

Neo4j: The Graph Database

Kavitha Raju

BridgeConn

February 6, 2019

Presentation Outline

1 Graph Database

2 Neo4j

3 A Sample Bible DB

4 Conclusion

Why a Graph Database?

Graph Databases are not replacement for the well established RDBMS system. They are suited for new use-cases where we have much more connected data to be handled.

Representing data as graph is more

- Simple
- Intuitive
- Expressive

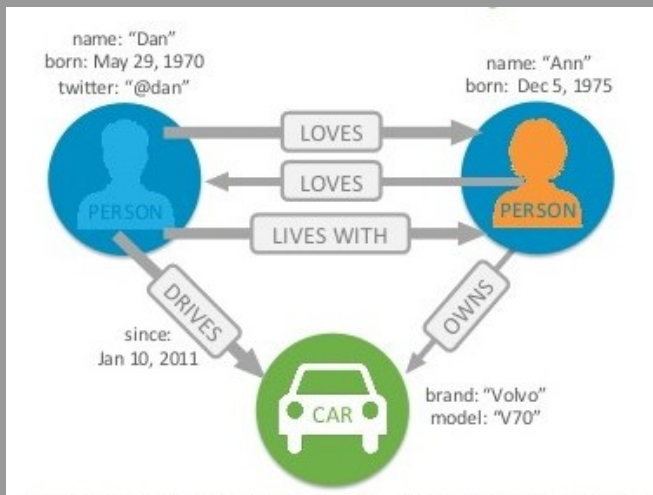
Labeled Property Graph

Charateristics

A Labeled Property Graph has the following characteristics:

- It contains nodes and relationships.
- Nodes contain properties (key-value pairs).
- Nodes can be labeled with one or more labels.
- Relationships are named and directed, and always have a start and end node.
- Relationships can also contain properties.

Labeled Property Graph



Nature and Power of Graph DB

High Performance

Has better performance with **larger connected data**. Though in relational databases, where join-intensive query performance deteriorates as the dataset gets bigger, with a graph database performance tends to remain relatively constant, even as the dataset grows

Agile and Flexible

The **schema-free nature** of graph DB makes it more flexible and agile. We can we can add new kinds of relationships, new nodes, new labels, and new subgraphs to an existing structure without disturbing existing queries and application functionality.

Presentation Outline

1 Graph Database

2 **Neo4j**

3 A Sample Bible DB

4 Conclusion

With native graph storage and native graph processing Neo4j has an architecture engineered for performance and scalability.

- Native graph storage

It has native graph storage. It stores and manages data as graph, and does not serialize the graph data into a relational database.

- Native graph processing

It employs native graph processing as well. That is, it uses index-free adjacency, meaning that connected nodes physically point to each other in the database and exposes a graph data model through CRUD operations.

Presentation Outline

1 Graph Database

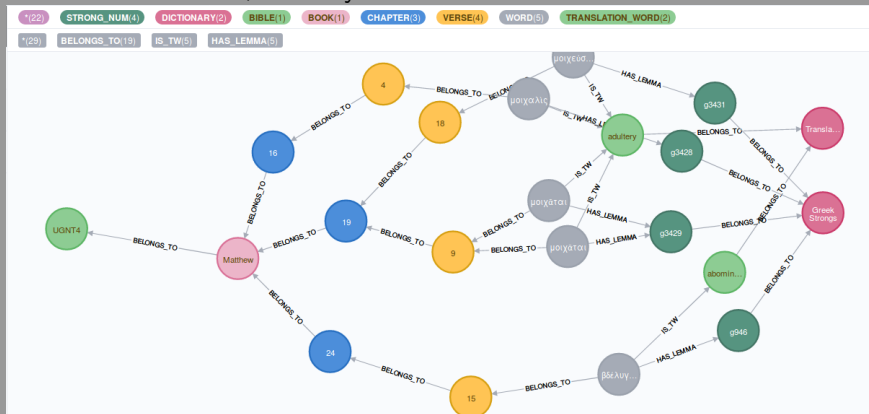
2 Neo4j

3 A Sample Bible DB

4 Conclusion

UGNT Bible in Neo4j DB

The following is a sample design to demonstrate the graph DB. It has the Greek Bible, word by word.



The following cypher query fetched the above graph

```
match p=((bib:BIBLE)<-[:BELONGS_TO]-(:BOOK)
<-[:BELONGS_TO]-(:CHAPTER)<-[:BELONGS_TO]-(:VERSE)
<-[:BELONGS_TO]-(w:WORD)-[:IS_TW]->(:TRANSLATION_WORD)
-[:BELONGS_TO]->(:DICTIONARY))
with p,w
match (w)-[:HAS_LEMMA]->(n:STRONG_NUM)-[:BELONGS_TO]
->(d:DICTIONARY)
return p,n,d limit 5
```

Presentation Outline

1 Graph Database

2 Neo4j

3 A Sample Bible DB

4 Conclusion

How to Utilise Graph DB?

Identify the areas where graph DB would be most appropriate.

- Data suited for Graph DB
 - Alignment Data
 - Syntax Trees
 - ...?
- Use cases for Graph DB based applications
 - Smart search in Bible
 - ...?

