

Converting Employee Relational Database into Graph Database

N.R.Prasanth, K.Arul

¹Under graduate Student, ²Assistant Professor

^{1,2}Dept. of CSE, Saveetha School of Engineering, Saveetha University

Abstract

Graph Database Management Systems provide an effective and efficient solution to data storage in current scenarios where data are more and more connected, graph models are widely used, and systems need to scale to large data sets. In this framework, the conversion of the persistent layer of an application from a relational to a graph data store can be convenient but it is usually an hard task for database administrators. In this proposal a methodology to convert a relational to a graph database by exploiting the schema and the constraints of the source. The approach supports the translation of conjunctive SQL queries over the source into graph traversal operations over the target. The experimental results are provided to show the feasibility of the solution and the efficiency of query answering over the target database.

Keywords

RDBMS, GDBMS, cipher

I. Introduction

There are various application that requires large data storage for storing their appropriate data. To do so the database management system (DBMS) is used. we use a software called SQL to create a database. Sometimes it is necessary to relate the two or more table in a database. To do so we use relational database system (RDBMS). However this relational database is not suitable for web applications, computer networks, geographical structure etc., moreover in these highly connected data applications requires complex join operation which can make typical operation on this kind of data inefficient application hard to scale. To overcome this problem we use graph database management system (GDBMS). In GDMS data are natively stored as graph and queries are expressed as graph traversal operation. This allows application to scale very large graph based data sets. In addition GDMS do not rely on any schema they provide more flexible solution in scenarios where the organization of data evolves rapidly. By using graph database rather than using the relational database is more beneficial. In graph database it follows a naive approach where tuples are mapped to nodes and foreign key is mapped into edges. In this paper the employee relational database is converted into the graph database for high performance. Specifically the relational database query is converted into the graph database query. The general graph model and generic query language for graph structures.

II. System Analysis

A. Existing system

A relational database is a database that has a collection of tables of data items, all of which is formally described and organized according to the relational model. Column to column relationship requires primary key. Row to row relationship requires foreign key

B. Proposed system

Graph Database Management Systems provide an excellent method to store the data .The graph models are widely used where the systems need to scale to large data sets .Here the relational database is converted into the graph database to provide efficiency of query answering. The approach supports the translation of conjunctive SQL queries over the source into graph traversal operations over

the target.

III. Requirement Specification

Hardware and Software Specification

1) Hardware Requirements

- Hard Disk: 80 GB and above.
- RAM: 2 GB DDR3
- Processor: i3 and above.

2) Software Requirements

- Mysql(Rdbms)
- Neo4j
- JVM
- 64 bit operating system

IV. Syetem Implementation

A. Module Explanation

1. RDBMS

A Relational database is a system in which data are arranged in the form of tables. The relationship between the tables can be maintained using primary key and foreign key.

2. Data Conversion

In this module the conversion of relational database to graph database is performed. The data in relational database in converted into graph database conversely, in our approach we try to aggregate values of different tuples in the same node to speed-up traversal operations. The basic idea is to try to store in the same node of g data values that are likely to be retrieved together in the evaluation of queries.

Schema of the source relational. For the sake of simplicity, we consider an intermediate step in which we map the SQL query in a graph-based internal structure, that we call query template (QT for short). Basically, a QT denotes all the sub-graphs of the target graph database that include the result of the query. A QT is then translated into a path traversal query. 4 .Graph DB A graph database is a type of database in which the data is stored in the form of nodes. In this database the retrival time for a data is very less when compared to relational database. It provides high efficient database in which data can be stored and retrieved easily.

This type of database can be used for processing both small and large sized database.

V. Conclusion

In this proposed system the employee relational database is converted into the graph database system. The graph database is considered to be the one of the emerging technology. The graph database is more effective form of database system. The time taken to add, manage and update a query in database gets very simple in graph database. It also requires only less coding to perform such operation when comparing to relational database. The comparison of relational database and graph database is shown in chart form and the above mentioned things are verified. In future works we intend to refine the technique proposed in this paper to obtain a more compact target database.

References

- [1] R. Angles and C. Gutierrez. *Survey of graph database models*. *ACM Comput. Surv.*, 40(1), 2008.
- [2] P. Atzeni, P. Cappellari, R. Torlone, P. A. Bernstein, and G. Gianforme. *Model-independent schema translation*. *VLDB J.*, 17(6):1347{1370, 2008.
- [3] C. Bizer. *D2r map - a database to rdf mapping language*. In *WWW (Posters)*, 2003.
- [4] C. Bizer and A. Schultz. *The berlin sparql benchmark*. *Int. J. Semantic Web Inf. Syst.*, 5(2):1{24, 2009.
- [5] F. Cerbah. *Learning highly structured semantic repositories from relational databases*. In *ESWC*, pages 777{781, 2008.
- [6] J. Coman and A. C. Weaver. *A framework for evaluating database keyword search strategies*. In *CIKM*, pages 729{738, 2010.
- [7] S. B. et al. *Keyword search over relational databases: a metadata approach*. In *SIGMOD*, pages 565{576, 2011.
- [8] M. Hert, G. Reif, and H. C. Gall. *A comparison of rdb-to-rdf mapping languages*. In *I-SEMANTICS*, pages 25{32, 2011.
- [9] F. Holzschuher and R. Peinl. *Performance of graph query languages - comparison of cypher, gremlin and native access in neo4j*. In *EDBT/ICDT Workshops*, pages 195{204, 2013.
- [10] W. Hu and Y. Qu. *Discovering simple mappings between relational database schemas and ontologies*. In *ISWC/ASWC*, pages 225{238, 2007.
- [11] J. B. Rodriguez and A. Gomez-Perez. *Upgrading relational legacy data to the semantic web*. In *WWW*, pages 1069{1070, 2006.
- [12] M. A. Rodriguez and P. Neubauer. *Constructions from dots and lines*. *CoRR*, abs/1006.2361, 2010.
- [13] J. Sequeda, M. Arenas, and D. P. Miranker. *On directly mapping relational databases to rdf and owl*. In *WWW*, pages 649{658, 2012.
- [14] P. T. Wood. *Query languages for graph databases*. *SIGMOD Record*, 41(1):50{60, 2012