

# NEO4J AS A SOLUTION TO HOSPITAL LOCALIZATION APPLICATION

Richa Kuriakose<sup>1</sup>, AnuSebastian<sup>2</sup> and Surekha Mariam Varghese<sup>3</sup>

Department of Computer Science and Engineering, M.A College of Engineering,  
Kothamangalam, Kerala, India

## ABSTRACT

*Graphs are efficient ways to visualize and represent real world data. Solutions to many real time scenarios can be easily provided when there is powerful graph databases like neo4j that can be used to efficiently query the graphs with multiple attributes. For instance, querying a system with medical and hospital data can be used to address the problem of location wise medical decision making. Here in this paper we present a neo4j as a solution to medical query.*

## KEYWORDS

*Graph databases, SQLite, cypher query language, Neo4j, Relational databases.*

## 1. INTRODUCTION

Nowadays people find it difficult to locate hospitals with a required specialization. For example, a patient who needs an appointment at the ophthalmologist, need to first identify those hospitals having eye specialists in his area. The information can be made available through an online application. The choice of database for the application is a determining factor in the speed, efficiency and ease of use of the online application.

The application that address the hospital and medical query can use any type of database. Relational Database Systems implement the relational model to work with the data. Relational databases are commonly used solution for data storage in almost all applications[16],[17]. SQL is the language used for querying and maintaining the relational databases [4],[13],[14]. SQLite is a relational database that is arguably the most widely deployed database engine, as it is used today by several operating system, browser and embedded systems. SQLite has bindings to many programming languages. As a self-contained, file-based database, SQLite offers an amazing set of tools to handle all sorts of data with much less constraint and ease compared to hosted, process based (server) relational databases. These features of SQLite made us to choose SQLite as the representative of relational databases to build Hospital localization application. But the relational databases have restriction in terms of size of the database. The connection between the entities in relational databases is done using the foreign key. But the connections and relations can be easily represented in the graph databases using the nodes and edges.

Graph databases enable us to build sophisticated models that map closely to our problem domain. Graph databases are schema less and efficient storage for semi structured data[12],[15]. They

express queries as traversals. They can do fast deep traversals instead of slow SQL joins. They allow ACID transactions with rollback support. The common graph databases includes Neo4j, Flockdb etc. Neo4j is an open-source graph database, implemented in Java. It employs the mathematics of graph and utilizes its huge potential for fast information extraction speeds to store information in the form of nodes and relationships. In this paper, we perform a comparison on both SQLite and Neo4j on the platform of hospital application.

## 2. PROBLEM DEFINITION

Graph databases are databases that uses graph structures for semantic queries with nodes edges and properties [1], [2], [12], [15]. In a graph database nodes represent entities such as people, business, accounts or any other item. Properties are information that relate to nodes. Edges are lines that connect to nodes or nodes to properties and they represent the relationship between the two. Most of the important information is really stored in the edges. Meaningful patterns emerge when one examines the connections and interconnections of nodes, properties, and edges. Figure 1 shows how data is represented in a graph database.

Neo4j is the most popular graph data base use in today [3]. Neo4j uses declarative query language called cypher query language for querying [5], [6]. Very complicated queries can be expressed through cypher. Cypher is relatively simple but still powerful. Cypher is designed to be a human query language. It makes simple things easy and complex things possible. Its constructs are based on English prose and queries are self-explanatory. Being a declarative language, Cypher focuses on the clarity of expressing what to retrieve from a graph, not on how to retrieve it. Cypher syntax provides a familiar way to match patterns of nodes and relationships in the graph.

Hospital localization is an application designed to find out the primary health centers and the specialized hospitals in an area, having the list of all hospitals, specialization and distance between the hospitals. The application allows fast and efficient best effort search in emergency situations. In situations like accident or other medical emergencies we may have to find the closest primary health center for getting first aid and after that from there it may be necessary to shift to the nearest hospital with required specialization. In such circumstances it is necessary to find the nearest health center and specialized hospital closest to the nearest health center. This type of query requires different levels of traversal, if we represent location wise hospital data and connection between them as a graph. For answering such queries we need a powerful graph database that combines the expressiveness of graph and dependability of database. Neo4j is the best choice. Because Neo4j is a graph database with powerful and simple query language (Cypher Query Language- CQL) capable of expressing complex queries. Relational databases cannot answer such queries that easily. In hospital localization application the distance between the hospitals need to be represented as an attribute to the edge. Neo4j allows to add properties to nodes as well as edges.

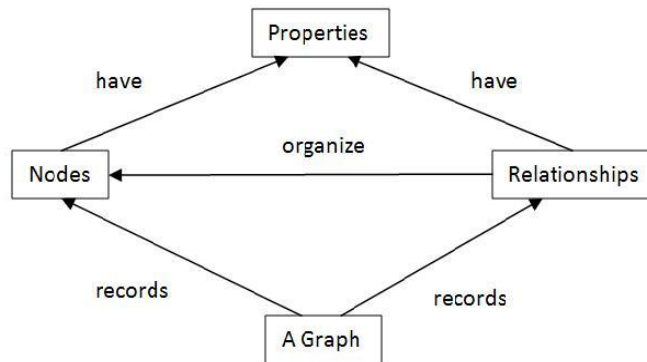


Fig. 1. Data representation in graph database

In relational database like SQLite we need different tables to represent the hospital localization application. If we use Neo4j for the hospital localization application the hospitals can be represented as nodes and the connection between them by road can be represented as the relationship between them. We can have specializations in a hospital as properties for the nodes and the distance between the hospitals as the property for the relationship. Figure.2 shows an instance of the graph for the hospital localization application including the hospitals in the Kothamangalam town within 100 meters from the National Highway 49.

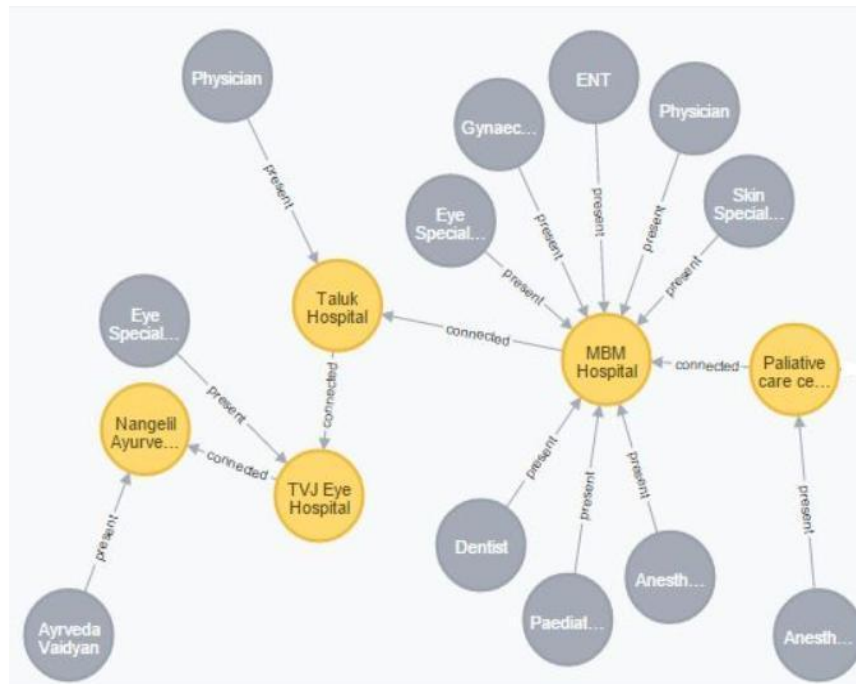


Fig. 2. Instance from Hospital Localization Application

The Figure depicts hospitals with their specialization and distance between them. The application allow different type of queries like, query for required specialization, query for hospitals within a given distance, query for the closest hospital or closest hospital with desired specialization etc.

### 3. METHODOLOGY

A detailed view of how hospital localization can be implemented with SQLite and Neo4j is discussed in this session. Let's first see the implementation with SQLite. It is necessary to have a table to store hospital information, another table to store distance information. For querying a hospital within a distance from a location it is necessary to perform JOIN operation on the tables. For example, if a patient wants to localize a hospital specialized in ophthalmologist, he could use a simple select query for determining those hospitals from a particular distance from his location Since JOIN operations are used in case of SQLite, it takes more accumulation of time.

In Neo4j we use cypher query language to develop the hospital localization application as a graph database [7]. It follows SQL like syntax, which is very simple and human readable. Let's now see the implementation of Neo4j. The procedure for implementing hospital localization application using Neo4j is given below:

1. Create nodes representing hospitals
2. Associate each hospital with the specializations as properties
3. Associate hospital nodes with relationship and add distance as property to the relationship.

#### *Step1. Create nodes representing hospitals*

Nodes for each hospital in an area are created using special queries. Each hospital node have a set of properties or attributes associated. The hospital nodes form the major nodes of a particular locality.

#### *Step2. Associate specializations to hospital nodes*

Specializations are added to each hospital nodes using links. Links determine the presence of specialization in a hospital. This association helps in fast checking of specializations in a hospital.

#### *Step3. Associate hospital nodes with relationship*

The hospital nodes of a locality are associated using relationship. The edges representing relationships as "connected" are created. This association helps in the faster traversal through hospitals in a locality. "Distance" is added as a property for the relationship. These are done using the "CREATE" constructs in CQL and Querying is done mainly using "MATCH", "WHERE", "WITH" and "RETURN".

### 4. PERFORMANCE EVALUATION

On analyzing the performance of both Neo4j and SQLite on hospital localization application, the performance of Neo4j is more when compared to that of SQLite. An ideal graph database should understand analytic queries that go beyond neighborhood selection. In relational databases like SQLite, the index represents pre-determined knowledge of the structure of the computation without knowing the specific input parameters. A relational query selects a subset of the data and

joins it by known fields [10], [11]. The JOIN operations decelerates the query by pre-computing a portion of the query. In a graph database like Neo4j, the equivalent index is the portion of an analytic or graph algorithm that can be pre-computed or kept updated regardless of the input parameters of the query [8], [9]. Examples include the connected components of the graph, to localize the hospitals. Hence on regard to the efficiency factor, SQLite is not a good choice.

Neo4j has more accuracy than SQLite. Neo4j is much faster than SQLite. It is easy to use Cypher query language when compared to JAVA. Cypher is a declarative graph query language for the graph database, Neo4j that allows for expressive and efficient querying and updating of the graph store. Cypher is a relatively simple but still very powerful language. Very complicated database queries can easily be expressed through Cypher. This allows users to focus on their domain instead of getting lost in database access.

For evaluating the performance of SQLite and Neo4j we took three queries. The results are shown in Fig 3.

- Query 1: Find all hospitals in a locality within a distance
- Query 2: Find the specializations in a particular hospital
- Query 3: Find the ophthalmologists within a restricted area

Table1: Neo4j Vs SQLite

Name	Neo4j	SQLite
Description	Open source graph Database	Widely used in-process RDMS
Database model	Graph DBMS	Relational DBMS
Implementation Language	Java	C
Server operating Systems	Linux OS X Windows	server-less
Data scheme	Schema-Free	yes
Typing	Yes	Yes
Secondary Indexes	Yes	Yes

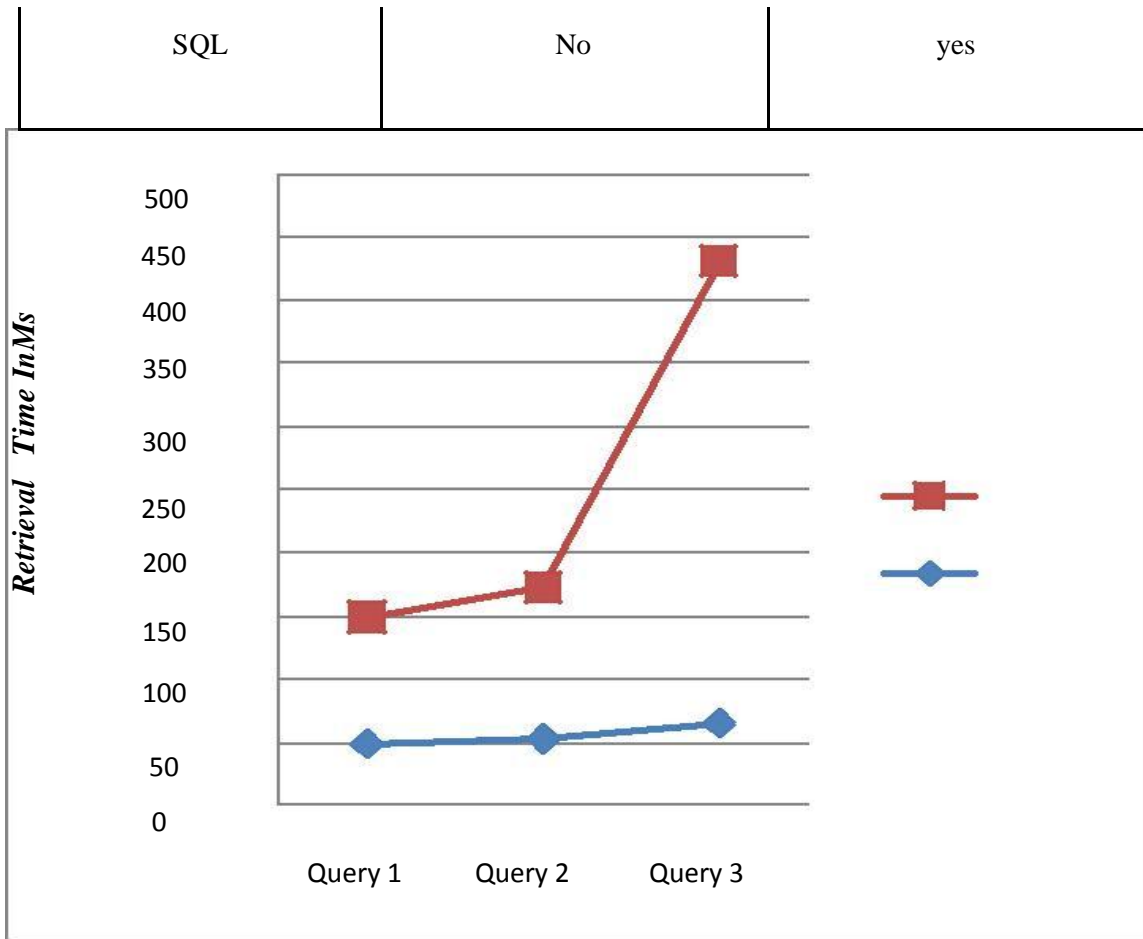


Fig 3. Retrieval time for queries in Neo4j and SQLite

Table 1 shows a comparison of Neo4j and SQLite based on some basic features. Figure.3 shows the retrieval time for the above mentioned queries for Neo4j and SQLite in milliseconds. A comparative rise can be seen for Query 3 for SQLite when compared to the Query 3 for Neo4j. This is due to the rise in retrieval time to select the ophthalmologists within a restricted area as per Query 3.

## 5. CONCLUSION

SQLite is a relational database that is arguably the most widely deployed database engine, as it is used today by several operating system, browser and embedded systems. SQLite has bindings to many programming languages. Graph databases enable us to build sophisticated models that map closely to our problem domain. Graph databases are schema less and efficient storage for semi structured data.

In this paper, we undergo performance evaluation between Neo4j and SQLite on hospital localization application. We created hospital localization application using both SQLite and

Neo4j and also plotted the retrieval time for three queries in Neo4j and SQLite. We came to a conclusion that when compared, Neo4j performs better than SQLite. Neo4j has more accuracy than SQLite. Neo4j is much faster than SQLite. It is easy to use Cypher query language when compared to JAVA. Cypher is a declarative graph query language for the graph database, Neo4j that allows for expressive and efficient querying and updating of the graph store. Cypher is a relatively simple but still very powerful language. Very complicated database queries can easily be expressed through Cypher. This allows users to focus on their domain instead of getting lost in database access.

Neo4j is a graph database whereas SQLite is a relational database. On comparison overall performance of graph databases exceeds the relational database. In conclusion, given a traversal of an artificial graph with natural statistics, the graph database Neo4j is more optimal than the relational database SQLite.

## REFERENCES

- [1] ShaliniBatra, CharuTyagi. Comparative Analysis of Relational And Graph Databases. International Journal of Soft Computing and Engineering (IJSCE), 2(2), May 2012.
- [2] Renzo Angles And Claudio Gutierrez.” Survey of Graph Database Models”ACM Computing Surveys, Vol. 40, No. 1, Article 1, Publication date: February 2008.
- [3] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. The 5th ACM/USENIX Internet Measurement Conference, 2007.
- [4] “NoSQL Databases”, <http://nosql-database.org/>
- [5] “Neo4j,” <http://neo4j.org/>.
- [6] Neo4jmanual, Internet: <http://docs.neo4j.org/chunked/stable/graphdb-neo4jnodes.html> ,2010
- [7] J. Paredaens and B. Kuijpers, “Data Models and Query Languages for Spatial Databases,” Data & Knowledge Engineering (DKE), vol. 25, no. 1-2, pp. 29–53, 1998
- [8] P. Urban, “Nosql graph database matrix,” <http://nosql.mypopescu.com/post/619181345/nosql-graph-databasematrix>, May 2010.
- [9] “Short overview on the emerging world of graph databases,”<http://www.graphdatabase.org/overview.html>
- [10] Michael Owens, The Definitive Guide to SQLite, USA: Apress, 2006, 341-362.
- [11] SQLite homepage [EB/OL] ,<http://www.sqlite.org>.
- [12] Florian Holzschuher, Prof. Dr. René Peinl. Performance of Graph Query Languages: Comparison of Cypher, Gremlin and Native Access in Neo4j.EDBT/ICDT`13March 18 - 22 2013.e
- [13] SzabolcsRozsnyai, AleksanderSlominski, YurdaerDoganata.Large-Scale Distributed Storage System for Business Provenance.CloudComputing (CLOUD), IEEE International Conference,2011.
- [14] IlyaKatsov. NoSql Data Modeling Techniques. March 2012.
- [15] Ian Robinson, Jim Webber, Emil Eifrem. Graph Databases(Early release revision 1).OReilly Media, Inc., 02-25 2013.
- [16] Apache Software Foundation.Home.<http://zookeeper.apache.org>, 2013.
- [17] YanmeiHuo, Hongyuan Wang, Liang Hu, Hongji Yang. A Cloud Storage Architecture Model for Data-Intensive Applications. 2011.

## AUTHORS

RichaKuriakose is currently pursuing M.Tech in Computer Science and Engineering in Mar Athanasius College of Engineering, Kothamangalam. She completed her B.Tech from Mar Athanasius College of Engineering, Kothamangalam. Her areas of research are DataMining and Machine Learning.



Anu Sebastian is currently pursuing M.Tech in Computer Science and Engineering in Mar Athanasius College of Engineering, Kothamangalam. She completed her B.Tech from Pulincunnoo Engineering College, Alappuzha. Her areas of research are DataStructures and Algorithms and Biocomputing.



Surekha Mariam Varghese is currently heading the Department of Computer Science and Engineering, M.A. College of Engineering, Kothamangalam, Kerala, India. She received her B-Tech Degree in Computer Science and Engineering in 1990 from College of Engineering, Trivandrum affiliated to Kerala University and M-Tech in Computer and Information Sciences from Cochin University of Science and Technology, Kochi in 1996. She obtained Ph.D in Computer Security from Cochin University of Science and Technology, Kochi in 2009. She has around 25 years of teaching and research experience in various institutions in India. Her research interests include Network Security, Database Management, Data Structures and Algorithms, Operating Systems, Distributed Computing and Machine Learning. She has published 17 papers in international journals and international conference proceedings. She has served as reviewer, committee member and session chair for many international conferences and journals.

