voir où l'information résidait à l'origine, où elle a été copiée et où elle a été utilisée. Avec toutes ces informations présentées dans un graphique, il devient plus

simple pour les professionnels des données de déterminer comment répondre aux demandes GDPR et rester conforme.

- **Confidentialité des données**

Le problème : Les organisations doivent limiter l'accès aux données. Par exemple, peut-être qu'ils veulent seulement autoriser certains ordinateurs personnels à ouvrir certains fichiers ou ils veulent que certaines équipes, départements et projets aient accès à certaines données. Les droits d'accès sont compliqués à gérer et la visibilité dans quelles équipes ont accès, quelles équipes ne devraient pas avoir accès plus, et quelles équipes ont besoin d'un meilleur accès pour effectuer leur travail, peut être très compliqué. Souvent, cette structure de données doit être fluide afin que la structure hiérarchique puisse changer dynamiquement. Mais faire cela de manière transparente est difficile, et une véritable perspicacité dans ce qui est changé et comment cela est changé est difficile à réaliser.

La solution graphique : Le graphique peut rendre une telle structure hiérarchique très dynamique et la requête graphique peut améliorer le temps de réponse pour modifier l'accès aux données. En raison des contrôles d'accès complexes et dynamiques, les applications doivent vérifier l'autorisation d'un matériau spécifique à chaque fois. Mais les requêtes graphiques peuvent suivre le réseau si efficacement que les applications peuvent trouver les autorisations en temps réel.

- **La cybersécurité**

Le problème : La cybersécurité est un sujet très important dans la bataille pour le cloud. Cela implique des domaines complexes tels que la détection du trafic invalide, la chasse aux cybermenaces et la détection des logiciels malveillants. Une solution pour aborder ces sujets consiste à utiliser la technologie graphique pour améliorer la cybersécurité.

La solution graphique : Les technologies graphiques capturent les connexions entre les entités de données, c'est-à-dire comment les ordinateurs sont connectés sur un réseau informatique. Ils exploitent des signaux supplémentaires du graphique pour la détection d'anomalies.

Les technologies graphiques Ils peuvent améliorer la détection des cybermenaces en permettant une exploration interactive et visuelle des données de sécurité. Cela crée un environnement idéal pour la chasse aux cybermenaces.

2.4.5 Le marketing

Le marketing concerne les relations que les spécialistes du marketing doivent comprendre avec leurs clients, les relations que leurs clients entretiennent entre eux, les produits, les relations entre différents produits et bien plus encore pour fournir efficacement aux clients ce qu'ils veulent.

- **Analyse client à 360 degrés**

Le problème : Aujourd'hui, les entreprises ont de plus en plus d'informations sur les clients, y compris:

• Données de base : nom, âge, sexe, adresse,

• Transactions : achats, types d'articles achetés, heures d'achat,

• BigData : journaux des centres d'appels, lignes de trafic, flux de clics Web, activités SNS,

• Prédictions : classification, signatures gustatives (souvent créées par différents des modèles).

Mais les entreprises n'utilisent souvent pas ces informations de manière aussi complète qu'elles devraient. Créer une véritable analyse client 360 est difficile.

La solution graphique : Lorsque toutes les données marketing listées ci-dessus sont collectées et intégrées dans la plateforme physique, il est généralement difficile de les analyser toutes. Mais ces ensembles de données peuvent être logiquement intégrés sur des graphiques et les utilisateurs du graphique peuvent simplement visualiser toutes les informations environnantes d'une entité (le client). Avec des graphiques, les marketeurs peuvent acquérir une vision plus complète de leurs clients - les relations que les clients entretiennent les uns avec les autres, les relations entre tous les produits achetés, et plus encore. Ensuite, les utilisateurs de graphiques peuvent exécuter des algorithmes pour découvrir plus de détails sur le client.

L'affichage de toutes les informations sur un client spécifique est important pour comprendre le client et effectuer une analyse client à 360°, afin de découvrir quelles prédictions (généralement créées grâce à l'apprentissage automatique) sont vraies et pourquoi.

- **Recommandations de produits**

Le problème : Les technologies non graphiques peuvent prendre en charge les moteurs de recommandation, mais le graphe crée un délai de rentabilisation plus rapide. Les bases de données de graphes sont construites pour que les relations entre les clients et les

produits qu'ils aiment acheter sont déjà configuré - il devient donc facile et rapide d'exécuter des algorithmes à travers les données pour trouver des recommandations.

De plus, les recommandations en temps réel deviennent de plus en plus importantes que jamais. Mais cela nécessite la capacité de corréler les informations sur les produits, inventaire des clients, comportement passé des clients, informations actuelles sur les fournisseurs, la logistique, et même les données sociales telles que les publicités cliquées et les produits explorés via les réseaux sociaux. Ceci est extrêmement difficile pour certains types de bases de données.

La solution graphique : La technologie pour collecter toutes ces données et établir des connexions pour obtenir un aperçu rapide des besoins des clients et des tendances des produits, et ensuite, pour fournir des recommandations en temps réel, c'est la base de données graphique. En fait, de nombreuses grandes entreprises s'appuient sur l'analyse graphique pour fournir leurs recommandations parce que les relations sont déjà établies, et l'analyse de ces relations pour fournir des recommandations est très rapide.

- **Média social**

Le problème : Les médias sociaux sont une partie croissante de notre monde aujourd'hui et les relations sont un élément clé de celle-ci. Il en va de même pour connecter les utilisateurs et garantir la validité de ces utilisateurs. Dans le monde des médias sociaux, les comptes « sockpuppet » sont un problème. Les Sockpuppets sont de faux comptes gérés par des robots. Ils travaillent pour s'assurer les sujets ou les mots-clés semblent plus importants en les aimant ou en les partageant, et leur donnant ainsi l'air d'être à la mode.

Parfois, cela est assez inoffensif, bien que toujours trompeur pour les détaillants et les clients. Pensez à un influenceur Instagram qui achète des abonnés et aime se faire paraître plus populaire. À d'autres moments, cela peut être très sérieux, avec des pays utilisant des bots pour encourager les sujets d'actualité qui déstabilisent les autres gouvernements.

La solution graphique : Les bases de données graphiques peuvent traverser les réseaux sociaux et les données associées très rapidement, c'est pourquoi les sociétés de médias sociaux telles que Facebook, LinkedIn, et Twitter exploitent tous une sorte de traitement graphique au sein de leur plateforme pour identifier les amis et les familles à travers le monde. Nous avons déjà les recommandations de produits mentionnées dans l'exemple précédent. Un similaire peut être utilisé pour fournir des recommandations d'utilisateurs, des images,

produits, et plus encore, ainsi que pour détecter les activités frauduleuses et comptes sockpuppet.

2.4.6 IA et l'apprentissage automatique

L'IA et l'apprentissage automatique sont généralement considérés comme des domaines de grand intérêt en raison de leur promesse d'améliorer les résultats commerciaux et de créer de nouveaux impactes. Le graphe peut être utilisé pour augmenter la science des données de plusieurs manières clés.

- **Ingénierie des fonctionnalités**

Le problème : En matière d'apprentissage automatique, les modèles d'apprentissage automatique reposent sur les données. Plus ces données sont bonnes, plus elles sont riches, profondes et complètes, meilleur est le modèle d'apprentissage automatique (généralement). Il y a toute une étape à la création d'un modèle d'apprentissage automatique, appelé ingénierie des fonctionnalités, qui consiste à enrichir les données.

Voici un exemple simplifié : un data scientist peut avoir l'adresse du domicile et l'adresse du bureau d'une personne, mais avoir la distance en miles serait meilleure pour le modèle d'apprentissage automatique. Le data scientist doit effectuer cette étape supplémentaire d'ingénierie des fonctionnalités pour trouver cette distance en miles et créer des données qui sont meilleures pour le modèle d'apprentissage automatique. Cependant, il existe certains types d'ingénierie de fonctionnalités qui peuvent être plus compliqué à réaliser, surtout quand il s'agit de chercher des relations avec les données et mettre ces relations au premier plan. Essayer de le faire peut nécessiter trop de jointures et être lent et fastidieux à accomplir.

La solution graphique : Le plus souvent, les fonctionnalités d'apprentissage automatique peuvent être créées via un graphe en exécutant des algorithmes de graphes sur un ensemble de données qui a été chargé dans une base de données de graphes et créer des données enrichies qui peuvent ensuite être utilisées pour l'apprentissage automatique. Cette étape de l'ingénierie des fonctionnalités fournit au modèle d'apprentissage automatique des informations plus complètes et utiles.

- **Réseaux de neurones graphiques**

Le problème : Nous avons déjà discuté de la façon dont les graphes peuvent aider avec les recommandations mais qu'en est-il des recommandations prédictives ? Par exemple,

que se passe-t-il si un magasin de vente au détail en ligne veut envoyer des recommandations à un client, avec le moment déterminé par le moment où le client devrait manquer de l'article? L'ajout de prédictions aux recommandations peut être compliqué mais cela peut aussi augmenter considérablement les profits. C'est souvent un domaine inexploité opportunité pour de nombreuses entreprises.

La solution graphique : De nombreux data scientists commencent à s'intéresser aux graphes neuronaux, qui peuvent capturer le graphe lui-même en tant qu'entrée de l'apprentissage automatique et les réseaux de neurones. Le graphique peut potentiellement contenir plus informations que les tableaux standard en raison de la flexibilité du modèle. Les modèles d'apprentissage automatique avec des informations capturées à partir de graphiques offrent souvent de meilleures performances que l'apprentissage automatique basé sur une entrée de forme de table. Ce type de réseau de neurones est déjà en cours d'évaluation dans toutes les industries, et certains résultats montrent qu'il améliore la précision, par exemple dans la détection de fraude financière. Pour exécuter de telles techniques, conserver les informations d'origine au format graphique pour plus de flexibilité est essentiel, donc la base de données graphique est le composant clé pour créer des flux de travail qui utilisent des techniques d'apprentissage automatique de pointe.

2.5 Limites des bases de données orientées graphes

Malgré la recherche et la pratique à long terme dans ce domaine, il existe de nombreux problèmes difficiles qui restent ouverts dans la gestion des données graphiques. Ils ont de l'influence sur les restrictions de fonctionnalité des bases de données de graphes. D'autres sont spécifiquement liées au Big Analytics. Défis concernant certains problèmes spécifiques. Les problèmes de la technologie des bases de données de graphes sont résumés comme suit. [11]

2.5.1 Restrictions de fonctionnalité

Interrogation déclarative : la plupart des bases de données de graphes commerciales ne peuvent pas être interrogées à l'aide d'un langage déclaratif. Seuls quelques fournisseurs proposent une interface de requête déclarative. Cela implique également un manque de capacités d'optimisation des requêtes.

Partitionnement des données : la plupart des bases de données de graphes n'incluent pas la fonctionnalité de partitionnement et distribuer des données dans un réseau informatique. Ceci est essentiel pour soutenir horizontalement l'évolutivité aussi. Il est

difficile de partitionner un graphe d'une manière qui n'entraînerait pas la plupart des requêtes devant accéder à plusieurs partitions.

Opérations vectorielles : elles prennent en charge une procédure qui écrit séquentiellement des données à partir de plusieurs tampons à un seul flux de données ou lit les données d'un flux de données à plusieurs tampons. Les bases de données NoSQL mises à l'échelle horizontalement prennent en charge ce type d'accès aux données. Ce semble que ce n'est pas le cas dans les bases de données de graphes aujourd'hui.

Restrictions du modèle : les possibilités de schéma de données et les définitions de contraintes sont restreint dans les bases de données de graphes. Par conséquent, les incohérences des données peuvent rapidement réduire leur utilité. Souvent, le modèle de graphe lui-même est restreint. Rappelons, par exemple, les nœuds Neo4jne peuvent ne pas se référencer directement. Il pourrait y avoir des cas réels où l'autoréférence est nécessaire.

2.5.2 Grandes exigences d'analyse

Extraction de graphe : une question est de savoir comment extraire efficacement un graphe ou une collection de graphes, à partir de magasins de données non graphiques. La plupart des systèmes d'analyse de graphes supposent que le graphe est fourni explicitement. Cependant, dans de nombreux cas, le graphe peut devoir être structuré en joignant et en combinant des informations provenant de différentes ressources qui ne sont pas nécessairement graphique. Même si les données sont stockées dans une base de données de graphes, nous ne faisons souvent que besoin de charger un ensemble de sous-graphes de ce graphique de base de données pour une analyse plus approfondie.

Coût élevé de certaines requêtes : la plupart des graphiques du monde réel sont très dynamiques et aurez de gros volumes de données à un rythme très rapide. Un défi ici est de savoir comment stocker la trace historique de manière compacte tout en permettant une exécution efficace des requêtes ponctuelles et tâches d'analyse globales ou centrées sur le quartier. Les principales différences par rapport aux SGBD temporels développés dans le passé sont l'échelle des données, l'accent mis sur les environnements distribués et en mémoire, et la nécessité de prendre en charge l'analyse globale. La dernière tâche nécessite généralement de charger des instantanés historiques entiers dans la mémoire.

Traitement en temps réel : Comme indiqué, la découverte de données de graphes se déroule essentiellement dans des environnements batch, par exemple dans Giraph. Certains produits destinés à la découverte de données et à des analyses complexes qui fonctionneront

en temps réel. Un exemple est uRIKA24 - Appliance BigData pour l'analyse de graphes. Il utilise la technologie de mémoire et le processeur multithread pour prendre en charge les opérations non batch sur les triplets RDF.

Algorithmes de graphes : Des algorithmes de graphes plus complexes sont nécessaires dans la pratique. L'idéal est que la base de données de graphes doit comprendre les requêtes analytiques qui vont au-delà des requêtes k-hop pour petit k. Les auteurs de ont comparé les performances de 12 bases de données graphiques open source en utilisant quatre algorithmes de graphes fondamentaux (par exemple, problème de plus court chemin source simple et Page Rank) sur des réseaux contenant jusqu'à 256 millions d'arêtes. Étonnamment, les bases de données de graphes les plus populaires ont atteint les pires résultats dans ces tests. Les bases de données de graphes courants (comme les bases de données relationnelles) ont tendance à donner la priorité à l'exécution de requêtes à faible latence sur l'analyse de données à haut débit.

Parallélisation : dans le contexte des grands graphiques, il est nécessaire de paralléliser les algorithmes de traitement des données de graphe lorsque les données sont trop volumineuses pour être gérées sur un seul serveur. Il est nécessaire de comprendre l'impact sur les performances de l'algorithme de traitement des données graphiques lorsque les données ne rentrent pas toutes dans la mémoire disponible et pour concevoir des algorithmes explicitement pour ces scénarios.

Données de graphes hétérogènes et incertaines : il est nécessaire de trouver des méthodes automatisées pour gérer l'hétérogénéité, l'incomplétude et l'incohérence entre les différents ensembles de données de BigGraph qui doivent être intégrés sémantiquement pour être efficacement interrogé ou analysé.

2.6 Conclusion

Dans ce chapitre nous avons présenté les bases de données NoSQL orientées graphe on les caractérise, nous avons détaillé leur structures conceptuelles, logique et physiques, nous avons expliqué leurs organisations en vue d'optimisation et enfin nous avons décrit les cas d'utilisations de ses bases et leurs limites. Dans le prochain chapitre nous présentons le cas d'étude à savoir le réseau de distribution des produits alimentaires au niveau national.

Chapitre 3

Réseau de Distribution des Produits Alimentaires

3.1 Introduction

Du fait de l'importance de l'alimentation pour la vie, les hommes s'organisent toujours de façon à ce que leur approvisionnement soit toujours le plus proche possible ; les marchés se situent à proximité des agglomérations. La distribution des produits alimentaires occupe une place importante dans l'activité de distribution en général. Les changements socio-économiques survenus et leurs perspectives d'évolution (la mondialisation, la croissance démographique, l'urbanisation, …) et l'importance que revêt la distribution sur le double plan micro-économique et macro-économique, donneront à la commercialisation des produits alimentaires dans les prochaines décennies un rôle capital dans l'accompagnement et l'impulsion du développement économique ainsi que dans la disponibilité d'une alimentation saine et de qualité. Les produits alimentaires peuvent subir plusieurs manipulations avant d'arriver au consommateur final, mettant ainsi à contribution plusieurs acteurs (producteur, grossiste, détaillant, …).

3.2 Des produits alimentaires et de la distribution

Une alimentation suffisante quantitativement et qualitativement est tout à la fois le facteur d'une bonne santé physique, mentale, d'un meilleur rendement économique, de moindre dépense en santé publique et un facteur de stabilité politique. Les produits alimentaires sont des composants biologiques qui possèdent des caractéristiques spécifiques qui doivent être prises en considération lors de leur commercialisation. Aussi, certains changements socio-économiques ayant une influence sur la fonction de distribution, nécessitent d'être pris en compte pour assurer une distribution efficace de ces produits. Cette section décrira les spécificités des produits alimentaires ayant une influence sur leur commercialisation et insistera sur l'importance de l'adaptation de celle-ci aux changements socio-économiques. Ce qui nous donne deux sous-sections consacrées respectivement aux caractéristiques des produits alimentaires affectant leur distribution et la description de la chaîne alimentaire [12].

3.3 Caractéristiques des produits alimentaires

L'alimentation est une question qui peut être analysée par différents spécialistes, car elle est un fait économique, social, culturel, religieux, technique, industriel… mais c'est avant tout, un besoin vital pour l'être humain et un facteur de bonne santé physique. Les produits alimentaires sont des matières biologiques qui doivent arriver en bon état au consommateur final pour au moins deux raisons : la santé du consommateur et la compétitivité sur le marché.

Tous les produits alimentaires sont périssables, ce qui est variable c'est la durée de conservation. Sur ce point, on distingue entre deux types de produits : les produits frais et les autres [12].

Denrées alimentaires fraiches ou conservées par le froid

Ces denrées sont généralement de courte durée de conservation (moins d'un mois, exception faite sur certains produits comme les semi-conserves), dont la commercialisation exige à tous les stades la mise en œuvre de moyens frigorifiques spécialisés.

L'épicerie et les boissons

Ces produits sont de durée de conservation moyenne ou longue, conservés par divers procédés (stérilisation, sucrage, embouteillage, cuisson, etc.). Leur commercialisation doit être effectuée à l'abri des variations excessives de température et d'hygrométrie (taux d'humidité dans l'air) et de la lumière. Les produits conservés par le froid sont les plus fragiles.

3.4 Chaine alimentaire, urbanisation et distribution

La capacité de la fonction de distribution à s'adapter aux changements est l'un des déterminants de son efficacité. Ce point de notre travail, examine quelques-uns des changements qui sont susceptibles d'influer sur l'organisation de la distribution. Au préalable, nous devons présenter la chaîne alimentaire [12].

3.4.1 La chaîne alimentaire

La chaîne alimentaire est une succession de phases (les maillons) que traversent les produits alimentaires depuis le producteur jusqu'au consommateur. Ces phases sont : la production agricole, la transformation (industrie agroalimentaire), la distribution et la consommation.

Les maillons sont séparés dans l'espace et dans le temps et les séparations se traduisent par un coût de transport et de stockage, mais aussi par des prises en considération spéciales quant à la viabilité des denrées alimentaires (conservation).

3.4.2 Distribution et urbanisation

La fonction de distribution est une fonction influencée par les changements qui surviennent dans son environnement. Plusieurs facteurs transforment la distribution à court-terme (la réglementation) et à moyen-long termes (changement des habitudes alimentaires, travail des femmes…). Nous mettrons, ici, l'accent sur le rôle de l'urbanisation dans le modelage futur des systèmes de distribution, spécialement dans les pays en développement.

La croissance urbaine dans les pays industrialisés est quasi-stable (les perspectives en sont de 1,1 milliard pour l'horizon 2050). Quant aux pays en développement, ils vont enregistrer une explosion de leur population urbaine qui va passer de 2,5 milliards en 2009 à5,19 milliards vers 2050, soit le double en 40 ans (ONU, Division de la Population, 2009).

Une telle situation ne peut pas être sans conséquence sur l'organisation de l'économie d'une manière générale, et sur la fonction alimentaire et commerciale, en particulier. L'urbanisation pourrait avoir deux conséquences sur l'alimentation : une transformation des habitudes alimentaires et des systèmes de distribution.

Par rapport à la demande, il faut se rappeler que la demande des citadins est très variée et s'oriente vers des produits tels les fruits et légumes frais, le lait, et les viandes... Une telle situation est amplifiée par les nouvelles tendances alimentaires à l'échelle mondiale. L'analyse de la composition des échanges mondiaux des produits alimentaires montre que ceux-ci s'orientent depuis les années 80 de plus en plus vers des produits frais au détriment des céréales.

Ainsi, l'amélioration du pouvoir d'achat et l'urbanisation remodèlent la consommation. Autrement dit, l'urbanisation contribue à modifier la qualité (les variétés) des produits demandés.

La seconde conséquence de l'urbanisation sur l'alimentation est celle de la nécessaire adaptation des systèmes de commercialisation à cette croissance urbaine, spécialement dans les pays en développement. L'explosion urbaine implique tout simplement qu'une grande partie de la population sera dépendante de systèmes d'approvisionnement.

3.5 La distribution en Algérie

L'Algérie est un très vaste pays dont le climat est chaud sur la majorité du territoire et dont la population a plus que triplé en un demi-siècle, avec une population urbaine qui augmente à un rythme supérieur à celui de la population totale. A la fin des années 80, le pays connaît une libéralisation de son économie, après deux décennies de dirigisme, mais en

matière de production, il ne couvre pas tous ses besoins que ce soit en produits alimentaires (de base) ou en biens d'équipement. Le déficit est flagrant et persistant, ce qui fait du pays l'un des premiers importateurs de produits alimentaires de base (céréales, lait, …). L'ensemble de ces facteurs a, ou peut avoir, une influence sur le système de distribution à court ou à long terme. Par ailleurs, l'organisation actuelle de la distribution en Algérie n'est que l'aboutissement des différentes politiques adoptées par le pays depuis l'indépendance [12].

3.5.1 Cadre général de la distribution en Algérie

Il est nécessaire de connaître les caractéristiques de l'environnement de la distribution qui peuvent avoir une influence sur cette fonction ainsi que des changements qui y surviennent. Nous présentons l'environnement de la distribution en Algérie, nous y examinerons le cadre naturel (l'espace géographique et le climat) ; la population et les tendances de sa croissance.

L'espace naturel

L'Algérie est un vaste pays d'une superficie de 2 381 741 de km2, riverain de la Méditerranée et occupant une place centrale au Maghreb. Le pays borde la Méditerranée au nord sur 1280 km, alors que la distance séparant son nord de son sud est de plus de 2000 km. L'espace y est contrasté, avec trois blocs naturels de différents climats et compositions physiques. Le territoire recèle des ressources naturelles intéressantes mais aussi des contraintes souvent difficiles à surmonter. Cette première sous-section présente les caractéristiques du territoire, en mettant l'accent sur ses atouts et ses contraintes par rapport à l'activité de distribution.

Les caractéristiques physiques et climatiques

Le territoire algérien est composé de trois grands blocs naturels qui présentent chacun des caractéristiques bien distinctes : le Nord (le Tell), les Hauts-Plateaux et le Sud.

Le Nord (Tell) : Le Tell représente 7 % du territoire algérien, large de 80 à 190 km et long de 1200 km. Ce relief est composé en grande partie de montagnes (Ouarsenis, Chenoua, Djurdjura, Baborset Bibans…). Le mont Lala Khedidja, dans le Djurdjura, en est le point culminant car il s'élève à 2 308 mètres d'altitude. Les plaines du Tell abritent avec les vallées voisines la grande majorité des terres fertiles : les plaines littorales étroites (Annaba, la Mitidja) et les plaines du bassin intérieur à l'ouest (Tlemcen, Sidi Bel Abbes, Chlef, Mascara). Le climat au nord de l'Algérie est méditerranéen donc tempéré, hiver froid, été

chaud et sec. Les températures hivernales varient entre 8°C et 15°C ; elles grimpent à 25°C en mai pour atteindre une moyenne de 28°C à 30°C en juillet et août, parfois elles dépassent les 30°C.L'étude physico-chimique de la qualité des sols au nord montre que ceux-ci sont plus adaptés aux cultures intensives irriguées : maraîchage intensive, cultures industrielles, fourrages irrigués pour l'élevage bovin laitier en intensif et des cultures arboricoles essentiellement des agrumes et arbres à pépins.

Les Hauts- Plateaux : Les Hauts-Plateaux sont la région intermédiaire entre le nord et le sud, ils représentent 9 %du territoire national et sont composés de reliefs hauts et plats comme leur nom l'indique. Leur altitude est variable de 800 et 1100 m. Le climat y est semi-aride (continental), chaud et sec en été avec des températures allant de 35 ° à 40° C, très froid en hiver avec des températures de 5° à -7°C. Cette région est aussi caractérisée par un vent très fort et de la grêle, qui ne sont pas sans conséquences sur l'environnement et la végétation. Les sols y sont fragiles et peu profonds (10 à 30 cm), cette fragilité est aggravée par la désertification, ce qui rend les terres peu favorables à l'agriculture, et fait que les cultures qui y sont effectuées sont faites à sec (céréales, légumes secs et fourrages).

Le Sud : La partie saharienne qui couvre 4/5 de la superficie du pays (quelques 2 millions de km²) est constituée principalement de regs (désert de pierres), d'ergs (désert de dunes), d'oasis et de massif montagneux. Elle est limitée au nord par l'Atlas saharien. Plus au sud, au cœur du Sahara, le massif du Hoggar, dont le point culminant est le plus haut sommet de l'Algérie avec 3 000 mètres au mont Tahat, est constitué de roches volcaniques formant des pics, des aiguilles volcaniques et de hauts plateaux désertiques. À l'est du Hoggar, le Tassili n'Ajjer, haut plateau aride perché à plus de 1 000 mètres d'altitude, dresse des formations rocheuses fortement érodées. En hiver les températures affichent 15 à28°C alors qu'en été elles atteignent 40 à 45°C, voire plus. Malgré son hostilité, cette région recèle des ressources naturelles indéniables : pétrolières, gaz, divers minerais, importante nappe phréatique mais aussi d'autres potentialités qui restent à exploiter : notamment le tourisme et l'énergie solaire. La présentation des ressources naturelles d'un territoire donné ne peut être complète sans celle des ressources hydriques, indispensables à la vie humaine et à l'activité économique.

Les ressources hydriques en Algérie

L'eau est la source de la vie sur la terre ; c'est un facteur déterminant de la concentration des populations et de l'activité économique. C'est, également un des enjeux majeurs du XXI° siècle. Une grande partie de l'humanité en est privée, spécialement dans les

pays en développement (en Afrique, au Moyen-Orient, une partie de l'Asie). En Algérie, la situation présente et à venir n'est pas rassurante.

Atouts et contraintes de l'espace naturel algérien

L'espace algérien est un espace contrasté sur plusieurs plans ; il offre de nombreux atouts mais impose aussi de nombreuses contraintes.

Les atouts : Le territoire jouit de plusieurs atouts qui lui confèrent d'importants avantages. Il y a, d'abord, sa situation géographique au nord-ouest de l'Afrique du Nord, au centre du Maghreb qui pourrait l'avantager si les relations intermaghrébines étaient plus prospères. Ensuite, il ya la proximité avec l'Union Européenne, premier client et premier fournisseur du pays. La longueur des côtes sur 1280 km est un autre atout et ce n'est pas pour rien que le transport maritime assure 95 % du commerce extérieur par le biais d'une dizaine de ports de commerce. Le littoral est, également, une énorme potentialité pour le tourisme et la pêche qui pour l'heure ne sont pas pleinement exploitées.

Les contraintes : L'espace algérien présente quelques difficultés dont certaines entravent directement l'activité de distribution. Certes, ces difficultés ne sont pas des fatalités, mais dans l'état actuel des choses, elles sont pesantes. En premier, l'étendue du territoire pose problème parfois. Il est très coûteux d'approvisionner des populations dispersées sur de grandes étendues alors que la production est essentiellement concentrée sur la bande littorale. En matière de distribution, la distance est un coût. Une grande distance implique plus de moyens techniques (transport, stockage, manutention s'il y a lieu et autres), nécessite, impérativement une fluidité dans la circulation de l'information mais aussi le recours à plusieurs intermédiaires, donc un circuit plus long. Outre la distance, il y a aussi le climat qui est chaud d'une manière générale sur une grande partie du territoire et pour une longue période de l'année. Pour la distribution, cela suppose la disponibilité de moyens de transport-stockage adaptés à différents échelons de la chaîne alimentaire. Au-dessus de toutes ces contraintes nous mettons celle de la rareté de l'eau. Les ressources hydriques limitées sont l'un des inconvénients majeurs à la croissance de la production agricole et donc à la disponibilité des produits.

3.5.2 La population

La distribution n'a pas lieu d'exister d'elle-même, sans les fonctions de production et de consommation elle n'aurait aucune utilité. Les changements qui surviennent au niveau de la demande doivent être nécessairement pris en considération en aval de la chaîne, la

distribution étant le maillon-relais transmetteur d'informations le long de la chaîne alimentaire. La croissance démographique et la croissance urbaine sont deux facteurs essentiels qui influent sur l'organisation de cette activité.

Croissance démographique

La population algérienne a connu une importante croissance démographique depuis l'indépendance. Elle est passée de 10 millions d'habitants en 1962 à 44,6 millions au 1erJanvier 2021.

Entre 1962 et 1984 la population a connu une croissance explosive avec un taux de 3,2%, mais depuis, ce taux ne cesse de baisser et il semble que l'Algérie soit entrée dans une transition démographique, puisque la tendance baissière est maintenue.

Il faut noter que la croissance démographique par région est très inégale et présente des écarts importants. Les wilayas du sud et des hauts- plateaux enregistrent jusqu'à présent un fort taux de croissance comparativement au nord, qui peut dépasser le double de la moyenne nationale dans certaines d'entre elle.

Croissance de la population urbaine en Algérie

A l'indépendance, la population algérienne est majoritairement rurale, un tiers seulement habite dans les villes, mais depuis la population citadine, n'a cessé de croitre pour dépasser de loin la moitié des algériens en 1998 (58,3 %). Aussi bien la population urbaine que rurale connaissent un mouvement ascendant, mais, la croissance de la population urbaine l'emporte nettement et le mouvement se prolonge dans le temps. Cette population est estimée en 2018 à plus de 70 % de la population totale et 85 % à l'horizon 2050.

Concernant les effets de la croissance démographique et de la croissance urbaine sur le système de distribution, nous ne pouvons que prendre acte du fait qu'un appareil de distribution traditionnel ne peut convenir à l'ère d'une consommation de masse. Une profonde refonte du système s'impose. La distribution est le maillon qui met en relation la fonction de production avec la fonction de consommation ; et c'est pourquoi le système de distribution doit se mettre au niveau de celui de la production.

A une production de masse, ne peut correspondre qu'une distribution de masse, en cas de surproduction. A l'inverse, dans une économie de pénurie, une distribution inefficace peut aggraver davantage la situation.

3.6 Organisation générale de la distribution

La distribution a fait l'objet d'une libéralisation et depuis, l'organisation de cette activité est caractérisée par :

- Le désengagement de l'Etat et l'amenuisement de son rôle direct dans cette activité ;
- L'essor du commerce privé, le commerce de gros, de détail et dans l'import-export.
- L'essor marqué du commerce informel et la difficulté marquée de son contrôle ou son intégration à l'économie réelle

  Essayons de décrire l'organisation générale du secteur [12].

Rôle de l'Etat dans la distribution

La réforme a conduit l'Etat à se retirer de l'activité de distribution (gros et détail). Une grande partie de structures publiques de commercialisation a disparu, alors que certaines autres sont maintenues et adaptées au nouveau contexte de transition vers l'économie de marché. L'Etat se cantonne dans l'activité des offices non dissous ainsi que la réglementation des prix des produits stratégiques (blé et produits dérivés, pain, lait, …). Reprenons ci-après les restructurations et rôles des structures publiques de commercialisation.

Activité des offices nationaux : Les offices nationaux de commercialisation ont été restructurés pour s'adapter au nouveau contexte de l'économie de marché, ceux qui ne sont pas cités ont cessé leur activité.

OAIC : Cet office est maintenu et continue à exercer son activité, mais avec un secteur privé de plus en plus important. A partir de 1995 le monopole de l'OAIC sur l'importation et la commercialisation des céréales est levé. La mission de l'OAIC est restée constante, celle de la collecte et de la commercialisation des céréales des producteurs nationaux et/ou de l'importation de céréales (et légumes secs) et dérivés. L'office joue le rôle de régulateur de l'offre, par l'écoulement de la production nationale sur le marché local en période de récolte, mais aussi par le recours aux importations pour combler les déficits.

GIPLAIT : Le groupe Industriel des Productions Laitières GIPLAIT SPA a été créé le 10 Mai 1998 à l'issue de la restructuration des ex- offices régionaux de lait (ORLAC, OROLAIT, ORELAIT). Cet office a pour mission, la production et la commercialisation du lait et des produits dérivés. GIPLAIT regroupe 19 filiales de production dispatchées à travers le territoire national et une société « Milk Trade » chargée des importations des matières premières pour approvisionner les unités industrielles.

ONAB : En 1998, l'ONAB est transformé en groupe. Il est le résultat de la restructuration des ex-offices régionaux avicoles (ORAC, ORAVIO et ORAVIE) ainsi que de la filière nutrition animale. Ses filiales spécialisées activent dans les domaines de l'importation et l'approvisionnement des matières premières, la production d'aliments de bétail (avicole et ruminants) et de condiments minéraux vitaminés, la production d'intrants avicoles : œufs à couver chair et ponte, poussins à chair et ponte, poulettes futures pondeuses, la production de poulet de chair et d'œufs de consommation, l'abattage et la transformation, la maintenance et le froid. Le groupe ONAB, est organisé dans un objectif d'encadrement plus poussé des activités de la filière, tant sur le plan de la maîtrise des coûts de production que sur celui de la qualité.

ONCV : Cet office est resté fonctionnel, il s'est transformé en société par action à partir de 1990.L'ONCV produit 500 000 hl de vin/an en moyenne, originaire de ses 4000 ha de vignoble ainsi que de ceux de ses 2 640 viticulteurs nationaux. Son chiffre d'affaires annuel est estimé à 106 millions de dollars. Cependant l'analyse de la situation de la viticulture montre que celle-ci a régressé considérablement depuis l'indépendance (faostat.fao.org), cela est dû à la réorientation des priorités en matière de production (opérations de reconversion).

ONILEV : L'office national interprofessionnel des légumes et des viandes est créé par les dispositions de la loi 08-16 du 27 septembre 2009 avec pour mission d'assurer un service public en matière de régulation, de constitution et de gestion des stocks des produits stratégiques conformément au cahier des charges de sujétions de service public. Les droits et obligations, induits par la mission de service public, font l'objet d'une convention entre l'Etat. L'office est sous tutelle du Ministère de l'Agriculture. La liste des légumes et des viandes concernés par l'opération de régulation, qui peut être élargie à certains fruits, sera fixée par arrêté conjoint des ministères chargés de l'agriculture et du commerce. A ce jour, l'office n'est pas fonctionnel.

L'Etat : assureur de la disponibilité et de l'accessibilité des produits alimentaires de base

Malgré, la libéralisation de la libéralisation de la distribution, trois produits alimentaires de base font exception à cette règle : le lait pasteurisé conditionné en sachet, la semoule et la farine et le pain (ainsi que d'autres produits non alimentaires, voire le site du ministère du commerce). Pour ces trois produits, l'approvisionnement extérieur, la

distribution sont assuré en partie par les entreprises publiques à côté du secteur privé. Les prix de ces produits restent jusqu'à ce jour subventionnés. La politique de l'Etat en la matière vise à assurer la disponibilité de ces produits de large consommation en Algérie, mais aussi à en assurer l'accessibilité à toutes les couches sociales.

Elaboration des règlements et veille à leur application

Depuis la libéralisation de l'économie nationale et de la distribution, le rôle de l'Etat a été restreint spécialement à la fonction de mise en place des règlements et la veille à leur application. L'institution officielle responsable de cette action est le ministère du commerce. Au niveau régional, les directions régionales de commerce et les directions de wilayas se chargent de cette mission. La direction régionale du commerce en liaison avec les structures centrales du ministère du commerce, a pour missions d'animer, d'orienter et d'évaluer les activités des directions de wilayas du commerce relevant de sa compétence territoriale et d'organiser et/ou de réaliser toutes enquêtes économiques sur la concurrence, le commerce extérieur, la qualité et la sécurité des produits. La direction de wilaya de commerce a pour missions de mettre en œuvre la politique nationale arrêtée dans les domaines du commerce extérieur, de la concurrence, de la qualité, de l'organisation des activités commerciales et des professions réglementées, du contrôle économique et de la répression des fraudes.

Malgré la libéralisation de l'activité commerciale et des prix, l'Etat peut intervenir, lorsqu'il le juge nécessaire pour surmonter une situation de crise inhabituelle. Une telle tâche est la prérogative du Conseil de la concurrence.

Le secteur privé

La libéralisation du secteur commercial privé depuis le début des années 1990 a permission essor de celui-ci. Mais, il faut savoir que cet essor ne concerne pas uniquement l'activité commerciale formelle, le secteur informel, a à son tour connu la même évolution.

L'activité formelle : Fin 2008, on compte en Algérie 1 213 8391 commerçants inscrits au Centre National du Registre du Commerce (CNRC) dont 1 104 611 (soit 90 % de l'ensemble des inscrits) sont des personnes physiques, donc des entreprises de petite taille, le reste soit 109 228 sont des personnes morales. La répartition géographique de l'activité commerciale est disparate sur le territoire national et elle suit la concentration géographique de la population et de l'activité économique. En effet, elle est prédominante dans les wilayas du Nord, spécialement au Centre ; Alger compte 154 297 commerçants (12,7 % du total), suivie de Sétif avec 55 547commerçants (4,6 %) et Oran avec 53 972 (4,4 %). Le nombre de

commerçants dans les wilayas du Sud reste modeste : Illizi avec 3 031commerçants (0,2 %), Tindouf avec 3 432, soit 0,3 % et El Bayadh avec 7 060, soit 0,6 %. La branche commerce réalise la moitié de la valeur ajoutée des services.

Concernant l'évolution des formules de distribution, celles-ci restent traditionnelles et prédominées par les petits magasins. La grande distribution, quant à elle, fait actuellement ses premiers pas, avec l'annonce d'un projet de 100 grandes surfaces à créer. L'objectif d'unetelle mesure est de raccourcir les circuits de distribution (en court-circuitant l'étape de gros) et de stimuler la concurrence entre les opérateurs privés. Le marché est ouvert aux opérateurs privés nationaux et étrangers.

L'activité informelle : L'importance du secteur informel dans l'économie des pays en développement suscite beaucoup de questions, aussi bien au niveau des organisations internationales (Bureau International du Travail) qu'au niveau des gouvernements des pays concernés. L'Algérie n'échappe pas à ce phénomène ; l'emploi informel y est évalué en 2003 à plus de 1,254million de personnes soit 17,2 % de l'emploi total, et connaît un taux de croissance moyen annuel de plus de 8% soit 2 fois celui de l'emploi structuré. La part de l'informel dans le commerce est estimée à plus du tiers de l'activité commerciale totale. Près de 200 000commerçants sur 826 470 recensés n'activent pas conformément à la loi et plus de 50 % des commerçants ne s'acquittent pas de leurs cotisations sociales. De plus, 700 marchés illégaux au sein desquels exercent 100 000 personnes, fonctionnent au su et au vu des autorités. Selon les estimations de l'UGCAA (Union Générale des Commerçants et Artisans Algériens) 850000commerçants exercent dans l'informel, alors qu'il y aurait 1500 marchés hebdomadaires informels, 28 marchés de semoule, et 100 000 revendeurs de tabac. Plusieurs facteurs peuvent expliquer l'essor du commerce informel, des facteurs économiques, sociaux et des facteurs liés à l'environnement économique général.

Malgré sa participation à l'absorption du chômage et le fait qu'elle fournisse des ressources aux populations démunies, l'activité informelle pose plusieurs problèmes sur les plans socio-économiques, en matière des conditions de travail, de qualité des biens mis en vente, de concurrence déloyale au secteur officiel et de soustractions aux charges sociales et fiscales. Cependant, lorsqu'il s'agit du commerce de produits alimentaires, les problèmes sont encore plus importants, spécialement ceux de la sécurité sanitaire des produits et même de la garantie de la qualité du produit. Face à une telle situation, le consommateur devient son propre protecteur, du moins, un consommateur expérimenté et avisé. Dans le cas contraire, le consommateur est lésé quant à la qualité (faux produit) ou quant à sa sécurité sanitaire.

Il y a lieu de faire le constat suivant concernant l'organisation actuelle des circuits de distribution en Algérie :

Rôle important de l'Etat dans l'activité

Malgré le rétrécissement du rôle de l'Etat dans l'activité de distribution, celui-ci joue un rôle capital dans l'approvisionnement des marchés, donc dans leur stabilité, spécialement pour les produits alimentaires de base (blé et dérivé, lait). Dans les moments de crise et de forte hausse des prix sur le marché international, le secteur privé se retire et laisse à l'Etat la mission de l'approvisionnement extérieur. L'importance croissante des besoins alimentaires de l'Algérie et sa dépendance du marché extérieur, font que le secteur privé ne peut à lui seul assurer l'approvisionnement du pays en biens alimentaires. Dans cet état des choses une libéralisation complète des marchés ne peut être envisagée pour les produits base. Le sort de la Nation ne peut être mis entre les mains de quelques opérateurs privés. Un autre rôle indéniable de l'Etat dans ce secteur, est la mise en place d'une réglementation adaptée et le suivi de son application. La réglementation est un facteur influent sur l'activité commerciale et il serait utile d'envisager les effets de la loi sur celle-ci. En matière d'application, il va du ressort de l'Etat de veiller à son application.

Organisation du secteur privé

Le commerce est l'une des plus anciennes activités exercées en société, il ne nécessite pas un haut niveau de qualification et tout le monde peut-y accéder. En Algérie, la libéralisation et le taux de chômage élevé ont fait de cette activité un refuge pour une grande partie de la population de différents âges et qualifications. Cela n'a pas été sans conséquences sur l'organisation des circuits qui sont rendus plus complexes.

Cette situation peut avoir un effet amplificateur sur les coûts et donc sur les prix au marché et la responsabilité des acteurs est difficilement déterminée. Il a été déjà expliqué que dans une économie de pénurie, l'inefficacité de la distribution est à l'origine de l'aggravation de la situation.

Importance de l'activité informelle

L'activité informelle dans la distribution a été une règle en Algérie durant la période de l'économie socialiste, elle représentait une voie alternative de travail pour le secteur privé mis à part dans la stratégie de développement du pays. Cependant, la libéralisation du secteur n'a pas fait reculer ce phénomène. L'activité informelle existe toujours, qu'elle soit visible ou pas. En matière de distribution l'importance du secteur informel veut dire que l'offre est

segmentée. Cette situation va à l'encontre du principe du regroupement de l'offre à l'origine d'économies d'échelles et qui est même l'une des raisons d'être des intermédiaires entre producteurs et consommateurs. D'autres problèmes de santé publique liés aux conditions d'hygiènes et de préparations des produits se posent aussi dans le secteur informel.

D'autres problèmes se posent aussi au secteur, ceux liés à l'environnement de la distribution : la persistance de l'état de pénurie, voire son aggravation qui est à l'origine de perturbations de l'offre ; les problèmes d'ordre techniques et logistiques à différents niveaux de la chaîne de distribution (capacité de réception des ports, moyens de stockages…), croissance démographique et urbaine…

La libéralisation de la distribution en Algérie a réduit le rôle direct de l'Etat dans cette activité et a donné un essor au secteur privé, mais aussi au secteur informel. Cette politique de libéralisation a été à l'origine de la résolution des difficultés antérieures posées par le circuit socialiste, mais ce n'est pas pour autant que tous les problèmes le soient, certains persistent dans le temps (notamment ceux liés à l'environnement de la distribution), alors que de nouveaux problèmes se posent, liés à l'inorganisation des circuits et à l'importance de l'activité informelle en la matière.

3.7 Circuits de distribution

Un circuit de distribution est constitué de l'ensemble des agents économiques utilisés par une entreprise productrice pour diffuser ses produits auprès des consommateurs. Un circuit de distribution se caractérise principalement par sa longueur, c'est-à-dire le nombre des agents économiques appartenant au circuit, et par la répartition des fonctions entre ces agents. Selon le critère de la longueur, on distinguera les circuits directs où le producteur assure l'ensemble des tâches de distribution auprès des consommateurs, des circuits indirects, c'est-à-dire comportant des distributeurs intermédiaires en nombre plus ou moins important. Ces circuits indirects peuvent être longs ou courts. Il est d'usage d'appeler circuit court, un circuit de distribution composé d'un producteur et d'un ensemble de détaillants. Un circuit long comporte, au minimum, un intermédiaire supplémentaire. Il s'agit, en général, d'un grossiste situé entre producteur et détaillants (voir Figure 3.13) [13].

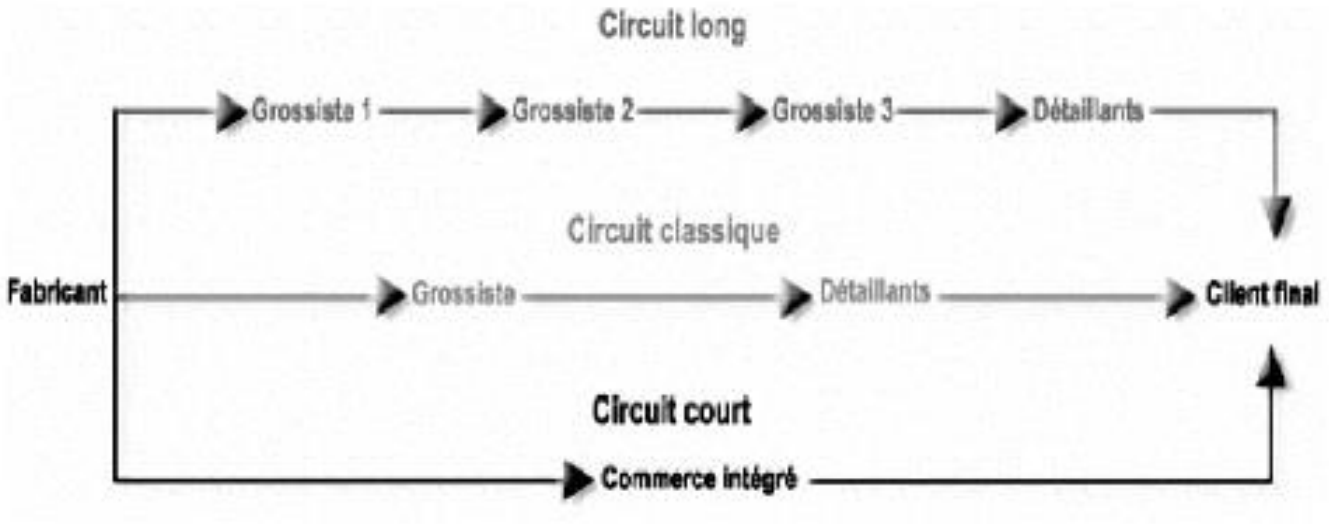

Figure 3.13 – Circuits de distribution [14]

3.8 Conclusion

A travers ce chapitre nous avons décrit la distribution des produits alimentaires en général et en Algérie en particulier, en concentrons sur les facteurs qui influencent la distribution ainsi que son organisation et les modes typiques de distribution (Circuits de distribution). Dans le prochain chapitre nous modélisons et nous implémentons ces circuits de distribution.

Chapitre 4

Modélisation et Implémentation

4.1 Introduction

Ce chapitre est consacré à la présentation des outils qui ont permis la réalisation de notre base de données décisionnelle NoSQL orientée graphe et l'ensemble des fonctionnalités qu'offre cette dernière sous forme de captures d'écran et des descriptions.

La base de données crée sert à modéliser les données de réseau de distribution des produits alimentaires au niveau national, pour cela nous établissons un modèle de l'existant puis on affaire a un schéma en étoile qui représente les différentes dimensions et les mesures pour la modélisation et des transformations sur ce schéma pour construire le modèle dimensionnel graphe (GDM), nous implémentons et nous instancions ce modèle en créant la base de données sur **Neo4j Desktop** puis on exploit les trois outils installées avec la base Neo4j a savoir **Browser** pour le développement, **Bloom** pour l'exploration et les analyses et enfin **Charts** pour créer les rapports en format tableau ou graphique d'aide à la décision en utilisant le langage Cypher.

4.2 Modélisation

4.2.1 Le modèle de données

A partir les circuits de distribution (Figure 3.13) notamment le circuit classique et les relations qui existent entre le grossiste et le producteur d'une part et entre le grossiste et le détaillant d'autre part, on établit le modèle de données statique (Figure 4.14), l'activité a analysée dans ce modèle c'est la distribution suivant plusieurs dimensions qui s'intersectent, ce qui nous conduit pour la modélisation de la base de données décisionnelle NoSQL orientée graphe a opté pour le schéma en étoile (Figure 4.15).

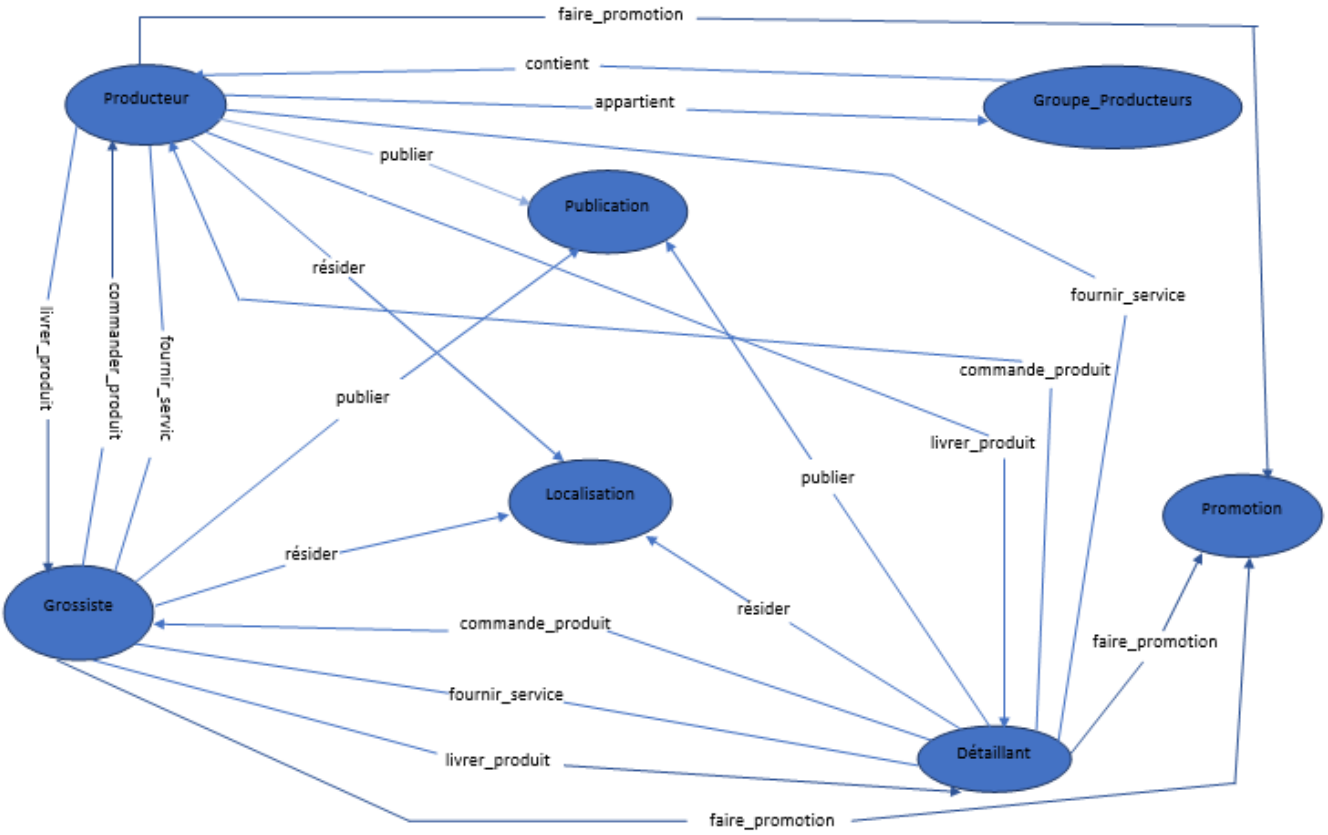

Figure 4.14 – Modèle de données

4.2.2 Schéma en étoile

Le schéma en étoile est simple permettant l'analyse de différentes mesures de la table des faits selon différentes dimensions qui représentent les axes d'analyse (voir Figure 4.15).

Notre schéma se compose de dimensions suivantes :

1- Date
2- Localisation
3- Producteur
4- Grossiste
5- Détaillant
6- Promotion

Et la table de fait : Forum_Distribution

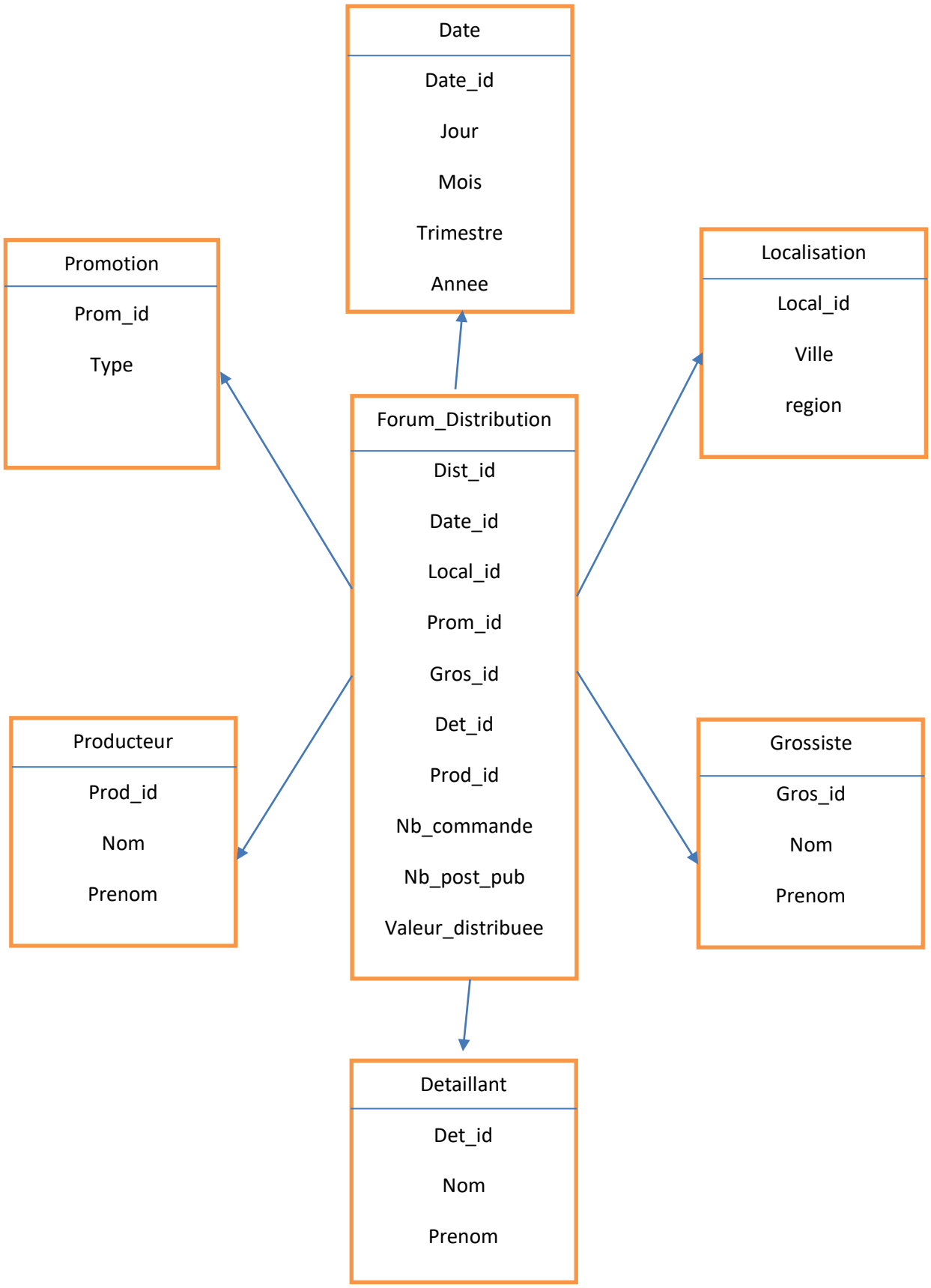

Figure 4.15–Schéma en étoile

4.2.3 Le modèle dimensionnel graphe

Le modèle dimensionnel graphe (GDM) est le modèle logique de l'entrepôt des données pour la base de données orientée graphe. Pour obtenir GDM, en utilisant des règles de transformation pour passer du modèle conceptuel (Schéma en étoile) au GDM. Il existe deux types de transformation : transformation normalisée et dénormalisée. Nous optons pour la transformation dénormalisée. Cette transformation assure le mappage au modèle NoSQL tout en mettant en évidence les concepts du Schéma Multidimensionnel mais sans détailler les hiérarchies [15].

4.2.4 Les règles de transformation Schéma en étoile vers GDM

Le modèle dimensionnel graphe (GDM) est le modèle logique de l'entrepôt des données pour la base de données orientée graphe. Pour obtenir GDM, nous utilisons les règles de transformation pour passer du modèle conceptuel (Schéma en étoile) au GDM [15].

Règle1 : Transformation des faits/mesures

Chaque fait est transformé en un nœud dont l'étiquette du nœud prend le type du concept du modèle multidimensionnel qui est 'Fait' puis on ajoute le nom du fait comme deuxième étiquette au même nœud. Chaque mesure est transformée par une propriété du nœud Fait.

Règle2 : Transformation Dimension/Paramètres

Chaque dimension est transformée en un nœud dont l'étiquette du nœud prend le nom du concept du modèle multidimensionnel (dans ce cas c'est la dimension). Ensuite, nous autorisons le nom de la dimension comme deuxième étiquette au même nœud. Ensuite, l'identifiant est transformé en une propriété dans le nœud. Enfin, tout attribut faible associé à l'identifiant est transformé en propriété dans le même nœud.

Chaque paramètre est transformé en une propriété dans le nœud (dimension). Ensuite, chaque attribut faible est représenté sous forme de propriété.

Règle3 : Transformation lien fait-dimension

Chaque lien entre fait et dimension est représenté comme une relation ayant pour noeud source le noeud modélisant le fait et pour noeud destination le noeud modélisant la dimension. La relation a pour nom 'lien fait-dimension'.

Le figure 4.16 représente le modèle dimensionnel graphe, les propriétés des nœuds sont les mêmes que dans le schéma en étoile.

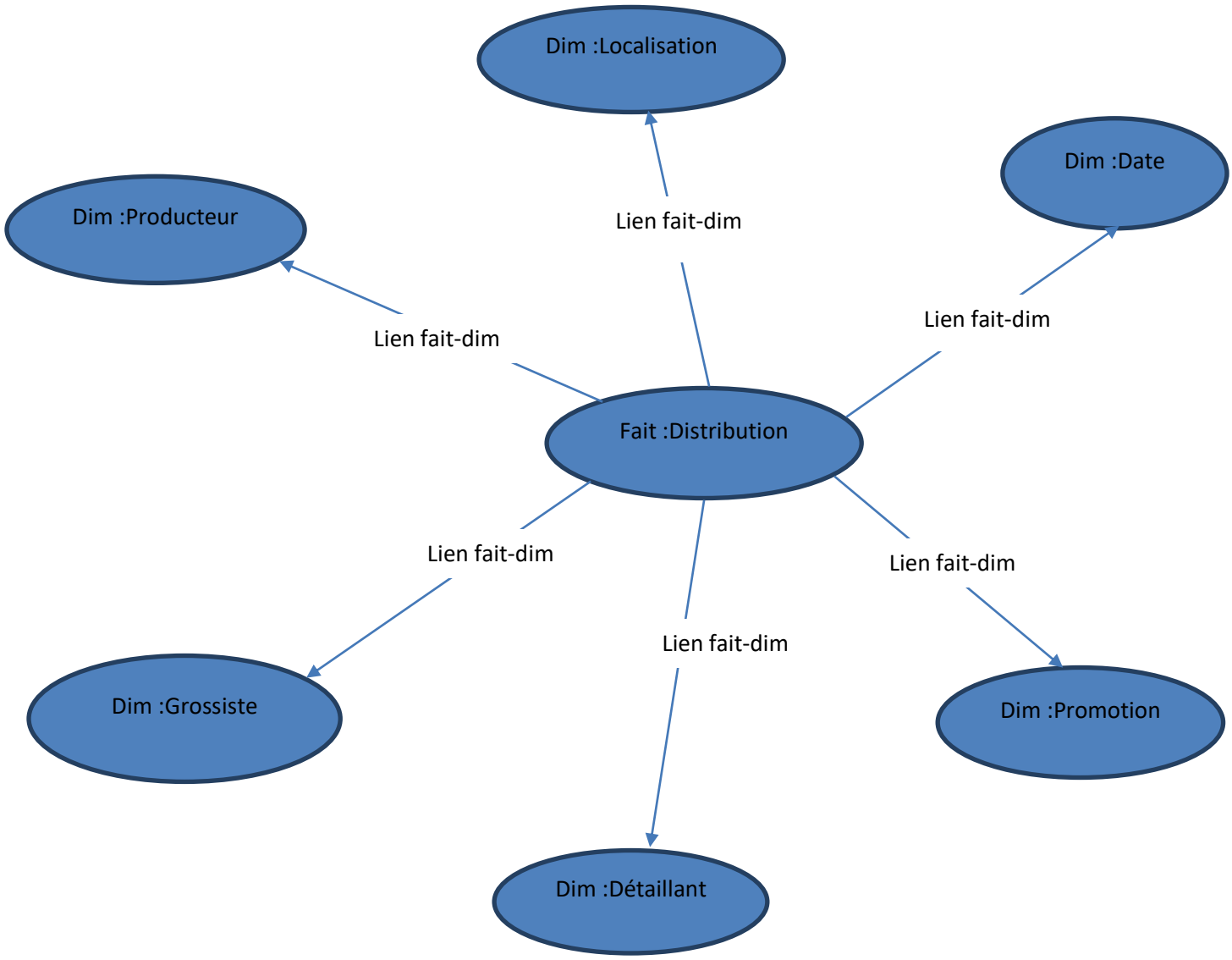

Figure 4.16 – Modèle Dimensionnel Graphe

4.3 Implémentation

Pour implémenter notre solution, nous avons utilisé un ensemble d'outils de développement et de visualisation des graphes. Pour tester notre solution nous avons utilisé un jeu de donnée relatives aux réseaux de distribution des produits alimentaires en Algérie. Ces outils sont énumérés ce qui suit :

4.3.1 Neo4j

Pour l'implémentation de notre base de données nous avons utilisé **Neo4j** (voir page 20), **Neo4j** permet de créer la base **DistributionPA** et la démarrer. Le Figure 4.17 montre l'interface de Neo4j ainsi que quelques outils de visualisation installée (Bloom, Charts, …).

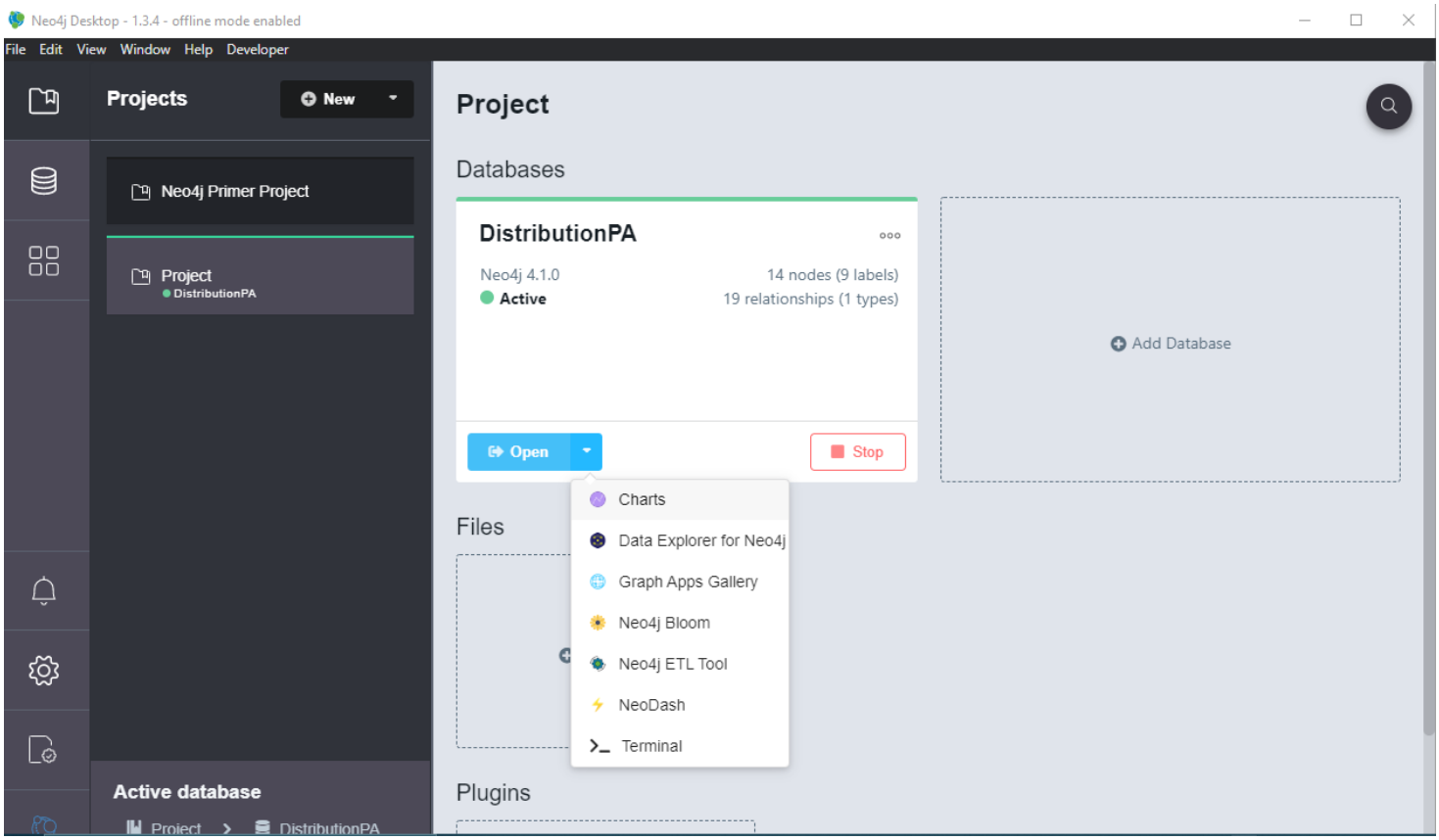

Figure 4.17 – Création de la base de données sur Neo4j

4.3.2 Browser

Outil de développement intégré au Neo4j qui permet de créer le graphe de la base en utilisant le langage Cypher, on l'accède avec le bouton Open dans l'interface Neo4j après le démarrage de notre base de données. Le Figure 4.18 montre le script Cypher pour créer une instance de graphe (les données). Le Figure 4.19 montre l'interface de Browser pour la création du graphe.

Cypher - Notepad

File Edit Format View Help

```
CREATE (n12:Dimension:Detaillant {detail_id: 01, nom: "ahmed", prenom: "abdellah"})
<-[:lien_Fait_Dim]-(n18:Fait:Forum_Distribution {nb_commande: 700, nb_post_publicitaire: 5,
valeur_distribuee: 2000, Dist_id: 03})-[:lien_Fait_Dim]->(:Dimension:Promotion {prom_id: 01,
categorie: "remise"})<-[:lien_Fait_Dim]-(n8:Fait:Forum_Distribution {nb_commande: 50,
nb_post_publicitaire: 5, valeur_distribuee: 1000, Dist_id: 01})-[:lien_Fait_Dim]
->(:Dimension:Date {date_id: 01012020, jour: 01, mois: 01, trimestre: 01, annee: 2020}),
(n12)<-[:lien_Fait_Dim]-(n8)-[:lien_Fait_Dim]->(:Dimension:Producteur {prod_id: 01, nom:
"mohamed", prenom: "kada"})<-[:lien_Fait_Dim]-(n15:Fait:Forum_Distribution {nb_commande: 70,
nb_post_publicitaire: 1, valeur_distribuee: 500, Dist_id: 02})-[:lien_Fait_Dim]->(n12),
(n15)-[:lien_Fait_Dim]->(n21:Dimension:Detaillant {detail_id: 2, nom: "nacer", prenom: "halim"})
<-[:lien_Fait_Dim]-(n8)-[:lien_Fait_Dim]->(:Dimension:Grossiste {gros_id: 01, nom: "ali", prenom:
"abdelkader"})<-[:lien_Fait_Dim]-(n15)-[:lien_Fait_Dim]->(:Dimension:Localisation {local_id: 01,
ville: "mostganem", region: "ouest"}),(n15)-[:lien_Fait_Dim]->(:Dimension:Date {date_id: 01012021,
jour: 05, mois: 04, trimestre: 02, annee: 2021})<-[:lien_Fait_Dim]-(n8),(:Dimension:Producteur {
prod_id: 02, nom: "abdelkader", prenom: "monir"})<-[:lien_Fait_Dim]-(n18)-[:lien_Fait_Dim]
->(:Dimension:Date {date_id: 01012020, jour: 01, mois: 01, trimestre: 01, annee: 2021}),
(n21)<-[:lien_Fait_Dim]-(n18)-[:lien_Fait_Dim]->(:Dimension:Grossiste {gros_id: 02, nom:
"alaaeddine", prenom: "madjed"})
```

Figure 4.18 – Script Cypher

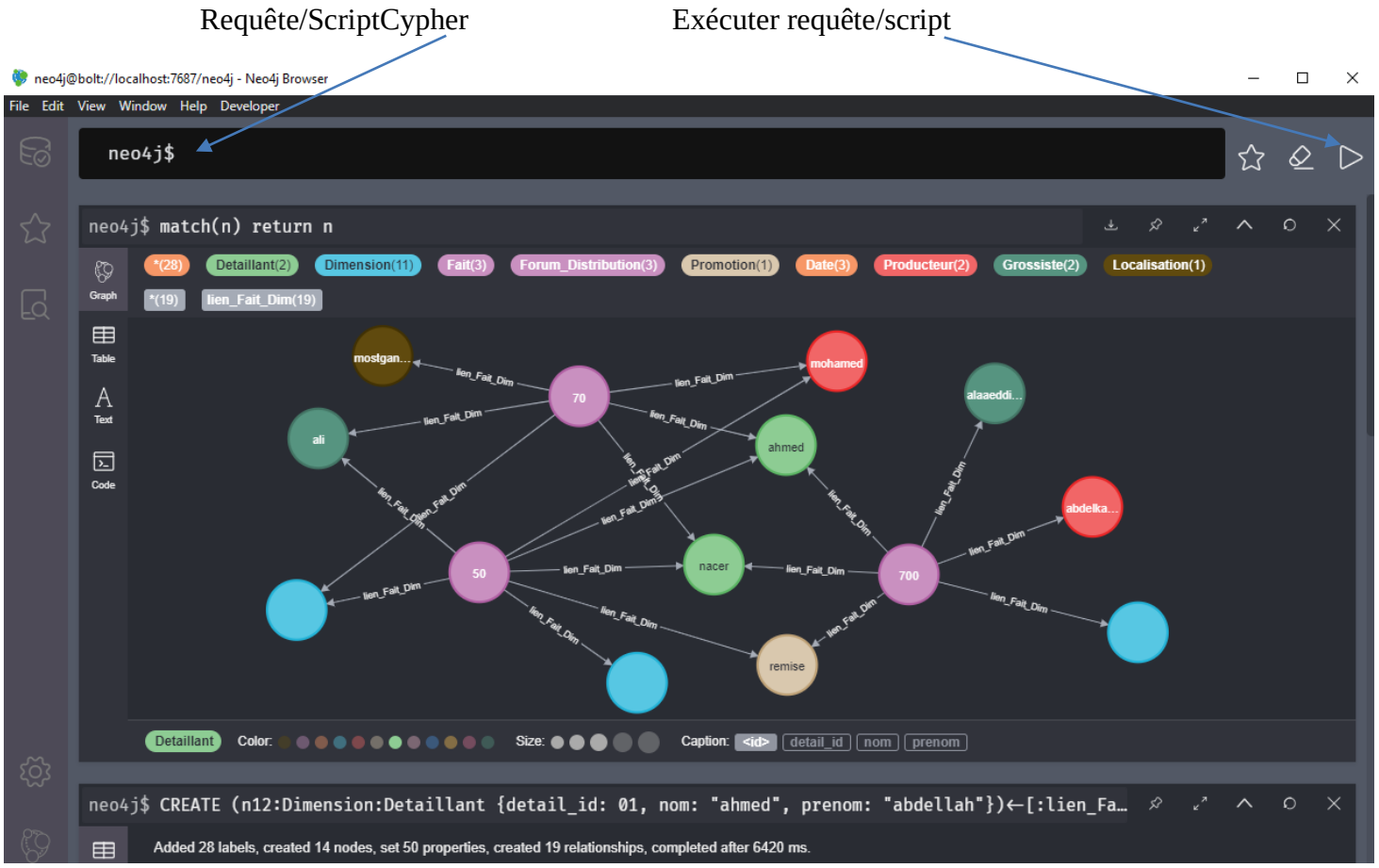

Figure 4.19 – Création de graphe sur Browser

4.3.3 Bloom

Outil qu'on installe sur Neo4j qui permet d'explorer et analyser le graphe de la base de données précédemment crée (voir Figure 4.20).

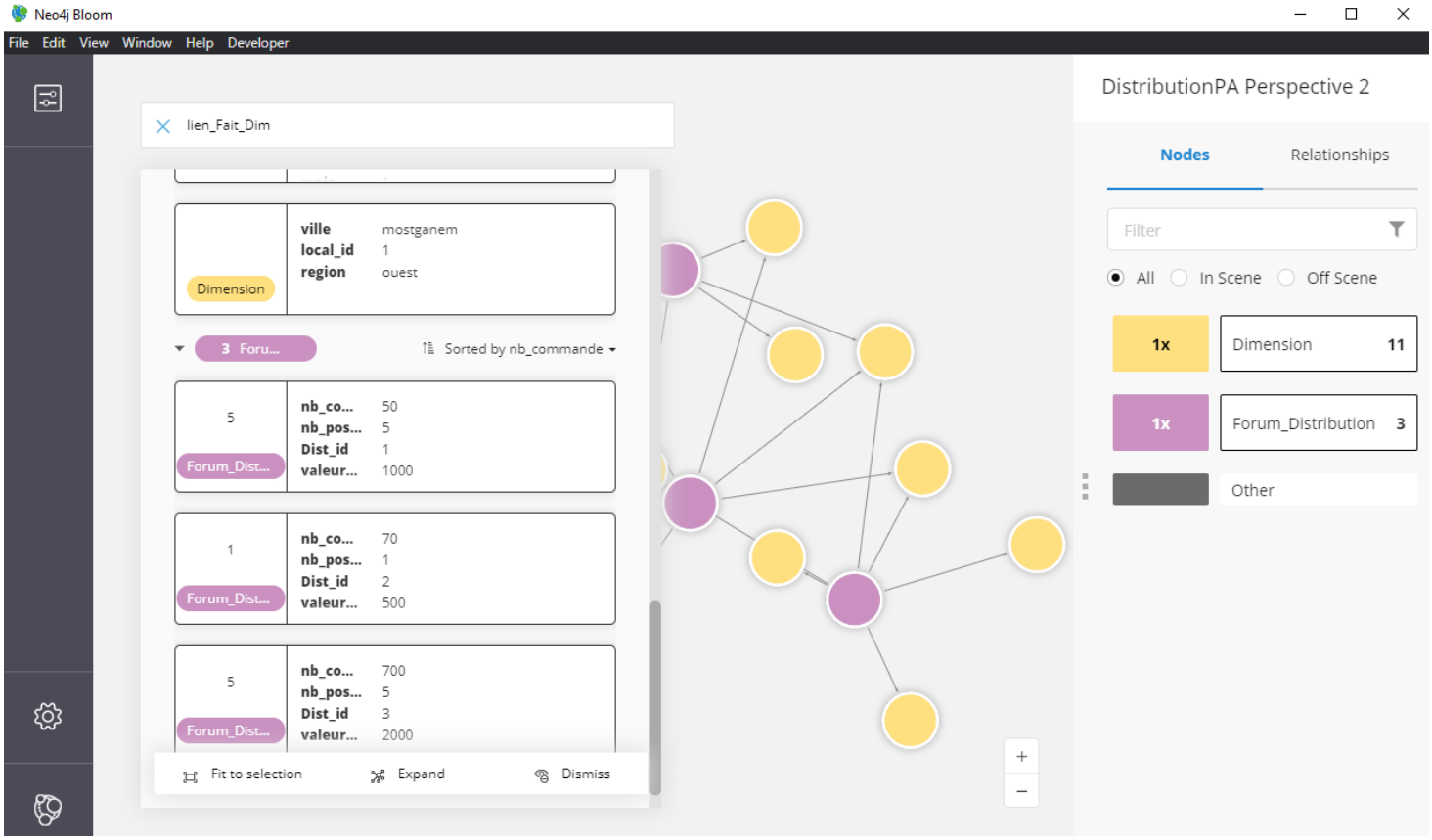

Figure 4.20 – Analyse de graphe à travers Bloom

4.3.4 Charts

Dans ce qui suit nous allons présenter un ensemble de requêtes analytiques d'aide à la décision sur **Charts** (outil que nous installons sur **Neo4j** qui permet de créer les rapports). Par exemples, la requête suivante **[MATCH(d:Dimension:Detaillant) RETURN d.nom AS nom,d.prenom AS prenom]** permet d'afficher tous les vendeurs détaillant (voir Figure 4.21 et 4.22) et la requête suivante **[MATCH(f :Fait :Forum_Distribution)-[lien_Fait_Dim]->(t :Dimension :Date) WHERE t.annee=2020 OR t.annee=2021 RETURN f.nb_commande AS nb_commande,t.jour AS jour,t.mois AS mois,t.annee AS annee]** permet d'afficher le nombre de commandes différentes passées durant les années 2020 et 2021 (voir Figure 4.23 et 4.24).

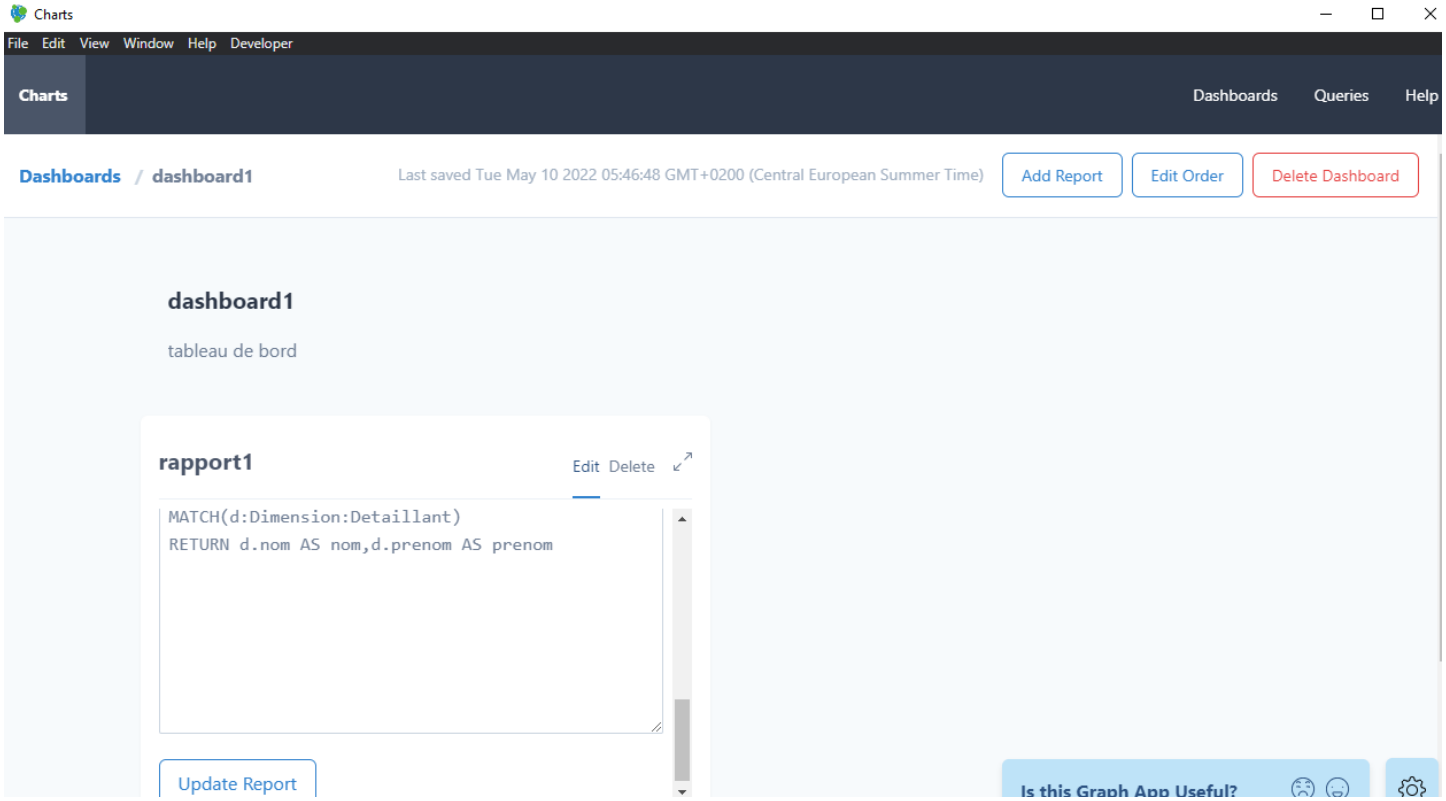

Figure 4.21 – Créer rapport1

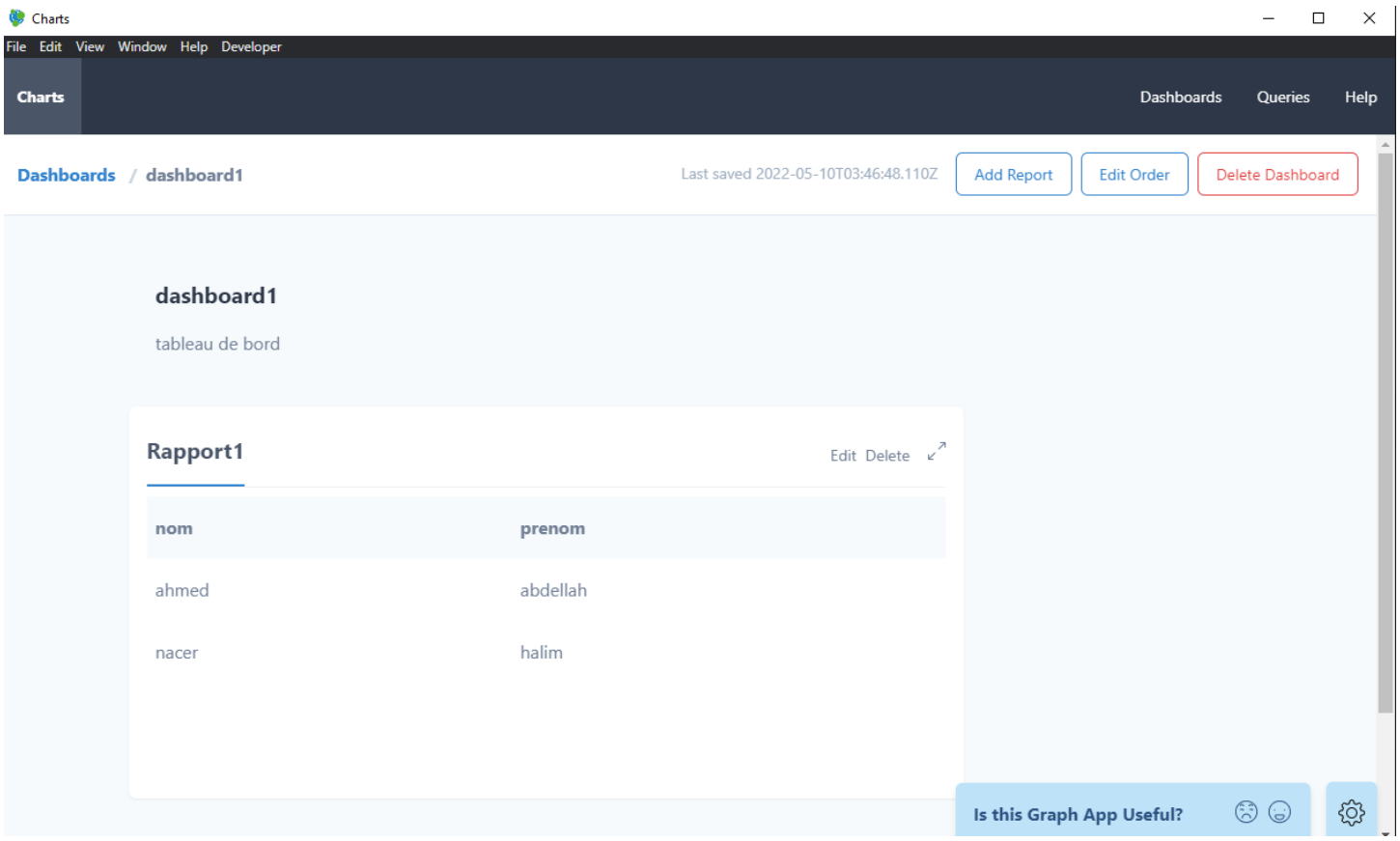

Figure 4.22–Afficher rapport1

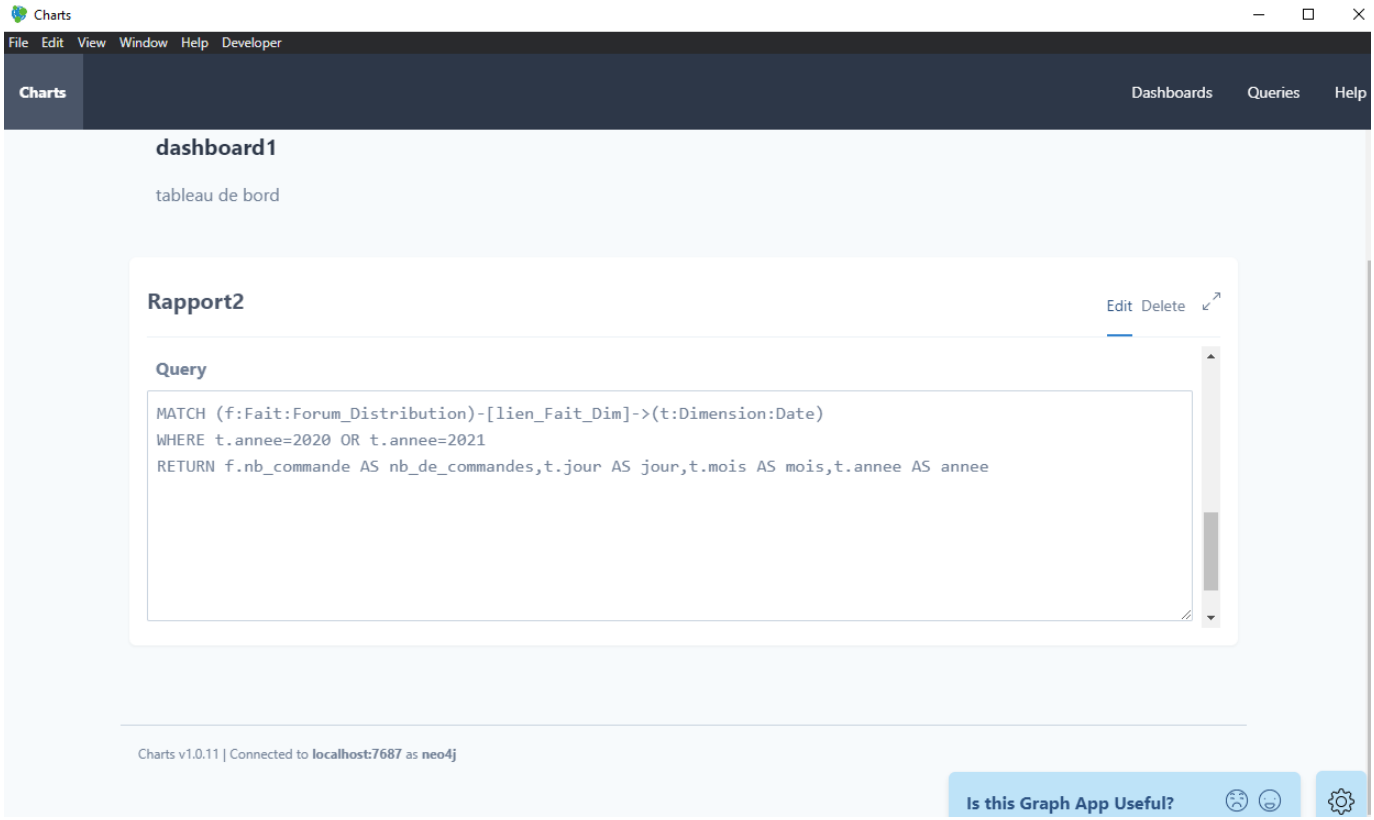

Figure 4.23 – Créer rapport2

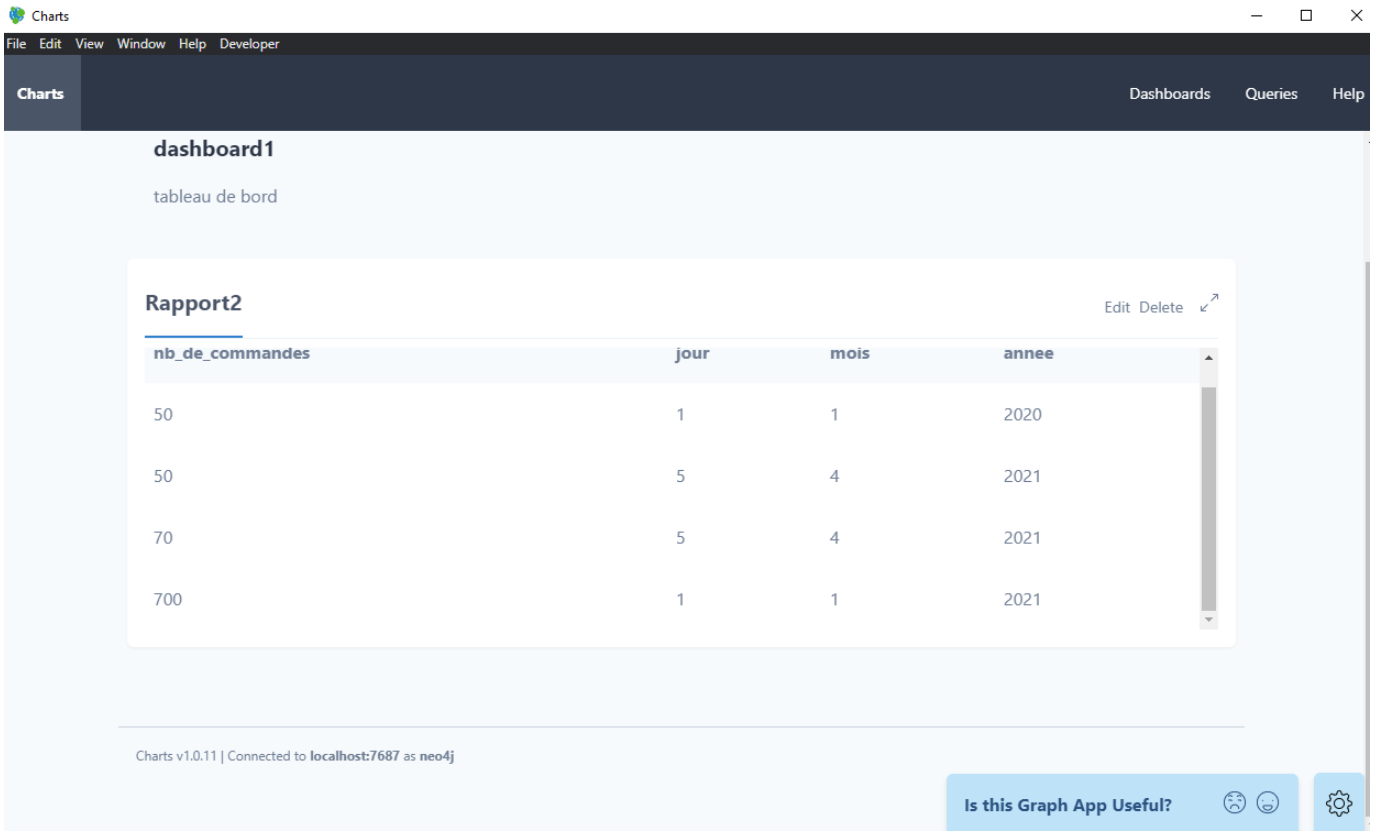

Figure 4.24 – Afficher rapport2

4.4 Conclusion

Dans ce chapitre nous avons modélisé le cas d'étude à savoir le réseau de distribution des produits alimentaires à un schéma en étoile puis nous faisons les transformations nécessaires pour aboutir à un modèle dimensionnel de graphe, enfin nous implémentons ce modèle avec des données relatives aux réseaux de distribution des produits alimentaires en Algérie sous forme d'une base de données décisionnelle orientée graphe sous Neo4j pour des explorations, analyses ou rapport avec les outils installés sur Neo4j(Browser, Bloom et Charts) utiles pour la prise des décisions.

Conclusion Générale

Les bases de données NoSQL constituent un domaine important de la recherche académique et des efforts de l'industrie différents. Ils sont utilisés pour maintenir, interroger et analyser de nombreux jeux de données dans différents domaines de l'industrie et des universités. De nombreuses bases de données NoSQL de différents types ont été développées, nous avons concentré notre étude sur les bases de données orientées graphes. Ils utilisent de nombreux modèles et représentations de données, ils sont construits à l'aide de choix de conception divers, et ils permettent un grand nombre de requêtes et de charges de travail.

Les bases de données orientées graphes deviennent courantes. Au fur et à mesure que les données deviennent connectées de manière plus compliquée et que la technologie des bases de données de graphes mûrit, leur utilisation augmentera. De nouveaux domaines d'application apparaissent, par ex. l'Internet des objets, ou plutôt l'Internet des objets connectés. Par rapport au SGBDR traditionnel, il est difficile pour les utilisateurs potentiels d'identifier les types particuliers de cas d'utilisation pour lesquels chaque produit est le plus adapté. Les performances varient considérablement d'un système de gestion de base de données orientée graphe à l'autre en fonction de la taille du graphe et du degré d'optimisation d'un outil donné pour une tâche particulière. Il semble que, en particulier pour les Big Graphs et Big Analytics, de nombreux résultats et conceptions antérieurs devront être reconsidérés et repensés dans les prochaines recherches et développements.

Dans ce travail, nous avons fournis une étude bibliographique sur les bases de données NoSQL on les compare avec les bases de données relationnelles (SQL) selon plusieurs critères, nous avons décrit ensuite les différents types des systèmes NoSQL utilisés on focalisons sur les bases données orientées graphes, en détaillons ses structures, ses cas d'utilisation et ses limites, enfin nous avons étudié le cas réseau de distribution des produits alimentaires pour modéliser et implémenter une solution NoSQL orientée graphe avec **Neo4j** et les outils de développement, d'exploration, d'analyses et de rapports associés pour l'aide à la prise de décision.

Bibliographie

[1] [A. CHIFU], NoSQL 101Premiers pas SQL vs. NoSQL, agcQui suis-je?

[2] [A. Oussous, F. Benjelloun, A. Ait Lahcen, S. Belfkih], Comparison and Classification of NoSQL Databases for Big Data Conference: International conference on Big Data, Cloud and Applications At: Tetuan, Morocco. Project: Big Data.May 2015.

[3] [A. Pore], NoSQL Data Architecture & Data Governance: Everything You Need to Know, February 16, 2018.

[4] [W. H. Inmon], Building the Data Warehouse Third Edition 2002.

[5] [O. Boussaid], Entrepôts de Données Avancés Partie 2 : Construction d'ED 2017

[6] [F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, R. E. Gruber], Bigtable: A Distributed Storage System for Structured Data 2006

[7] [A. W. Services], Amazon SimpleDB Developer Guide API Version 2009

[8] [M. Besta, R. Gerstenberger, E. Peter, M. Fischer, M. Podstawski, C. Barthels, G. Alonso, T. Hoefler], Demystifying Graph Databases: Analysis and Taxonomy of Data Organization, System Designs, and Graph Queries arXiv:1910.09017v5 [cs.DB], PRODYNA, ETH Zurich, Schweiz 16 Sep 2021.

[9] [I. Robinson, J. Webber & E. Eifrem], O˙REILLY Graph Databases 2nd Edition,2015.

[10] [ORACLE],17 Use Cases for Graph Databases and Graph Analytics, 2021.

[11] [J. Pokorný], Graph Databases: Their Power and Limitations
Department of Software Engineering, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic 2015.

[12] [K. Houda], Mémoire Magister Analyse de la fonction de distribution et son rôle dans le développement. Cas des produits des industries agroalimentaires à Béjaia 2012.

[13] [J. TARONDEAU], QUE SAIS - JE ? La distribution, l'Université de Paris X-Nanterre, Troisième édition mise à jour 1992.

[14] [F. Belhadj], Les circuits de distribution des produits alimentaires. Cas Pratique : Danone Djurdjura Algérie, Licence en sciences commerciales, option marketing, Université Abderrahmane Mira de Béjaia Algérie 2009

[15] [C. Mai], Graph NoSQL Data Warehouse Creation iiWAS ’20, November 30 - December 2, 2020, Thailand.