

Graph Datenbanken - Neue & aktuelle Neo4j Tools für den schnellen Einstieg in Neo4j

Marco De Luca

Neo4j Field Engineering

21. November 2022

Folien, Daten und Datenmodelle finden Sie auf: github.com/luzidl/it-novum-webinar_112022

Agenda

- Neo4j Graph Datenbank - Was oft unklar ist
- Knowledge Graph als Grundlage für weiter Mehrwerte
- Vorgehensweise - Aufbau eines Knowledge Graphen (KG)
- Tools für den Einstieg in Neo4j
 - Daten Modellierung
 - Daten laden
 - Data analysieren
- Demo: Neo4j Workspace*

Neo4j Graph Datenbank - Was oft unklar ist

Neo4j ist eine **NORMALE**
Datenbank (DB)



- Vergleichbar mit anderen DBs wie Postgres, MySQL / MariaDB, Oracle, HANA DB, etc.
- Daten sicher speichern, verwalten und abfragen
- Backup + Recovery
- Hoch skalierbar



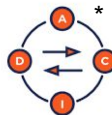
Neo4j Graph Datenbank - Was oft unklar ist

Neo4j ist eine NORMALE Datenbank (DB)



- Vergleichbar mit anderen DBs wie Postgres, MySQL / MariaDB, Oracle, HANA DB, etc.
- Daten sicher speichern, verwalten und abfragen
- Backup + Recovery
- Hoch skalierbar

Neo4j ist “ACID compliant” und somit transaktionssicher



- Auf deutsch AKID konform
- Daten werden auf Transaktionsebene sicher gespeichert ohne Verlust bei Ausfällen
- **AKID** - Atomarität, Konsistenz, Isolation und Dauerhaftigkeit

Neo4j Graph Datenbank - Was oft unklar ist

Neo4j ist eine NORMALE Datenbank (DB)



- Vergleichbar mit anderen DBs wie Postgres, MySQL / MariaDB, Oracle, HANA DB, etc.
- Daten sicher speichern, verwalten und abfragen
- Backup + PiT Recovery
- Hoch skalierbar durch Clustering

Neo4j ist “ACID compliant” und somit transaktionssicher



- Auf deutsch AKID konform
- Daten werden auf Transaktionsebene sicher gespeichert ohne Verlust bei Ausfällen
- **AKID - Atomarität, Konsistenz, Isolation und Dauerhaftigkeit**

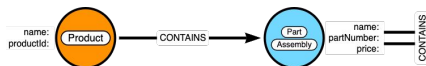
Neo4j wird zu 80%+ für OLTP Workloads genutzt



- > 80% der Kunden nutzen Neo4j in erster Instanz als normale Datenbank
- Daten speichern, ändern und löschen
- Mit gleicher oder oft besserer Performance als bei relationale DBs

Aber was macht eine native Graph-Datenbank anders?

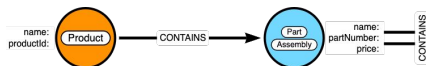
Die Speicherung der Daten



- Daten werden verknüpft gespeichert
- Daten liegen auf der Platte nah “beieinander”
Stichwort: “Index-free Adjacency”
- Sie speichert **Knoten** und **Verbindungen** statt Zeilen und Spalten
- **Semantik** ist abbildbar!

Aber was macht eine native Graph-Datenbank anders?

Die Speicherung der Daten



- Daten werden verknüpft gespeichert
- Daten liegen auf der Platte nah “beieinander”
Stichwort: “Index-free Adjacency”
- Sie speichert **Knoten** und **Verbindungen** statt Zeilen und Spalten
- **Semantik** ist abbildbar!

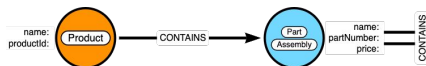
Die Abfragen der Daten

`(:Product) -[:CONTAINS]->(:Part)`

- Cypher Query Language statt SQL (ISO -> GQL)
- Einfacher, weniger Code-Zeilen, besser lesbar
- Abfragen bis in Tiefen von 100+ Hops, vergleichbar von SQL Joins über 100+ Tabellen!

Aber was macht eine native Graph-Datenbank anders?

Die Speicherung der Daten



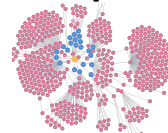
- Daten werden verknüpft gespeichert
- Daten liegen auf der Platte nah “beieinander”
Stichwort: “Index-free Adjacency”
- Sie speichert **Knoten** und **Verbindungen** statt Zeilen und Spalten
- **Semantik** ist abbildbar!

Die Abfragen der Daten

`(:Product) - [:CONTAINS] -> (:Part)`

- Cypher Query Language statt SQL (ISO -> GQL)
- Einfacher, weniger Code-Zeilen, besser lesbar
- Abfragen bis in Tiefen von 100+ Hops, vergleichbar von SQL Joins über 100+ Tabellen!

Komplexe Daten-netzwerke speichern



- Speicherung und Analyse von komplexen Zusammenhängen
- Auswertung von Daten, die bis jetzt nie/unzureichend miteinander verknüpft wurden
- Erweiterbar mit Data Science Algorithmen

Was und Warum - Knowledge Graphen



Vorgehensweise - Aufbau eines Knowledge Graphen (grob)

1. Prüfung des Problems - Ist Graph die Lösung?
2. Fachbereich + IT Personal - Anforderungen niederschreiben
3. Aus den Anforderungen ein Datenmodell ableiten



Vorbereitung

Vorgehensweise - Aufbau eines Knowledge Graphen (grob)

1. Prüfung des Problems - Ist Graph die Lösung?
2. Fachbereich + IT Personal - Anforderungen niederschreiben
3. Aus den Anforderungen ein Datenmodell ableiten



4. Daten präparieren und laden
5. Fragen/Anforderungen versuchen zu beantworten
6. Datenmodell überarbeiten, für bessere/schnellere Antworten



Vorbereitung



Rekursive Arbeit

Vorgehensweise - Aufbau eines Knowledge Graphen (grob)

1. Prüfung des Problems - Ist Graph die Lösung?
2. Fachbereich + IT Personal - Anforderungen niederschreiben
3. Aus den Anforderungen ein Datenmodell ableiten



4. Daten präparieren und laden
5. Fragen/Anforderungen versuchen zu beantworten
6. Datenmodell überarbeiten, für bessere/schnellere Antworten



7. ETL Pipeline aufbauen
8. Skripte, Monitoring und ggf. Applikation entwickeln/einkaufen



Vorbereitung



Rekursive Arbeit



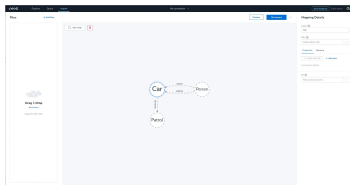
Erster Pilot

Tools für den Einstieg

Daten modellieren



arrows.app



[Neo4j
Workspace*](https://neo4j.com/workspace)

Daten laden

Daten speichern

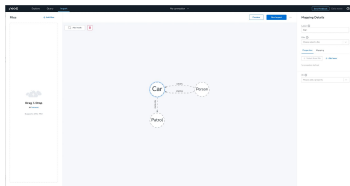
Daten analysieren

Tools für den Einstieg

Daten modellieren



arrows.app

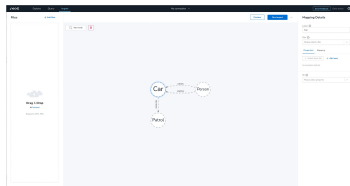


[Neo4j
Workspace*](#)

Daten laden

- > Cypher CSV Load
- > APOC Library
- > neo4j admin import
- > ETL Tool
- > Python API

[Daten laden, etc.](#)



[Neo4j
Workspace*](#)

Daten speichern

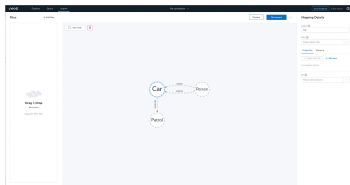
Daten analysieren

Tools für den Einstieg

Daten modellieren



arrows.app

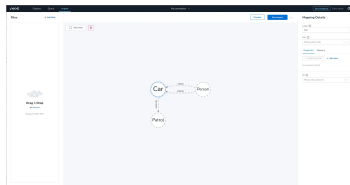


Neo4j
Workspace*

Daten laden

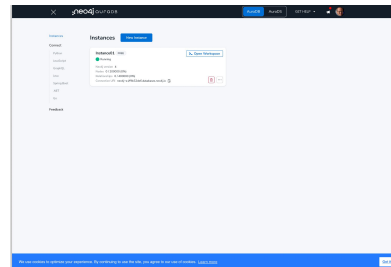
- > Cypher CSV Load
- > APOC Library
- > neo4j admin import
- > ETL Tool
- > Python API

Daten laden, etc.

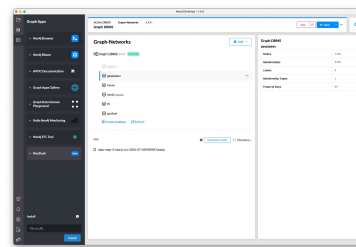


Neo4j
Workspace*

Daten speichern



Neo4j Aura



Neo4j Desktop

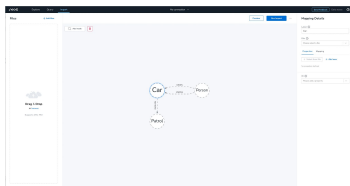
Daten analysieren

Tools für den Einstieg

Daten modellieren



[arrows.app](#)

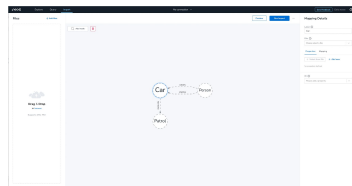


[Neo4j
Workspace*](#)

Daten laden

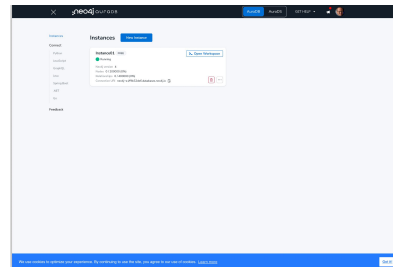
- > Cypher CSV Load
- > APOC Library
- > neo4j admin import
- > ETL Tool
- > Python API

[Daten laden, etc.](#)

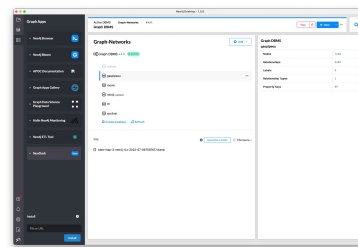


[Neo4j
Workspace*](#)

Daten speichern



[Neo4j Aura](#)

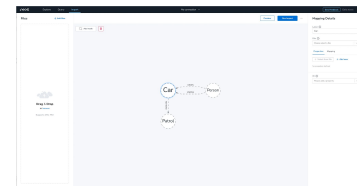


[Neo4j Desktop](#)

Daten analysieren



[Neo4j Bloom](#)



[Neo4j
Workspace*](#)



DEMO TIME!

[NEO4j Workspace
\(Preview\)](#)

[NEO4j Aura \(SaaS\)](#)

[NEO4j Desktop](#)

Demo Daten, Folien, Datenmodelle, etc. auf Github unter:

github.com/luzidl/it-novum-webinar_112022

Danke!

Contact us at
marco.deluca@neo4j.com
heiko.schoenfelder@neo4j.com

