Storing in Semantic Web – Vertical Partitioning Approach

Ashraf F. A. Mahmoud
PhD. Student in
AL-Neelain university

Al Tyeb E. Abdelgabar
Associate Professor in
AL-Neelain university

ABSTRACT
This paper aims to provide storing solution for the Semantic Web which has some constraints in terms of efficiency and scalability because of the nature of resource description framework. RDF based on triple format (subject, property, object). The solution presented here is vertical partitioning approach in which two columns are used for each property. On the other hand, by having a look into property table technique in terms of concept, pros, and cons. Besides, a brief talk is provided about column-store approach and C-Store DBMS.

Keywords
semantic web, column- store, C-Store, resource description framework, property table technique, vertical partitioning approach.

1. INTRODUCTION
The Semantic Web is a web 3.0 technology makes machine thinks like humans. It is also, defined as a method of linking information between systems or entities that allow for rich relative data on the web [1].

The Semantic Web is a dream by the W3C that perspectives the Web as an enormous information mix issue. The objective is to free information from the applications that control them, with the goal that information can be effectively portrayed and traded. This is refined by supplementing normal dialect found on the Web with machine meaningful metadata in articulation frame (e.g., X is a man, X has-name "John", X has-age "35") and empowering portrayals of information ontologies so that information from various applications can be incorporated into philosophy mapping or applications can fit in with panel institutionalized ontologies. Executing Semantic Web information administration utilizing a segment store is not direct. The announcement organization of Semantic Web information is known as the Resource Description Framework (RDF) [2].

The W3 (World Wide Web Consortium) is a universal standard community; It evolves standards for the web to guarantee the long-term development on the web. To have the capacity to Link Data together W3 made the standard RDF which is the root of the Semantic Web making conceivable to make metadata (data about data) about resources on the web so that machines can understand them. What the resource is and how it interfaces to other resources [3].

RDF databases are collections of seeming "triples of knowledge", where each triple is of the form (subject, predicate, object) and models the binary relation predicate between the subject and the object. It can be seen as labeled directed graphs.

An effective RDF storage scheme should support a quick assessment of graph patterns and an extent to RDF databases containing millions of triples. The direct relational implementation, specifically a single Triples relation with three columns subject, predicate, and object that carries all RDF triples, looks not very favorable: The key problem with this approach is that the estimation of collected graph patterns usually requires a big amount of costly self-joins on this table.

A conceptually less complicated idea is to set up one table for every specific predicate in the data, which can be seen as complete vertical partitioning on the predicates. Every such predicate desk includes columns (subject, object) and incorporates all subject-object pairs connected thru the particular predicate. Data is then distributed across several littler tables and when the predicate is settled, joins don't include the entire group of triples. By physically sorting data at the subject column, subject-subject joins among two tables, a repeated operation, may be recognized in linear time by combining their subject columns[4].

2. RELATED WORKS
2.1 Column-oriented database systems (column-stores)
There is an enormous amount of excitation and recent work on column-orientated database systems (column-stores). These database structures were proven to perform greater than an order of volume better than conventional row-orientated database structures (row-stores) on analytical workloads like the ones discovered in data warehouses, decision support, and business intelligence applications [5].

Data is stored in a chain of "triples" enclosing a subject, property, and object. Storing these triples in a column-store with a three column table (one column for a subject, one for a property, and one for an object) does not minimize a lot of joins that are requisite in the execution of most queries. So,

a data rearranged in which the single three column tables is rearranged into a vast, sprinkled table, comprising one column for each unique property. This large table is then vertically partitioned to minimize the scattering. Column-stores get better by more than an order of size more than row-stores [2].

Properties are stored on disk in discrete files, in blocks of 64 KB. Each property includes two columns such as the vertically partitioned store above. Each property has

a clustered B+ tree on the subject, and single-valued, fewer originality properties have a bitmap index on the object[6].

2.2 C-Store
Is a design of a column store that encloses a number of fresh characteristics related to current systems. C-Store physically stores a set of columns, each sorted on some attribute(s).

C-Store is a column-oriented DBMS that is architected to decrease the number of disk accesses per query. The creative characteristics of C-Store contain a column-oriented optimizer and executor, with different primitives than in a row-oriented system[7].

3. RESOURCE DESCRIPTION FRAMEWORK

Resource description framework is a standard for describing resources on the semantic web[8]. It represents data as statements approximately the source of which is the use of graph connecting resources nodes and their properties values with categorized arcs representing properties. Syntactically, this graph may be represented by using XML syntax (RDF/XML). This is normally the format for RDF data exchange, however, structurally; the graph may be parsed into a chain of triples, each representing a statement of the form (subject, property, object). Those triples can then be stored in a relational database with a three-column schema [6].

The property is typically a verb denotes the relationship that exists between the subject and the object. Each element of a triple is identified by a URI (Universal Resource Identifiers). It utilizes web identifiers (URIs) to distinguish resources, describes resources with properties and property values which a property is a Resource that has a name (such as employee) and a property value is the value of a property (such as "Ahmed Omer" or another resource, for example, http://www.Ashrafpage.com) [8].

For example, to represent the fact that Ahmed Omer, Ali Mohammed, John Krees, James Kan are employees in a Company called “Blue Nile”, fifteen triples will be used:

person1 isNamed “Ahmed Omer”

person2 isNamed “Ali Mohammed”

person3 isNamed “John Krees”

person4 isNamed “James Kan”

person1 hasAdress “KSA”

person3 hasAddress “Canada”

person4 hasAddress “London”

person1 hasPhone “0147895478”

person2 hasPhone “0145258147”

person4 hasPhone “0547896214”

company hasEmployee person1

company hasEmployee person2

company hasEmployee person3

company hasEmployee person4

company isCalled “Blue Nile

Table 1: RDF triples example

Subject	Property	Object
NO1	isNamed	“Ahmed Omer”
NO2	isNamed	“Ali Mohammed”
NO3	isNamed	“John Krees”
NO4	isNamed	“James Kan”
NO5	hasEmployee	Person1
NO6	hasEmployee	Person2
NO7	hasEmployee	Person3
NO8	hasEmployee	Person4
NO9	has Address	Person1
NO10	has Address	Person3
NO11	has Address	Person4
NO12	hasPhone	Person1
NO13	hasPhone	Person2
NO14	hasPhone	Person4
NO15	isCalled	“Blue Nile”

The above table represents triples store concept in the semantic web. It consists three columns as such: the subject,

a property and the object. The subject column contains persons’ numbers. Property column is a description of the subject. Object column includes the values of the properties.

4. PROPERTY TABLE TECHNIQUE

Property table is used to increase the efficiency of retrieving information from triple-stores. There are two types of property tables. The primary type, which called a clustered property table, contains clusters of properties that have a tendency to be described together. The second one shows the sort of property table, termed as a property class table, which exploits the type property of subjects to cluster comparable groups of subjects together in the same table [9].

Table 2. Clustered Property table

Subject	isNamed	hasEmployee	isCalled	hasAddress	hasPhone
NO1	“Ahmed Omer”	person1	“Blue Nile”	“KSA”	0147895478
NO2	“Ali Mohammed”	person2	“Blue Nile”	NULL	0145258147
NO3	“John Krees”	Person3	“Blue Nile”	“Canada”	NULL
NO4	“James Kan”	Person4	“Blue Nile”	“London”	0547896214

The above table explains how data stored using clustered property table technique. Basically, it consists of the subject column and (N) property columns. The subject column contains the employee's number and property columns contain the values or data that belong to each employee.

Table 3. Property class table (hasAddress class)

Subject	isNamed	hasEmployee	isCalled	hasPhone
NO1	“Ahmed Omer”	person1	“Blue Nile”	0147895478
NO3	“John Krees”	Person3	“Blue Nile”	0145258147
NO4	“James Kan”	Person4	“Blue Nile”	0547896214

Table 3 uses property class table technique. It contains the data of all employees that have the property "hasAddress"

without including the property itself (i.e. hasAdress property omitted from the table). Employee number two (NO2) (subject) "Ali Mohammed" (object) has no address (property) that is why it did not show up in the table.

Table 4. Property class table (hasPhone class)

Subject	isNamed	hasEmployee	isCalled	hasAddress
NO1	"Ahmed Omer"	person1	"Blue Nile"	"KSA"
NO2	"Ali Mohammed"	person2	"Blue Nile"	"Canada"
NO4	"James Kan"	Person4	"Blue Nile"	"London"

Table 4 also uses property class table technique, but this time with different property type called hasPhone. The same idea applied in the table above which presents all data of all employees with the property "hasPhone". Employee number three – NO3 (subject) "John Krees" (object) has no phone number (property) that is why it did not show up in the table

4.1 Advantages of property table technique

There are multiple benefits of using property table technique: Firstly, it is very general (almost any type of data can be represented in this format it is easy to shred both relational and XML databases into RDF triples) and it is easy to construct tools that manipulate RDF. Secondly, it opens up the possibility for attribute typing. Rather than storing objects of many different types in a single column in a triple-store [10].

4.2 Disadvantages property table technique

This approach has several problems: the first one is, NULLs as a matter of fact, not all properties can be defined for all subjects within the subject cluster so huge tables could have NULLs. The space overhead of those NULLs can potentially dominate the space of the data itself. The other disadvantage is multi-valued attributes: Many-to-many relationships are somewhat awkward to explicit in a clear illustration (such as projects have multiple employees and employees have multiple projects). The last one is the clustering algorithm must typically be utilized to discover gatherings of properties that tend to be defined together or classes of subjects with similar property definitions. Furthermore, re-clustering might become necessary as the data characteristics change over time [10].

5. VERTICAL PARTITIONING APPROACH

To address the above limitations, the researcher will use vertical partitioning approach which creates a two-column table for each unique property in the RDF dataset where the first column in close subjects that define the property and the second column contains the object values for those subjects [11].

Every table is sorted by subject so that specific subjects can be situated fastly, and that quick combination joins can be used to rebuild information about multiple properties for subgroups of subjects. The value column for each table can also be optionally indexed [10]. There is a clustered B+ tree index on a subject and an unclustered B+ tree index on the object[6].

Table 5. Vertical partitioning (isNamed)

Subject	isNamed
NO1	"Ahmed Omer"
NO2	"Ali Mohammed"
NO3	"John Krees"
NO4	"James Kan"

Table 5 uses vertical partitioning technique. It contains two columns, the subject which includes all employees' numbers and the property "isNamed" which includes all the employees' names.

Table 6. Vertical partitioning (hasEmployee)

Subject	hasEmployee
NO1	person1
NO2	person2
NO3	person3
NO4	Person4

Table 6 uses vertical partitioning technique. It contains two columns, the subject which includes all employees' numbers and the property 'hasEmployee" which includes all the employees' description.

Table 7. Vertical partitioning (isCalled)

Subject	isCalled
NO1	"Blue Nile"
NO2	"Blue Nile"
NO3	"Blue Nile"
NO4	"Blue Nile"

Table 7 uses vertical partitioning technique. It contains two columns the first one is the subject which includes all employees' numbers and the second one is the property "isCalled" which includes the company name.

Table 8. Vertical partitioning (hasAddress)

Subject	has Address
NO1	"KSA"
NO3	"Canada"
NO4	"London"

Table 8 uses vertical partitioning technique. It contains two columns the first one is the subject which includes all employees' numbers which has address and the second one is

property "hasAddress" which includes the employees' addresses.

Table 9. Vertical partitioning (hasPhone)

Subject	hasPhone
NO1	0147895478
NO2	0145258147
NO4	0547896214

Table 9 uses vertical partitioning technique. It contains two columns the first one is the subject which includes all employees' numbers which has phone number and the second one is property "hasPhone" which includes the employees' phone numbers

5.1 Advantages of vertical partitioning approach

This approach has multiple advantages which include the support for multi-valued attribute - multi-valued subjects are thus represented as multiple rows in the table with the same subject and different values (see table 10), support for heterogeneous records - subjects (for e.g. NO15) that do not define a particular property (hasAddress/hasPhone) are simply omitted from the table for that property (see table 8 and 9), only those properties accessed by a query need to be read. I/O costs will be well decreased, no clustering algorithms are needed. The algorithm for creating tables in the vertical partitioning approach is straightforward and needs no changes over time and fewer unions (relative to the property-class schema approach). All data for a specific property is situated in the same table, union clauses in queries are less common [10].

Table 10. Vertical partitioning multi-valued

Subject	hasEmployee
NO15	"Ahmed Omer"
NO15	"Ali Mohammed"
NO15	"John Krees"
NO15	"James Kan"

The previous table illustrates how multi-valued attribute stored using vertical partitioning technique. It consists of two columns, firstly, the company id number, and secondly, the employees' names. It presents relation between company and employees (one to many).

5.2 Disadvantages of vertical partitioning approach

This approach has some disadvantages summarized as follows: increased number of joins when a query accesses several properties, multiple two column vertical partitions have to be merged, and inserts can be slow for vertically

partitioned schemes since inserted statements about the same subject end up needing to access multiple different partitions [9].

6. CONCLUSION

Semantic Web is an important step toward web evaluation; it will improve search results for all users' type (people and developers) due its understanding of user's intention. Semantic Web differs considerably from traditional web in the storage model; it applies RDF to store data. Search engine needs to work differently with Semantic Web in order to retrieve data because it deals with SWDs instead of HTML pages. This paper presents different types of storing techniques as such triples store, property table and vertical partitioning approach. Vertical partitioning technique handled with multivalued attribute when it is compared with property table technique. Also, it indicates better performance contrasted with a triple store and property tables especially when it is used with c-store.

Future work will focus on solving the limitations of vertical partition technique toward joins and insertion.

7. REFERENCES

[1] http://www.linkeddatatools.com/semantic-web-basics

Semantic web definition Day: 13-08-201 05:22:25 PM.

[2] Abadi, Daniel J. Query execution in column-oriented database systems. Diss. Massachusetts Institute of Technology, 2008.

[3] Gylfason, B. P. "The future of the web: The semantic web." (2010).

[4] Sheth, Amit P., et al. "The Semantic Web-ISWC 2008." (2008).

[5] Samuel R. Madden, Abadi, Daniel J., and Nabil Hachem. "Column-stores vs. row-stores: How different are they really?." Proceedings of the 2008 ACM SIGMOD international conference on Management of data. ACM, 2008.

[6] Adam Marcus, et al. "Scalable semantic web data management using vertical partitioning." Proceedings of the 33rd international conference on Very large data bases. VLDB Endowment, 2007.

[7] Stonebraker, Mike, et al. "C-store: a column-oriented DBMS." Proceedings of the 31st international conference on Very large data bases. VLDB Endowment, 2005.

[8] http://www.w3schools.comResource Description Framework definition Day: 15-08-2016 09:26:15 AM. Resource Description Framework triples Day: 16-08-2016 12:20:10 PM.

[9] Weiss, Cathrin, Panagiotis Karras, and Abraham Bernstein. "Hexastore: sextuple indexing for semantic web data management." Proceedings of the VLDB Endowment 1.1 (2008): 1008-1019.

[10] Madden, S. R., et al. "SW-Store: a vertically partitioned DBMS for Semantic Web data management." The VLDB Journal—The International Journal on Very Large Data Bases 18.2 (2009): 385-406.

[11] Curé, Olivier, and Guillaume Blin. RDF Database Systems: Triples Storage and SPARQL Query Processing. Morgan Kaufmann, 2014.