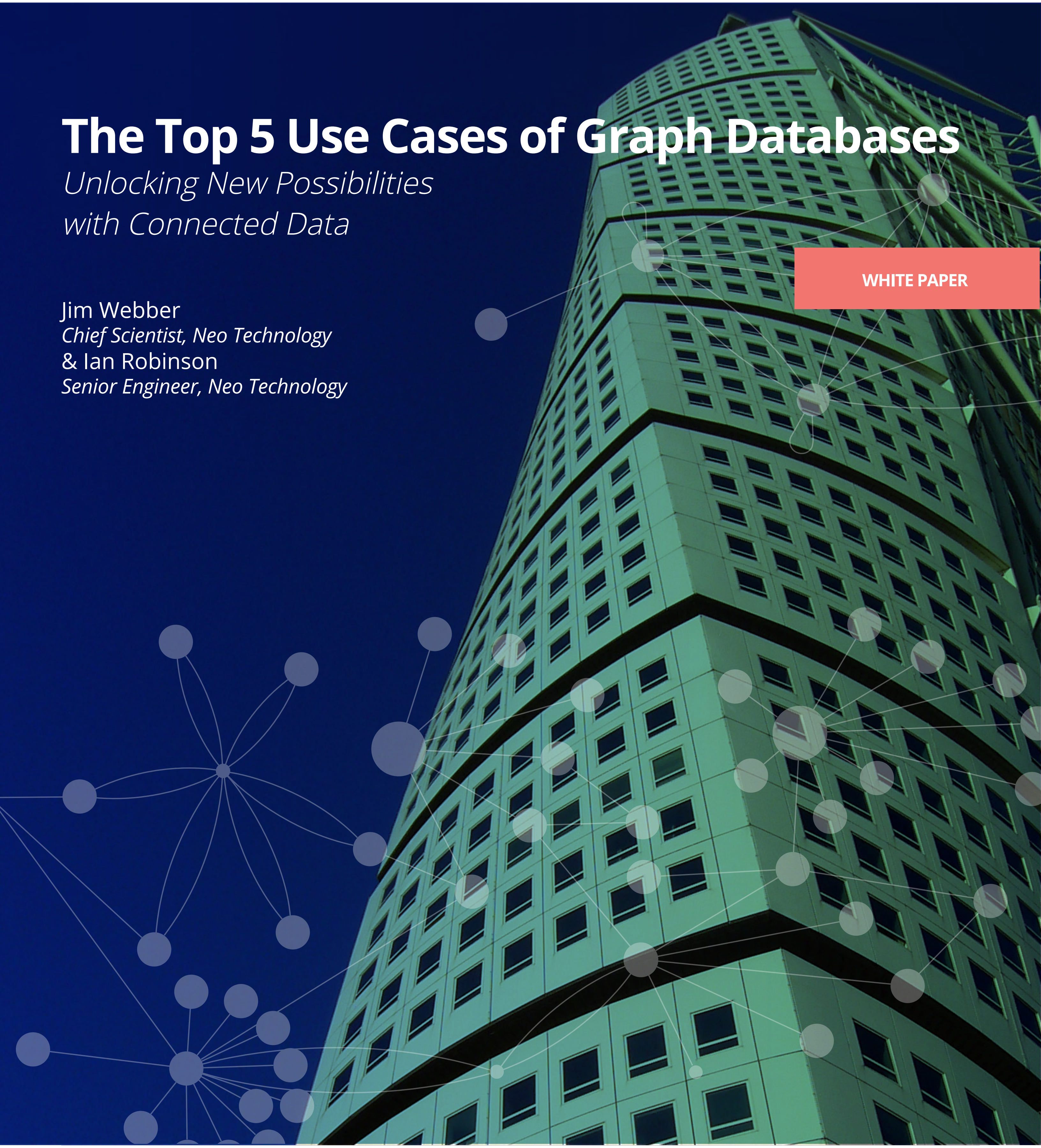
The #1 Platform for Connected Data



The Top 5 Use Cases of Graph Databases

Unlocking New Possibilities with Connected Data

WHITE PAPER

Jim Webber
Chief Scientist, Neo Technology
& Ian Robinson
Senior Engineer, Neo Technology

TABLE OF CONTENTS

"Stop merely *collecting* data points, and start *connecting* them."

The Top 5 Use Cases of Graph Databases

Unlocking New Possibilities with Connected Data

Jim Webber & Ian Robinson

Introduction

"Big data" grows bigger every year, but today's enterprise leaders don't only need to manage larger volumes of data, but they critically need to generate insight from their existing data. So how should CIOs and CTOs generate those insights?

To paraphrase Seth Godin, businesses need to stop merely collecting data points, and start connecting them. In other words, the relationships between data points matter almost more than the individual points themselves.

In order to leverage those data relationships, your organization needs a database technology that stores relationship information as a first-class entity. That technology is a graph database.

Ironically, legacy relational database management systems (RDBMS) are poor at handling relationships between data points. Their tabular data models and rigid schemas make it difficult to add new or different kinds of connections.

Graphs are the future. Not only do graph databases effectively store the relationships between data points, but they're also flexible in adding new kinds of relationships or adapting a data model to new business requirements.

So how might your enterprise leverage graph databases to generate competitive insights and significant business value from your connected data?

Here are the top five use cases of graph database technologies:

Use Case #1: Fraud Detection

Banks and insurance companies lose billions of dollars every year to fraud. Traditional methods of fraud detection fail to minimize these losses since they perform discrete analyses that are susceptible to false positives and negatives. Knowing this, increasingly sophisticated fraudsters develop a variety of ways to exploit the weaknesses of discrete analysis.

Graph databases offer new methods of uncovering fraud rings and other complex scams with a high level of accuracy through advanced contextual link analysis, and they are capable of stopping advanced fraud scenarios in real time.

Why Use a Graph Database for Fraud Detection?

While no fraud prevention measures are perfect, significant improvements occur when you look beyond individual data points to the connections that link them.

Understanding the connections between data, and deriving meaning from these links, doesn't necessarily mean gathering new data. You can draw significant insights from your existing data simply by reframing the problem in a new way: as a graph.

Unlike most other ways of looking at data, graphs are designed to express relatedness. Graph databases uncover patterns that are difficult to detect using traditional representations such as tables. An increasing number of companies use graph databases to solve a variety of connected data problems, including fraud detection.

Example: E-commerce Fraud

As our lives become increasingly digital, a growing number of financial transactions are conducted online. Fraudsters have adapted quickly to this trend and have devised clever ways to defraud online payment systems.

While this type of activity can and does involve criminal rings, even a single well-informed fraudster can create a large number of synthetic identities and to carry out sizeable schemes.

Consider an online transaction with the following identifiers: user ID, IP address, geo location, a tracking cookie and a credit card number. Typically, the relationships between these identifiers should be (almost) one-to-one. Some variations naturally account for shared machines, families sharing a single credit card number, individuals using multiple computers and the like.

However, as soon as the relationships between these variables exceed a reasonable number, fraud should be considered as a strong possibility. The more interconnections exist amongst identifiers, the greater the cause for concern. Large and tightly-knit graphs are very strong indicators that fraud is taking place.

See the graphic on the following page for an example:

Fraud Detection Challenges:

Complex link analysis to discover fraud patterns

- Uncovering fraud rings requires you to traverse data relationships with high computational complexity – a problem that's exacerbated as a fraud ring grows.

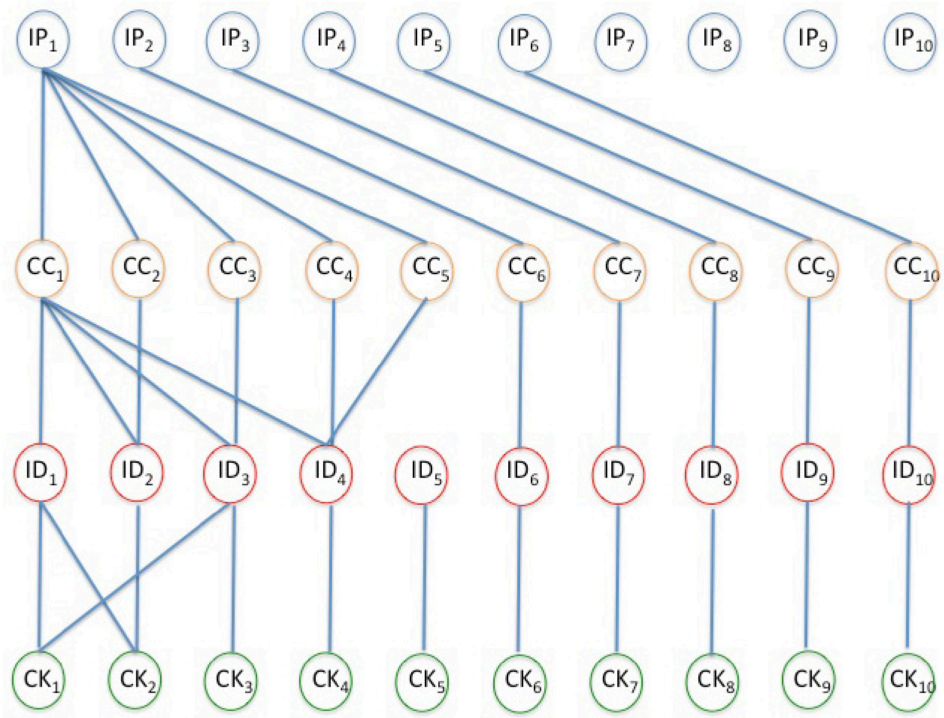
Detect and prevent fraud as it happens

- To prevent a fraud ring, you need real-time link analysis on an interconnected dataset, from the time a false account is created to when a fraudulent transaction occurs.

Evolving and dynamic fraud rings

- Fraud rings are continuously growing in shape and size, and your application needs to detect these fraud patterns in this highly dynamic and emerging environment.

Use Case #1: Fraud Detection



A graph of a series of transactions from different IP addresses with a likely fraud event occurring from IP_1, which has carried out multiple transactions with five different credit cards.

By putting checks into place and associating them with the appropriate event triggers, such schemes can be uncovered before they are able to inflict significant damage. Triggers can include events such as logging in, placing an order or registering a credit card – any of which can cause the transaction to be evaluated against the fraud graph. Fan-out might be skipped, but complex graphs can be flagged as a possible instance of fraud.

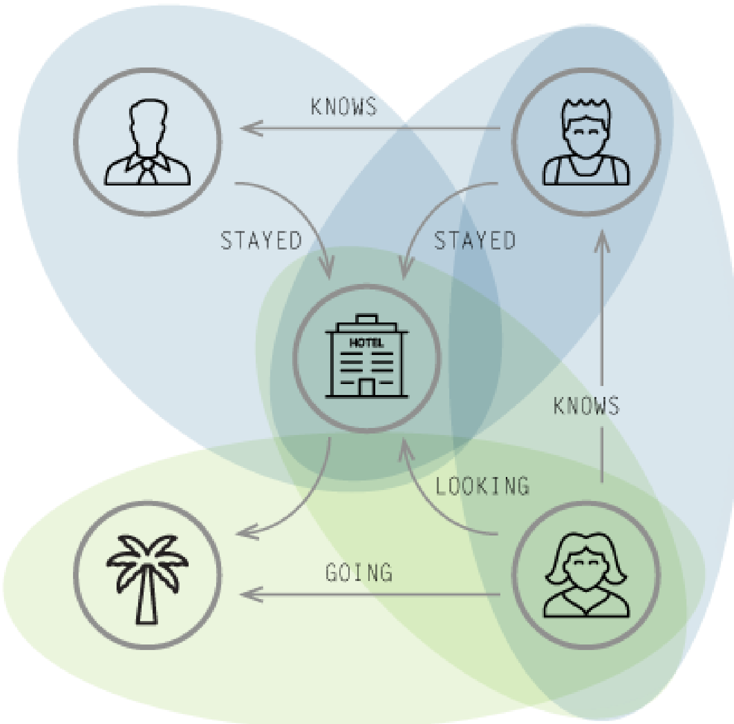
Conclusion

When it comes to graph-based fraud detection, you need to augment your fraud-detection capability with link analysis. That being said, two points are clear:

- As business processes become faster and more automated, the time margins for detecting fraud are narrowing, increasing the need for a real-time solution.
- Traditional technologies are not designed to detect elaborate fraud rings. Graph databases add value through analysis of connected data points.

Graph databases are the ideal enabler for efficient and manageable fraud detection solutions. From fraud rings and collusive groups, to educated criminals operating on their own, graph databases uncover a variety of important fraud patterns – and all in real time.

Use Case #2: Real-Time Recommendation Engines



Whether your enterprise operates in the retail, social, services or media sectors, offering your users highly targeted, real-time recommendations is essential to maximizing customer value and staying competitive. Unlike other business data, recommendations must be inductive and contextual in order to be considered relevant by your end consumers.

With a graph database, you're able to capture a customer's browsing behavior and demographics and combine those with their buying history to instantly analyze their current choices and then immediately provide relevant recommendations – all before a potential customer clicks to a competitor's website.

Why Use a Graph Database to Power Real-Time Recommendation Engines?

The key technology in enabling real-time recommendations is the graph database. Graph databases also out-class other database technology for connecting masses of buyer and product data (or connected data in general).

Making effective real-time recommendations depends on a database that understands the relationships between entities, as well as the quality and strength of those connections. Only a graph database efficiently tracks these relationships according to user purchase, interactions, and reviews to give you the most meaningful insight into customer needs and product trends.

Graph-powered recommendation engines can take two major approaches: identifying resources of interest to individuals; or identifying individuals likely to be interested in a given resource. With either approach, graph databases make the necessary correlations and connections to serve up the most relevant results for the individual or resource in question.

Real-Time Recommendation Challenges:

Process large amounts of data and relationships for context

- Collaborative and content-based filtering algorithms rely on rapid traversal of a continually growing and highly interconnected dataset.

Offering relevant recommendations in real time

- The power of a suggestion system lies in its ability to make a recommendation in real time using immediate history.

Accommodate new data and relationships continuously

- The rapid growth in the size and number of data elements means the suggestion system needs to accommodate both current and future requirements.

Use Case #2: Real-Time Recommendation Engines

Examples: Walmart and eBay

Retail industry leader Walmart has sales of more than $460 billion and employs 2.2 million associates worldwide, serving more than 245 million customers weekly through its 11,000 stores in 27 countries and e-commerce websites in 10 countries. Their development team has decided to use a graph database to serve up real-time product recommendations by using information about what users prefer.

Walmart Software Developer Marcos Wada states that a graph database “helps us to understand our online shoppers’ behavior and the relationship between our customers and products, providing a perfect tool for real-time product recommendations.”

E-commerce giant eBay has also found success using a graph-powered suggestion engine, in this case, for a sophisticated real-time courier/package routing solution.

Senior Developer Volker Pacher at eBay says his team found a graph database “to be literally thousands of times faster than our prior MySQL solution, with queries that require 10-100 times less code. Today, our graph database provides eBay with functionality that was previously impossible.”

Conclusion

Storing and querying recommendation data using a graph database allows your application to provide real-time results rather than precalculated, stale data. As consumer expectations increase – and their patience decreases – providing these sorts of relevant, real-time suggestions will become a greater competitive advantage than ever before.

Real-time recommendation engines provide a key differentiating capability for enterprises in retail, logistics, recruitment, media, sentiment analysis, search and knowledge management.

Use Case #3: Master Data Management

Master data is the lifeblood of your enterprise, including data such as:

- Users
- Customers
- Products
- Accounts
- Partners
- Sites
- Business units

Many business applications use master data and its often held in many different places, with lots of overlap and redundancy, in different formats, and with varying degrees of quality and means of access. Master data management (MDM) is the practice of identifying, cleaning, storing, and – most importantly – governing this data.

MDM best practices vary along the spectrum of merging all master data into a single location to managing data assets for easy access from a single service or application. In both cases (or any hybrid solution), enterprise data architects need a data model that provides for ad hoc, variable and exceptional structures as business requirements change. That sort of rapidly evolving model fits best with a graph database.

Why Use a Graph Database for Master Data Management Solutions?

Because master data is highly connected and shared, poorly built MDM systems cost business agility in a way that ripples throughout your enterprise. Most legacy MDM systems rely on a relational database which isn't optimized for traversing relationships or rapid responsiveness.

These data connections and relationships in your master datasets are essential to competitive advantage as business analytics evolve. The good news is that graph databases are ideal for modeling, storing and querying the hierarchies, metadata and connections in your master data.

With graph databases, your master data is much easier to model, costing you fewer resources (modelers, architects, DBAs and developers) than building a relational solution. In addition, with a graph database, you don't have to migrate all of your master data into a single location. Graph relationships easily connect your siloed data between CRM systems, inventory systems, accounting and point-of-sale systems to provide a consistent vision of your enterprise data.

Master Data Management Challenges:

Complex and hierarchical datasets

- Managing the top-down hierarchies of master data with a relational database results in complex and unwieldy code that is slow to run, expensive to build and time-consuming to maintain.

Real-time storage and query performance

- The master data store must integrate with and provide data to a host of applications within the enterprise. Providing real-time information on complex and highly interconnected dataset is a significant challenge.

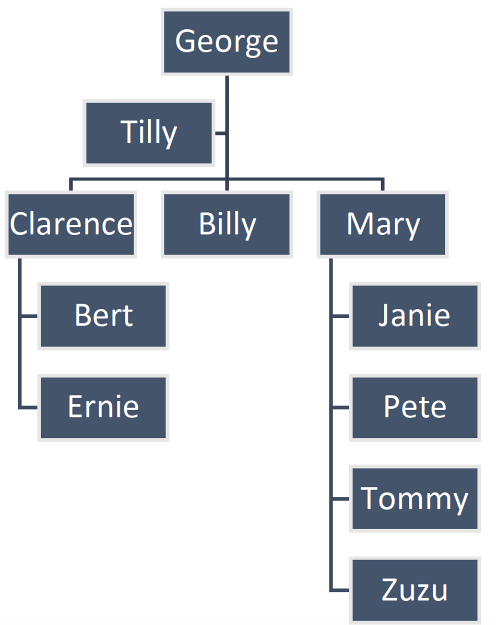
Dynamic structure

- Master data is dynamic in nature, making it harder for developers to design systems that accommodate its evolution.

Use Case #3: Master Data Management Solutions

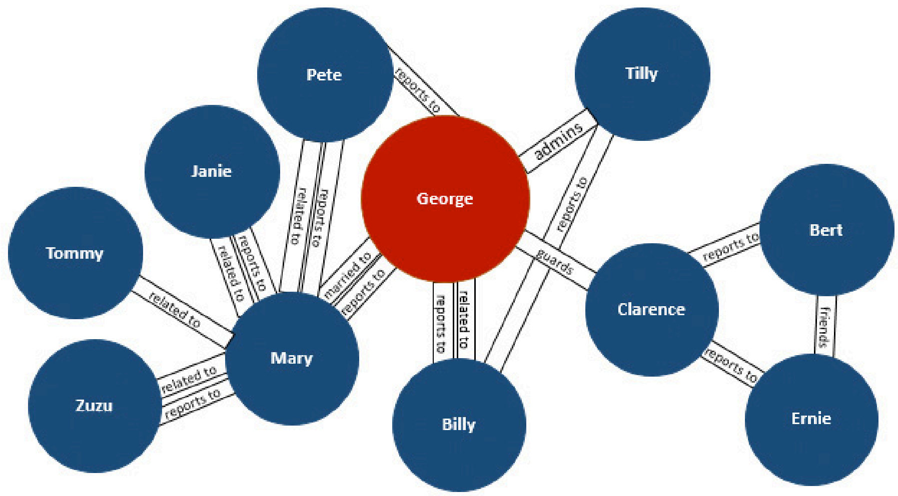
Example: Employee Hierarchy Data

In your master data, a hierarchy is any structure where nodes have other nodes above and below them, possibly with multiple branches. One example of a master data hierarchy is employee reporting and supervisory structures like the one to the below left.



A master data hierarchy illustrating employee reporting and supervisory structures. This hierarchy would traditionally serve as a model in a relational database.

A small hierarchy, such as the one to the left, is easy enough to model and maintain in a relational database. But as soon as we model a much larger set of employees, both querying and maintaining the data gets more expensive. For example, if an employee gets a promotion, every relationship must be reset for every hierarchy in which the employee participates.



A master data network detailing employee reporting and supervisory relationships, this time with more real-life complexity.

Of course, such pure hierarchies rarely exist in the real world. Employees often report to a multiple people, and sometimes reporting relationships exist only for transitional reasons (such as job shadowing or coverage). In fact, most business hierarchies are actually networks filled with real-life complexities and many kinds of relationships. See the second figure to the above right as an example of our earlier hierarchy re-envisioned as a more realistic network (or graph).

Traditional hierarchies need to be reimagined as networks that are easier and more flexible to model with a graph database as business needs change. While the example discussed has to do with employee reporting relationships, the same principle of master data networks applies to product listings, document relationships and sales or customer data.

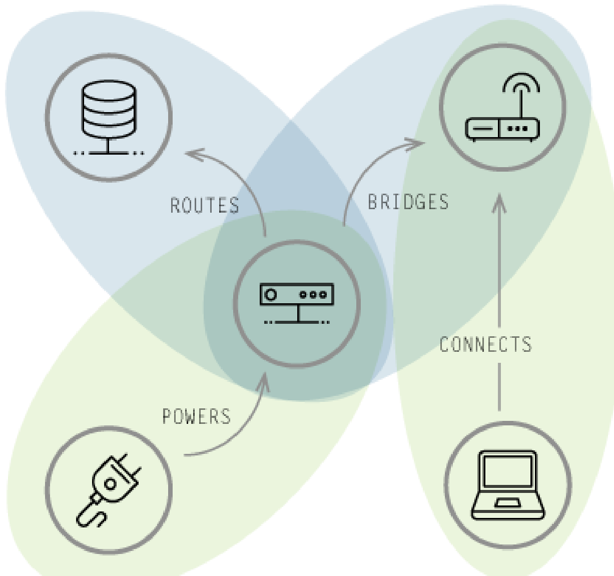
Conclusion

The best data-driven business decisions aren't based on stale information silos. Instead, you need real-time master data with information about data relationships.

Graph databases are built from the ground up to support data relationships. With more efficient modeling and querying, organizing your master data in a graph yields relevant answers faster and with more flexibility than ever before.

Use Case #4: Network and IT Operations

By their nature, networks are graphs. Graph databases are, therefore, an excellent fit for [modeling, storing and querying network and IT operational data](#) no matter which side of the firewall your business is on – whether it's a communications network or a data center.



Today, graph databases are being successfully employed in the areas of telecommunications, network management, impact analysis, cloud platform management and data center and IT asset management.

In all of these domains, graph databases store configuration information to alert operators in real time to potential shared failure modes in the infrastructure and to reduce problem analysis and resolution times from hours to seconds.

Why Use a Graph Database for Network and IT Operations?

As with master data, a graph database is used to bring together information from disparate inventory systems, providing a single view of the network and its consumers – from the smallest network element all the way to the applications, services and customers who use them.

A graph representation of a network enables IT managers to catalog assets, visualize their deployment and identify the dependencies between the two. The graph's connected structure enables network managers to conduct sophisticated impact analyses, answering questions like:

- Which parts of the network – which applications, services, virtual machines, physical machines, data centers, routers, switches and fiber – do particular customers depend on? (Top-down analysis)
- Conversely, which applications and services, and ultimately, customers in the network will be affected if a particular network element – such as a router or switch – fails? (Bottom-up analysis)
- Is there redundancy throughout the network for the most important customers?

A graph database representation of the network can also be used to enrich operational intelligence based on event correlations. Whenever an event correlation engine (such as a [Complex Event Processor](#)) infers a complex event from a stream of low-level network events, it assesses the impact of that event against the graph model and triggers any necessary compensating or mitigating actions.

Challenges in Network and IT Operations:

Troubleshooting a network

- Physical and human interdependencies are extremely complex in any network or IT environment, making it difficult to troubleshoot.

Impact analysis

- Relationships among network nodes are neither purely linear nor hierarchical, making it difficult to determine the interdependencies of network elements on each other.

Growing number of physical & virtual nodes

- The rapid growth in the size and number of data elements means the suggestion system needs to accommodate both current and future requirements.

Use Case #4: Empowering Network and IT Operations

Example: A Large European Telecom Provider

To showcase the use of a graph database in the IT and network operations sector, here is an excerpt from an interview with a software consultant who helped implement a graph database solution for one of Europe's largest telecommunication providers.

"This telecom provider had a very large complex network with many silos and processes – including network management information spread across more than thirty systems. The large number of data sources was in part due to network complexity, and in part due to different business units, as well as organic growth through mergers and acquisitions. These different sources also created a very non-linear fabric that had to be modeled and understood from various dimensions.

"Prior to using a graph database, they had different network layers stored in different systems – for instance, one system might be dedicated to cell towers, another to fiber cables and another devoted to information about consumers or enterprise customers.

"One of their business challenges was around maintenance and ensuring redundancy – they needed to know if they took a device down for maintenance, exactly who might be impacted and what the penalties might be, as well as what alternate routes might better mitigate the impact.

"[Implementing a graph database solution] was almost a dream business case because you could measure the benefit of the project as the telecommunications provider began to manage production-level changes that impacted its many actual customers.

"After implementation of the graph database model and the impact analysis queries, it was easy to extend the application to support single point of failure (SPOF) detection thanks to the flexibility of the graph model. Also, by providing an effectively unified cross-domain view, experts from different silos could work together for the first time and agree on a common domain terminology."

Conclusion

Discovering, capturing and making sense of complex interdependencies is central to effectively running Network and IT operations are a critical part of running an enterprise. Whether it's optimizing a network or application infrastructure or providing more efficient security-related access – these problems involve a complex set of physical and human interdependencies that are a challenge to manage.

The relationships between network and infrastructure elements are rarely linear or purely hierarchical. Graph databases are designed to store that interconnected data, making it easy to translate network and IT data into actionable insights.

Identity and Access Management Challenges:

Highly interconnected identity and access permissions data

- To verify an accurate identity and its access permissions, the system must traverse a highly interconnected dataset constantly growing in size and complexity.

Productivity and customer satisfaction

- As users, products and permissions grow, traditional systems no longer deliver responsive query performance, resulting in frustration for users.

Dynamic structure and environment

- With rapid growth in the size of users and their associated metadata, your application needs to accommodate both current and future identity management requirements.

Use Case #5: Identity & Access Management

Identity and access management (IAM) solutions store information about parties (e.g., administrators, business units, end-users) and resources (e.g., files, shares, network devices, products, agreements), along with the rules governing access to those resources. IAM solutions apply these rules to determine who can or can't access or manipulate a resource.

Traditionally, identity and access management has been implemented either by using directory services or by building a custom solution inside an application's backend. Hierarchical directory structures, however, can't cope with the complex dependency structures found in multi-party distributed supply chains. Custom solutions that use non-graph databases to store identity and access data become slow and unresponsive as their datasets grow in size.

Why Use a Graph Database for Storing Identity and Access Data?

A graph database can store complex, densely connected access control structures spanning billions of parties and resources. Its richly and variably structured data model supports both hierarchical and non-hierarchical structures, while its extensible property model allows for capturing rich metadata regarding every element in the system.

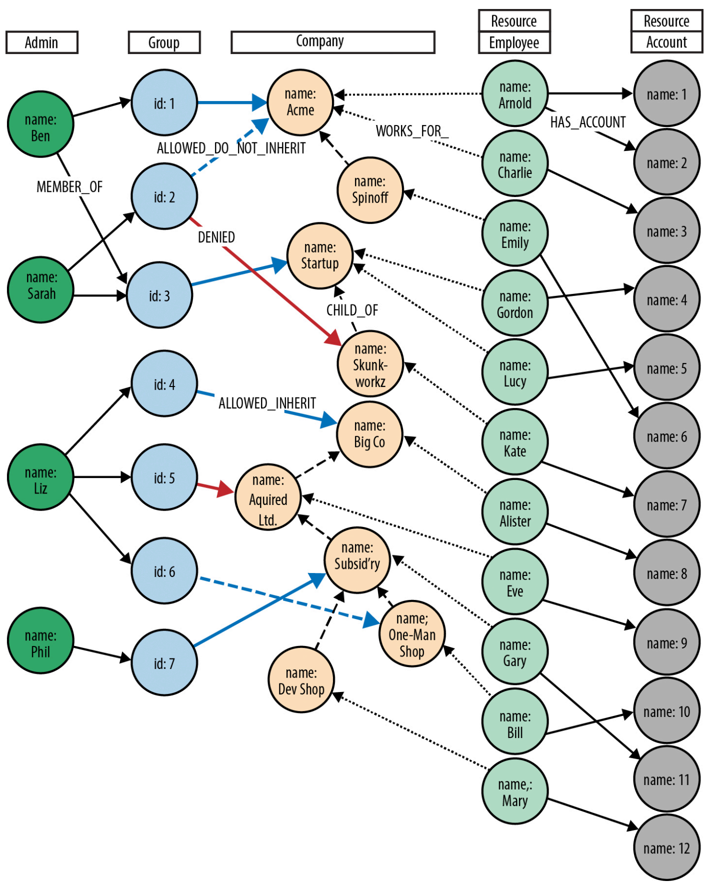
With a query engine that can traverse millions of relationships per second, graph database access lookups over large, complex structures execute in milliseconds not minutes or hours.

As with network and IT operations, a graph database access control solution allows for both top-down and bottom-up queries:

- Which resources – company structures, products, services, agreements and end users – can a particular administrator manage? ***(Top-down)***
- Given a particular resource, who can modify its access settings? ***(Bottom-up)***
- Which resource can an end-user access?

Access control and authorization solutions powered by graph databases are particularly applicable in the areas of content management, federated authorization services, social networking preferences and software as a service (SaaS) offerings, where they realize minutes-to-milliseconds increases in performance over their relational database predecessors.

Use Case #5: Identity and Access Management



A sample of Telenor Norway's data model showing their identity and access management application.

Example: Telenor Norway

Telenor Norway is an international communications services company. For several years, it has offered its largest business customers the ability to self-service their accounts. Using a browser-based application, administrators within each of these customer organizations can add and remove services on behalf of their employees.

To ensure users and administrators see and change only those parts of the organization and the services they are entitled to manage, the application employs a complex identity and access management system which assigns privileges to millions of users across tens of millions of product and service instances.

To the left is an example of Telenor's data model.

Due to performance and responsiveness issues, Telenor decided to replace its existing IAM system with a graph database solution. Their original system used a relational database, which used recursive JOINs to model complex organizational structures and product hierarchies.

Because of the join-intensive model, their most important queries were unacceptably slow.

In contrast, once they implemented a graph database solution, Telenor realized the performance, scalability and adaptiveness necessary for handling their identity and access management needs, reducing queries that once took many minutes to milliseconds.

Conclusion

For your enterprise organization, managing multiple changing roles, groups, products and authorizations is an increasingly complex task. Relational databases simply aren't up to the task of managing your identity and access needs as queries are far too slow and unresponsive.

Using a graph database, you seamlessly track all of your identity and access relationships real-time results, connecting your data along intuitive relationships. With an interconnected view of your data, you have better insights and controls than ever before.

"Graph databases allow data professionals at every level to exploit the potential of their data *relationships* rather than just individual data points."

In Review: The Graph Database Competitive Advantage

These five use cases of graph databases are hardly a comprehensive list, but they do highlight some impactful and profitable applications of graph technologies.

Nearly every enterprise benefits from fraud detection, master data management and real-time recommendation engines. In addition, no major corporation is without a growing IT network or an increasing number of user identities to be managed and monitored.

Even so, there are plenty of other use cases for graph technologies, including logistics and routing, the life sciences, social networking, gaming, government, sports and even non-profit.

Today's CIOs and CTOs are under increasing pressure to provide actionable insights from their big data even as datasets grow larger and more unwieldy. What they need is a technology that determines the connections between data points and derives appropriate cogent conclusions.

Graph databases ***are*** that technology solution. They allow data professionals at every level to exploit the potential of their data ***relationships*** rather than just individual data points, and the only limit to how those relationships might be harnessed is the imagination of the database user.

Graph databases are a rising tide – not merely a passing fad – in the world of big data insights, and the enterprises that tap into their power realize significant competitive advantages.

Questions about Neo4j?

Contact us:
1-855-636-4532
info@neo4j.com