

Graphdatenbanken für alle

Warum PostgreSQL mit Apache AGE eine echte Alternative zu Neo4j ist

Matthias Grömmner
Head of Consulting



PROUD CONTRIBUTOR TO





Get your tickets!



 Vienna

 4 September 2025

 pgday.at





Open Alliance

For PostgreSQL Education

Scan for Updates



oapg-edu.org

- Independent industry certification for PostgreSQL

- First launch at pgconf.de 2025

- For more information visit CYBERTEC booth

AGENDA

1. Motivation

Warum Graphdatenbanken?

2. Grundlagen

Graphdatenbank vs. relationales DBMS

3. Apache AGE & openCypher

Vorstellung & Beispiel der Nutzung

4. Fazit / Q&A

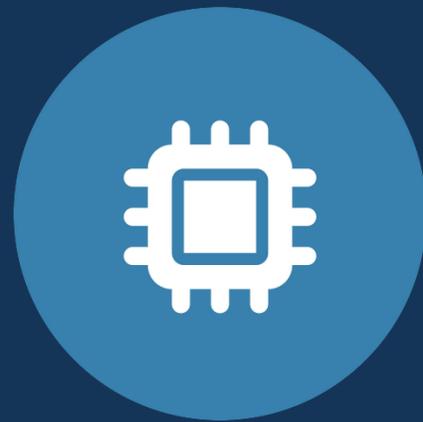
Graphdatenbank vs. relationales DBMS



1. Warum Graphdatenbanken?



Modellierung komplexer und zahlreicher Beziehungen



Performante Abfragen bei vernetzten Daten



Datenspeicherung auf Basis von JSON-Dokumenten (schemaless)



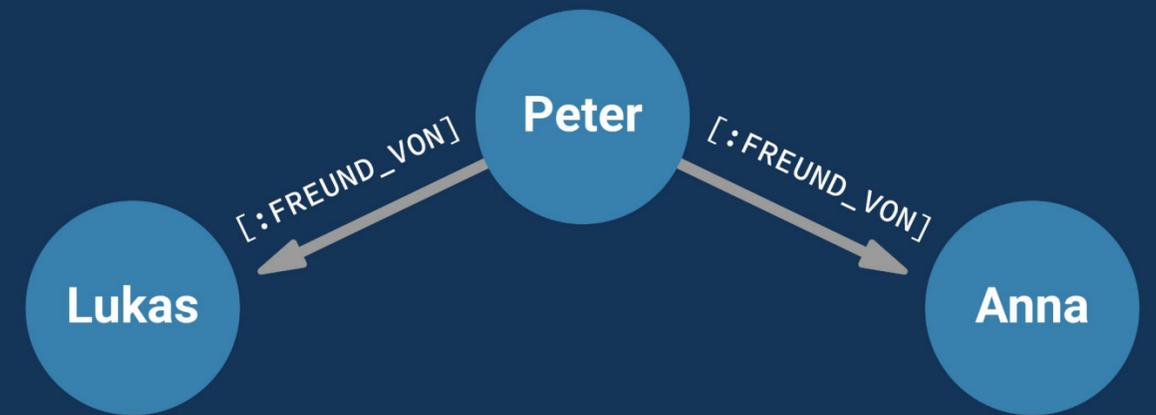
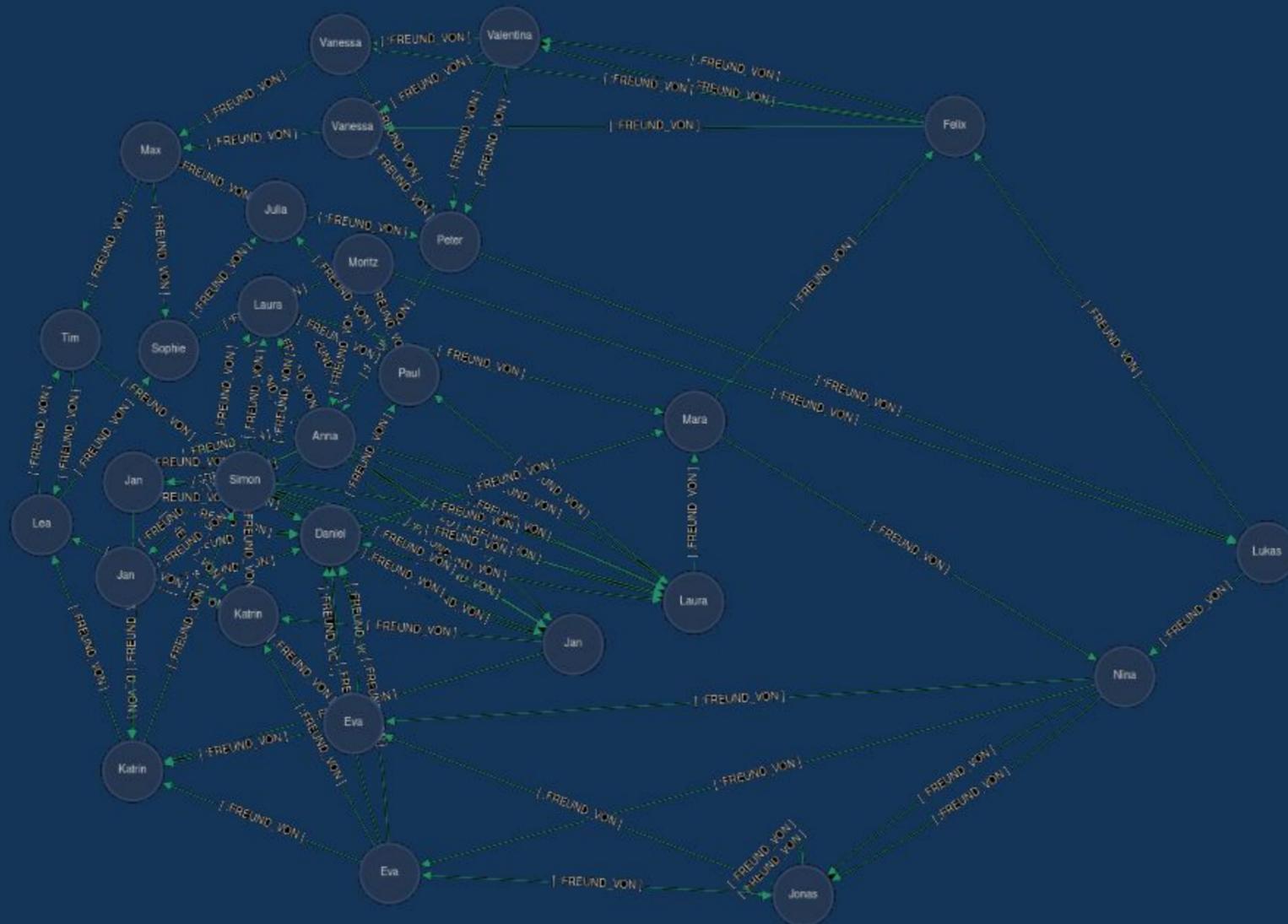
Einfache Visualisierung und Analyse



Native Algorithmen für Graphnetzwerke



1. Warum Graphdatenbanken?



2. Grundlagen

- Knoten entsprechen einem Objekt mit Eigenschaften
Üblicherweise JSON
- Kanten entsprechen einer Beziehung mit Eigenschaften
Üblicherweise JSON
- Zusammenspiel ermöglicht das Erzeugen von Graphen/Netzwerken
- Hierarchien und m:n-Beziehungen problemlos möglich



2. Grundlagen

- Tiefe oder rekursive Beziehungsketten
 - Identifiziere alle Freunde von Valentina, welche zum Swiss PGDAY gehen wollen
- Dynamische Beziehungen (agile Datenmodelle)
- Komplexe Netzwerk-Analysen
 - Supply-Chain-Netzwerk / Empfehlungssysteme
 - Soziale Netzwerke / Knowledge-Graphen
- Kausalketten
 - Fraud-Detection / AI-Systeme



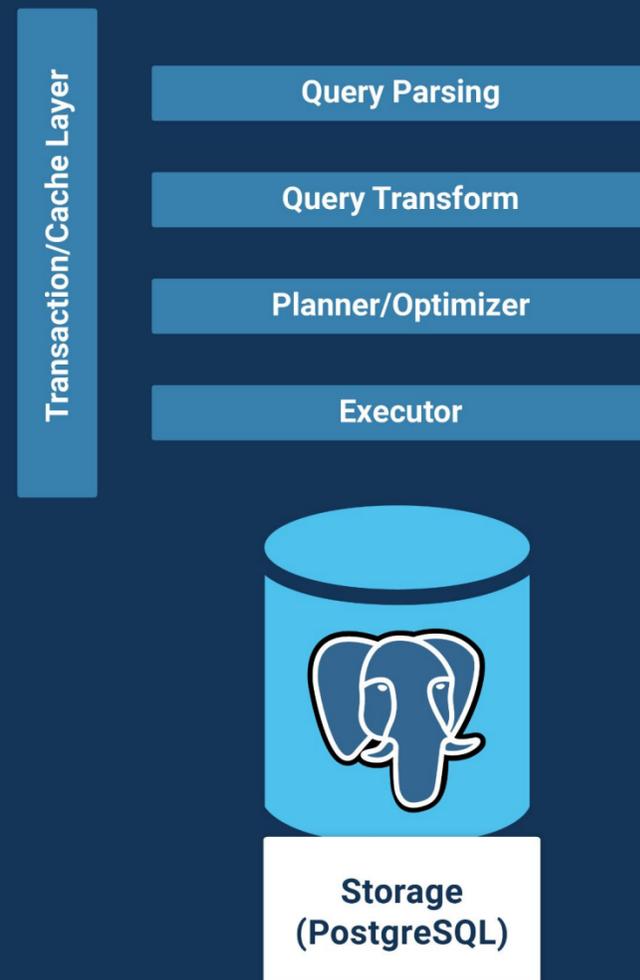
3. Apache AGE & openCypher

- PostgreSQL-Extension - Apache License 2.0
- Basiert auf der Idee von AgensGraph
- Kombination relationale Datenbank und Graphmodell
- Kombination von SQL mit OpenCypher
- Aktuelle Version 1.5.0 (max PG 16)
- Letzter Commit: vor 2 Wochen
- <https://age.apache.org>



3. Apache AGE & openCypher

AGE Architecture



1

Parses Cypher queries embedded in cypher function calls. Here we implement the grammar for openCypher.

2

Transforms a Cypher query into a Query tree that will be attached as a subquery node.

3

Understands some graph operations and produces plan nodes that are related to graph operations.

4

Executes plan nodes that are related to graph operations.

5

Cypher queries work with Postgres existing fully transactional system (ACID).



3. Apache AGE & openCypher

- Deklarative Abfragesprache für Graphdatenbanken
- OpenSource (Entwickelt von Neo4j)
- Grundkonzept
 - Daten sind Knoten (Nodes) und Kanten (Relationships)
 - Knoten und Kanten verfügen über Eigenschaften (Properties) (Key-Value-Paare)
 - Knoten haben Labels, Kanten haben Typen (Types)



3. Apache AGE & openCypher

- openCypher-Syntax:
 - CREATE (OBJEKT:LABEL {properties})
CREATE (a:Person {name: 'Valentina'})
 - MERGE erzeugt ein Objekt (falls es nicht existiert), sonst UPDATE
MERGE (a:Person {name: 'Valentina'})
 - MATCH KNOTEN-KANTE->KNOTEN RETURN
MATCH (V)-[R]-(V2) RETURN V,R,V2
 - WHERE KNOTEN.properties =|<|> WERT
MATCH (V)-[R]-(V2) WHERE V.name = 'Valentina' RETURN V,R,V2



3. Apache AGE & openCypher

- openCypher-Syntax:

- `MATCH (OBJEKT:LABEL {properties}) SET OBJEKT.properties = value`
`MATCH (p:Person {name: 'Valentina'}) SET p.age = 35`

- `MATCH (OBJEKT:LABEL {properties})`
`MATCH (p:Person {name: 'Valentina'}) DETACH DELETE p`

- flexible Pfadlängen

`MATCH (a:Person)-[:FRIEND_OF*1..3]->(b:Person)`

`MATCH path = (a:Person {name: 'Valentina' })-[:FREUND_VON*3]->(b:Person)`
`RETURN path`



3. Apache AGE & openCypher

- Apache Age-Syntax:

- `SELECT * FROM cypher ('GRAPH_NAME', $$ OPENCYPHER $$) AS (Return Types)`

```
SELECT * from cypher('social_graph', $$ MATCH (V)-[R]-(V2) RETURN V,R,V2
$$) AS (V agtype, R agtype, V2 agtype);
```

```
SELECT * FROM cypher('social_graph', $$
  MATCH path = (a:Person {name: 'Valentina' })-
  [:FREUND_VON*3]->(b:Person)
  RETURN path
$$) AS (path agtype);
```



3. Apache AGE & openCypher

- Age Viewer:
 - Tool zur Visualisierung und Interaktion mit Graphdaten
 - Unterstützt die Ausführung von Cypher-Queries direkt aus dem Viewer
 - Ermöglicht die interaktive Analyse von Graphstrukturen durch selektierbare Knoten und Kanten
 - Webapp auf Basis von Node.js (Für Server oder Container-Umgebungen)



3. Apache AGE & openCypher

- Algorithmen:
 - DBMS entscheidet über verfügbare Algorithmen, nicht openCypher
 - Beispiel Neo4J:
 - Kürzeste Wege:
 - Dijkstra
 - A* Shortest Path
 - Minimum Weight Spanning Tree
 - Similarity algorithms
 - Node Similarity
 - K-Nearest Neighbor



3. Apache AGE & openCypher

- Algorithmen:
 - Weitere Typen von üblichen Algorithmen:
 - Centrality algorithms
 - Community detection algorithms
 - DAG Algorithmen (in gerichteten Graphen)
 - Apache AGE unterstützt derzeit:
 - Die Suche mit flexiblen Pfadlängen
 - Degree Centrality (Centrality Algorithms)
 - Weitere Algorithmen sind über Python-Packages oder via Cypher nutzbar



3. Apache AGE & openCypher

- Treiber/SDK:
 - Go
 - Java
 - NodeJS
 - Python
 - Rust



3. Apache AGE & openCypher

- Praxisbeispiel



Fazit

Q&A

