

Ihre Stammdaten sind ein Graph: Sind Sie bereit?

Karen Lopez, InfoAdvisors
März 2015

InfoAdvisors

www.datamodel.com
karenlopez@Infoadvisors.com
[@datachick](#)

*Dieses Whitepaper wird von Neo Technology, Inc. gesponsert.
Neo4j ist eine eingetragene Marke von Neo Technology, Inc*

DATENGESCHICHTE

Ihre Daten wollen eine Geschichte erzählen. Sie haben kein *Datenproblem*, sondern eine *Datengeschichte*.

Ihre Stammdaten sind ein Graph – Sind Sie bereit?

MDM Heute

Ihr MDM-Programm nutzt wahrscheinlich die gleiche Datenbanktechnologie wie Ihre anderen Anwendungen: ein ausgereiftes, voll optimiertes relationales Datenbankmanagementsystem (Relational Database Management System, RDBMS). Da Sie schon seit etlichen Jahren damit arbeiten und einen Großteil Ihrer Daten darin abspeichern, sind Sie mit relationalen Datenbanken bestens vertraut. Es macht also Sinn, auch die Stammdaten relational zu speichern und zu verwalten. **Aber was geschieht, wenn die Beziehungen innerhalb Ihrer Stammdaten – sowohl bekannte als auch unbekannte – ebenfalls verwaltet werden müssen?** Genau diese Daten werden in Zukunft in der Daten- und Unternehmensanalyse an Bedeutung gewinnen – vor allem wenn es darum geht, sich einen Wettbewerbsvorsprung zu verschaffen.

Gewöhnlich umfasst MDM Daten von Kunden, Produkten, Buchhaltung, Lieferanten, Partnern und andere gemeinsam nutzbare Daten innerhalb eines Unternehmens. In einigen Fällen werden auch Referenzdaten in die Stammdaten miteinbezogen.

In letzter Zeit tauchten eine ganze Reihe neuer Ansätze im Bereich Data Management auf. MDM bildet hier keine Ausnahme. Die unterschiedlichen Definitionen fokussieren sich entweder auf die rein operationale Verwaltung von Daten in einer einzelnen Quelle oder Datei. Oder aber es handelt sich um einen eher konzeptuellen Ansatz für die Verwaltung ganzer Datenbestände, um den Zugriff auf einen einzelnen Dienst oder eine Anwendung zu ermöglichen. In beiden Fällen liegt der Fokus jedoch auf den Daten an sich als Entscheidungskriterium.

Bislang taten sich Datenspezialisten schwer, mit traditionellen Ansätzen für Stammdaten auf Veränderungen des Geschäfts und des Geschäftsumfelds zu reagieren. Stammdaten sind gewöhnlich einer Vielzahl von Personen im Unternehmen zugänglich. Ein unflexibler Umgang mit ihnen wirkt sich daher über alle Bereiche hinweg negativ aus. Systemarchitekturen zielten bisher darauf aus, Daten mit einer einzigen, allumfassenden Lösung zu managen. Dass dies auf lange Sicht nicht die „wahre“ Lösung ist, ist längst bekannt.

Die Zukunft von MDM

MDM-Programme, die versuchen Daten an einem einzigen Ort zu speichern, kommen mit der Realität moderner Informationstechnologien nicht zurecht. Die meisten Unternehmen verwenden Anwendungen wie CRM-Systeme, Work-Management-Systeme, Kreditoren- und Debitorenbuchhaltungssysteme oder PoS-Systeme. Aufgrund dieser Herangehensweise ist es nicht immer möglich, alle Stammdaten an einem einzigen Ort zu speichern. Selbst wenn ein CRM-System vorhanden ist, finden sich Kundeninformationen gewöhnlich auf unterschiedlichen Systemen wieder. Das gilt auch für Produkt- und Buchhaltungsdaten.

Die erfolgreichsten Programme versuchen nicht, einen einzigen physischen Ort für alle Daten zu finden. Vielmehr bieten sie Standards, Tools und Dienste, mit denen sich Unternehmensdaten einheitlich abbilden lassen. Dank Technologien, die am besten zur ihrer Datengeschichte passen, können manche Daten weiterhin an einem Ort gespeichert werden. Andere Daten werden sich mit hoher Wahrscheinlichkeit in verschiedenen physikalischen Systemen wiederfinden, was auf die vermehrte Nutzung von Anwendungspaketen sowie auf gestiegene Bearbeitungsanforderungen in Bezug auf Performance und geographische Verteilung zurückzuführen ist. Erst wenn wir ein Verständnis für unsere Umgebung gewonnen haben, können wir Lösungen entwickeln, die auf diesen Anforderungen aufbauen.

Die erfolgreichsten Programme versuchen nicht, einen einzigen physischen Ort für alle Daten eines Konzepts zu finden. Vielmehr bieten sie Standards, Tools und Dienste, die für eine einheitliche Abbildung der Unternehmensdaten erforderlich sind.

In Zukunft wird MDM seinen Nutzen sowohl aus Daten als auch den Datenbeziehungen ableiten. Im Vordergrund steht eine schlüssige und aussagekräftige Darstellung von Stammdaten. In vielen Fällen wird es möglich sein, Daten an einem Ort zusammenzufassen und damit Abfragegeschwindigkeit und Datenintegration zu optimieren. Graphdatenbanken bieten genau diese Art von Daten-/Leistungsintegration. Dieses Whitepaper zeigt, weshalb Ihre Stammdaten ein Graph sind und warum Graphdatenbanken wie Neo4j die beste Lösung für Stammdaten darstellen.

Bessere Einblicke dank Datenbeziehungen

Denkt man an Datenbeziehungen, fallen einem meist zuerst Fremdschlüssel ein, also jene Funktionen von relationalen Datenbanken, mit deren Hilfe sich Datenreihen miteinander verbinden lassen. Dies ist jedoch nur eine Art von Beziehung. **Tatsächlich sind diese Fremdschlüssel technisch betrachtet lediglich Einschränkungen unserer Daten.** Die Fremdschlüsselbeziehung zwischen TRANSAKTION und TRANSAKTIONSPOSITIONEN stellt eine Einschränkung dar, die garantiert, dass alle Transaktionspositionen einer Transaktion zugeordnet sind. Die Datengeschichten mit dem meisten Mehrwert konzentrieren sich auf weit komplexere und vielschichtige Fragen:

- Welche Produkte werden üblicherweise zusammen erworben?
- Welche Produkte werden regelmäßig vom selben Kunden und den Mitgliedern seines Haushalts erworben? Wie verhält es sich mit Kundengruppen?
- Welche Produkte und Dienstleistungen sollten dem Kunden einzeln bzw. als Paketlösung angeboten werden?
- Welche Faktoren spielen bei Kaufentscheidungen eine Rolle?

Auf diese Fragen werden wir später noch genauer eingehen.

Graphen: Theorie und Praxis

Graphen sind Datenstrukturen, die Daten und ihre Beziehungen beschreiben. Die bekanntesten Graphformen sind Netzwerke und Hierarchien. Graphen enthalten Knoten und deren Beziehungen zueinander. Sowohl die Knoten als auch die Beziehungen können über Properties/Eigenschaften verfügen. Zudem lassen sich Knoten auch mit Labels versehen.

Netzwerke

Ein Netzwerk besteht aus einer Reihe von Knoten, die durch Beziehungen miteinander verbunden sind.

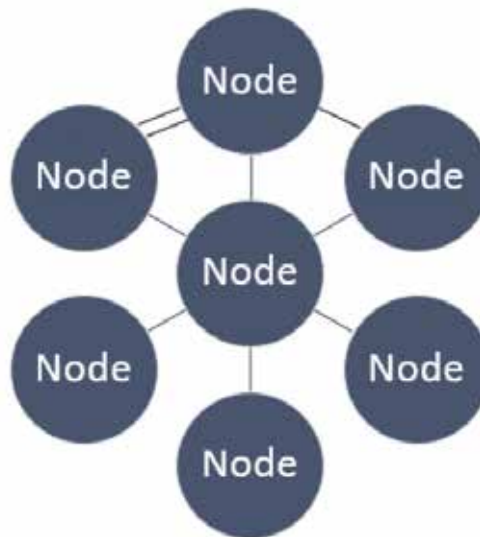


Abbildung 1 – Netzwerk

Beispiele dafür sind ein Netzwerk von Freunden, Servern, Datenbanken oder Kunden. In einer Graphdatenbank müssen nicht alle Knoten gleich sein. Dasselbe gilt für die Beziehungen. Anstatt ein Objekt zu definieren, das als Standard für sämtliche Instanzen dient, lassen sich Knoten erstellen, die individuelle Eigenschaften aufweisen. Bei den Knoten in Abbildung 1 kann es sich um Kunden, Bestellungen, Produkte oder Werbeangebote handeln. Alle diese Knoten können über unterschiedliche Eigenschaften verfügen. Das gilt nicht nur für eine Kategorie von Knoten, sondern für jede einzelne Instanz. Für den Kundenknoten lassen sich zum Beispiel der Vor- und Nachname, das Geburtsdatum sowie das Kaufdatum als Eigenschaften festlegen. Doch möglicherweise sind diese Eigenschaften nicht für alle Kunden bekannt. In einer relationalen Datenbank könnte man allgemeine Eigenschaften definieren und diejenigen auf NULL setzen, die keine Werte für diese Kategorie besitzen. Bei den Beziehungen zwischen Knoten eröffnen sich die gleichen Möglichkeiten: Eigenschaften lassen sich für jede Beziehung speichern, ohne dass diese Eigenschaften auf alle Beziehungen angewendet werden müssen. Richtig: Wir können auch Eigenschaften für Beziehungen speichern. Das ist ein wesentlicher Unterschied zu relationalen Datenbanken, bei denen eine Beziehung in eine Tabelle umgewandelt werden muss, um ihre Metadaten zu speichern.

Eine Graphdatenbank benötigt nicht dieselbe Struktur für alle Instanzen. Wie wir weiter unten sehen werden, zahlt sich diese Flexibilität bei realen Anwendungsbeispielen aus.

Hierarchien und Bäume

Eine Hierarchie ist eine Struktur von Knoten, die aus über- und untergeordneten Knoten besteht. Ein Baum ist eine Hierarchie mit Verzweigungen (mehrere Knoten, die zu einem übergeordneten Knoten gehören). Natürlich sind reine Hierarchien in der Wirklichkeit eher selten (mehr dazu im nächsten Abschnitt). Beispiele für Hierarchien sind betriebliche Strukturen für das Berichts- und Kontrollwesen. In einer typischen Aufsichtshierarchie ist ein Mitarbeiter einem anderen Mitarbeiter unterstellt. Dieser hat wiederum mehrere andere Personen unter sich. Genau so könnten die Unternehmensregeln für das Berichtswesen aussehen.

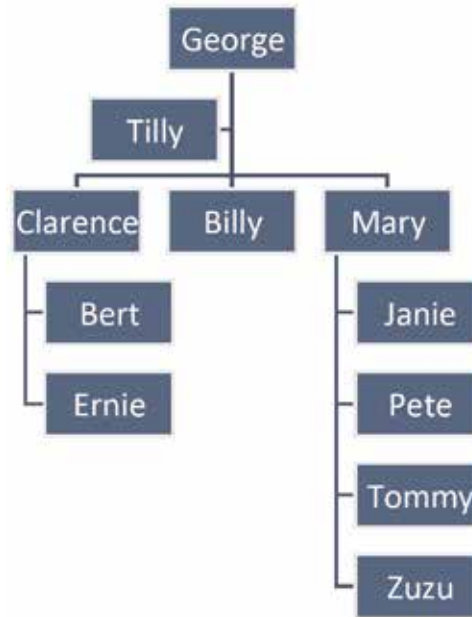


Abbildung 2 – Mitarbeiterhierarchie

Mit einer rekursiven Beziehung zu einer Tabelle MITARBEITER lassen sich diese Regeln einfach in eine relationale Datenbank übertragen. Bei einer überschaubaren Hierarchie, wie in Abbildung 2, können diese Berichtsstrukturen einfach aufrechterhalten werden. Wird die Hierarchie jedoch stark vergrößert, steigen auch die Kosten für die Datenpflege. Denn um die verschiedenen Stufen und Verweise im Auge zu behalten, müssen Ausweidlösungen wie besonderen Hierarchie-Datentypen oder berechnete Spalten bereit stehen. Und was geschieht, wenn ein Knoten befördert wird? Alle vorherigen Beziehungen müssen zurückgesetzt und neu berechnet werden. Ist dieser Knoten an vielen unterschiedlichen Hierarchien beteiligt, ist dies unter Umständen auch für weitere Beziehungen erforderlich. **Die Datengeschichte ist so komplex, dass zahlreiche Fachbücher und Tutorials zu Entwicklung und Aufrechterhaltung dieser Lösungen existieren.** Doch selbst mit allen Tricks ist es schwierig, die Performance von Daten und Anwendungen hoch zu halten.

Hierarchien in der Realität sind eigentlich Graphen

Jeder Mitarbeiter in Abbildung 2 ist ein Knoten. Die Linien in der Berichtsstruktur sind angedeutete Beziehungen: „Ist unterstellt“. Folgt man der Berichtsstruktur nach oben und unten finden sich alle Mitarbeiter, die einem anderen Mitarbeiter unterstellt sind und umgekehrt. In der Realität sind feste Hierarchien jedoch selten. Sogar in unserem Berichtsbeispiel haben Mitarbeiter häufig mehrere Vorgesetzte. Manchmal handelt es sich bei diesen Beziehungen nur um Übergangslösungen (Job-Shading, Job-Sharing, Stellvertretungen). In anderen Fällen bestehen diese Beziehungen aufgrund mehrerer vorhandener „Ist unterstellt“-Beziehungen (Vorgesetzte, Mentoren, Projektmanager usw.). Abbildung 3 zeigt ein realitätsnäheres und vielfältigeres Netz von Beziehungen. Zu sehen sind nicht nur die hierarchischen Beziehungen, sondern auch administrative, protektive und sogar familiäre Verbindungen zwischen Mitarbeitern. Es existieren mehrere Beziehungen innerhalb eines Unternehmens, wobei einige – wie im vorherigen Beispiel von eingeschränkten Beziehungen – Stufen überspringen.

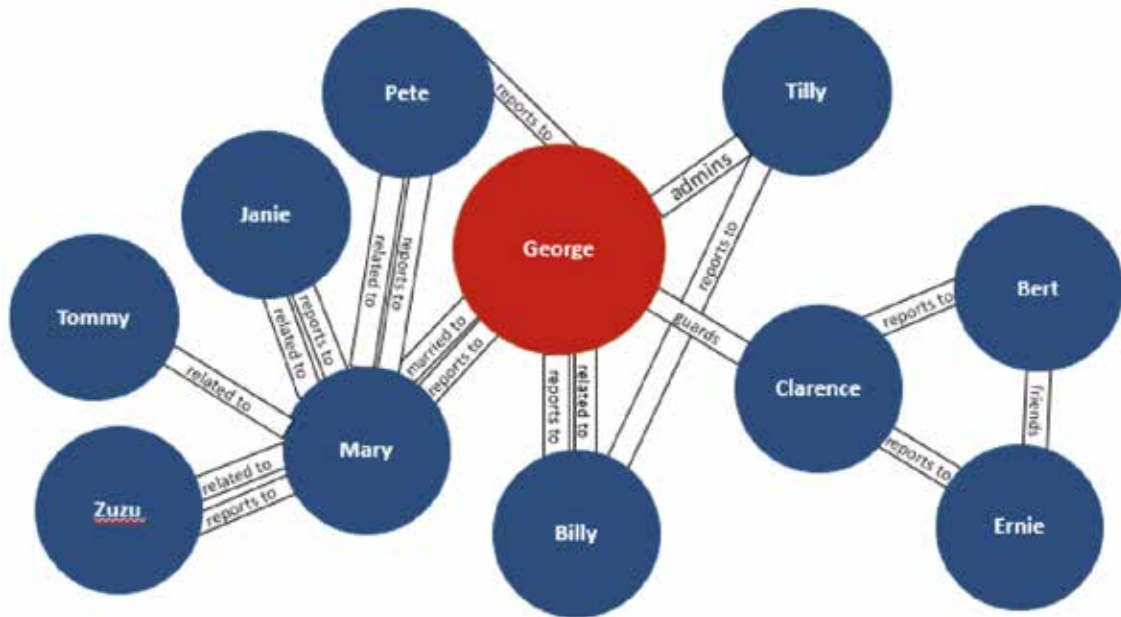


Abbildung 3 – „Ist unterstellt“-Netzwerk

Wenn sie auch Hierarchien ähneln, handelt es sich bei den meisten Unternehmensstrukturen eigentlich um Netzwerke. Sobald in der Realität mehrere komplexe Reportingstrukturen auf unterschiedliche Arten von Beziehungen treffen, zerfällt die Vision von einer perfekten Hierarchie. Das ist bei Organigrammen, „Produkt Hierarchien“, Standorten und Dokumenten der Fall. Betrachten wir unsere Umgebung nicht mehr als Teil einer Hierarchie, erkennen wir alle Graphen darin.

Datenmodellierung und Graphen

In der klassischen Datenverwaltung existieren logische und physische Datenmodelle. Das logische Datenmodell beschreibt die geschäftlichen Anforderungen an eine Datengeschichte, während das physische Datenmodell das Abspeichern von Daten in einer Datenbank festlegt. In einer relationalen Datenbank wird eine gemeinsame Struktur auf jede Instanz einer Entität angewandt. So besteht eine CUSTOMER-Entität und alle diese Entitäten teilen dieselben Eigenschaften bzw. Attribute. Alle zu unterstützenden Eigenschaften müssen daher gefunden und dokumentiert werden, bevor die Datenbank erstellt und Daten importiert werden können.

In einer Graphdatenbank entspricht das logische Modell dem physischen Modell. Man kann das Graphmodell sogar als Servicemodell bezeichnen. Knoten und Beziehungen lassen sich auf dem Whiteboard zeichnen, Eigenschaften und Bezeichnungen anschließend hinzufügen. An dieser Stelle ist die gesamte für die Implementierung in eine Graphdatenbank erforderliche Datenmodellierung abgeschlossen. Bei herkömmlichen logischen Datenmodellen lassen sich sogar bereits bekannte Eigenschaften in das Graphmodell miteinbeziehen. Da das logische Modell das einzige Modell ist, sind für den Übergang von Daten zu Datenbank wesentlich weniger Zeit und Ressourcen (Modellierer, Architekten, DBAs und Entwickler) erforderlich als bei der Erstellung relationaler Stammdatenlösungen.

Zudem können Anwender jede Instanz kennzeichnen und so ihre Daten rollenspezifisch abfragen. Bei bestimmten Abfragen lassen sich beispielsweise nur Unternehmenskunden heranziehen.

In Abbildung 4 sehen wir, dass sowohl Knoten als auch Beziehungen unterschiedliche Eigenschaften aufweisen. Das ist unser Datenmodell wie es auf einem Whiteboard aussehen könnte.

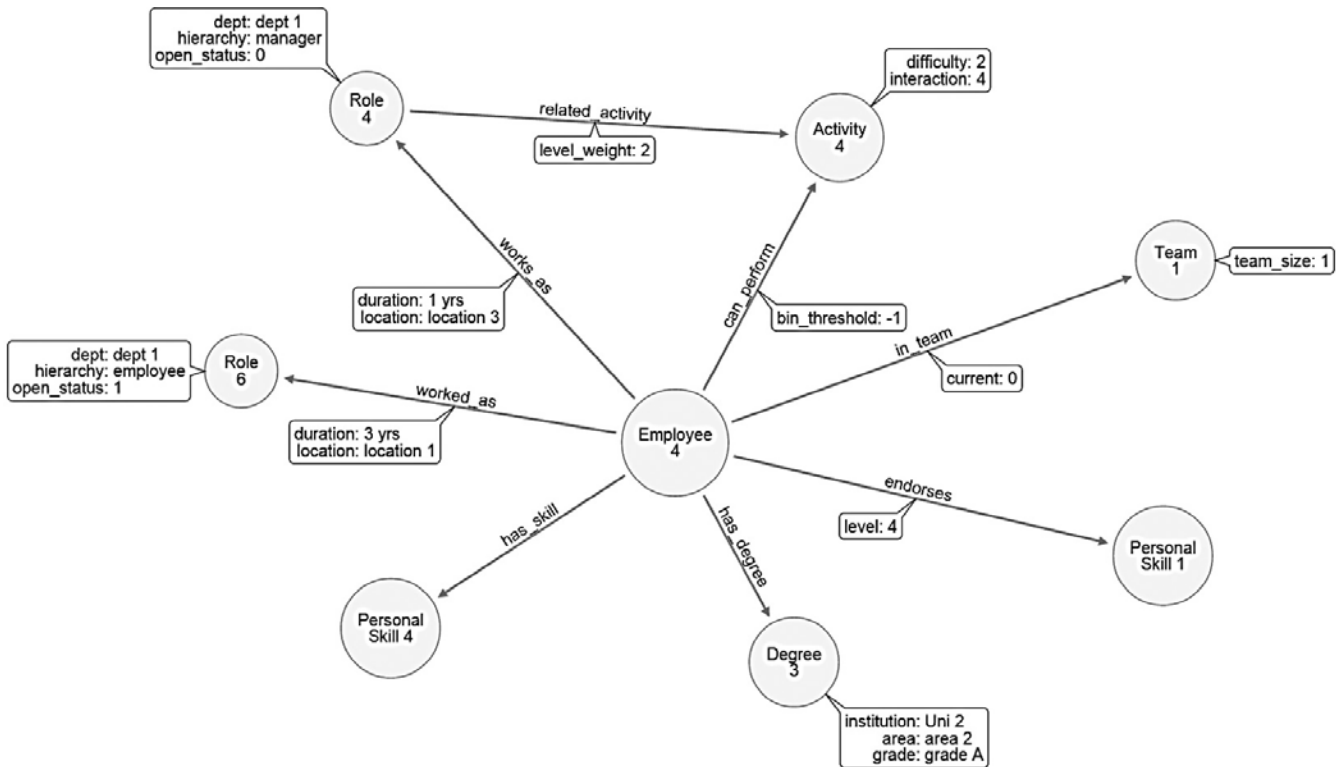


Abbildung 4 – Graph der Mitarbeiterrollen, Aktivitäten, Fähigkeiten, Abschlüsse, Teams

Graphen im Praxiseinsatz

Viele Unternehmen verwenden Graphdatenbanken, um sich auf die Datenbeziehungen innerhalb ihrer Stammdaten zu fokussieren und sich so einen Wettbewerbsvorsprung zu verschaffen.

Neue Verkaufsstrategie für Kommunikationsunternehmen

Ein Informations- und Kommunikationsunternehmen, das Mobilfunk-, TV- und digitale Dienstleistungen anbietet, konnte das Einkaufserlebnis für seine Kunden verbessern, indem es Neo4j im Einzelhandel einsetzte. Zuvor konnten Mitarbeiter Mobiltelefone, Zubehör sowie Sonderaktionen und Abonnements nur über unterschiedliche Systeme an die Laufkundschaft verkaufen. Ihre Datentechnologien waren darauf ausgerichtet Produktdaten zu sammeln. Nicht aber um einen effizienten Kundenservice aufzubauen oder um dem Produktmanagement zu helfen, sinnvolle Produktpakete zu entwickeln. Die Aufgabe bestand darin feste Beziehungen zwischen Produkten und Dienstleistungen herzustellen. Allerdings mussten diese festen Beziehungen stets neu angepasst werden, sobald neue Eigenschaften hinzukamen. Dem Unternehmen war klar, dass es angesichts der sich ständig verändernden Produkt- und Angebotslandschaft eine flexiblere, reaktionsschnellere Lösung benötigte.

Durch die Verwendung von Neo4j können Käufer heute Gruppen von Artikeln im Vorfeld zusammenstellen, so dass Verkäufer Produktpakete direkt an ihre Kunden weitergeben können. Dank der Flexibilität und der Fokussierung auf Produkt- und Angebotsbeziehungen in Neo4j konnte das Unternehmen die für die Zusammenstellung von Produktpaketen erforderliche Zeit von einem Monat auf zehn Tage verkürzen. Im Geschäft können Verkäufer ihre Kunden bei der Auswahl von Artikeln helfen und ihnen darauf abgestimmte zusätzliche Dienstleistungen oder Artikel anbieten. Der Einsatz von vielen unterschiedlichen Datenbeziehungen innerhalb der Produkt- und Kundenstammdaten trägt entscheidend zur Verbesserung des Kundenservice und der Bestellabwicklung, zu niedrigeren Verkaufskosten sowie zu deutlich kürzeren Wartezeiten in den Verkaufsfilialen bei.

Das Einzelhandelsunternehmen verwaltet beinahe eine Million Knoten sowie Millionen von Beziehungen zwischen diesen Knoten, und rüstet dank steigendem Umsatz und einer wachsenden Zahl von Benutzern weiter auf. Um in anderen Geschäftszweigen den gleichen Wettbewerbsvorteil zu erzielen, soll Neo4j zukünftig auch für Dienstleistungen und Produkte im Bereich Kabel- und digitales Fernsehen zum Einsatz kommen.

Produktkombinationen mehrerer Anbieter bei Polyvore

POLYVORE

Polyvore ermöglicht Kunden, Kleidung und Produkte in einem optisch ansprechenden und interaktiven Format zusammenzustellen. Ein Kunde (von Polyvore „Stylist“ genannt) kann mit einem Clipper-Tool Produkte sowie die dazugehörigen Metadaten und Preise von einer beliebigen Website abrufen und diese Kombination mit Grafiken, Texten und persönlichen Fotos ergänzen. Anschließend lassen sich diese Kreationen zu Kollektionen zusammenfassen, womit die Zahl der Beziehungen zwischen den Artikeln weiter wächst.



Jährlich erstellen Polyvore-Benutzer Millionen solcher Kollektionen und regen damit unzählige Kunden zum Kauf an. Derzeit befinden sich Millionen von Artikeln (Bestandseinheiten) in dieser mit benutzergenerierten Inhalt gefüllten Datenbank. Dabei erstellen die Benutzer die Kollektionen basierend auf Trends und ihrer persönlichen Geschmacksrichtung.

So kann Polyvore abfragen, welche Artikel zueinanderpassen. Die Antworten verraten, welche anderen Produkte möglicherweise Interesse wecken und den Verkauf zusätzlicher Artikel ankurbeln. Zudem können Benutzer Produkte mit entsprechenden Merkmalen kennzeichnen. Diese Tags stellen Beziehungen zwischen Produktdaten her, wodurch der durch die Benutzer erstellte Datensatz an Umfang zunimmt. Auch Einzelmarken können Teil der Polyvore-Community werden, indem sie ihre Artikel in die Datenbank stellen und dann anhand von den Empfehlungen und „Likes“ ihrer Produkte im Verhältnis zu anderen Produkten Trends und Beliebtheitsgrad verfolgen.

In diesem Anwendungsfall mit Produkt- und Metadaten sowie mit den von Stylisten vergebenen Tags dreht sich alles um die Beziehungen zwischen diesen Konzepten. Ein herkömmlicher Produktkatalog konzentriert sich auf die Produkte und ihre Eigenschaften. Doch im Fall von Polyvore liegt das große Potenzial in den benutzergenerierten Datenbeziehungen. Es geht nicht nur um die Daten – es geht vor allem um Beziehungen. Mit dem leistungsstarken Graphen von Neo4j ist Polyvore in der Lage, nicht nur Trends zu erkennen, sondern auch die beeinflussenden Faktoren. Dazu zählen Preis, Style, Farbe, Größe und alle anderen von der Community gewählten Eigenschaften. Marken erhalten Informationen darüber, wie beliebt ihre Produkte bei den einflussreichsten Käufern im E-Commerce sind.

Warum eine Graphdatenbank?

Nach diesen Praxisbeispielen einer Graphdatenbank, kommen wir nun zu den Szenarien, in denen ihre Anwendung am sinnvollsten ist. In vielen Fällen ist das Potenzial für Graphdatenbanken klar erkennbar: soziale Medien, Freundesnetzwerke, reale Technologienetzwerke. Hier sind die Beziehungen zwischen Benutzern und Objekten eindeutig Schlüsseldatenstrukturen, die man sogar als Netzwerke bezeichnen kann. Doch es gibt viele Anwendungsfälle in Unternehmen, deren Datenbeziehungen zwar ebenso wertvoll sind, die aber nicht als Netzwerk betrachtet werden.

Neue Wechselbeziehungen innerhalb von Daten erkennen

Eine Kombination der Einkaufsdaten von stationären Geschäften mit den Daten des Online-Shoppings eines Kunden könnte so aussehen:

- Besuche der Website des Geschäfts
- Angezeigte, angeklickte, bewertete, auf die Wunschliste gesetzte oder über soziale Medien geteilte Artikel
- Zum Einkaufswagen hinzugefügte Artikel sowie die Zahl der dazu nötigen Klicks

Zwischen diesen Daten bestehen wertvolle Beziehungen, die ein wettbewerbsorientierter Händler identifizieren möchte. Zwar ist es möglich, die Daten in einer relationalen Datenbank zu verwalten und abzufragen, dies wäre jedoch sehr zeitaufwändig und kostspielig. Mit wachsendem Datenvolumen steigen zudem die Kosten einer relationalen Datenbank hinsichtlich der Verarbeitungszeit. Je mehr Daten gesammelt werden, desto schwieriger lassen sich Echtzeitdaten erzielen. Dabei sind Fragen interessant wie:

- Welche Arten von Artikeln werden von einer bestimmten Kundengruppe häufig zusammen gekauft? Welche Rolle spielt dabei der zeitliche Aspekt?
- Welche Mitglieder eines Haushalts kaufen welche Art von Artikeln an welchen Tagen? Wie hängt dieses Einkaufsverhalten mit der Entfernung zu einem lokalen Geschäft zusammen?

Deduplikation von Daten

Zur Duplikation von Stammdaten kommt es durch den Zukauf von Drittdaten und durch Unternehmenszusammenschlüsse. Auch die Nutzung von Herstellerpaketen kann eine Stammdatenduplikation zur Folge haben. Mit Graphdatenbanken lässt sich schneller und einfacher feststellen, ob die Kundin „Karen Lopez“ in Toronto, Ontario, identisch ist mit „Karen Lopez“ in Scarborough, Ontario, einem Vorort von Toronto. Das gelingt dank der Vorteile von Graphabfragen. Dabei wird durch Analyse aller Datenbeziehungen in mehreren Instanzen ein „Vertrauensergebnis“ ermittelt, das besagt, dass es sich bei jedem Datensatz um die gleiche Person handelt. Die Erkenntnis, dass zwei Karen Lopez mit unterschiedlichen Adressen Wartungsarbeiten am gleichen Fahrzeug durchführen lassen oder zur selben Zeit eine Veranstaltung besuchen, ermöglicht ein Verständnis der Muster und Beziehungen in den Daten. Solche Datenbeziehungen ermöglichen es Stammdaten zu ermitteln, selbst wenn noch keine allgemeinen Kundeninformationen vorliegen.

Auswirkung und Was-wäre-wenn-Analysen

Unternehmen müssen Stammdaten nicht nur verwalten, sondern auch Änderungen berücksichtigen. So sind etwa Produktdaten häufig das Ziel von Umgestaltungen und Rebranding. Marketing- und Produktmanager verlangen beispielsweise Antworten auf folgende Was-wäre-wenn-Fragen:

- Welche Folgen hätte eine Regeländerung einer Werbekampagne zu bestimmten Produkten?
- Wie würde sich eine Änderung in der Bündelung von Produkten und Dienstleistungen auf die durchschnittliche Zahl an Bestellungen auswirken?
- Wie viele Kunden wären betroffen, wenn ein Produktpaket aus dem Programm genommen wird? Und welche Produktpakete bieten sich als Ersatz an?
- Welche Produkte und Werbeaktionen können angesichts der Kundendemografie eines neuen Verkaufsgebiets die besten Ergebnisse erzielen?

Alle diesen Fragen konzentrieren sich auf Datenbeziehungen über verschiedene Stammdatensätze hinweg.

360-Grad-Ansicht von Stammdaten

Es ist eine Sache eine Abbildung von KUNDE oder PRODUKT zu modellieren und zu erstellen. Unternehmen aber treffen keine Entscheidungen auf Grundlage von Datensilos. Wir haben Kunden, weil sie Produkte kaufen oder kaufen möchten. Wir haben Produkte, weil wir erwarten, dass diese gekauft werden.

Das Beispiel der genannten Datengeschichten zeigt, dass bestimmte Fragen an die Masterdaten sich in einem Graph einfacher und flexibler beantworten lassen und so eine schnellere Reaktion ermöglichen.

Moderne hybride Datenarchitekturen

Die moderne Stammdatenarchitektur verlässt sich nicht länger nur auf eine Technologie. Die wettbewerbsstärksten Unternehmen haben erkannt, dass sie mit dem richtigen Instrument für den richtigen Job Antworten schneller und mit einem höheren Erkenntniswert finden.

Ansicht einer modernen Stammdatenarchitektur

Abbildung 5 zeigt Anwendungsfälle, die sich weiterhin am besten für relationale Datenbanken eignen. In anderen Fällen ist der Einsatz von NoSQL-Technologien, einschließlich Graphdatenbanken, die optimale Lösung. Ein Bericht von Gartner vom Januar 2015 zeigt sogar, dass CIOs und Datenprofis eine Vielzahl von Datenbanktechnologien benötigen:

Durch den Anstieg polyglotter Persistenz müssen CIOs und IM-Führungskräfte neue Ansätze für die Datenkonsistenz, den Datenzugriff und die Datenintegration über mehrere Persistenz-Typen hinweg berücksichtigen.

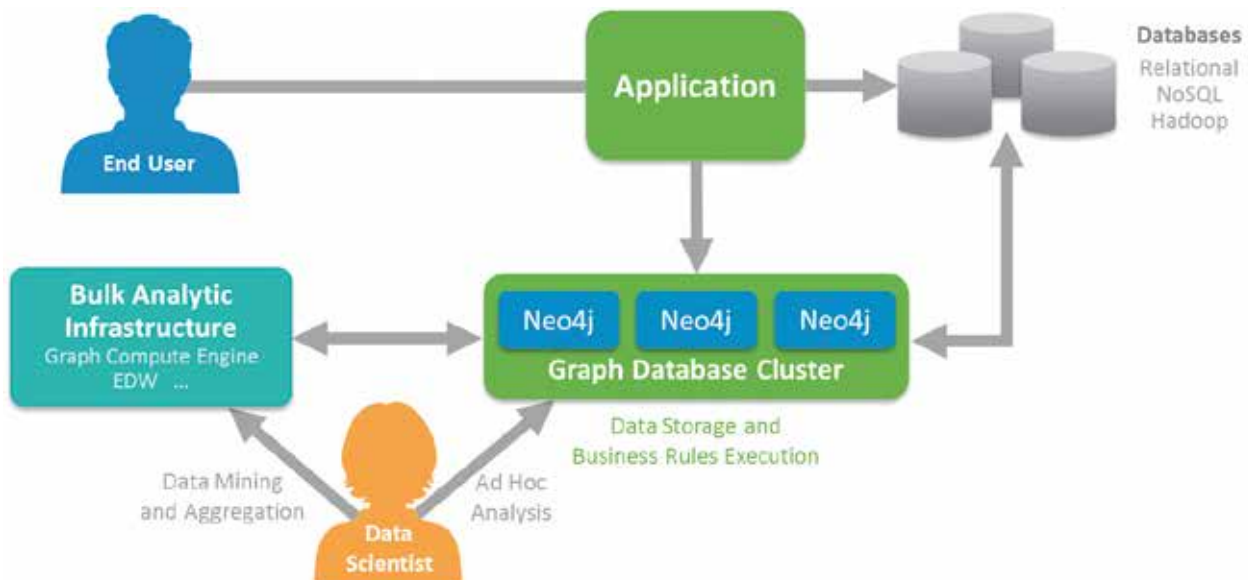


Abbildung 5 – Moderne Datenarchitektur
(Neo4j. Mit Genehmigung verwendet)

Unternehmen wollen Daten nicht nur erfassen, sondern auch zur Unterstützung von Geschäftsentscheidungen in Echtzeit heranziehen. Das bedeutet den Einsatz von Technologien, mit denen sich am besten Fragen an den Datensatz stellen lassen. Relationale Datenbanken werden dadurch nicht überflüssig. Nach wie vor eignen sie sich für hochstrukturierte und nach klaren Unternehmensregeln ausgerichteten Datenspeicher. Diese Anwendungen verlangen die Art von Einschränkungen und „Gleichheit“, für die relationale Datenbanken ausgerichtet sind. Sie sind für eine hohe Integrität in unseren Daten erforderlich, garantieren, dass alle notwendigen Daten zur Fertigstellung einer Transaktion erfasst werden, und liefern in hohem Maße vorhersehbare, stabile Datensätze für Geschäftsprozesse. Diese Stärken eines relationalen Systems können bei einem anderen Anwendungsfall zur Datenverwaltung und -analyse jedoch stören. Darum verlangen einige Datenspeicher weniger Struktur, Beschränkungen und Konsistenz zwischen den Dateninstanzen. Tatsächlich ist dies einer der Hauptgründe für die Implementierung einer NoSQL-Lösung.

Performance

Dank der Skalierbarkeit des zugrundeliegenden Designs können Stammdatenlösungen mit Neo4j in der gleichen Geschwindigkeit wie das Unternehmen wachsen. Speziell zur Speicherung und Verarbeitung von Graphabfragen entwickelt, entschlüsselt die Datenbank den Mehrwert der Datenbeziehungen erheblich schneller als andere SQL oder NoSQL-Lösungen.

Flexibilität

Eine Graphdatenbank ermöglicht die Verwaltung von Daten ohne einen Standardsatz an Attributen oder Eigenschaften, die für jeden Knoten oder selbst für eine Beziehung zwischen Knoten gleich sind. Das ist vor allem dann von Vorteil, wenn Daten aus mehreren Quellen zusammen fließen. Kundendaten können aus dem Online-Handel, den Ladengeschäften sowie aus E-Mail-Verteilerlisten stammen. Jede dieser Datenquellen besitzt wahrscheinlich gemeinsame und unterschiedliche Attribute mit einem unterschiedlichen Grad an Vollständigkeit. Beim Versuch diese Daten in eine relationale Datenbank zu importieren, sehen sich Anwender gezwungen Regeln zur Datenqualität auszuschalten oder bestimmte Daten zu bereinigen und zur Einhaltung der Integrität zurückzuweisen. Das ist nicht das Ziel. Unternehmen wollen alle Daten in die Datenbank importieren, um mit ihnen in ihrer ursprünglichen Form arbeiten zu können.

Zudem besteht die Möglichkeit im laufenden Betrieb neue Daten hinzuzufügen, da nicht alle Knoten oder Beziehungen den gleichen Metadatentyp aufweisen müssen. Es lassen sich alle gewünschten Daten importieren, mit bestehenden Daten verknüpfen und abfragen. Da das logische Modell dem physischen Modell entspricht, müssen Anwender Lösungen nicht jedes Mal umgestalten, wenn sie neue Erkenntnisse über ihre Stammdaten gewinnen wollen.

Dank dieser praxisnahen und flexiblen Verwaltung von Daten und Datenbeziehungen lassen sich mehr Fragen stellen und mehr Antworten erhalten als zum Zeitpunkt der Datenerfassung geplant war. Durch die Flexibilität, weitere Datenbeziehungen in Echtzeit hinzuzufügen, können Unternehmen ihre Datengeschichten weiter verbessern.

Zusammenfassung

Das MDM der Zukunft wird als eine hybride und polyglotte Architektur für wettbewerbsorientierte Unternehmen angesehen. Dabei liegt der Wert des MDMs nicht nur in den Daten selbst, sondern auch in den leistungsstarken Beziehungen zwischen den Daten. Eine Herausforderung stellt die Entwicklung schneller und flexibler Systeme zur Durchsuchung all dieser Beziehungen dar. Herkömmliche relationale Systeme haben ihre Vorzüge, doch das Abfragen eines Netzwerks von Stammdaten ist mit ihnen kostspielig, unflexibel und langsam.

Graphdatenbanken wie Neo4j wurden von Beginn an für die Unterstützung von „Graphdatengeschichten“ entwickelt – genau die Geschichten, die Daten erzählen wollen. Da das logische Datenmodell dem physischen Modell entspricht, können Datenprofis schneller und flexibler auf Fragen zu diesen Datenbeziehungen antworten.

Wie geht es weiter?

Graphdatenbanken sind Teil einer modernen Datenarchitektur in Unternehmen. Was sind also Ihre nächsten Schritte?

1. Lesen Sie das Buch „Graph Databases“ (<http://graphdatabases.com>) für einen tieferen Einblick in die Theorie und die praktische Anwendung von Graphtechnologien.
2. Absolvieren Sie den Online-Kurs zu Graphdatenbanken: <http://neo4j.com/graphacademy/online-course>
3. Sprechen Sie mit geschäftlichen Anwendern von Schlüsseldaten in Ihrem Unternehmen über Fragen, die sie ihren Daten gerne stellen würden, die sie aber aufgrund von technischen Einschränkungen oder zu hohen Kosten nicht stellen können.
4. Informieren Sie sich über relationale Datenbankmanagementsysteme (RDBMS), Graphdatenbankenkonzepte und -tools: <http://neo4j.com/developer/graph-db-vs-rdbms>
5. Helfen Sie den Geschäftskunden zu verstehen, dass es Datenbeziehungen gibt, die sich in den bestehenden Datenstrukturen nicht veranschaulichen lassen.
6. Laden Sie Neo4j unter <http://neo4j.com> herunter oder verwenden Sie Amazon Machine Image (AMI) (unter <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html> erhältlich), um eine Umgebung in der Cloud zu erstellen.
7. Gewinnen Sie neue Erkenntnisse aus Ihren Stammdaten, die Ihrem Unternehmen einen Wettbewerbsvorteil auf Grundlage von Beziehungen in den bestehenden Daten verschaffen.

Über die Autorin

Karen López ist Senior Project Manager und Datenarchitekt bei InfoAdvisors. Sie verfügt über mehr als 20 Jahre Erfahrung in der Betreuung von Unternehmen beim Implementieren großer, vielschichtiger Programme.

InfoAdvisors ist ein in Toronto ansässiges Beratungsunternehmen für Datenverwaltung. Unser Schwerpunkt liegt auf der praktischen Anwendung der Datenverwaltung. Im Rahmen unserer Tätigkeit bewerten wir Kosten, Vorteilen und Risiken einer bestimmten Technologie, um den jeweiligen Anforderungen unserer Kunden gerecht zu werden.

Wir möchten, dass Sie Ihre Daten lieben.

Besuchen Sie uns unter datamodel.com

Info zu Neo Technology

Neo Technology entwickelt die weltweit führende Graphdatenbank Neo4j, mit der sich Datenverbindungen optimal nutzen lassen. Dazu zählen Anwendungen wie das Angebot von personalisierten Empfehlungen für Produkte und Dienstleistungen, zusätzliche soziale Medienfunktionen für Websites, Analyse von Telekommunikationsnetzwerken oder die Neuorganisation von Stammdatenverwaltung sowie Identity und Access Management. Unternehmen nutzen Graphdatenbanken, um Datenvernetzungen und -zusammenhänge abbilden, abspeichern und abfragen zu können. Großunternehmen wie Walmart, eBay, UBS, Nomura, Cisco, HP und Telenor sowie Startups wie CrunchBase, Medium, Polyvore und Zephyr Health verwenden Neo4j, um erfolgsentscheidenden Nutzen aus Datenverbindungen zu ziehen. Weitere Informationen erhalten Sie unter www.neo4j.com.