

# Neo4j

M. Gorlas, N. Adamavicius

FHTenL

March 28, 2024

# Roadmap

1. Data model and schema
2. Consistency and Replication
3. Security and Performance
4. Specific use cases and bit of history
5. Demo

# Data model

Neo4j is a **graph database**.

There are two main types of graph databases:

- Property graph model
- RDF graph model

# Property graph model

In Neo4j information is organized as nodes, relationship and properties.

# Property graph model

In Neo4j information is organized as nodes, relationship and properties.

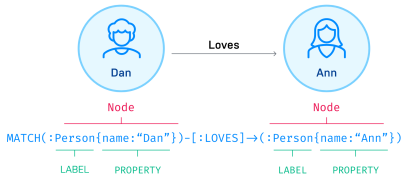


Figure 1: Example of the property graph model<sup>1</sup>

In the property graph model, **nodes** are the **entities** in the graph. **Relationship** provide directed, named connections between two nodes entities.

<sup>1</sup>Neo4j Inc. What is a graph database? URL: <https://neo4j.com/docs/getting-started/get-started-with-neo4j/graph-database/>.

# Database schema

```
1 CREATE (forrestGump:Movie {title: 'Forrest Gump',  
2 released: 1994})  
3 CREATE (robert:Person:Director {name: 'Robert Zemeckis',  
4 born: 1951})  
5 CREATE (tom:Person:Actor {name: 'Tom Hanks', born: 1956})  
6 CREATE (tom)-[:ACTED_IN {roles:  
7 ['Forrest']}]->(forrestGump)  
8 CREATE (robert)-[:DIRECTED]->(forrestGump)
```

2

# Consistency

How is the CAP theorem applied to the DBMS? What type of consistency is maintained in the system? Can you choose?

# Replication

Describe how replication works in the DBMS.



# Security

- Schema-based Security<sup>3</sup>
- Role-based access control<sup>4</sup>

---

<sup>3</sup>Neo4j Inc. *Fine-Grained Access Control for Better Security and Privacy*. URL:

<https://neo4j.com/product/neo4j-graph-database/security/>.

<sup>4</sup>Neo4j Inc. *Built-in roles and privileges*. URL:

<https://neo4j.com/docs/operations-manual/current/authentication-authorization/built-in-roles/#auth-built-in-roles>.

# Schema-based Security

- Protect the nodes and relationships by controlling users' ability to traverse and read from different parts of the graph.
- Ensures that only authorized users have access to the data they need to protect sensitive data.

# Role-based access control

- An approach, where you can apply restrictions to roles assigned to users at any level of granularity throughout the graph.
- Simplifies the task of assigning permissions and helps ensure that your data is secure.

# Performance

- Compared to relational DBMS (MySQL in this case)
- Compared to other NoSQL DBMS

## Compared to MySQL

Based on the benchmark using real-world data from Career Village, the experiment done by Rodrigues et. al, showed that Neo4j was faster than MySQL in most cases, particularly in pattern matching and recursive queries. However, MySQL has advantages in terms of data consistency and transactional support.

Category	Query	Neo4j	MySQL
Selection	Q1	2ms	31ms
	Q2	8ms	323ms
	Q3	32ms	438ms
Recursive	Q4	2ms	757ms
	Q5	2ms	290ms
	Q6	3ms	305ms
Aggregation	Q7	43ms	146ms
	Q8	18ms	40ms
	Q9	62ms	290ms
Pattern Matching	Q10	5ms	360ms
	Q11	10ms	455ms
	Q12	1ms	68ms

Figure 2: Performance comparison between Neo4j and MySQL<sup>5</sup>

<sup>5</sup>Cajetan Rodrigues, Mit Ramesh Jain, and Ashish Khanchandani. *Performance*

*Comparison of Graph Database and Relational Database.* May 2023. doi:

10.13140/RG.2.2.27380.32641.

# Compared to MySQL

In the benchmark the following types of queries were used:

- selection/search
- recursion
- aggregation
- pattern matching

# Compared to other NoSQL DBMS (1/2)

Based on the WDBench, a benchmark for graph databases focused on querying the Wikidata, Neo4j was the slowest of all tested graph databases, on all types of queries.<sup>6</sup>

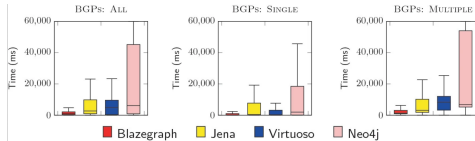


Figure 3: Basic Graph Patterns queries

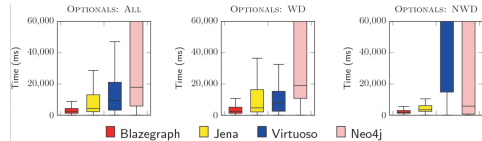


Figure 4: Optional Graph Patterns queries

<sup>6</sup>Renzo Angles et al. *WDBench: A Wikidata Graph Query Benchmark*. Oct. 2022. DOI:

## Compared to other NoSQL DBMS (2/2)

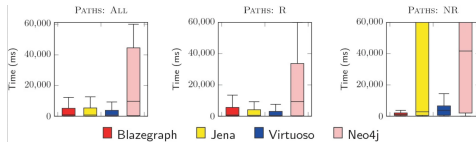


Figure 5: Path Patterns queries

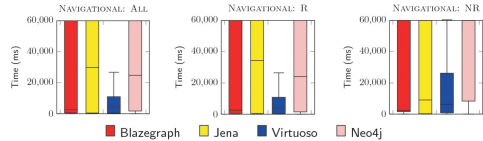


Figure 6: Navigational Graph Patterns queries



# Specific use cases

For which kind of use cases is the DBMS specifically good for, and why?

# History

Provide some economic information, e.g. market share, history, and any famous case(s)

# Demo