MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE
LA RECHERCHE SCIENTIFIQUE
UNIVERSITE ABDELHAMID IBN BADIS - MOSTAGANEM

Faculté des Sciences Exactes et de l'Informatique

Département de Mathématiques et d'Informatique

Filière : Informatique

MEMOIRE DE FIN D'ETUDES

Pour l'Obtention du Diplôme de Master en Informatique

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

THEME :

Optimisation du traitement centralisé des requêtes SPARQL

Etudiants : ARBAOUI MOHAMED RIAD

BOUKERROUCHA YACINE OUSSAMA

Encadrante: M^me^BENHAMED Siham

Année Universitaire 2018-2019

REMERCIEMENTS

À nos chers parents

Nous voulons remercier nos très chers parents, qui étaient toujours là pour nous « Vous avez tout sacrifié pour vos enfants n'épargnant ni santé ni efforts. Vous nous avez donné un merveilleux modèle de travail et de persévérance. Nous sommes redevables à Vous ».

À notre encadrant Mme Benhamed Siham

Nous vous adressons nos vifs remerciements. En tant que Directeur de mémoire. Nous avons eu la chance et le privilège de travailler sous votre direction. Nous avons apprécié votre aide et vos conseils si précieux. Veuillez trouver dans ce travail l'expression de notre estime et de notre considération. Que ce travail soit pour vous un témoignage profond gratitude.

Enfin, notre gratitude est exprimée envers :

Monsieur le Doyen de la Faculté de Mathématiques et informatique,
Monsieur le chef de Département Informatique ainsi qu'à l'ensemble du staff pédagogique et administratif de notre faculté.
Que ce modeste travail vous honore et vous témoigne notre reconnaissance.

Sommaire

Chapitre 2 : Systèmes de traitement de données Sémantiques

Chapitre 3 : Conception d'une approche pour le traitement de requête SPARQL

LISTE DE FIGURE

Liste des abréviations :

RDF:	Resource Description Framework.
RDFS:	Resource Description Framework Schema.
OWL:	Ontology Web Language.
XML:	eXtensible Markup Language.
SGML:	Standard Generalized Markup Language.
API :	Application Programming Interface.
W3C:	World Wide Web Consortium.
ISO:	International Organization for Standardization.
HTML:	Hyper Text Markup Language.
SPARQL:	Protocol and RDF Query Language.
Turtle:	Terse RDF Triple Language.
URI:	Uniform Resource Identifier.
URL:	Uniform Resource Locator.
IDE:	Integrated Development Environment.

Résumé :

La représentation sémantique des données dans le web à travers le RDF (Resource Description Framework) est en forte augmentation ces dernières années, ce qui a permis aux machines l'exploitation de la sémantique d'une manière formelle. Dans le web sémantique, la popularité de RDF est de plus en plus grande, ce qui a mené l'augmentation du volume globale de ces données avec un rythme sans précédent, et a la surcharge des ressources de mémorisation et de calcul qui est constatée. Dans de telles situations, les données générées ne peuvent pas être traitées par une seule machine, ce qui nécessite à notre avis, l'utilisation des techniques de partitionnement de graphe qui sont mises au point pour pouvoir interroger ce graphe et l'exploiter de manière fiable.

Dans notre travail, le traitement de ces ensembles de données RDF par les solutions classiques des systèmes centralisés engendre différents problèmes au niveau de l'accès é la donnée et sa récupération. L'objectif de notre travail est de proposer un mécanisme de traitement de grandes quantités de données sémantiques, en utilisant des mécanismes d'Optimisation de requêtes SPARQL et partitionner les requêtes pour obtenir un taux d'exécution optimale.

Introduction générale :

Actuellement, le Web est principalement syntaxique, dans le sens où la structure des documents en particulier et des ressources en général est bien définie, mais leurs contenus restent quasi inaccessibles aux traitements machines. Seuls les humains peuvent interpréter leurs contenus. Pour ce faire, la nouvelle génération de Web « le Web sémantique » a pour ambition de lever cette difficulté [1]. Dans ce cas, les ressources du Web seront plus aisément accessibles aussi bien par l'homme que par la machine, grâce à la représentation sémantique de leurs contenus.

Le Web sémantique représente une infrastructure qui permet l'utilisation de connaissances formalisées en plus du contenu informel actuel du Web. Cette nouvelle infrastructure permet d'abord de localiser, d'identifier, et de transformer des ressources de manière robuste et saine tout en renforçant l'esprit d'ouverture du Web avec sa diversité d'utilisateurs. Le web sémantique s'appuie sur un certain niveau de consensus portant, par exemple, sur les langages de représentation ou sur les ontologies utilisées [2]. Il contribue ainsi, à assurer l'automatisation des traitements et à garantir l'interopérabilité et les transformations entre les différents formalismes et les différentes ontologies. Il facilite la mise en œuvre de calculs et de raisonnements complexes tout en offrant des garanties supérieures sur leur validité. Il offre aussi, un ensemble de mécanisme permettant de qualifier les connaissances pour augmenter le niveau de confiance des utilisateurs.

Pour la représentation des données, le Web sémantique adopte le RDF [3]. Ce dernier, est le modèle de données le plus largement adopté sur Web sémantique. Il est basé sur la description des relations entre les sujets, Prédicats et objets (triplet) liés dans des graphes RDF. La nouvelle connaissance peut être déduite par le raisonnement tiré des graphes RDF avec le langage standard d'ontologie (OWL) ontologies et les langues basées sur les règles.

Le Web sémantique, concrètement, est d'abord une infrastructure qui permet l'utilisation de connaissances formalisées en plus du contenu informel actuel du Web. Cette nouvelle infrastructure permet d'abord de localiser, d'identifier et de transformer des ressources de manière robuste et saine tout en renforçant l'esprit d'ouverture du Web avec sa diversité d'utilisateurs. Le web sémantique s'appuie sur un certain niveau de consensus portant, par exemple, sur les langages de représentation ou sur les ontologies utilisées. Il contribue ainsi, à assurer l'automatisation des traitements et à garantir l'interopérabilité et les transformations

entre les différents formalismes et les différentes ontologies. Il facilite la mise en œuvre de calculs et de raisonnements complexes tout en offrant des garanties supérieures sur leur validité. Il offre un ensemble de mécanismes de protection (droits d'accès, d'utilisation et de reproduction), ainsi que des mécanismes permettant de qualifier les connaissances pour augmenter le niveau de confiance des utilisateurs.

Afin de traiter les données sémantiques pour accéder à l'information, il est plus facile aux machines de traiter les données structurées en graphe que de traiter un document. Le traitement des graphes est plus rapide et plus lisible par les machines que le traitement des documents numériques.

Les systèmes de traitement de données sémantiques permettent aux opérateurs la jointure adaptatifs proposés présentent le meilleur compromis entre le temps de réponse et le temps d'exécution, l'efficacité des opérateurs adaptatifs, plus rapide que les approches existantes.

Afin d'éclaircir l'état de l'art relative à ce domaine, ce rapport est articulé autour des deux chapitres suivants :

Chapitre 1 : Web Sémantique Dans ce chapitre nous présentons le web sémantique avec les différents formats de représentation des données sémantiques.

Chapitre 2 : Systèmes de traitement de données Sémantiques : Ce chapitre décrit le langage d'interrogation de données sémantiques SPARQL et il présente un ensemble de systèmes de traitement de données sémantiques afin de faire une comparaison entre les différentes approches utilisées dans ces systèmes. Ceci nous permet de bien comprendre le fonctionnement des systèmes de traitement de données sémantiques.

Chapitre 3 : Conception d'une approche pour le traitement de requête SPARQL : décrit le modèle qu'on propose pour perfectionner le traitement des données RDF ayant une taille massive.

Chapitre 4 : Implémentation du modèle de traitement des données sémantique : Présente la mise en œuvre des différents composants de notre modèle en utilisant le langage Python.

Enfin, nous concluons notre étude en donnant un résumé de notre travail, ainsi que, des perspectives futures à nos travaux.

1 Introduction :

De nos jours, la capacité d'échange discrète des informations et des données entre de différentes applications, des clients, et des partenaires via le Web devient de plus en plus vitale. Cependant la majorité des organisations utilisent une variété d'applications hétérogènes qui met en œuvre différentes formes et méthodes de stockage et d'accès aux données.

Le Web actuel est essentiellement syntaxique, dans le sens où la structure des documents en particulier et des ressources en général est bien définie, mais leurs contenus restent quasi inaccessibles aux traitements machines. Seuls les humains peuvent interpréter leurs contenus. Pour ce faire, la nouvelle génération du Web « le Web sémantique » a pour ambition de lever cette difficulté. Dans ce cas, les ressources du Web seront plus aisément accessibles aussi bien par l'homme que par la machine, grâce à la représentation sémantique de leurs contenus.

L'expression Web sémantique est proposée par Tim Berners-Lee au sein du W3C [4]. Ce dernier, fait d'abord référence à la vision du Web de demain comme un vaste espace d'échange de ressources entre les êtres humains et les machines, dans le but de permettre une exploitation, qualitativement supérieure, de grands volumes d'informations et de services variés. Espace virtuel, il devrait voir, à la différence du Web que nous connaissons aujourd'hui, les utilisateurs déchargés d'une bonne partie de leurs tâches de recherche, de construction et de combinaison des résultats, grâce aux capacités accrues des machines à accéder aux contenus des ressources et à effectuer des raisonnements sur ceux-ci.

Ainsi, les données sémantiques adoptent des formats très hétérogènes, autrement dit ces données utilisent des modèles différents pour la représentation de l'information et la relation entre les informations, ce qui impose l'intégration de nouveau modèle de représentation dans les applications web sémantique. Dans un tel contexte, le besoin d'intégration de la sémantique des données s'impose de plus en plus.

2 Définition du Web Sémantique :

Le terme Web sémantique a été inventé par Tim Berners-Lee qui signifie le traitement d'un réseau de données par des machines. Cette nouvelle version du Web a subi plusieurs critiques qui se douter de son utilité, elle demeure adoptée dans plusieurs secteurs tels que l'industrie, la biologie et la recherche en sciences humaines qui ont déjà prouvé la validité du concept original. Dans une publication scientifique de Berners-Lee, Hendler et Lassila en 2001, les auteurs décrivent l'évolution et l'amélioration attendue du Web existant en intégrant une

Web sémantique. En 2006, Berners-Lee et ses collègues ont déclaré que : « Cette idée simple reste en grande partie non réalisée ». En 2013, plus de quatre millions de domaines Web contenait le balisage Web sémantique [5].

Le Web sémantique offre une promesse passionnante d'un monde dans lequel les ordinateurs et les humains peuvent coopérer ensemble avec une compréhension commune de la signification des données, de même, il est considéré comme « Une extension du Web actuel par les normes du World Wide Web Consortium (W3C), dans lequel l'information reçoit une signification bien définie, permettant ainsi aux ordinateurs et aux personnes de travailler en coopération » [6].

Afin d'éclaircir le principe du Web sémantique, on propose l'exemple suivant : La faculté ex 'INES' est située à Mostaganem et Mostaganem est située en Algérie. Le but du Web sémantique est de donner la possibilité d'extraire automatiquement qu'INES est située en Algérie comme illustré dans la figure 1 ci-dessous.

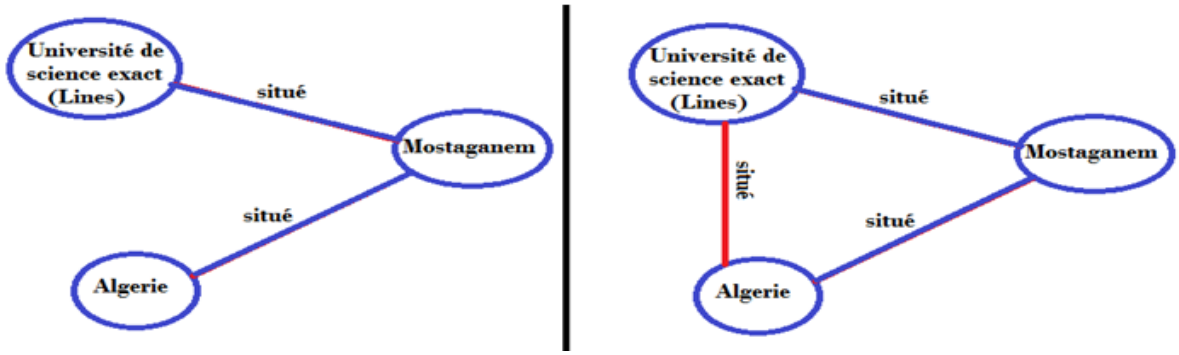


Figure 1: Exemple explicatif du web sémantique.

Le but du Web sémantique est de favoriser les formats de représentation de données et les protocoles d'échange communs sur le Web. Parmi les formats de représentation des données sémantiques, on trouve le RDF « Resource Description Framework » qui est le plus fondamental.

Selon le W3C, « Le Web sémantique fournit un cadre commun qui permet de partager et de réutiliser les données à travers des applications, des entreprises et des communautés » [4]. Toutefois, au cours de la décennie écoulée depuis l'entrée en vigueur du terme, les applications du Web sémantiques ont pris beaucoup de temps pour sortir des laboratoires de recherche.

Dans un future proche, le Web sémantique pourra devenir un outil de gestion avancée de l'information sur Internet, ce qui nous permettra de lancer des requêtes plutôt que de parcourir les documents, de connaitre les faits existants et d'identifier l'incohérence.

Le célèbre diagramme "Web Layer", montré dans la figure 2, donne une vue d'ensemble de la hiérarchie des principaux langages, chacun exploitant les caractéristiques des niveaux dessous. Elle renforce également le fait que le Web sémantique n'est pas séparé de la bande existante, mais elle est en effet une extension de ses capacités.

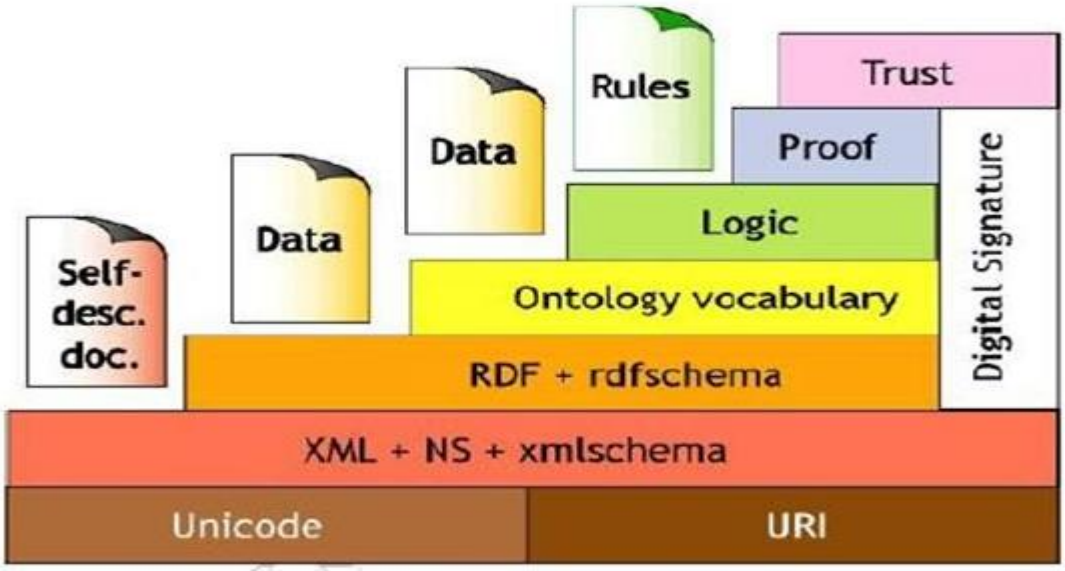


Figure 2: Les couches du Web sémantique [7].

3 Les Technologies de Base du Web Sémantique :

3.1 RDF :

3.1.1 Définition :

Le RDF (Resource Description Framework) est un modèle de graphe destiné à décrire de façon formelle les ressources Web et leurs métadonnées, de façon à permettre le traitement automatique de telles descriptions. Développé par le W3C, RDF est le langage de base du Web sémantique. L'une des syntaxes de ce langage est RDF/XML. D'autres syntaxes de RDF sont apparues ensuite, cherchant à rendre la lecture plus compréhensible [8].

RDF est un modèle standard pour l'échange de données sur le Web. RDF dispose de fonctionnalités qui facilitent la fusion des données, même si les schémas sous-jacents sont différés, et il supporte spécifiquement l'évolution des schémas au fil du temps sans exiger que tous les utilisateurs de données soient modifiés.

RDF étend la structure de liaison du Web pour utiliser les URI afin de nommer la relation entre les ressources ainsi que les deux extrémités du lien (ce qui est habituellement appelé « triplet »). En utilisant ce modèle simple, il permet de combiner, d'exposer et de partager des données structurées et semi-structurées entre différentes applications. Cette structure de liaison forme un graphe dirigé et marqué, où les arêtes représentent le lien nommé entre deux ressources, représentées par les nœuds graphiques (figure 3). Cette vue graphique est le modèle mental le plus simple possible pour RDF. Ceci est souvent utilisé dans des explications visuelles puisqu'il est facile à comprendre par la machine [4].

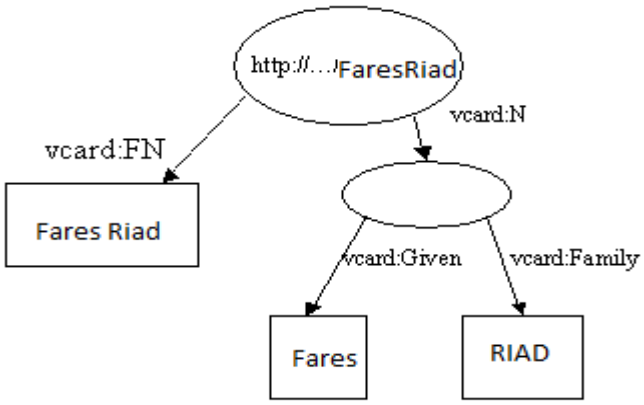


Figure 3: Exemple de graphe RDF décrivant Fares Riad.

3.1.2 Les formats de représentation de RDF :

Le format de représentation de RDF est de type XML. Ce dernier permet de stocker des données de manière relationnelle grâce à sa structure graphique. Il permet à n'importe quel programme de connaître les relations entre telle ou telle donnée.

Les formats sont normalisés par le W3C, d'autres syntaxes peuvent être intégrées dans les registres de l'analyseur et de l'auteur. Parmi les syntaxes on trouve : Tutrule, RDF/XML, N-Triples, JSON-LD, RDF/JSON, TriG, TriX, RDF Binary, et NQuads.

Un document structuré en RDF est un ensemble de triplets. Ce dernier, est une association (sujet, prédicat, objet) :

- Le « sujet » représente la ressource à décrire.
- Le « prédicat » représente un type de propriété applicable à cette ressource.
- L'« objet » représente une donnée ou une autre ressource : c'est la valeur de la propriété.

Dans la figure 4, nous présentons les différentes formes de la représentation d'un triplet RDF.

En allant de haut en bas, on voit : la notation graphique dans laquelle le sujet et l'objet sont liés par une flèche marquée. Dans le format triplet, chaque composant est un URI ou une valeur de données dans le cas d'un objet. On présente ainsi, la forme relationnelle qui serait l'équivalent dans Prolog, le format d'échange RDF / XML, et le format Turtle qui, dans le cas d'un triplet unique, diffère seulement de la notation triplet par son utilisation du préfixe d'espace de noms dans les URI.

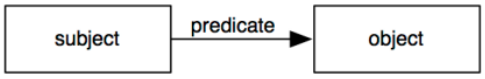
Forme graphique	
Triplet	Objet sujet prédicat
Forme relationnelle	Prédicat (sujet, objet)
RDF/XML	<rdf:Descriptionrdf:about="subject"> <ex:predicate> <rdf:Descriptionrdf:about="object"/> </ex:predicate> </rdf:Description>
Tutrle	Objet ex: objet prédicat.

Figure 4: Représentation de triplet RDF sous différentes formes.

3.2 RDFS :

3.2.1 Définition de RDFS :

RDF Schéma ou RDFS « Resource Description Framework » est un langage extensible de représentation des connaissances [4]. Il appartient à la famille des langages du Web sémantique publiés par le W3C. RDFS fournit des éléments de base pour la définition d'ontologies ou de vocabulaires destinés à structurer des ressources RDF notamment sous la forme d'un triplestore[1], ce qui permet, grâce au langage de requête SPARQL[2]de les atteindre à travers le Web.

[1]Une base de données spécialement conçue pour le stockage et la récupération de données RDF.

[2]Un langage de requête et un protocole qui permet de rechercher, d'ajouter, de modifier ou de supprimer des données RDF disponibles à travers Internet.

La figure 5 montre la relation entre RDF et RDFS. Le RDFS permet d'exprimer et d'échanger des méta-données, et les mécanismes de requête et d'inférence disponibles pour le formalisme des graphes conceptuels.

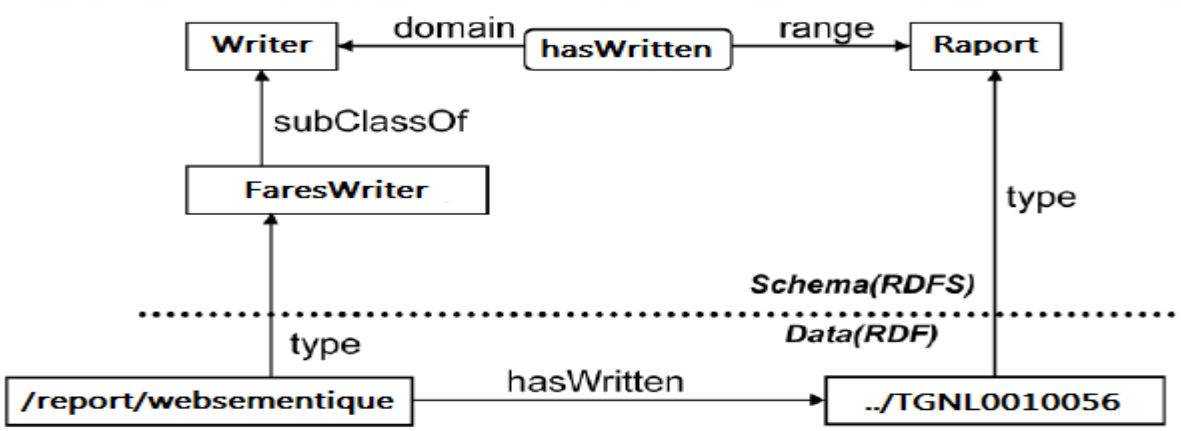


Figure 5: Relation entre RDF et RDFS.

3.3 L'ontologie :

3.3.1 Définition de l'ontologie :

Le terme est repris en informatique et en science de l'information, où une ontologie est l'ensemble structuré des termes et concepts représentant le sens d'un champ d'informations, que ce soit par les métadonnées d'un espace de noms, ou par les éléments d'un domaine de connaissances [9].

L'ontologie constitue en soi un modèle de données représentatif d'un ensemble de concepts dans un domaine, ainsi que des relations entre ces concepts. Elle est employée pour raisonner à propos des objets du domaine concerné. Plus simplement, on peut aussi dire que l'« ontologie est aux données ce que la grammaire est au langage».

Le langage d'ontologie le plus largement utilisé est le « WebOntologyLanguage», Qui a curieusement l'acronyme «OWL».

3.4 OWL :

3.4.1 Définition d'OWL :

OWL (ou Ontology Web Language) est une recommandation émanant du W3C permettant de définir et d'instancier des ontologies [4]. Ce langage constitue une extension des langages RDF et RDFS et permet de combler leur manque d'expressivité. OWL 'introduit notamment des notions de classes ou de propriétés équivalentes, d'égalités entre instances, de propriétés symétriques, de restrictions de valeurs.

3.4.2 Les sous langage d'OWL :

OWL « Web OntologyLanguage comprend trois sous langages [10] :

- OWL Lite, tout d'abord, est le moins expressif des trois. Ce langage est particulièrement adapté aux personnes souhaitant bénéficier d'une expressivité plus importante que RDF/RDFS tout en conservant une certaine simplicité d'utilisation.
- OWL DL est un sous-langage offrant une expressivité maximale. Toutes les propriétés OWL sont ainsi présentes dans ce langage. Toutefois, un ensemble de contraintes a été fixé afin de garantir deux propriétés :
  - OWL DL peut résoudre l'ensemble des problèmes d'inférences (complétude).
  - OWL DL assure que ces problèmes peuvent être résolus en un temps fini (décidabilité).
- OWL Full, un sous-langage permettant d'avoir une expressivité maximale et une grande liberté dans la conception des ontologies. OWL Full lève la plupart des verrous fixés dans OWL DL. Cette liberté syntaxique a toutefois un prix : il est impossible de garantir que les problèmes d'inférences concernant une ontologie utilisant OWL Full pourront être résolus en un temps fini. Parmi les libertés offertes par OWL Full, on trouve notamment la possibilité d'utiliser une classe comme une instance ou encore la possibilité d'intégrer plus facilement des éléments de RDF/RDFS dans l'ontologie.

4 Les Formes de Représentation des Données Sémantiques :

A ce jour, de nombreux systèmes de compréhension du langage ont été réalisés, illustrant à leur façon, différentes ambitions. Dans la mesure où ces systèmes manipulent du web, ils contiennent quelque part un composant pour une « représentation sémantique ». Dans ce qui va suivre nous allons détailler les deux formes de représentation des données qui nous intéressent dans le web sémantique : l'XML et les graphes.

4.1 XML :

Le langage XML est un langage de balisage extensible [10]. XML fournit une syntaxe unique pour les données sans adhérence avec une plateforme logicielle particulière. XML présente comme HTML la particularité de véhiculer les données et leurs descriptions. On rappelle que nous adoptons la définition suivante donnée par les W3C.

4.1.1 Définition :

XML « eXtensible Markup Language » est un langage simple et très flexible, dérivé de SGML « Standard Generalized Markup Language, ISO 8879 », pour le formatage de texte [11]. Originalement, L'XML est conçu pour être à la hauteur des défis que présentent les publications électroniques à grande échelle, XML également un rôle de plus en plus important dans l'échange d'une grande variété sur le Web et ailleurs.

4.1.2 Les principales caractéristiques de XML :

De nos jours, l'XML est devenu omniprésent dans le monde de l'informatique. De nombreux standards sont apparus et permettent à des applications différentes de stocker mais surtout de partager des documents.

- XML est un langage de marquage (balisage) comme HTML.
- XML a été conçu comme véhicule de données et ne s'intéresse pas à leur présentation.
- Les balises XML ne sont pas prédéfinies : l'utilisateur définit ses propres balises.
- XML est conçu pour être autodescriptif.
- XML est une recommandation du W3C.

XML peut être présenté comme un « SGML réformé et modernisé », ou comme un « HTML rendu générique et extensible », apte à supporter les nouvelles applications Web.

Nous présentant dans la figure ci-dessous, un exemple document XML qui permet de décrire une personne :

```
< ?xml version="1.0" encoding="UTF-8"?>
<Personne>
        <Nom>Riad</Nom>
        <Prénom>Fares</Prénom>
        <Naissance>
                <Lieu>
                    <Ville>Mostaganem</Ville>
                    <Pays>Algerie</Pays>
                </Lieu>
        </Naissance>
</Personne >
```

Figure 6 : Exemple de fichier XML simple d'un document RDF.

4.2 Les graphes :

4.2.1 Définition :

Un graphe est un ensemble de points nommés nœuds (parfois sommets ou cellules) reliés par des traits (segments) ou flèches nommées arêtes (ou liens ou arcs). L'ensemble des arêtes entre nœuds forme une figure similaire à un réseau. Il existe deux types de graphe : orientés et non orientés. Le web sémantique utilise les graphes orientés pour la représentation de ses données. Un graphe orienté est un graphe dirigé sans paire symétrique de bords dirigés.

4.2.2 Les graphes sémantiques :

Un graphe sémantique est tout simplement un graphe qui schématise les triplets RDF. Afin d'éclaircir le principe des graphes sémantiques, Nous présentons dans la figue 7 un graphe sémantique qui est constitué de deux nœuds, " Mostaganem " et "Algérie", reliés par un arc nommé "est_situé_en". En d'autres termes :

- Mostaganem : est-ce qu'on appelle une ressource, ou encore un sujet une source ?
- Est_situé_en : est-ce qu'on appelle un prédicat ou encore une propriété ?

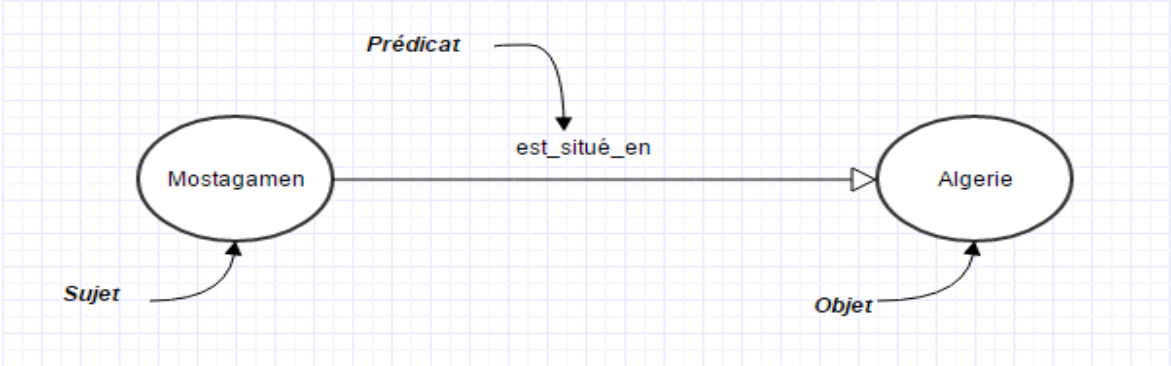


Figure 7: Graphe RDF orienté.

4.2.3 Les avantages des graphes :

Dans le Web sémantique en utiliser la représentation graphique (les graphes) grâce aux graphes, il est devenu possible aux utilisateurs de saisir visuellement le message qui doit être transmis. Ainsi, les graphes sont particulièrement utiles s'il y avait beaucoup de détails qui nécessiteraient trop de temps à expliquer et qu'il faut compacter l'information sur une synthèse visuelle. Connaître votre public aide à créer le bon type de graphiques et de graphiques à utiliser.

5 Les Systèmes de gestion de Bases de données sémantiques :

5.1 Définition :

Les systèmes de gestion de bases de données représentent un ensemble de programmes qui permettent d'assurer la structuration, le stockage, la maintenance, la mise à jour et la recherche des données d'une base avec des interfaces nécessaires aux différentes formes d'utilisation de la base. Les propriétés des SGBD sont les suivantes :

- Usage multiple des données.
- Accès facile, rapide, protégé, souple, puissant.
- Coût réduit de stockage, de mise à jour et de saisie.
- Disponibilité, exactitude, cohérence et protection des données non redondance.
- Évolution aisée et protection de l'investissement de programmation.
- Indépendance des données et des programmes.
- Conception a priori.

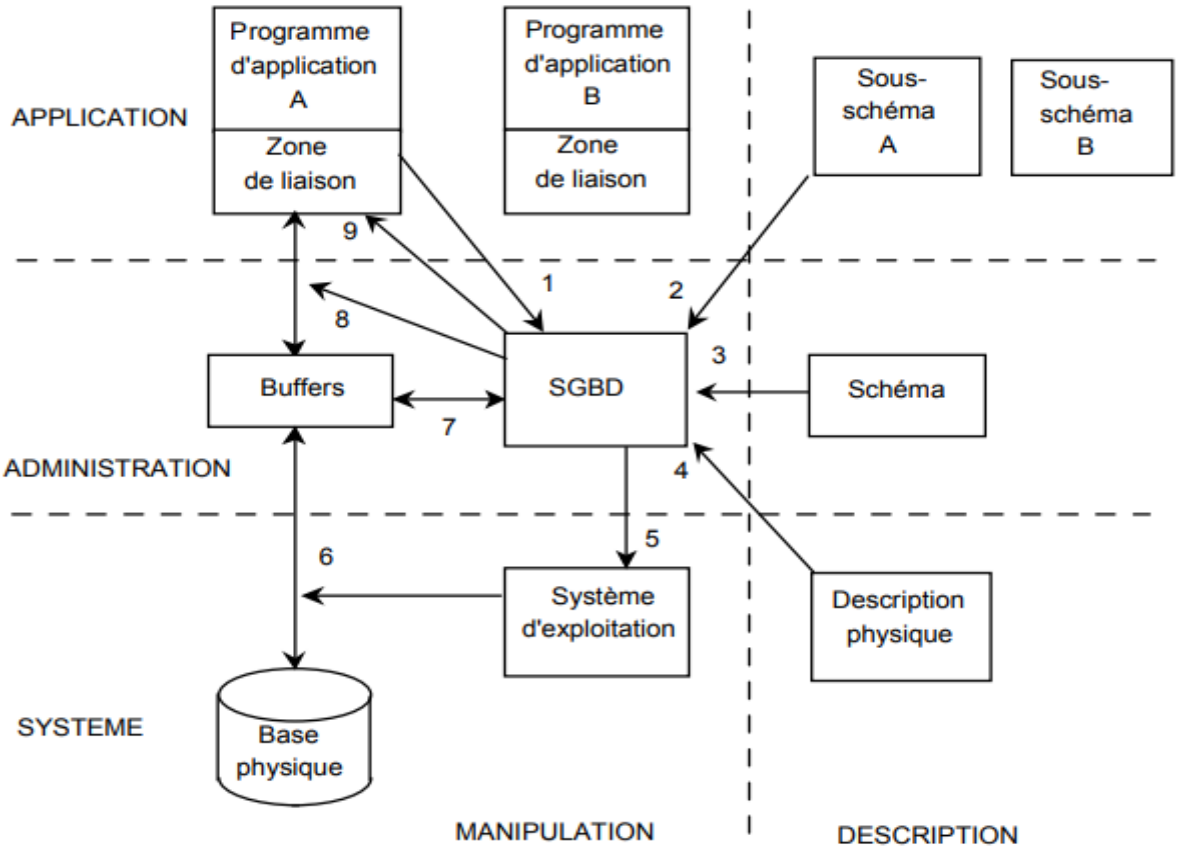


Figure 8: Fonctionnement d'un SGBD [20]

Le programme d'application A émet une demande de lecture à l'intention du SGBD et consulte le sous-schéma relatif à A pour obtenir la description logique de ses données.

Le SGBD consulte le schéma et détermine la structure logique des données à extraire et examine la description physique de la base et détermine les enregistrements physiques à lire ensuite le SGBD lance une commande au système d'exploitation pour provoquer la lecture de l'enregistrement physique.

Le système d'exploitation provoque le transfert de l'enregistrement entre la base physique et les buffers du SGBD et à partir du sous-schéma A, extrait les données à communiquer au programme d'application A.

Le SGBD provoque le transfert des données dans la zone de liaison de A ensuite il retourne au programme d'application les informations d'état relatives à l'échange (en particulier les codes des erreurs éventuelles).

Les SGBD assurent : la rapidité de création d'IHM et la mise en œuvre, en plus la facilité de maintenance ou reprise.

5.2 La base de données No SQL :

5.2.1 Définition :

Appelée également « Not Only SQL » (pas seulement SQL), la base de données NoSQL est une approche de la conception des bases et de leur administration particulièrement utile pour de très grands ensembles de données distribuées. NoSQL est particulièrement utile lorsqu'une entreprise doit accéder, à des fins d'analyse, à de grandes quantités de données non structurées ou de données stockées à distance sur plusieurs serveurs virtuels du Cloud.En fait, le terme base NoSQL définit une nouvelle génération de produits qui ne suivent pas le modèle relationnel. Mais l'architecture de ces produits varie beaucoup entre eux. [13]

5.2.2 Type de BD No SQL :

Les bases NoSQL peuvent être réparties en quatre grandes familles [14] :

A. Clé–valeur :

Les données sont représentées par un couple clé–valeur, la valeur pouvant être une simple chaîne de caractères ou un objet. Ce modèle n'a pas la complexité du paramétrage à l'origine des bases de données transactionnelles, il offre une forte évolutivité grâce à l'absence de structure ou de typage.

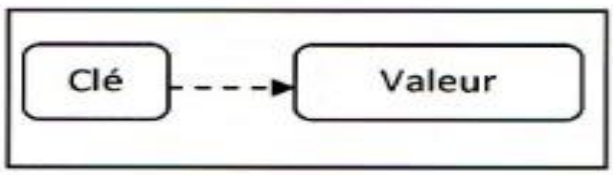


Figure 9: BD Clé-valeur [14]

B. Orientée colonnes :

Une base de données orientée colonnes est une base de données qui stocke les données par colonne et non par ligne. L'orientation colonne permet d'ajouter des colonnes plus facilement aux tables (les lignes n'ont pas besoin d'être redimensionnées).Elle permet de plus une compression par colonne, efficace lorsque les données de la colonne se ressemblent.

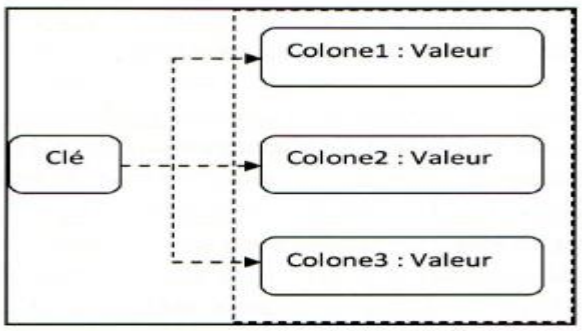


Figure 10: BD Orientée colonnes [14]

C. Orientée documents :

Cette base remplace la valeur par un document de type JSON ou XML. Une seule clé permet ainsi de récupérer l'ensemble des informations de manière hiérarchique, ce qui imposerait plusieurs jointures dans le monde SQL.

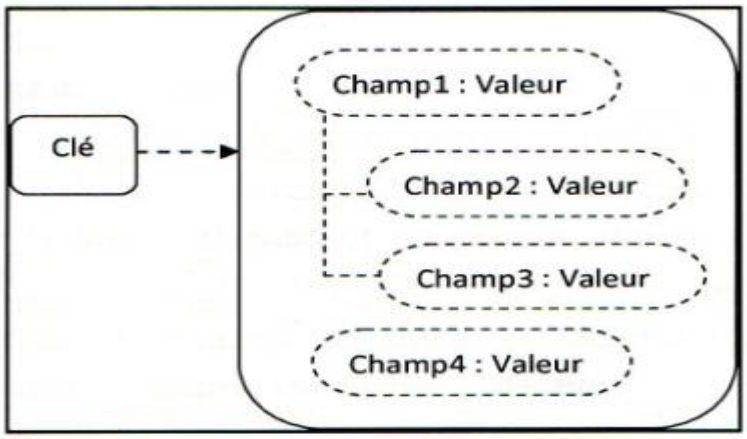


Figure 11: BD Orientée document [14]

D. Orientée graphe :

Ce modèle de représentation des données se base sur la théorie des graphes, à savoir la notion de nœuds, de relations et de propriétés, ce qui facilite la représentation du monde réel, par exemple dans les réseaux sociaux.

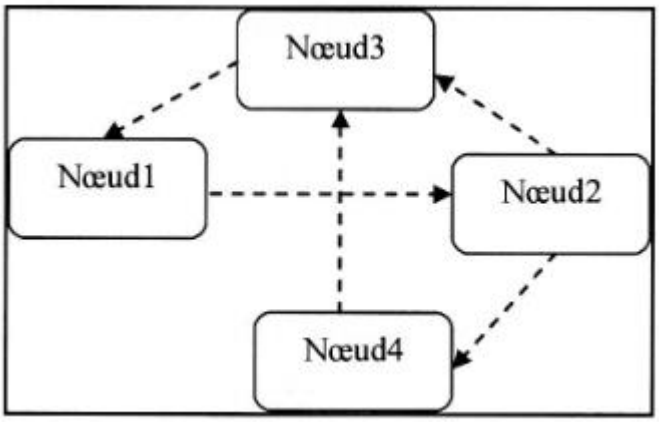


Figure 12: BD Orientée graphe [14]

6 BIG DATA:

6.1 Définition :

Littéralement, ces termes signifient mégadonnées, grosses données ou encore données massives. Ils désignent un ensemble très volumineux de données qu'aucun outil classique de gestion de base de données ou de gestion de l'information ne peut vraiment travailler. En effet, il existe environ 2,5 trillions d'octets de données tous les jours. Ce sont les informations provenant de partout : messages que nous nous envoyons, vidéos que nous publions, informations climatiques, signaux GPS, enregistrements transactionnels d'achats en ligne et bien d'autres encore. Ces données sont baptisées Big Data ou volumes massifs de données. Les géants du Web, au premier rang desquels Yahoo (mais aussi Facebook et Google), ont été les tous premiers à déployer ce type de technologie [15]

Le big data rassemble tous les types des bases de données tels que les données RDF qui sont volumineuses, ce qui nécessite un mode de stockage particulier, pour ce nouveau mode de stockage, il faut respecter la règle des 3V [16]

- Le volume de données de plus en massif.
- La variété de ces données qui peuvent être brutes, non structurées ou semi-structurées.
- La vélocité qui désigne le fait que ces données sont produites, récoltées et analysées en temps réel.

6.2 Utilisation :

Les Big Data sont souvent liés à des applications d'analyses permettant le traitement des données contenues dans les Big Data. En analysant ces données il devient possible de déterminer les tendances religieuses, culturelles, politiques afin anticiper des actions possibles. On peut trouver l'utilisation des Big Data dans la vente en ligne (améliorer l'expérience du client, optimiser ses processus, sa performance opérationnelle, méthodes de ventes), par la nsa dans la médecine analytique afin de visualiser l'activité cérébrale.

6.3 Problématique de gestion des données massives (Big data) :

Les Big data soulèvent des défis à toutes les étapes de la gestion des données : le stockage, le traitement, l'analyse...etc. Les Big data ne sont pas simplement de grands volumes de données, elles se déplacent rapidement, sont difficiles à valider et à valoriser [17]. Le stockage du Big data est une chose, son traitement est une autre. Maintenant Il faut alors s'adapter et tenter de nouvelles méthodes de traitement. La question n'est donc plus d'identifier quelles données stocker, mais, qu'est-ce qu'on peut faire avec ces données ? Cette masse de données qui arrive en flot continu, provenant des sources très diverses, son traitement pose des problèmes en particulier dans l'extraction de connaissances [18]

6.4 La sémantique dans le Big data :

Le Big data se réfère ainsi à ce qui peut être accompli à grande échelle et ne peut pas l'être à une échelle plus petite. Le Big data s'appuie sur le développement d'applications à visée analytique, qui traitent les données pour en extraire de la sémantique. Une chaîne de transformations bien connue existe dans le domaine du management de l'information, c'est la chaîne [22] : « donnée → information → connaissance → sagesse » que représente le modèle DIKW (Data, Information, Knowledge, Wisdom). La représentation graphique la plus populaire pour DIKW est une pyramide, avec les données à la base et la sagesse à son sommet. Cette représentation suppose implicitement que les éléments les plus hauts dans la pyramide nécessitent les éléments inférieurs pour être définis, et qu'ils peuvent être atteints après un processus de transformation des éléments inférieurs. Le modèle DIKW est alors une chaîne où l'information est le résultat du traitement des données, la connaissance est le résultat du traitement de l'information, et la sagesse est le résultat du traitement de la connaissance.

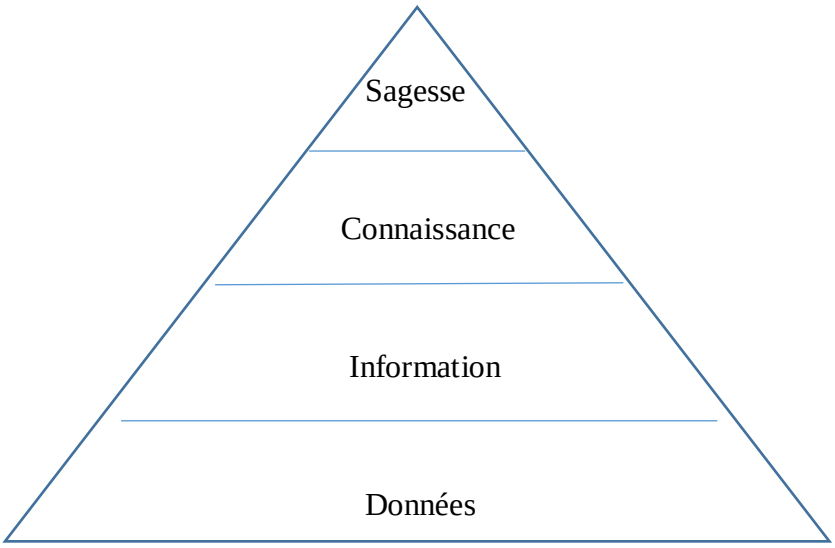


Figure 13: La pyramide DIKW [18]

7 Conclusion :

Certains spécialistes du Web affirment que le Web sémantique offre toutes les possibilités pour favoriser l'évolution de la connaissance humaine dans son ensemble. En effet, si l'on songe que la "domestication" du Web mettra à la portée de ses utilisateurs une information élargie sur des domaines spécifiques qui jusqu'à maintenant n'était repérable que partiellement, il s'agit d'un changement très important dans le monde de l'information.

En structurant les données non-structurées, le système RDF propose de définir les métadonnées propres à encourager un repérage efficace. En fait, le RDF est l'un des outils qui permettra de faciliter l'ensemble des opérations en lien avec le Web sémantique : recherche, indexation, condensation, etc. Mais il ne faudrait pas trop attendre pour élargir son utilisation puisque des concurrents commerciaux sont à l'affût. Topic Maps en est le plus sérieux : il veut s'accaparer le marché des métadonnées sur le Web et il possède déjà sa certification ISO. Par conséquent, le W3C et les autres organismes voués aux logiciels libres ont avantage à redoubler d'efforts pour faire valoir des solutions ouvertes comme le RDF afin que le Web continue partiellement d'échapper à la spéculation financière. Une partie de l'avenir de l'accès à l'information appartient à la bonne marche libre et ouverte du Web sémantique.

Les documents RDF peuvent être rédigés en différentes syntaxes, y compris en XML. Mais RDF en soi n'est pas un dialecte XML. Il est envisageable d'avoir recours à d'autres syntaxes pour exprimer les triplets. RDF est simplement une structure de données constituée de nœuds et organisée en graphe. Quoique RDF/XML — sa version XML proposée par le W3C

— ne soit qu'une sérialisation du modèle, elle est fréquemment nommée RDF. Un abus de langage sert à désigner à la fois le graphe de triplets et la présentation XML qui lui est associée.

1 Introduction :

Chaque jour un nombre incalculable de pages Web viennent se greffer à cette masse informe qu'est le Web. Difficile donc pour un être humain, normalement constitué, de s'y retrouver dans cette jungle !

Heureusement, des solutions existent pour retrouver de l'information pertinente dans tout ce contenu. Aujourd'hui les moteurs de recherche, grâce à leurs crawler, sont capables de parcourir récursivement les liens de milliards de pages Web et d'indexer leur contenu dans de gigantesques bases de données. Ainsi un utilisateur effectuant une recherche obtiendra une liste de résultats classée dans un ordre de pertinence correspondant à des critères spécifiques au moteur de recherche tels que la fréquence des mots-clés, l'indice de densité, etc.

Le langage de requête est un langage informatique utilisé pour accéder aux données d'une base de données ou d'autres systèmes d'information. Il permet d'obtenir les données vérifiant certaines conditions, permet les langages qui existe dans le web est [21] :

- Datalog pour les bases de données déductives.
- DMX pour les modèles d'exploration de données (Data Mining).
- MDX pour les bases de données multidimensionnelles OLAP.
- OQL pour les bases de données orientées objet.
- SPARQL pour les graphes RDF.
- SQL pour les bases de données relationnelles.
- XQuery pour les données XML.
- XPath pour parcourir le DOM.

Victime de son succès, le web est devenu un gigantesque réservoir d'informations rendant parfois la recherche d'information laborieuse, d'autant plus lorsqu'il s'agit de retrouver des informations fiables et pertinentes.

Le problème de l'accès à l'information n'est pas neuf. Il a déjà été abordé dans le domaine des sciences documentaires, pour des collections de documents-papier dans un premier temps, pour des ensembles de ressources électroniques ensuite. Avec l'avènement du réseau Internet et du Web1, c'est un nouveau type de collection documentaire qui est apparu. Son importance en ce qui concerne le nombre de documents et d'utilisateurs ainsi que l'accès largement public, au contraire de certaines archives présentes dans les entreprises et autres grandes organisations, ont alors entraîné une concentration importante des innovations dans ce

secteur. Si le web reste un cas particulier de collection de documents, les technologies développées pour y accéder sont néanmoins souvent applicables d'une manière générale à tous ensemble documentaire numérique.

2 Langage d'interrogation SPARQL :

2.1 Définition :

Un langage d'interrogation de données est un langage informatique, destiné à la recherche, extraction, tri et mise en forme, de données dans une base de données.

Une interrogation abstraite SPARQL est un uplet *(tuple)* (E, DS, R) où [22] :

- E est une expression algébrique SPARQL ;
- DS est un ensemble de données RDF ;
- R est une forme d'interrogation.

L'exécution d'une interrogation SPARQL est définie par une série d'étapes, commençant à l'interrogation SPARQL en tant que chaîne, en transformant cette chaîne en une forme de syntaxe abstraite, puis en transformant la syntaxe abstraite en une interrogation abstraite SPARQL comprenant des opérateurs de l'algèbre SPARQL. Cette interrogation abstraite est alors évaluée sur un ensemble de données RDF.

2.2 Forme d'interrogation SPARQL :

Le langage SPARQL est destiné à devenir le standard dans le domaine de l'interrogation de données stockées au format RDF.

SPARQL possède quatre formes de résultat : SELECT, CONSTRUCT, DESCRIBE, ASK [23].

a) **SELECT :** Identifie quelles variables nommées sont dans l'ensemble de résultats. « * » signifiant « toutes les variables nommées » (les nœuds anonymes dans la requête agissent comme des variables d'appariement, mais ne sont jamais retournés).

Avantage : traitement séquentiel simple des résultats.

Inconvénient : la structure / les relations sont perdues entre les expressions du résultat.

b) **CONSTRUCT :** Construit un graphe RDF basé sur un modèle de graphes. Le modèle de graphes peut avoir des variables qui sont liées par une clause WHERE. L'effet est de calculer

le fragment de graphe, donné par le modèle, pour chaque solution de la clause WHERE, après la prise en compte des modificateurs de solution. Les fragments de graphe, un par solution, sont fusionnés en un seul graphe RDF qui est le résultat. Tous les nœuds anonymes explicitement mentionnés dans le modèle de graphes sont créés de nouveau pour chaque fois que le modèle est utilisé pour une solution.

Avantage : données de résultats structurés avec des relations entre les éléments.

Inconvénient :

- Le traitement séquentiel des résultats est plus difficile.
- Aucun traitement des variables non liées (les triples sont omis).

c) **DESCRIBE :** La forme DESCRIBE crée également un graphe, mais la forme de ce graphe est fournie par le processeur de requêtes, pas par l'application. Pour chaque URI trouvée, ou explicitement mentionnée dans la clause DESCRIBE, le processeur de requêtes devrait fournir un fragment utile de RDF, comme tous les détails connus d'un livre. ARQ permet aux gestionnaires de descriptions spécifiques au domaine d'être écrits.

d) **ASK :** Le formulaire de résultat ASK retourne un booléen, vrai pour un motif qui correspond, faux sinon.

2.3 Définition de SPARQL :

SPARQL (acronyme dérivé de SQL) est un langage de requêtes destiné à interroger les bases de données (fichiers) RDF, standardisé par le W3C et implémenté sur les systèmes de base de données fournissent la récupération efficace des données par son langage d'interrogation sous la forme de langage d'interrogation structuré (SQL), l'ensemble de données dans un document RDF peut être interroger par le langage d'interrogation appelé le SPARQL. C'est une composante clé de la technologie sémantique de Web. Comme langage d'interrogation, SPARQL « orienté-donnée » parce qu'il interroge seulement l'information tenue dans les modèles, il n'y a aucune inférence dans le langage d'interrogation lui-même.

2.4 Avantage de SPARQL :

a) SPARQL permet de découvrir la structure d'une base de données. Cela servira à l'avenir à des agents (machines) sur le Web qui pourront ainsi découvrir les données disponibles à travers le Web pour répondre à des questions complexes. SPARQL ouvre ainsi les portes au Web des données (*Linked Data*), qui permettra à l'homme et à la machine de mieux interpréter les informations à travers le Web, sans service intermédiaire comme Google.

b) Il peut être utilisé pour exprimer des interrogations à travers diverses sources de données, que les données soient stockées nativement comme RDF ou vues comme du RDF via un logiciel médiateur *(middleware)*.

c) SPARQL est capable de rechercher des motifs de graphe *(graph patterns)* obligatoires et optionnels ainsi que leurs conjonctions et leurs disjonctions. Il gère également le test extensible des valeurs et la contrainte des interrogations par un graphe RDF source. Les résultats des interrogations SPARQL peuvent être des ensembles de résultats ou des graphes RDF.

d) Si nous considérons le Web sémantique comme une collection globale de bases de données, SPARQL peut faire en sorte que la collection ressemble à une grande base de données. Il nous permet de profiter des avantages de la fédération. Exemples :

- Fédérer des informations à partir de plusieurs sites Web (mashups)
- Fédérer des informations provenant de plusieurs bases de données d'entreprise (par exemple, systèmes de fabrication et de commande client et d'expédition)
- Fédérer des informations entre des systèmes internes et externes (par exemple, pour la sous-traitance, des bases de données Web publiques (par exemple, NCBI), des partenaires de la chaîne logistique).

e) SPARQL permet de gagner du temps et de réduire les coûts de développement en permettant aux applications client de travailler uniquement avec les données qui les intéressent. (Cela ne signifie pas tout réduire, mais aussi dépenser du temps et de l'argent pour écrire un logiciel permettant d'extraire les informations pertinentes.)

Exemple : recherchez le tarif de la population, de la région et du transport en commun (bus) des villes algériennes afin de déterminer s'il existe un lien entre la densité de population et les coûts de transport en commun.

- **Sans SPARQL :** vous pouvez résoudre ce problème en écrivant une première requête pour extraire des informations des pages des villes sur Wikipedia, une seconde requête pour extraire des données de transport en commun à partir d'une autre source, puis un code pour extraire les données de tarifs de population et de région et de bus pour chaque ville.
- **Avec SPARQL :** cette application peut être réalisée en écrivant une requête SPARQL unique qui fédère la source de données appropriée. Le développeur de l'application n'a besoin que d'écrire une seule requête et aucun code supplémentaire.

f) SPARQL s'appuie sur d'autres normes, notamment RDF, XML, HTTP et WSDL. Cela permet de réutiliser les outils logiciels existants et favorise une bonne interopérabilité avec d'autres systèmes logiciels.

Exemples : Les résultats SPARQL sont exprimés en XML : XSLT peut être utilisé pour générer des affichages de résultat de requête conviviaux pour le Web.

g) Il est facile de lancer des requêtes SPARQL, étant donné l'abondance de la prise en charge des bibliothèques HTTP en Perl, Python, PHP, Ruby, etc.

3 Les systèmes de traitement de données sémantique :

3.1 Définition et Objectifs :

a) Mapping :

Un mapping objet-relationnel est une technique de programmation informatique qui crée l'illusion d'une base de données orientée objet à partir d'une base de données relationnelle en définissant des correspondances entre cette base de données et les objets du langage utilisé.

L'objectif de ce système est de trouver les éléments de la B.C. (c'est-à-dire concepts, individus, relations et littéraux) correspondants à chaque mot clé de la requête de l'utilisateur.

b) Sempala :

Est une approche SPARQL sur SQL permettant le traitement de requêtes SPARQL en temps interactif sur Hadoop. Il stocke les données RDF dans une disposition en colonnes (Parquet) sur HDFS et utilise Impala ou Spark comme couche d'exécution par-dessus. Les requêtes SPARQL sont traduites en Impala / Spark SQL pour exécution.

L'objectif : Redéfinir notre structure de données RDF de Sempala en incorporant des structures de données imbriquées.

c) Méthodes d'optimisation pour le traitement de requêtes réparties à grande échelle sur des données liées :

Il est utilisé via un moteur de requêtes fédéré qui vise à minimiser le temps de réponse du premier tuple du résultat et le temps d'exécution pour obtenir tous les tuples du résultat.

Ce système fait la gestion des différents taux d'arrivée des données.

d) **SPARQLGX :** Un outil d'évaluation efficace des requêtes SPARQL sur les jeux de données RDF distribués. Il est également livré avec un évaluateur direct basé sur la même traduction SPARQL processus et appelé SDE, pour les situations où le prétraitement le temps compte au moins autant que le temps d'évaluation des requêtes.

e) Répondre aux requêtes par reformulation dans les bases de données RDF :

La première partie se concentre sur l'apport de réponse aux requêtes sur les données soumises à des contraintes RDFS, stockées dans un système de gestion de données relationnelles.

L'information implicite, résultant du raisonnement RDF est nécessaire pour répondre correctement à ces requêtes. Nous introduisons le fragment des bases de données RDF, allant au-delà de l'expressivité des fragments étudiés précédemment.

L'objectif :

- Optimisation de la maintenance de saturation lors de mise-a-jour.
- Comparer l'efficacité des techniques (taille et la fréquence des mises à jour)

f) Auto-complétions :

Elle propose l'utilisateur des mots clés ou des éléments de syntaxe du langage SPARQL. Elle peut se calculer sans difficulté à l'aide de notation EBNF du langage SPARQL. Néanmoins, en pratique, chaque service SPARQL ne supporte pas nécessairement l'ensemble de la syntaxe du langage.

Par exemple, le service SPARQL de Wikidata ne supporte pas les requêtes contenant le mot-clé GRAPH.

L'objectif : est de Proposer des complétions d'une requête en cours de rédaction en exploitant de nombreux types d'auto complétion et ce dans un contexte multi-services.

g) Qtor :

Ce système est porté par ses utilisateurs-trices et basé sur les similitudes entre requêtes. Les relations d'équivalences entre les différentes requêtes permettent de réunir les participants au sein de communautés d'intérêt.

L'objectif :

- Des outils numériques de plus en plus puissants.
- Réaliser des simulations locales.

3.2 Comparaison entre les systèmes :

Après la recherche et l'analyse sur quelques systèmes de traitements de données on les a regroupés par :

A. Stratégie pour les requêtes :

a) Le Rappel et MRR sont indispensables dans le système Mapping, afin de réaliser une stratégie pour les requêtes.

b) Une fédération de requêtes pour le système (Méthodes d'optimisation pour le traitement de requêtes réparties à grande échelle sur des données liées).

c) Répondre aux requêtes par saturation et par reformulation pour le système (Répondre aux requêtes par reformulation dans les bases de données RDF).

d) Le système auto-complétions générique de SPARQL :

- L'utilisation d'un éditeur SPARQL pour la rédaction d'une requête.
- Un service SPARQL va répondre à une requête décrite au sein d'un éditeur SPARQL.

B. Partitionnement de données :

Il existe deux types de partitionnements des données, l'un est thématique et un autre vertical. Le 1er se conçoit dans le système QTor et le second dans les systèmes SPARQLGX, R-Type, Sempala.

Par contre dans les systèmes (Répondre aux requêtes par reformulation dans les bases de données RDF et Mapping) Il n'existe aucun partitionnement de données.

C. Partitionnement de la requête :

On trouve cette option chez les systèmes (Méthodes d'optimisation pour le traitement de requêtes réparties à grande échelle sur des données liées, répondre aux requêtes par reformulation dans les bases de données RDF).

DataSets) on remarque l'absence de cette option.

D. SGBD :

Le stockage des données se fait pour tous les systèmes, donc chacun possède un SGBD et, on remarque que les systèmes proposés dans notre étude exigent le HDFS (SPARQLGX, Sempala, R-Type).

E. BDD :

Les systèmes de traitement de données exigent d'intégrer des bases de données pourqu'ils puissent faire leurs traitements, tout on savant qu'il y a des différents langages des bases de données, nous citons dans cette étude les exemples suivants :

- CQL pour le système QTor.
- Apache HBase pour le système (Storing, Indexing and Querying Large Provenance DataSets)

On constate qu'il y a des systèmes de traitements de données disposant plus d'un langage de base donnée tel que :

- SPARQLGX qui dispose LUBM et Watdiv comme BDD.
- DBPSB et LUBM pour le système R-Type.
- Le système de Répondre aux requêtes par reformulation dans les bases de données RDF qui exige les deux bases de données DBpedia et DBLP.

F. Type système (centralisé, distribué, autre) :

Dans l'étude qu'on vient le faire les deux systèmes utiliser sont le système centralisé et le système distribué.

a) Centralisé pour :
   - Mapping.
   - Répondre aux requêtes par reformulation dans les bases de données RDF.
   - R-Type.

b) Distribué pour :
   - Méthodes d'optimisation pour le traitement de requêtes réparties à grande échelle sur des données liées.
   - QTor.
   - SPARQLGX.
   - Storing, Indexing and Querying Large Provenance DataSets.
   - Sempala.
   - Une auto complétion générique de SPARQL.

Il existe d'autre système mais ne figure pas dans cette étude vue que l'échantillant pris ne les intègres pas.

G. Outils de distribution du traitement

Aucun outil de distribution de traitement n'est utilisé dans le système centralisé par contre dans le système distribué on trouve :

- Le NAT pour QTor.
- Apache Spark pour SPARQLGX.
- Le bitmap pour le systèmeStoring, Indexing and Querying Large Provenance DataSets.

H. Outil de développement :

Chaque système dispose d'un seul outil (FleDDi pour le système QTor) sauf le système mapping qui dispose deux outil Java et Jena.

Avantage :

1. Les opérateurs de jointure adaptatifs proposés dans la méthode d'optimisation pour le traitement de requêtes réparties à grande échelle sur des données liées présentent le meilleur compromis entre le temps de réponse et le temps d'exécution
    - L'efficacité des opérateurs adaptatifs proposés.
2. Pour le système QTor l'organisation des requêtes est la plus simple.
3. Approche est efficace et évolutive pour le système Storing, Indexing and Querying Large Provenance DataSets.
4. Le système R-Type est le plus rapide que les approches existantes.
5. Répondre aux requêtes par reformulation dans les bases de données RDF est basé sur deux techniques :
    - Technique01 :sa facilité de mise en œuvre.
    - Technique02 : est que la saturation n'a pas à être calculée.

Inconvénient :

1. Pour le système auto complétion Les services SPARQL ne supportent pas toute la syntaxe du langage.
2. La taille des données est très petite et ne permet pas de montrer l'efficacité du système Mapping.

3. Répondre aux requêtes par reformulation dans les bases de données RDF (sur deux techniques) :

- Technique01 : la saturation nécessite du temps pour être calculé, de l'espace pour être stockée, et que celle-ci doit être recalculée lors de mises-à-jour.
- Technique02 : chaque requête doit être reformulée, et que sa reformulation résulte généralement en une requête plus complexe à évaluer.

En faisant la comparaison des systèmes de traitements de données sémantique, et sur proposition de notre encadreur des articles a étudié, nous avons élaboré un tableau cité ci-après qui comporte toutes les informations nécessaires sur les systèmes de traitements de données.

Tableau 1 : Comparaison de la base de données, SGBD et le but des systèmes.

Nom du système	But	BDD	SGBD
Mapping Système d'interrogation basé sur l'annotation sémantique	Trouver les éléments de la B.C. (c'est-à-dire concepts, individus, relations et littéraux) correspondants à chaque mot clé de la requête de l'utilisateur	Base d'annotations sémantiques	_____
Méthode d'optimisation pour le traitement de requêtes réparties à grande échelle sur des données liées	Gérer différents taux d'arrivée des données	Les bases de données relationnelles (Parallèles)	_____
toring, Indexing and Querying Large Provenance DataSets		Apache HBase	_____
Une auto complétion générique de SPARQL	Proposer des complétions d'une requête en cours de rédaction en exploitant de nombreux types d'auto	Wikidata	_____

	complétion et ce dans un contexte multi-services		
Répondre aux requêtes par reformulation dans les bases de données RDF	-Optimisation de la maintenance de saturation lors de mise-a-jour. -Comparer l'efficacité des techniques (taille et la fréquence des mises à jour)	Graphe DBpedia et DBLP	_____
QTor	Des outils numériques de plus en plus puissants Réaliser des simulations locales	CQL	Relationnelle
Sempala	Redéfinir notre structure de données RDF de Sempala en incorporant des structures de données imbriquées	BigData	HDFS
SPARQLGX		LUBM, Watdiv	HDFS
R-Type	Améliorer l'efficacité de la correspondance de motif de graphe	DBPSB, LUBM	HDFS

Tableau 2: Comparaison de Partitionnement de données, partitionnement de la requête et la stratégie pour les requêtes des systèmes

Nom du système	Partitionnement de données	Partitionnement de la requête	Stratégie pour les requêtes
Mapping Système d'interrogation basé sur l'annotation sémantique	Non	Non	Rappel et MRR
Méthodes d'optimisation pour le traitement de requêtes réparties à grande échelle sur des données liées	Non	La sélection de sources de données, l'optimisation et l'exécution de requêtes	La fédération de requêtes
Storing, Indexing and Querying Large Provenance DataSets	Oui	Non	_____
Une auto complétion générique de SPARQL	Non	Non	L'utilisation d'un éditeur SPARQL pour la rédaction d'une requête.
Répondre aux requêtes par reformulation dans les bases de données RDF	Non	Oui	Répondre aux requêtes par saturation et par reformulation
QTor	Partitionnement thématique	Non	_____
Sempala	Vertical	Non	_____
SPARQLGX	Partitionnement vertical	Non	_____
R-Type	Vertical	Non	_____

Tableau 3: Comparaison des Outils de distribution du traitement et Type système.

Nom du système	Type système (centralisé, distribué, autre)	Outils de distribution du traitement
Mapping Système d'interrogation basé sur l'annotation sémantique	Centralisé	Non
Méthodes d'optimisation pour le traitement de requêtes réparties à grande échelle sur des données liées	Distribué	Moteur de requêtes fédéré
Storing, Indexing and Querying Large Provenance DataSets	Distribué	Bitmap
Une auto complétion générique de SPARQL	Distribué	_____
Répondre aux requêtes par reformulation dans les bases de données RDF	Centralisé	Non
QTor	Distribué	NAT
Sempala	Distribué	Non
SPARQLGX	Distribué	Apache Spark
R-Type	Centralisé	Non

4 Conclusion :

De nombreuses technologies de base de données distinctes sont utilisées et il est bien entendu impossible de dicter une technologie de base de données unique à l'échelle du Web. RDF (le modèle de données Web sémantique), cependant, sert de lingua franca (plus petit dénominateur commun) standard dans laquelle les données de systèmes de bases de données disparates peuvent être représentées. SPARQL est donc le langage de requête pour ces données. En tant que tel, SPARQL masque les détails de la gestion des données et des détails de structure d'un serveur. Cela réduit les coûts et augmente la robustesse des logiciels émettant des requêtes.

Il y a aussi d'autre avantage de SPARQL sur l'API on va vous citez les plus intéressant [24] :

a. Pas besoin de lire la doc sur l'API : nous lançons une requête SPARQL et nous voyons ce qui sort. Avec une API, on ne peut pas lancer de requête à l'aveugle, il faut d'abord savoir comment elle fonctionne

b. Le langage de requête est standard, il suffit de l'apprendre.

c. On peut envisager n'importe quelle requête sur les données, même celles non envisagées par le fournisseur. Seule condition : que les données soient là.

d. On peut décider de récupérer les informations qu'on veut rapatrier dans les résultats.

Dans notre mémoire nous sommes faits des recherches sur les systèmes de traitement de données sémantiques, et donc on a trouvé que les opérateurs de jointure adaptatifs proposés présentent le meilleur compromis entre le temps de réponse et le temps d'exécution, l'efficacité des opérateurs adaptatifs, plus rapide que les approches existantes.

La saturation nécessite du temps pour être calculé, de l'espace pour être stockée, et que celle-ci doit être recalculée lors de mises-à-jour, chaque requête doit être reformulée, et que sa reformulation résulte généralement en une requête plus complexe à évaluer. La taille des données est très petite et ne permet pas de montrer l'efficacité du système et à la fin les services SPARQL ne supportent pas toute la syntaxe du langage.

1 Introduction :

Le Web sémantique a permis à l'utilisateur d'utiliser la machine pour trouver, partager et combiner les informations afin d'exploiter de nouvelle connaissance et pour les transférer d'un endroit à l'autre très facilement. Le Web sémantique permet de rendre Web actuel rend plus compréhensible par la machine. Ceci permet aux utilisateurs d'être autorisés à utiliser le sens des choses dans le Web pour exécuter les tâches de recherche d'informations sémantiquement. Pour ce faire, le Web sémantique fournit des instructions pour que la machine exécute différentes tâches de recherche d'informations en fournissant une requête sémantique qui peut interpréter la recherche sémantiquement. Par conséquent, les machines peuvent effectuer la tâche fournie par le Web sémantique pour trouver, combiner et agir sur les informations présentes sur le Web.

Dans ce chapitre nous allons décrire un ensemble d'étapes nécessaires pour l'interrogation d'une base de données RDF. Le but de ce chapitre est de proposer un modèle qui permet d'optimiser le traitement des données sémantique RDF. Initialement, nous allons donner la structure générale de notre modèle avec ses différents composants, ensuite nous expliquerons le fonctionnement de chaque composant en détails.

2 Le modèle de traitement de données sémantique optimisé :

Le modèle proposé permet d'effectuer un traitement sémantique sur un ensemble de données RDF. Le traitement est optimisé via deux points :

Premièrement, l'utilisation du partitionnement du traitement sémantique en un ensemble de sous traitements. Deuxièmes, l'exécution des traitements sur la représentation graphique des données sémantiques.

Ceci nous permet de partager le traitement sur plusieurs nœuds de calcul afin d'alléger la charge de travail et avoir des résultats dans un temps de réponse raisonnable ce qui nous permet d'améliorer ce dernier.

Afin de montrer le fonctionnement de notre modèle nous proposons dans le schéma ci-dessous l'architecture générale de notre modèle figure 14.

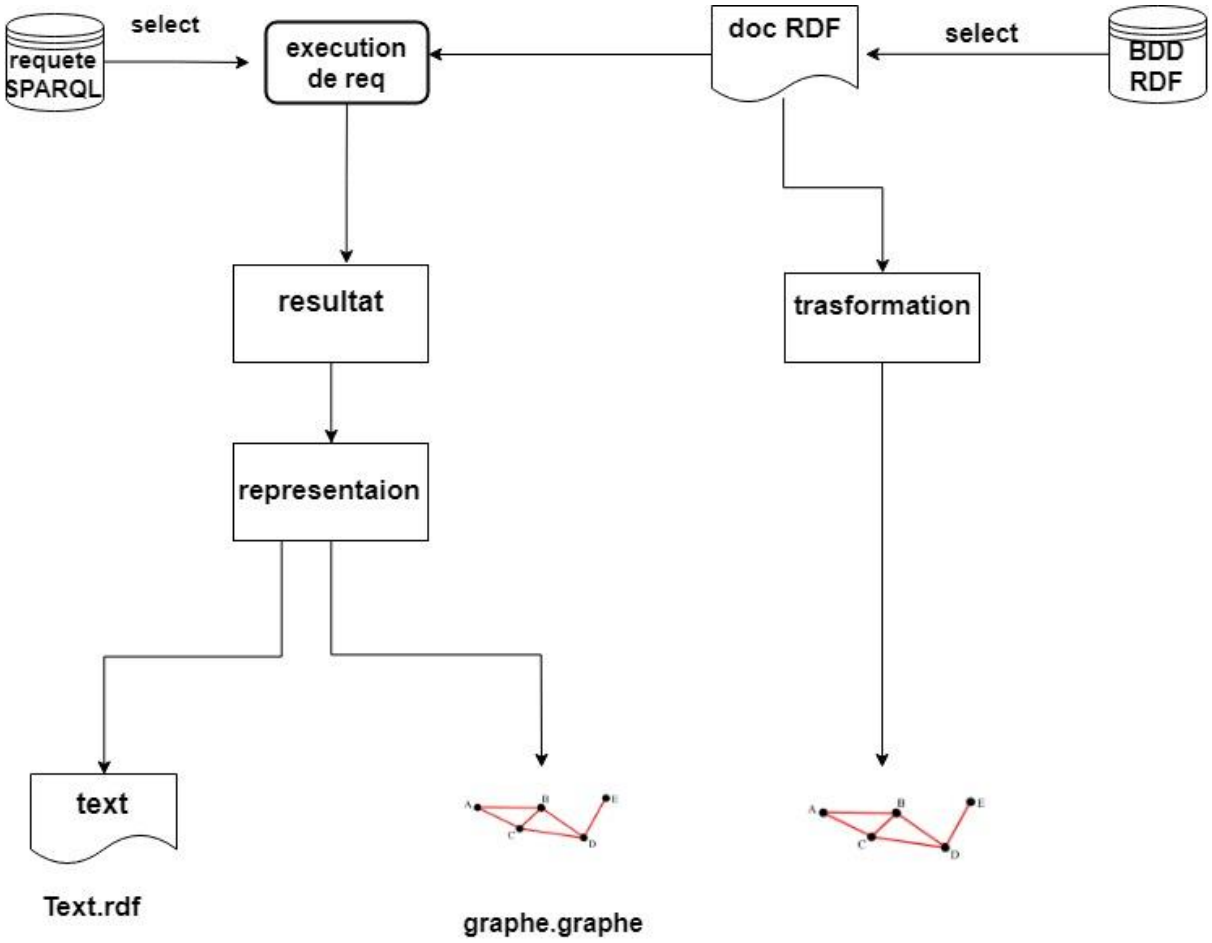


Figure 14 : Le modèle général

Initialement, le modèle donne la possibilité de choisir un document RDF parmi un ensemble de documents RDF qu'il contient dans la BDD RDF par une simple sélection, ensuite, le document est représenté sous forme graphique en passant par un processus de transformation graphique.

Dans l'étape suivante et suite à un événement déclenché dans le système, une requête parmi un ensemble de requêtes sera exécutée sur le document RDF sélectionné via deux modes d'exécution : avec ou sans partitionnement. Le résultat de l'exécution de la requête sera soumis sous une représentation graphique. Par la suite, une comparaison sera effectuée sur les deux graphes obtenus et une analyse sur les deux modes d'exécution de requêtes dans le but de déterminer laquelle de ces deux modes offre un temps d'exécution plus court, en visualisant le résultat obtenu.

2.1 Ensemble des bases de données :

Dans notre modèle, il y a deux types différents de base de données : La base de données qui contient les documents RDF appelée BDD RDF, et la base de données qui contient l'ensemble des requêtes SPARQL appelée BDD requêtes.

2.1.1 BDD RDF :

Lorsque le modèle reçoit en entrée des données sémantiques qui parviennent de plusieurs sources sous différentes formes, ces ensembles de données sémantiques sont transformées en un ensemble de documents RDF ce qui nous permet de mettre en évidence la sémantique des données. Ensuite, les documents RDF sont stockés dans la base de données de RDF de types TripleStore [2].

Le document RDF contient un ensemble de déclaration « statement », chaque déclaration comprend un sujet, un prédicat, et un objet. Formellement un document RDF peut être représenté comme suit figure 15 :

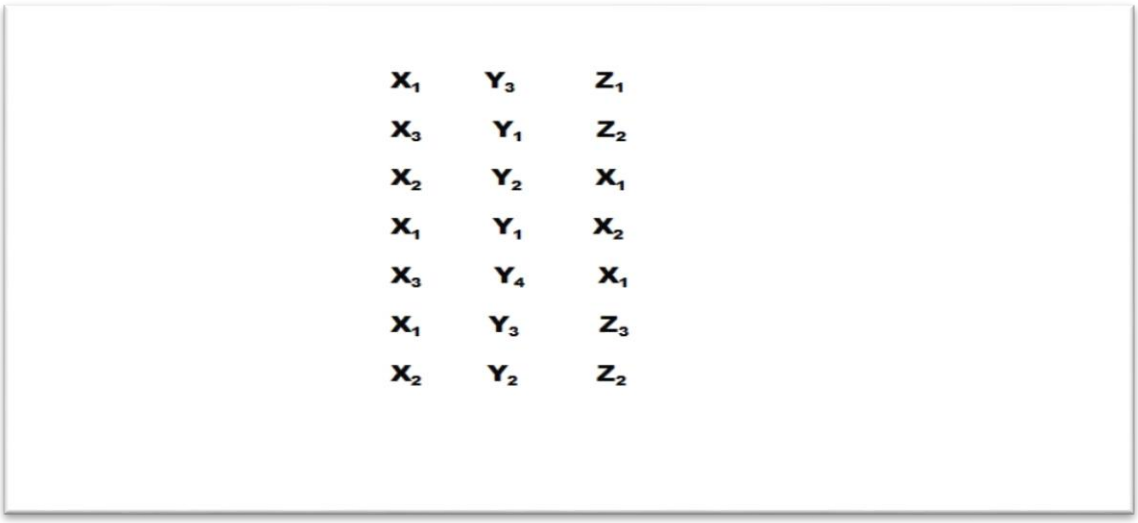


Figure 15 : Document RDF

2.2.2 BDD requêtes :

La BDD requêtes contient contiens un ensemble de requête exprimer via SPARQL. Chaque document RDF lui est attribuer un ensemble de requêtes adéquates avec le contenu du document.

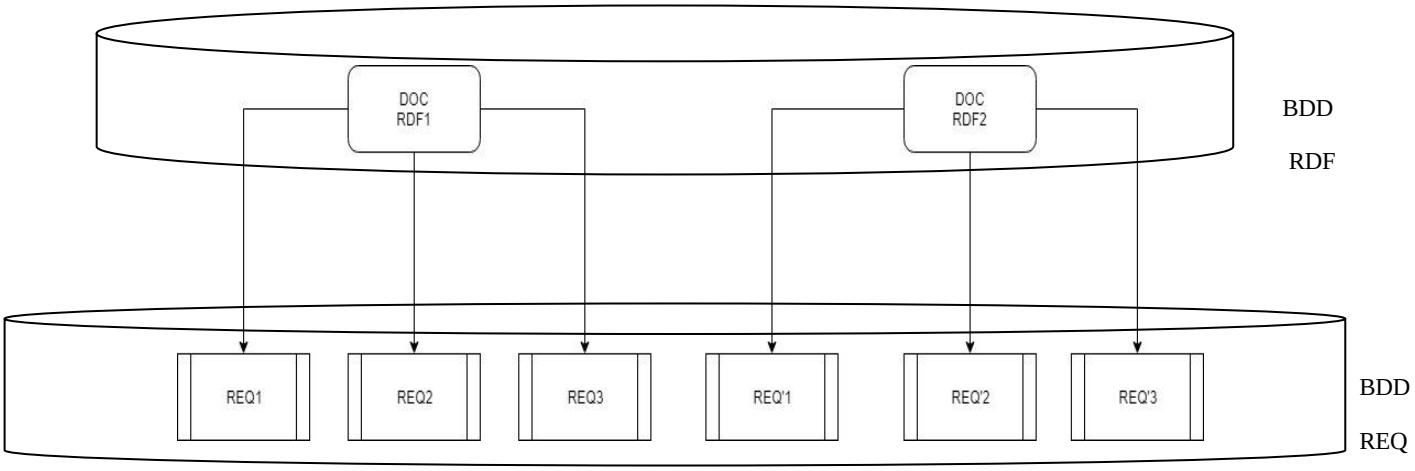


Figure 16 : BDD REQUETE

Par exemple, une requête de BDD requête peut-être exprimer de la forme suivante :

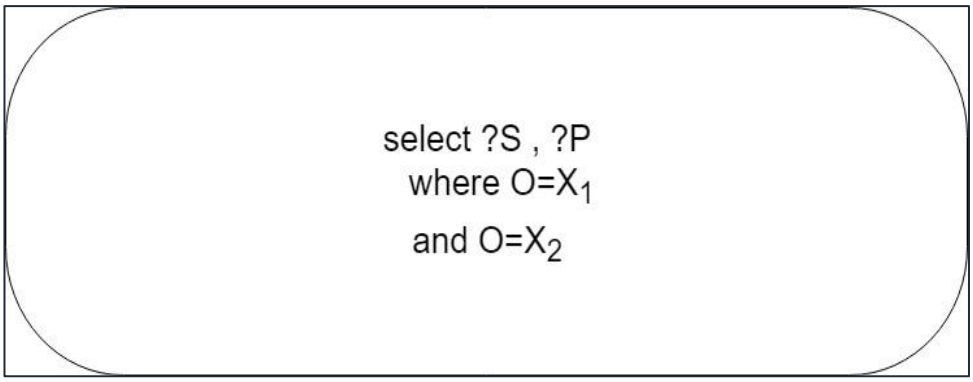


Figure 17 : Requête de BDD

Cette requête cherche les différentes valeurs des sujets et des prédicats à partir d'un ensemble de triplets d'un document RDF particulier lorsque la valeur de l'objet d'un triple est égale à la valeur X_1 et à la valeur X_2.

2.3 Transformation :

Lorsque le modèle reçoit en entrée les données d'un document RDF qui parviennent de plusieurs sources sous différentes formes. Ces données sont stockées dans une base de données sémantique sous forme de documents. Ceci permet à un utilisateur par la suite de sélectionner un document de la base de données. Avant la représentation graphique du document, le modèle teste si le document a été est déjà traité et transformer en graphe, si c'est le cas, les résultats de traitement qui ont été stocké auparavant dans la base de données seront affichées directement. Dans le cas contraire, le document sélectionné sera traité en identifiant ses ressources. Pour ce faire, le modèle test la nature des ressources, si les ressources sont des objet ou sujet, un nœud

serra crée sinon si c'est un prédicat, une arrête serra crée et un lien serra établi entre les nœuds de type sujet et les nœuds de type objet. Enfin on fait la fusion des nœuds et des arrêts et qui nous permet la conception du graphe qui va être affiché.

Afin de montrer le processus de la transformation d'un document RDF en graphe nous proposons l'organigramme ci-dessous figure 18.

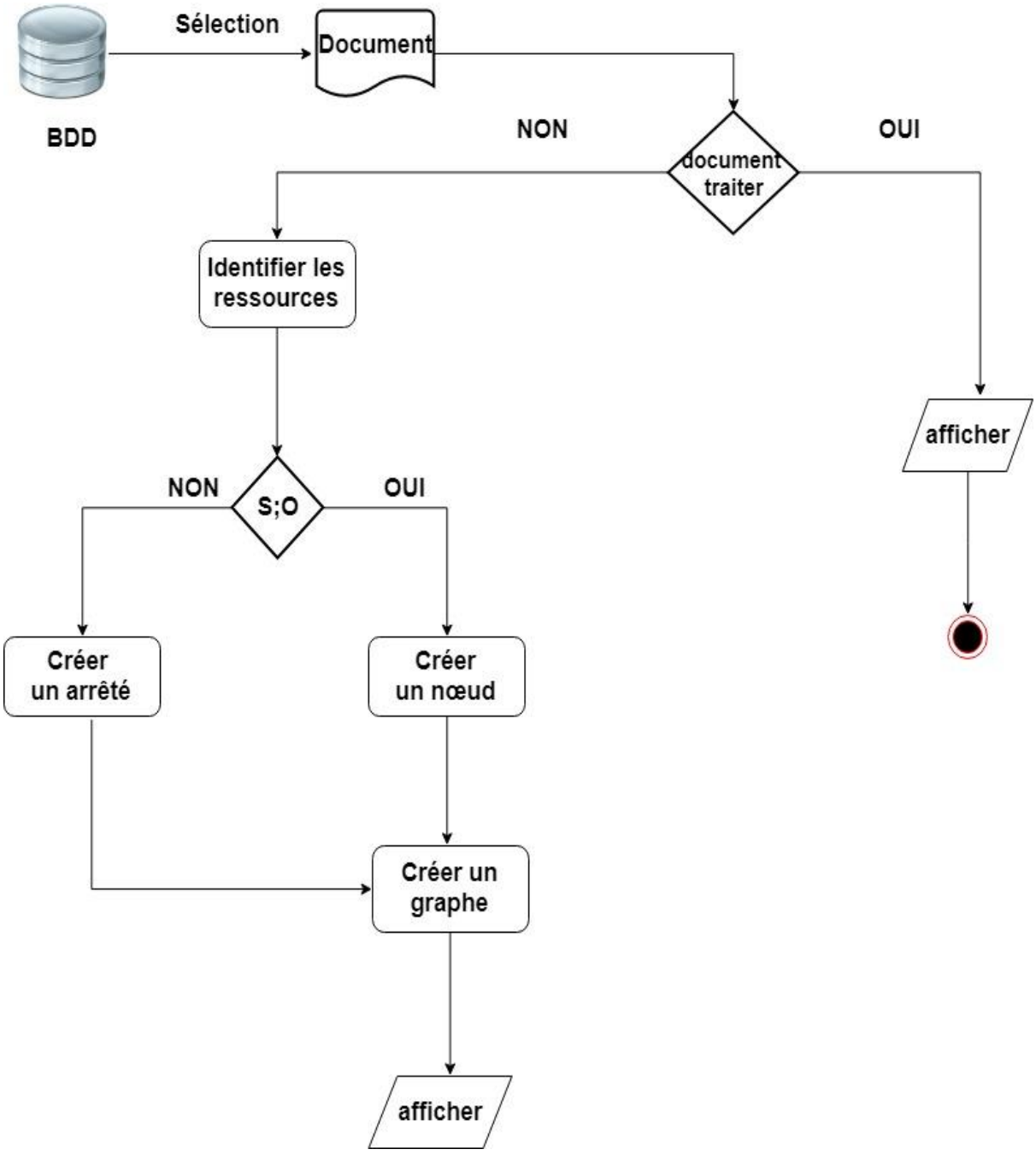


Figure 18 : L'organigramme de modèle

Par exemple la transformation de l'exemple de la figure 19 est le graphe de la figure ci-dessous.

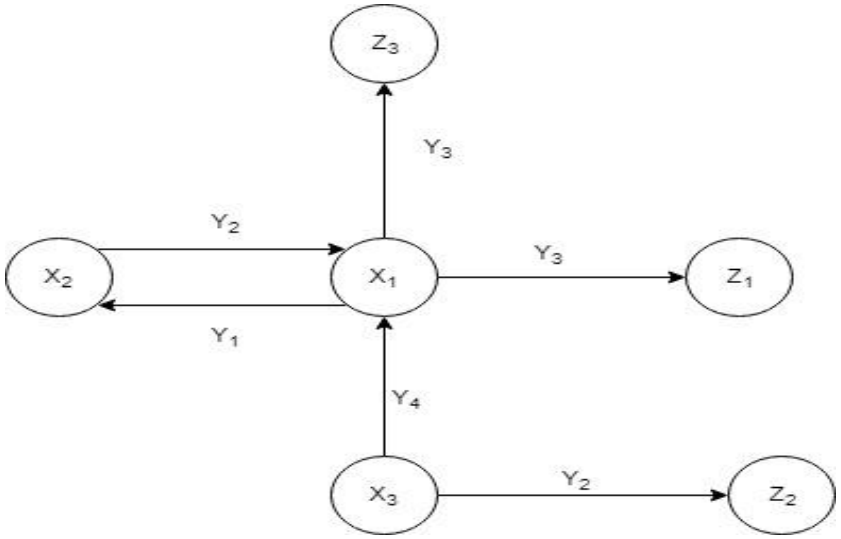


Figure 19 : Résultat d'un document en graphe

2.4 Exécution de requête :

Le composant exécution de requête de notre modèle figure 20 permet à la fois d'extraire des connaissances sémantiques prérequis et d'accélérer l'accès à un petit ensemble de données à partir d'une grande masse de connaissances sémantiques. Lorsque l'exécution de la requête est activée, la requête SPARQL va générer un graphe à partir de la déclaration de la requête. Dans ce cas, l'utilisateur a la possibilité de déclencher l'exécution de la requête et il peut en plus choisir le mode d'exécution soit avec ou sans partitionnement.

S'il le mode d'exécution sans partition est sélectionnée, la requête serra exécuter immédiatement sans avoir recours à une optimisation, et ensuite l'utilisateur pourra visualiser le résultat.

Sinon lorsque le mode d'exécution avec partitionnement est activé, la requête déclenchée serra subdiviser en un ensemble de sous requêtes et chaque sous requête serra exécuter indépendamment des autres sous requêtes. Ensuite, le résultat de chaque sous requête serra regrouper et fusionné pour concevoir le résultat final de l'exécution de la requête.

Dans la figure 20 ci-dessous nous résumant le contenu du composant exécution de requête

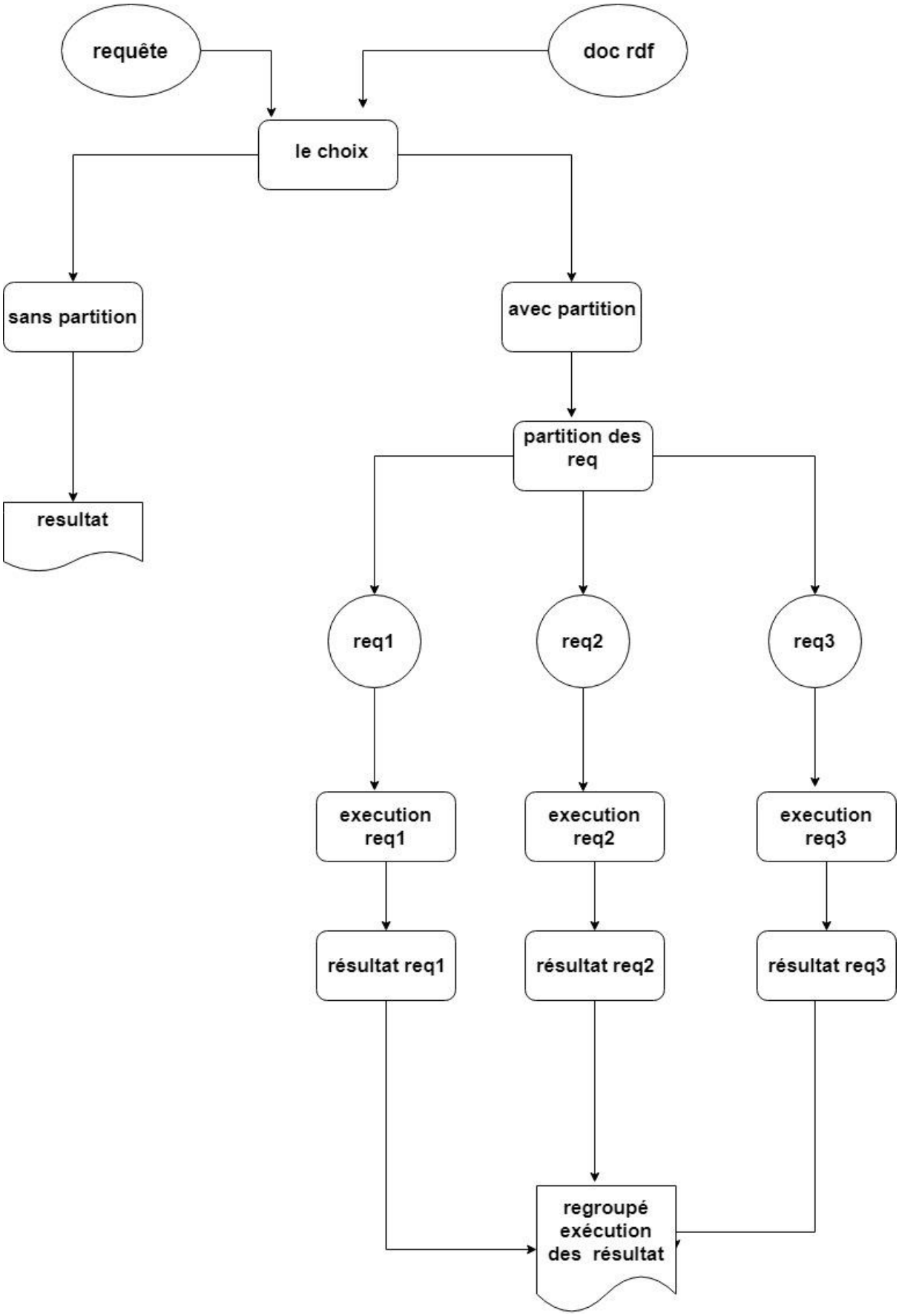


Figure 20: Exécution de requête

2.4.1 Exécution sans partition :

L'exécution d'une requête sans partition consiste à traiter la requête SPARQL déclenché sans modification. Par exemple, la requête ci-dessous sera lancer et exécuter entièrement en une seule action. Et par conséquent nous obtiendrons un seul résultat.

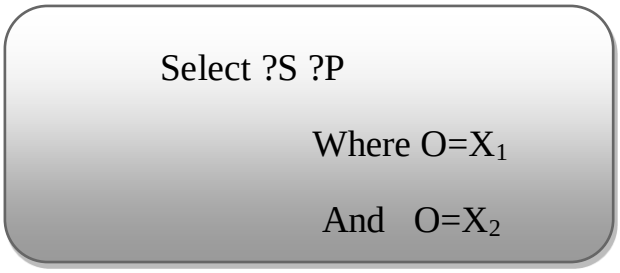


Figure 21: Requête SPARQL

2.4.2 Exécution avec partition :

Ce mode d'exécution consiste on va divise la requête SPARQL déclenché en un ensemble de sous requêtes. La division de la requête et le nombre de sous requête dépond du nombre de clauses de la partie condition de la requêtes SPARQL. Par exemple, la division de la requête de l'exemple de la figure ci-dessus est représentée dans la figure ci-dessous.

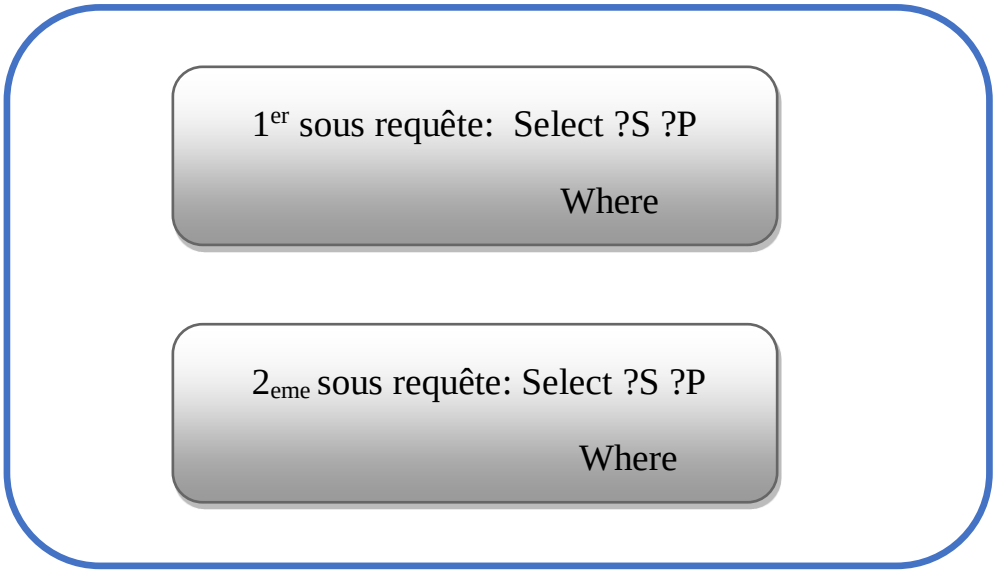


Figure 22: Requête SPARQL devisé

2.4.3 Résultat & Représentation :

La déclaration d'une requête SPARQL va générer un graphe, RDF tel que le sujet, le prédicat et l'objet peuvent être des variables comme illustré précédemment.

Après l'affichage de la requête SPARQL l'utilisateur a la possibilité de l'exécuter de plus il peut choisir le type d'exécution soit avec ou sans partitionnement.

Une comparaison sera effectuée entre ces deux méthodes d'exécution, par la suite l'utilisateur pourra visualiser le résultat de la comparaison dans un document.

Un document RDF récupéré de la base de données RDF ne peut pas être directement partitionné, une transformation est nécessaire pour rendre notre graphe manipulable par le programme de partitionnement. Le fichier d'entrées généré, il est maintenant possible d'entamer le partitionnement de notre graphe en utilisant le programme de partitionnement de graphe, le résultat obtenu est une partition P= {REQ'1, ….., REQ'n}, la requête sera exécuté sur chaque partition ainsi on obtiendra un ensemble de résultats qui sera fusionné pour former le résultat finale.

La figure ci-dessous montre la différence entre exécution sans et avec partition dans le cas avec partition chaque sous requête sera exécuté tout seul et nous obtiendrons les résultats et ont fusionné les résultats obtenus pour former le résultat final, dans le cas contraire la requête sera exécutée et on obtiendra un seul résultat qui est finale.

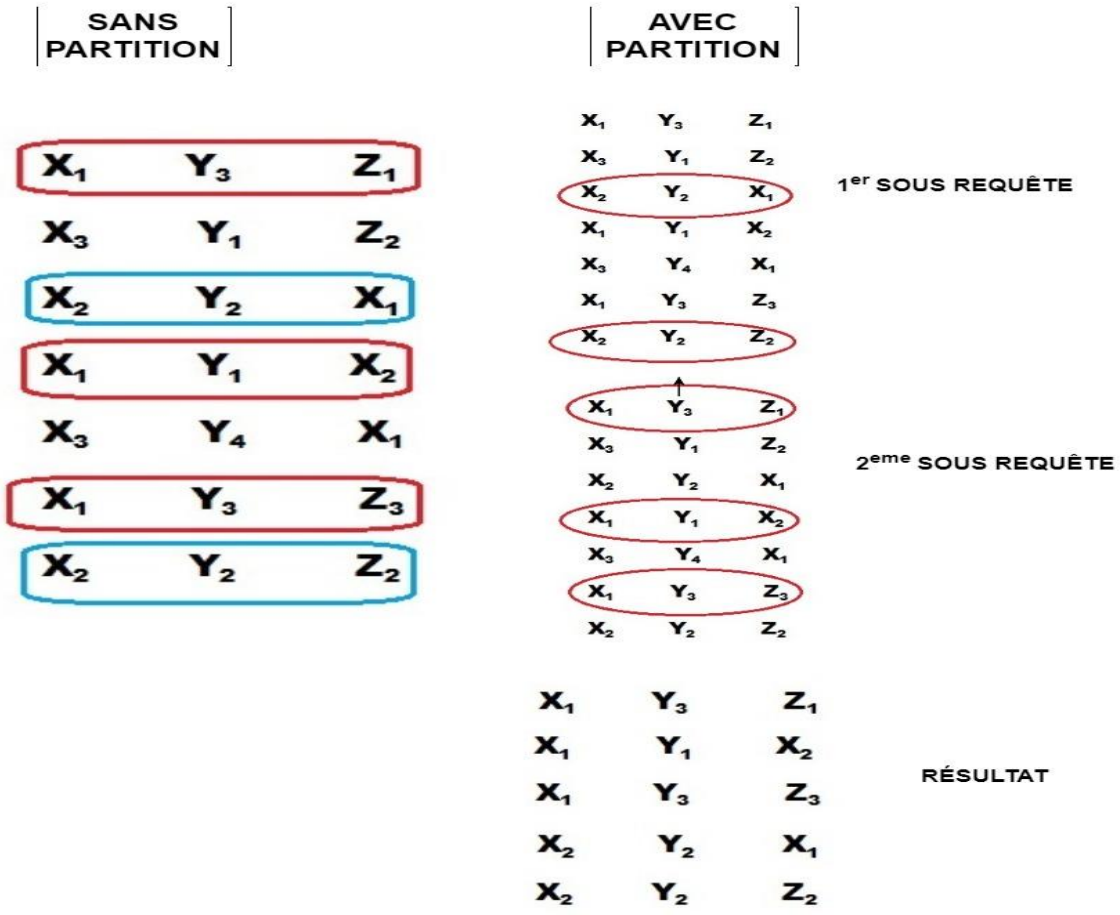


Figure 23:Résultat de la requête avec et sans partition

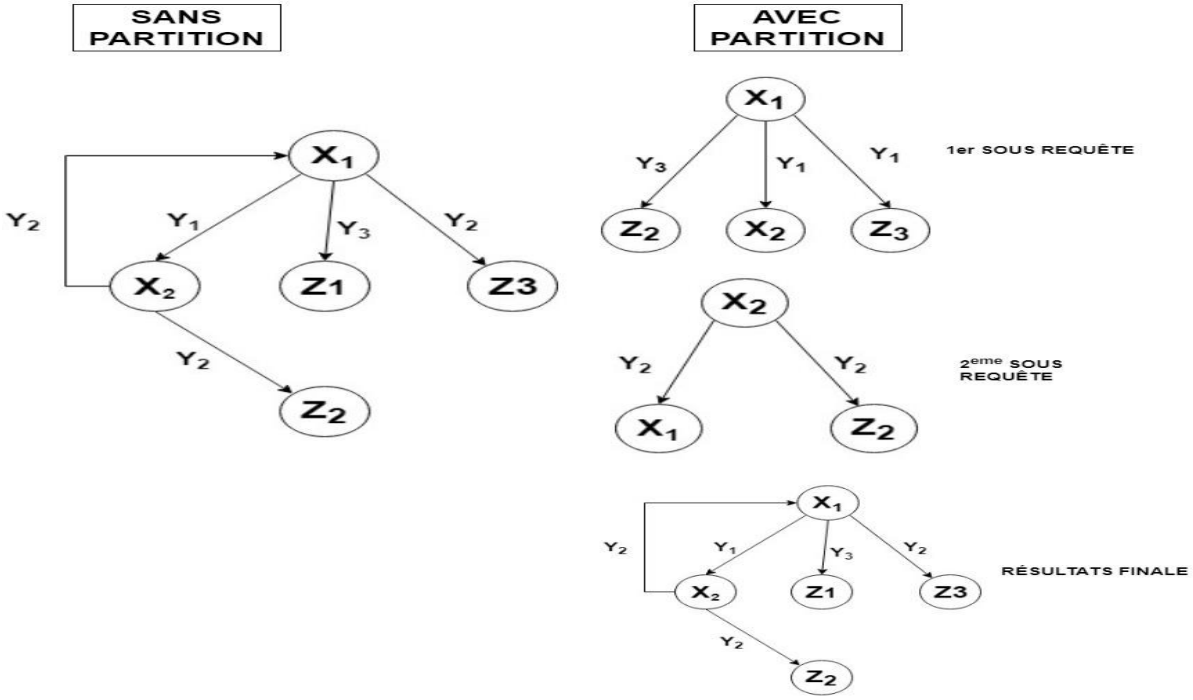


Figure 24 : Résultat de la requête avec et sans partition en graphe

2.5 Algorithme de la représentation des résultats :

Algorithme ci-dessous figure 25 montre le fonctionnement de notre modèle. On utilise dans cet algorithme une fonction figure 26 qui s'appelle ***convertir_en graphe (Triplet)*** qui permet de convertir les triplets in dépendamment en graphe après l'identification des ressources. Cette fonction admet en entrée un Triplet composé d'un sujet S, prédicat P, objet O.

Dans l'algorithme on a en entrée la requête, le document RDF sélectionné, et choix de l'exécution de la requête et en sortie nous avons la représentation graphique du résultat.

Si le choix de l'exécution de la requête SPARQL est sans partition, dans ce cas la requête serra exécutée, et on appliquera pour chaque triplet de l'ensemble des triplets du résultat la fonction ***convertir_en_graphe*** (Triplet) afin d'obtenir leurs représentations graphiques, après on fait l'union des sous graphes. Sinon dans le cas où la requête est exécutée avec partitionnement, la requête SPARQL serra partitionné en un ensemble de sous requêtes et chaque partition qui représente une sous requête serra exécuter en parallèle. Ensuite, pour chaque résultat nous allons appliquer la fonction ***convertir_en_graphe*** (Triplet)) pour chaque résultat de toutes les sous requêtes et on fait l'union des sous graphes pour concevoir le graphe des résultats.

Entrée : requête ; doc ; choix de l'exécution

Sortie : graphe

DEBUT

SI choix de l'exécution == sans partition alors

Exécuter la requête

POUR Chaque triplet ∃ résultat

Sous graphe == convertir_en_ graphe (Triplet)

FIN POUR

Union des sous graphes.

SINON

Partitionné la requête

POUR chaque sous requête

Exécuter sous requête

POUR Chaque triplet ∃ résultat

Sous graphe == convertir_en_ graphe (Triplet)

Union des sous graphes.

FIN POUR

FIN

FIN POUR

Union des sous graphes.

FIN

Figure 25 : L'algorithme de fonctionnement de modèle.

Convertir_en_graphe (Triplet)

S → NOEUD

O → NOEUD

P →ARRET

Figure 26 : L'algorithme de fonction Convertir_en_graphe

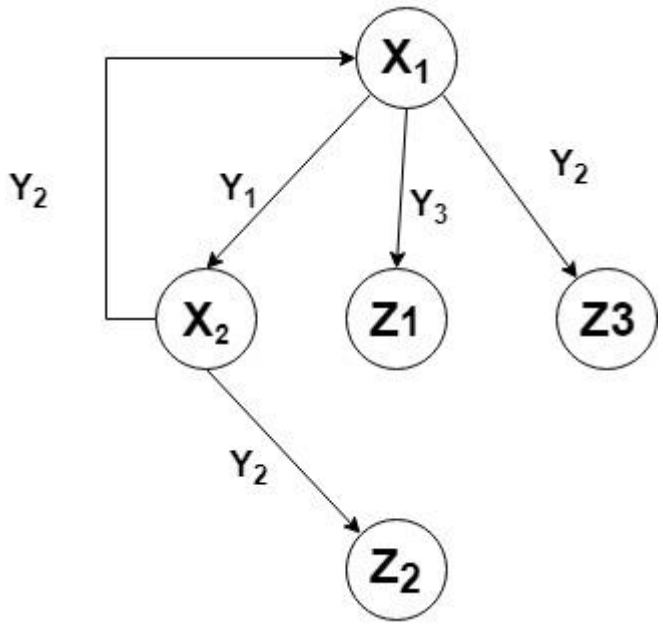


Figure 27: Requête SPARQL devisé

2.6 Traitement des données RDF en graphe :

En résumé, afin de réaliser la transformation de documents sémantiques en graphes, notre modèle passe par plusieurs étapes suivant le modèle proposé. Au début, il faut charger un document sémantique RDF, une fois le chargement effectué on a la possibilité de le visualiser sous forme de graphe c'est à dire tous les ressources et les objets serons transformer en nœuds et les prédicats en arrêtes. Ensuite effectuer des requêtes sur notre graphe, ces requêtes sont importées d'un document externe, après le choix d'une de ces requêtes il faut déterminer le type d'exécution de cette dernière soit avec ou sans partitionnement.

Si l'exécution sans partition, fait le traitement de la requête et afficher le résultat ; Dans l'autre cas c'est-à-dire avec partition divise la requête sur plusieurs sous requête et fait le traitement sur chaque sous requête et faire la fusion des résultats et l'affiché.

Afin de montrer la chronologie des différentes étapes de notre modèle nous proposons dans la figure ci-dessous le diagramme de séquence correspondant.

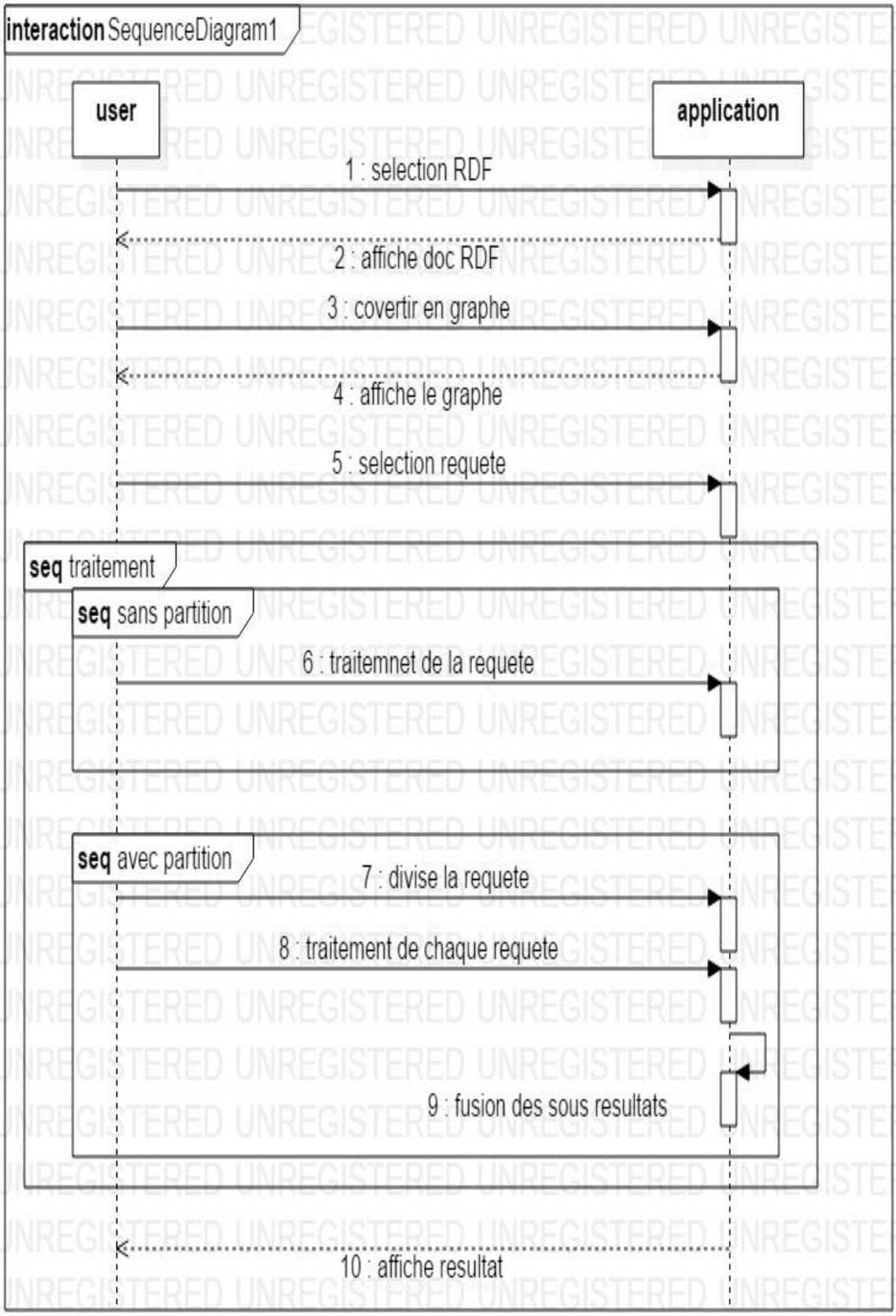


Figure 28 : Diagramme de séquence du modèle

3 Conclusion :

Au terme de ce chapitre nous concluons que la représentation des données sur le web avec un modèle unique (RDF), permet de favoriser l'enrichissement et la réutilisation de l'information.

Ainsi il est maintenant possible de stocker un grand nombre de triplets dans un TripleStore, de plus la flexibilité et la simplicité de ce modèle facilite son interrogation avec le langage de requête SPARQL, Même si SPARQL reste un langage d'interrogation fiable et sur son utilisation sur des graphes énormes risque d'avoir un temps de réponse assez long pour cela on est dans l'obligation de diminuer le volume du graphe pour minimiser l'espace de recherche, donc on utilise la méthode de partitionnement de graphe qui a pour avantage de réduire l'espace de recherche pour les requêtes SPARQL.

Afin de montrer l'intérêt de cette méthode, notre modèle permet d'illustrer ses principaux avantages.

Dans le prochain chapitre nous allons montrer les outils logiciels utilisées, pour implémenter notre modèle.

1 Introduction :

Afin de mettre en place le modèle proposé dans le chapitre précédent ainsi que ses différents composants, nous allons présenter dans ce chapitre les différents outils de développement et les logiciels utilisés pour l'implémentation du modèle, ensuite, nous allons présenter notre application en indiquant en détail les étapes nécessaires pour son utilisation.

Dans ce chapitre nous allons présenter l'implémentation de notre modèle en présentant les outils, les API, et les logiciels utilisés dans cette réalisation, ensuite nous allons présenter notre application en indiquant en détail les étapes nécessaires pour l'utilisation de l'application, et on terminera ce chapitre par une conclusion.

Nous avons utilisé le langage python pour l'implémentation de notre application, ce choix est pris parce qu'on le métrise et ce parce que ce langage est beaucoup utilisé au monde presque dans tous les domaines. Ce langage nous a fait gagner du temps (pas de compilation, un typage dynamique, une syntaxe succincte, un debugger intégrer). Grâce à la simplicité de python, la lisibilité du code est plus puissante, en plus, il y a beaucoup d'expérience derrière ce langage et son environnement ''Python est là depuis longtemps, et pour longtemps [25] '' c'est la preuve qu'il est capable de tenir la longueur. Et il a une forte relation avec l'environnement LINUX (l'environnement utilisé).

Notre application fournit un aperçu général sur les méthodes de partitionnement de graphe, elle permet d'illustrer ces avantages à l'aide de différents API (Application Programming Interface) utilisées.

2 Outils de développements :

L'environnement de développement comporte un ensemble d'outils qui permet d'augmenter la productivité des programmeurs qui développent des logiciels. Pour ce faire nous avons utilisé python avec Pycharm.

2.1 Python :

Python est un langage puissant, à la fois facile à apprendre et riche en possibilités. Il dispose de nombreuses fonctionnalités intégrées au langage. Il est, en outre, très facile d'étendre les

fonctionnalités existantes [26]. Ainsi, il existe ce qu'on appelle des bibliothèques qui aident le développeur à travailler sur des projets particuliers. Plusieurs bibliothèques sont installées pour, développer nos interfaces graphiques en python telles que : (os, tkinter, time, rdflib, graph, matplolib. pyplot, networkx, counte ,et graphx).

Concrètement, voilà ce qu'on a pu faire avec python :

- De petits programmes très le chargement des données et des graphes sémantiques
- Des programmes complets, comme le transfert des données sémantiques en un seul graphe.
- Des projets très complexes, comme la distribution du traitement.

Voici quelques-unes des fonctionnalités que nous a été offertes par python et ses bibliothèques :

- Créer des interfaces graphiques.
- Faire circuler des informations au travers d'un réseau.
- Dialoguer d'une façon avancée avec le système d'exploitation.

Remarque : le choix du langage de programmation influence dans notre cas sur le temps du traitement des données sémantiques.

2.2 Pycharm :

PyCharm est l'environnement de développement que nous avons intégré pour programmer en python. Il offre l'analyse de code, un débogueur graphique, la gestion des tests unitaires, l'intégration de logiciel de gestion de versions, et supporte le développement web avec Django [27].

PyCharm est développé par l'entreprise tchèque JetBrains. Il est multi-plateforme et fonctionne sous Windows, Mac OS X et Linux. Il est décliné en édition professionnelle, réalisé sous licence propriétaire, et en édition communautaire réalisée sous licence Apache.

3 Le système d'exploitation :

Le système d'exploitation nous a permis l'utilisation des ressources matérielles de deux ordinateurs. Pour ce faire nous avons utilisé Ubuntu une distribution Linux open source de type Unix basée sur Debian. Ubuntu est considéré comme une bonne distribution pour les débutants. Le système d'exploitation était principalement destiné aux ordinateurs personnels PC. Linux est utilisé parce qu'il est un système d'exploitation complet. Il comprend une interface utilisateur, un environnement graphique X Windows System, une connectivité TCP/IP, l'éditeur Emacs et d'autres composants que l'on retrouve généralement dans un système Unix exhaustif [28].

Caractéristique des machines :

Dans l'implémentation, nous avons utilisé deux PC, chacune caractérisé par :

- Processeur : Intel (R) Core (TM) i3-3110M CPU @ 2.40GHz 2.40GHz.
- Mémoire installée : 8.00 GB (7.88GB usable).
- Type de système : 64- bits système d'exploitation, x64- processeur basé.
- Processeur : Intel (R) Core (TM) i5-5200M CPU @ 2.20GHz 2.20GHz.
- Mémoire installée : 6.00 GB
- Type de système : 64- bits système d'exploitation, x64- processeur basé.

4 Apache Spark :

Spark présente plusieurs avantages par rapport aux autres technologies big data et MapReduce comme Hadoop et Storm. D'abord, Spark propose un Framework complet et unifié pour répondre aux besoins de traitements Big Data pour divers jeux de données, divers par leur nature (texte, graphe, etc.) aussi bien que par le type de source (batch ou flux temps-réel). Ensuite, Spark permet à des applications sur clusters Hadoop d'être exécutées jusqu'à 100 fois plus vite en mémoire, 10 fois plus vite sur disque. Il vous permet d'écrire rapidement des applications en Java, Scala ou Python et inclut un jeu de plus de 80 opérateurs haut-niveau. De plus, il est possible de l'utiliser de façon interactive pour requêter les données depuis un Shell [29].

Apache Spark est un système informatique en grappes rapide et polyvalent. Il fournit des API de haut niveau en Java, Scala, Python et R, ainsi qu'un moteur optimisé prenant en charge les graphiques d'exécution généraux. Il prend également en charge un ensemble complet d'outils de niveau supérieur, notamment Spark SQL pour SQL et le traitement de données structurées, MLlib pour l'apprentissage automatique, GraphX pour le traitement de graphes et Spark Streaming [30].

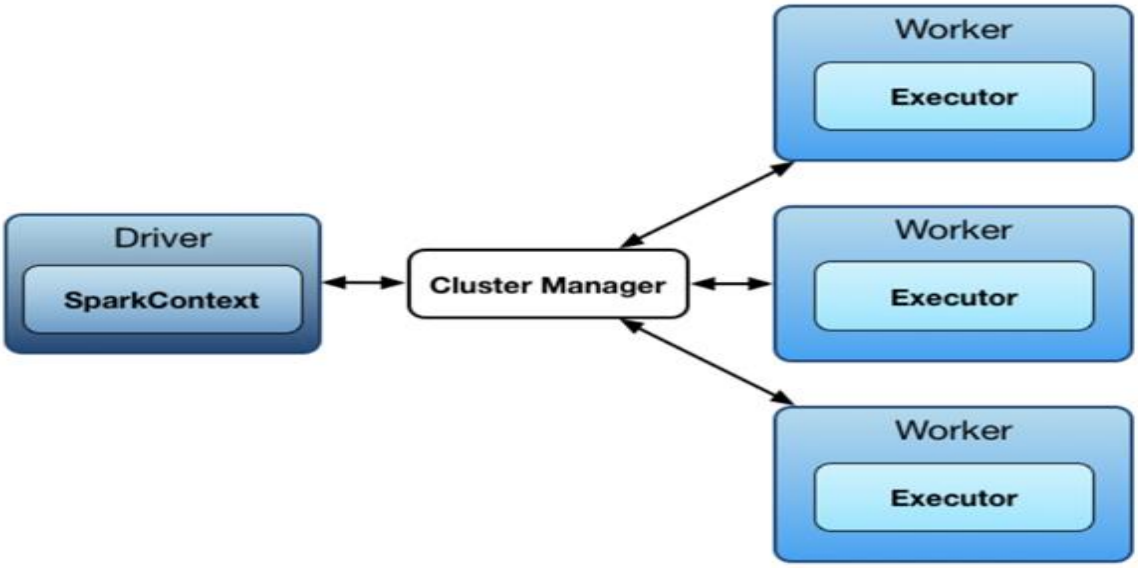


Figure 29 : Architecture Apache Spark.

Spark s'exécute en mode maître esclave, c'est-à-dire un master et un ou plusieurs workers. On peut exécuter des programmes en mode standalone scheduler (le mode natif qui gère un cluster Spark) ou bien en se basant sur un cluster manager qui gère les ressources du cluster (Yarn, Mesos ou Kubernetes) [31].

5 Bases de données et requête :

La base de données est la collection de nos données sémantique stockées dans des fichiers et accessibles à la demande pour plusieurs utilisateurs et des besoins divers. Pour notre modèle on a utilisé 3 différentes bases de données (DBPedia, BSBM, LUBM). Chaque BDD lui est attribuée un ensemble de requêtes.

5.1 La bases de données DBPédia :

DBpedia est un projet universitaire et communautaire d'exploration et extraction automatique de données dérivées de Wikipédia. Son principe est de proposer une version structurée et sous forme de données normalisées au format RDF des contenus encyclopédiques de chaque page de Wikipédia. Il existe plusieurs versions de DBpedia et dans plusieurs langues. Les trois versions principales[f][g][h] sont la version anglaise (http://dbpedia.org/sparql), la versions française (http://fr.dbpedia.org) et la version allemande (http://de.dbpedia.org/) [32].

5.2 La bases de données LUBM :

The Lehigh University Benchmark (LUBM). L'université de Lehigh est développée pour faciliter l'évaluation des référentiels de Web sémantique de manière standard et systématique. L'indice de référence est destiné à évaluer les performances de ces référentiels par rapport à de grands ensembles de données. Il comprend un domaine technique, des données synthétiques personnalisables et répétables, un ensemble de requêtes de test et plusieurs métriques de performance [33].

5.3 La bases de données BSBM :

Le Berlin SPARQL Benchmark (BSBM) est un benchmark permettant de comparer les performances des systèmes de stockage qui exposent des points de terminaison SPARQL. Ces systèmes comprennent les magasins RDF natifs, les magasins Name Graph, les systèmes mappant des bases de données relationnelles dans RDF et les wrappers SPARQL autour d'autres types de sources de données. L'indice de référence est construit autour d'un scénario d'utilisation du commerce électronique, dans lequel un ensemble de produits est proposé par différents fournisseurs et par lequel les consommateurs ont publié des avis sur les produits [34].

6 Système de traitement de données sémantique optimisé :

Notre application définit un ensemble de traitements accessible par des opérations déclenchés sur les documents RDF dont le but est de montrer les avantages d'une méthode de partitionnement de graphe.

Figure 30: Fenêtre principale d'application.

Chaque menu effectue une tâche bien définie, on distingue aussi deux emplacements textuelles destinée pour l'affichage de documents RDF se trouve dans un fichier externe. Le premier menu « Fichier » permet à l'utilisateur de charger un document parmi les documents présents dans la base.

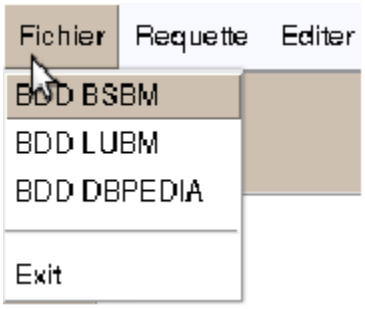


Figure 31 : Menu fichier

Le menu « Requête » permet à l'utilisateur de choisir une requête qui est adapter au document RDF choisi.

Figure 32: Menu requête.

Pour convertir en graphe, un bouton « Convert » est affiché pour visualiser la document RDF, un bouton « vider » est affiché pour vider les champs, comme le montre la figure ci-dessus.

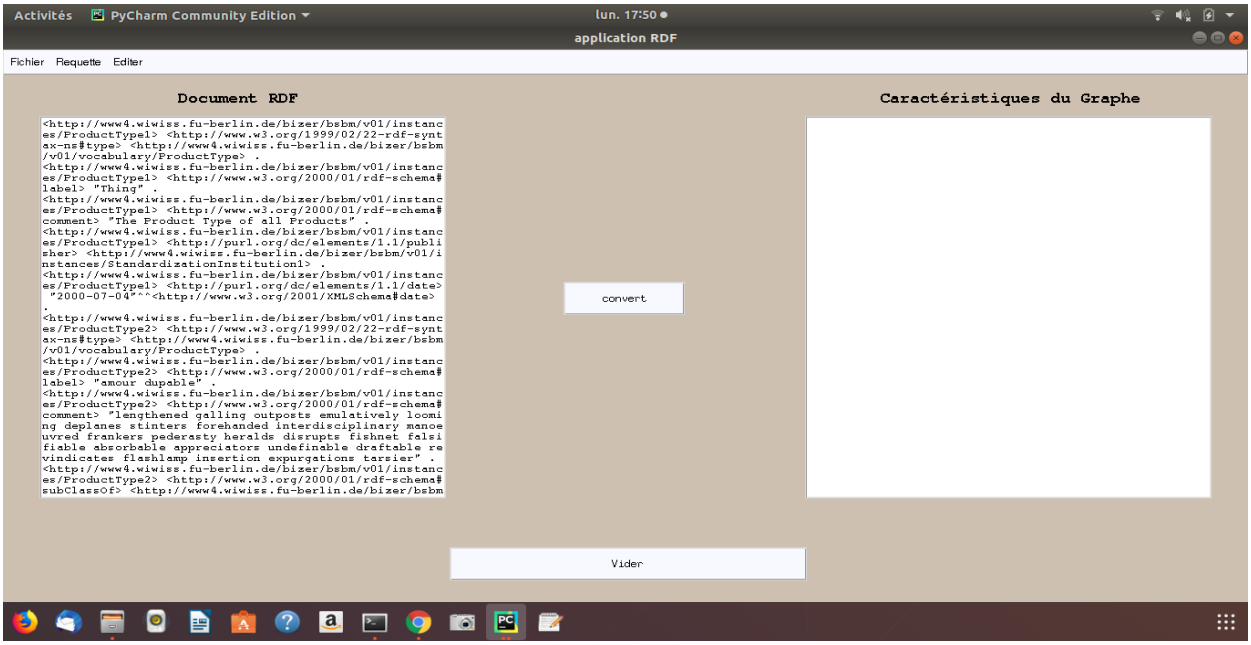


Figure 33 : Fenêtre pour affiche le document RDF

Voici le résultat du graphe du document RDF chargé ;

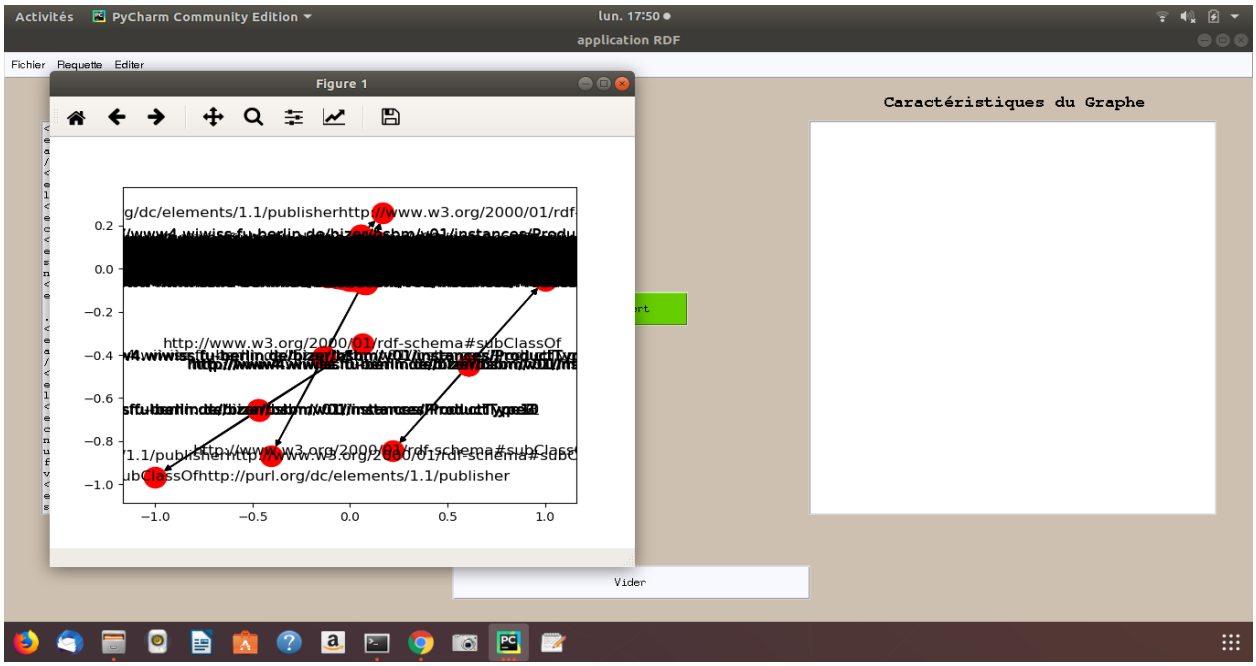


Figure 34 : Fenêtre pour affiche le résultat graphique du document RDF

Et afficher la statistique du graphe obtenu

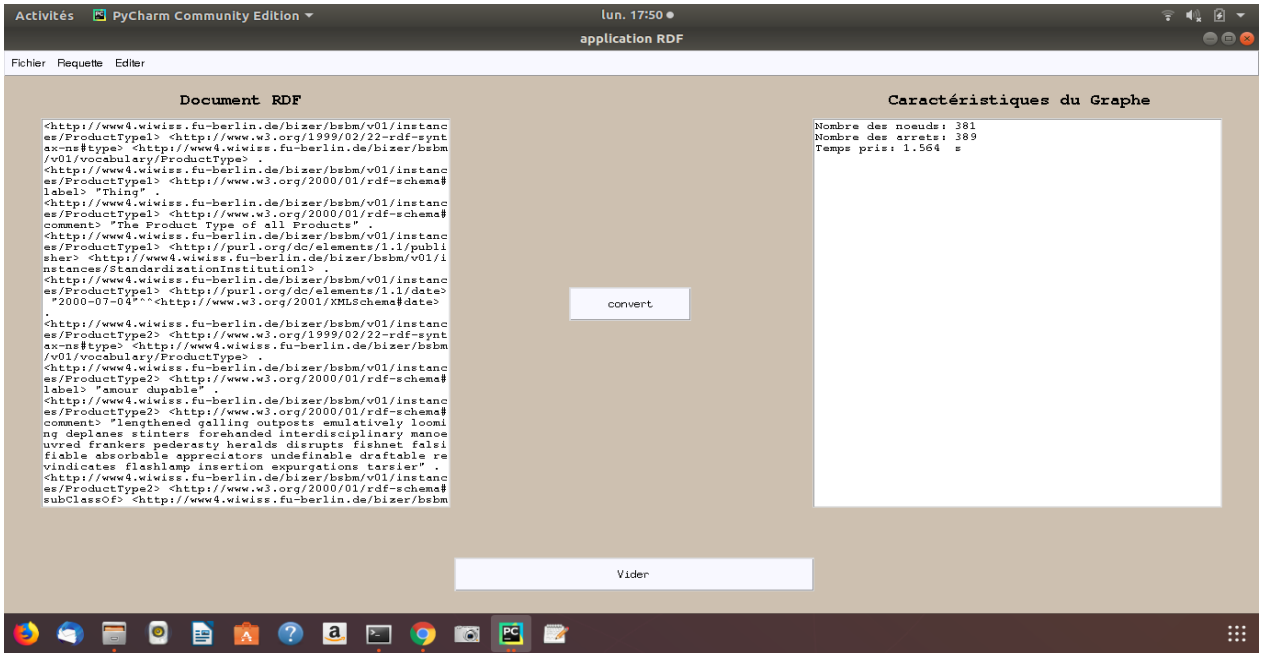


Figure 35 : Caractéristique du graphe

Qu'on choisît la requête, une boite de dialogue apparaît lorsqu'on appuis sur le bouton oui elle permet d'ouvrir la 2eme fenêtre

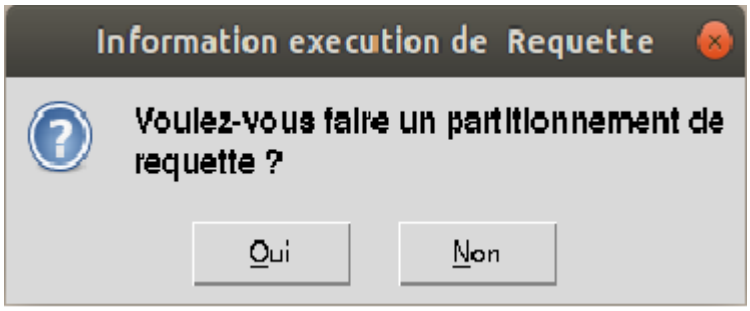


Figure 36 : Choix du traitement de la requête

La 2eme fenêtre

Pour exécuter la requête, un bouton « Convertir » est affiché pour visualiser le résultat, un bouton « vider » est affiché pour vider les champs, ensuite, Un bouton « statistique » est affiché pour visualiser les caractéristiques du graphe.

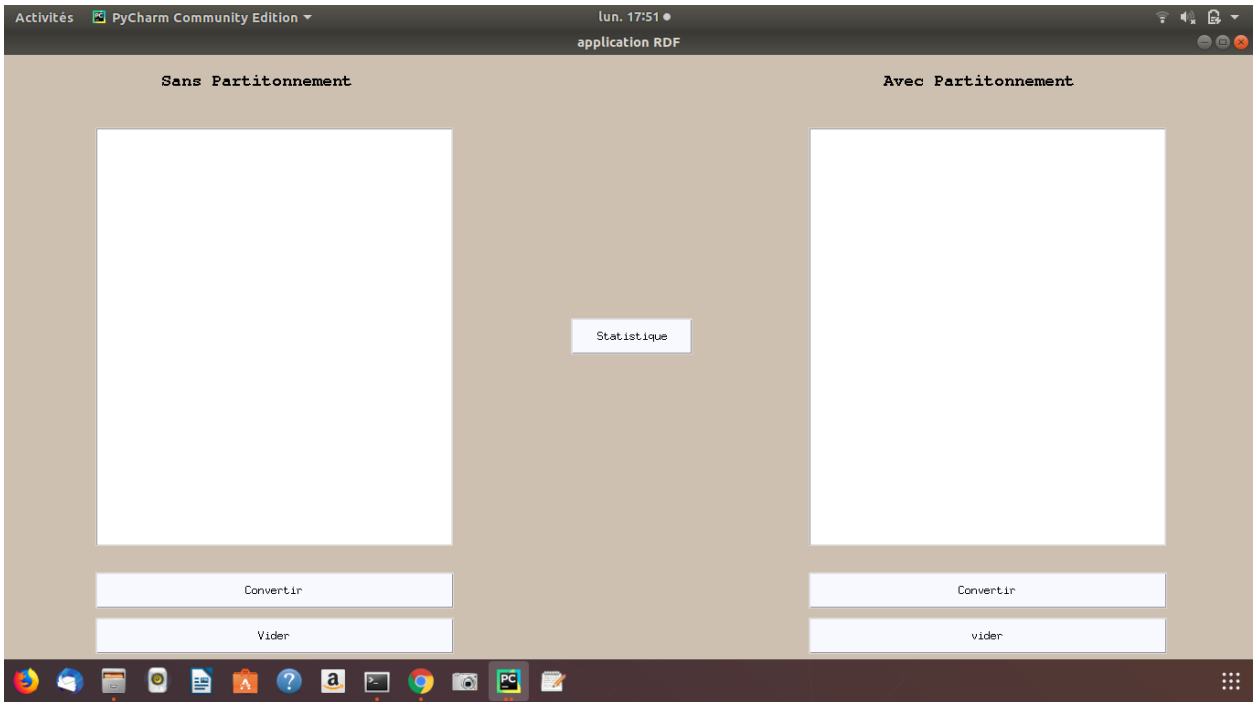


Figure 37: Partitionnement des requêtes

Quand nous cliquons sur le bouton convertir pour visualiser les résultats de la requête sans partitionnement

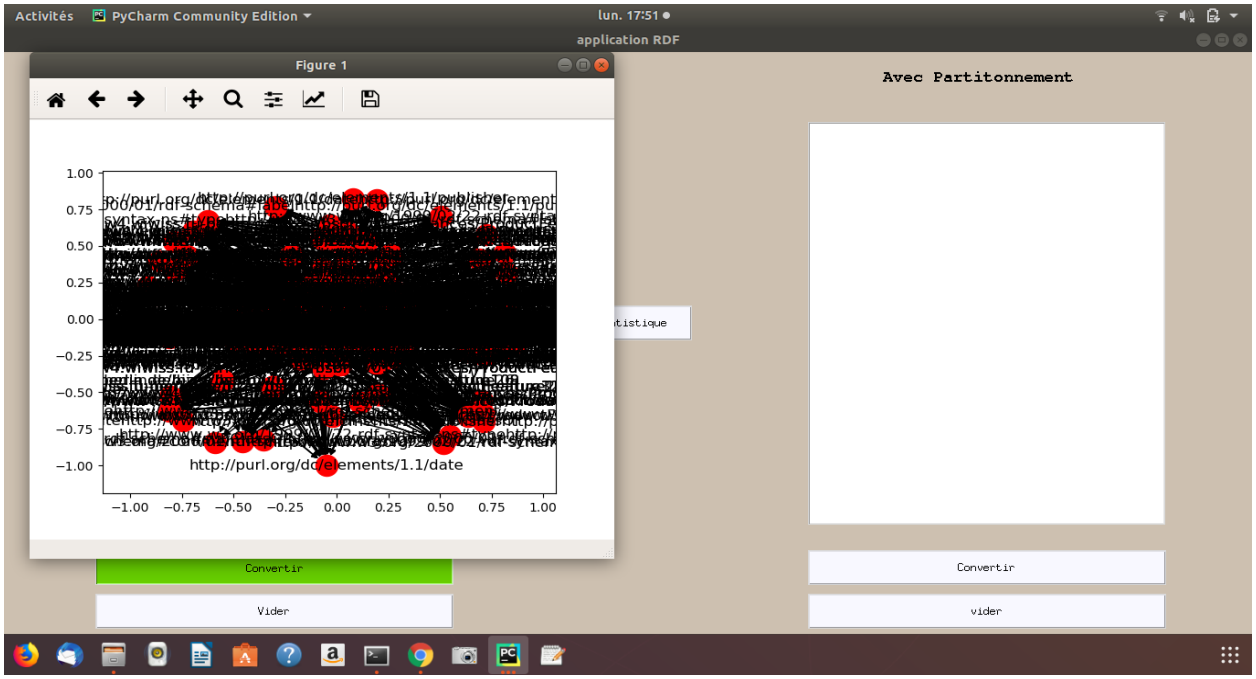


Figure 38 : Résultat graphique de la requête sans partitionnement

Et afficher le statistique du graphe obtenu

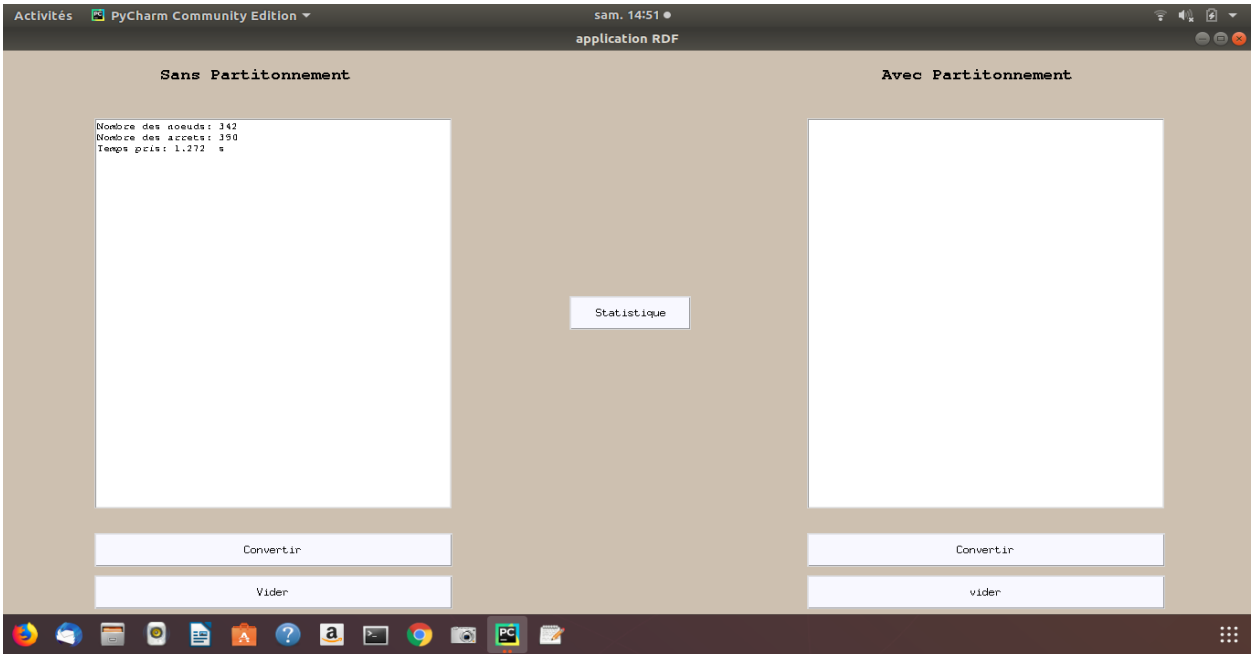


Figure 39 : Statistiques de la requête sans partitionnement

Quand nous cliquons sur le bouton convertir pour visualiser les résultats de la requête avec partitionnement

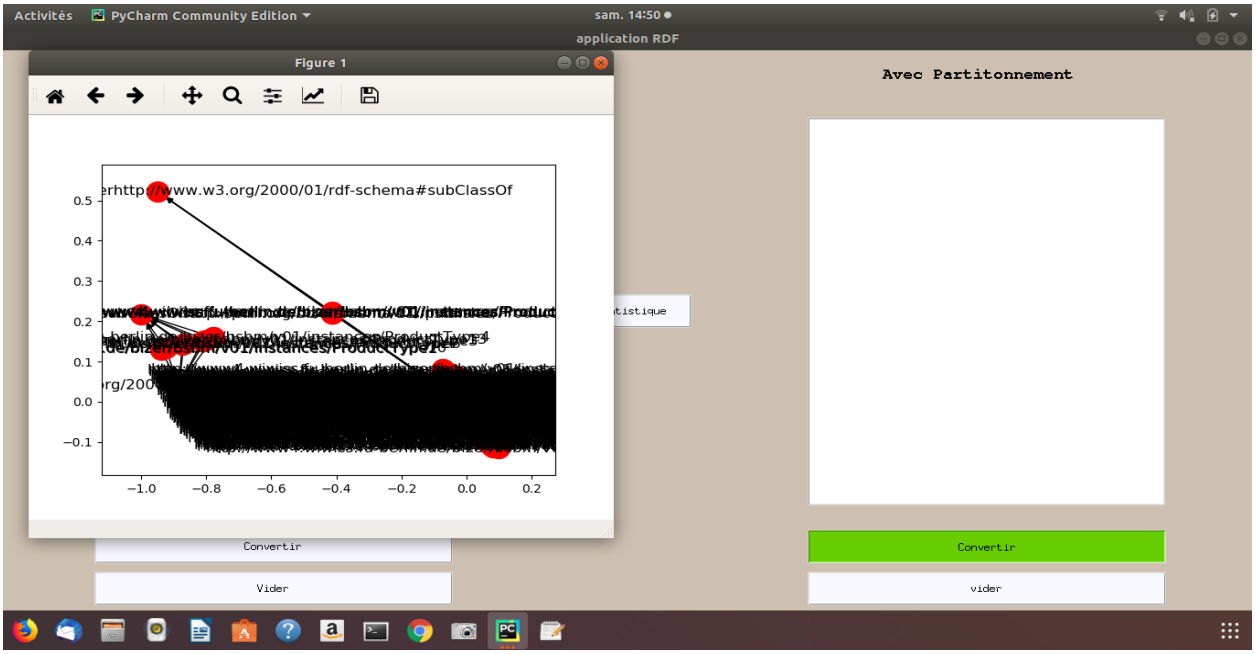


Figure 40: Résultat graphique de la requête avec partitionnement

Et afficher le statistique du graphe obtenu

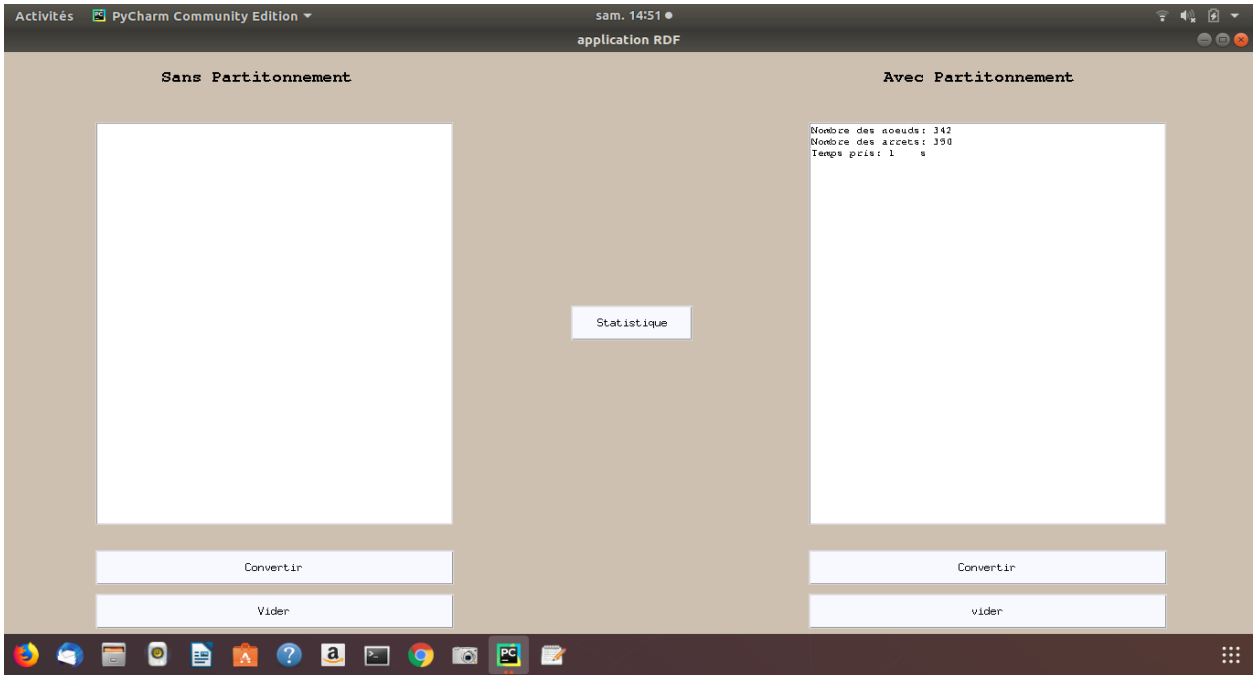


Figure 41 : Statistiques de la requête avec partitionnement

Le résultat obtenu sera un graphe comme le résultat précédent mais la différence entre les deux méthodes dans le temp exécution car le temp d'exécution de la méthode avec partitionnement est moins importante que sans partitionnement.

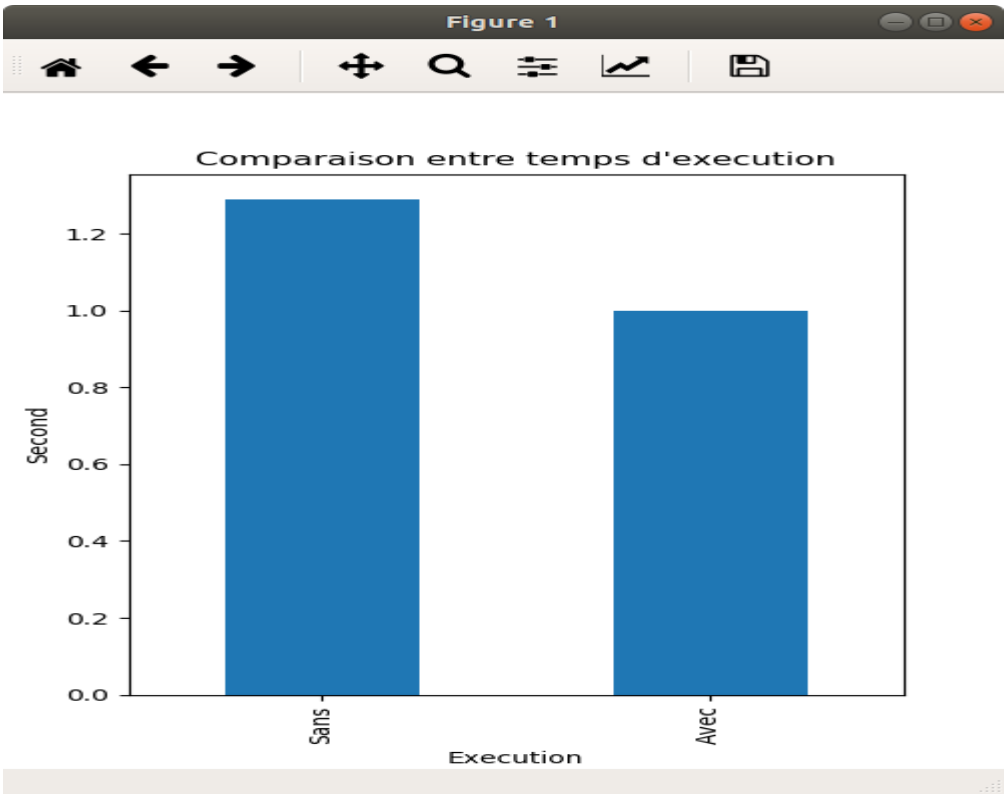


Figure 42: Comparaison temp d'exécution entre les méthodes de partitionnement

D'après les résultats nous avons remarqué que temp l'exécution de la méthode avec partitionnement (1s) et sans partitionnement (1.2s) donc il est clair que nous avons gagné (0.2s) et sa c'est notre but.

7 Conclusion :

Dans ce chapitre nous avons présenté l'implémentation de notre modèle, nous avons utilisé le langage Python pour obtenir de bons résultats pendant l'utilisation de l'outil de partitionnement de graphe et ses statistiques pour faire la comparaison entre les deux méthodes de partitionnement pour connais la meilleure méthode entre les deux.

L'avantage de notre application utilise deux méthodes de partitionnement et faire une comparaison entre eux pour découvrir la plus rapide et performent (temps d'exécutions).

Conclusion générale :

Le web sémantique, propose une nouvelle plateforme permettant une gestion plus intelligente du contenu, à travers sa capacité de manipuler les ressources sur la base de leurs sémantiques. Le web sémantique favorise ainsi les coopérations Homme/Machine et permet de s'ouvrir à de nouvelles possibilités d'automatisation sur le Web.

Le Web sémantique vise à permettre aux machines d'utiliser la sémantique, c'est-à-dire la signification de l'information, sur le Web. Il étend le réseau des hyperliens entre des pages Web classiques par un réseau de liens entre données structurées, permettant ainsi à des agents automatisés d'accéder plus intelligemment aux différentes sources de données contenues sur le Web et, de cette manière, d'effectuer des tâches (recherche, apprentissage, etc.) plus précises pour les utilisateurs.

Le but de notre travail est d'orienter l'évolution du Web pour permettre aux utilisateurs sans intermédiaires de trouver, partager et combiner l'information plus facilement. Les êtres humains sont capables d'utiliser le Web pour effectuer des tâches telles que trouver le mot 'Mostaganem' pour réserver un livre à la bibliothèque, trouver un plan et réserver son billet de transport. Cependant, les machines ne peuvent pas accomplir toutes ces tâches sans direction humaine, parce que les pages web sont conçues pour être lues uniquement par des personnes et non par des machines. Aussi le Web sémantique est critiqué à cause de sa lourdeur : les langages utilisés pour le Web sémantique sont très verbeux, car dérivés du XML et donc souvent pénibles à utiliser. De ce fait, l'écriture d'ontologies est souvent très problématique, car elle exige une spécialisation dans un domaine particulier et lorsque l'on ne maîtrise pas ce domaine, elle devient très difficile à créer. Ainsi, certaines personnes disent qu'il est préférable d'utiliser des « Word tags » (ce sont une série de mots clés qui permettent de qualifier une ressource) à la place des ontologies.

Ainsi, puisque les machine comprenne plus les formes graphiques, nous avons opté pour transformer les documents sémantiques en graphe.

Nous avons adopté un moyen simple de donner un sens à des graphes sémantiques relativement complexes. Ce sens n'inclut pas les cas de quantificateurs ramifiés, mais en

contrepartie, la formule logique produite (ou les formules, en cas de sous-spécification) est à même d'être utilisée dans un solveur de la logique du premier ordre. Les perspectives de ce travail incluent notamment une analyse des hypothèses de bonne formation d'un graphe prédicatif quantifié, une étude plus poussée de la construction des informations de restriction à partir de la syntaxe, ainsi qu'un traitement des déterminants complexes. À plus long terme, il serait intéressant d'étudier dans quelle mesure des inférences logiques standards pourraient être effectuées directement sur la structure de graphe sémantique.

Pour réaliser la transformation de documents sémantiques en graphes, notre application passe par plusieurs étapes selon le modèle proposé. Initialement, il faut charger un document sémantique, ce dernier provient d'une base de données interne. Une fois le chargement effectué on a la possibilité de le visualiser sous une syntaxe XML et le transformer ensuite en graphe. Si notre document chargé a déjà été traité, l'application affichera la représentation XML et graphique du document.

Perspectives

Les perspectives liées à ce travail consistent à traiter le résultat de ce travail, c'est-à-dire le traitement des graphes sémantiques, en exécutant des requêtes sur ces graphes afin d'accéder aux données pour ajouter, supprimer, ou modifier. Ensuite, il est possible de lancer les requêtes sur les documents sémantiques avant la conversion en graphe et de comparer le temps d'exécution des requêtes lancées sur les documents et le temps d'exécution des requêtes lancer sur le graphe.

Bibliographies:

[1] S. Antipolis, Research Challenges and Perspectives of the Semantic Web, http://www.ercim.org/EU-NSF/semweb.html, 2001.

[2] P. Hitzler, M Krotzsch, Fondation of Semantic Web Technologie&, Sebastian Rudolph, Chapman and Hall/CRC, 2009.

[3] J. Domingue, D. Fensel, J. A. Hendler, Handbook of Semantic Web Technologie, Springer-Verlag Berlin Heidelberg, 2011.

[4] World Wide Web Consortium (W3C), w3c.org, Consulté le 01 Mars 2019.

[5] Définition de web sémantique par Tim Berners-Lee président de W3C.

[6] Lee Feigenbaum, « The Semantic Web in Action » Scientific American, 1er mai

[7] World Wide Web Consortium (W3C), OWL Web Ontology Language Overview, http://www.w3.org/TR/2004/REC-owl-features-20040210/, 10 février 2004.

[8] World Wide Web Consortium (W3C), Resource Description Framework, https://www.w3.org/2001/sw/wiki/RDF, 25/02/2014,

[9] E.Simperl, Reusing ontologies on the Semantic Web: A feasibility study ; Data & Knowledge Engineering, Volume 68, Issue 10, Pages 905-925, October 2009.

[10]: Government of Canada, Public Works and Government Services Canada, Translation Bureau, TERMIUM, http://www.btb.termiumplus.gc.ca,

[11] W3C, Recommendation: Extensible Markup Language (XML) 1.0 (Fifth Edition), http://www.w3.org/TR/2008/REC-xml-20081126/, 26 Novembre 2008.

[12] CNAM Centre associé de Clermont-Ferrand Cycle A – Année 1997-98J. Darmont

[13] https://www.lemagit.fr/definition/NoSQL-base-de-donnees-Not-Only-SQL.

[14] http://blog.jemsdatafactory.com/2016/09/23/les-bases-de-donnees-nosql-cle-valeur-et-la-base-cassandra-open-source/

[15] BIG DATA https://www.lebigdata.fr/definition-big-data.

[16] https://www.futurasciences.com/tech/definitions/informatique-big-data-15028/

[17] Émilie Baro, Vers une définition des Big data en santé basée sur la littérature, thèse de doctorat, Université Lille 2 droit et santé Faculté de médecine Henri Warembourg, 11 mai 2015 à 16h, 69 pages

[18] Jean-Louis Monino, Big data open data et valorisation des données, article, Réseau de Recherche sur l'innovation. France, 2015, 17 pages

[19] Jean-Louis Ermine et Al, une chaîne de valeur de la connaissance, article, Management international, 7 Février 2016 04 :29, p. 29-40.

[20] (J.Darmont, Bases de données, 1997-98)

[21] (2018, aout 04). Récupéré sur wikipedia : https://fr.wikipedia.org/wiki/Langage_de_requ%C3%AAte

[22] Andy Seaborne, H.-P. L. (2008, janvier 15). Récupéré sur http://www.yoyodesign.org/doc/w3c/rdf-sparql-query/#defn_algOrdered

[23] Cavalié, E. (2012, novembre 22). Récupéré sur https://bibliotheques.wordpress.com/2012/11/22/utiliser-les-sparql-endpoint-comme-si-cetait-des-api/

[24] developpez.com. (s.d.). Récupéré sur https://web-semantique.developpez.com/tutoriels/jena/arq/introduction-sparql/

[25] http://sametmax.com/10-raisons-pour-lesquelles-je-suis-toujours-marie-a-python/

[26] https://openclassrooms.com/fr/courses/235344-apprenez-a-programmer-en-python/230659-decouvrez-python

[27] https://fr.wikipedia.org/wiki/PyCharm

[28] https://www.universalis.fr/encyclopedie/systemes-d-exploitation-informatique/

[29]https://www.infoq.com/fr/articles/apache-spark-introduction/

[30]https://spark.apache.org/docs/latest/

[31]https://meritis.fr/bigdata/larchitecture-framework-spark/

[32] http://fr.dbpedia.org/sparqlTuto/tutoSparql.html#/h.i9fnke60pvyd

[33] http://swat.cse.lehigh.edu/projects/lubm/

[34] https://virtuoso.openlinksw.com/bs/bs-sparql-results-04-04-2013