

Einführung und Theorie

Contents

1	Einführung und Theorie	2
2	Inhalt	2
3	Vorbemerkung	2
4	Herkunft und Idee	3
5	Graphtypen und Strukturen	5
5.1	Graph und nicht Graph	5
5.2	Labeled Property Graph	6
5.3	Einfache Graphtypen	6
5.4	Verbundene und nichtverbundene Graphen	9
5.5	Gewichtete und ungewichtete Graphen	11
5.6	Gerichtete und ungerichtete Graphen	13
6	Installation und Start	15
7	Zusammenfassung	16

1 Einführung und Theorie

2 Inhalt

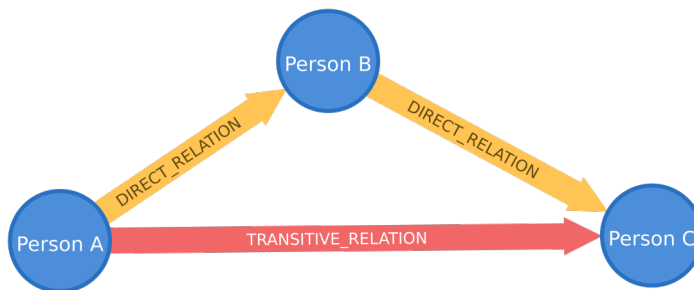
{:.no_toc}

- Will be replaced with the ToC, excluding the “Contents” header {:.toc}

[Dieser Abschnitt ist in Bearbeitung]

3 Vorbemerkung

Dieser Abschnitt gibt eine kurze Einführung zu Graphen.¹ Gegenüber relationalen Datenbanken, in denen die Daten in Tabellen abgelegt und verknüpft sind, werden in Graphen Knoten für die Speicherung der Daten verwendet und diese dann mit Kanten in Relation gesetzt. Aber auch in relationalen Datenbanken kann man natürlich Daten verknüpfen. Wo liegt also der eigentliche Vorteil.



(Quelle: Kuczera)

Der wirkliche Mehrwert bei Graphdatenbanken ergibt sich aus gerichteten und transitiven Beziehungen. Hat A eine direkte Kante zu B und B eine direkte Kante zu C dann ist A zwar nicht direkt sondern transitiv mit C verbunden. Gerade wenn ein Graph sehr viele verschiedene solcher transitiven

¹Dieser Abschnitt beruht in Teilen auf den Kapiteln “Introduction” und “Graph Theory and Concepts” des Buches Graph Algorithms: Practical Examples in Apache Spark and Neo4j von Mark Needham und Amy E. Hodler, O’Reilly Media 2019 (<https://www.oreilly.com/library/view/graph-algorithms/9781492047674/>), S. xiii-xv und S. 1-26.

Verbindungen hat lassen sich Muster und Verbindungen identifizieren die in relationalen Modellen oft unentdeckt bleiben. Darüber hinaus bietet der Graph eine optimale Ausgangslage für anschließende Netzwerkanalyse des gesamten Graphen oder von ausgewählten Subgraphen. War es in den digitalen Geisteswissenschaften bis vor einigen Jahren noch höchste Priorität überhaupt digitale Forschungsdaten bereitzustellen ist es heute die Herausforderung Daten in ihrem Kontext zu erfassen. Hierfür lassen sich Graphdatenbanken hervorragend nutzen.

4 Herkunft und Idee

Graphen gehen zurück auf die erste Hälfte des 18. Jahrhunderts. 1736 löste Leonhard Euler das “Königsberger Brückenproblem”, das fragte ob es möglich sei, die 4 durch den Fluß getrennten Stadtbereiche, die über 7 Brücken verbunden waren jeweils einmal zu besuchen, ohne eine der Brücken zweimal zu nutzen.²

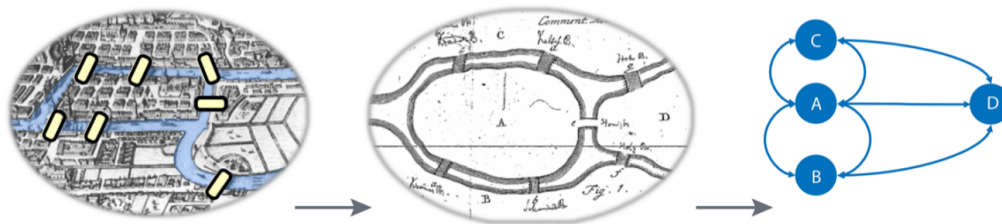


Figure 1: Eulersche Brücken

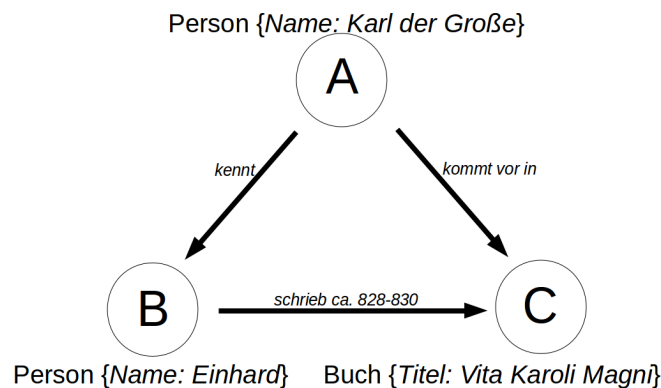
Quelle: <https://www.oreilly.com/library/view/graph-algorithms/9781492047674/>, S. 2, Bild 1-1.

Euler kam zu der Einsicht, dass nur die Verbindungen relevant sind und legte damit den Grundstein für die Graphtheorie und ihre Mathematik. Er zeigte, dass das Brückenproblem nicht lösbar war, da zu jedem Stadtbereich eine ungerade Anzahl von Brücken führte, es dürfe aber nur zwei Ufer mit einer ungeraden Anzahl geben.³

²Vgl. https://de.wikipedia.org/wiki/K%C3%B6nigsberger_Br%C3%BCckenproblem.

³Vgl. https://de.wikipedia.org/wiki/K%C3%B6nigsberger_Br%C3%BCckenproblem.

Das folgende Beispiel modelliert einen Zusammenhang aus dem Frühmittelalter und erklärt kurz die Grundprinzipien in der geisteswissenschaftlichen Domäne.



(Quelle: Kuczera)

Der Beispielgraph zeigt oben einen Knoten (engl. Nodes) vom Typ Person mit der Eigenschaft Name. Diese hat den Wert “Karl der Große”. Links unten ist ein weiterer Knoten vom Typ Person mit dem Namen “Einhard”. Rechts unten ist ein Knoten vom Typ Buch und dem Titel “Vita Karoli Magni” abgebildet. Die Kanten (engl. Edges) geben an, dass Karl der Große Einhard kannte, Einhard ca. 828-830 das Buch “Vita Karoli Magni” schrieb und Karl der Große in dem Buch vorkommt.

Knoten und Kanten können also noch zusätzliche Eigenschaften besitzen, in denen weitere Informationen gespeichert sind. Diese Eigenschaften sind spezifisch für die jeweiligen Knotentypen. So sieht man in der Abbildung, dass die beiden Knoten vom Typ Person jeweils noch die Eigenschaft Namen haben, deren Wert dann die Namen der Person angibt, während der Knoten vom Typ Buch die Eigenschaft Titel trägt, in dem der Titel des Buches abgespeichert wird.

5 Graphtypen und Strukturen

5.1 Graph und nicht Graph

Auch wenn Graphen ihren Ursprung in der Mathematik haben so sind sie doch ein pragmatisches Werkzeug um Informationen zu modellieren und zu analysieren.

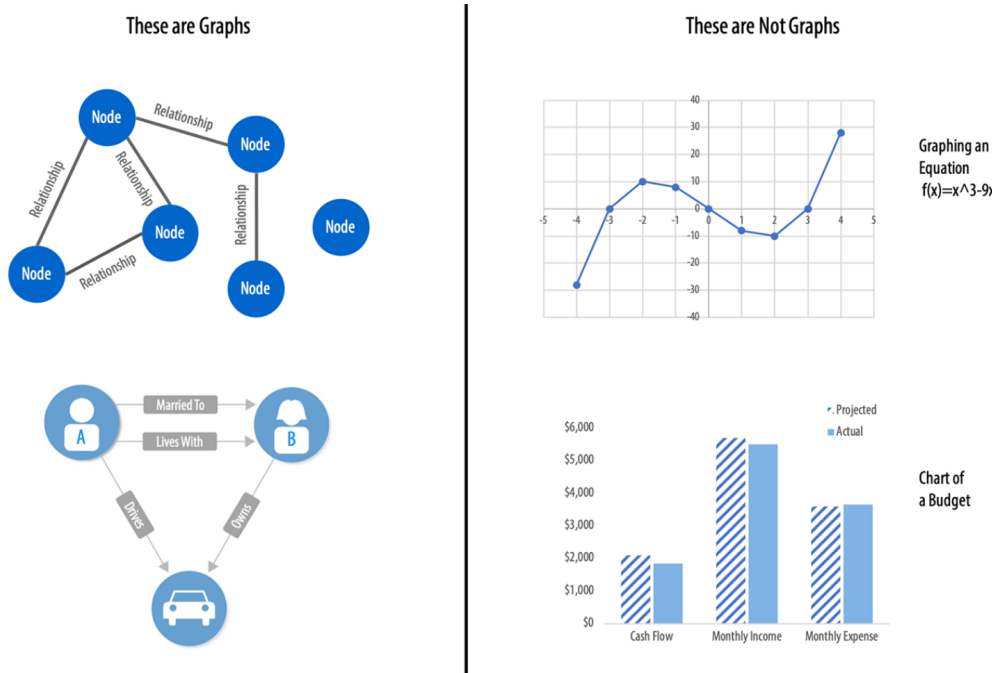


Figure 2: Graphen und nicht Graphen

Quelle: <https://www.oreilly.com/library/view/graph-algorithms/9781492047674/>, S. 3, Bild 1-2

In diesem Buch geht es nicht um Graphen im Sinne von Lösung von Gleichungen wie im Bild rechts. Es geht um Informationen, die durch Knoten und Kanten dargestellt werden, wie links im Bild.

5.2 Labeled Property Graph

Einen Graphen in denen die Knoten und Kanten Typen (Labels) und Eigenschaften (Properties) besitzen nennt man “Labeled Property Graph” (LPG).

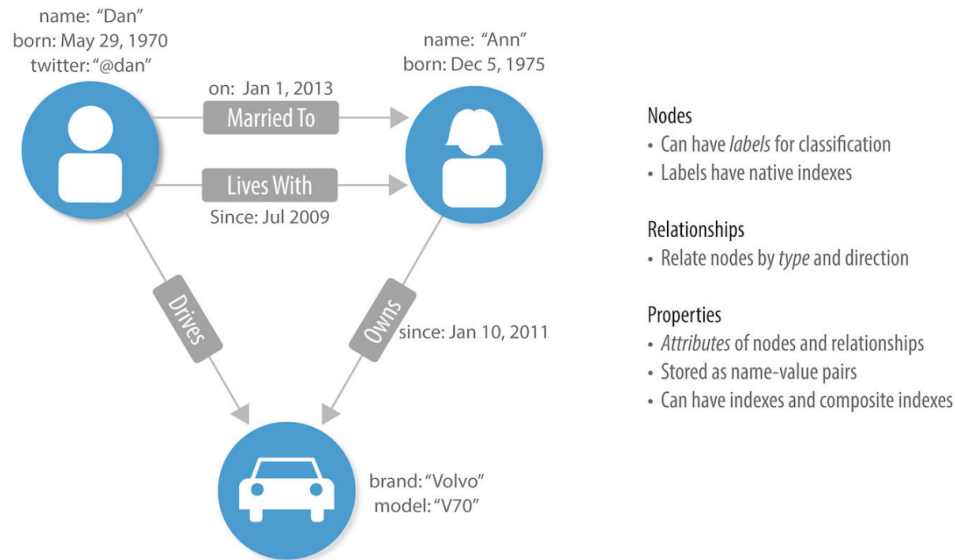


Figure 3: Beispiel für einen Labeled-Property-Graphen

Quelle: <https://www.oreilly.com/library/view/graph-algorithms/9781492047674/>, S. 16, Bild 2-1

Der o.a. Graph sagt beispielsweise aus, dass der Knoten A mit dem Label Person und den zugehörigen Properties (name: 'Dan', born etc.) mit Person B zusammenlebt und verheiratet ist. Person B gehört gleichzeitig ein Auto (mit der Property Marke: 'Volvo'), dass von Person A gefahren wird. Hier werden in einfachen Schritten Informationen aus der “realen Welt” abgebildet. In den nächsten Abschnitten geht es nun um die Konzepte dahinter.

5.3 Einfache Graphtypen

In der folgenden Abbildung sind verschiedene Graphtypen abgebildet. Im Beispiel links ist ein einfacher Graph dargestellt, in dem jedes Knotenpaar nur eine Verbindung haben kann.

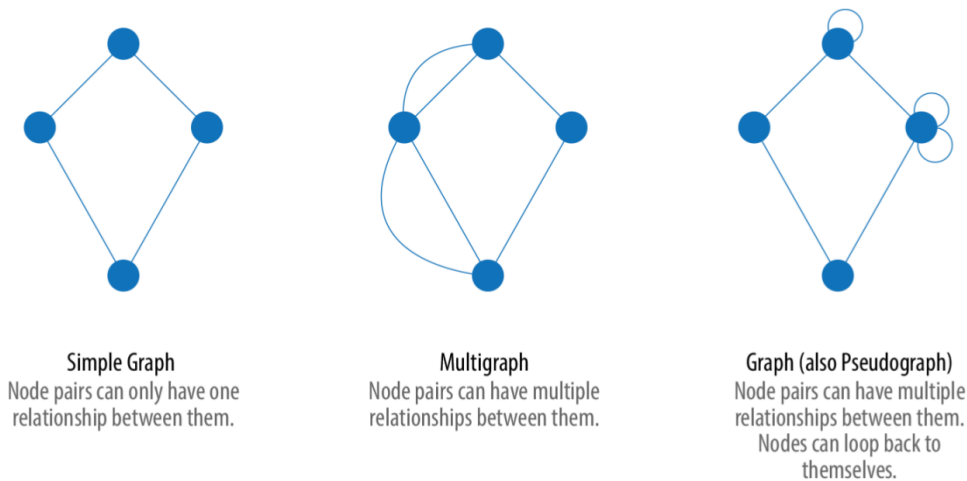
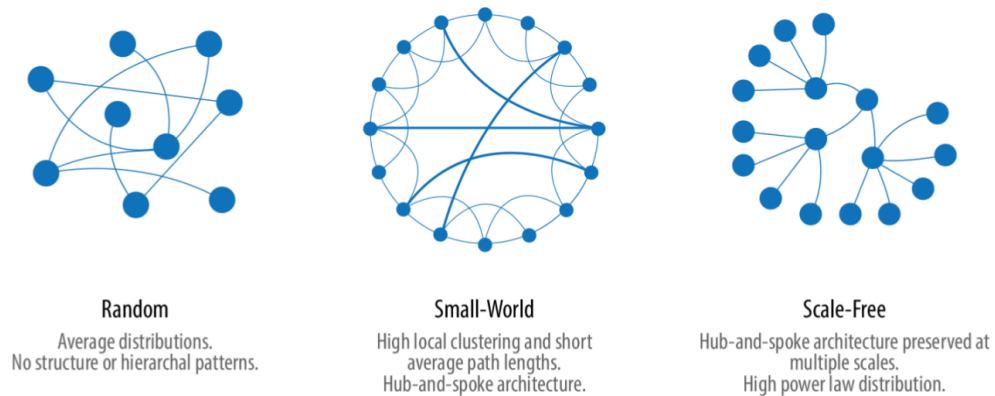


Figure 4: Einfache Graphtypen

Quelle: <https://www.oreilly.com/library/view/graph-algorithms/9781492047674/>, S. 17, Bild 2-2

Im mittleren Beispiel sind mehrere Verbindungen zwischen Knotenpaaren möglich. Das rechte Beispiel ergänzt dich noch um die Möglichkeit, Verbindungen von Knoten wieder zu ihnen selbst zurück zu ermöglichen.

Die folgende Abbildung zeigt links ein Beispiel für einen zufälligen Graphen, aus dem sich keine hierarchischen oder strukturellen Informationen ablesen



lassen.

Quelle: <https://www.oreilly.com/library/view/graph-algorithms/9781492047674/>, S. 17, Bild 2-3.

In der mittleren Abbildung ein Small-World-Graph abgebildet, der lokal stark geclustert ist und eine geringe durchschnittliche Pfadlänge hat. Die rechte Abbildung zeigt ein skalenfreies Netzwerk, wie es das Word-Wide-Web eines ist.

In der folgenden Tabelle werden charakteristische Graphtypen gegenübergestellt.

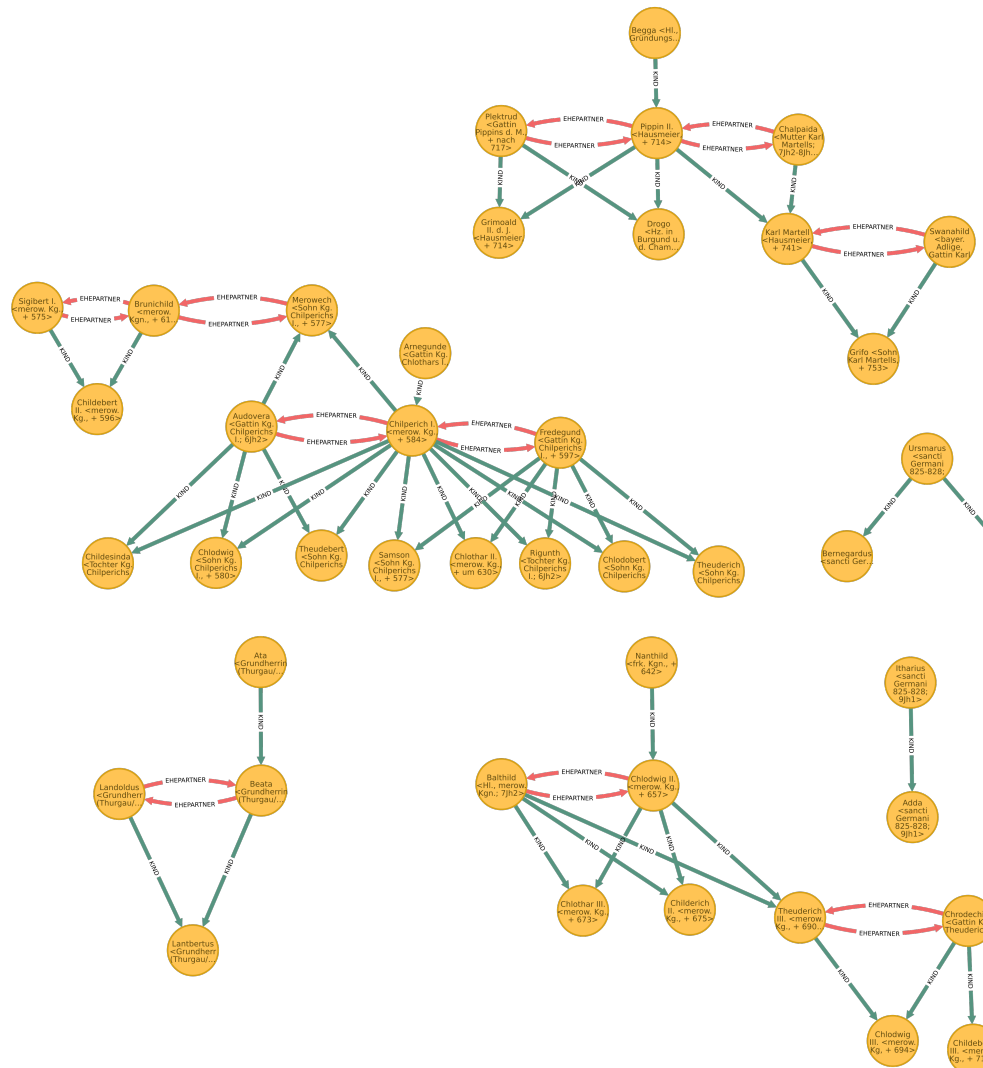
Graphattribute	Unterschiede	Überlegungen zum Algorithms
Verbundene und nichtverbundene Graphen	Gibt es einen Pfad (Länge egal) zwischen zwei beliebigen Knoten eines Graphen	Knoteninseln können zu unvorhergesehenem Verhalten, wie Abbruch oder Auslassen von Knoteninseln
Gewichtete und ungewichtete Graphen	Gibt es (domänen-spezifische) Werte für Wichtungen an Knoten und Kanten	Werte machen den Graphen reicher an Informationen; Viele Algorithmen benötigen gewichtende Werte
Gerichtete und ungerichtete Graphen	Ist eine Verbindung hierarchisch oder gegenseitig	Die Richtung einer Kante sind für viele Auswertungen von Relevanz
Zyklische und nicht zyklische Graphen	In zyklischen Graphen können Pfade wieder zum Startknoten zurückkehren	Zyklische Graphen sind weit verbreitet

Graphattribute	Unterschiede	Überlegungen zum Algorithmus
Graphdichte	Das Verhältnis von Kanten zu Knotenzahl	Extrem dichte oder extrem dünne Graphen können die Analyse erschweren. Ggf. können Änderungen an der Modellierung helfen, sofern es die Domäne zulässt
Monopartite, bipartite, and k-partite Graphen	Gibt es einen, zwei oder mehrere Knotentypen	Mehr Knotentypen erleichtern in den Geisteswissenschaften die Modellierung, Graphalgorithmen arbeiten oft aber nur mit einem Knotentyp

Quelle: <https://www.oreilly.com/library/view/graph-algorithms/9781492047674/>, S. 18-19.

5.4 Verbundene und nichtverbundene Graphen

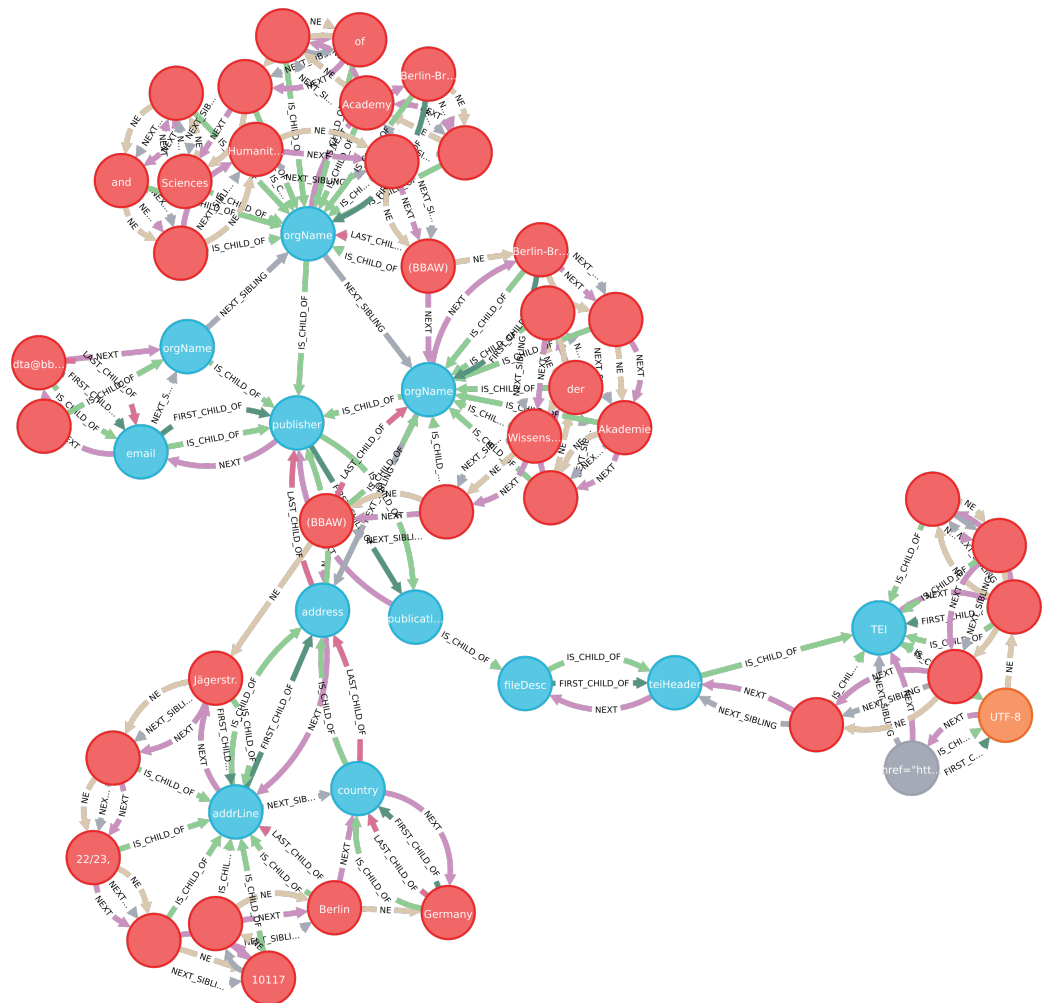
Sind in einem Graphen mehrere Gruppen von verbundenen Knoten vorhanden und es gibt zwischen den Gruppen keine Verbindungen, handelt es sich um



nichtverbundene Graphen.

Das Beispiel zeigt einen Graphen mit Personen und ihren Verwandtschaftsbeziehungen. Es gibt mehrere Stammbäume, die nicht miteinander verbunden sind.

Das folgende Bild zeigt einen verbundenen Graphen, bei dem alle Knoten mindestens eine Kante haben und jeder Knoten jeden anderen über einen Pfad erre



ichen kann.

Hier wird ein Ausschnitt aus einer Graphdatenbank gezeigt, in die eine XML-Text-Datei als XML-Baum importiert wurde. Da es sich bei jeder XML-Datei um einen Baum handelt und jeder XML-Knoten Teil dieses Baumes ist, gibt es keine nichtverbundenen Teile.

5.5 Gewichtete und ungewichtete Graphen

In ungewichteten Graphen besitzen die Kanten keinen Wert zu Wichtung.

Quelle: Kuczera

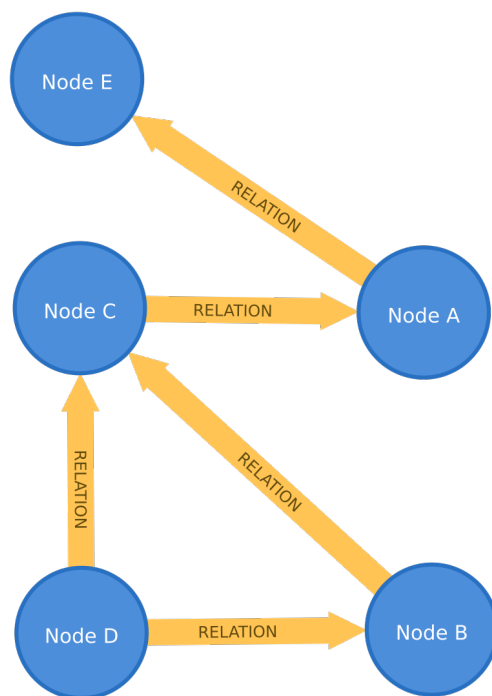


Figure 5: Ungewichteter Graph

Im Graphen werden Beziehungen zwischen Personen abgebildet. Über die Intensität der Beziehungen wird nichts gesagt. In der folgenden Abbildung wurde den Beziehungen zwischen jeweils ein Wert zwischen 0 und 100 zugeordnet.

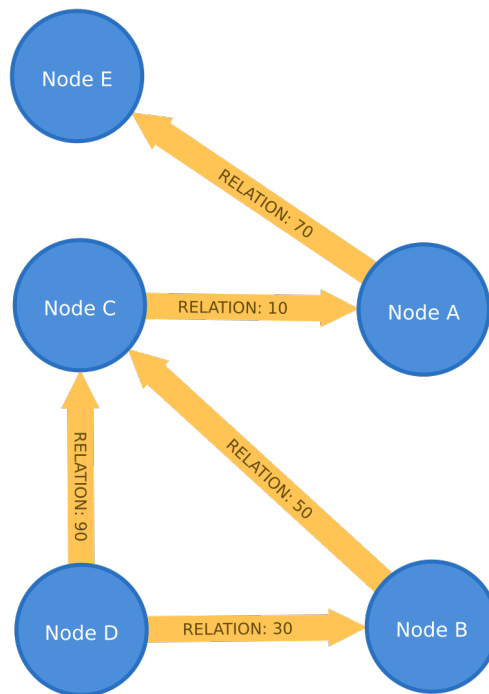


Figure 6: Gewichteter Graph

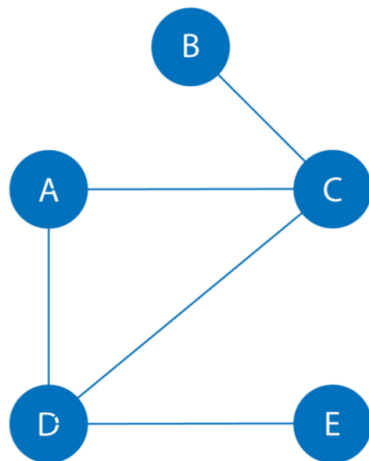
Quelle: Kuczera

Werden solche Angaben ergänzt, werden die Informationen im Graph wertvoller. Werte für Wichtungen können beispielsweise Entfernungen, Kosten, Kapazitäten oder auch domänenspezifische Priorisierungen sein.

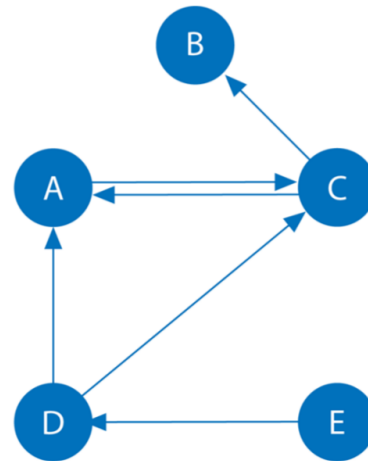
5.6 Gerichtete und ungerichtete Graphen

In einem ungerichteten Graphen geht eine Beziehungen immer in beide Richtungen (beispielsweise EHEPARTNER_VON). In einem gerichteten Graphen haben Beziehungen eine Richtung. Betrachtet man einen Knoten gibt es eingehende Kanten, die auf den Knoten zeigen und ausgehende Kanten, die von

dem Knoten ausgehen. Mit der Angabe von Richtung wird eine zusätzliche Informationsdimension hinzugefügt. Angenommen im linken Beispiel der folgenden Abbildung würden die Knoten Personen und die ungerichtete Kante ihre gegenseitige Freundschaft darstellen. Damit ergibt sich, dass Person A mit Person B befreundet ist.



Undirected

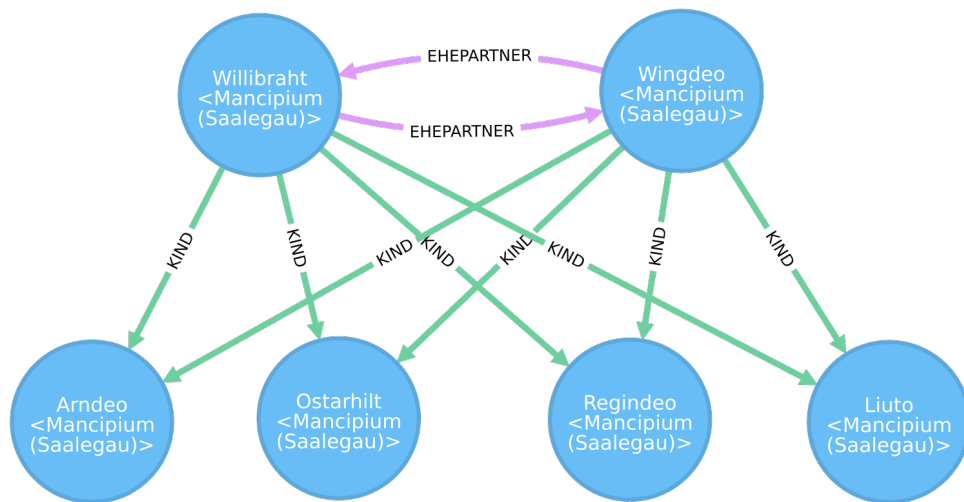


Directed

(Quelle: <https://www.oreilly.com/library/view/graph-algorithms/9781492047674/>, S. 21, Bild 2-7)

Erweitern wir das Beispiel für die rechte Abbildung, so dass die Freundschaftskanten gerichtet sind, so wird es einerseits möglich, zu zeigen, dass Person D für Person A freundschaftliche Gefühle hegt, die A aber nicht erwidert. Nehmen wir noch eine Wichtung der Freundschaftskanten hinzu, könnte man modellieren, dass die freundschaftlichen Gefühle von Person C gegenüber A mit 0,8 gewichtet sind, umgekehrt die Zuneigung von Person A zu Person C aber nur mit 0,3 gewichtet sind.

Auch bei der Modellierung von Verwandtschaft kommen gerichtete und ungerichtete Verbindungen vor.



(Quelle: Kuczera)

In der Graphdatenbank neo4j müssen Kanten eine Richtung haben. Ungerichtete Beziehungen kann man mit zwei Kanten vom gleichen Typ aber unterschiedlicher Richtung zwischen einem Knotenpaar modellieren. Im Verwandtschaftsbeispiel sind die zwei Personenknoten der Eltern mit zwei EHEPARTNER-Kanten unterschiedlicher Richtung verbunden. Es handelt sich um eine ungerichtete Beziehung, da es rechtlich nicht möglich ist, dass eine Person mit einer anderen Person verheiratet umgekehrt aber nicht. Für die Eltern-Kind-Beziehung reicht eine Kante, da diese Beziehung hierarchisch ist.

6 Installation und Start

Informationen zur Installation von neo4j finden Sie auf den Dokumentationsseiten unter <https://neo4j.com/docs/operations-manual/current/installation/>. Für den normalen Nutzer empfiehlt sich die Installation von neo4j-Desktop. Unter <https://neo4j.com/blog/this-week-in-neo4j-getting-started-with-neo4j-desktop-and-browser-graphileon-personal-edition-intuitive-detections-research-with-neo4j/?ref=twitter#features-1> finden sich Videos, in denen die Installation von neo4j-Desktop und erste Schritte im neo4j-Browser erklärt werden.

7 Zusammenfassung

In diesem Kapitel wurde kurz auf die Ursprünge von Graphen eingegangen und verschiedene Typen von Graphen vorgestellt. Festzuhalten ist, dass es sich in den Digitalen Geisteswissenschaften in der Regel um Mischformen der vorgestellten Graphtypen handelt. Je nach Domäne werden Graphen hoher Dichte, mit vielen Knotentypen oder auch hierarchischen Strukturen benötigt. Modelle in den Digitalen Geisteswissenschaften haben meist mehrere Knotentypen und viele Kantentypen. Dies macht wiederum die direkte Anwendung von Graph-Algorithmen schwierig, so dass Graph-Transformationen notwendig werden können.