International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 2 Issue: 7 1879 – 1882

Comparative Study Of Relational Database Model and Resource Description Framework(RDF) Model

Mradula singh
Computer science and engineering
Shri Vaishnav Institute of Technology and Science Indore
(M.P), India
mradula4singh@gmail.com

Miss. Pallavi Jain
Lect., Computer science and engineering
Shri Vaishnav Institute of Technology and Science Indore
(M.P), India

Abstract **-** In this modern era, World Wide Web Consortium (W3C) found a new way to store, represent and manage data. Resource Description Framework (RDF) is that progressing path. From a very long time we all are dependent upon relational database for effective storage and management of our data. So to find out which one is better RDF or RDB? Basic Model on which Relational Database and Resource Description Framework work is explained in this paper.

Keywords-Relational *Database, Resource Description Framework (RDF), Relational Data Model, RDF data model;*

*****

I. INTRODUCTION

Relational Database theory was given to us by Edgar F. Codd in 1969. Since then relational database has become an important part of our data storage and management. All the data in relational database was stored in tabular form. Codd gave some rules according to which databases should be created. This leads to the development of various commercial relational database management systems products like DB2, Oracle etc.

Resource description framework's foundation was established by Ora Lissila and Ralph Swick in 1997. Since then it kept maturing. Many new documents were released by W3C to support and describe RDF, make it more understandable.

II. RELATIONAL DATABASE

Relational database is DBMS whose essential data structure has relational tables. RDBMS is based on relational model (or relational data model) which was introduced by E.F Codd. Currently relational data model is used in many of the popular databases.

It's easier to understand and it's good for the storage of information like financial records, personal data etc. As it is already been mentioned that relational database stores information in tabular form so the tables are referred as relations. We can also say that relations are set of tuples having same attributes. Tuples represent object (mainly physical objects or concepts) or information about the object. Derived relations are computed by applying relational operations to other relation. Even though derived relations gather information from various relations still they act like a single relation.

To understand relation database firstly we need to understand the model on which this database is based on, the relational data model.

III. RELATIONAL DATA MODEL

Relational data model is represented with tables, tuples, attributes etc. Rows in the tables are called tuples and columns are called attributes. Description of a data in terms of data models is called a schema. In case of the relational data model schema specifies relations names, attributes names their types etc. Tables are logical structures and 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). Terms which are used by model, users and programmers:-

User	Model	Programmer
Row	Tuple	Record
Column	Attribute	Field
Table	Relation	File

Relational data model is a combination of three components namely:

A. STRUCTURAL PART

The part of database which is defined as a collection of relations comes under structural part.

B. INTEGRITY PART

Keys and constraints maintain database integrity in a relational model.

C. MANIPULATIVE PART

Manipulation in data of a database is done by using tools like relational algebra and relational calculus.

IV. KEY FEATURES OF RELATIONAL DATABASE MODEL

- Tuples: Rows in the table are called tuples.

- Attribute: Columns in the tables are called attributes.
- Data values: Intersection of row with column will contain data values.
- In relational model we can have rows and columns in any order.
- In a relation no two rows can be exactly same. All rows are distinct.
- Every relation must have a key which can uniquely define that relation. Key can also be a set of attributes.
- There is a set of possible values for each and every column which is called its domain.
- Domain: It's a set of valid values for an attribute.
- No. of attributes in a relation tells degree of that relation.
- No. of tuples tells cardinality of that relation

A. STRUCTURAL PART

TABLE AND RELATION:

What's the difference between table and relation? For any table to be relation there are some rules which should be followed:

- Single value (atomic values) should be contained at the intersection of rows with column.
- In a column all entries should be of same type.
- There is unique name for each column.
- No two rows can be identical in a relation.

STUDENT

Roll. No	First name	Last name	Login	Age	G.P.A
0601	David	John	john.david01@cs	20	4.6
0602	Smith	Sen	sen.smith02@math	19	5.6
0603	Rahul	Seth	Seth.rahul03@ee	18	7.2

Tabular representation of relational data model

In Fig.1, STUDENT (roll no, first name, last name, login, age, G.P.A) are the attributes of the relation. STUDENT is the name of the relation. The data types, primary key, name of attributes etc were defined at the time of its creation.

The degree of relation (no. of column in the relation) is 6.

And cardinality of relation (no. of rows in the relation) is 3.

B. INTEGRITY PART

KEYS:

Keys are attributes or group of attributes which are used for uniquely identifying a row in a relation.

Keys are broadly classified into

- Super key
- Primary key
- Candidate key
- Foreign key

i. *SUPER KEY:*

Super keys are mainly constraints on relations. It doesn't allow two entities to have same values for that particular subset of attributes.

ii. *CANDIDATE KEY:*

Candidate key is minimal super key. In a relational schema candidate key is that minimal set of attributes which uniquely identify tuples of agreeing relation.

iii. *PRIMARY KEY:*

Primary key is specified or designated candidate key. It should not be null.

In Fig.1 Super key can be Roll no., First name, Last name, Login, Age etc. Candidate key can be Roll no., First name and Login. Primary key is Roll no.

iv. *FOREIGN KEY:*

Foreign key is set of attributes that is used to refer a tuple of other relation.

INTEGRITY CONSTRAINTS:

Data integrity gives a mechanism to maintain data consistency. Types of data integrity constraints are namely:

- Entity integrity
- Referential integrity

i. *ENTITY INTEGRITY*:

Entity integrity implies that a primary key cannot be null. Since it uniquely identify a row in a relation.

E.g. consider a relation FOOTBALL; the attributes of the PLAYER are Name, Age, Nation, and Rank. In this example, let us consider PLAYER's name as the primary key even though two players can have same name. We cannot insert any data in the relation FOOTBALL without entering the name of the player. This implies that primary key cannot be null.

ii. *REFRENTIAL INTEGRITY:*

In relational data model we make associations between tables through foreign keys. So the referential integrity states that their must not be any unmatched foreign key values. This doesn't mean that a foreign key cannot be null. For every foreign key value there must be a primary key value in other relation that matches it or the foreign keymust be null.

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

Resource description framework is a framework for describing resources. Its approach is based on the idea of making statements about resources. In subject-predicate-object expression. These statements are called triples and they are stored in triple store. These triples are represented by directed graphs. The subject refers to resource and predicate denotes the relationship between subject and object.

 ISSN: 2321-8169
1879 – 1882

E.g. "The rose has the color red" if we change this statement to a RDF triple: the subject will be "the rose", predicate will be "has the color", and object will be "red".

RDF can be serialized with several serialization formats like XML, N3 and Turtle etc. This way the encoding of triples will vary from format to format.

Graph representation of RDF.

To understand RDF we first need to understand the model on which it is based on, the RDF data model.

I. RESOURCE DESCRIPTION FRAMEWORK DATA MODEL (BASIC RDF MODEL)

RDF data model is a way of representing RDF expressions. RDF Core Working Group decided that through directed graph we can easily represent and understand RDF expressions. Directed graphs doesn't assume about the data is holds. This decision was based on two reasons:

1. Graphs are easy to read and understand. There is no confusion about the subject, subject's values, and subject's properties. No confusion about the statements made, even in a complex RDF data model.

2. Graph is the default description techniques for RDF because there are RDF data models that can be represented in RDF graphs, but cannot be expressed in RDF/XML.

RDF directed graph contains a set of nodes which are connected by arcs in **node -arc- node** pattern. Node comes in three varieties:

1. URI (Uniform Resource Identifier) reference

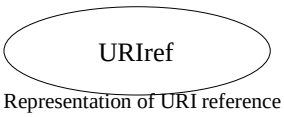
2. Blank nodes

3. Literals

I. URI Reference

Uniform Resource Identifier provides a unique identification to a node. It should point on something that is accessible on web. It may or may not have direct connectivity to web resource. If a resource is identified in a specific graph a URI is given to that resource.

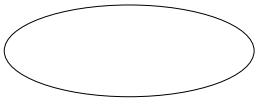
Uriref are drawn with an ellipse around them with URI written in them.



Representation of URI reference

I. Blank Node

Nodes which do not have URI's are referred as blank nodes. If identification of a node does not exist in a graph at the time graph was recorded or the given node isn't meaningful then it is diagrammed as blank node. Blank node is shown by empty circle.

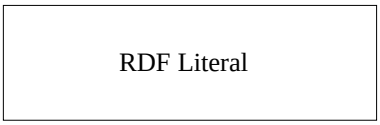


Representation of Blank Node

I. Literals

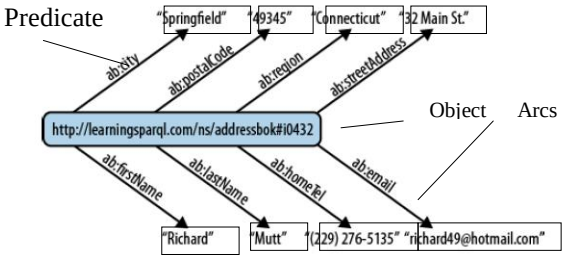
Targets of the graph can be pieces of text instead of resources; those pieces of text are called literals. Literal values represent RDF objects only. They do not represent subject or predicate.

Literals mainly consist of three parts: a character string, an optional language tag and data type. Literals are drawn with rectangles around them.



Representation of Literal

The arcs are directional. Arcs are drawn starting from resource and end at object. It also has an arrow to show the direction from resource to object with predicate written on it. If information is in pieces then each piece is connected to other through arbitrary relationship. Relationship can also operate in both directions.

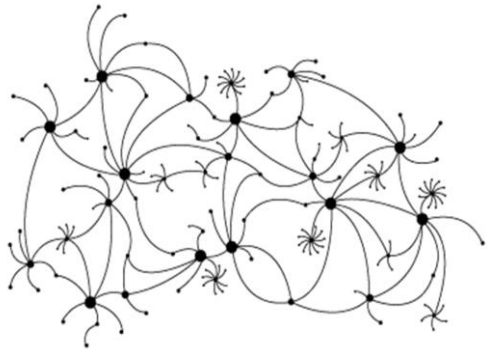


Graph representation of address book entry ab:i0432

In fig. 6 the object is represented with an URIref, arcs are drawn starting from subject and ending on object. Predicate describes the relationship between subject and object.

If we want to add more information to the graph we just need to create two new objects and then connect them to the original record object. There is no limit to the no. of relationship that two piece of data may share.

Without having prior knowledge of the data and schema we can still build a user interface using directed graphs. RDF is a boon for the industries whose data are complex and there are rapidly changing relationships in data objects.



View of how pieces of data can easily connect to the each other.

This database can grow easily and new resources can be added and process without changing the underlining architecture of the database.

CONCLUSION

From the above study of both the models this paper comes to a conclusion that RDF should be used where data is complex and data objects change their relationship rapidly. As RDF database leaves the data intact and allows maximum flexibility with data. And it is made according to the W3C standards. But we cannot say that it can universally replace relational database. Relational database has proved itself a good database with structured data. It stores data in tables which saves space, ideal for the enterprises etc. RDF is new and still growing.

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] Raghu Ramakrishnan, Johannes Gehrke "Database Management System" Second Edition, pp. 51-54

2] Shelley Powers, "Practical RDF", First edition 2003, pp.1-20.

3] http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/

4] http://www.w3.org/TR/rdf-concepts/

5] Hugh Darwen , "A Introdction to Relational Database Theory,"2010 in press.

6] S. Sumathi, S. Esakkirajan, "Fundamentals of Relational Database Management System,"2007.

en.wikipedia.org/wiki/Relational_database