

Graph-Datenbanken

- basiert auf Beziehungen
- Schemalos und flexibel
- Joins müssen nicht teuer berechnet werden
- Sharding aufwändig

Knoten

- repräsentieren Objekte
- eindeutiger Identifier
- können mit einer Menge von Typen und Attributen verknüpft werden

Kanten

- Beziehungen zwischen Knoten
- eindeutiger Identifier
- genau einen Typ
- können mit einer Menge von Attributen verknüpft werden
- haben Start-, Endknoten und eine Richtung

Attribut

- Key-Value-Paare
- Key ist ein String
- Value ist ein atomarer Wert eines primitiven Datentyps

Beispiele

```
MATCH (tom:Person {name:"Tom Hanks"})-[r:ACTED_IN]->(m)
WHERE m.released >= 1900 AND m.released < 2000
RETURN tom,m;
```

```
CREATE (TheMatrix:Movie {title:'The Matrix', released:1999, Tagline:'Welcome to the Real World'})
```

```
CREATE (Keanu)-[:ACTED_IN {roles:['Neo']}]->(TheMatrix)
```

Allgemeine Syntax

- Knoten: „(“ variable? (“:“ label)* properties? „)“
- label: alle müssen erfüllt werden
- properties: {k:v, k:v, ...}, alle müssen erfüllt werden
- Kanten: („-“ | „<-“) („[„ kantenDetails „]“)? („-“ | „->“)
- Kantendetails: - Variablenname - :label - Liste von Typen (getrennt durch „|“; nur ein Typ muss erfüllt werden) - Pfadlänge - Beispiele: *, *4, *2..4, *..6 - properties – siehe Knoten

Revision #2

Created 24 July 2024 19:31:04 by Kuchenmampfer

Updated 25 July 2024 13:06:44 by Kuchenmampfer