Interaktion zwischen Abgeordneten des Bundestages

- Gruppe 4 -

Information Systems (WiSe 2020)

Jennifer Vormann, Rico Stucke, Florian Thom

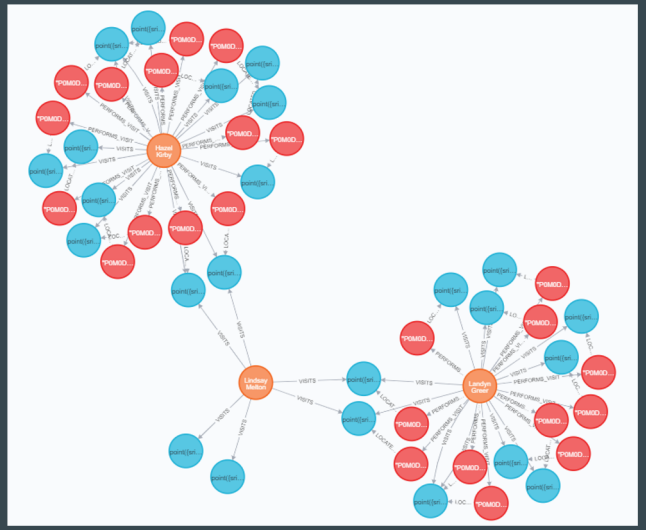
Supervisor: Prof. Dr. Hoppe

13. November 2020

Gliederung

1. Graphdatenbanken
    a. Allgemeines
    b. Funktionsweise
2. Ausgangssituation
3. ER Modell Entwurf
4. Aufgaben
5. Zeitplan
6. Quellen

Graphdatenbanken: Allgemeines



Graph

- Ein Graph G ist ein Paar (V,E)
  - V Menge von Knoten/Ecken.
  - E Menge von Kanten.
- Unterscheidung: gerichtet, ungerichtet

Graphdatenbanken

- “A graph database management system [...] is an online database management system with Create, Read, Update, and Delete (CRUD) methods that expose a graph data model” - I. Robinson
  - Speicher-Model
  - Processing-Engine

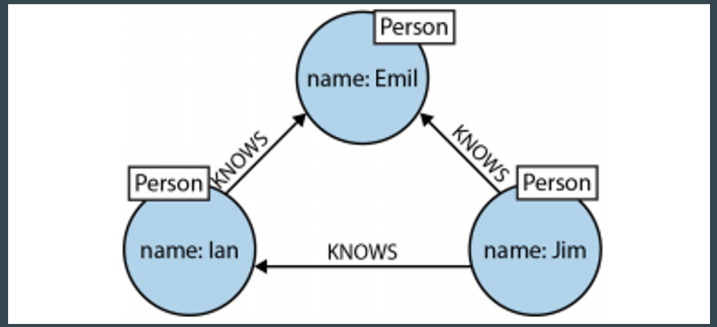
Graphdatenbanken: Allgemeines

Graphdatenbanken

- Vorteil:
  - Ermöglichen Analyse tiefgehender Relationsverbindungen
  - Zunehmende Größe der Datenbank wirkt sich unwesentlich auf die Performance aus

Graph-Modelle

- Labeled Property Graph
- Weitere



I. Robinson, J. Webber, E. Eifrem: “Graph Databases” (2013), O'Reilly Media

Graphdatenbanken: Funktionsweise 1

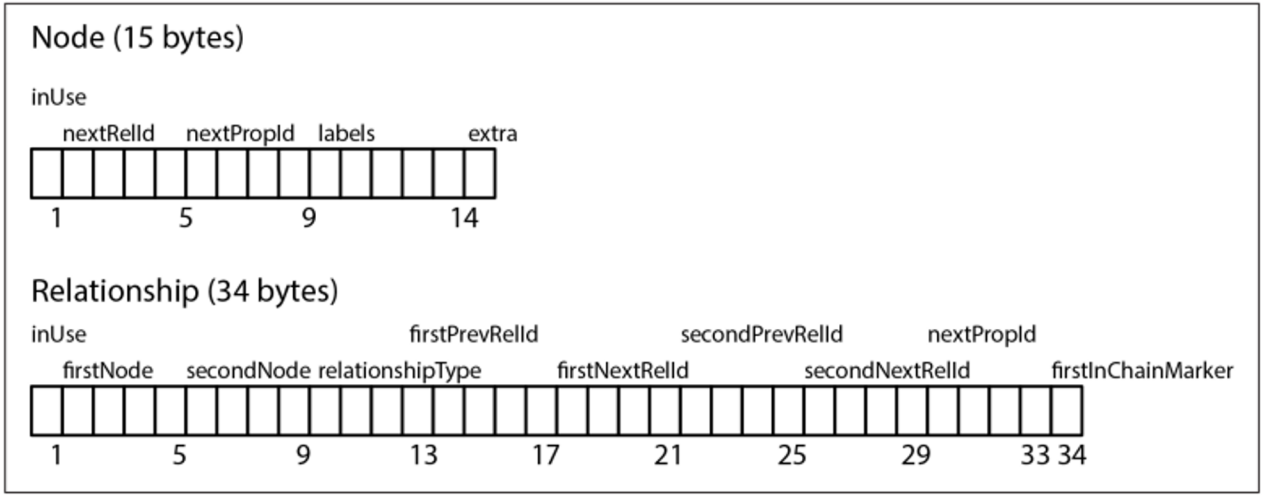
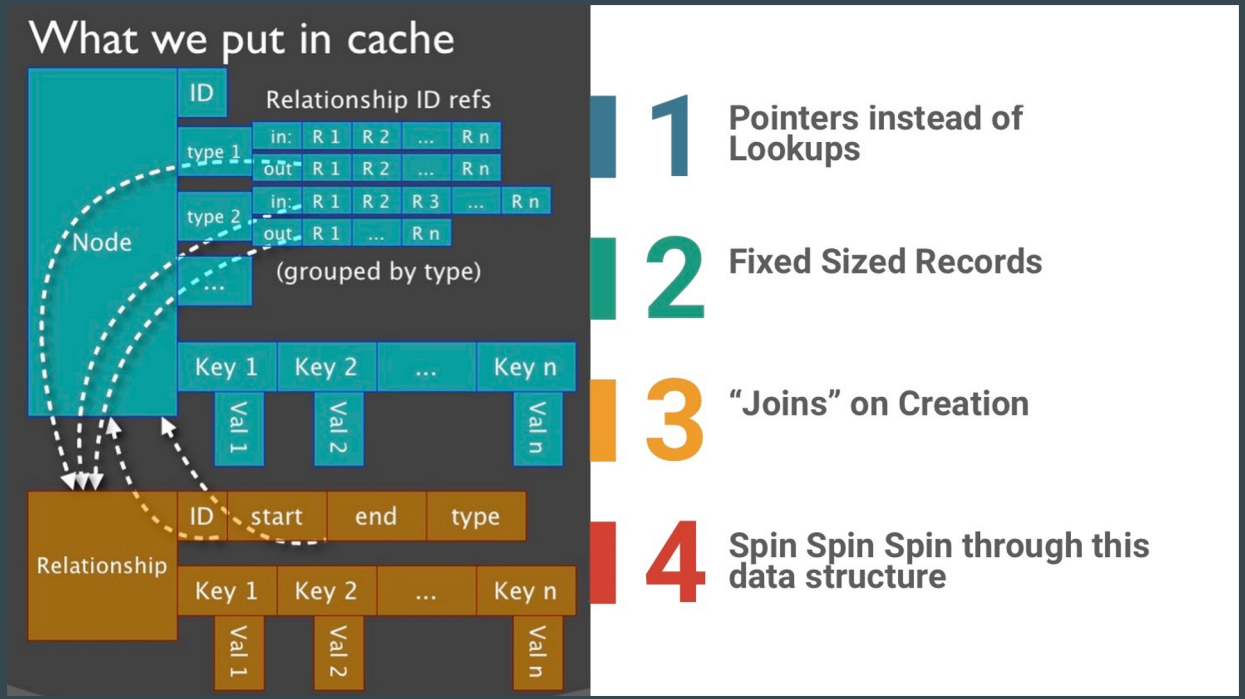


Figure 6-4. Neo4j node and relationship store file record structure

I. Robinson, J. Webber, E. Eifrem: "Graph Databases" (2013), O'Reilly Media

Graphdatenbanken: Funktionsweise 2



Quelle: Talk von Max De Marzi (https://neo4j.com/blog/secret-sauce-neo4j-modeling-graphconnect/)

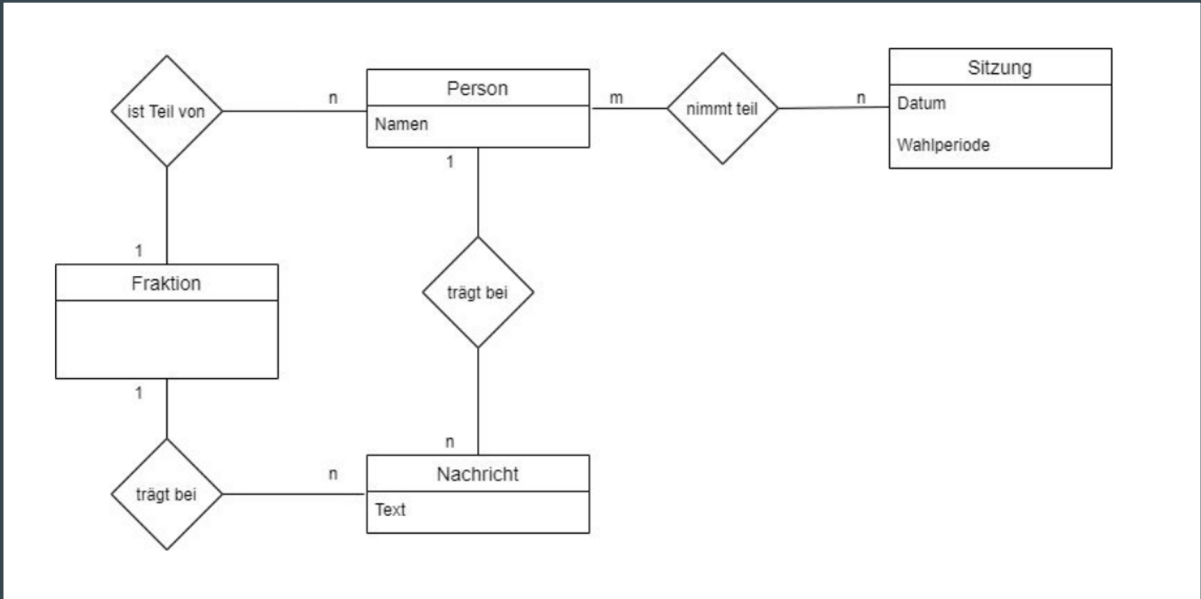
Ausgangssituation 1:

Schema ist problembezogen → aufgabenbezogener Input ist notwendig:

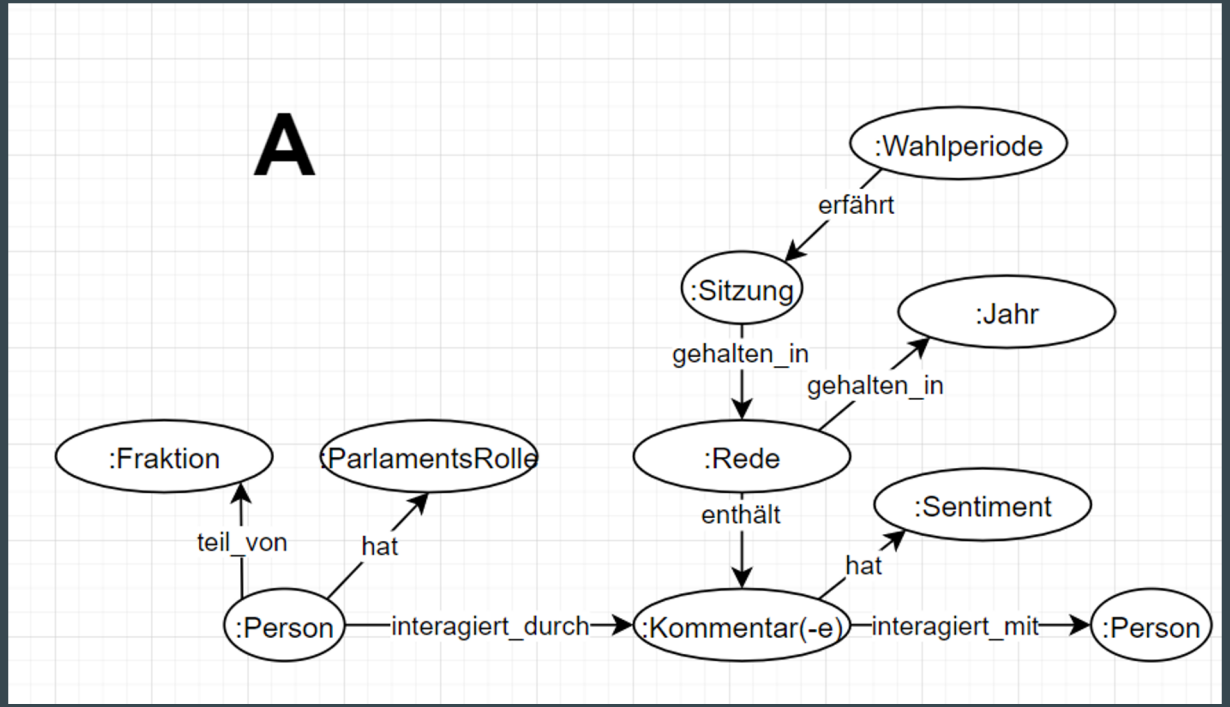
1. Angebotene Datenbasis
   a. Vorhandenes Datenbankmodell im ERM-Format Gruppe 2 / 3
2. Allgemeines:
   a. These des Moduls Der Ton im Bundestag ist rauer geworden
   b. Konkrete Aufgabenstellung: Interaktion zwischen Abgeordneten
3. Konkrete Fragestellungen durch Gruppe 7
   a. Fragestellung 1
   b. Fragestellung 2
   c. Fragestellung 3
   d. ...

Ausgangssituation 2:

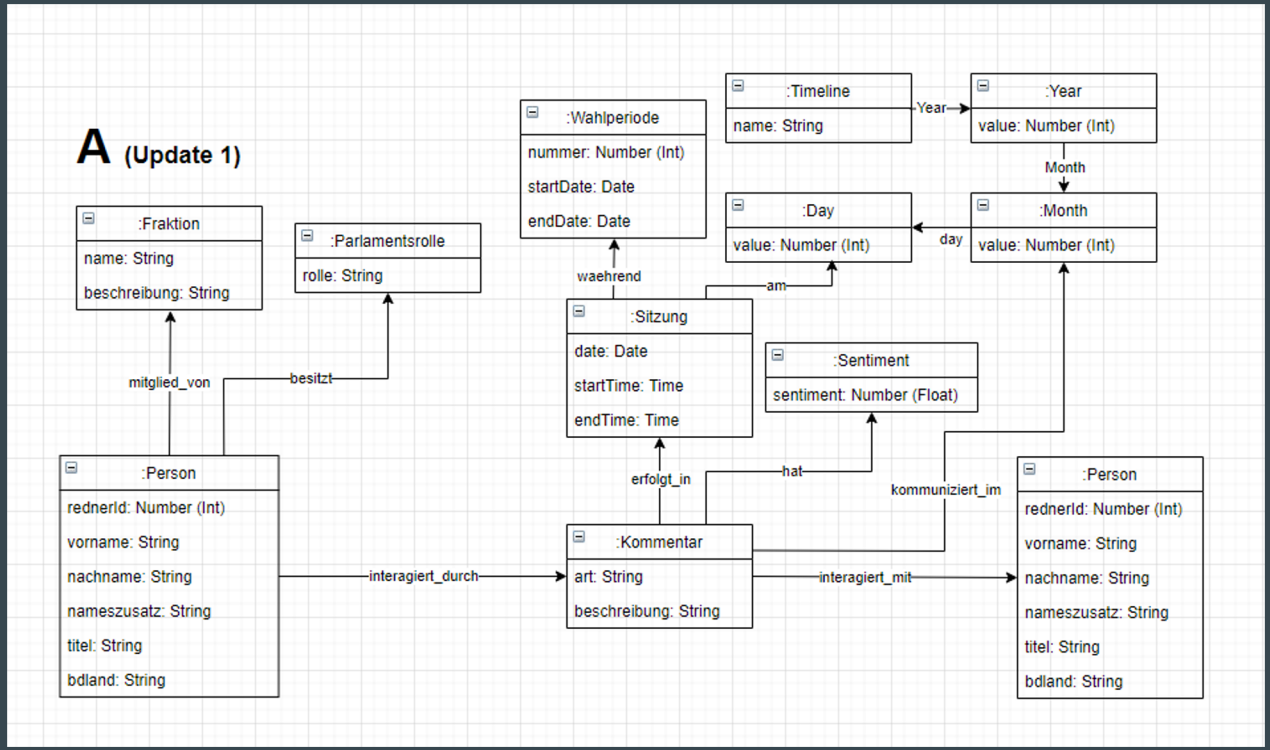
Datenbankschema Gruppe 2 (Format: ERM)



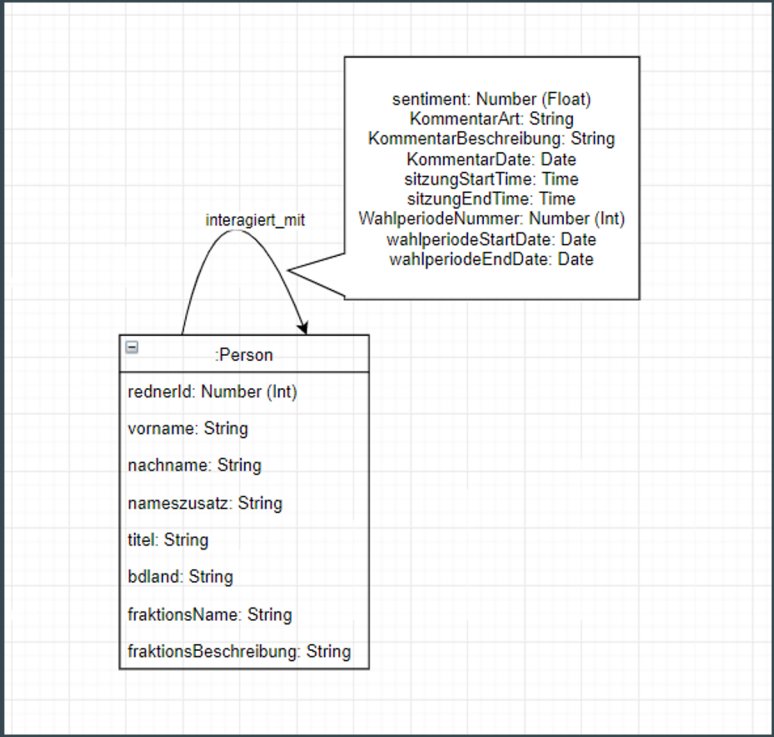
ER Modell Entwurf: Entwurf 1



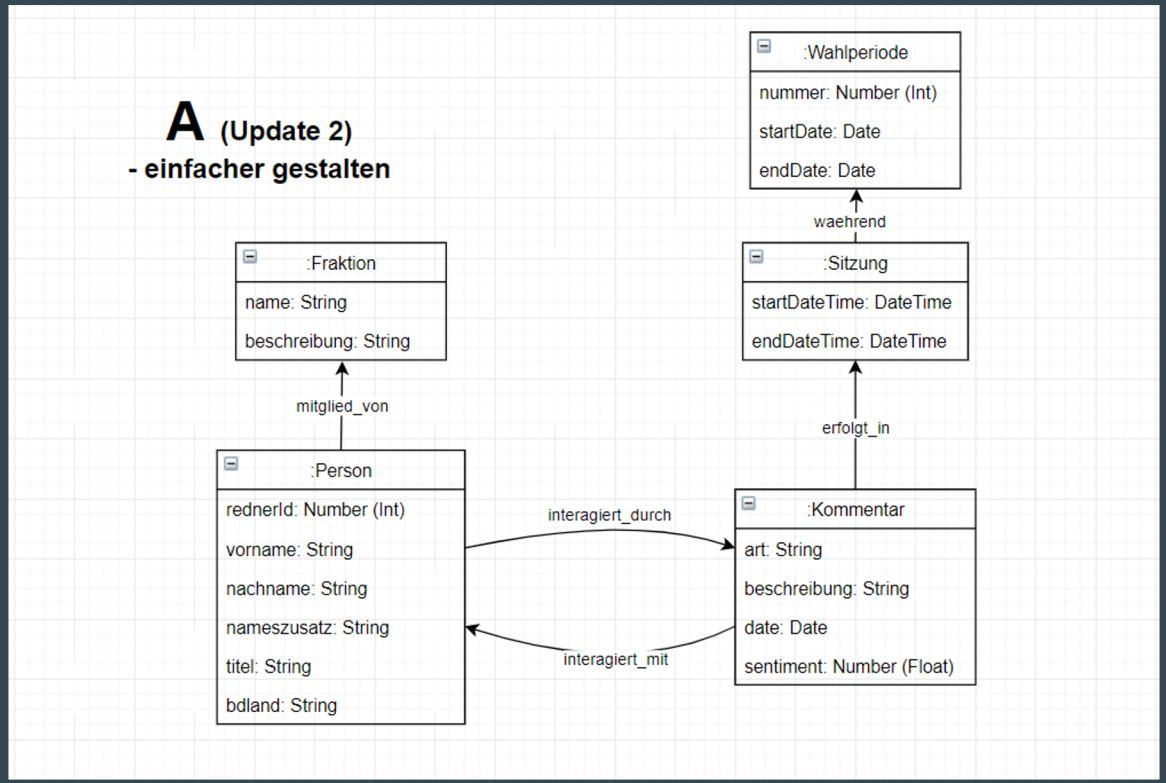
ER Modell Entwurf: Entwurf 2



ER Modell Entwurf: 3



ER Modell Entwurf: Aktueller Stand



Aufgaben

1. Theorie: aktuelles Datenbankschema weiter verbessern
2. Praxis: Verarbeiten und Übertragen von Daten zwischen Datenbanken
    a. Graph-Datenbank physisch einrichten
    b. Script implementieren, welches auf Anfrage
        i. Verbindung zu den Datenbanken (MongoDB, Neo4j) aufbaut
        ii. Daten einzeln von MongoDB einliest
        iii. Daten umformt: Nodes, Properties, Relationships, Labels hinzufügt/entfernt
        iv. Daten in die Graphdatenbank Neo4j schreibt

Zeitplan

- Wann kann mit Umsetzung begonnen werden?
  - Jetzt
  - Besser: Sobald erste Daten-Ausschnitte von Gruppe 3 vorhanden sind

- Wann ist unsere Graphdatenbank voll analyse-fähig?
  - Fertigstellungstermin von Gruppe 3 + 1 Woche

Quellen

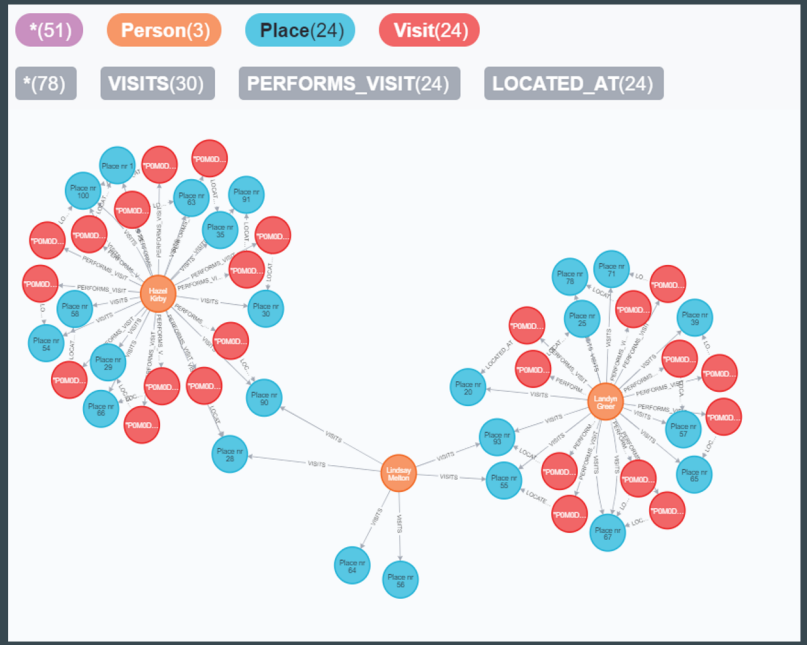
- I. Robinson, J. Webber, E. Eifrem: “Graph Databases” (2013), O'Reilly Media
- Talk Max De Marzi: https://neo4j.com/blog/secret-sauce-neo4j-modeling-graphconnect/
- Talk Peter Olson: https://vimeo.com/79399404
- https://neo4j.com/developer/kb/understanding-data-on-disk/
- https://www.slideshare.net/thobe/an-overview-of-neo4j-internals
- https://neo4j.com/blog/neo4j-genericvague-relationship-names/?_ga=2.1966065.1196726123.1604256717-1400350775.1602246988&_gac=1.195750366.1602247016.CjwKCAjwlID8BRAFEiwAnUoK1Z5e4V0ZdNrtxmmYoreFpB0QhNer6UlY2zi9RwBoffRiZqI3v4sG2hoCavAQAvD_BwE

Vielen Dank für die Aufmerksamkeit

Fragen?

Appendix

Appendix: Beispiel



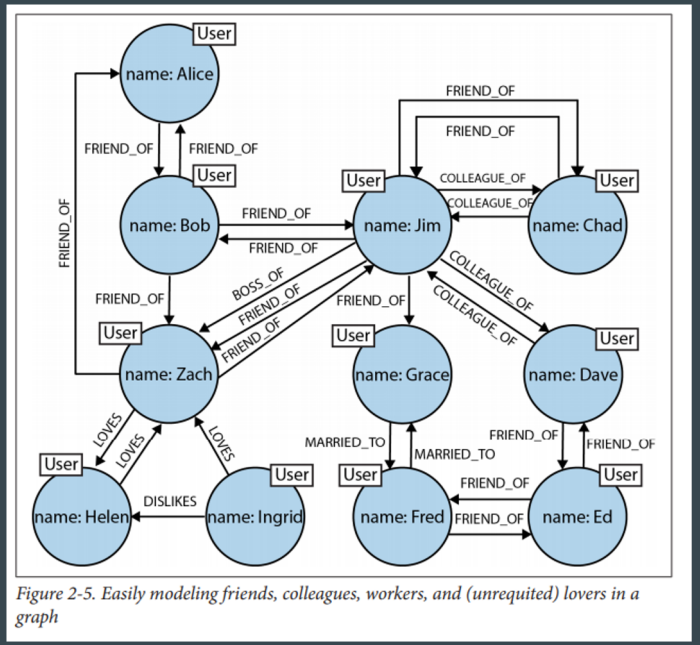


Figure 2-5. Easily modeling friends, colleagues, workers, and (unrequited) lovers in a graph

I. Robinson, J. Webber, E. Eifrem: "Graph Databases" (2013), O'Reilly Media

Appendix: Storage 1

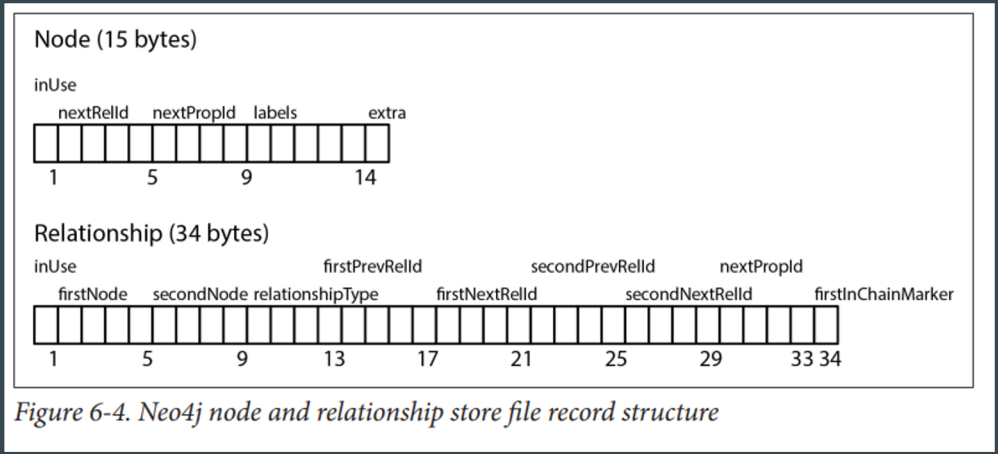


Figure 6-4. Neo4j node and relationship store file record structure

Scenario #1 – Initial status

- Node count: 4M nodes
  - Each node has 3 properties (12M properties total)
- Relationship count: 2M relationships
  - Each relationship has 1 property (2M properties total)

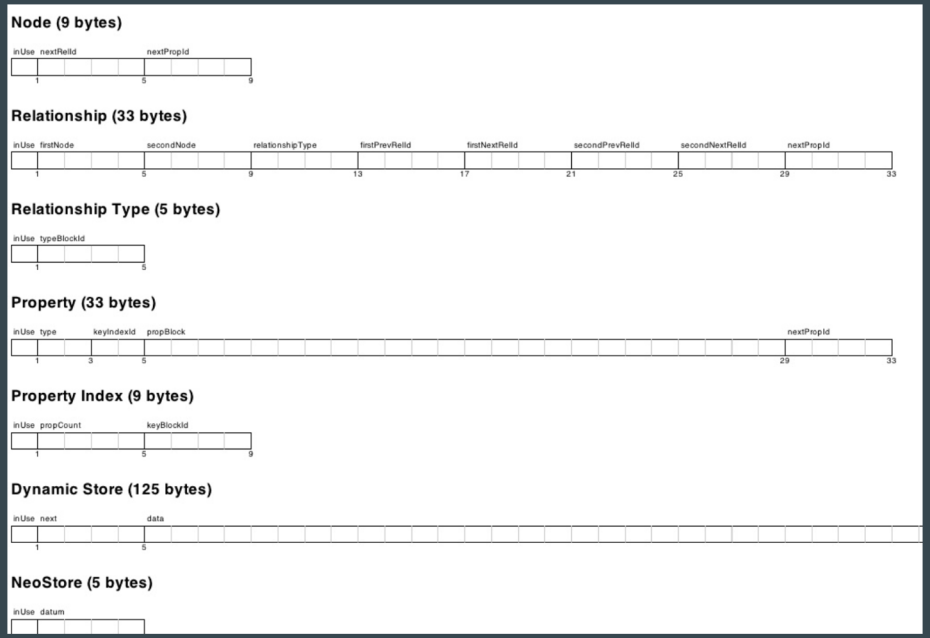
This is translated to the following size on disk:

- Nodes: 4.000.000x15B = 60.000.000B (60MB)
- Relationships: 2.000.000x34B = 68.000.000B (68MB)
- Properties: 14.000.000x41B = 574.000.000B (574MB)
- TOTAL: **703MB**

Quelle: https://neo4j.com/developer/kb/understanding-data-on-disk/

Wie werden nun Relationships nach Type, In- und Outgoing gruppiert?
→ Einführung einer neuen Speichereinheit zwischen Node und Relationship
→ Quellen:
https://github.com/neo4j/neo4j/commit/366d30928d1b1590eba5daef92116ecc15aa36b1
https://neo4j.com/blog/the-neo4j-2-1-0-milestone-1-release-import-and-dense-nodes/

Appendix: Storage 2



Quelle: https://www.slideshare.net/thobe/an-overview-of-neo4j-internals

Appendix: Storage 3

Store File	Record size	Contents
neostore.nodestore.db	15 B	Nodes
neostore.relationshipstore.db	34 B	Relationships
neostore.propertystore.db	41 B	Properties for nodes and relatior
neostore.propertystore.db.strings	128 B	Values of string properties
neostore.propertystore.db.arrays	128 B	Values of array properties
Indexed Property	1/3 * AVG(X)	Each index entry is approximately

Quelle: https://neo4j.com/developer/kb/understanding-data-on-disk/

Appendix: Beispiel - Verarbeitungszeit im Sozialen Netzwerk

- Studie: Analyse eines Social Network
- Quelle
  - Buch: Neo4j in Action
  - Autoren: J. Partner, A. Vukotic, N. Watt
- Ziel: Finde Freunde-von-Freunden bis zu einer max-death von 5
- Datenbank
  - 1.000.000 people
  - Each person: approximately 50 friends

Table 2-1. Finding extended friends in a relational database versus efficient finding in Neo4j

Depth	RDBMS execution time(s)	Neo4j execution time(s)	Records returned
2	0.016	0.01	~2500
3	30.267	0.168	~110,000
4	1543.505	1.359	~600,000
5	Unfinished	2.132	~800,000

I. Robinson, J. Webber, E. Eifrem: “Graph Databases” (2013), O’Reilly Media

Appendix: Relationship-Label

```
Using a generic relationship type and then filtering by end node label
50%ile: 6.0    75%ile: 6.0    99%ile: 402.60999999999825

Using a generic relationship type and then filtering by relationship property
50%ile: 21.0   75%ile: 22.0   99%ile: 504.85999999999785

Using a generic relationship type and then filtering by end node label
50%ile: 4.0    75%ile: 4.0    99%ile: 145.65999999999931

Using a specific relationship type
50%ile: 0.0    75%ile: 1.0    99%ile: 25.749999999999872
```

First-row: *end node property

Quelle: Blog-Post of Mark Needham on the Neo4J-Website: https://neo4j.com/blog/neo4j-genericvague-relationship-names/?_ga=2.1966065.1196726123.1604256717-1400350775.1602246988&_gac=1.195750366.1602247016.CjwKCAjwlID8BRAFEiwAnUoK1Z5e4V0ZdNrtxmmYoreFpB0QhNer6UlY2zi9RwBoffRiZqI3v4sG2hoCavAQAvD_BwE

Appendix: Email-Problem

- Fehlende Informationen über die Email selbst

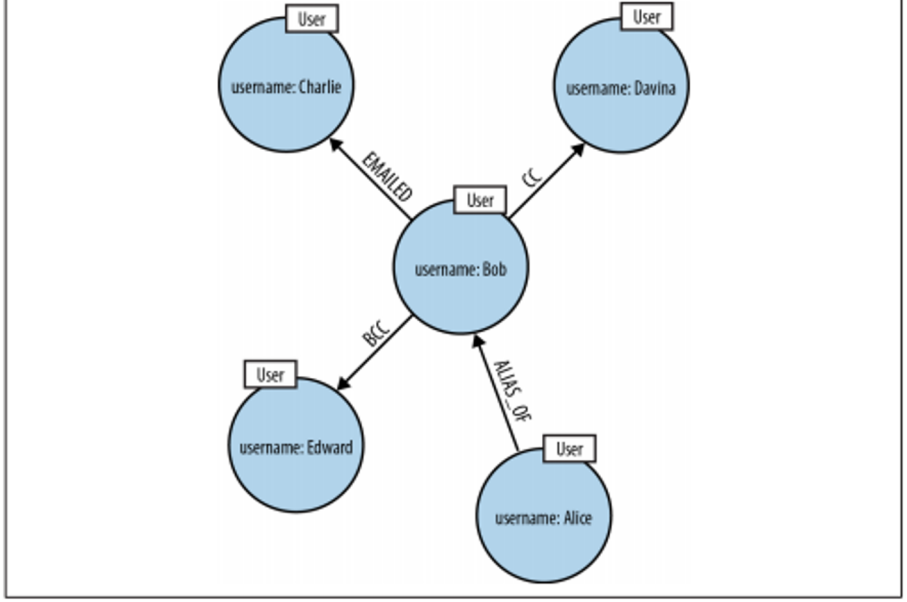


Figure 3-8. Missing email node leads to lost information

I. Robinson, J. Webber, E. Eifrem: “Graph Databases” (2013), O'Reilly Media

Appendix: Datenbank-Driver

MongoDB:

- Besonders: Node.js [~7.7k] (> Java) > Python > C# [~3.6k]
- https://docs.mongodb.com/drivers/

Neo4J:

- Besonders: (Java [~2.1k] >) C# - JS > Python [~1.1k]
- https://neo4j.com/developer/language-guides/