Qualitative analysis of

RDF, Property Graph, and Domain Graph data models

using Wikidata

Hina Shahzad

Thesis submitted for the degree of
Master in Informatics: Programming and System
Architecture 60 credits

Department of Informatics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Autumn 2022

Qualitative analysis of RDF, Property Graph and Domain Graph data models

using

Wikidata

Hina Shahzad

Abstract

Knowledge Graphs (KG) have been widely popular for presenting real-world entities as nodes and edges according to semantic web rules and regulations. Many organizations and industries have used Knowledge Graphs (KGs) for publishing their datasets according to Linked Open Data (LOD) principles. Many Graph data models, e.g., RDF, Property Graph, and Domain Graph data model, have been used to model real-world entities as Knowledge Graphs (KGs). However, each data model has represented the knowledge differently, which sometimes affects the performance of the Knowledge Graph, especially in data storage and retrieval. The selection of an exemplary graph data model for representing Knowledge Graphs (KGs) plays a vital role in extracting and integrating data from various sources.

Wikidata (Vrandečić & Krötzsch, 2014) is a Knowledge Graph representing real-world entities and connecting them to Wikipedia articles. Wikidata entities are defined by the Pages, describing the information as statements. Each statement has some additional information, e.g., qualifiers and references. Wikidata is one of the most extensive Knowledge Graphs where the data is updated daily. Hence, the representation of Wikidata entities in the different graph data models is challenging and costly in terms of data storage and data retrieval. So, the thesis represents the Wikidata in three graph data models, e.g., RDF, Property Graph, and Domain Graph, and does a qualitative analysis of three graph data models by conducting comparison and describe their advantages and disadvantages.

The RDF data model represents Wikidata as triples (subject-predicate-object) and uses RDF reification to model Wikidata complex statements. The property Graph data model uses node and edge labels to represent Wikidata entities and model the complex statement as edge attributes and a compact data model. The Domain Graph data model uses the edges as nodes to model Wikidata statements. The RDF reification lacks internal structure and generates many redundant triples, which increases the data storage and reduces the query response time. The Property Graph data model needs to be fully represented Wikidata statements as edge attributes. However, the Domain Graph data model facilitates the edges as nodes, fully represents Wikidata statements, and provides better storage and query response time than RDF and PG. In addition, the thesis represents the general qualitative analysis between three graph data models (RDF, Property Graph, and Domain Graph), which helps the readers to select the best graph data model for modeling Knowledge Graphs (KGs).

Acknowledgments

I would like to thank Francisco Martin -Recuerda and Dr. Arne J. Berre for providing sincere guidance and help in various ways for preparing my thesis. Your availability for regular meetings has been beneficial for me. I would also like to extend my gratitude to Dr. Dumitru Roman for his feedback on the thesis.

I am profoundly grateful to SINTEF for providing me with the resources to work on my thesis and the University of Oslo (UiO) for providing the best learning environment.

Finally, I would like to thank my husband, my mother, and my children for supporting me throughout the study.

Hina Shahzad
November 2022

Contents

List of Figures

List of Tables

Abbreviations and Acronyms

KGs	Knowledge Graphs
RDF	Resource Description Framework
PG	Property Graph
DG	Domain Graph
LOD	Linked Open Data
W3C	World Wide Web Consortium
YAGO	Yet Another Great ontology
RDFS	Resource Description Framework Schema
IRIs	International Resource Identifiers
TSV	Tab Separated Values
KGTK	Knowledge Graph Tool Kit
OWL	Ontology Web Language

Part I
Introduction and Background

Chapter 1

Introduction

1.1 Context

Graphs data models have been used to store real-world entities. Considerable advancement has been made in graph databases. Many companies and industries have managed their Knowledge Graphs (KGs) and information by using the different features of graph data models. Several Knowledge Graphs (KGs), i.e., Wikidata (Vrandečić & Krötzsch, 2014), DBpedia (Lehmann et al., 2015), YAGO (Hoffart et al., 2011), Bing (Shrivastava, 2017), Google (Singhal, 2012), and many others have been developed since the 1980s, which have been used to represent the data and knowledge as a graph. In addition, to manipulate the Knowledge Graphs (KGs), several powerful hardware has been used to store and retrieve a piece of knowledge from them efficiently. In addition, many robust sensors and machines have been established to present and analyze information from Knowledge Graphs (KGs) (Angles & Gutierrez, 2017) (Hogan et al., 2022).

A Knowledge Graph (Hogan et al., 2022) is an example of the graph and has been widely used to model real-world entities where each node represents an entity, and the edges have defined the relationship between the two nodes. Different methods, e.g., human-driven, semiautomated, and fully automated, have been proposed to add the information to Knowledge Graphs (KGs). The goal is to provide the information in Knowledge Graphs (KGs) in such a manner that is understood by humans. In recent era, searching and storing information in the KGs is a big challenge. Different graph data models have been widely used to model knowledge graphs, e.g., the Resource Description Framework (RDF) (Hogan, Arenas, Mallea, & Polleres, 2014), Property Graph data model (Angles, 2018), and the Domain Graph data model (Vrgoč et al., 2021). Each model has advantages and disadvantages when representing the knowledge graphs (Chaudhri et al., 2022).

In addition, Wikidata (Vrandečić & Krötzsch, 2014) as Knowledge Graph (KG) representing real-world entities such as Wikipedia articles. Each page is defined against each Wikipedia article and the pages are represented as Wikidata entities. In addition, Links have been established from Wikidata entities page to Wikipedia articles and have been available in more than 36 million languages. Wikidata has been developed since October 2012, has proliferated, and more than 45,6 million contributors have been actively involved with Wikidata. There are more than 14.5 million entities represented by Wikidata. Entities have been defined by the statements and more than 30 million statements have been developed (Malyshev, Krötzsch, González, Gonsior, & Bielefeldt, 2018). Furthermore, modeling the Wikidata in a different graph data model is a big challenge. Each data model represents Wikidata differently, affecting the performance of Wikidata, especially data storage and retrieval (Chaudhri et al., 2022). This thesis places itself to analyze three graph data models (RDF, Property Graph, and Domain

Graphs) for Wikidata. It explores which graph data model is more suitable in order to represent Wikidata statements, qualifiers, and references within the given context.

1.2 Motivation

The value, volume, and significance of the knowledge graphs have increased during the last decades, affecting the complexity of the knowledge graphs in research and business areas. With the introduction of Google knowledge graphs (Uyar & Aliyu, 2015) in 2012, the popularity of knowledge Graphs (KGs) has increased. Many communities have selected Knowledge Graphs (KGs) to distribute their dataset by using the Linked Open Data (LOD) principles (Ehrlinger & Wöß, 2016).

The Knowledge Graph (KG) closely relates to semantic web technologies and linked data. Several large projects are based on Linked Open Data (LOD) by using the semantic web technology and datasets are published on the web. Furthermore, the Resource Description Framework (RDF) is a data model in order to model the Knowledge Graphs using semantic web technology and to publish them according to the Linked Open Data principles. It is also recommended by the World Wide Web Consortium (W3C) (Fernández-Álvarez, Frey, Labra Gayo, Gayo-Avello, & Hellmann, 2021) (Berners-Lee, Hendler, & Lassila, 2001).

Wikidata as a knowledge graph is popular among others because it provides the feature to integrate and organize information by sharing its resources openly on the World Wide Web (Chaudhri et al., 2022). In addition, Wikidata is widely prevalent among other knowledge graphs and is being used by many industries for intelligent assistance, information retrieval and knowledge integration (Guo et al., 2022). Representing massive datasets e.g., Wikidata in a graph data model, is quite a big challenge for developers. The selection of the correct data model is also a challenge for the communities before publishing the datasets on the web because the datasets are often updated over time. Ensuring the correct and updated information is being published at the right time is a challenge for developers and communities (Melnik, Mitra & Decker, 2000).

Consequently, the selection of a data model has played a vital role in representing a knowledge graph and is being challenged by the communities and organizations. They want a data model representing their Knowledge Graphs (KGs) according to semantic web rules and regulations. In addition, they want a data model which consumes the minimum data storage and fastest data retrieval. Many graph data models, e.g., Resource Description Framework (RDF) (Hogan, Arenas, Mallea, & Polleres, 2014), Property Graph data model (Angles, 2018), and Domain Graph data model (Vrgoč et al., 2021), have provided different features in order to model Knowledge Graphs (KGs). However, on the other side, they have limitations and incompleteness while representing the Knowledge Graphs (KGs) (Melnik, Mitra & Decker, 2000).

1.3 Problem statement

Representing Knowledge Graphs (KGs), e.g., Wikidata, different graph data models, is challenging because each data model has different expressivity. For instance, the RDF data model represents information using triples of the form (subject, predicate, object) (Hogan, Arenas, Mallea, & Polleres, 2014). The Property Graph data model allows the possibility to add attribute-value pairs to nodes and edges (Angles et al., n.d.). The Domain Graph data model represents statements using quads (Ilievski et al., 2020) (Angles et al., n.d.).

Based on the given context of the thesis, there is a need to examine which graph data model is the best choice for representing information and particularly the statement-level data which is very important for Wikidata.

The following research questions have been formulated considering the problem statement associated with the data model: -

1. Which of the three graph data models is more suitable for representing the information stored in Wikidata?
2. What are the main advantages and disadvantages of each of the three data models?

1.4 Thesis scope

The thesis aims to present the main differences between RDF, Property Graph, and Domain Graph data models and evaluates their advantages and disadvantages when representing the information in Wikidata due to Wikidata being very big. A small subset of the graph was selected involving only the statement related to a single entity. Each subgraph was manually edited to produce a specific version for each data model. The differences between subgraphs were analysed to establish relevant advantages and disadvantages of each data model.

1.5 Research methodology and work plan

In addition to an evaluation based on the ability of each data model to represent Wikidata statements, a qualitative analysis was also conducted. This analysis is based on the opinions of different authors discussing each data model (RDF, Property Graph, and Domain Graph) (Solheim & Stølen, 2007). During the preparation of this thesis, the following steps were completed:

Problem statement: This step includes understanding the notion of knowledge graphs and relevant data models, including RDF, Property Graphs and Domain Graphs. A literature review was conducted, and relevant inputs have been summarized in Chapter 2.

Experiment on Wikidata: During this step, an experimental setup was created. The original idea of working with a complete version of Wikidata was discarded due to the inability of the actual representation of the knowledge graph, e.g., Wikidata in graph data models (RDF, Property Graph, and Domain Graphs). It also presents each model by discussing their process for Wikidata.

Qualitative Evaluation: In this step, qualitative evaluation based on a literature review was conducted. The qualitative analysis compares the data models RDF, Property Graph and Domain Graph.

1.6 Thesis outline

This thesis includes the following five additional chapters.

Chapter 2: Background This chapter introduces the relevant notion for understanding the other two chapters. It introduces the data models RDF, Property Graph and Domain Graph. It also describes Wikidata knowledge graph and how information is represented.

Chapter 3: Analysis of RDF, PG, and DG for modelling information in Wikidata This chapter includes an analysis of the suitability of each data model considered in this thesis to represent information in Wikidata. The focus has been made in the ability to represents Wikidata statements.

Chapter 4: Qualitatively Analysis of RDF, PG and DG This chapter compares and describes the advantages and disadvantages of each data model based on the input collected during the literature review.

Chapter 5: Conclusion and Future Work This chapter summarizes the main contributions of the thesis and discusses future work.

Chapter 2

Background

This chapter briefly introduces the three graph data models analysed in the thesis: RDF, Property Graph and Domain Graph. In addition, the chapter includes a short introduction of Wikidata, a reference knowledge graph created and maintained by Wikimedia Foundation. This chapter illuminates core terms that help to better understand the material presented in the following chapters. This chapter has been divided into four sections. Section one explains the RDF data model; Next section two discusses the Property Graph data model; Next section three discusses the Domain Graph data model. To conclude, Section four introduces Wikidata, and in particular, the representation of Wikidata statements.

2.1 The Semantic Web and Knowledge Graphs (KGs)

According to Lampropoulos, Keramopoulos, and Diamantaras (2020), semantic knowledge is vital in enhancing the research quality in a knowledge graph. At the same time, the knowledge graph combines data and semantics that collects information from various sources and produces new knowledge by analysis and reasoning. According to Buchgeher, Gabauer, Martinez-Gil, and Ehrlinger (2021), Knowledge Graph (KG) consists of nodes and edges where each node is an entity and defines the relationship between each entity. Knowledge graphs are widely popular nowadays and used in different domains. Wikidata (Vrandečić & Krötzsch, 2014) is an example of a well-adopted knowledge graph that defines real-world entities as labeled nodes and Wikidata properties as labeled edges that define relationships between real-world entities.

The semantic web (Berners-Lee et al., 2001) aims to give meaning to the information, and its purpose is to enable automate information processing. Hence, the objective of the Semantic web is to develop a knowledge management system that organizes the knowledge according to its meaning (Antoniou & Van Harmelen, 2004).

Kroetzsch and Weikum (2016) define the knowledge graph as follow.

> *"Knowledge graphs are large networks of entities, their semantic types, properties, and relationships between entities."*

2.2 Graph database model

A graph database model (Angles et al., 2018) is used to model the data structures for the schema as a graph and represents the information and the knowledge in the form of real-world entities. The graph database model is very good at handling unstructured data, especially interconnected data. In addition, the nodes represent the complete information about the entity and are interconnected with the edges, forming a graph. There are different types of graphs, e.g., directed and undirected graphs, graphs labeled on nodes and edges, hypergraphs, and hypernodes. The most straightforward graph is a plain labeled graph; however, the hypergraphs are used to model the complex relations, and hypernodes are used to model the nested graphs inside the nodes. The RDF data model and the Property Graph data model are label-directed graphs, and the Domain Graph is an example of a hypergraph. Different query languages are used to interact or to extract the data from the different graph data models (Angles & Gutierrez, 2008), (Angles & Gutierrez, 2017).

2.3 Section I

2.3.1 RDF data model

The RDF data model is a graph data model defined by the World Wide Web Consortium (W3C) for exchanging data on the web (Gutierrez et al., 2011).

Hogan, Arenas, Mallea, and Polleres (2014) defines the RDF formal definition:

> *"Pairwise disjoint sets U (URIs), L (Literals), and B (Blank nodes). UB represents the union of U and B. An RDF triple is a tuple (s, p, o) ∈ UB × U × UBL where s is called the subject, p the predicate, and o the object. "*

Further, Hayes (2004) defines the formal syntax of the RDF graph by explaining the mapping function.

> "***M*** *is a mapping function from a set of blank nodes to some set of literals, blank nodes, and URI references; then any graph obtained from a graph G by replacing some or all the blank nodes* ***N*** *in* ***G*** *by* ***M(N)*** *is an instance of* ***G****."*

Figure 1 shows the graphical representation of the formal syntax of the RDF graph (Hayes, 2004).

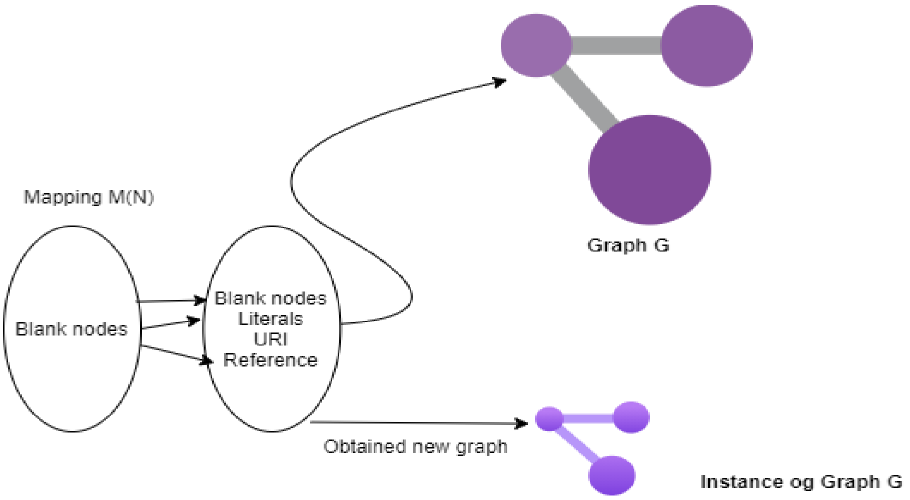


Figure 1: Formal syntax of RDF graph (Hayes,2004).

RDF (Resource Description Framework) is a simple data model created to define statements about resources. Resources can be human, object, location etc. The resources are uniquely identified using International Resource Identifiers (IRIs) [1] ("IRIs/RDFConceptsProposal - RDF Working Group Wiki", 2022). The statements are defined using triples which consist of a subject, a predicate, and an object. Each triple defines a logical statement between two resources or between a resource and a literal which defines the resources datatypes in form of strings. A subject in the RDF data model is represented by either a resource or an anonymous object also known as blank node (Hogan, Arenas, Mallea, & Polleres, 2014). A predicate can only be represented by a resource. An object can be represented either as a resource or as a literal. RDFS and OWL are two standards that define modelling languages to represent more complex relations between resources. This includes, for instance, sub-class and sub-property relations (Gutierrez et al., 2011).

2.3.2 Blank nodes

Hogan, Arenas, Mallea, and Polleres (2014) formally defines a blank node as follows:

> *"Let G be an RDF graph and I = (Res, Prop, Ext, Int) be a simple interpretation. Let A: B → Res be a function from blank nodes to resources and let Int_A denote an amended version of Int that includes B as part of its domain such that $Int_A(x) = A(x)$ for $x \in B$ and $Int_A(x) = Int(x)$ for $x \in UL$. We say that I is a model of G if I is an interpretation over voc(G) and there exists a mapping A such that for each (s, p, o) $\in$ G, it holds that Int(p) $\in$ Prop and ($Int_A(s)$, IntA(o)) $\in$ Ext (Int(p))."*

Blank nodes are unnamed resources that can occur either on the subject position or on the object position in the RDF triples. Blank nodes cannot represent the predicates. The purpose of the Blank nodes in the RDF data model is to break down the binary relation and provides the facility to define the structural information about the resources on the web and usually use to model complex statements or statements overstatements. Table 1 shows the example of an RDF graph which shows the simple statement (subject-predicate-object), and Table 2 shows the decomposition of Table 1 into many triples by using a blank node to display the RDF graph in a more suitable structure (Manola & Miller, 2014), (Angles & Gutierrez, 2017).

Abraham	address	"Jacobine Ryes vie, 0976 Oslo Akershus."

Table 1: RDF simple statement (Manola & Miller, 2014).

Abraham	address	_: AbrahamAdress
_: AbrahamAdress	street	"Jacobine Ryes vie"
_: AbrahamAdress	city	"Oslo."
_: AbrahamAdress	state	"Akershus."
_: AbrahamAdress	postalCode	"0976"

Table 2: n-ary relations using Blank node (Manola & Miller, 2014).

[1] https://www.w3.org/2011/rdf-wg/wiki/IRIs/RDFConceptsProposal

Figure 2 displays the dual representation of the RDF data model. The first diagram shows a simple RDF graph, where Abraham is the subject, address is the predicate, and “Jacobine Ryes vie, 0976 Oslo Akershus” is the object. The second diagram, a blank node defines n-ary relation. So, _: Abraham is the blank node, and through this blank node, the different elements of the address are represented separately, e.g., postal code, street, city, and state.

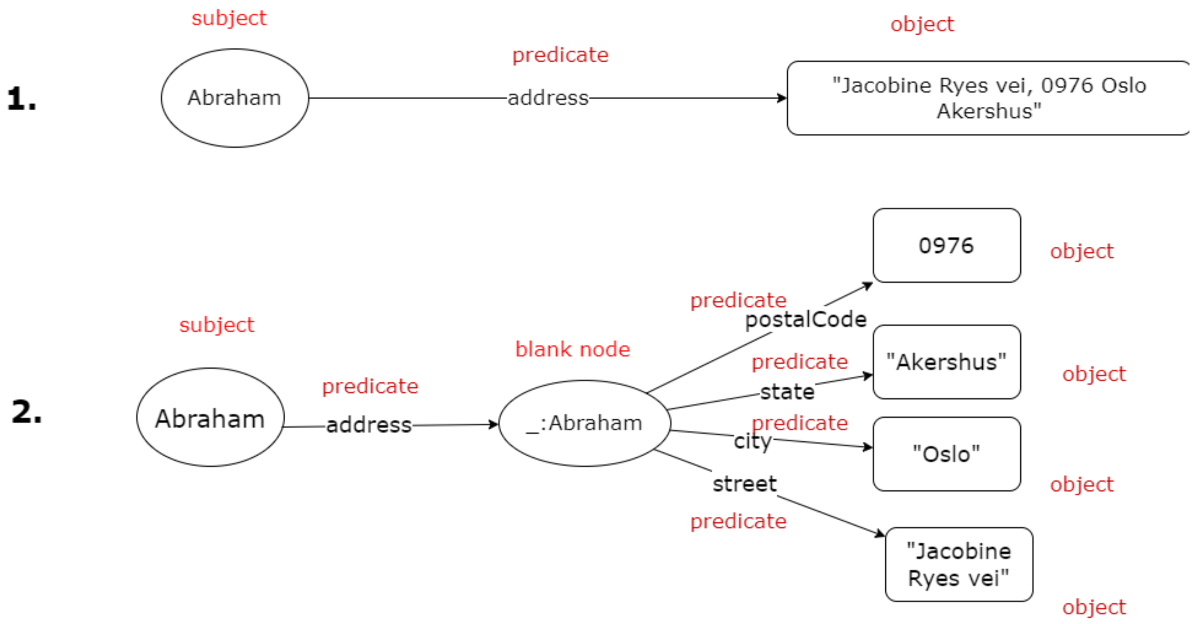


Figure 2: Dual representation of the RDF data model (Angles & Gutierrez, 2017).

RDF Schema (or RDFS) is an extension of RDF, which provides additional vocabulary to represent more complex relations. With RDFS is possible to define classes and properties, which are also defined in terms of classes. The definition of properties might include a domain and a range. Figure 3 includes some reference classes and properties defined by the standard RDFS.. The yellow nodes refer to classes, and the red nodes refer to properties in Figure 3 (Brickley, Guha, & McBride, 2014).

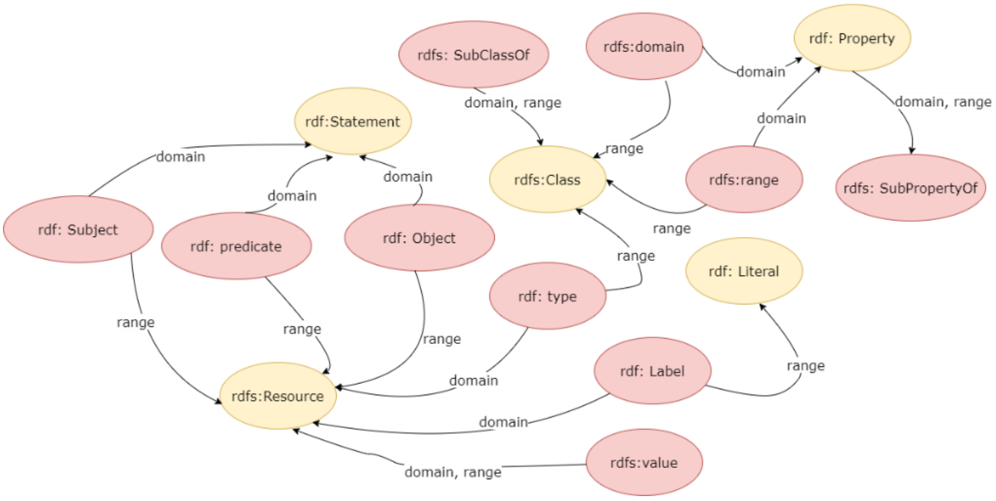


Figure 3: RDF Schema classes and properties (Brickley, Guha, & McBride, 2014).

2.3.3 OWL (Web Ontology Language)

OWL is an extension of RDF Schema; it is a modelling language and a recommendation of the World Wide Web Consortium (W3C). OWL defines the knowledge structure in various domains by defining the classes, subclasses, properties, and sub-properties like RDFS and introduces many more new constructors. It provides formal semantics for the meaningful representation of data (McGuinness & Harmelen, 2004).

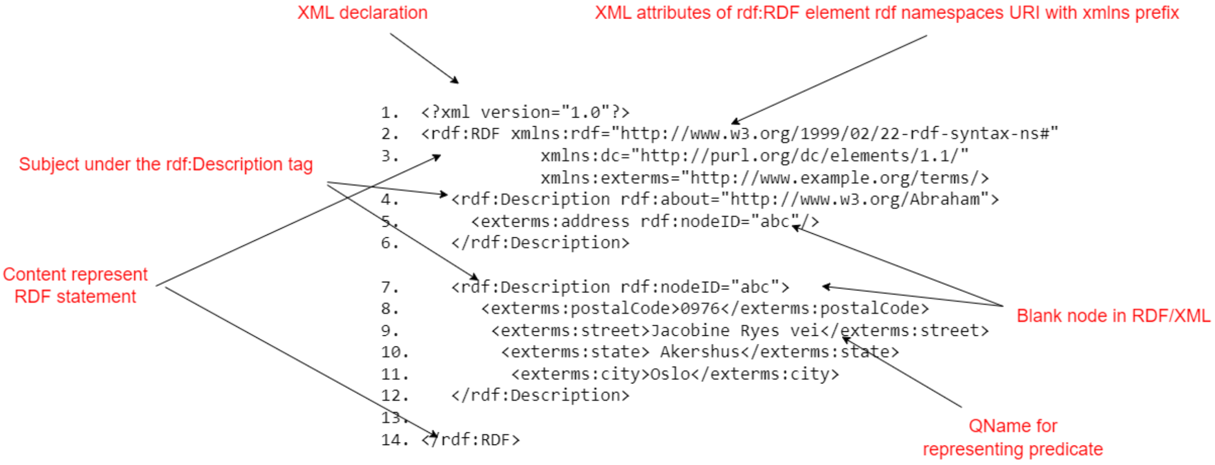
2.3.4 RDF serialization

There are many serialization formats available which can be used to write RDF graphs in a file. The different RDF serialization formats presented in this chapter are defined as recommendations by the World Wide Web Consortium (W3C).

2.3.4.1 RDF/XML

The RDF/XML format encodes the RDF triples in XML form. This is the first serialization format defined as a recommendation by the World Wide Web Consortium (W3C) recommendation. QNames and namespaces were included following the XML specification. In addition, RDF/XML facilitates the blank nodes and the complex statements using reification (Manola & Miller, 2014), (Hayes & Patel-Schneider, 2014).

Figure 4 shows an example of RDF graph represented using RDF/XML format by Manola and Miller (2014). The example includes some blank nodes.



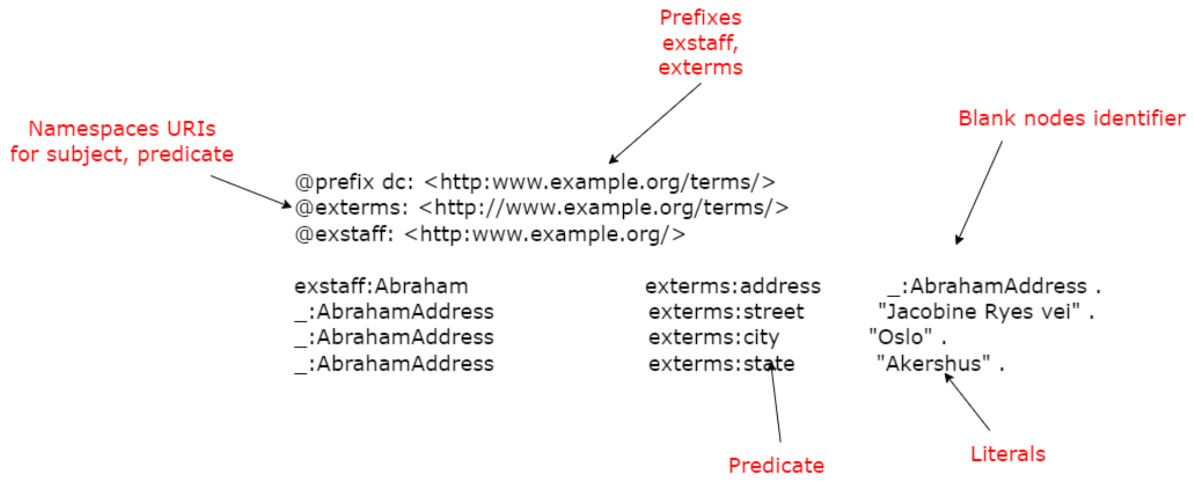
```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
            xmlns:dc="http://purl.org/dc/elements/1.1/"
            xmlns:exterms="http://www.example.org/terms/>
   <rdf:Description rdf:about="http://www.w3.org/Abraham">
     <exterms:address rdf:nodeID="abc"/>
   </rdf:Description>

   <rdf:Description rdf:nodeID="abc">
      <exterms:postalCode>0976</exterms:postalCode>
       <exterms:street>Jacobine Ryes vei</exterms:street>
        <exterms:state> Akershus</exterms:state>
         <exterms:city>Oslo</exterms:city>
   </rdf:Description>

</rdf:RDF>
```

XML declaration
XML attributes of rdf:RDF element rdf namespaces URI with xmlns prefix
Subject under the rdf:Description tag
Content represent RDF statement
Blank node in RDF/XML
QName for representing predicate

Figure 4: Example of an RDF graph represented using RDF/XML (Manola & Miller, 2014).

2.3.4.2 Turtle

This is another serialization format that is more user-friendly than the RDF/XML format. It is also a recommendation of the World Wide Web Consortium (W3C) (Manola & Miller, 2014), (Hayes & Patel-Schneider, 2014). Figure 5 shows the example of RDF graph in Figure 2 using the turtle syntax instead.

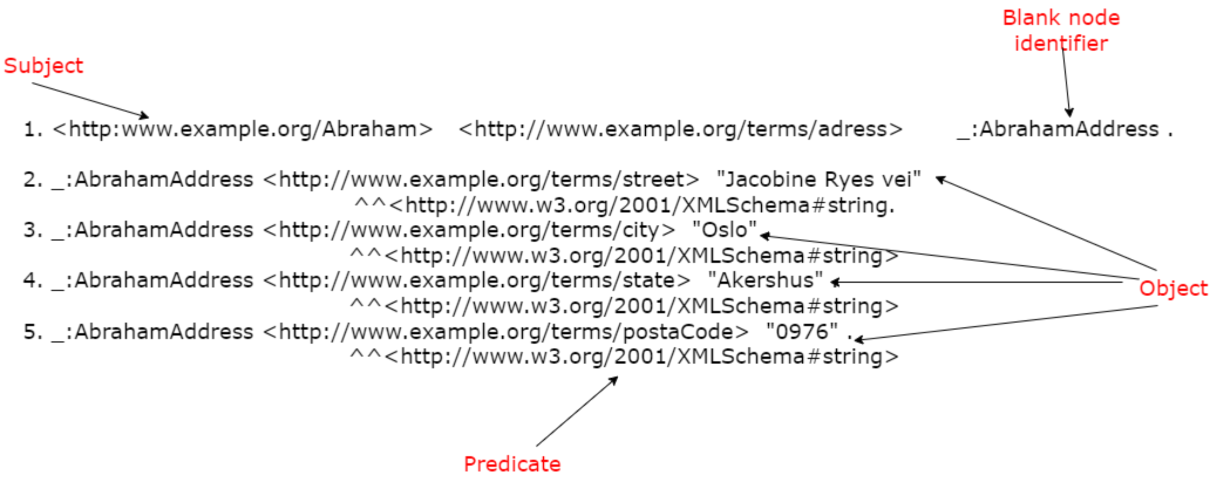


```
@prefix dc: <http:www.example.org/terms/>
@externs: <http://www.example.org/terms/>
@exstaff: <http:www.example.org/>

exstaff:Abraham              externs:address      _:AbrahamAddress .
_:AbrahamAddress              externs:street     "Jacobine Ryes vei" .
_:AbrahamAddress              externs:city      "Oslo" .
_:AbrahamAddress              externs:state      "Akershus" .
```

Figure 5: Example of an RDF graph represented using Turtle format (Manola & Miller,2014).

2.3.4.3 N-Triples

This serialization format uses URIs instead of prefixes (Manola & Miller, 2014), (Hayes & Patel-Schneider, 2014). Figure 6 shows the N-Triples representation of figure 2.

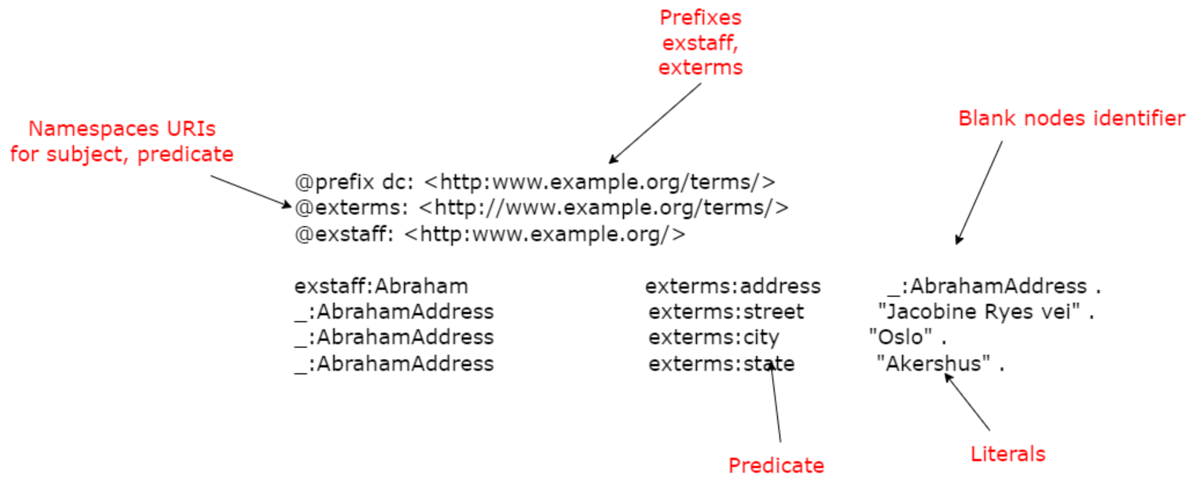


```
1. <http:www.example.org/Abraham>   <http://www.example.org/terms/adress>      _:AbrahamAddress .

2. _:AbrahamAddress <http://www.example.org/terms/street>  "Jacobine Ryes vei"
                              ^^<http://www.w3.org/2001/XMLSchema#string.
3. _:AbrahamAddress <http://www.example.org/terms/city>  "Oslo"
                              ^^<http://www.w3.org/2001/XMLSchema#string>
4. _:AbrahamAddress <http://www.example.org/terms/state>  "Akershus"
                              ^^<http://www.w3.org/2001/XMLSchema#string>
5. _:AbrahamAddress <http://www.example.org/terms/postaCode>  "0976" .
                              ^^<http://www.w3.org/2001/XMLSchema#string>
```

Figure 6: Example of an RDF graph represented using N-Triple (Manola & Miller, 2014).

2.3.4.4 N3 format

It is similar to turtle format and easy to read. N3 notation and turtle serialization format supports the namespaces, QNames and URIs and has an independent language. It represents RDF triples using the prefixes, (Manola & Miller, 2014), (Hayes & Patel-Schneider, 2014). Figure 7 is the N3 serialization format of figure 2.



```
@prefix dc: <http:www.example.org/terms/>
@exterms: <http://www.example.org/terms/>
@exstaff: <http:www.example.org/>

exstaff:Abraham              exterms:address     _:AbrahamAddress .
_:AbrahamAddress             exterms:street     "Jacobine Ryes vei" .
_:AbrahamAddress             exterms:city      "Oslo" .
_:AbrahamAddress             exterms:state      "Akershus" .
```

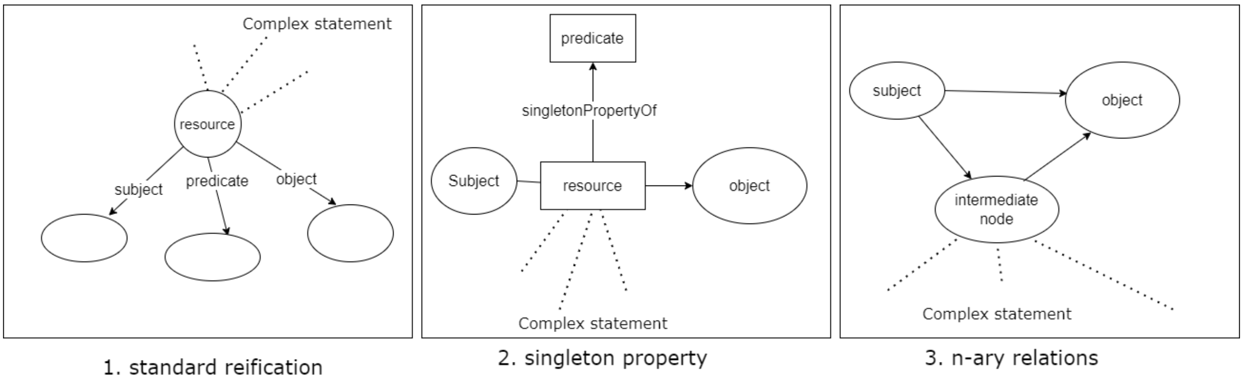
Figure 7: Example of an RDF graph represented using N3-Triples (Manola & Miller, 2014).

2.3.5 RDF reification

According to Hernández et al. (2015), RDF data model uses the binary relation between subject and object to model statements, and for complex statements, the RDF data model provides reification techniques. Three reification techniques are as follows:

- standard reification (Manola & Miller, 2014) represents a resource; these resources specify the triples (subject-predicate-object) to model complex statements.
- n-ary relations (Hernández et al., 2015) uses the intermediate resource to create a relationship with the subject, and intermediate resources further define the triples.
- singleton properties (Nguyen, Bodenreider, & Sheth, 2014) defines the unique predicates via singletonPropertyOf; resource becomes the instance of predicate and models the complex statement.

Figure 8 shows the three RDF reification techniques (Manola & Miller, 2014), (Hernández et al., 2015), (Nguyen, Bodenreider, & Sheth, 2014).



1. standard reification

2. singleton property

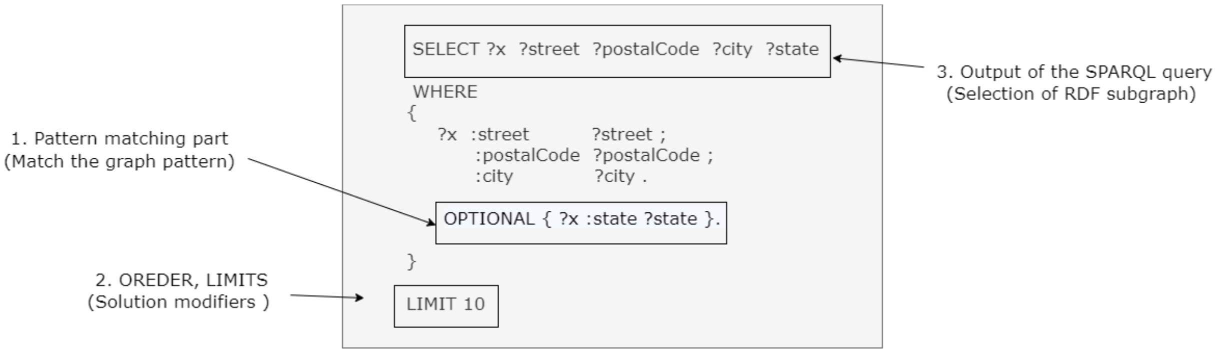
3. n-ary relations

Figure 8: Three reification techniques (Manola & Miller, 2014), (Hernández et al., 2015), (Nguyen, Bodenreider, & Sheth, 2014).

2.3.6 SPARQL

The SPARQL query language is used to query RDF graphs and is favored for querying semantic web data. SPARQL is a W3C standard and provides three significant parts while extracting subgraphs and data from the RDF triples. First, the pattern matching part is used to match the graph patterns, e.g., OPTIONAL, Filtering values, UNIONS of patterns, etc. Second, the solution modifier takes the output/result from the pattern-matching part and applies further classical operations, e.g., projection, distinct, order, and limit. Third, the output of the SPARQL query, e.g., it can be the subgraph of RDF, selection of some values, etc., and displays the final query answer. The SPARQL query language facilitates the query of a blank node and uses the constant variable for the blank node inside the query instead of the blank node identifier. The scope of the Blank nodes in SPARQL is local to the scoping graph and does not provide the scope of a blank node outside the RDF dataset (Arenas et al., 2022), (Prud'hommeaux & Seaborne, 2008).

Figure 9 shows the SPARQL query parts to extract the RDF subgraph



```
SELECT ?x  ?street  ?postalCode  ?city  ?state
WHERE
{
    ?x  :street       ?street ;
        :postalCode  ?postalCode ;
        :city         ?city .
    OPTIONAL { ?x :state ?state }.
}
LIMIT 10
```

3. Output of the SPARQL query
(Selection of RDF subgraph)
1. Pattern matching part
(Match the graph pattern)
2. OREDER, LIMITS
(Solution modifiers)

Figure 9: SPARQL query to extract the RDF subgraph (Prud'hommeaux & Seaborne, 2008).

2.4 Section II

2.4.1 Property Graph data model

A formal definition of the Property Graph data model is proposed by (Angles, 2018): -

> "A Property Graph is a tuple G = (N, E, ρ, λ, σ) where: N is a finite set of nodes (also called vertices); E is a finite set of edges such that E has no elements in common with N; ρ: E → (N × N) is a total function that associates each edge in E with a pair of nodes in N (i.e., ρ is the usual incidence function in graph theory); λ : (N ∪ E) → SET+(L) is a partial function that associates a node/edge with a set of labels from L (i.e., λ is a labelling function for nodes and edges); σ : (N ∪E)×P → SET+(V) is a partial function that associates nodes/edges with properties, and for each property, it assigns a set of values from V."

The Property Graph data model is a labeled graph containing nodes and edges. Nodes are used to model real-world entities, and edges define their relationship. Each node and edge can have properties in the form of key-value pairs. The properties of a node are used to model metadata about entities, while the properties of an edge are used to model additional information about the relations between entities (Angles & Gutierrez, 2017).

Figure 10 shows an example of a Property Graph. The graph has five nodes and five relationships. Each node and edge have pairs of properties and values. (Angles & Gutierrez, 2017), (Rodriguez & Neubauer, 2010), (Angles, 2018).

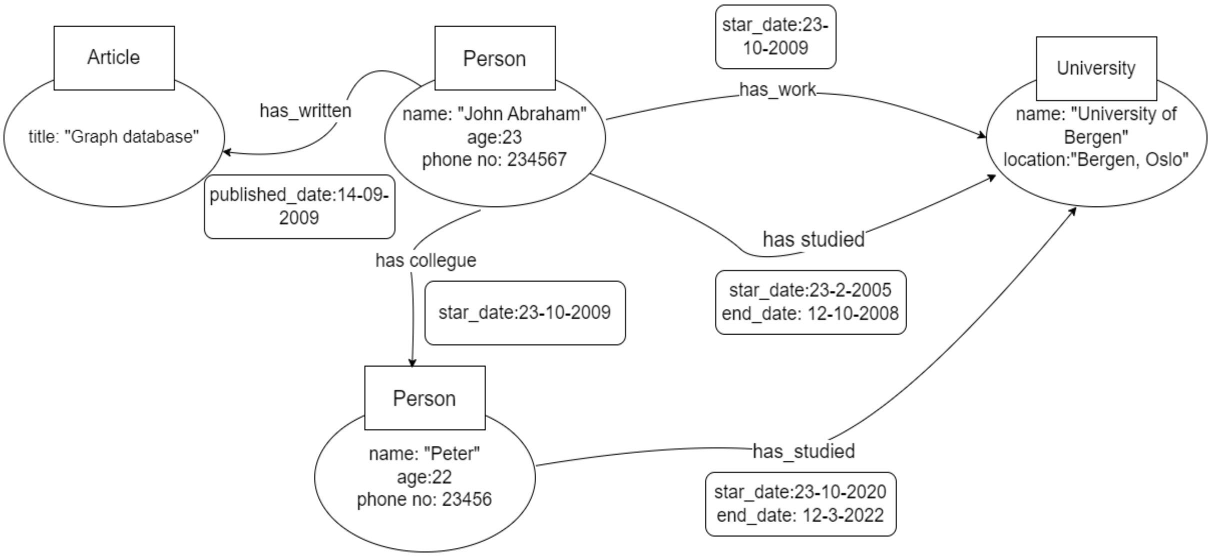
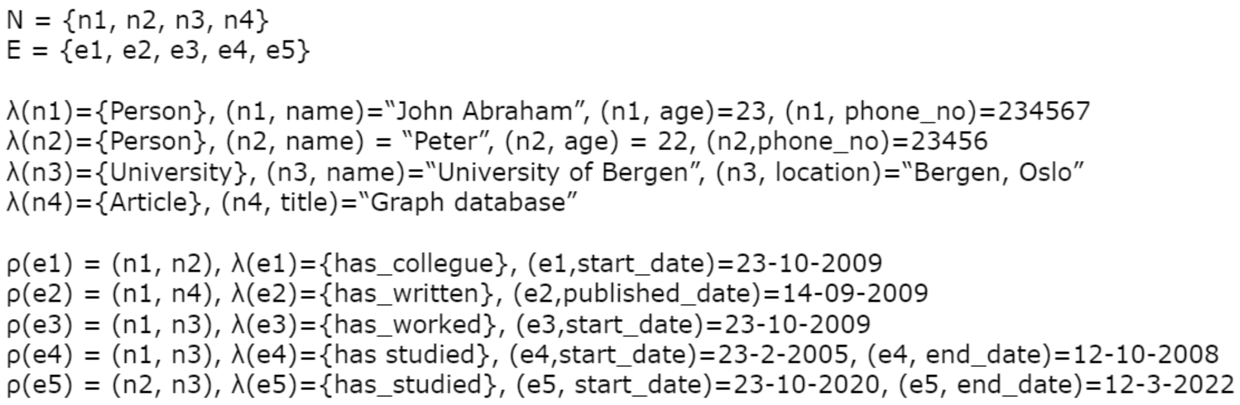


Figure 10: Example of a Property Graph (Angles & Gutierrez, 2017), (Angles, 2018).

The specification of the example presented in Figure 10 is described in Figure 11, where the set N and E represent the nodes and edges. The partial function λ: (N ∪ E) → SET+(L) defines the node label and the node properties, and the function σ: (N ∪ E) ×P → SET+(V) defines the complete path between the two nodes concerning its relationship (Angles, 2018).



N = {n1, n2, n3, n4}
E = {e1, e2, e3, e4, e5}

λ(n1)={Person}, (n1, name)=“John Abraham”, (n1, age)=23, (n1, phone_no)=234567
λ(n2)={Person}, (n2, name) = “Peter”, (n2, age) = 22, (n2,phone_no)=23456
λ(n3)={University}, (n3, name)=“University of Bergen”, (n3, location)=“Bergen, Oslo”
λ(n4)={Article}, (n4, title)=“Graph database”

ρ(e1) = (n1, n2), λ(e1)={has_collegue}, (e1,start_date)=23-10-2009
ρ(e2) = (n1, n4), λ(e2)={has_written}, (e2,published_date)=14-09-2009
ρ(e3) = (n1, n3), λ(e3)={has_worked}, (e3,start_date)=23-10-2009
ρ(e4) = (n1, n3), λ(e4)={has studied}, (e4,start_date)=23-2-2005, (e4, end_date)=12-10-2008
ρ(e5) = (n2, n3), λ(e5)={has_studied}, (e5, start_date)=23-10-2020, (e5, end_date)=12-3-2022

Figure 11: Specification of example presented in Figure 10 (Angles, 2018), (Angles & Gutierrez, 2017).

2.4.2 Property Graph schema

The schema defines the structure of the data, and the Property Graph data model uses the node type and edge type and the properties of such types to define the schema. The label defines each node and edge in the Property Graph data model; sometimes, one node may have multiple labels. The node label and edge label decide the node type in the Property Graph data model (Angles & Gutierrez, 2017), (Angles, 2018).

Figure 12 shows the schema representation of the Property Graph data model for figure 11.

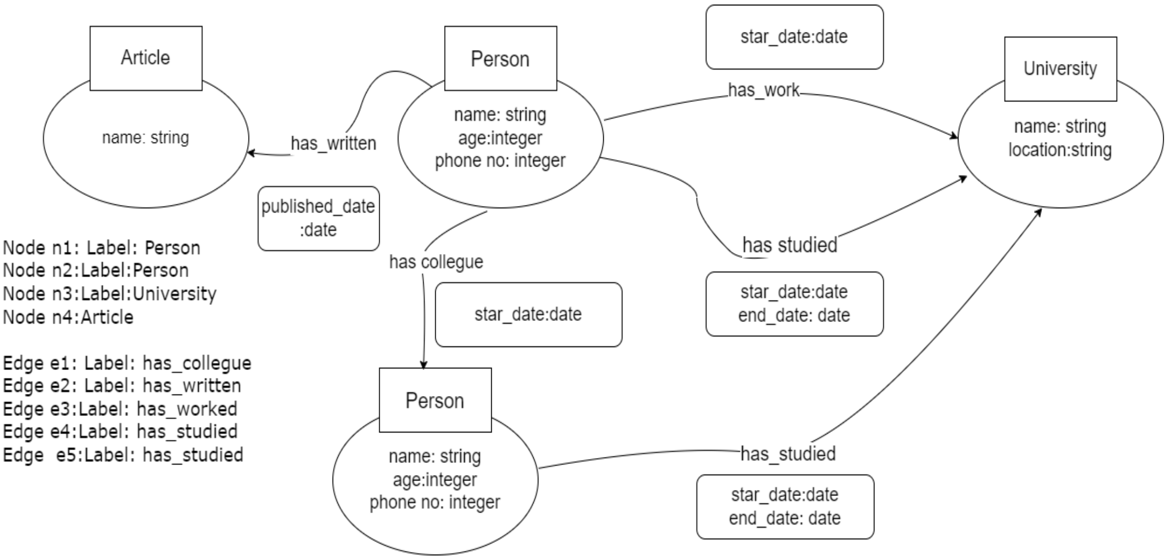
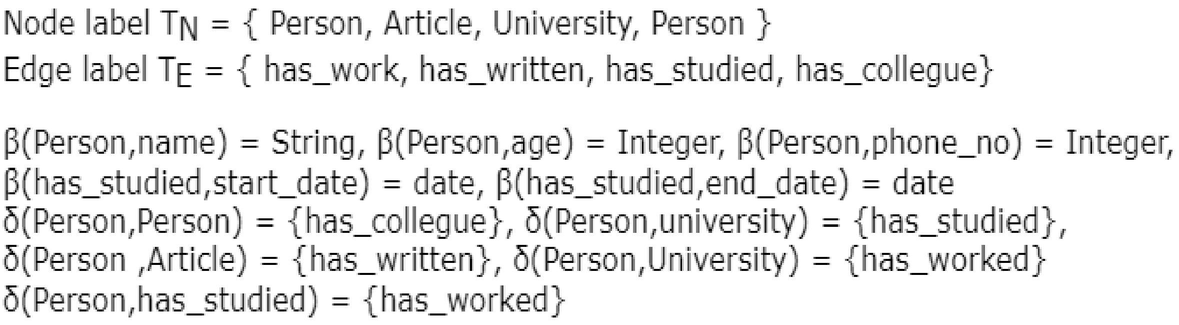


Figure 12: Example of a Property Graph schema (Angles & Gutierrez, 2017), (Angles, 2018).

According to Angles and Gutierrez (2017), the schema of a Property Graph is defined by three sets L, P, and T. L is the infinite set that defines the label of nodes and edges, P is the infinite set of properties, and T is the set which defines the data type of the values. So, the schema of the Property Graph presented in Figure 12 is represented in Figure 13 as follows.



Node label T_N = { Person, Article, University, Person }
Edge label T_E = { has_work, has_written, has_studied, has_collegue}

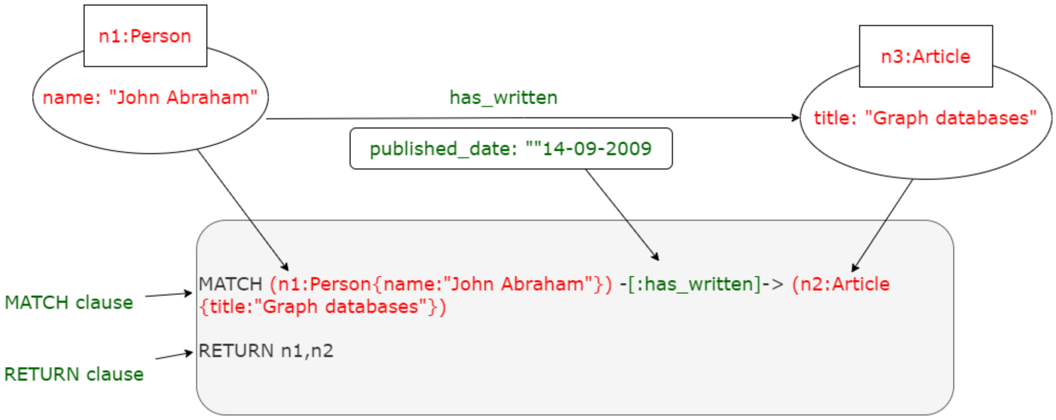
β(Person,name) = String, β(Person,age) = Integer, β(Person,phone_no) = Integer,
β(has_studied,start_date) = date, β(has_studied,end_date) = date
δ(Person,Person) = {has_collegue}, δ(Person,university) = {has_studied},
δ(Person ,Article) = {has_written}, δ(Person,University) = {has_worked}
δ(Person,has_studied) = {has_worked}

Figure 13: Schema representation of the example presented in Figure 12 (Angles, 2018).

2.4.3 Cypher

Cypher [2] (Cypher query language - developer guides, 2022) is a query language to extract graphs from the Property Graph data model. It is a declarative language and uses patterns to determine nodes and edges. Cypher uses the MATCH and RETURN clause to select the nodes and the relationship between them and return the graphs. Figure 10 defines (n1: Person); n1 is the node labeled Person. Similarly, [e1: has_written] defines the edge e1 with the label has_written. Cypher uses the "." to extract the properties of nodes and edges. In addition to MATCH and RETURN clauses, Cypher supports UNION, DIFFERENCE, OPTIONAL, CREATE, and DELETE clauses to operate on the Property Graph (Angles et al., 2018).

Figure 14 shows the Cypher query for Figure 10.



```
MATCH (n1:Person{name:"John Abraham"}) -[:has_written]-> (n2:Article
{title:"Graph databases"})

RETURN n1,n2
```

Figure 14: Cypher query to extract the graph of the example presented in Figure 10 (Angles et al., 2018).

[2] https://neo4j.com/developer/cypher/

2.5 Section III

2.5.1 Domain Graph data model

According to Vrgoč et al. (2021), the formal definition of the Domain Graph is

> "A Domain Graph $G = (O, \gamma)$ consists of a finite set of objects $O \subseteq$ Obj and a partial mapping $\gamma: O \rightarrow O \times O \times O$.
> Intuitively, O is the set of objects that appear in our graph database, and γ models edges between objects. If $\gamma\ (e) = (n1,\ t,\ n2)$, this states that the edge $(n1,\ t,\ n2)$ has id e, type t, and links the source node $n1$ to the target node $n2$. We can analogously define our model as a relation:
> DomainGraph (source, type, target, eid). "

The Domain Graph uses a quad representation, where the fourth element represents a unique identifier of the other three elements (triple). Similar as RDF and Property Graph data models, the Domain Graph data model is also used to represent real-world entities, and relationships between them (Vrgoč et al., 2021).

Figure 15 shows the Domain Graph for the statement < Abraham address Jacobine Ryes vei, 0976 Oslo.>

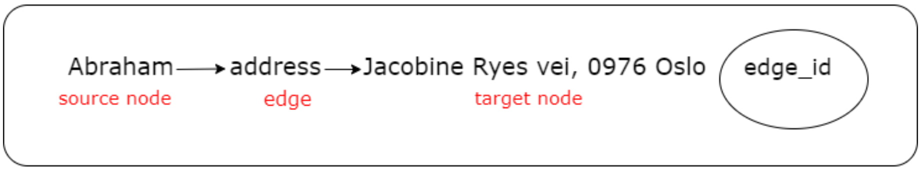


Figure 15: Domain Graph data model (Vrgoč et al., 2021).

The Domain Graph data model has been recommended as a more suitable alternative than RDF and Property Graph data models to represent the state of the art knowledge Graph Wikidata (Ilievski et al., 2020), (Angles et al., 2022), (Vrgoč et al., 2021).

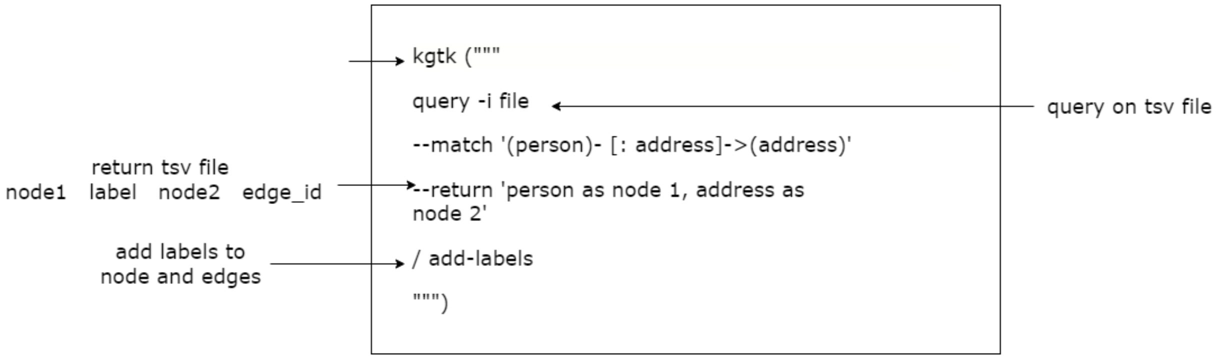
2.5.2 Knowledge Graph Tool Kit

The Knowledge Graph Tool Kit (KGTK) was created to load, query and analyse Wikidata using limited computer resources. KGTK also includes support for Python and a python library has been implemented to performs many different operations on graphs, such as cleaning, validation, extracting subsets, etc. KGTK also offers exports capabilities to popular formats such as Tsv, Neo4j and N-triples. When KGTK imports the knowledge graph Wikidata three files are generated. The first file contains Wikidata QNodes and PNodes. The second file contains the edges, which consist of all Wikidata statements concerning its entities. The third file contains all the qualifiers of the statements. (Ilievski et al., 2020), (Angles et al., 2022).

2.5.3 Kypher query language

Kypher (Ilievski et al., 2020) is the query language included in KGTK. Kypher is based on the Cypher query language (Angles et al., 2022). Kypher extracts those real-world entities which reside on the edges as nodes. In addition, Kypher does not use all the Cypher commands, and the syntax of Kypher is also different than Cypher. Kypher uses the KGTK file, e.g., tsv (Tab-separated values) format, and the output file is also the tsv file. Operating on KGTK files requires no server installation (Chalupsky & Szekely, 2022).

Figure 16 shows the Kypher query to extract Figure 15.



```
kgtk ("""
query -i file
--match '(person)- [: address]->(address)'
--return 'person as node 1, address as
node 2'
/ add-labels
""")
```

Figure 16: Kypher query for Figure 15 (Chalupsky & Szekely, 2022).

2.6 Section IV

This section introduces the knowledge graph Wikidata. This section explains how Wikidata entities are represented in the Wikibase data model.

2.6.1 Wikidata Knowledge Graph (KG)

Wikidata (Vrandečić & Krötzsch, 2014) is a sister project of the well-known project Wikipedia. Wikimedia introduced Wikidata in October 2012, and initially number of editors were less to connect items to Wikipedia article, but in January 2013 Wikidata has grown increasingly with the collaboration of three Wikipediaes Hungarian, then Hebrew and Italian. Now, the Wikidata community has already created 100,284,087 items with a high number of editors. The latest statistics [3] (Statistics,2022) shows there are a total of 24,027 active editors who edit Wikidata contents. Wikidata export services support different RDF syntaxes including JSON-LD and N-triples. In addition, Wikidata Query service [4] (Wikidata query service, 2022) is available to query over Wikidata, and Wikidata Toolkit [5] (Wikidata toolkit – mediawiki, 2022) is a JAVA library and facilitates the users to download the Wikidata and use in other applications (Malyshev, Krötzsch, González, Gonsior, & Bielefeldt, 2018), (Hall et al., 2018).

2.6.2 Wikibase Data model

The wikibase data model [6] (projects, 2022) is used to model the structure of real-world entities in Wikidata. Wikidata is a collection of entities; each entity is presented as a webpage with the relevant information about the entity. There are two types of entities: items and properties.- An item is either a class or an individual. A property defines the relationship between items defined in Wikidata. Figure 17 provides a graphical representation of the Wikibase data model (Eells et al., 2021), (Erxleben et al., 2014).

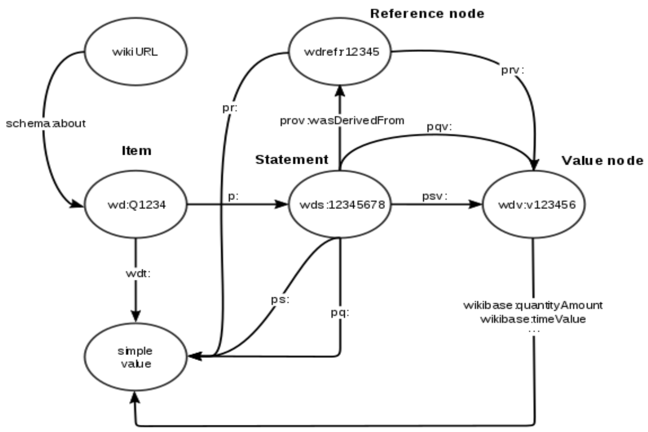


Figure 17: Wikibase Data Model (projects, 2022) (Eells et al., 2021).

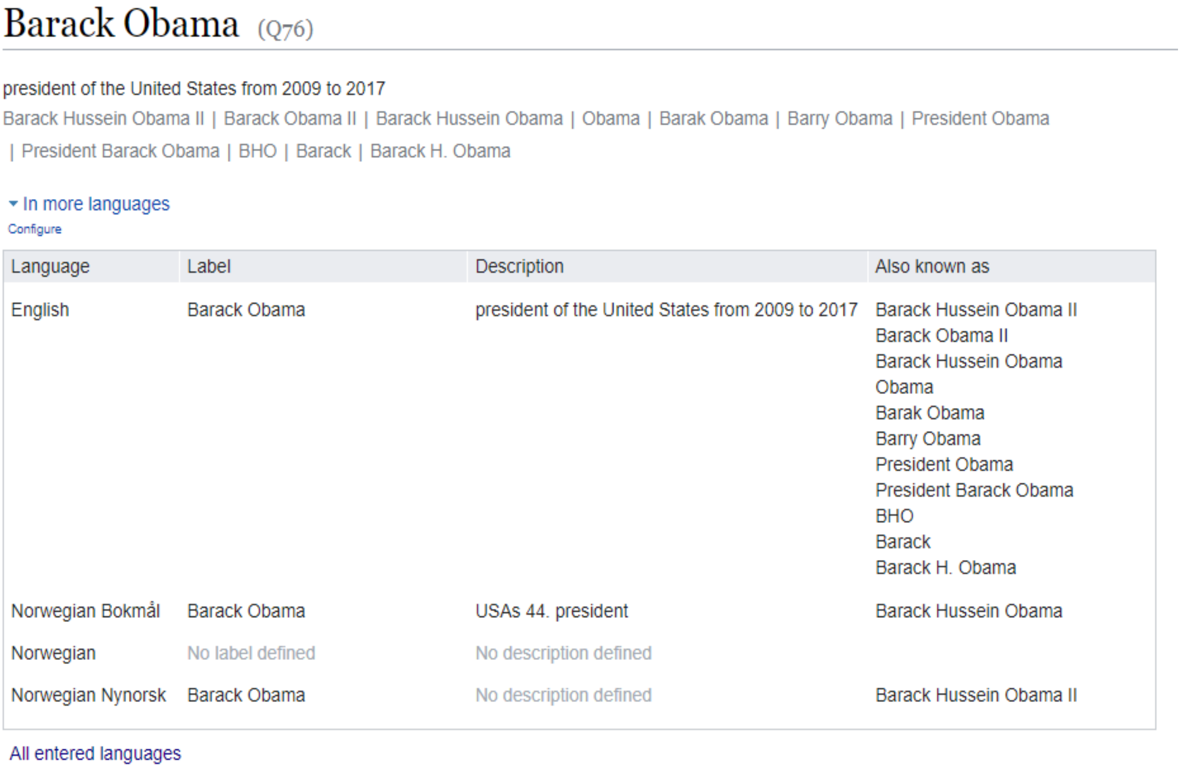
[3] https://www.wikidata.org/wiki/Wikidata:Statistics

[4] https://query.wikidata.org/

[5] https://www.mediawiki.org/wiki/Wikidata_Toolkit

[6]https://www.mediawiki.org/wiki/Wikibase/Indexing/RDF_Dump_Format

Figure 18 shows how Wikidata presents the entity Barack Obama (wd:Q76)



Barack Obama (Q76)

president of the United States from 2009 to 2017

Barack Hussein Obama II | Barack Obama II | Barack Hussein Obama | Obama | Barak Obama | Barry Obama | President Obama | President Barack Obama | BHO | Barack | Barack H. Obama

▾ In more languages

Configure

Language	Label	Description	Also known as
English	Barack Obama	president of the United States from 2009 to 2017	Barack Hussein Obama II Barack Obama II Barack Hussein Obama Obama Barak Obama Barry Obama President Obama President Barack Obama BHO Barack Barack H. Obama
Norwegian Bokmål	Barack Obama	USAs 44. president	Barack Hussein Obama
Norwegian	No label defined	No description defined	
Norwegian Nynorsk	Barack Obama	No description defined	Barack Hussein Obama II

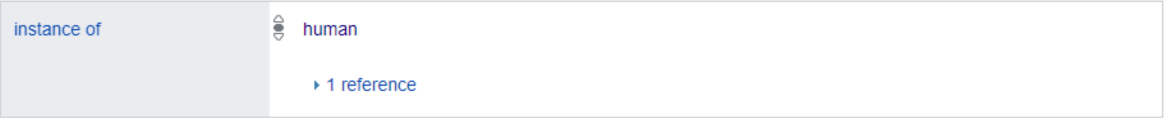
All entered languages

Figure 18: Wikidata entity Barack Obama (wd: Q76) [7] (Barack Obama, 2022).

Wikidata items are identified by Q<number>, which defines a unique identifier. For instance, wd:Q76 is the unique identifier of a Wikidata item representing the former president of the United States "Barack Obama". The specifrication of each Wikidata item includes a label, a title, one or more statements, a description, and one or more sitelinks. The label defines the name of Wikidata item and the description provide a short overview of the item. For instance, in Figure 18:

- the entity page is http://www.Wikidata.org/wiki/Q76
- the label is Barack Obama, and
- the description defines the short introduction about the item Q76.

Figure 19 shows an example of statement for the item Barack Obama (wd:Q76), and Figure 20 shows a graphical representation of the item Barack Obama based on the Wikibase data model.

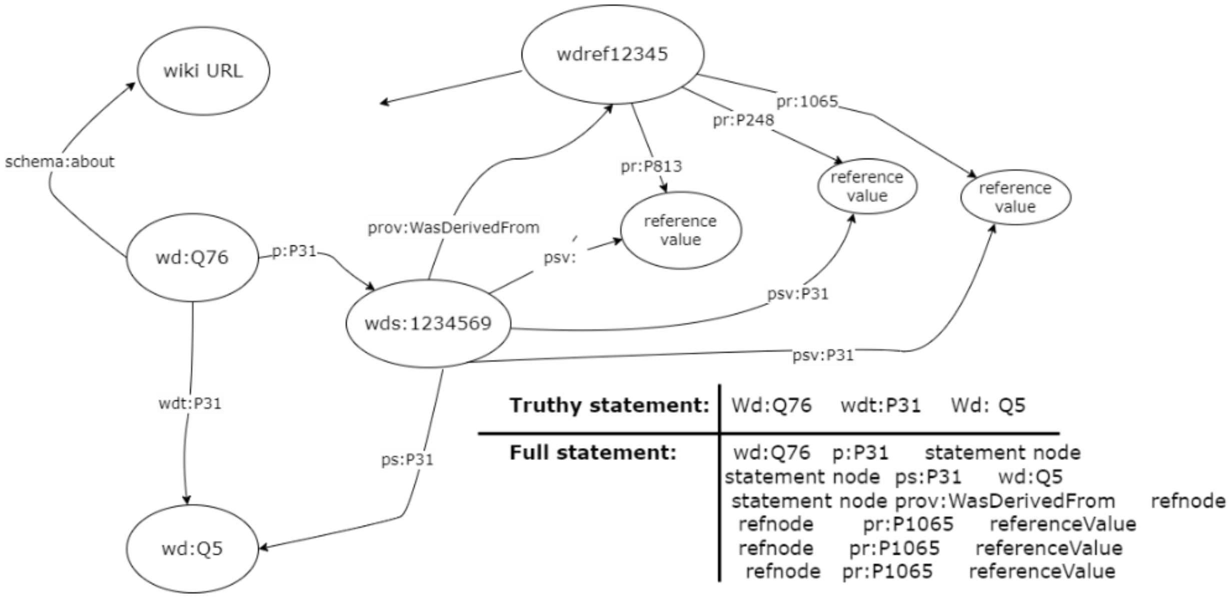


instance of	human ▸ 1 reference

Figure 19: Barack Obama Q76 Statement

[7] https://www.wikidata.org/wiki/Q76

There are two common representations of Wikidata statements: truthy representation and full representation. Truthy statements only include subjects (entity title), properties, and values. On the other hand, full statements also include additional information, such as qualifiers and references (Erxleben et al., 2014).



Truthy statement:	Wd:Q76 wdt:P31 Wd: Q5
Full statement:	wd:Q76 p:P31 statement node statement node ps:P31 wd:Q5 statement node prov:WasDerivedFrom refnode refnode pr:P1065 referenceValue refnode pr:P1065 referenceValue refnode pr:P1065 referenceValue

Figure 20: Wikidata entity wd:Q76 in Wikibase Data model (Eells et al., 2021) (Erxleben et al., 2014).

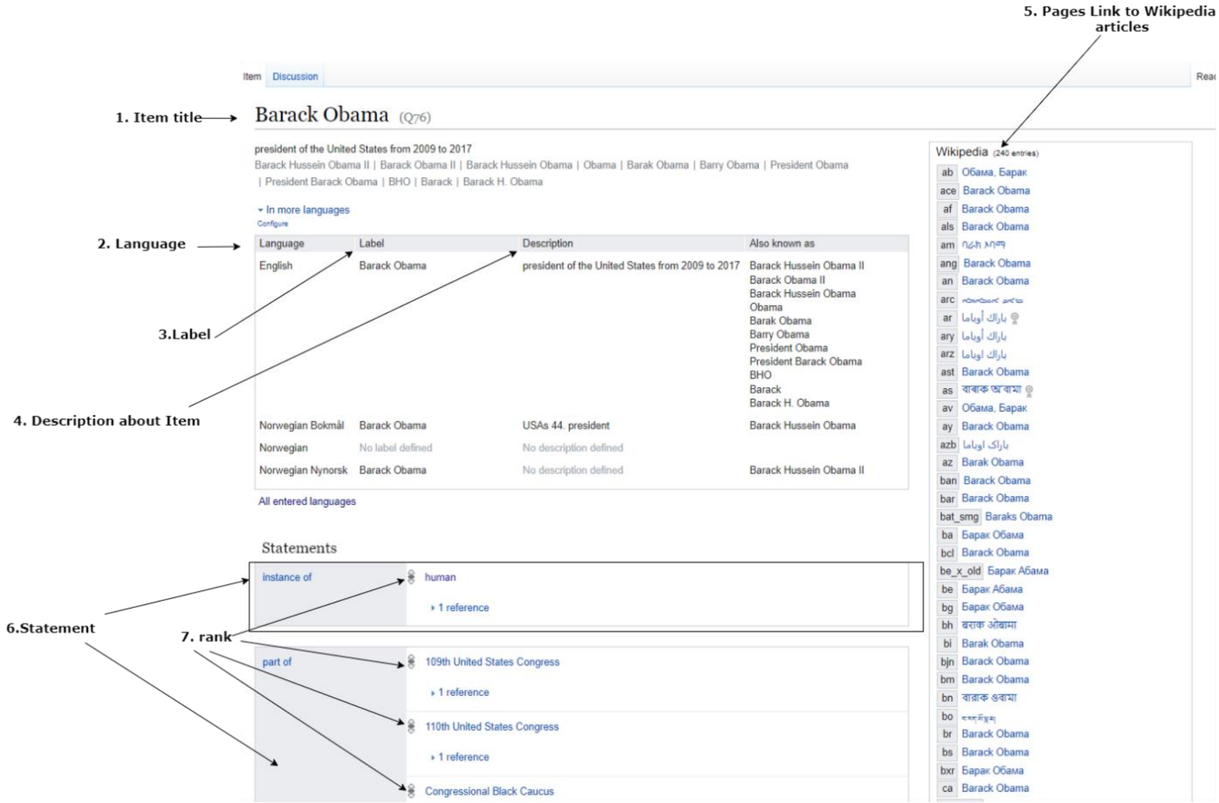
Wikidata properties also have unique identfiers. For instance, wdt:P31 is a Wikidata property that states that an item is part of a class. Like any Wikidata item, a Wikidata property can be also displayed as a webpage. Each Wikidata property has associated a datatype that determines the type of values supported by this property (Wikiproject properties/reports/Datatypes, 2022), (Erxleben et al., 2014).

Table 3 shows the data type of Wikidata properties (Wikiproject properties/reports/Datatypes,2022) [8] (Erxleben et al., 2014).

Wikidata properties	**Datatype**
Date of birth P569	Wikibase: Time
Coordinate location P625	Wikibase: GlobeCoordinate
Wikidata form P625	Wikibase: WikibaseForm
Weather P4150	Wikibase: TabularData
Geoshape P3896	Wikibase: Geoshape
Reference URL P854	Wikibase: URL
Formula P274	Wikibase: String
Code P267	Wikibase: External id
Formula P2534	Wikibase: Math

Table 3: Wikidata properties concerning its data type (Wikiproject properties/reports/Datatypes, 2022).

[8] https://www.wikidata.org/wiki/Wikidata:WikiProject_Properties/Reports/Datatypes


5. Pages Link to Wikipedia articles
1. Item title
2. Language
3.Label
4. Description about Item
6.Statement
7. rank

Item Discussion Read

Barack Obama (Q76)

president of the United States from 2009 to 2017
Barack Hussein Obama II | Barack Obama II | Barack Hussein Obama | Obama | Barak Obama | Barry Obama | President Obama | President Barack Obama | BHO | Barack | Barack H. Obama

▾ In more languages
Configure

Language	Label	Description	Also known as
English	Barack Obama	president of the United States from 2009 to 2017	Barack Hussein Obama II Barack Obama II Barack Hussein Obama Obama Barak Obama Barry Obama President Obama President Barack Obama BHO Barack Barack H. Obama
Norwegian Bokmål	Barack Obama	USAs 44. president	Barack Hussein Obama
Norwegian	No label defined	No description defined	
Norwegian Nynorsk	Barack Obama	No description defined	Barack Hussein Obama II

All entered languages

Statements

instance of	human ▸ 1 reference
part of	109th United States Congress ▸ 1 reference 110th United States Congress ▸ 1 reference Congressional Black Caucus

Wikipedia (240 entries)

ab	Обама, Барак
ace	Barack Obama
af	Barack Obama
als	Barack Obama
am	ባራክ ኦባማ
ang	Barack Obama
an	Barack Obama
arc	[illegible]
ar	باراك أوباما
ary	باراك أوباما
arz	باراك اوباما
ast	Barack Obama
as	বাৰাক অ'বামা
av	Обама, Барак
ay	Barack Obama
azb	باراک اوباما
az	Barak Obama
ban	Barack Obama
bar	Barack Obama
bat_smg	Baraks Obama
ba	Барак Обама
bcl	Barack Obama
be_x_old	Барак Абама
be	Барак Абама
bg	Барак Обама
bh	बराक ओबामा
bi	Barak Obama
bjn	Barack Obama
bm	Barack Obama
bn	বারাক ওবামা
bo	[illegible]
br	Barack Obama
bs	Barack Obama
bxr	Барак Обама
ca	Barack Obama

Figure 21: Representation of Wikidata entity Barack Obama wd:Q76 in order (Erxleben et al., 2014).

Figure 21 is presented Wikidata entity's information in order. For instance, all the information is presented in proper order and rank, including Wikidata entity title, description, aliases, statements, qualifiers, and references. Each statement is divided into three ranks which differentiate the statements from each other. While extracting the dataset over large Wikidata, ranks play a vital role, e.g., the default statements used the average rank, and referred rank means select that statement which has preferred over average rank. Deprecated means the statement is kept in the system for some reason; otherwise, it has no meaning (Erxleben et al., 2014).

2.7 Summary

This chapter has provided a short introduction to three popular graph data models: RDF, Property Graph, and Domain Graph. Each data model has its own peculiarities when representing information in a knowledge graph. For instance, the RDF data model uses triples (subject-predicate-object) to model statements. The Property Graph data model allows to define edges attributes to extend the definition of an statement. The Domain Graph data model represents statements using quads. Wikidata is a very popular knowledge graph which contains millions of definitions of real-world entities.Information is defined in Wikidata using the Wikibase data model. In the next chapter, the RDF, Property Graph and Domain Graph data models will be evaluated when representing information in Wikidata.

PART II
Qualitative analysis of RDF, PG and DG for Wikidata

Chapter 3

Analysis of RDF, PG, and DG for modelling information in Wikidata

This chapter analysis the suitability of three graph data models (RDF, Property Graph, and Domain Graph) for representing information in Wikidata based on the background from Chapter 2. Section 3.1 demonstrates the evaluation setup e.g., hardware and software configuration, selection of Wikidata subgraph before presenting Wikidata in RDF, PG and DG. Section 3.2,3.3, and 3.4 covers the representation of one of the graph data models considered. The RDF data model has limited expressivity to natural model information in Wikidata. To alleviate this limitations, different reification techniques has been proposed including standard reification, n-ary relations and singleton properties. In the case of the Property Graph Wikidata statements are represented using edges and qualifiers and references are represented as edge attributes in the form of key-value pairs. Finally, the Domain Graph data model is suitable for the ability of representing statements using quads, where a quad contains an edge id, which can be used when modelling qualifiers and references. Further, section 3.5, 3.6 and 3.7 presents the analysis of three graph data model RDF, PG and DG respectively for Wikidata subgraph based on the representation from section 3.2, 3.3 and 3.4;The result of the analysis of these three graph data models will help the reader to better understand which graph data model best represents Wikidata statements.

3.1 Evaluation set-up

Underlying hardware and software set-up was used to evaluate the representation of the three graph data models for Wikidata. Wikidata dumps were enormous and difficult to download on the local machine. However, different Wikidata entities were easily available and downloaded on the local machine via the Wikidata API services. Further, these Wikidata entities were examined in graph databases with regard to the models.

3.1.1 Hardware setup

All Wikidata entities were examined on the local machine. Table 4 shows the hardware configuration for Wikidata representation in the three graph data models.

Item	Specification
Machine type	macOS Catalina
CPU	Intel x86 CPU
Processor	2,7GHz Dual-Core Intel Core i5
Memory	8 GB MHz DDR3
OS	macOS
Disk space	2.5 GB

Table 4: Hardware specification for examining the Wikidata in three graph data models.

3.1.2 Software setup

Table 5 shows the software specification for examining the Wikidata in the three graph models.

Software	Version
Python 2	2.7
Python 3	3.11
Anaconda	2.31
Jupyter Notebook	-

Table 5: Software specifications for examining the three graph data models for Wikidata.

3.1.3 Graph databases

Table 6 shows the graph databases and their graph data models used throughout the thesis analysis of the three graph models for Wikidata. Stardog database 9 (Stardog Union, 2022), including the collection of Wikidata entities in the turtle format, was created. Neo4j database 10 (Neo4j documentation - NEO4J documentation, 2022) was also created to examine the representation of Wikidata entities in the property graph data model. Knowledge Graph Tool Kit (KGTK) was installed to study the Domain graph data model representation for Wikidata. Installation of

[9] https://www.stardog.com/

[10] https://neo4j.com/docs/

KGTK does not require any special hardware or database. KGTK provides the feature of running Wikidata using the Google Colab [11] (usc-isi-i2, 2022).

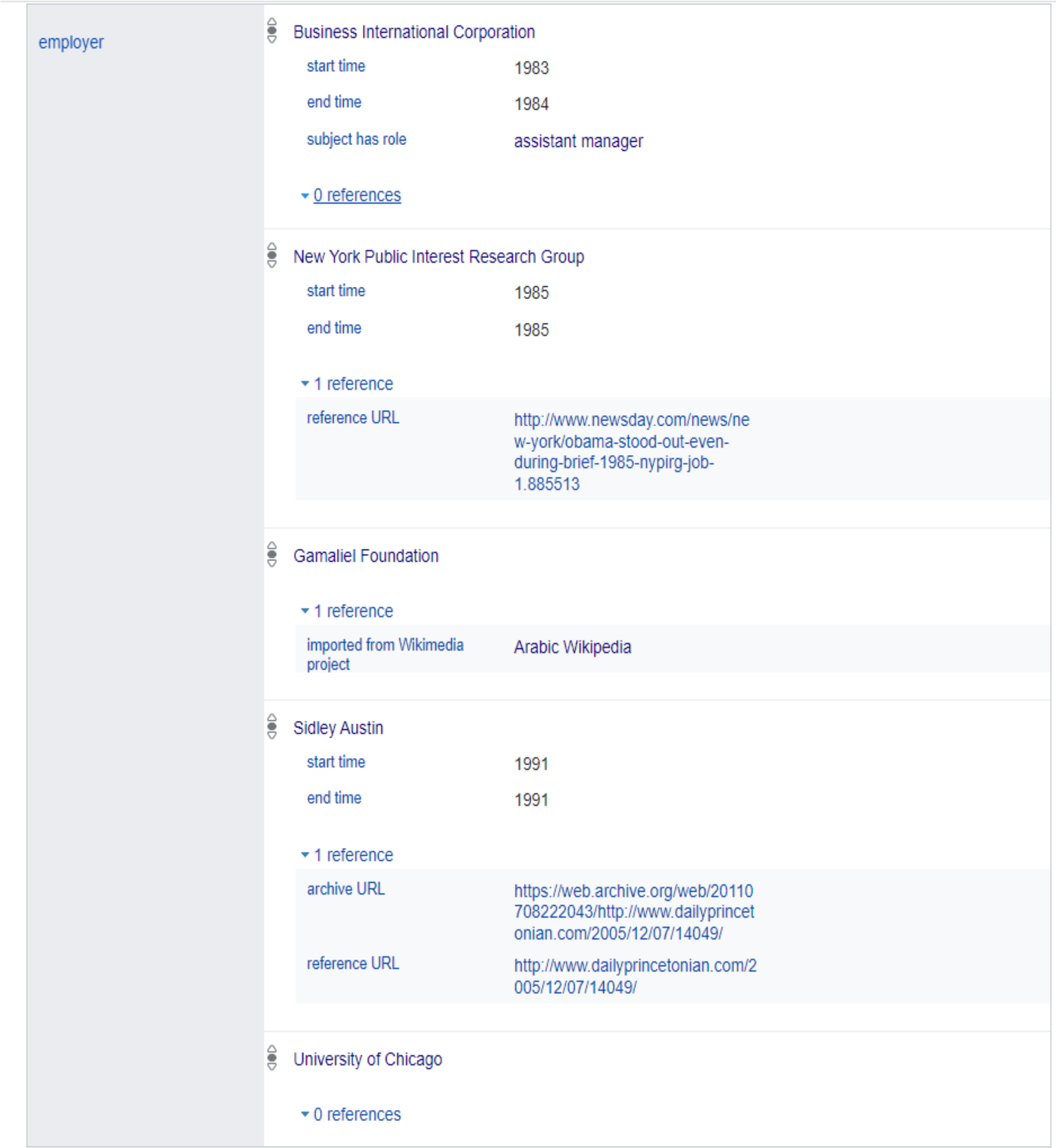
Graph data model	Graph database
Stardog	RDF data model
Neo4j	Property Graph data model
KGTK	Domain Graph data model

Table 6: Graph data models and corresponding graph databases.

[11] https://github.com/usc-isi-i2/kgtk-notebooks

3.1.4 Selection of subgraph from Wikidata

Figure 22 [12] (Barack Obama, 2022) shows the subgraph of the Wikidata entity Barack Obama (Q76), which displays the employment history of Barack Obama (Q76) . The subgraph consists of five statement nodes, and each statement further defines the additional information, e.g., qualifiers and references.



employer

Business International Corporation

start time	1983
end time	1984
subject has role	assistant manager

0 references

New York Public Interest Research Group

start time	1985
end time	1985

1 reference

reference URL	http://www.newsday.com/news/new-york/obama-stood-out-even-during-brief-1985-nypirg-job-1.885513

Gamaliel Foundation

1 reference

imported from Wikimedia project	Arabic Wikipedia

Sidley Austin

start time	1991
end time	1991

1 reference

archive URL	https://web.archive.org/web/20110708222043/http://www.dailyprincetonian.com/2005/12/07/14049/
reference URL	http://www.dailyprincetonian.com/2005/12/07/14049/

University of Chicago

0 references

Figure 22: Subgraph of Wikidata entity Barack Obama (Q76).

[12] https://www.Wikidata.org/wiki/Q42

3.1.5 Extraction of subgraphs from Wikidata

Table 7 shows the graph data model and graph databases concerning their query language, which is used to interact with the database.

Graph database	Graph data model	Query languages
Stardog	RDF data model	SPARQL
Neo4j	Property Graph data model	Cypher
KGTK	Domain Graph data model	Kypher

Table 7: Graph data model with corresponding graph databases, and query languages

3.1.5.1 SPARQL

Table 8 shows the SPARQL query to extract the subgraph of figure 22 in the RDF data model. The SPARQL query consisted of lines over five Wikidata statements in figure 22.

```
SELECT DISTINCT *
WHERE
{
wd:Q76 p:P108 ?statement.
?statement ps:P108 ?job location.
OPTIONAL{?statement pq:P580 ?start Time.}
OPTIONAL{?statement pq:P582 ?end Time.}
OPTIONAL{?statement pq:P2868 ?role.}
OPTIONAL { ?statement Prov:wasDerivedFrom
?refnode.
OPTIONAL{?refnode pr:P854 ?refurl.}
OPTIONAL{?refnode pr:P1065 ?archieveurl .}
OPTIONAL{?refnode pr: P143?Wikimedia
project .}}
}
```

Table 8: SPARQL query for Wikidata entity Barack Obama subgraph of figure 22.

3.1.5.2 Cypher

Table 9 shows the Cypher query to extract the subgraph of figure 22 in the Property Graph data model.

```
MATCH s= (:ns0__Item {uri: "http://www.wikidata.org/entity/Q76"})-
[:ns1__P108]-> ()
MATCH q=(n)-[:ns6__P108]->(statement)-[]->()
RETURN s,q
```

Table 9: Cypher query for Wikidata entity Barack Obama subgraph of figure 22.

3.1.5.3 Kypher

Table 10 and Table 11 shows the Kypher query to extract the subgraph of figure 22 in the Domain Graph data model.

```
kgtk("""
    query -i all
        --match '
            (node1:Q76)-[id:P108]->(n2)'
            --return 'node1 as node1, id as id, n2 as n2'
            / add-labels
""")
```

Table 10: Kypher query for Wikidata entity Barack Obama subgraph of figure 22.

```
kgtk("""
    query -i all
        --match '
            (:Q76)-[id:P108]->(n2),
            (id)-[qualifier_id]->(qualifier_value)'
            --
return 'id as id, qualifier_id as qualifier_id, qualifier_value as qua
lifier_value'
            / add-labels
""")
```

Table 11: Kypher query for Wikidata entity Barack Obama extracting qualifier of figure 22.

3.2 Wikidata in RDF data model

Because RDF statements are defined in the form of triples, it is difficult to define statements about statements as it is required by the example in Figure 22. RDF proposes three reification techniques to overcome this limitation. These three reification techniques are illustrated in Figure 23, Figure 24, and Figure 25 (Eells et al., 2021), (Hernández et al., 2015), (Manola & Miller, 2014), (Nguyen, Bodenreider, & Sheth, 2014).

The three reification techniques were proposed for the Wikidata entity Barack Obama (Q76) subgraph of figure 22 in order to describe how the RDF data model represents Wikidata statements, qualifiers and references.

Figure 23 (Manola & Miller, 2014) shows the standard reification, figure 24 (Hernández et al., 2015) shows the n-ary relations and figure 25 (Nguyen, Bodenreider, & Sheth, 2014), (Hernández et al., 2015), (Orlandi et al., 2021) shows the singleton properties for figure 22.

3.2.1 standard reification

In standard reification, five different statement nodes indicate the triples, e.g., the statement node (q76-66C211F7-A8BB-42F6-98C8-3451C112629C) declares the triple [subject: wd: Q76), (predicate: P108). (Object: wd: Q4537781]. Moreover, the statement node(q76-66C211F7-A8BB-42F6-98C8-3451C112629C) generates triples for qualifiers and references; this way, figure 22 was thus represented through the standard reification in the RDF data model (Manola & Miller, 2014).

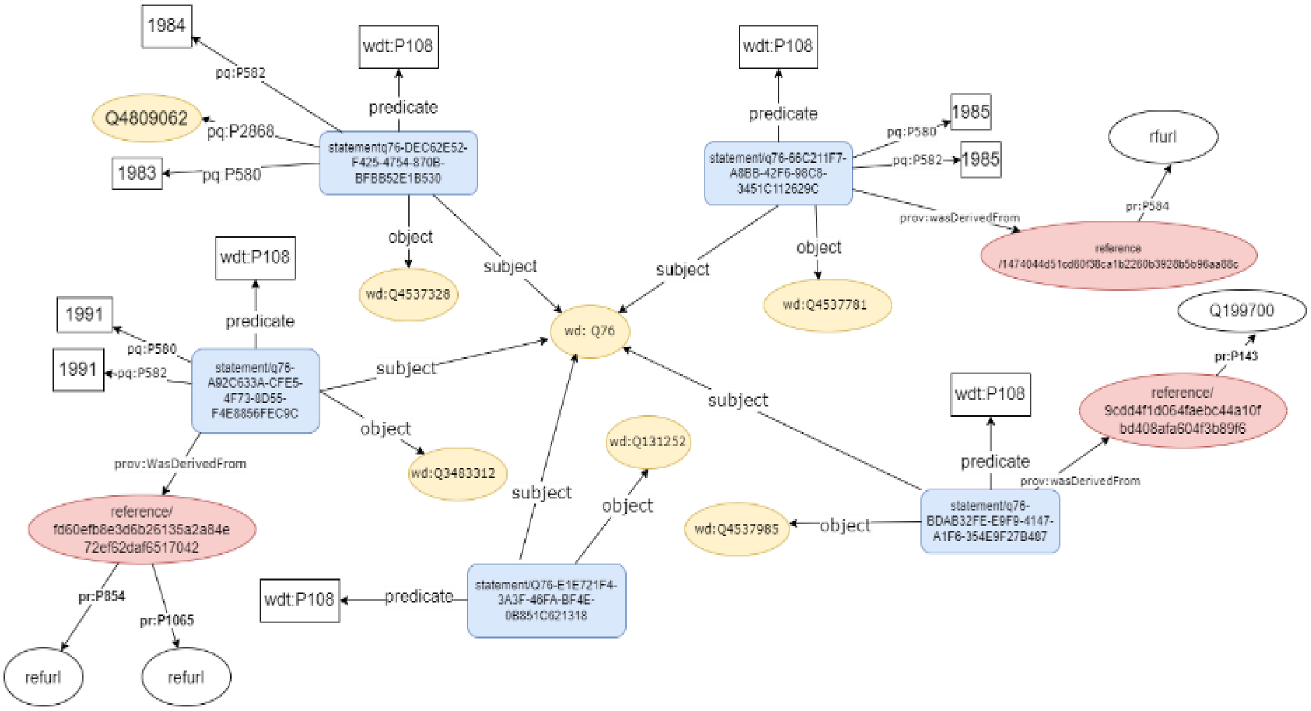


Figure 23: Examples of standard reification (Manola & Miller, 2014).

.

Table 15 (Manola & Miller, 2014) in appendix shows the statement node generating the triples and represents the additional information qualifiers and references for figure 22. A total of 29 triples were generated when representing the Wikidata entity Barack Obama (Q76) subgraph of figure 22 through standard reification.

3.2.2 n-ary relations

In n-ary relations reification technique (Hernández et al., 2015), RDF binary relation is decomposed into the n-ary relations to model the complex statements of figure 22. The subject (wd: Q76) was related to the five different statement nodes using the property p: P108, and this relation further describes the values, e.g., wd: Q3305213, wd: Q4537328 wd: Q3483312, wd: Q131252, wd: Q4537781 from the statement's nodes to the value nodes using the predicate ps: P108. Further, the statement nodes have represented the n-ary relations to model the complex statements (qualifiers and references). Figure 24 shows the statement node pointing to the triples for accessing the qualifiers and references (Hernández et al., 2015).

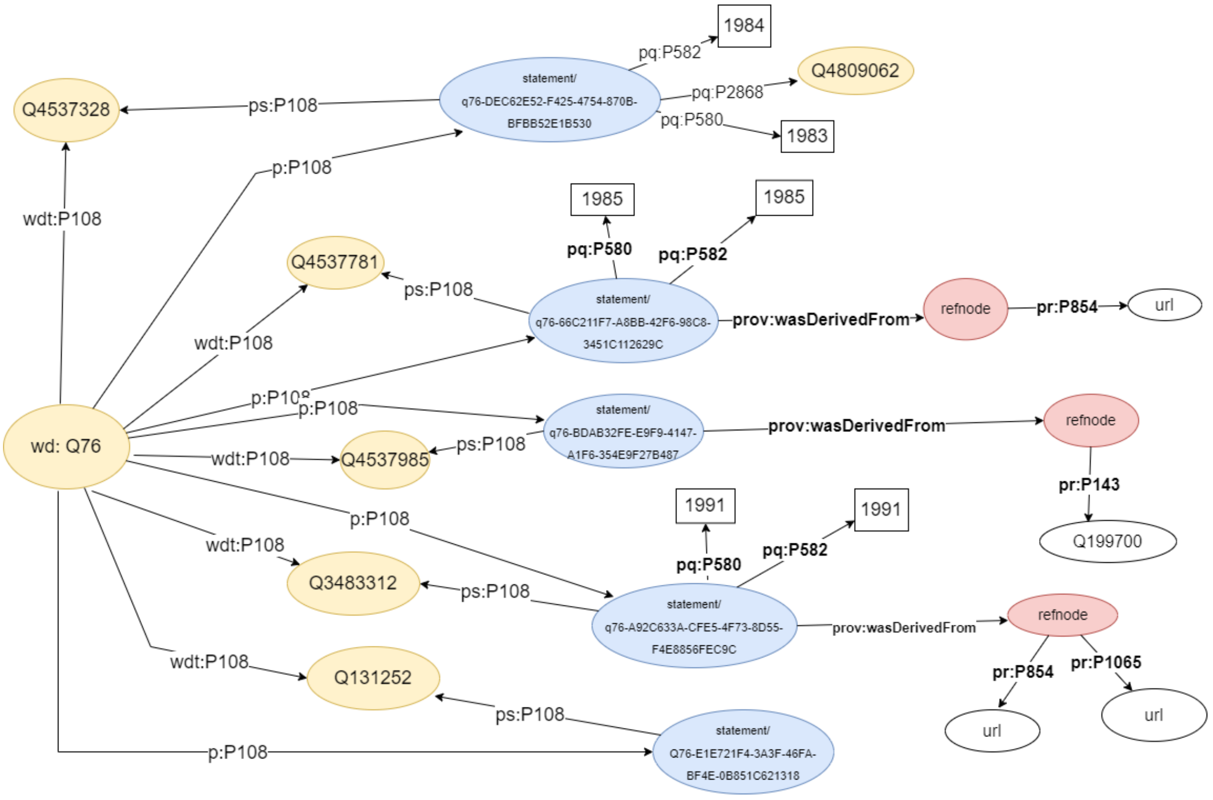


Figure 24: Examples of n-ary relations (Hernández et al., 2015).

Table 16 (Hernández et al., 2015) in the appendix shows the 24 triples using the n-ary relations technique for the Wikidata subgraph of figure 22.

3.2.3 singleton properties

When representing figure 22 in singleton properties (Hernández et al., 2015), the five statement nodes became the instances of the predicate/Wikidata property (wdt: P108) by creating the relation "*singletonPropertyOf*" from statement nodes to predicate/Wikidata property node (wdt: P31). The Wikidata property wdt: P108 became the unique property representing statements, qualifiers, and references (Orlandi et al., 2021), (Nguyen, Bodenreider, & Sheth, 2014).

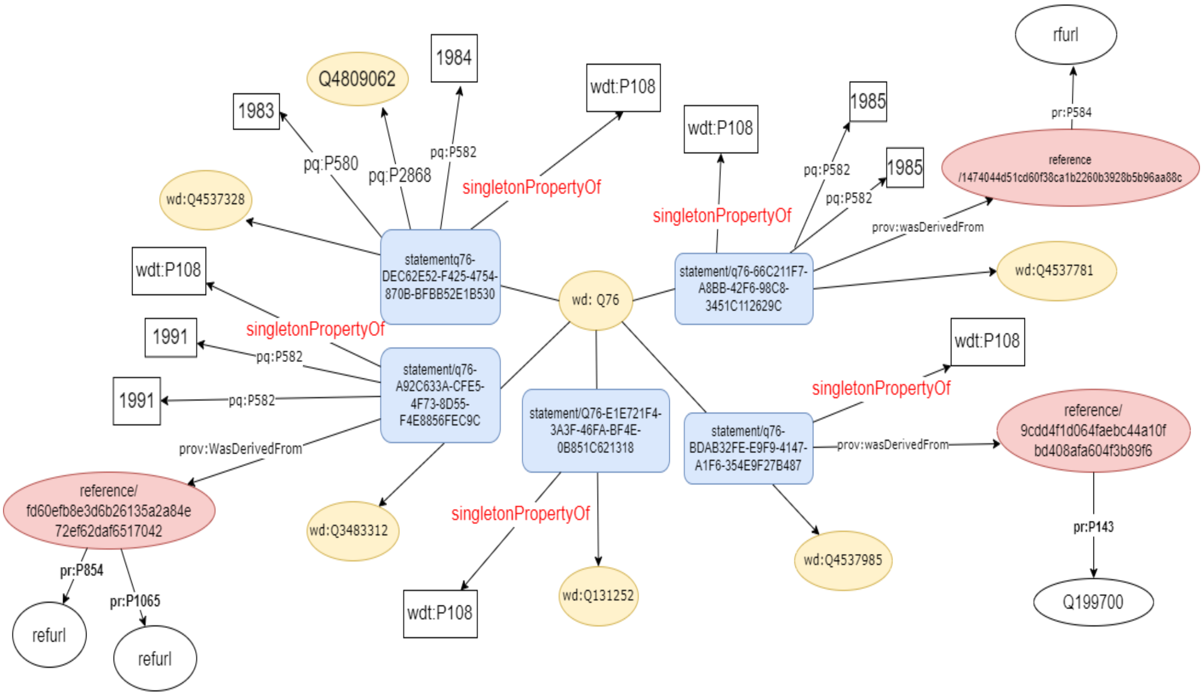


Figure 25: RDF singleton properties to model Wikidata entity Barack Obama (Q76) subgraph of figure 22 (Hernández et al., 2015).

Table 17 in the appendix shows the total 24 triples using the singleton properties for figure 22.

3.3 Wikidata in Property Graph data model

Figure 26 shows an example of a Property Graph based on the example presented in Figure 22. Wikidata real-world entities Barack Obama (wd:Q76), University of Chicago (wd:Q131252), Business International Corporation (wd:Q4537328), Sidley Austin (wd:Q3483312), New York Public Interest Research Group (wd:Q4537781) and Gamaliel Foundation (wd:Q131252) were represented using a node in the Property Graph, whereas Wikidata property (wdt:P108) was represented using several edges. Each node and edge had properties in the form of a key-value pair. The node properties defined the metadata of Wikidata entities, e.g., rdfs__label, id, URI, etc. However, Wikidata's additional information, e.g., qualifiers and references, was presented by the attributes of the edges (Angles et al., n.d.).

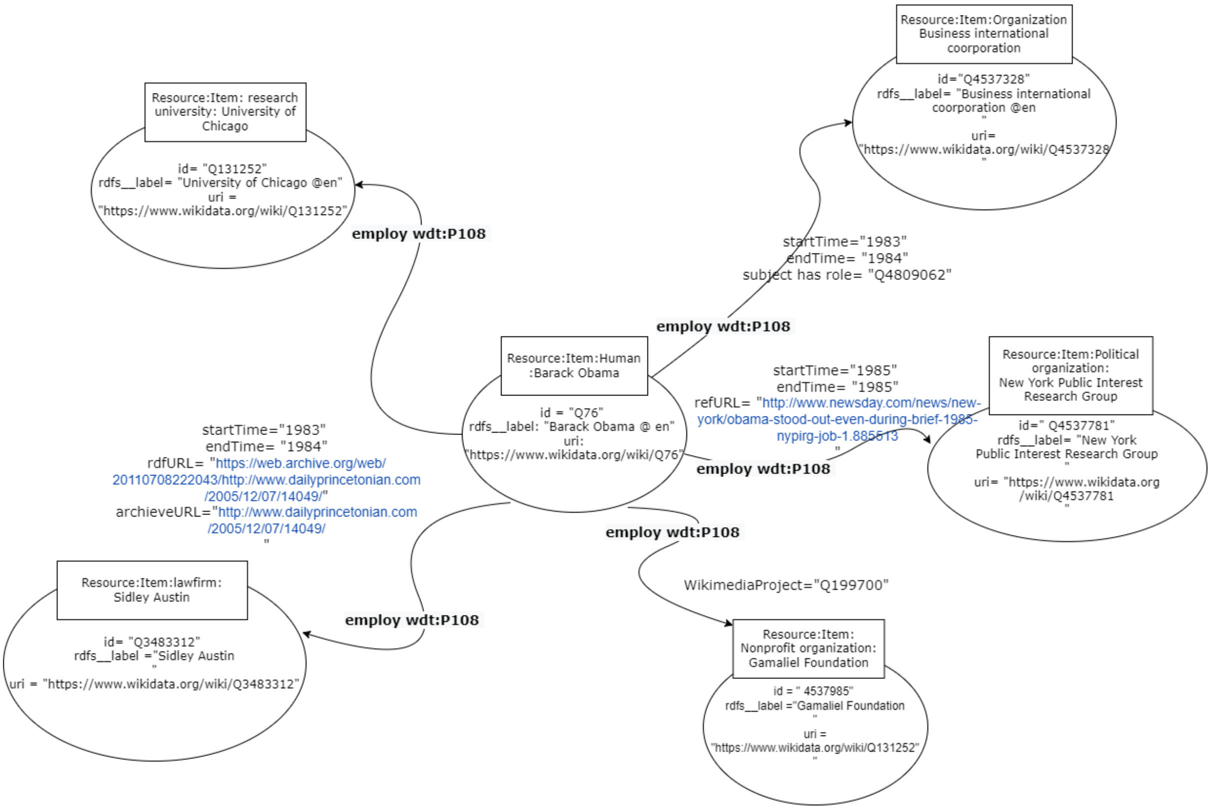

Resource:Item:Organization Business international coorporation
id="Q4537328"
rdfs__label= "Business international coorporation @en "
uri= "https://www.wikidata.org/wiki/Q4537328 "
Resource:Item: research university: University of Chicago
id= "Q131252"
rdfs__label= "University of Chicago @en"
uri = "https://www.wikidata.org/wiki/Q131252"
employ wdt:P108
startTime="1983"
endTime= "1984"
subject has role= "Q4809062"
employ wdt:P108
Resource:Item:Human :Barack Obama
id = "Q76"
rdfs__label: "Barack Obama @ en"
uri: "https://www.wikidata.org/wiki/Q76"
startTime="1985"
endTime= "1985"
refURL= "http://www.newsday.com/news/new-york/obama-stood-out-even-during-brief-1985-nypirg-job-1.885513 "
employ wdt:P108
Resource:Item:Political organization: New York Public Interest Research Group
id=" Q4537781"
rdfs__label= "New York Public Interest Research Group "
uri= "https://www.wikidata.org /wiki/Q4537781 "
startTime="1983"
endTime= "1984"
rdfURL= "https://web.archive.org/web/20110708222043/http://www.dailyprincetonian.com /2005/12/07/14049/"
archieveURL="http://www.dailyprincetonian.com /2005/12/07/14049/ "
employ wdt:P108
WikimediaProject="Q199700"
Resource:Item:lawfirm: Sidley Austin
id= "Q3483312"
rdfs__label ="Sidley Austin "
uri = "https://www.wikidata.org/wiki/Q3483312"
employ wdt:P108
Resource:Item: Nonprofit organization: Gamaliel Foundation
id = " 4537985"
rdfs__label ="Gamaliel Foundation "
uri = "https://www.wikidata.org/wiki/Q131252" "

Figure 26: Property Graph data model for Figure 22 (Angles et al., n.d.).

3.4 Wikidata in Domain Graph data model

Figure 27 illustrates how to represent the example displayed in Figure 22 using a Domain Graph. This graph representation facilitates the representation of complex Wikidata statements. Figure 22 has five statements, and in the example of Domain Graph, each statement was represented as a quad that includes an edge id. In Wikidata, references were defined by the reference id, like the statement id. The Domain Graph data model uses the edge as nodes and is best fitted to model complex Wikidata statements that include references and qualifiers (Vrgoč et al., 2021).

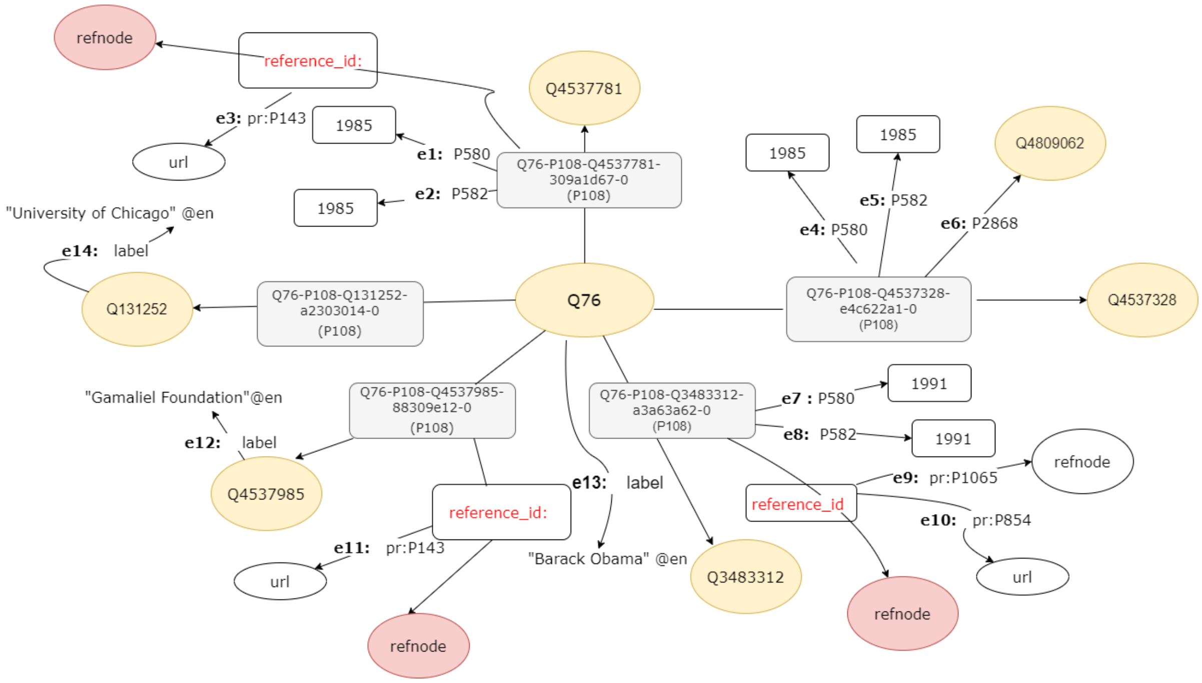


Figure 27: Domain Graph data model for figure 22 (Vrgoč et al., 2021).

3.5 Analysis of the suitability of the RDF data model for Wikidata

After representing the Wikidata statement of figure 22 in the RDF data model through different reification techniques, the following points are shown as the representation results.

1. The RDF data model presents data in two forms. One type is that the subject is presented as a predicate object. Moreover, the second form is that the data is represented as property-value paired, e.g., the subject is the same in both forms, but the predicate and object can represent property and value (Bergman, 2009).

   Figure 28 shows the two representations of the Wikidata statement in the RDF data model, e.g., predicate as property and the object as the value for Wikidata entity Barack Obama (wd: Q76) statement (Bergman, 2009).

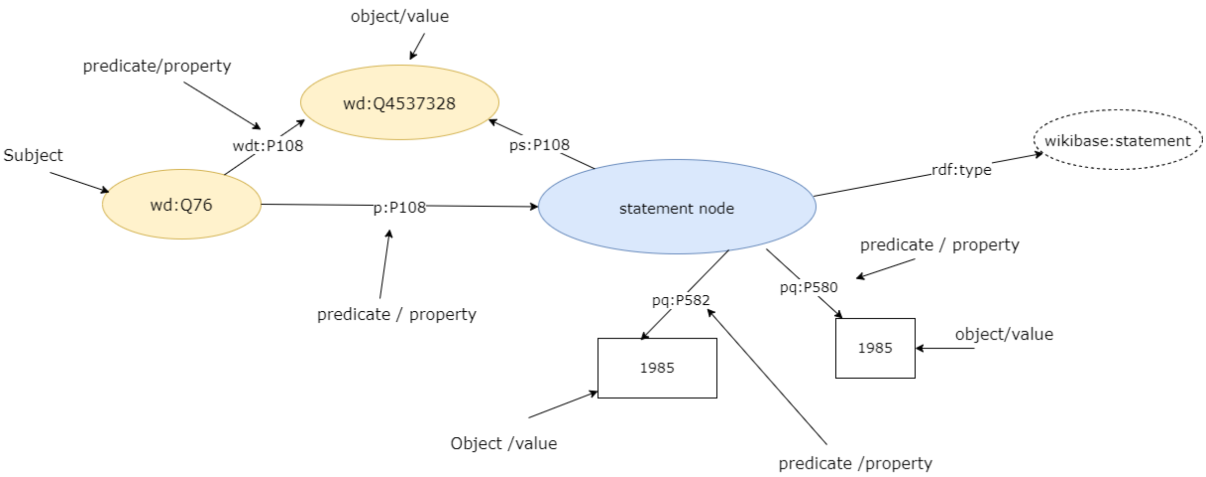


Figure 28: Wikidata statement (Subject-Predicate-Object) and (Property-value pair) (Bergman, 2009).

2. The RDF reification techniques make it possible to represent Wikidata. However, at the same time, it has several issues, e.g., limited scalability, requiring more storage capacity, longer query response time, and long query length (Nguyen, Bodenreider, & Sheth, 2014).

   - The standard reification and n-ary relations have no formal semantics and produce many redundant triples, which causes to increase in the data storage capacity and turns in scalability issues (Nguyen, Bodenreider, & Sheth, 2014). According to (Govindapillai et al., 2021), the knowledge graph is vast in size, and Wikidata is one of the most extensive knowledge graphs on which information is often updated daily. After standard reification and n-ary Relations in figure 23 and figure 24 (Hernández et al., 2015), the triples count were 29, which is a large number of triples over Wikidata's three statements, increasing the knowledge graph size by four times. Due to this, Wikidata dumps in the RDF data model takes a massive amount of time to download on the machines. In addition, there is a need for additional memory requirements, which must be fulfilled to download and store the Wikidata dumps in RDF data format, as well as required custom configurations (Ilievski et al., 2020).

- In addition, Wikidata in RDF data model makes the searching and identification tasks harder which increases the query response time and slows down the data retrieval process, and sometimes it is impossible to extract larger subgraph. According to Ilievski et al. (2020) extracting the Wikidata articles through the SPARQL query endpoint is a big challenge in RDF data model, because RDF data model deals with the Wikidata small graphs and for extracting subgraphs where more than 100 000 SPARQL queries are written which is impossible. (Nguyen, Bodenreider, & Sheth, 2014), (Orlandi et al., 2021), (Govindapillai et al., 2021).

- There are more than 24,027 editors (Vrandečić & Krötzsch, 2014) who are actively involved to update Wikidata entities on daily basis which increases the query length especially when extracting the qualifiers and references (Nguyen, Bodenreider, & Sheth, 2014), (Orlandi et al., 2021), (Govindapillai et al., 2021).

- Compared to the standard reification and n-ary relations, the singleton properties reification technique has formal semantics. It uses unique predicates to model Wikidata additional information, e.g., qualifiers and references in the RDF data model. The singleton properties reification reduces the query response time because it produces fewer redundant triples. Figure 25 shows the smaller number of triples compared to the standard reification and n-ary relations. In addition, it reduces the storage capacity, but this reification approach produces a high number of unique predicates, which affects the indexing strategy in triple stores. In short, the number of unique predicates after singleton properties reification equals the redundant triples in the standard and n-ary relations reification technique (Nguyen, Bodenreider, & Sheth, 2014), (Orlandi et al., 2021), (Govindapillai et al., 2021).

Table 12 compares the three reification techniques in the RDF data model while representing the Wikidata statement qualifiers and references (Orlandi et al., 2021).

	Standard reification	**n-ary Relations**	**Singleton properties**
Scalability issue	✓ High	✓ High	✓ Medium
Data storage issue	✓ High	✓ High	✓ Medium
Data retrieval issue	✓ High	✓ High	✓ Medium
Query size	Lengthy	Lengthy	short
Formal semantics	No	No	Yes

Redundancy in triples	High four times	High four times	Less 40%
Reification approach	Wikidata statements on subject and object	Wikidata statement on subject and object	Wikidata statement on predicates
No of predicates	No	No	✓ Very high
Required additional memory to download Wikidata RDF dump	Yes	Yes	✓ Yes

Table 12: Compare the three RDF reification techniques to model Wikidata statements in the RDF data model for Wikidata (Orlandi et al., 2021).

- The reification technique is ambiguous and unclear when one qualifier property has two different values. Figure 29 shows the subgraph of a Wikidata entity (Star Trek the Next Generation Q16290),showing fuzzy reification behaviour. For instance, one cast member has many character roles (pq: P453)(Erxleben et al., 2014).

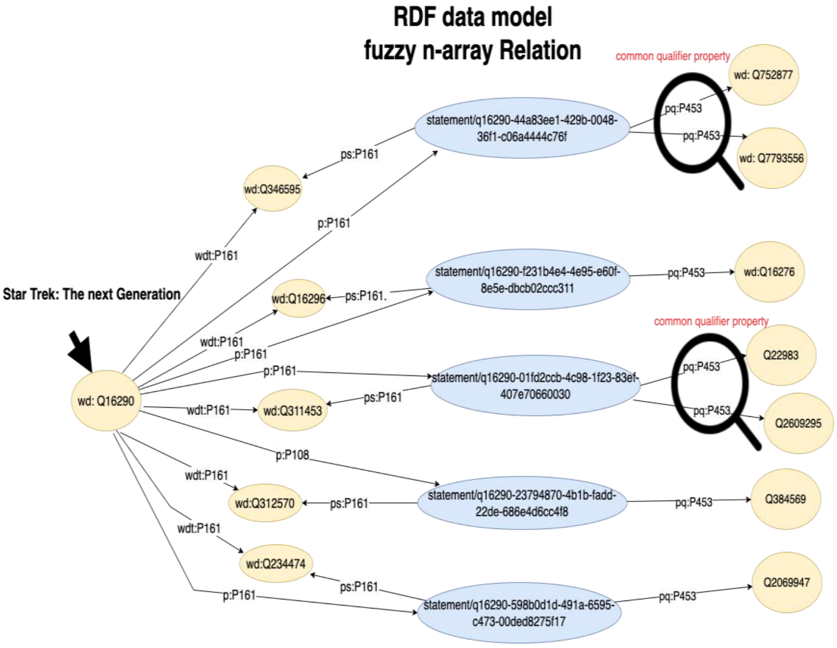
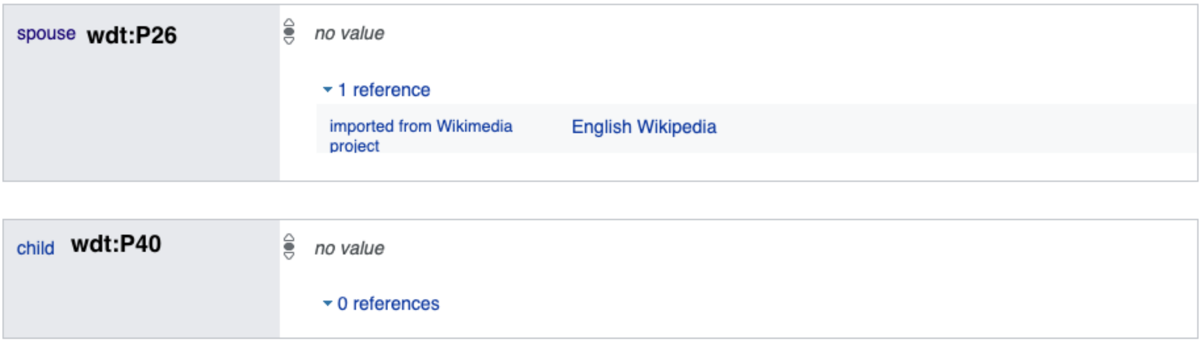

RDF data model
fuzzy n-array Relation
common qualifier property
wd: Q752877
pq:P453
pq:P453
wd: Q7793556
statement/q16290-44a83ee1-429b-0048-36f1-c06a4444c76f
ps:P161
wd:Q346595
p:P161
wdt:P161
Star Trek: The next Generation
statement/q16290-f231b4e4-4e95-e60f-8e5e-dbcb02ccc311
pq:P453
wd:Q16276
wd:Q16296
ps:P161.
common qualifier property
Q22983
wdt:P161
p:P161
p:P161
statement/q16290-01fd2ccb-4c98-1f23-83ef-407e70660030
pq:P453
pq:P453
Q2609295
wd: Q16290
wdt:P161
wd:Q311453
ps:P161
p:P108
wdt:P161
wd:Q312570
ps:P161
statement/q16290-23794870-4b1b-fadd-22de-686e4d6cc4f8
pq:P453
Q384569
wdt:P161
wd:Q234474
ps:P161
p:P161
statement/q16290-598b0d1d-491a-6595-c473-00ded8275f17
pq:P453
Q2069947

Figure 29: fuzzy RDF reification[13] (Star Trek the Next Generation, 2022).

[13] https://www.Wikidata.org/wiki/Q16290

3. The RDF data model also supports those Wikidata properties that have no values. Figure 30 is the subgraph of Wikidata entity Elizbeth I England (wd: Q7207) which has two properties, P26 (spouse) and P40 (child), and it has "no value." When the SPARQL query is written to extract the subgraph of the Elizbeth I England (Q7207) (Elizbeth I England, 2022), it shows nothing in the results. Such properties with no values are like negation in OWL and RDF data model handles such statements efficiently (Erxleben et al., 2014).



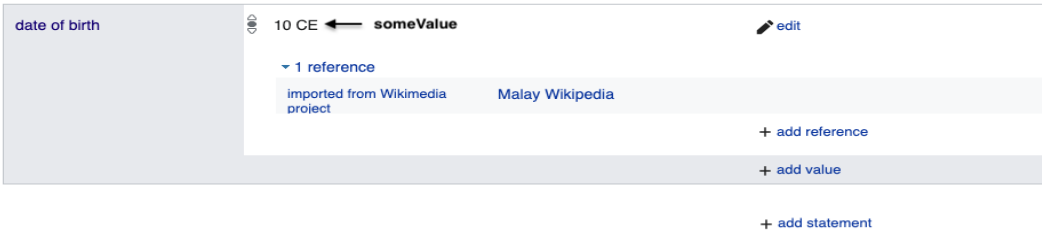
spouse wdt:P26 — no value
▾ 1 reference
imported from Wikimedia project — English Wikipedia

child wdt:P40 — no value
▾ 0 references

Figure 30: RDF facilitates no values in Wikidata (Elizbeth I England, 2022).

4. Wikidata uses the property constraint (P2302) instead of domain and range properties. After the semantic interoperability, the RDF data model can use Wikidata property constraints both as a domain and range. Further, the domain property is defined by the type constraints, and similarly, the value-type constraint property in Wikidata defines the range property (Haller, Polleres, Dobriy, Ferranti & Rodríguez Mendez, 2022).

5. Many Wikidata properties have some values, but these values are unknown. Figure 31 shows the subgraph of Wikidata entity Linus (wd: Q47144). The subgraph has one statement which defines Linus's date of the birth property, and the property's value has some value (10CE). The RDF data model supports such values as a blank node in the RDF data model. In other words, the RDF data model can handle Wikidata values and *someValuesFrom* restriction in OWL and facilitates them in the model (Erxleben et al., 2014).



date of birth — 10 CE ← someValue
✎ edit
▾ 1 reference
imported from Wikimedia project — Malay Wikipedia
+ add reference
+ add value
+ add statement

Figure 31: Subgraph of Wikidata entity Linus (Linus, 2022).[14] .

[14] https://www.Wikidata.org/wiki/Q47144

6. The RDF data model only facilitates the label on the edges and does not provide the label on the nodes. Wikidata items are defined by their label, e.g., Wikidata item wd: Q76 has the label Barack Obama @en and does not represent a node in the RDF data model (Vrgoč et al., 2021), (Angles et al., 2022).

7. Wikidata follows the proper order, and each Wikidata item page is defined in an order, e.g., the entity is explained by its label. Second, the description part further elaborates on the Wikidata entity. Third, the alias's part gives some other name for Wikidata entities. Fourth, statements define further information about the entities. In addition, qualifiers and references also come with statements and proof of the statements. Fifth, sitelinks determine Wikidata entities represented in other projects, e.g., Wikimedia, etc. Syntactic interoperability has some missing functionalities. For instance, when Wikidata has been converted from the wikibase data model to the RDF data model through the RDF exports, some functionalities are ignored by the RDF exports, e.g., the order of Wikidata. In other words, the RDF exports are not reliable using syntactic interoperability (Erxleben et al., 2014).

8. Wikidata has Wikidata ontology [15] (Wikiproject Ontology, 2022), which is not complicated and easily presentable to another schema language. RDF data model easily represents the Wikidata ontology class hierarchy and facilitates semantic interoperability between two data models. For instance, the "instance of" is a Wikidata property like the "rdf: type" in the RDF Schema (Brickley, Guha, & McBride, 2014). Similarly, the Wikidata property "subClassOf" is the same as "rdfs: subClassOf" in the RDF Schema. Like the RDFS/OWL, Wikidata ontology defines the relationship of Wikidata items in the graph and does not provide a formal predefined Wikidata ontology; hence it is easily expressible in RDFS/OWL (Piscopo & Simperl, 2018), (Erxleben et al., 2014), (Baskauf & Baskauf, 2021).

   Figure 32 shows Wikidata human class representation and mapping Wikidata properties to RDFS/OWL properties. RDFS/OWL property starts with the prefixes rdfs and owl, and on the other hand, the prefix wdt defines Wikidata properties. Both properties (RDFS/OWL and Wikidata) are written on the edges so that mapping explains easily (Piscopo & Simperl, 2018), (Haller, Polleres, Dobriy, Ferranti & Rodr´ıguez Mendez, 2022).

[15] https://www.wikidata.org/wiki/Wikidata:WikiProject_Ontology

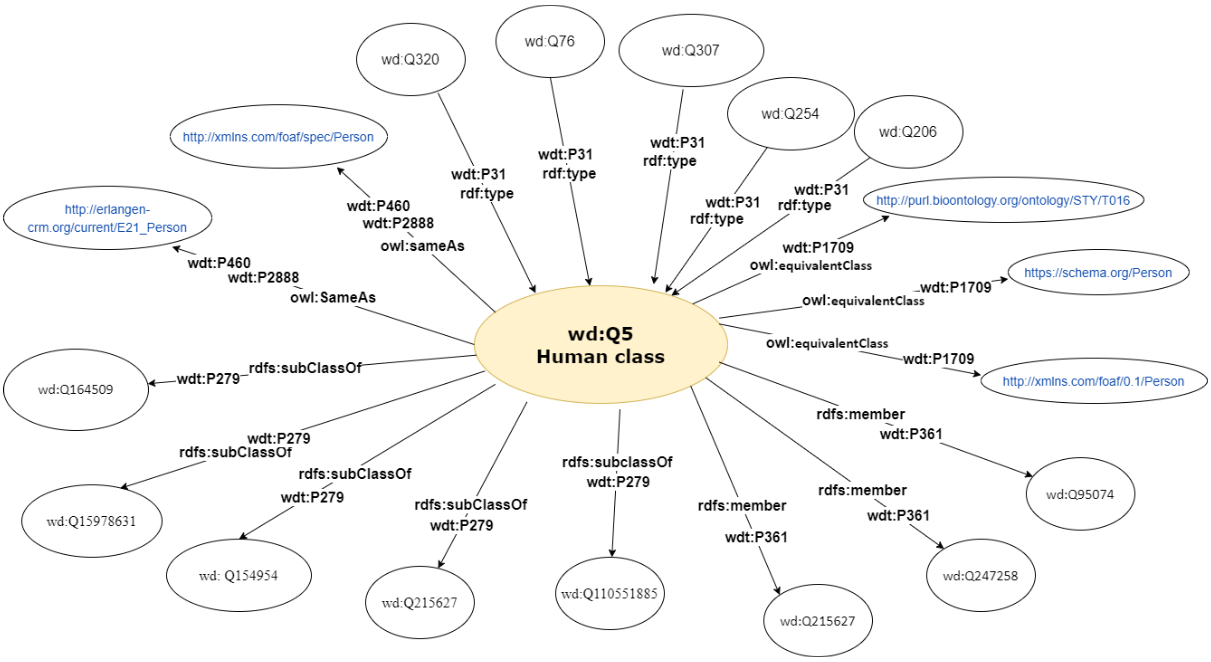


Figure 32: Wikidata properties mapping to RDF properties (Piscopo & Simperl, 2018).

Figure 33 shows the mapping of Wikidata properties into the RDFS/OWL during the semantic interoperability in the RDF data model. The statement [Barack Obama (Q76) instance of (P31) Human (wd: Q5)] has been taken for the mapping of Wikidata properties to the RDF data model (Haller, Polleres, Dobriy, Ferranti & Rodr´ıguez Mendez, 2022).

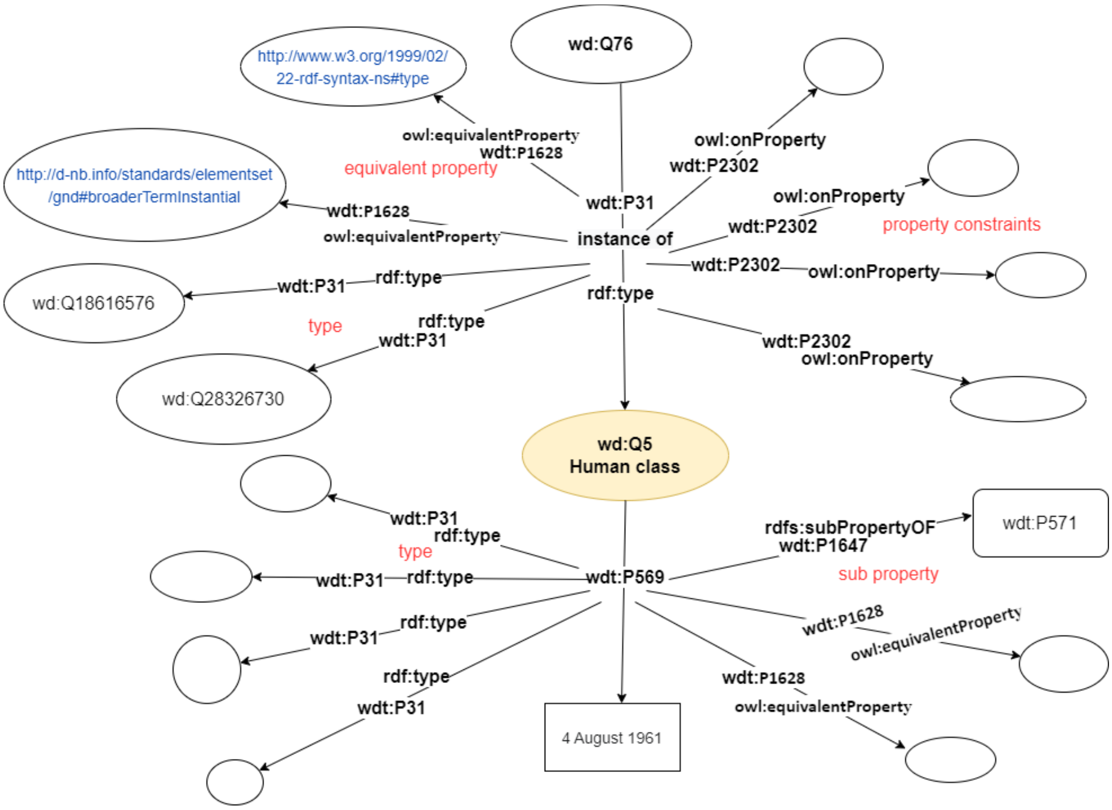


Figure 33: Wikidata properties and constraints representation in RDF/OWL (Haller, Polleres, Dobriy, Ferranti & Rodr´ıguez Mendez, 2022).

3.6 Analysis of the suitability of the PG data model for Wikidata

The following results are produced after the representation of figure 22 in the Property Graph data model which are as follows:-

1. The representation of the example using a Property Graph is more compact because qualifiers and references are defined as key-value pairs on edges. This reduces the complexity of searching information, and the storage capacity needed, because it generates a smaller number of nodes. However, there is no serialization format to represent Wikidata graphs using the Property Graph data model (Vrgoč et al., 2021).

   - The size of the Property Graph data model for Wikidata does not change whenever new qualifiers and references are added to Wikidata. The Property Graph data model facilitates Wikidata qualifier and references on the edge attribute as a key-value pair. Only the edges attributes are updated whenever new qualifiers are added to the statement. In addition, the size of the graph remains the same, and query response is faster because the Property Graph has one big graph, which makes searching easy (Angles & Gutierrez, 2017).

   - The Cypher query length is usually smaller in size. It remains the same whenever the new statement and qualifiers are added (Angles et al., 2022).

2. The Property Graph data model is unable to fully represent Wikidata qualifiers and references on the edges. However, the Property Graph data model represents Wikidata entities as a string rather than QNode (Angles et al., n.d.), (Vrgoč et al., 2021).

Figure 34 shows one of the statements of figure 22, which doesn't show a complete Property Graph data model.

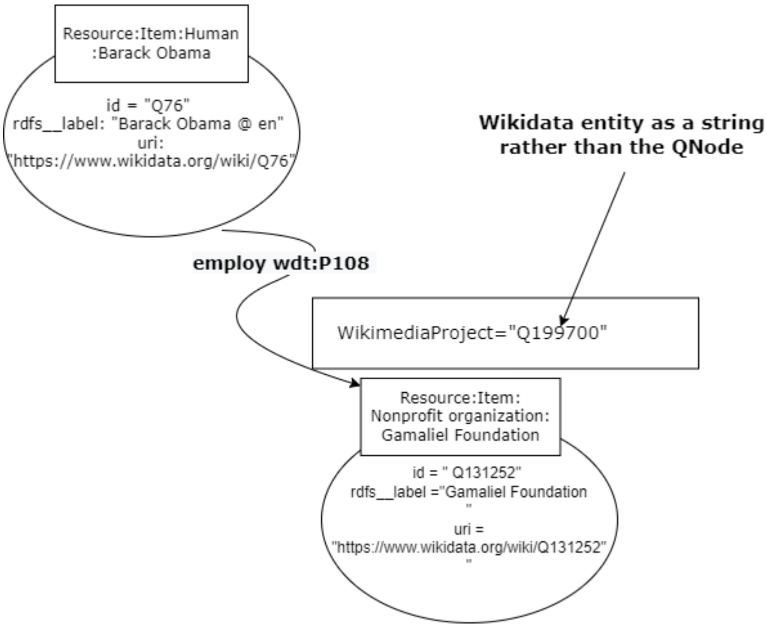

Resource:Item:Human :Barack Obama
id = "Q76"
rdfs__label: "Barack Obama @ en"
uri: "https://www.wikidata.org/wiki/Q76"
Wikidata entity as a string rather than the QNode
employ wdt:P108
WikimediaProject="Q199700"
Resource:Item: Nonprofit organization: Gamaliel Foundation
id = " Q131252"
rdfs__label ="Gamaliel Foundation "
uri = "https://www.wikidata.org/wiki/Q131252" "

Figure 34: Incomplete data model representing the Wikidata qualifiers and references (Vrgoč et al., 2021).

3. Wikidata entity has many labels in different languages, e.g., English, Spanish, etc. However, the Property Graph data model does not facilitate to model all the languages. It considers only the first language and skips the rest of the languages while representing the metadata about the Wikidata entities in the Property Graph data model (Angles & Gutierrez, 2017), (Eells et al., 2021).

   Figure 35 shows example of not supporting multivalued attributes of nodes for defining metadata of Wikidata.

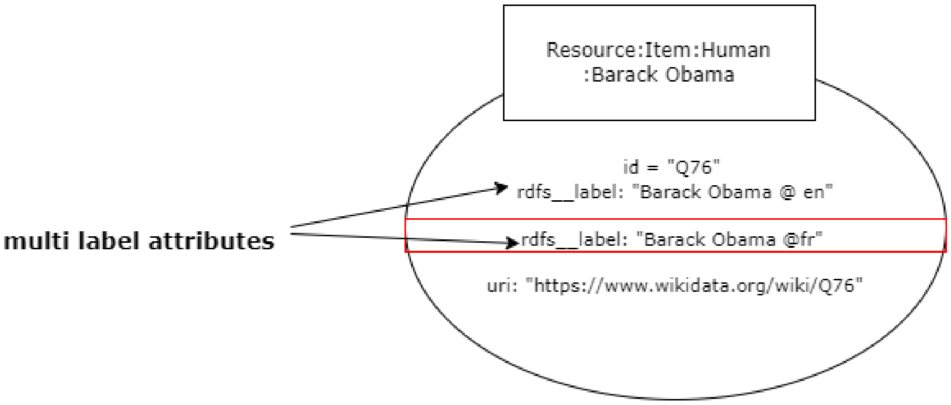

Resource:Item:Human
:Barack Obama
id = "Q76"
rdfs__label: "Barack Obama @ en"
multi label attributes
rdfs__label: "Barack Obama @fr"
uri: "https://www.wikidata.org/wiki/Q76"

Figure 35: Example of not supporting Wikidata multilabel in Property Graph data model (Angles & Gutierrez, 2017) (Eells et al., 2021).

4. The Property Graph data model does not support multivalued properties on the edges. The Wikidata entity *The Next Generation* has a statement containing qualifier P453 which is presenting two different values "William Riker" and "Thomas Riker". While representing such statements in PG, it ignores the second value Thomas Riker and only considers the first value (Albertus Donkers, Yang & Baken, 2020).

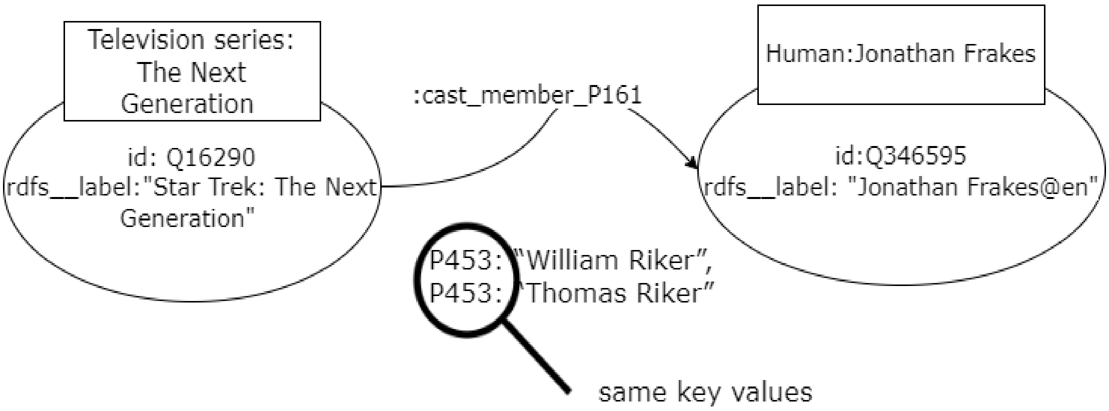


Figure 36: Multivalued property in Property Graph data model on edges (Albertus Donkers, Yang & Baken, 2020).

5. Property Graph data model uses the node label and edge label to model the real-world entities, e.g., Wikidata items (Vrgoč et al., 2021), (Angles et al., 2022).

6. The node type in Property Graph data model defines the schema and class hierarchy of Wikidata, and the key-value pair concerning the conceptual attributes inherent in the Wikidata entity. Wikidata new statements, qualifiers, and references for a Wikidata entity can be easily added and generated in Property Graph data model by using the node label in Property Graph data model (Zaho, Kook Han & Ri Kim, 2018).

3.7 Analysis of the suitability of the DG data model for Wikidata

The following results are produced after the representation of figure 22 using the Domain Graph data model which are as follows:-

1. The Domain Graph data model extends the RDF and Property Graph data models. For instance, the Property Graph data model uses the edges attributes to model the complex statement as a key-value pair. However, the Domain Graph data model provides more than the edge attributes by providing the edges as relationships and values as nodes. On the other hand, the Domain Graph data model facilitates the subject-predicate-object as target node, relation, source node, and rdf: statement is used to model the edges over edges and uses the reification approach (Vrgoč et al., 2021).

   Figure 37 shows one of the statements of above illustrated figure 22, which demonstrates how the Domain Graph is extended to the RDF and Property Graph data model (Vrgoč et al., 2021).

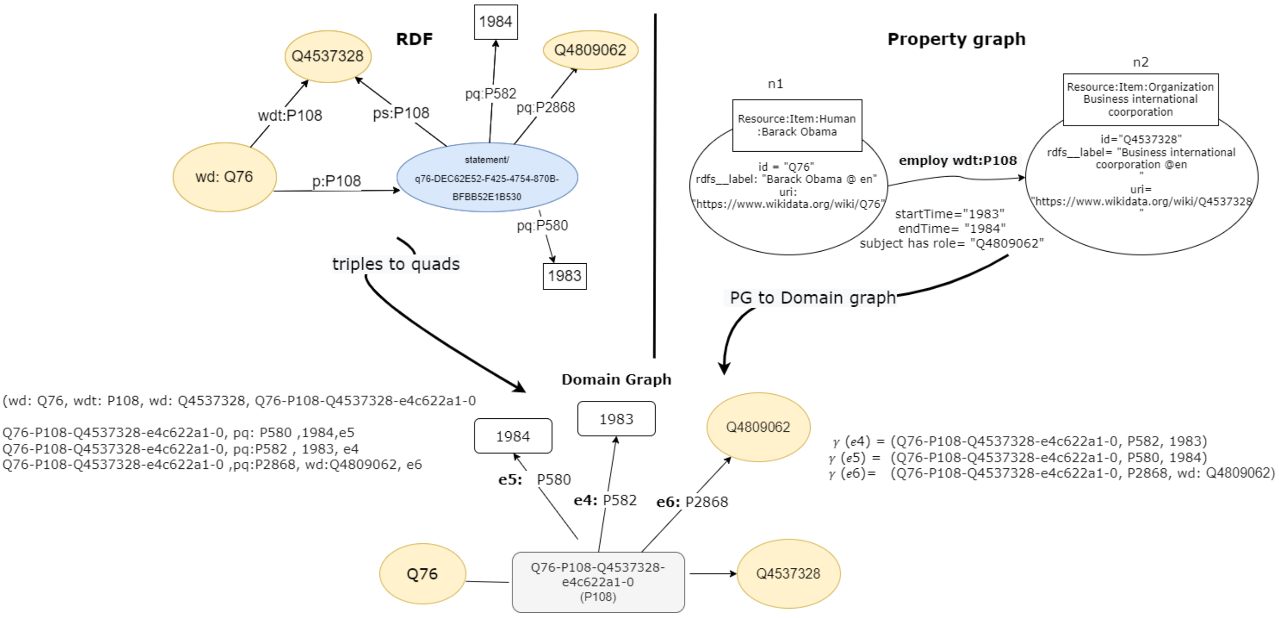

RDF
Q4537328
1984
Q4809062
wdt:P108
ps:P108
pq:P582
pq:P2868
wd: Q76
p:P108
statement/
q76-DEC62E52-F425-4754-870B-
BFBB52E1B530
pq:P580
1983
triples to quads
Property graph
n1
n2
Resource:Item:Human
:Barack Obama
id = "Q76"
rdfs__label: "Barack Obama @ en"
uri:
"https://www.wikidata.org/wiki/Q76"
employ wdt:P108
Resource:Item:Organization
Business international
coorporation
id="Q4537328"
rdfs__label= "Business international
coorporation @en
"
uri=
"https://www.wikidata.org/wiki/Q4537328
"
startTime="1983"
endTime= "1984"
subject has role= "Q4809062"
PG to Domain graph
Domain Graph
(wd: Q76, wdt: P108, wd: Q4537328, Q76-P108-Q4537328-e4c622a1-0
Q76-P108-Q4537328-e4c622a1-0, pq: P580 ,1984,e5
Q76-P108-Q4537328-e4c622a1-0, pq:P582 , 1983, e4
Q76-P108-Q4537328-e4c622a1-0 ,pq:P2868, wd:Q4809062, e6
1984
1983
Q4809062
e5: P580
e4: P582
e6: P2868
γ (e4) = (Q76-P108-Q4537328-e4c622a1-0, P582, 1983)
γ (e5) = (Q76-P108-Q4537328-e4c622a1-0, P580, 1984)
γ (e6)= (Q76-P108-Q4537328-e4c622a1-0, P2868, wd: Q4809062)
Q76
Q76-P108-Q4537328-
e4c622a1-0
(P108)
Q4537328

Figure 37: Extension of RDF and PG (Domain Graph) (Vrgoč et al., 2021).

2. Domain Graph data model uses quads representation to model Wikidata statements. It uses the edges as nodes and represented Wikidata statements, qualifiers and references. Each quad has defined by the edge id. The quad representation in Domain Graph data model make it easier for indexing over Wikidata statement (Vrgoč et a., 2021).

3. Representing Domain Graphs through KGTK (Knowledge Graph Tool Kit) shows the best performance in storing Wikidata statements and fast data retrieval (Chalupsky et al., 2021), (Chalupsky & Szekely, 2022), (Vrgoč et al., 2021), (Angeles et al., 2022). The representation of Wikidata in the Domain Graph through KGTK produces the four tab-

separated values (tsv) files. Each file contains separate information about the Wikidata statement. Each edge id stores the triples in a separate file. Only four separate files are updated whenever new qualifiers and references are added to Wikidata (Ilievski et al., 2020).

4. Knowledge Graph Tool Kit (KGTK) uses the kypher query language over the Domain Graph data model. Kypher is a simple and user-friendly query language, like the Cypher query language (Chalupsky et al., 2021). Kypher query extracts the larger subgraphs of Wikidata in a very short time. For instance, counting the number of instances in Wikidata, calculating the date of birth of all human beings in Wikidata provides the quick response by using the Kypher through KGTK in Domain Graph data model (Usc-isi-i2, 2022).

5. Kypher query extracts Wikidata entities specific information and can be useable in other queries and other applications. For instance, wd: Q76 (Barack Obama, 2022) is Wikidata entity and through KGTK Kypher query the Barack Obama family can be extracted which is further useful in other Kypher query (Usc-isi-i2, 2022) (Ilievski et al., 2020).

3.8 Result: Which model is best for Wikidata?

In this research, different graph models have been tested in order to find their suitability for Wikidata. The result shows that the Domain Graph data model through Knowledge Graph Tool Kit (KGTK) is the best option to represent Wikidata statement-level data because Domain Graph can extract the larger subgraph of Wikidata in seconds. In addition, no RAM requirement is required for downloading Wikidata dumps and using google collab; the Domain Graph data model is accessible on the local machine, and the RDF data model uses much memory to store Wikidata dumps and takes many days to store and retrieve Wikidata dumps. Most importantly, the Domain Graph data model facilitates Wikidata statement, i.e., qualifiers and references, by presenting the edges as nodes that have not been provided in the Property Graph data model.

Figure 38 shows how Domain Graph data model overcame the incompleteness in property graph data model representing edges as nodes.

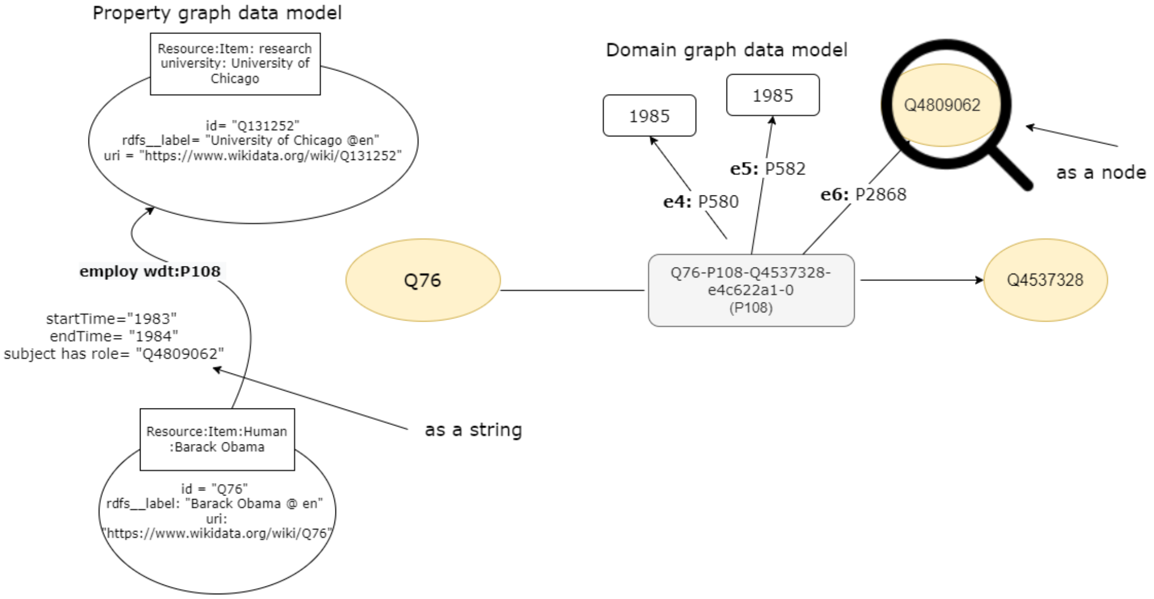

Property graph data model
Resource:Item: research university: University of Chicago
id= "Q131252"
rdfs__label= "University of Chicago @en"
uri = "https://www.wikidata.org/wiki/Q131252"
employ wdt:P108
startTime="1983"
endTime= "1984"
subject has role= "Q4809062"
as a string
Resource:Item:Human :Barack Obama
id = "Q76"
rdfs__label: "Barack Obama @ en"
uri: "https://www.wikidata.org/wiki/Q76"
Domain graph data model
1985
1985
Q4809062
as a node
e4: P580
e5: P582
e6: P2868
Q76
Q76-P108-Q4537328-e4c622a1-0 (P108)
Q4537328

Figure 38: demonstrates the incompleteness of the Property Graph data model while representing Wikidata qualifiers and the representation of edges as nodes in the Domain Graph data model (Vrgoč et al., 2021) (Angeles et al. l, 2022).

Table 13 summarizes the analysis results of three graph data models (RDF, PG, and DG) for Wikidata.

	RDF	**PG**	**Domain Graph**
Wikidata representation	Reification - n-ary relations - standard reification - singleton properties (Subject-predicate-object)	Node property, edge property as a key-value pair	Edges as nodes (Statement-id, node, label, node2) (Statement-id, qualified property, qualifier value, edgeId-qualifier) (Statement-id, reference property reference value, edgeId-reference)
Query language	SPARQL	Cypher	Kypher
Wikidata file format	In RDF, many formats - N-triples - N3 - Turtle - RDF/XML Standard format and W3C recommendation	No serialization format	Uses KGTK file format (tsv) to represent Domain Graph data models
Query response time	Very slow	medium	fast
Hardware requirement	maximum	maximum	minimum
Import and export Wikidata	Wikidata is available in RDF in a different format	No format is available for the Property Graph	Import Wikidata dumps in tsv file format and can export on tsv JSON.
Query length	High Challenging to write Complex queries, especially extracting subgraphs	Medium	Medium
Completeness	Yes,	No, it does not facilitate the edges as nodes and considers the QNodes as strings.	Yes

Table 13: shows the comparison between the three graph data models (RDF, Property Graph, and Domain Graph) (Angeles et al. l, 2022) (Vrgoč et al., 2021) (Ilievski et al., 2020).

3.9 Summary

This chapter had focus on analysis of different graph data models such as RDF, Property Graph and Domain Graph in order to explore which graph data model is more suitable for representing information in Wikidata. The example of Wikidata entity Barack Obama was selected to analyse the representation of Wikidata in the three graph data models. The results of each data model showed that the RDF data model has limited expressivity while representing Wikidata qualifiers and references. To do so, three reification techniques were discussed, but these techniques have substantial disadvantages. In addition, the Property Graph data model used the node label and edge label. It represented Wikidata qualifiers and references as edge attributes in the form of key-value pair, which is compact to model the statement level data. However, it is incomplete to model Wikidata entities on qualifier and reference values. Whereas Domain Graph data model has solved this problem by introducing the edges as nodes. Finally, the Domain Graph data model proved to be the best model to represent Wikidata statements.

PART III
General qualitative analysis of RDF, PG and DG

Chapter 4
Advantages of RDF, PG, and DG

The analysis presented in Chapter 3 showed what are the limitations of RDF, Property Graph and Domain Graph data models for representing information in Wikidata. This chapter presents a more general qualitative analysis of the three data models considered earlier. Chapter has two sections, and section 4.1 shows the qualitative comparison of RDF, Property Graph, and Domain Graph based on the results from chapter 3. In addition, section 4.2 shows more qualitative analysis by explaining the advantages and disadvantages of three graph data models for Wikidata.

4.1 Comparison of RDF, PG and DG

1. RDF data model uses the edge-centric approach, e.g., the node represents the subject, the edge represents the predicate, and the object is either the node or the literal values. Moreover, the RDF data model uses lists of edges; many of them are the properties of the nodes, which is why the cost of the RDF graph traversing is logarithmic. On the other hand, the Property Graph data model is the node-centric approach, the node represents the resources, and the edges define the relationship; each node and edge have its key-value pair. Compared to the RDF data model, the Property Graph data model considers the best graph traversal because it has only one big graph (Albertus Donkers, Yang & Baken, 2020) (Alocci et al., 2015). In addition, the Domain Graph data model uses the quads to model complex statements and facilitates edges as nodes and Knowledge Graph Tool Kit (KGTK) over Domain Graph provides the efficient data retrieval (Ilievski et al., 2020) (Angles et al., 2022), (Vrgoč et al., 2021).

2. RDF data model is a framework, and Knowledge Graphs uses the RDF framework to publish data so that it is exchangeable among different stakeholders. On the other hand, the primary purpose of the PG and DG development is to represent the data, store the data and efficiently query the data (Albertus Donkers, Yang & Baken, 2020). (Vrgoč et al., 2021), (Hernández et al., 2015), (Ilievski et al., 2020), (Angles et al., 2022).

3. The Property Graph data model provides the node and edge properties best for creating the temporal and weighted relationship and handling complex data efficiently. On the other hand, the RDF data model does not provide the node and edges properties and uses reification to model the complex data (Albertus Donkers, Yang & Baken, 2020), (Angles et al., 2022). Domain Graph data model uses both data model to represent Knowledge Graphs (Ilievski et al., 2020).

4. The RDF data model uses the SPARQL query language, the standard query language supported by W3C (World Wide Web Consortium). The SPARQL query provides application portability and can be used in many other RDF implementations. Wikidata query service can access SPARQL endpoints, one example of application portability. On the other hand, the Property Graph data model uses many graph query languages. One is a Cypher query language, and the Property Graph data model provides limited application portability, e.g., massive changes are required in the software before going to the alternative implementations. Moreover, the SPARQL query language implemented by the W3C standard makes SPARQL a good choice in software production. However, in the Property Graph data model, Neo4j develops many add-on components, e.g., plugins. The RDF triples can be accessed using these plugins as a Property Graph data model. These plugins are implemented by third-party contributors and do not guarantee future development (Alocci et al., 2015). The Domain Graph data model uses the kypher query language to extract the data, which is less similar to the Cypher query language (Chalupsky et al., 2021), (Ilievski et al., 2020).

5. The RDF data model uses a schema (RDFS and OWL) to represent the other data model, which is sometimes time-consuming and challenging to manage. In the case of Wikidata, it has its own ontology (Wikidata ontology) and the RDF data model does not take much time to represent Wikidata ontology; however, there are some features of Wikidata ontology that are difficult to be expressed in OWL axioms. Moreover, the RDF data model is suitable for building the ontologies from scratch and converting the other data structure into RDF triples. On the other hand, the Property Graph data model has ready-to-use solutions for substructure searches. Any data model can use this solution to represent the data in the PG data model (Alocci et al., 2015).

6. The RDF data model provides data interoperability which makes it the best model as compared to the Property Graph data model because the RDF data model provides syntactic, semantic and query interoperability as it has a formal data format, e.g., turtle, RDF/XML, N-Triples. Many other formats have the formal RDFS/OWL vocabularies which lead to the semantic and syntactic interoperability in RDF data model. However, on the other hand, the Property Graph data model lacks semantic and syntactic interoperability because of no standard vocabulary and data format (Alocci et al., 2015). In addition to this, due to the syntactic interoperability, Wikidata dumps are available in RDF different formats, e.g., turtle-triples, etc. Wikidata can access the SPARQL query endpoints. On the contrary, when importing Wikidata dumps in Neo4j, the neosemantics plugin must be installed in the Neo4j database, and the representation of Wikidata is not like the Property Graph data model. However, it is like the reified Property Graph data model due to the drawbacks of neo4j (Alocci et al., 2015).

7. The RDF data model supports the ontology modeling language, which gives meaning to the data. However, the Property Graph data model does not support the modeling language; therefore, Wikidata uses the node type to differentiate the Wikidata entities in

the Property Graph data models. For instance, the node label *Resource: Item: Human: Barack Obama* defines that Barack Obama is the Human and Wikidata item, Wikidata resource and RDF data model uses Wikidata ontology. Moreover, the existence of the ontology first leads to the schema interoperability framework. Different stakeholders can use this framework to model the data because the RDF data model is popular in publishing new data. Moreover, ontology enables the inferencing and reasoning of data (Albertus Donkers, Yang & Baken, 2020).

8. In data retrieval, the performance of the RDF data model is slower than the Domain Graph data model and Property Graph data model, due to RDF's internal structure. For instance, domain graph data model uses the kypher query language, which is also the KGTK representation and can query the other RDF knowledge graphs. Kypher is a simple and user-friendly query language, like the Cypher query language. Neither administrative setup nor database installation requires for the kypher. More significant Wikidata subset extracts with no time in the KGTK model by using the kypher. Kypher provides the facility to extract Wikidata entity subgraphs and uses them in another dataset for other purposes (Chalupsky et al., 2021).

9. Compared to the RDF and Property Graph data model, the Domain Graph uses KGTK and does not consume data storage capacity and also provides an ideal development environment for the users by running the jupyter notebook. Users can use the google Collab notebooks on the laptop and run the KGTK examples. Kypher query language is used to briefly run the more significant Wikidata subgraphs (Chalupsky et al., 2021).

4.2 RDF data model

4.2.1 Advantages

1. The RDF data model is the recommendation of the W3C (World Wide Web Consortium). It provides many standards and languages which make it possible to interchange data on the web (RDF - Semantic Web Standards, 2022), (Bergman, 2009), (Brickley & Guha, 2003).

2. RDF data model is quite a simple model. It can represent any data and distribute data among different applications (RDF - Semantic Web Standards, 2022). In addition, the RDF data model can also represent data that does not provide detailed information and is updated gradually over time. For instance, the representation of Wikidata in the RDF data model shows that the RDF data model is more accessible than the other data models. Wikidata itself has a data model (wikibase data model). Through the RDF exports, the wikibase data model can be represented in the RDF data model. Additionally, Wikidata has a Wikidata query service that extracts the different Wikidata subgraphs in the RDF data model with the help of SPARQL query (Baskauf & Baskauf, 2021).

3. The RDF data model is extendible, e.g., the schema and the detailed data information can be added anytime in the RDF data model. In other words, the RDF data model can extend, update, and adapt the data at any point. Wikidata in the RDF data model is an example because Wikidata updates over time through the RDF exports provided in the RDF data model (Bergman, 2009).

4. The RDF data model has many serialization formats, e.g., RDF/XML, n-triples, turtle, n3, JSON-LD, etc., representing the data in the RDF document. Each serialization format has advantages, e.g., the turtle serialization format is user-friendly and understandable. The n-triple format is straightforward, and the N-quads serialization format represents multiple RDF graphs—the JSON-Ld. Serialization format interacts with the database via API. Additionally, RDF/XML serialization format is the standard format and has been introduced by the W3C (World Wide Web Consortium) (RDF - Semantic Web Standards, 2022) (Bergman, 2009) (Beckett, Prud'hommeaux & Carothers, 2014). Furthermore, Wikidata dumps are available in different RDF serialization formats. They are updated daily because Wikidata uses the central Linked open data approach rather than the distributed, and the RDF data model has a standard data format ("Zenodo - Research. Shared.", 2022), (Angles et al., n.d.).

5. RDF data model has a schema unbound, and this feature can make it among the best model. The schema in the RDF data model has overcome data integration issues. The RDF data model is just a model, but the RDF schema gives added power and semantics to this model. The RDF schema gives the semantics to the data represented in the RDF

data model before publishing it to the web. However, RDFS has a non-standard and a non-fixed layer of metamodeling which leads to the dual representation of the elements in metamodeling architecture and creates a problem for the modelers while giving the semantics to the data (Pan & Horrocks, 2003). However, the OWL is the extended version of RDFS. RDF, RDFS, and SPARQL are used inside the OWL. OWL is the recommendation of the W3C, and it is the standard language to give semantics to the data (McGuinness & Harmelen, 2004), (Bergman, 2009), (Brickley, Guha and McBride, 2014).

6. RDF data model has a standard query language which is the recommendation by W3C (World Wide Web Consortium). It extracts the RDF triple from the RDF triple stores. SPARQL query can extract the subgraph of RDF without knowing about the data. Furthermore, SPARQL query extracts whatever is represented in the RDF databases (Bergman, 2009) (Hernández, Hogan, Riveros, Rojas & Zerega, 2016). Wikidata has a query service to extract the subgraph of Wikidata entities via SPARQL endpoints. This is due to the query interoperability provided by the RDF data model (Wikidata query service, 2022).

Figure 39 shows the data interoperability in the RDF data model by providing syntactic, semantic, and query interoperability.

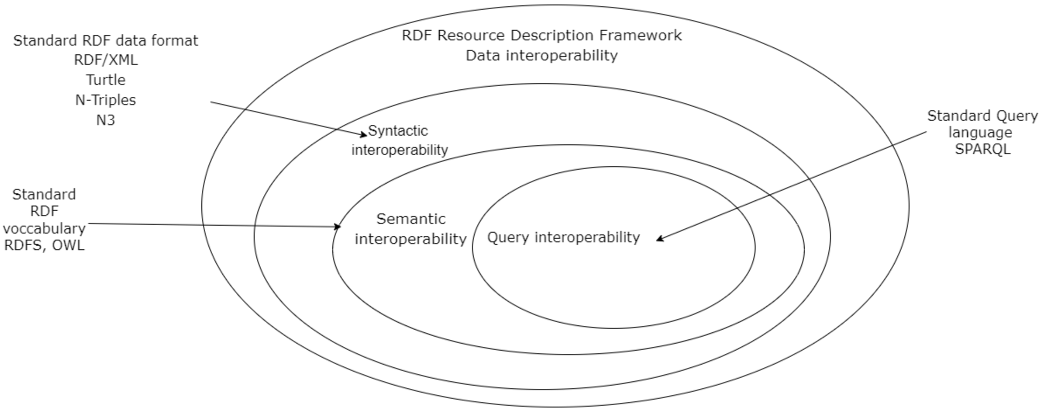


Figure 39: RDF Data interoperability (Angles et al., n.d.)

4.2.2 Disadvantages

1. The RDF data model does not represent the data in a specific order, and serialization formats do not establish mechanisms to indicate a particular order (Erxleben et al., 2014).

2. There is a need for extra memory for the storage of RDF dumps, especially the large Knowledge Graphs (KGs) (Vrgoč, D. et al., 2021).

4.3 Property Graph data model

4.3.1 Advantages

1. The Property Graph data model can efficiently represent and manage complex and extensive data structures in key-value pairs. Additionally, it provides a non-relational and schema-less database system that can easily manage different data that resides on the internet and updates over time (Zaho, Kook Han & Ri Kim, 2018), (Angles, 2018).

2. The Property Graph data model is beneficial to model the diverse, informative metadata and their relationship because it provides the labels and properties on both the nodes and the edges. Node properties are the key-value pair, and the key is used to represent the metadata and ontological vocabularies. Moreover, the node label is vital because it efficiently provides the conceptual schema and hierarchy. The node label in the Property Graph data model is like the rdf: type, which decides the node type, but node labels are much more efficient than the rdf: type. For instance, new subgraphs and schema can be easily added and generated by using the node label in PG. (Zaho, Kook Han & Ri Kim, 2018).

4.3.2 Disadvantages

1. The property graph data model has no standard format to serialize data. (Angles et al., n.d.), (Hernández, Hogan, Riveros, Rojas & Zerega, 2016).

2. The property graph data model has no standard semantics or foundational ontology to represent the data. Due to the lack of standard semantics or the standard data format, syntactic and semantic interoperability is not possible. Additionally, there is no concept of constraints on the classes and the properties when representing information using the Property Graph data model (Zaho, Kook Han & Ri Kim, 2018), (Angles et al., n.d.).

3. Property graph data model does not facilitate the multivalued attributes (Vrgoč, D. et al., 2021).

4.4 Domain Graph data model

4.4.1 Advantages

1. The Domain Graph data model is suitable for modelling knowledge graphs that include statements about statements and good to represent the higher-arity relations by modeling the real-world entities (Chalupsky et al., 2021).
2. The Domain Graph data model is suitable to model the real-world entities by defining the edges as nodes (Vrgoč et al., 2021).

3. Kypher is one of the query languages available for querying domain graphs. It has been created by Cypher to better support queries and updates operations on domain graphs. Kypher is proprietary and only supported by the tool KGTK (Chalupsky et al., 2021).

4.4.2 Disadvantages

1. Difficult to manage the domain graph data model (hypergraphs) due to its complex modeling structure (Chalupsky et al., 2021).

4.5 Results: General qualitative of three graph data models

RDF	**PG**	**DG**
1. It is a standard data model and recommendation of W3C (*World Wide Web Consortium*). **2.** RDF is supported by additional modelling languages such as RDFS and OWL **3.** It has a demand in the industry due to the standard. **4.** It has standard query language, e.g., SPARQL language **5.** It uses reification to model statement-level data, e.g., Wikidata. **6.** It supports data interoperability (syntactic, semantic, query). **7.** It is No good to follow the order. **8.** It does not keep any specific order for the triples exported using an existent serialization format. **9.** Reification generates many nodes and literals, which affect the performance.	**1.** It is not a standard model. **2.** It has no schema language. **3.** It has a many different query languages (some of them are proprietary, such as Cypher). **4.** It presents the knowledge graphs as a real-world entity where each node is used to define the entities, and edges define the relationship between the entity. Node and edge have attributes as key-value pairs. **5.** The node properties are used to define the metadata about the real-world entities as a key-value pair. **6.** It provides the facility to model the additional information on the attribute of the edges. **7.** Each node has a type that defines the schema class hierarchy etc. **8.** Support multilabel for one node. **9.** It does not support multivalued attributes on edges. **10.** It does not facilitate the edge as nodes.	**1.** Highly recommended to represent complex knowledge graphs that represent statements about statement. **2.** Statement are represented as quads of the form <Node1 label node2 ids> where label defines the relation between node one and node 2, and each statement is defined by the id, which further defines additional information of the statement. **3.** Efficient data model.

Table 14: Analysis result of three Graph data models (RDF, PG, and DG) (Angeles et al., 2022) (Vrgoč et al., 2021) (Ilievski et al., 2020).

4.6 Summary

This chapter discussed the general qualitative comparison, advantages, and disadvantages of three graph data models (RDF, Property Graph, and Domain Graph). The qualitative analysis showed that the RDF data model is widely prevalent among stakeholders compared to the Property Graph and Domain Graph data models. In addition, the RDF data model is the recommendation of the World Wide Web (W3C) standard, and many organizations and companies are using the RDF framework to publish their Knowledge Graphs (KGs). However, the RDF data model increased the cost of manipulating KGs. Moreover, the analysis of the Property Graph showed that it is a compact model and provided feasibility to manage large Knowledge Graphs (KGs). It has lower cost as compared to the RDF data model. On the contrary, the Property Graph data model needs more formal semantics and ontology, and due to this it is not so popular among the stakeholders. The analysis of the Domain Graph data model showed that KGTK uses the Domain graph data model, which is efficient for modeling Knowledge Graphs and supported edges as nodes to model complex statements.

PART IV
Summary and outlook

Chapter 5
Conclusion and Future Work

The qualitative analysis presented in Chapter 4 showed the comparison and what are the advantages and disadvantages of three graph data models (RDF, PG and DG). This chapter concludes the analysis of RDF, PG and DG for representing Wikidata. In addition, which data model shows the best representation of KGs. Chapter has four sections and sections 5.1 defines the summary of the thesis, section 5.2 defines the limitations of the thesis and section 5.3 defines the Future work and section 5.4 concludes the thesis research.

5.1 Summary

The thesis analyzed the three graph data models (RDF, Property Graph, and Domain Graph) based on the problem statement and the context of the thesis. Two qualitative analyses were conducted in this thesis. The research showed the representation of Wikidata in RDF, Property Graph, and Domain Graph. The research analysis showed that the Domain graph data model represented the Wikidata statement in the best way compared to the RDF and Property Graph data models. The RDF data model produced many redundant triples, provided limited scalability, and required more hardware, e.g., RAM, to store Wikidata RDF dumps and retrieve data. The Property Graph data model was not a complete data model for representing Wikidata entities. The Domain Graph data model overcame the limitations presented in the Property Graph, RDF graph data model and facilitated the edges as nodes to model Wikidata. In addition, the thesis conducted a general qualitative analysis and presented the comparison between three graph data models, their advantages, and disadvantages. (Results of qualitative analysis)

5.2 Limitations

During the thesis analysis, the selection of Wikidata entities was minimal because Wikidata as Knowledge Graph is vast in size, and Wikidata dumps are available in RDF data format. However, they all have massive sizes, and loading Wikidata for the first time in Stardog database takes a minimum of five days. A lot of memory is required for downloading Wikidata dumps, and at least 512GB RAM is required to analyze Wikidata. So, the analysis was conducted on a limited number of Wikidata entities due to memory insufficiency.

5.3 Future work

The representation of the knowledge graphs in the Domain Graph data model is the best option compared to other graph data models because it provides the minimum data storage and fastest data retrieval compared to RDF and PG data models. In addition, KGTK (Knowledge Graph Tool Kit) uses the Kypher query language to extract Wikidata qualifiers and references, which gives the best results for larger Wikidata graphs. However, there is a need to develop more operations on the Kypher query. For instance, kypher does not facilitate writing nested queries; however, it is possible to extract the subgraphs of Wikidata and present this subgraph in a separate KGTK file which can be further used in another database (Chalupsky et al., 2021). However, the nested queries for Wikidata are not supported by the KGTK, which may be developed in the future. Moreover, many new features can be introduced in the KGTK pipeline for efficiently representing Knowledge Graphs (KGs).

5.4 Conclusion

In conclusion, the thesis represented the qualitative analysis of three technologies RDF, Property Graph data model, and Domain Graph data model.

Conclusion for Q1: Different reification techniques showed the representation of Wikidata in the RDF data model, and the limitations in the RDF data model affected Wikidata's performance. For instance, whenever new nodes are added to the graph, it increases the number of triples in the RDF graph. So, the Property Graph data model overcame the limitations of the RDF data model by presenting Wikidata qualifiers as an edges attribute which increased the performance of Wikidata but was unable to present Wikidata ultimately. The Property Graph data model has not supported the edges as nodes and multivalued attributes. The Domain Graph data model used the quads in order to model the hypergraphs, mainly designed for Wikidata. It uses both the RDF data model and the Property Graph data model. The Domain Graph data model overcame the limitations presented in RDF and the Property Graph for Wikidata. In addition, the Domain Graph provided the best query response time compared to RDF and Property Graph data models. Domain Graph data model in KGTK used significantly less storage and hardware requirement than RDF, Property Graph and it was the best option for Wikidata.

Conclusion for Q2: The general qualitative analysis has been shown by comparing three graph data models. The author's opinion showed that each data model has several advantages and disadvantages representing complex statements and helps the reader to select the best option for model Knowledge Graphs. For instance, the best point of the RDF data model is a standard data model and recommendation of the World Wide Web Consortium (W3C), and this has been widely popular among different stakeholders. The Property Graph's best point is that it provides the node and edge properties to model complex statements. Schema can be easily represented by adding the label to the node and edge. Hence, the PG data model is compact. The Domain graph data model is the best for modeling Wikidata and makes it user-friendly for the readers by executing more significant Wikidata subgraphs on a Local machine. However, the Domain Graph data model is a complex structure to model Knowledge graphs because these are hypergraphs and challenging to visualize the larger graphs.

Reference

Albertus Donkers, A., Yang, D., & Baken, N. (2020). Linked Data for Smart Homes: Comparing RDF and Labeled Property Graphs. Conference: LDAC2020 - 8Th Linked Data in Architecture And Construction Workshop at Dublin, Ireland. Retrieved from http://linkedbuildingdata.net/ldac2020/files/papers/02paper.pdf

Alocci, D., Mariethoz, J., Horlacher, O., Bolleman, J., Campbell, M., & Lisacek, F. (2015). Property Graph vs RDF Triple Store: A Comparison on Glycan Substructure Search. PLOS ONE, 10(12), e0144578. doi: 10.1371/journal.pone.0144578

Angles, R., & Gutierrez, C. (2017). An introduction to Graph Data Management. doi: https://doi.org/10.48550/arXiv.1801.00036

Angles, R., Thakkar, H., & Tomaszuk, D. (n.d). RDF and Property Graphs Interoperability: Status and Issues (Vol-2369).

Angles, R., & Gutierrez, C. (2008). Survey of graph database models. ACM Computing Surveys, 40(1), 1-39. doi:10.1145/1322432.1322433

Angles, R., Hogan, A., Lassila, O., Rojas, C., Schwabe, D., Szekely, P., & Vrgoč, D. (2022). Multilayer graphs: A unified data model for graph databases. Proceedings of the 5th ACM SIGMOD Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA). doi:10.1145/3534540.3534696

Angles, R., Arenas, M., Barceló, P., Hogan, A., Reutter, J., & Vrgoč, D. (2018). Foundations of modern query languages for graph databases. ACM Computing Surveys, 50(5), 1-40. doi:10.1145/3104031

Angles, R. (2018). The Property Graph Database Model, Vol-2369. http://ceur-ws.org/Vol-2369/paper01.pdf

Antoniou, G., & Van Harmelen, F. (2004). A semantic web primer. MIT press.

Arenas, M., Gutierrez, C., & P´erez, J. (2022). Foundations of RDF Databases. Retrieved 11 October 2022, from https://users.dcc.uchile.cl/~jperez/papers/reasoning-web09.pdf. ez/papers/reasoning-web09.pdf.

Barack Obama. (2022, November 16). Wikidata entity Barack Obama. Retrieved November 16, 2022, from https://www.wikidata.org/wiki/Q76

Baskauf, S., & Baskauf, J. (2021). Using the W3C Generating RDF from Tabular Data on the Web recommendation to manage small Wikidata datasets. Semantic Web, 1-23. DOI: 10.3233/sw-210443

Beckett, D., Prud'hommeaux, E., Carothers, G., & Berners-Lee, T. (2014). RDF 1.1 Turtle. W3C, Inc. Retrieved from https://www.w3.org/TR/turtle/

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. Sci Am. 2001; 284:34-43.

Bergman, M. (2009). Advantages and Myths of RDF. Structured Dynamics LLC. Retrieved from https://web.archive.org/web/20180412044958id_/http://www.mkbergman.com/wp-content/themes/ai3v2/files/2009Posts/Advantages_Myths_RDF_090422.pdf.

Brickley, D., Guha, R., & McBride, B. (2014). RDF Schema 1.1 W3C Recommendation 25 February 2014. Retrieved from https://www.w3.org/TR/rdf-schema.

Brickley, D., & Guha, R. (2003). RDF Vocabulary Description Language 1.0: RDF Schema. Retrieved from https://www.w3.org/2001/sw/RDFCore/Schema/200212bwm/

Buchgeher, G., Gabauer, D., Martinez-Gil, J., & Ehrlinger, L. (2021). Knowledge graphs in manufacturing and production: A systematic literature review. IEEE Access, 9, 55537-55554. doi:10.1109/access.2021.3070395

Chalupsky, H. and Szekely, P. (2022). Hybrid Structured and Similarity Queries over Wikidata plus Embeddings with Kypher-V. Retrieved from: https://wikidataworkshop.github.io/2022/papers/Wikidata_Workshop_2022_paper_7722.pdf

Chalupsky, H., Szekely, P., Ilievski, F., Garijo, D. and Shenoy, K., (2021). Creating and Querying Personalized Versions of Wikidata on a Laptop. Retrieved from https://arxiv.org/pdf/2108.07119.pdf

Chaudhri, V., Baru, C., Chittar, N., Dong, X., Genesereth, M., Hendler, J., . . . Wang, K. (2022). Knowledge graphs: Introduction, history and, Perspectives. AI Magazine, 43(1), 17-29. doi:10.1609/aimag.v43i1.19119

Cypher query language - developer guides (2022) Neo4j Graph Data Platform. Retrieved: November 14, 2022, from https://neo4j.com/developer/cypher/

Eells, A., Shimizu, C., Zhou, L., Hitlzer, P., Gonzales, S., & Rehberger, D (2021). Aligning Patterns to Wikibase Model *. Retrieved from https://par.nsf.gov/servlets/purl/10342675.

Ehrlinger, L. and Wöß, W., (2016). Towards a Definition of Knowledge Graphs. https://ceur-ws.org/Vol-1695/paper4.pdf?ref=https://githubhelp.com

Elizbeth I England. (2022). Wikidata entity Elizbeth I England. Retrieved September 22, 2022, from https://www.wikidata.org/wiki/Q7207

Erxleben, F., Günther, M., Krötzsch, M., Mendez, J., & Vrandečić, D. (2014). Introducing wikidata to the linked data web. The Semantic Web – ISWC 2014, 50-65. doi:10.1007/978-3-319-11964-9_4

Fernández-Álvarez, D., Frey, J., Labra Gayo, J. E., Gayo-Avello, D., & Hellmann, S. (2021). Approaches to measure class importance in Knowledge Graphs. PLOS ONE, 16(6), e0252862. https://doi.org/10 Ehrlinger and Wöß.1371/journal.pone.0252862

Govindapillai, S., Soon, L., & Haw, S. (2021). An empirical study on resource description framework reification for trustworthiness in knowledge graphs. F1000Research, 10, 881. doi:10.12688/f1000research.72843.2

Guo, K., Khanna, D., Diefenbach, D., Perevalov, A., & Both, A. (2022). WikidataComplete – an easy-to-use method for rapid validation of text-extracted new facts applied to the Wikidata knowledge graph. The Semantic Web: ESWC 2022 Satellite Events, 118-122. doi:10.1007/978-3-031-11609-4_22

Gutierrez, C., Hurtado, C. A., Mendelzon, A. O., & Pérez, J. (2011). Foundations of Semantic Web Databases. Journal of Computer and System Sciences, 77(3), 520-541. doi:10.1016/j.jcss.2010.04.009

Hall, A., Terveen, L., & Halfaker, A. (2018). BOT detection in wikidata using behavioral and other informal cues. Proceedings of the ACM on Human-Computer Interaction, 2(CSCW), 1-18. doi:10.1145/3274333

Haller, A., Polleres, A., Dobriy, D., Ferranti, N., & Rodr´ıguez Mendez, S. (2022). The Semantic Web. Lecture Notes in Computer Science, 43-58. doi: 10.1007/978-3-031-06981-9

Hayes, P. (2004). RDF Semantics. W3C Recommendation 10 February 2004. Retrieved from https://www.w3.org/TR/rdf-mt/

Hayes, P.J. & Patel-Schneider, P.F. (2014). RDF 1.1 Semantics W3C Recommendation 25 February 2014. Retrieved from https://www.w3.org/TR/rdf11-mt/#literals-anddatatypes

Hernández, D., Hogan, A., Riveros, C., Rojas, C., & Zerega, E. (2016). Querying wikidata: Comparing SPARQL, relational and graph databases. Lecture Notes in Computer Science, 88-103. doi:10.1007/978-3-319-46547-0_10

Hernández, D., Hogan, A., & Krötzsch, M. (2015). Reifying RDF: What Works Well with Wikidata? Retrieved from https://www.researchgate.net/publication/283865828_Reifying_RDF_What_works_well_with_wikidata

Hoffart, J., Suchanek, F. M., Berberich, K., Lewis-Kelham, E., De Melo, G., & Weikum, G. (2011). YAGO2: Exploring and Querying World Knowledge in Time, Space, Context, and Many Languages. Proceedings of the 20th International Conference Companion on World Wide Web - WWW '11. doi:10.1145/1963192.1963296

Hogan, A., Blomqvist, E., Cochez, M., D'amato, C., Melo, G. D., Gutierrez, C., . . . Zimmermann, A. (2022). Knowledge graphs. ACM Computing Surveys, 54(4), 1-37. doi:10.1145/3447772

Hogan, A., Arenas, M., Mallea, A., & Polleres, A. (2014). Everything you always wanted to know about blank nodes. Journal of Web Semantics, 27-28, 42-69. doi:10.1016/j.websem.2014.06.004

Ilievski, F., Garijo, D., Chalupsky, H., Divvala, N. T., Yao, Y., Rogers, C., . . . Szekely, P. (2020). KGTK: A toolkit for large knowledge graph manipulation and analysis. Lecture Notes in Computer Science, 278-293. doi:10.1007/978-3-030-62466-8_18

IRIs/RDFConceptsProposal - RDF Working Group Wiki. W3.org. (2022). Retrieved 2 September 2022, from https://www.w3.org/2011/rdf-wg/wiki/IRIs/RDFConceptsProposal.

Kroetzsch , M. and Weikum, G. (2016) "Journal of Web Semantics: Special Issue on Knowledge Graphs,"

Lampropoulos, G., Keramopoulos, E., & Diamantaras, K. (2020). Enhancing the functionality of augmented reality using Deep Learning, semantic web and knowledge graphs: A Review. Visual Informatics, 4(1), 32–42. doi: 10.1016/j.visinf.2020.01.001

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., . . . Bizer, C. (2015). DBpedia – a large-scale, multilingual knowledge base extracted from Wikipedia. Semantic Web, 6(2), 167-195. doi:10.3233/sw-140134

Linus. (2022). Wikidata entity Barack Obama. Retrieved August 16, 2022, from https://www.wikidata.org/wiki/ Q47144

Malyshev, S., Krötzsch, M., González, L., Gonsior, J., & Bielefeldt, A. (2018). Getting the most out of Wikidata: Semantic technology usage in Wikipedia's knowledge graph. Lecture Notes in Computer Science, 376-394. doi:10.1007/978-3-030-00668-6_23

Manola, F., & Miller, E. (2014). RDF Primer W3C Recommendation 10 February 2004. Retrieved from https://www.w3.org/TR/rdf-primer/

McGuinness, D., & Harmelen, F. (2004). OWL Web Ontology Language W3C Recommendation 10 February 2004.

Melnik, S., Mitra, P., & Decker, S. (2000). Framework for the semantic Web: an RDF tutorial. IEEE Internet Computing, 4(6), 68-73. doi: 10.1109/4236.895018

Neo4j documentation - NEO4J documentation. (2022). Retrieved November 12, 2022, from https://neo4j.com/docs/

Nguyen, V., Bodenreider, O., & Sheth, A. (2014). Don't like RDF reification? Proceedings of the 23rd International Conference on World Wide Web - WWW '14. doi:10.1145/2566486.2567973

Orlandi, F., Graux, D., & O'Sullivan, D. (2021). Benchmarking RDF Metadata Representations: Reification, Singleton Property, and RDF*. IEEE 15 International Conference On Semantic Computing (ICSC). Retrieved from https://fabriziorlandi.net/pdf/2021/ICSC2021_REF-Benchmark.pdf.

Pan, J. Z., & Horrocks, I. (2003). RDFS(FA) and RDF MT: Two semantics for RDFS. Lecture Notes in Computer Science, 30-46. doi:10.1007/978-3-540-39718-2_3

Piscopo, A., & Simperl, E. (2018). Who Models the World? Proceedings Of The ACM On Human-Computer Interaction, 2(CSCW), 1-18. doi: 10.1145/3274410 projects, C.to W. (2022) Wikibase/indexing/RDF dump format - mediawiki, Powered by MediaWiki. Wikimedia Foundation, Inc. Available at: https://www.mediawiki.org/wiki/Wikibase/Indexing/RDF_Dump_Format (Accessed: November 14, 2022).

Prud'hommeaux, E. and Seaborne, A., (2008). SPARQL Query Language for RDF W3C Recommendation 15 January 2008.Retreived from https://www.w3.org/TR/rdf-sparql-query/#introduction

RDF- Semantic Web standards (2022). RDF - Semantic Web Standards. Retrieved 31 August 2022 from https://www.w3.org/RDF/

Rodriguez MA, Neubauer P (2010) Constructions from dots and lines. Bull Am Soc Inf Sci Technol 36(6):35–41

Shrivastava, S. (2017). Bring rich knowledge of people, places, things and local businesses to your apps. Bing Blogs. July 12, 2017

Singhal, A. (2012). Introducing the Knowledge Graph: Things, not strings. Google Blog. May 16, 2012

Solheim, I., & Stølen, K. (2007). Technology research explained.

Statistics. (2022). Retrieved November 12, 2022, from https://www.wikidata.org/wiki/Wikidata:Statistics

Stardog Union. (2022). The enterprise knowledge graph platform: Stardog. Retrieved November 12, 2022, from https://www.stardog.com/

Star Trek the Next Generation (2022, November 1). Wikidata entity Star Trek the Next Generation. Retrieved November 1,2022, from https://www.Wikidata.org/wiki/Q16290

Usc-isi-i2. (2022). USC-ISI-i2/KGTK-notebooks: Tutorial and hands-on notebook on using the Knowledge Graph Toolkit (KGTK). Retrieved November 12, 2022, from https://github.com/usc-isi-i2/kgtk-notebooks

Uyar, A. and Aliyu, F.M. (2015) “Evaluating search features of Google Knowledge graph and Bing Satori,” Online Information Review, 39(2), pp. 197–213. Available at: https://doi.org/10.1108/oir-10-2014-0257.

Vrgoč, D. et al. (2021). “MillenniumDB: A Persistent, Open-Source, Graph Database.” Available at: https://arxiv.org/pdf/2111.01540.pdf.

Vrandečić, D., & Krötzsch, M. (2014). Wikidata: A free collaborative knowledgebase. Communications of the ACM, 57(10), 78-85. doi:10.1145/2629489

Wikidata query service. (2022). Wikidata Query Service. Retrieved November 8, 2022, from https://query.wikidata.org/

Wikidata toolkit - mediawiki. (2022). Retrieved November 12, 2022, from https://www.mediawiki.org/wiki/Wikidata_Toolkit

Wikiproject properties/reports/Datatypes. (2022). Retrieved November 12, 2022, from https://www.wikidata.org/wiki/Wikidata:WikiProject_Properties/Reports/Datatypes

Wikiproject Ontology. (2022). Retrieved November 12, 2022, from https://www.wikidata.org/wiki/Wikidata:WikiProject_Ontology

Zaho, Z., Kook Han, S., & Ri Kim, J. (2018). LPG Representation of the Reification of RDF. International Journal Of Engineering &Amp; Technology, 7(3.34), 562. doi: 10.14419/ijet.v7i3.34.19382

Zenodo - Research. Shared. (2022). Retrieved 5 September 2022, from https://zenodo.org/

Appendix A

Triples count after standard reification, n-ary relations and singleton property

Table 15: Standard reification (Manola & Miller, 2014).

Triple	Subject	Predicate	Object
1	q76-66C211F7-A8BB-42F6-98C8-3451C112629C	Subject	Wd: Q76
2	q76-66C211F7-A8BB-42F6-98C8-3451C112629C	Predicate	Wdt: P108
3	q76-66C211F7-A8BB-42F6-98C8-3451C112629C	Object	wd: Q4537781
4	-q76-66C211F7-A8BB-42F6-98C8-3451C112629C	Pq:P580	1985
5	q76-66C211F7-A8BB-42F6-98C8-3451C112629C	Pq:P582	1985
6	q76-66C211F7-A8BB-42F6-98C8-3451C112629C	Prov:wasDerivedFrom	reference /1474044d51cd60f38ca1b2260b3928b5b96aa88c
7	reference /1474044d51cd60f38ca1b2260b3928b5b96aa88c	Pr: P584	Reference_url
8	statement/q76-BDAB32FE-E9F9-4147-A1F6-354E9F27B487 (2)	Subject	Wd: Q76
9	statement/q76-BDAB32FE-E9F9-4147-A1F6-354E9F27B487	Predicate	Wdt: P108
10	statement/q76-BDAB32FE-E9F9-4147-A1F6-354E9F27B487	Object	wd: Q4537781
11	statement/q76-BDAB32FE-E9F9-4147-A1F6-354E9F27B487	Prov:wasDerivedFrom	reference/ 9cdd4f1d064faebc44a10fbd408afa604f3b89f6
12	reference/ 9cdd4f1d064faebc44a10fbd408afa604f3b89f6	Pr:P143	Q199700
13	statement/Q76-E1E721F4-3A3F-46FA-BF4E-0B851C621318	Subject	Wd: Q76
14	statement/Q76-E1E721F4-3A3F-46FA-BF4E-0B851C621318	Predicate	Wd: P108
15	statement/Q76-E1E721F4-3A3F-46FA-BF4E-0B851C621318	Object	wd: Q131252

16	statement/q76-A92C633A-CFE5-4F73-8D55-F4E8856FEC9C	Subject	Wd: Q76
17	statement/q76-A92C633A-CFE5-4F73-8D55-F4E8856FEC9C	Predicate	Wdt: P108
18	statement/q76-A92C633A-CFE5-4F73-8D55-F4E8856FEC9C	object	wd: Q3483312
19	statement/q76-A92C633A-CFE5-4F73-8D55-F4E8856FEC9C	Pq:P582	1991
20	statement/q76-A92C633A-CFE5-4F73-8D55-F4E8856FEC9C	Pq:P580	1991
21	statement/q76-A92C633A-CFE5-4F73-8D55-F4E8856FEC9C	Prov:wasDerivedFrom	reference/ fd60efb8e3d6b26135a2a84e 72ef62daf6517042
22	reference/ fd60efb8e3d6b26135a2a84e 72ef62daf6517042	Pr:P854	Refurl
23	reference /1474044d51cd60f38ca1b2260b3928b5b96aa88c	Pr: P1065	refurl
24	statementq76-DEC62E52-F425-4754-870B-BFBB52E1B530	Subject	Wd: Q76
25	statementq76-DEC62E52-F425-4754-870B-BFBB52E1B530	Predicate	Wdt:P108
26	statementq76-DEC62E52-F425-4754-870B-BFBB52E1B530	Object	wd: Q4537328
27	statementq76-DEC62E52-F425-4754-870B-BFBB52E1B530	Pq:P582	1984
28	statementq76-DEC62E52-F425-4754-870B-BFBB52E1B530	Pq:580	1983
29	statementq76-DEC62E52-F425-4754-870B-BFBB52E1B530	Pq: P2868	wd: Q4809062

Table 16: n-ary relations (Hernández et al., 2015) (Nguyen, Bodenreider, & Sheth, 2014).

Triples	Subject	Predicate	object
1.	Q76	p: P108	statement/q76DEC62E52-F425-4754-870BBFBB52E1B530
2.	statement/q76-DEC62E52-F425-4754-870B-BFBB52E1B530	Ps:P108	Wd: Q4537328
3.	statement/q76-DEC62E52-F425-4754-870B-BFBB52E1B530	Pq: P582	1984
4.	statement/q76-DEC62E52-F425-4754-870B-BFBB52E1B530	Pq:P580	1983
5.	statement/q76-DEC62E52-F425-4754-870B-BFBB52E1B530	Pq: P2868	Wd: Q4809062
6.	Wd: Q76	P:P108	statement/q76-66C211F7-A8BB-42F6-98C8-3451C112629C
7.	statement/q76-66C211F7-A8BB-42F6-98C8-3451C112629C	Ps:P108	Q4537781
8.	statement/q76-66C211F7-A8BB-42F6-98C8-3451C112629C	Pq:P580	1985
9.	statement/q76-66C211F7-A8BB-42F6-98C8-3451C112629C	Pq:P582	1985
10.	statement/q76-66C211F7-A8BB-42F6-98C8-3451C112629C	Prov: WasDerivedFrom	Refnode
11.	Refnode	Pr:P854	Ref URL
12.	Wd: Q76	Ps:P108	statement/q76-BDAB32FE-E9F9-4147-A1F6-354E9F27B487
13.	statement/q76-BDAB32FE-E9F9-4147-A1F6-354E9F27B487	Ps:P108	Q4537985
14.	statement/q76-BDAB32FE-E9F9-4147-A1F6-354E9F27B487	Prov: WasDerivedFrom	Refnode
15.	Refnode	Pr: P143	Q199700
16.	Wd: Q76	P:P108	statement/q76-A92C633A-CFE5-4F73-8D55-F4E8856FEC9C
17.	statement/q76-A92C633A-CFE5-4F73-8D55-F4E8856FEC9C	Ps:P108	Q3483312

18.	statement/q76-A92C633A-CFE5-4F73-8D55-F4E8856FEC9C	Pq:P580	1991
19.	statement/q76-A92C633A-CFE5-4F73-8D55-F4E8856FEC9C	Pq:P582	1991
20.	statement/q76-A92C633A-CFE5-4F73-8D55-F4E8856FEC9C	Prov: WasDerivedFrom	Refnode
21.	Refnode	Pr:P854	URL
22.	Refnode	Pr: P1065	URL
23.	Wd: Q76	P:P108	statement/Q76-E1E721F4-3A3F-46FA-BF4E-0B851C621318
24.	statement/Q76-E1E721F4-3A3F-46FA-BF4E-0B851C621318	Ps:P108	Q131252

Table 17 Singleton Properties (Hernández et al., 2015), (Nguyen, Bodenreider, & Sheth, 2014) (Orlandi et al., 2021).

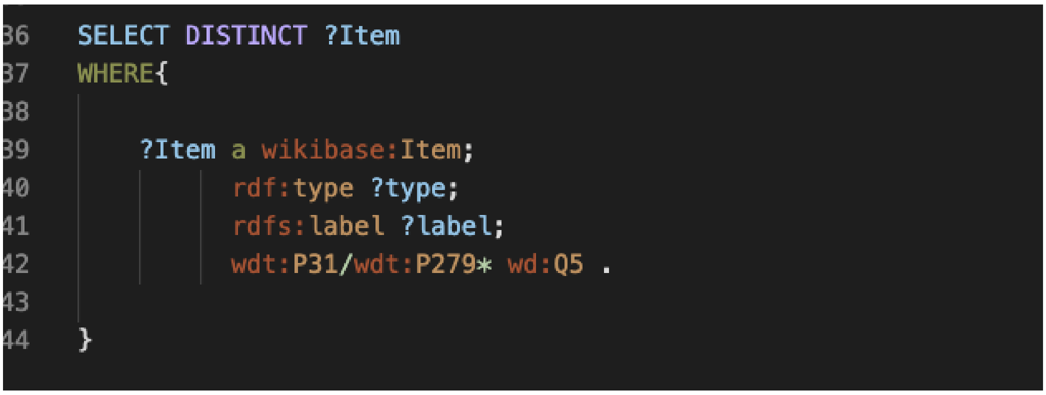
Triples	Subject	Predicate	object
1	Wd: Q76	statement/q76-66C211F7-A8BB-42F6-98C8-3451C112629C	Q4537781
2	statement/q76-66C211F7-A8BB-42F6-98C8-3451C112629C	singletonPropertyOf	Wdt: P31
3	statement/q76-66C211F7-A8BB-42F6-98C8-3451C112629C	Pq:	1985
4	statement/q76-66C211F7-A8BB-42F6-98C8-3451C112629C	Pq:	1985
5	statement/q76-66C211F7-A8BB-42F6-98C8-3451C112629C	Prov:wasDerivedFrom	reference /1474044d51cd60f38ca1b2260b3928b5b96aa88c
6	reference /1474044d51cd60f38ca1b2260b3928b5b96aa88c	Pr:P584	Ref_url
7	Wd: Q76	statement/q76-BDAB32FE-E9F9-4147-A1F6-354E9F27B487	wd: Q4537985
8	statement/q76-BDAB32FE-E9F9-4147-A1F6-354E9F27B487	Singleton Properties	Wdt: P31
9	statement/q76-BDAB32FE-E9F9-4147-A1F6-354E9F27B487	Prov:wasDerivedFrom	reference/ 9cdd4f1d064faebc44a10f bd408afa604f3b89f6
10	reference/ 9cdd4f1d064faebc44a10f bd408afa604f3b89f6	Pr:P143	Ref_url
11	Wd: Q76	statement/q76-A92C633A-CFE5-4F73-8D55-F4E8856FEC9C	wd: Q3483312
12	statement/q76-A92C633A-CFE5-4F73-8D55-F4E8856FEC9C	singletonPropertyOf	Wdt:P108
13	statement/q76-A92C633A-CFE5-4F73-8D55-F4E8856FEC9C	Pq:P580	1991
14	statement/q76-A92C633A-CFE5-4F73-8D55-F4E8856FEC9C	Pq:P582	1991

15	statement/q76-A92C633A-CFE5-4F73-8D55-F4E8856FEC9C	Prov:wasDerivedFrom	reference/fd60efb8e3d6b26135a2a84e 72ef62daf6517042
16	reference/fd60efb8e3d6b26135a2a84e 72ef62daf6517042	Pr:P854	Ref_url
17	Refnode	Pr: P1065	Ref_url
18	Wd: Q76	statement/Q76-E1E721F4-3A3F-46FA-BF4E-0B851C621318	Wd: Q131252
19	statement/Q76-E1E721F4-3A3F-46FA-BF4E-0B851C621318	singletonPropertyOf	Wdt:P108
20	Wd: Q76	statementq76-DEC62E52-F425-4754-870B-BFBB52E1B530	Wd: Q4537328
21	statementq76-DEC62E52-F425-4754-870B-BFBB52E1B530	singletonPropertyOf	Wdt:P108
22	statementq76-DEC62E52-F425-4754-870B-BFBB52E1B530	Pq:P582	1984
23	statementq76-DEC62E52-F425-4754-870B-BFBB52E1B530	Pq:P580	1983
24	statementq76-DEC62E52-F425-4754-870B-BFBB52E1B530	Pq: P2868	Q4809062

Appendix B

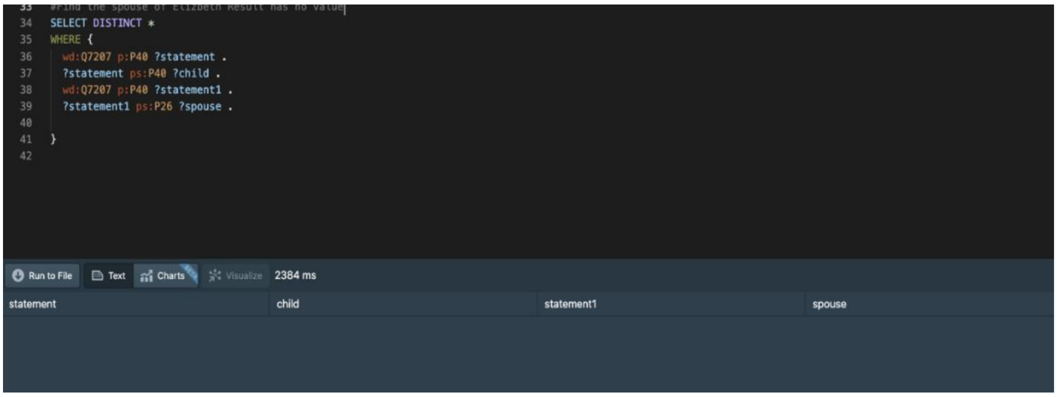
Experiment Results on Wikidata

1. SPARQL Query to find all the humans in Wikidata.



```
SELECT DISTINCT ?Item
WHERE{

    ?Item a wikibase:Item;
          rdf:type ?type;
          rdfs:label ?label;
          wdt:P31/wdt:P279* wd:Q5 .

}
```

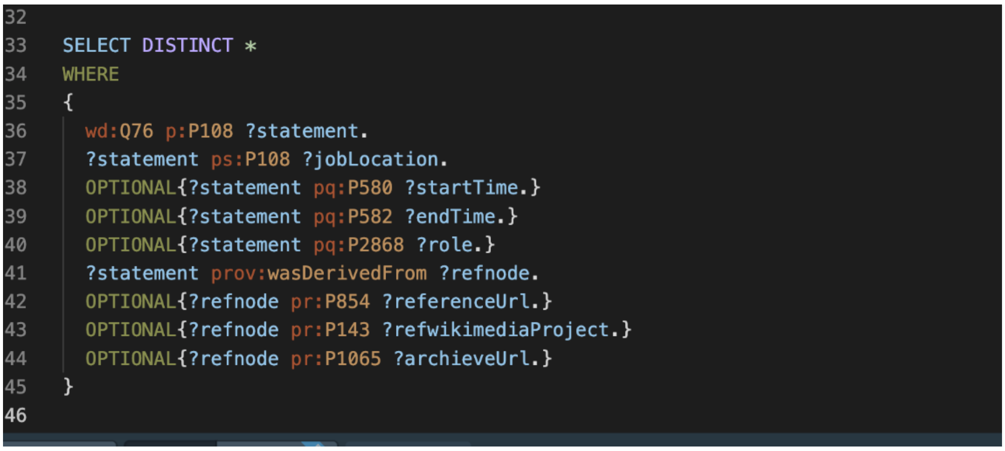
2. SPARQL Query to find the no values in RDF data model over Wikidata entity Elizbeth I England (Q7207) (Elizbeth I England, 2022)



```
#Find the spouse of Elizbeth Result has no value
SELECT DISTINCT *
WHERE {
  wd:Q7207 p:P40 ?statement .
  ?statement ps:P40 ?child .
  wd:Q7207 p:P40 ?statement1 .
  ?statement1 ps:P26 ?spouse .

}
```

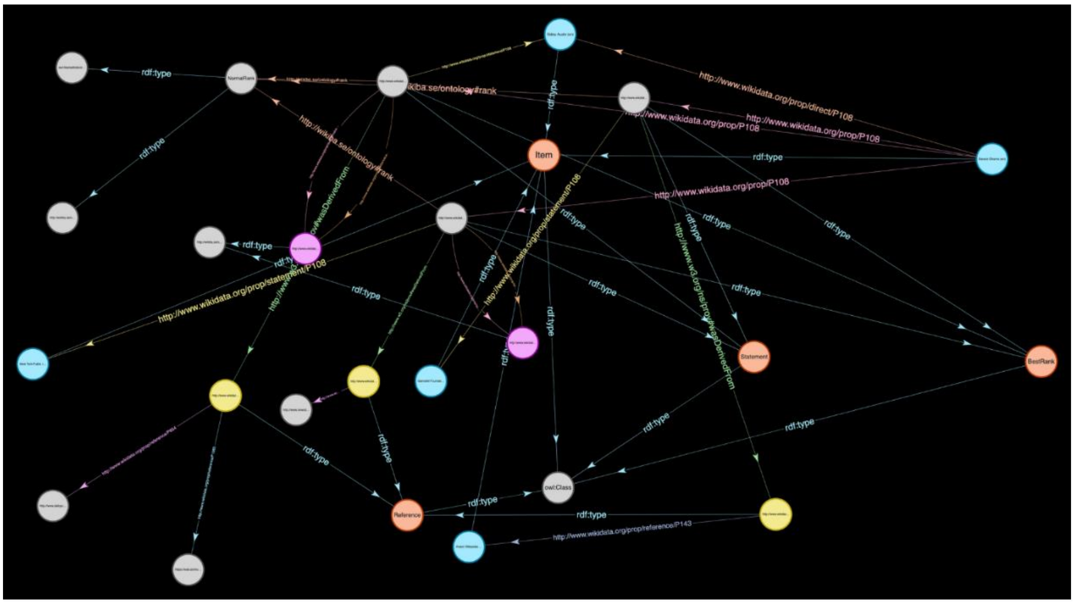
statement	child	statement1	spouse

3. SPARQL query to extract RDF subgraph of Wikidata entity Barack Obama Q76 of figure 22

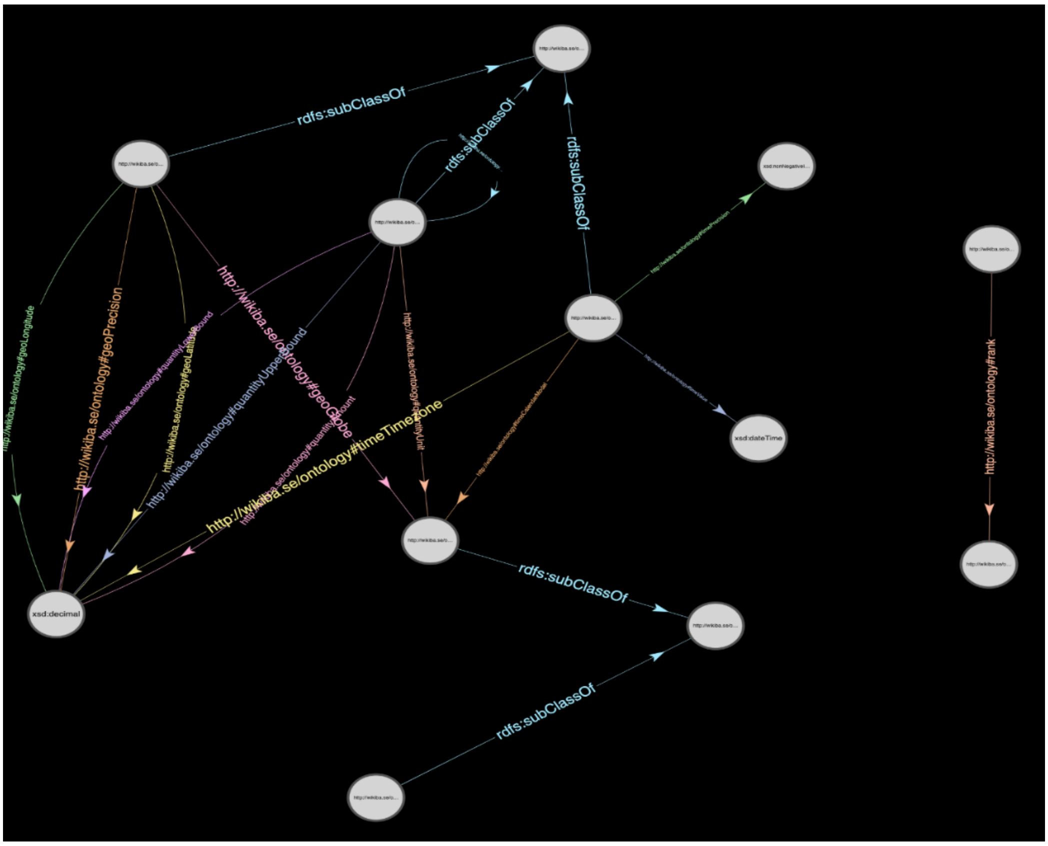


```

SELECT DISTINCT *
WHERE
{
  wd:Q76 p:P108 ?statement.
  ?statement ps:P108 ?jobLocation.
  OPTIONAL{?statement pq:P580 ?startTime.}
  OPTIONAL{?statement pq:P582 ?endTime.}
  OPTIONAL{?statement pq:P2868 ?role.}
  ?statement prov:wasDerivedFrom ?refnode.
  OPTIONAL{?refnode pr:P854 ?referenceUrl.}
  OPTIONAL{?refnode pr:P143 ?refwikimediaProject.}
  OPTIONAL{?refnode pr:P1065 ?archieveUrl.}
}

```

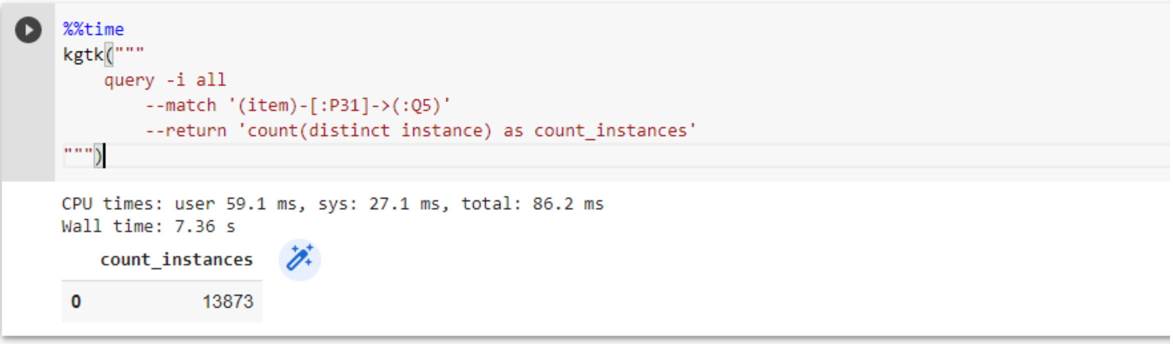
4. SPARQL query result for extracting subgraph of figure 22.


NormalRank
rdf:type
http://wikiba.se/ontology#rank
Item
http://www.wikidata.org/prop/direct/P108
http://www.wikidata.org/prop/P108
http://www.wikidata.org/prop/statement/P108
http://www.w3.org/ns/prov#wasDerivedFrom
Statement
BestRank
owl:Class
Reference
http://www.wikidata.org/prop/reference/P143

5. Wikidata ontology representation in RDF data model.


http://wikiba.se/o...
rdfs:subClassOf
rdfs:subClassOf
rdfs:subClassOf
http://wikiba.se/o...
http://wikiba.se/o...
xsd:nonNegativel...
http://wikiba.se/o...
http://wikiba.se/ontology#geoLongitude
http://wikiba.se/ontology#geoPrecision
http://wikiba.se/ontology#quantityLowerBound
http://wikiba.se/ontology#geoLatitude
http://wikiba.se/ontology#quantityUpperBound
http://wikiba.se/ontology#geoGlobe
http://wikiba.se/ontology#quantityAmount
http://wikiba.se/ontology#quantityUnit
http://wikiba.se/ontology#timeTimeZone
http://wikiba.se/ontology#timePrecision
http://wikiba.se/ontology#timeCalendarModel
http://wikiba.se/ontology#timeValue
http://wikiba.se/o...
xsd:dateTime
http://wikiba.se/ontology#rank
http://wikiba.se/o...
http://wikiba.se/o...
xsd:decimal
rdfs:subClassOf
http://wikiba.se/o...
rdfs:subClassOf
http://wikiba.se/o...

6. Kypher query to count the number of humans in Domain Graph data model.

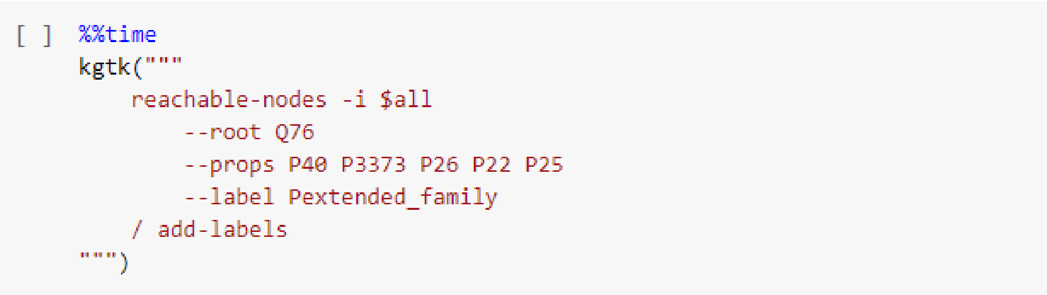


```
%%time
kgtk("""
    query -i all
        --match '(item)-[:P31]->(:Q5)'
        --return 'count(distinct instance) as count_instances'
""")
```

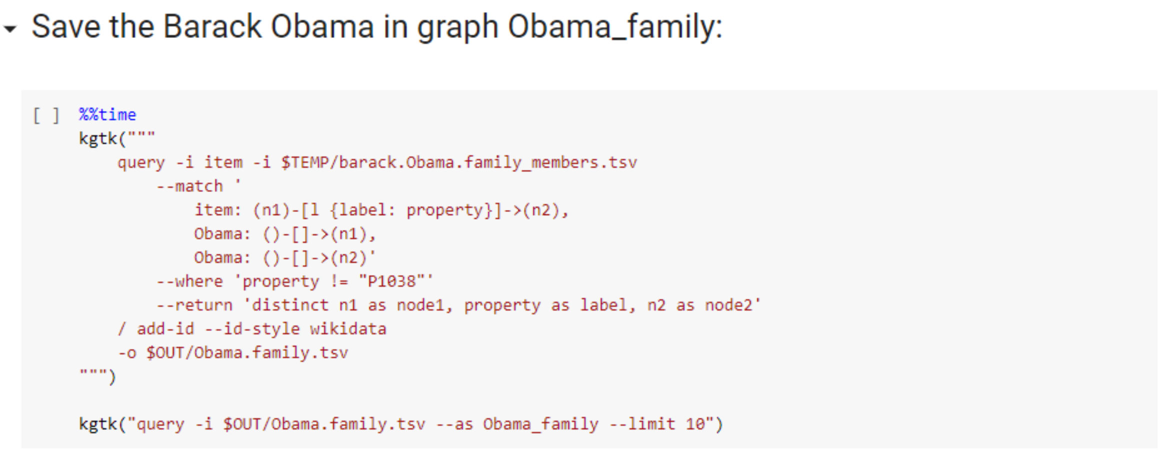
```
CPU times: user 59.1 ms, sys: 27.1 ms, total: 86.2 ms
Wall time: 7.36 s
```

	count_instances
0	13873

7. Create a network of Barack Obama in Domain Graph data model

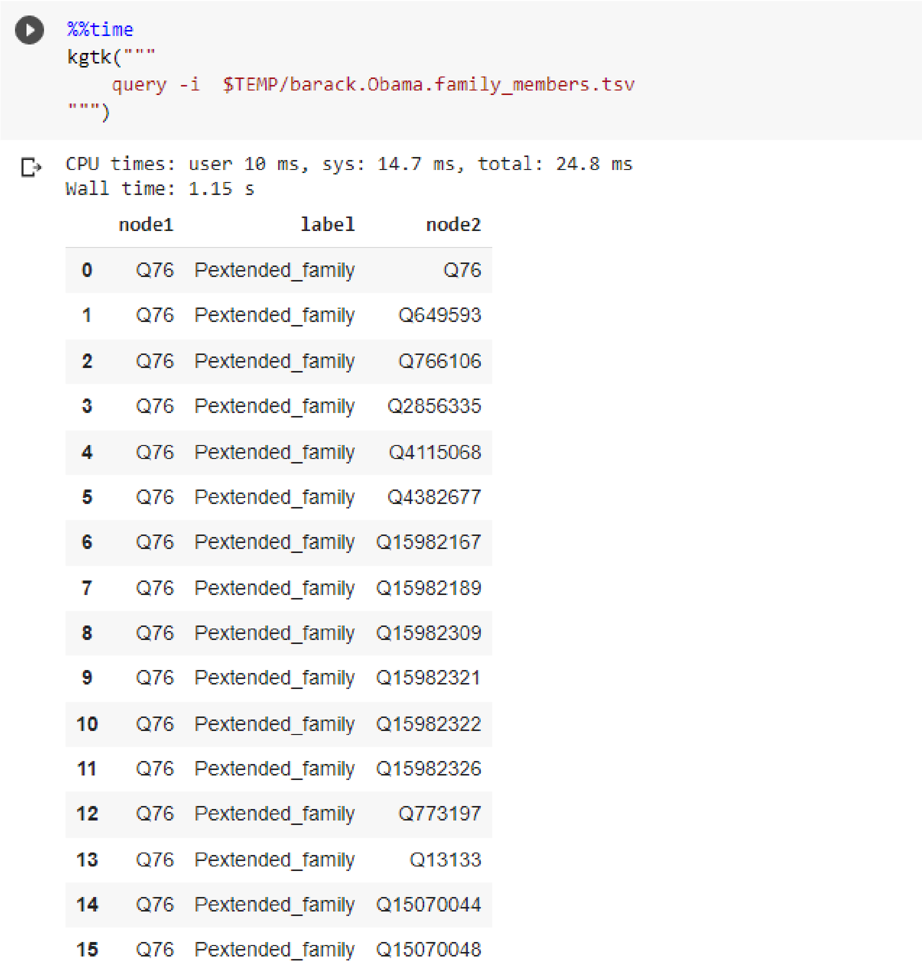


```
[ ] %%time
    kgtk("""
        reachable-nodes -i $all
            --root Q76
            --props P40 P3373 P26 P22 P25
            --label Pextended_family
        / add-labels
    """)
```



▾ Save the Barack Obama in graph Obama_family:

```
[ ] %%time
    kgtk("""
        query -i item -i $TEMP/barack.Obama.family_members.tsv
            --match '
                item: (n1)-[l {label: property}]->(n2),
                Obama: ()-[]->(n1),
                Obama: ()-[]->(n2)'
            --where 'property != "P1038"'
            --return 'distinct n1 as node1, property as label, n2 as node2'
        / add-id --id-style wikidata
        -o $OUT/Obama.family.tsv
    """)

    kgtk("query -i $OUT/Obama.family.tsv --as Obama_family --limit 10")
```

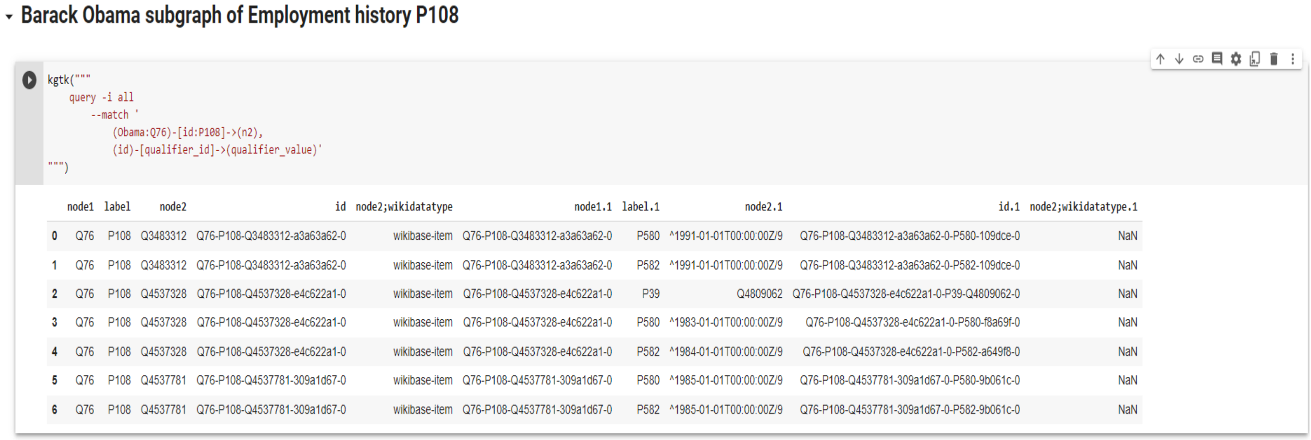


```
%%time
kgtk("""
    query -i  $TEMP/barack.Obama.family_members.tsv
""")
```

```
CPU times: user 10 ms, sys: 14.7 ms, total: 24.8 ms
Wall time: 1.15 s
```

	node1	label	node2
0	Q76	Pextended_family	Q76
1	Q76	Pextended_family	Q649593
2	Q76	Pextended_family	Q766106
3	Q76	Pextended_family	Q2856335
4	Q76	Pextended_family	Q4115068
5	Q76	Pextended_family	Q4382677
6	Q76	Pextended_family	Q15982167
7	Q76	Pextended_family	Q15982189
8	Q76	Pextended_family	Q15982309
9	Q76	Pextended_family	Q15982321
10	Q76	Pextended_family	Q15982322
11	Q76	Pextended_family	Q15982326
12	Q76	Pextended_family	Q773197
13	Q76	Pextended_family	Q13133
14	Q76	Pextended_family	Q15070044
15	Q76	Pextended_family	Q15070048

8. Extract the Barack Obama (Figure 22) subgraph

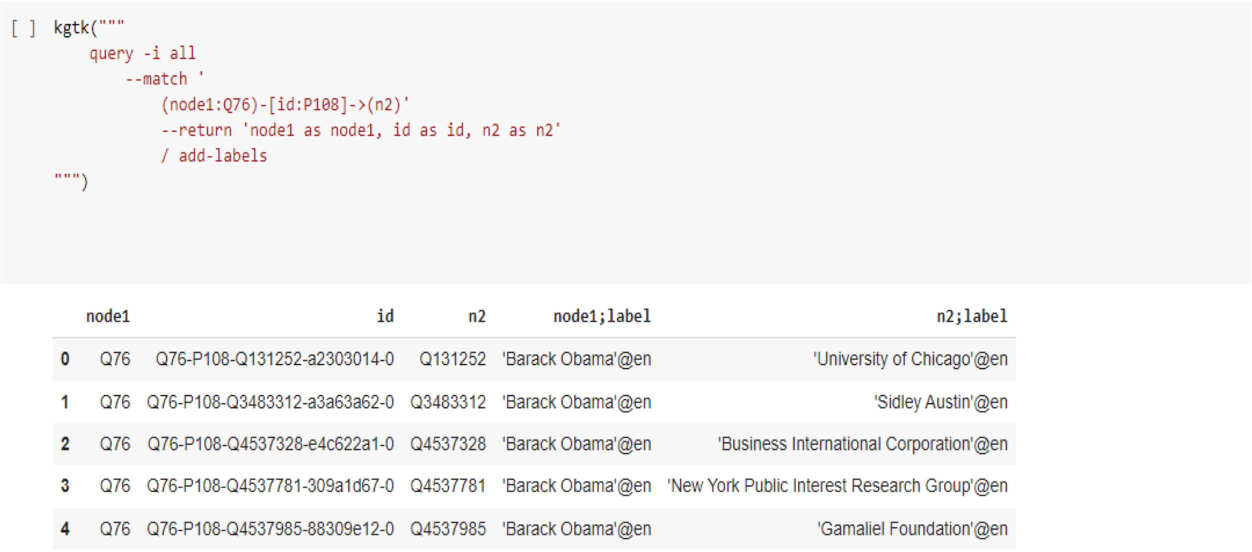


Barack Obama subgraph of Employment history P108

```
kgtk("""
    query -i all
        --match '
            (Obama:Q76)-[id:P108]->(n2),
            (id)-[qualifier_id]->(qualifier_value)'
""")
```

	node1	label	node2	id	node2;wikidatatype	node1.1	label.1	node2.1	id.1	node2;wikidatatype.1
0	Q76	P108	Q3483312	Q76-P108-Q3483312-a3a63a62-0	wikibase-item	Q76-P108-Q3483312-a3a63a62-0	P580	^1991-01-01T00:00:00Z/9	Q76-P108-Q3483312-a3a63a62-0-P580-109dce-0	NaN
1	Q76	P108	Q3483312	Q76-P108-Q3483312-a3a63a62-0	wikibase-item	Q76-P108-Q3483312-a3a63a62-0	P582	^1991-01-01T00:00:00Z/9	Q76-P108-Q3483312-a3a63a62-0-P582-109dce-0	NaN
2	Q76	P108	Q4537328	Q76-P108-Q4537328-e4c622a1-0	wikibase-item	Q76-P108-Q4537328-e4c622a1-0	P39	Q4809062	Q76-P108-Q4537328-e4c622a1-0-P39-Q4809062-0	NaN
3	Q76	P108	Q4537328	Q76-P108-Q4537328-e4c622a1-0	wikibase-item	Q76-P108-Q4537328-e4c622a1-0	P580	^1983-01-01T00:00:00Z/9	Q76-P108-Q4537328-e4c622a1-0-P580-f8a69f-0	NaN
4	Q76	P108	Q4537328	Q76-P108-Q4537328-e4c622a1-0	wikibase-item	Q76-P108-Q4537328-e4c622a1-0	P582	^1984-01-01T00:00:00Z/9	Q76-P108-Q4537328-e4c622a1-0-P582-a649f8-0	NaN
5	Q76	P108	Q4537781	Q76-P108-Q4537781-309a1d67-0	wikibase-item	Q76-P108-Q4537781-309a1d67-0	P580	^1985-01-01T00:00:00Z/9	Q76-P108-Q4537781-309a1d67-0-P580-9b061c-0	NaN
6	Q76	P108	Q4537781	Q76-P108-Q4537781-309a1d67-0	wikibase-item	Q76-P108-Q4537781-309a1d67-0	P582	^1985-01-01T00:00:00Z/9	Q76-P108-Q4537781-309a1d67-0-P582-9b061c-0	NaN

9. Extracting qualifiers of Barack Obama subgraph



```
[ ]  kgtk("""
         query -i all
             --match '
                 (node1:Q76)-[id:P108]->(n2)'
                 --return 'node1 as node1, id as id, n2 as n2'
                 / add-labels
     """)
```

	node1	id	n2	node1;label	n2;label
0	Q76	Q76-P108-Q131252-a2303014-0	Q131252	'Barack Obama'@en	'University of Chicago'@en
1	Q76	Q76-P108-Q3483312-a3a63a62-0	Q3483312	'Barack Obama'@en	'Sidley Austin'@en
2	Q76	Q76-P108-Q4537328-e4c622a1-0	Q4537328	'Barack Obama'@en	'Business International Corporation'@en
3	Q76	Q76-P108-Q4537781-309a1d67-0	Q4537781	'Barack Obama'@en	'New York Public Interest Research Group'@en
4	Q76	Q76-P108-Q4537985-88309e12-0	Q4537985	'Barack Obama'@en	'Gamaliel Foundation'@en