Volume 4, Issue 12, December 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Time Based Query Comparison of Relational Database and Resource Description Framework(RDF)

Mradula Singh
Computer Science and Engineering
Shri Vaishnav Institute of Technology and Science,
Indore (M.P), India

Miss. Pallavi Jain
Assistant Prof., Computer Science and Engineering
Shri Vaishnav Institute of Technology and Science,
Indore (M.P), India

Abstract — Many researchers have enlightened us with their new methods to map data between Relational Database (RDB) and different serialization formats of Resource Description Frameworks (RDF). Relational Database uses SQL for its query processing and RDF uses SPARQL. SQL and SPARQL are very similar to each other, the way it is written, its keywords etc. Seeing this similarity, in this paper we have performed a time based query comparison between RDF and RDB.

Keywords—Relational Database, Resource Description Framework (RDF), SPARQL, SQL;

I. INTRODUCTION

As we all are already familiar with Relational Database theory which was given to us by Edgar F. Codd in 1969. There are many tools which we can use for storing relational databases like DB2, Oracle, Access etc. All the data in RDB is stored in tabular form. With columns containing the attributes and rows containing values of those attributes. We use SQL for querying our relational database.

On the other hand RDF is relatively new its foundation was established in 1997. And since then it is progressing. W3C has shown its keen interest towards the development of RDF. W3C has also released a series of documents so that it becomes easier for everyone to understand the concepts of RDF. There are many languages from which we can query RDF Store but SPARQL is similar to SQL and it's easy to understand it. We will be using it for querying RDF Store.

II. RELATIONAL DATABASE

Relational database is DBMS in which all the data is in stored in tabular form. RDBMS was introduced by E.F Codd. It's based on relational data model. It is understandable and used for the storage of information like account records, personal data etc.

As we all know that relational database stores information in tabular form and those tables are known as relations. Relations contain rows which are called tuples and columns are known as attributes. Tuples are used to represent object (physical objects or concepts) or information about the object.

Some relations gather information from various relations still they act like a single relation. All relations are queried by structural query language.

III. RELATIONAL DATA MODEL

Relational database is based on relational data model in which data is arranged in tabular form and columns of the table are called attributes. Name to table is called relation and rows are referred as tuples.

Collection of tables is used to represent both data and relationships among those data. Information from a relational database is queried with the help of SQL (Structured Query Language).

If we want to create a relational database we first have to define datatypes of all the attributes the approximate size of those datatypes, we have to create an empty structure of table with columns telling the properties/attributes. Then we insert information in form of tuples.

First Name	Last Name	Address	Phone	Book Title	Due Date
Bob	Smith	123 Main St.	555-1212	Don Quixote	7-14-09
Alicia	Petersohn	136 Oak St.	555-1234	Three Men in a Boat	7-16-09
Bob	Smith	123 Main St.	555-1212	Things Fall Apart	8-15-09
Bob	Smith	123 Main St.	555-1212	Anna Karenina	8-15-09
Zayn	Murray	248 Pine Dr.	555-1248	Heidi	8-17-09
Bob	Smith	123 Main St.	555-1212	The Old Man and the Sea	9-10-09

Fig. 1.Tabular view of RDB.

IV. SQL (STRUCTURED QUERY LANGUAGE)

SQL is a language which provides interface to RDBMS (Relational Database Management System), to manage its data. It was developed by IBM in 1970 and became ANSI standard in 1986. A year later it became an ISO standard. Originally it was based on relational algebra and relational calculus. Since then many new features were added to SQL. Data is defined using the DDL (Data Definition Language) commands and its manipulated using DML (Data Manipulation Language) commands.

A SQL query is written using keywords SELECT, FROM, WHERE in which we define what data we want from which table/tables and in "where" we give conditions. E.g. of a SQL query is shown below:

```
SELECT fname, lname, dob_inc "birthdate", occup
FROM cust_mstr
WHERE fname like 'ch%';
```

To end SQL query we use semicolon (";"). In the above query we want the first name, last name, date of birth and occupation from customer master table and in where clause we gave the condition. The database will search the suitable data which satisfies all the conditions and will give the result accordingly.

In SQL we can perform aggregation, match patterns, can order data in ascending or descending order etc. there are many features using which we can get desired output. We can also put constraints on the data.

V. RESOURCE DESCRIPTION FRAMEWORK (R.D.F)

RDF is a used for describing resources. It makes statements about resources. The statements are in subject – predicate - object expression which are referred as triples and stored in triple store. Through directed graph we represent these triples. In which predicate represents relationship between subject and object.

E.g. "*Bob DuCharme is author of Learning SPARQL*." RDF triple of the above statement will be: the subject will be "Bob DuCharme", predicate will be "is author of", and object will be "Learning SPARQL".

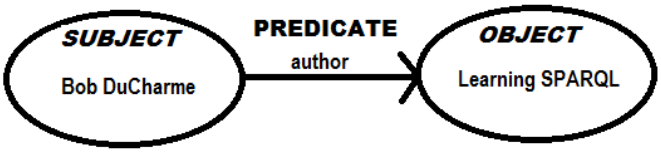

Fig. 2. Graph representation of RDF.

We can serialize RDF with several serialization formats like XML, N3 and Turtle etc. Hence which change in format the encoding of triples will also change. In this paper we will see Ntriple format.

To understand RDF and its serialization format we need to see the model on which RDF is based on.

VI. RESOURCE DESCRIPTION FRAMEWORK DATA MODEL (RDF MODEL)

RDF data model is used to represent RDF expressions as directed graphs as they are easy to understand. Directed graphs are used because:

1. Graphs are easy to read and understand. We can easily find subject predicate and object from the graph.
2. RDF data models that can be represented in RDF graphs, but not in RDF/XML format.

RDF directed graph are connected by arcs in **subject-predicate-object** pattern. Subject predicate and objects are written in URI pattern. A URI provides unique identification to node.

In Ntriple format we will write URI's inside "< >" angular brackets. Example of a single triple in Ntriple format is shown below:

<http://resource.com/name/Bob_DuCharme><http://property.com/author><http://value.com/Learning_SPARQL>.

A triple always ends with a full stop. The above triple explains Ntriple for fig. 2. These Ntriple files are saved with .ntriple extension.

VII. SPARQL (SPARQL PROTOCOL AND RDF QUERY LANGUAGE.)

SPARQL allows user to query RDF graphs by writing query and the data which satisfy query are returned. SPARQL is very similar to SQL. SPARQL also make query using SELECT, WHERE, FROM keywords.

```
PREFIX ab: <http://learningsparql.com/ns/addressbook#>
SELECT  ?craigEmail
WHERE
{
 ab:craig  ab:email  ?craigEmail. }
```

or

```
SELECT  ?craigEmail
WHERE
```



```
{
<http://learningsparql.com/ns/addressbook#craig>
<http://learningsparql.com/ns/addressbook#email>
?craigEmail .
}
```

Above is an example of a SPARQL query[8]. In this query we are asking for the email id of Craig. We can write SPARQL query in either ways. We use PREFIX so that we don't have to write the URI again and again. Both the queries will return same information.

WHERE conditions are always written between curly brackets "{ }". The information in is in triple format. Information which we want to search are written with a question mark in front of them like ?craigEmail.

If any information belongs to a particular file we mention the file name with FROM keyword. FROM keyword is placed after SELECT.

```
SELECT  ?craigEmail  FROM <ex002.ttl>
WHERE
{
<http://learningsparql.com/ns/addressbook#craig>
<http://learningsparql.com/ns/addressbook#email>
?craigEmail .
}
```

This was the syntax in which SPARQL is written and the way it is similar to SQL. Now we will query same data set with both the query languages to find the difference in the execution timing.

There is a data set of National Stock Exchange of India Ltd on which queries are performed and timing is recorded.

VIII. SPARQL (SPARQL PROTOCOL AND RDF QUERY LANGUAGE.) VS SQL (STRUCTURED QUERY LANGUAGE)

Here are few queries which gave same result but there is difference in execution time. SQL queries are performed on Oracle 10g and SPARQL queries on Gruff 5.5.0. For SPARQL queries mentioned below we have some prefixes:

```
PREFIX res: <http://Sample.Thesis/resource/>
       prop: <http://Sample.Thesis/property/>
       val: <http://Sample.Thesis/value/>
```

1. SQL:

Select * from APH where ID = 'APH10';	0.05sec

SPARQL:

select * where { res:APH10 prop:OPEN_PRICE ?OPEN_PRICE . res:APH10 prop:CLOSE_PRICE ?CLOSE_PRICE. res:APH10 prop:HIGH_PRICE ?HIGH_PRICE. res:APH10 prop:LOW_PRICE ?LOW_PRICE. res:APH10 prop:DATE ?DATE. res:APH10 prop:LAST_TRADE_PRICE ?LAST_TRADE_PRICE. }	0.004sec

2. SQL:

select max(open_price) "maxOPEN_PRICE" from techm; Ans: 2492	1.17sec

SPARQL:

select ?OPEN_PRICE from <http://Sample.Thesis/TECHM> where {?x prop:OPEN_PRICE ?OPEN_PRICE.} ORDER BY DESC(?OPEN_PRICE) LIMIT1 Ans :2492	0.046sec

3. SQL:

select MIN (open_price) "MINOPEN_PRICE" from TCS; Ans: 2525.4	0.05sec

SPARQL:

select ?OPEN_PRICE from <http://Sample.Thesis/TCS> where { ?x prop:OPEN_PRICE ?OPEN_PRICE.} ORDER BY (?OPEN_PRICE) LIMIT1 Ans: 2525.4	0.005sec

4. SQL:

select CLOSE_PRICE from ITC order by CLOSE_PRICE;	0.04sec

SPARQL:

select * from <http://Sample.Thesis/ITC> where {?x prop:CLOSE_PRICE ?CLOSE_PRICE.} ORDER BY (?CLOSE_PRICE)	0.004sec

5. SQL:

select ID, OPEN_PRICE from HUL union select ID,OPEN_PRICE from APH union select ID,OPEN_PRICE from ITC;	0.03sec

SPARQL:

select * from <http://Sample.Thesis/HUL> from <http://Sample.Thesis/APH> from <http://Sample.Thesis/ITC> where { ?x prop:OPEN_PRICE ?OPEN_PRICE.}	0.006sec

In the tool named Gruff 5.5.0 AllegroGraph 3.3 we can see both tabular result of SPARQL query and graphical result too. We can compare graphs. And it has many inbuilt features which reduce our efforts. We don't have to write queries for additions of triples, renaming, deletion of triples etc.

IX. CONCLUSION

These were some examples which clearly show the difference in execution time of both the query language. SPARQL give output that can be changed to graphs and can be compared easily. On the other hand SQL give output which is in tabular form. SQL is used for highly structured and regular data and SPARQL is good for data that changes frequently. It's easy to add node to the graph or remove nodes.

ACKNOWLEDGMENT

I would like to thank whole Computer Science Department of Shri Vaishnav Institute of Technology and Science Indore (M.P) for all of its support. I would also like to thank my guide Miss Pallavi Jain for her efforts and guidance.

REFERENCES

[1] Shelley Powers, "Practical RDF", First edition 2003, pp.1-20.
[2] Raghu Ramakrishnan, Johannes Gehrke "Database Management System" Second Edition, pp. 51-54
[3] en.wikipedia.org/wiki/SPARQL.
[4] http://www.w3.org/TR/rdf-concepts/
[5] Hugh Darwen , "A Introdction to Relational Database Theory,"2010 in press.
[6] http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/
[7] en.wikipedia.org/wiki/Relational_database.
[8] Bob DuCharme "Learning Sparql" First Edition 2011, pp 1-5.
[9] S. Sumathi, S. Esakkirajan, "Fundamentals of Relational Database Management System," 2007

