

Neo4j Graph Database Implementation for LinkedIn

Punam Nikam, Sachin Bhoite, Anuj Shenoy

Department of Computer Science, MIT-WPU, Pune, Maharashtra, India

ABSTRACT

Article Info

Volume 6, Issue 6

Page Number: 339-342

Publication Issue :

November-December-2020

Article History

Accepted : 20 Dec 2020

Published : 30 Dec 2020

The presented article mainly circumspect the idea for the use and implementation of graph database in the most social media of today. Currently many companies are using neo4j graph database for their workouts for data management. We shall stick with social media for this particular paper. We have used neo4j graph database for maintaining data of LinkedIn user and pages in a very systematic manner. Facebook and Twitter do currently use graph databases so we thought of implementing the same for LinkedIn.

Keywords: Neo4j, Graph Database, Computer Science, LinkedIn, Information Technology

I. INTRODUCTION

LinkedIn is a social platform mostly used for a professional purpose with users connecting to each other for jobs, business etc, and businesses searching for the right professionals for their organization. A modern way of job hunt for users and recruiting for companies.

LinkedIn started slowly in 2002 and then around 2008 marked its rapid increase in terms of user accounts registered. The graph below shows exactly how rapidly the growth increased from 2009 to 2016.

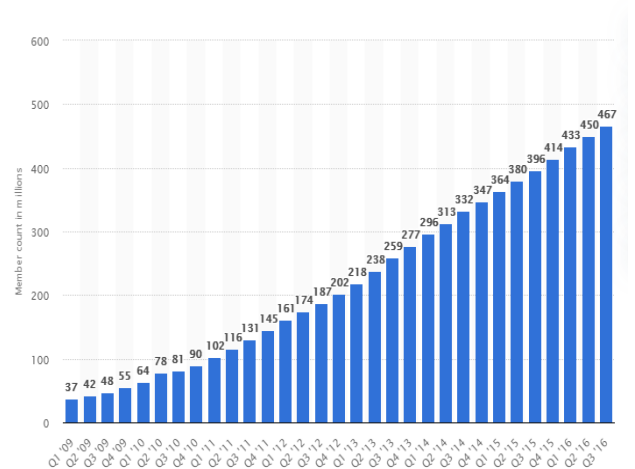


Figure 1: (Data represents millions per quarter)

LinkedIn traditionally started with using Relational Database Management Systems (RDBMS) such as Oracle and key-value stores such as Voldemort. AS they kept growing in numbers data collection was a headache in terms of horizontal scaling, having zero downtime and many more. LinkedIn then came up

with a solution of using a NoSQL database named Espresso.

Espresso is LinkedIn's online, distributed, fault-tolerant NoSQL database that currently powers approximately 30 LinkedIn applications including Member Profile, InMail, portions of the Homepage and applications, etc. Espresso has a large production footprint at LinkedIn with over a dozen clusters in use. It hosts some of the most heavily accessed and valuable datasets at LinkedIn serving millions of records per second at peak. It is the source of truth for hundreds of terabytes (not counting replicas) of data. This gave us a fair idea of trying to implement Neo4j to try and maintain all applications for LinkedIn.

Neo4j is a graph database which was initiated in 2010. It is implemented in Java and uses Cypher Query Language (CQL). It went through lots of changes and many enhanced versions were launched of which the latest and most stable was launched in November 2020.

Neo4j stores data in a node, attribute, edge form of data structure, where each node and edge can have multiple number of attributes. Cypher Query Language is a declarative Graph query language that allows for expressive and efficient data querying in a property graph. Cypher organizes data into nodes and edges that are called relationships. It shares some similar keywords to that of SQL. So, it is supposed to be quite similar to SPARQL and Gremlin languages for querying graph data.

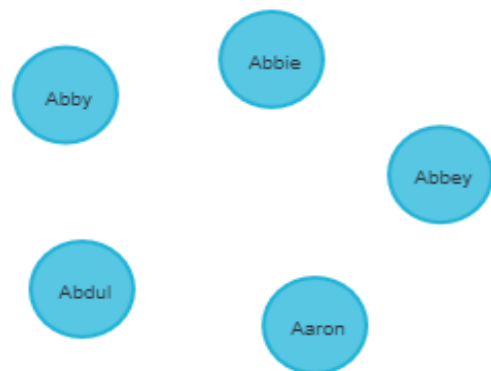
II. APPROACH AND WORKING

Using Neo4j, we decided to replicate a small version of a LinkedIn database to see if CQL can be implemented on the same using graphs. The two main entities that we decided here were, the user accounts and company pages. The user accounts label

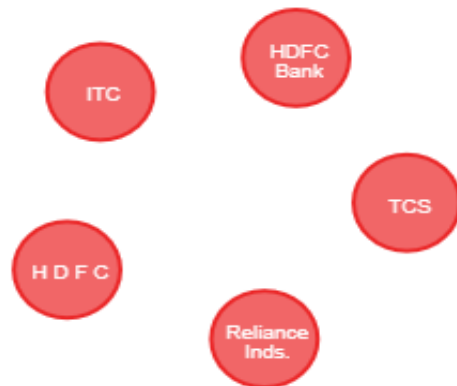
will have multiple nodes with all the users that are registered. The pages label will contain all the nodes of the company pages that are registered. Thereon, we start with creating all the relationships between all the users and also with the pages. A brief implementation of the graphs and relationships is shown below that are panned upon all nodes.

A. Queries and Graphs.

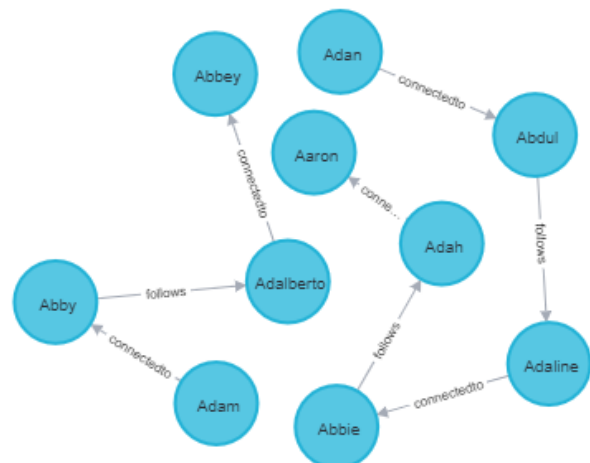
Account Nodes (Sample)



Page Nodes (Sample)



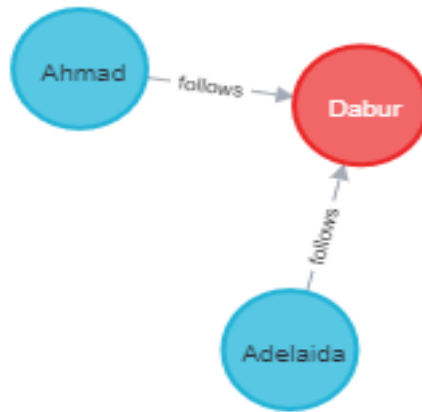
Account ->Account (Connection Relationship)



Account->Account (Connection + Follow)

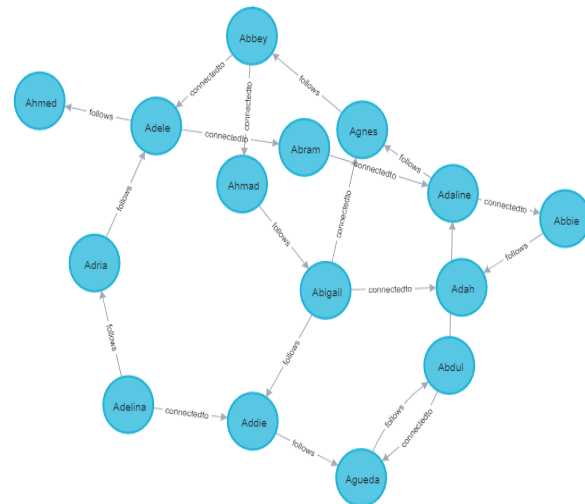
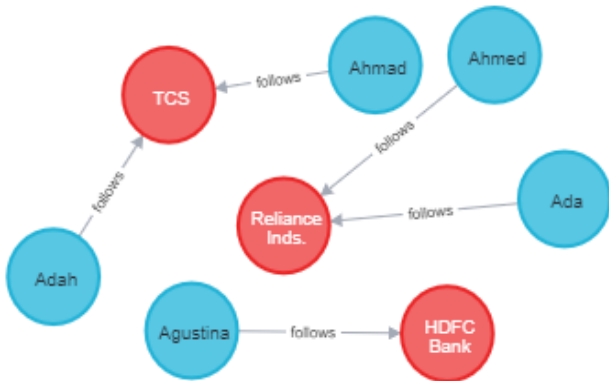


2. List all people that follow Dabur.



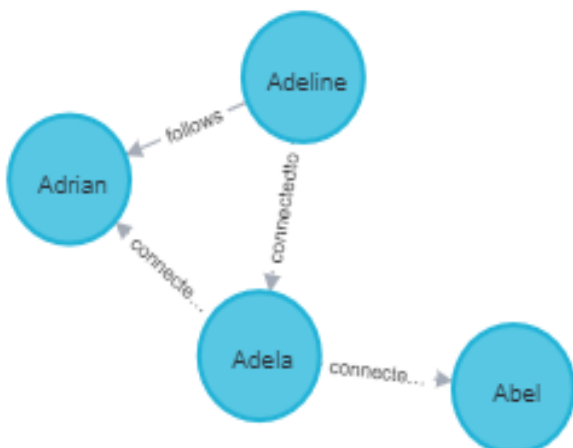
3. List all people having minimum experience of 4 years.

Account -> Page(Follow)

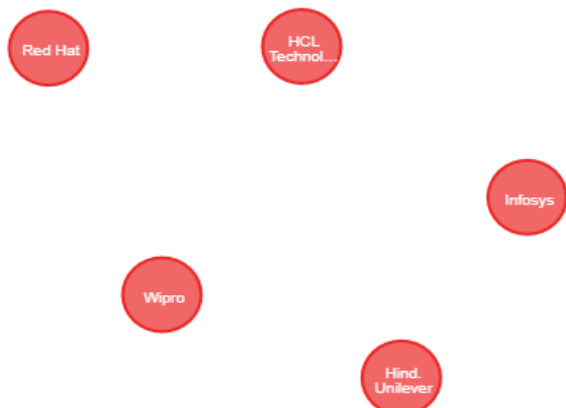


B. Query Implementation.

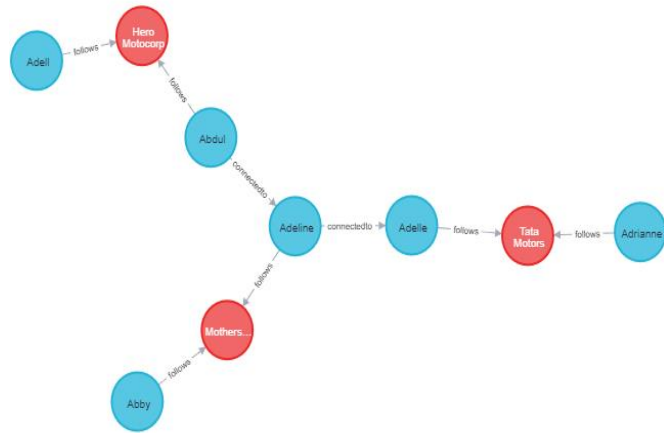
1. List all people connected to Adela.



4. List all pages that are of IT and FMCG sector.



5. List all people following HDFC or Companies from automobile sector.



III.RESULTS AND DISCUSSION

As shown above with all the result display of query implementation we can see how we can store and manage data in an easy and precise manner. Data retrieval is also very fast compared to NoSQL. Clearly Neo4j holds the edge over Espresso and can be used as well.

IV.CONCLUSION

We hereby conclude that Neo4j Graph Database can be a strong alternative for NoSQL or MYSQL when it comes to social media platforms data management. As mentioned earlier data management and retrieval is far faster and visually precise to get the desired output.

The future scope for the implementation would be to try and get the InMail, post and messaging applications that are provided by LinkedIn and store them as Graphs in Neo4j.

This will be a significant and very important stepping stone for the social media platforms data management.

V. REFERENCES

- [1]. Bangera, The Neo4j Cypher Manual v4.2 (<https://neo4j.com/docs/cypher-manual/current/>)
- [2]. Neo4j Official Documentation (<https://neo4j.com/docs/>)

Cite this article as :

Punam Nikam, Sachin Bhoite, Anuj Shenoy, "Neo4j Graph Database Implementation for LinkedIn ", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 6 Issue 6, pp. 339-342, November-December 2020. Available at doi : <https://doi.org/10.32628/CSEIT206665>
 Journal URL : <http://ijsrcseit.com/CSEIT206665>