The #1 Platform for Connected Data

WHITE PAPER

Driving Innovation in Retail with Graph Technology

How Top Retailers Use Neo4j

Philip Rathle, VP of Products

neo4j.com

White Paper

Driving Innovation in Retail with Graphs

How Top Retailers Use Neo4j

Philip Rathle, VP of Products

TABLE OF CONTENTS

What is a graph database?

A graph database is a highly scalable transactional database that leverages data relationships as first-class entities.

Introduction

Today's retailers face a number of complex and emerging challenges. Thanks to lower overhead and higher volume, online behemoths like Amazon can deliver products faster and at a lower price – driving smaller retailers out of business.

To remain viable, retailers must be nimble enough to face their colossal online competition while also addressing another new reality of retail: The customer is now at the center of the value chain. In order to adapt to these new realities, retailers must have real-time control of inventory, payment, and delivery systems. However, real-time responsiveness is difficult for traditional retailers slowed down by legacy infrastructure.

In order to re-invent the value chain from linear to circular and highly connected, retailers need to modernize their infrastructure rapidly and cost-effectively. In addition, web-based retailers must find a way to handle scale and sophistication to remain competitive.

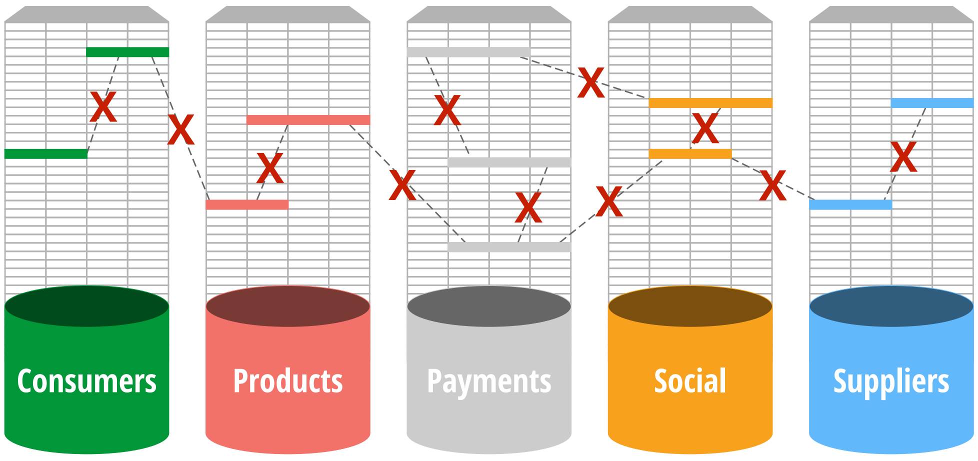
The purpose of this white paper is to illustrate how real-world retailers address these pressing challenges using the power of graph database technology – specifically Neo4j.

Personalized Product and Promotion Recommendations

Delivering real-time recommendations to online shoppers is a proven way to maximize revenue. It improves the customer experience and increases sales. However, shoppers expect finely tuned recommendations and react poorly to one-size-fits-all or uninformed recommendations (e.g., ***"I've already bought that. Why are they showing it to me again?"***). To be effective, recommendations must be personalized based on the individual consumer's preferences, shopping history, interests and needs – in addition to what's already in their current shopping cart.

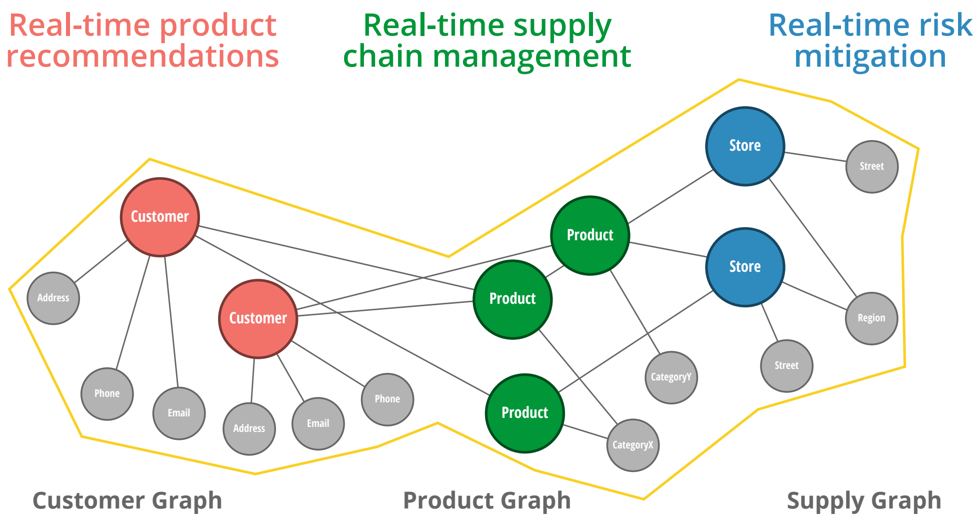
Real-time recommendations require data products that connect masses of complex buyer and product data (and connected data in general) to gain insight into customer needs and product trends. This cannot be achieved with relational database (RDBMS) technology; the SQL queries are too complex and take too long to provide recommendations in real-time. The same goes for big data processing technologies like Hadoop and Spark. These technologies work well for something like email recommendations, which are delivered once a day, but they are not real-time.

Challenge with traditional systems



Data stored in relational databases and other silos

By design, graph databases quickly query customers' past purchases, as well as instantly capture any new interests shown in their current online visit, both of which are essential for making real-time recommendations. Because relationships are treated as first-class entities in a graph database, retailers can connect customers' browsing history with their purchasing history, as well as their offline product and brand interactions. This enables a real-time recommendation algorithm to utilize a customer's past and present choices to offer personalized recommendations. No offline pre-compute is necessary, eliminating the associated delay.



KEY RETAIL USE CASES

- Recommendations, such as product-, coupon-, and content-recommendations
- Real-time pricing engines
- Customer 360 for personalized experiences, loyalty programs, and call centers
- Logistics for ecommerce order delivery
- Supply chain visibility and traceability
- Network and IT infrastructure monitoring to ensure system uptime and uncover attack paths

INDUSTRY-REPORTED BENEFITS OF REAL-TIME RECOMMENDATIONS

- Amazon saw double-digit revenue growth
- Between 20-40% increase in average order value and items per order
- Between 2-4x conversion rates
- Increase in staff efficiency by reducing manual and batch oriented tasks

Furthermore, in order to counter dynamic pricing from the likes of Amazon, retailers need the ability to change pricing and promotions at any level of a product hierarchy in real time. For example, they must be able to mark down all 60-inch televisions by 10% for the next two hours if the right economic and competitive factors indicate such a move is necessary. Similarly, retailers must be able to implement competing promotions. They might reduce all smart phone prices except Apple iPhones due to Apple's strict pricing guidelines.

Real-time promotions such as these involve complex rules that are made simple when handled by a graph database. The database may hold millions of relationships that have only one parent node. With a graph database, retailers can change one relationship type rather than a thousand products and all their prices.

CASE STUDY:

Walmart

Walmart became the world's largest retailer by understanding its customers' needs better than any competitor. An important tool in achieving that understanding is the Neo4j graph database.

Walmart's Brazilian ecommerce group wanted to understand the behavior and preferences of its online buyers with enough speed and in enough depth to make real-time, personalized "you may also like" recommendations. However, Walmart quickly recognized that it would be difficult to deliver such functionality using traditional relational database technology.

"A relational database wasn't satisfying our requirements about performance and simplicity, due to the complexity of our queries," said Marcos Wada, software developer at Walmart.

To address this, Marcos's team decided to use Neo4j, the leading graph database. Matching a customer's historical and session data is trivial for graph databases, enabling them to easily outperform relational and other NoSQL data products.

Walmart deployed Neo4j in its remarketing application run by the company's ecommerce IT team based in Brazil, and it has been using Neo4j in production since early 2013. Neo4j enables Walmart to understand online shoppers' behavior, as well as the relationship between customers and products. As a result, the retailer has also been able to up- and cross-sell major product lines in core markets.

"With Neo4j we could substitute a complex batch process that we used to prepare our relational database with a simple and real-time graph database. We could build a simple and real-time recommendation system with low latency queries," Marcos said.

CASE STUDY: TOP 10 RETAIL COMPANY

One Top 10 US-based, bricks-and-mortar retailer turned to Neo4j after its burgeoning online operation was almost overwhelmed by the volume of customer traffic it attracted on Cyber Monday 2015.

The company was running its site on an IBM DB2 relational database, and on Cyber Monday 2015, it offered an across-the-board 15% discount to site visitors. While the retailer had pulled in more customers than any other bricks-and-mortar rival, the price paid was unacceptable.

The site's checkout function kept working that day, but 90% of customer traffic was delayed. As one senior company executive said: "We pushed a lot of guests to the site and we were very successful in terms of volume. But the reality was we got significantly more traffic than we ever projected, and we couldn't handle it. We protected checkout so the site functioned. But we disappointed way too many guests, and that's never okay, period."

The biggest bottleneck was the crucial but complex promotions process, where a company invites shoppers to add last-minute extras to their online cart. To flash up exactly the right recommendations requires software that can instantly analyze the shopper's cart contents and their buying history, and dig through 15-30 layers of data – such as promotion types, qualifying manufacturers, product names and categories – all in real time.

This proved beyond a conventional relational database like DB2. So the retailer considered Neo4j, which is optimized to rapidly carry out such complex searches among masses of connected data.

The company already knew its biggest rival, Walmart, had turned to Neo4j to provide the best web experience for its customers (see case study above), so in mid-2016 the company rolled out both a new Neo4j-based front-end and backend to its website, transforming the company's real-time promotions engine and online cart promotion calculations.

Neo4j now processes 90% of the retailer's 35 million-plus daily transactions – which involve between three and 22 hops across different layers of data – in 4 milliseconds or less. And during Q4 2016, the vital Christmas retail period, the company's digital sales rose 34% to a record high, helped by the friction-free Neo4j solution.

The company's digital sales rose 34% to a record high, helped by the friction-free Neo4j solution.

Personalized Experiences Powered by a 360-Degree View of the Customer

Retailers can personalize the online customer experience by serving relevant content based on the customer's desires, interests, and needs. Doing so improves customer engagement and is likely to lead to increased revenue and customer loyalty. For example, by serving relevant blog posts beside product descriptions, retailers can portray themselves as experts on how to use a particular product. Customers will increase their visits — and purchases — because they know they can get valuable information from a reliable source.

Retailers can also use path analytics to help improve outcomes. This involves analyzing customer behavior leading up to a purchase and using that data to guide customers along a more profitable path. This may entail adjusting content, or changing where a link takes future customers.

When it comes to combining all these data sources into a personalization engine, relational databases can't do complex recommendation computations in real-time.

Retailers can also identify a dimension shared by a group of customers and cluster them based on these attributes. For example, customers could be clustered around the attribute of having (or not having) children, or customers could be clustered based on profession and tenure, such as an early-career engineer versus a seasoned VP of marketing. Different dimensions of consumers have different responsibilities and incomes, and therefore different buying habits. Retailers can use this information to personalize content for each customer.

Retailers have plenty of data that can be used to determine the best paths and content to serve customers. That includes data pertaining to products, markets, social media, master data, digital assets, and the like. However, this data often resides in information silos, making it difficult to consolidate and identify opportunities to serve customers the most relevant content.

When it comes to combining all these data sources into a personalization engine, relational databases can't do complex recommendation computations in real-time. You could move the data into Hadoop or a data warehouse to pre-compute recommendations for each customer, but the recommendations will always be slightly out of date. In addition, it is inefficient to pre-compute recommendations for an entire customer base every day when only a small portion of the customer base visits the website on any given day.

Rather than forklift all customer data into a centralized system, a graph database allows you to keep data where it is and add a graph analysis overlay. Each customer can be given a department identifier, which is then tied back to the main customer identifier. The identifier for each department or line of business consists of individual identifiers, resulting in a two-overlap graph of each customer. This allows you a view of the bigger picture of the customer relationship and to quickly navigate back into the original systems anytime a customer interacts with the company.

CASE STUDY: ADIDAS

The adidas group, a global leader in the sporting goods industry, wanted to offer a more personalized experience to its online customers. Unlike other online retailers that offer static content to all website visitors, the adidas Group wanted to serve content based on user interests, local languages, regional sporting news and market-specific product offerings. There was just one problem: The data required to provide personalized web experiences was spread across various information silos.

"We have many different silos, many different data domains, and in order to make sense out of our data, we needed to bring those together and make them useful for us," said Senior Project Manager Sokratis Kartelias. On a technical level, data models didn't align between information silos, and there wasn't a standard, consistent way to communicate between the different data domains.

Rather than consolidate all the data into a single place, Kartelias wanted to create a "Shared Metadata Service" that would allow employees to categorize and search for content across every platform and division within the enterprise. The Shared Metadata Service would also allow the Group to target audiences with content organized by language, country, sport and athlete. In addition, the Service would need to include search engine optimization (SEO) for content and be able to govern the roles for who has rights to change data and ownership of employees to ensure high quality data.

The Neo4j graph database proved to be ideal for creating the Service, offering access and searchability to all relevant data, along with support for emerging services. In order to implement the Shared Metadata Service on Neo4j, Kartelias had to first unify the different models between content, product data and master data. With the help of Neo4j consultants, Kartelias's team defined an optimal data model that connects all three domains, relating information as diverse as marketing campaigns, product specifications, contracted athletes and associated teams, sports categories, gender and more.

Today, the Neo4j-powered Shared Metadata Service has two million nodes with nearly ten million relationships, but for Kartelias, this is only the first step. The ultimate goal is to build a recommendation engine that uses Neo4j to offer relevant, real-time suggestions to online shoppers.

The Neo4j graph database proved to be ideal, offering access and searchability to all relevant data, along with support for emerging services.

Ecommerce shoppers aren't willing to wait any longer than two days to receive their online purchases. As a result, retailers must meet or beat the standard — or risk losing customers to Amazon.

Graphs for Ecommerce Delivery Service Routing

Amazon has set the standard for shipping and delivery. Thanks to its free two-day shipping for Amazon Prime members, ecommerce shoppers aren't willing to wait any longer than two days to receive their online purchases. As a result, retailers must meet or beat the standard — or risk losing customers to Amazon.

To shorten delivery times, retailers must have visibility into inventory at storefronts and distribution centers, as well as the transit network. They need to know, for example, whether a routing problem could delay a product shipped from a distribution center located closer to the customer, or whether a shortage of products makes it impossible to meet a specific delivery date altogether. Identifying the fastest delivery route requires support for complex routing queries at scale with fast and consistent performance.

Ecommerce delivery service routing is a natural fit for graph databases given the highly connected nature of the data. It's not just that it requires a lot of "hops" across data points, but that there can be many different paths with any number of permutations. Those permutations may be optimized and deemed the best path at different times of the year and for different products, even within a single order. A graph database can take these various factors into account and support complex routing queries to streamline delivery services.

CASE STUDY:

ebay

Even before its acquisition by global ecommerce leader eBay, London-based Shutl sought to give people the fastest possible delivery of their online purchases. Customers loved the one-day service, and it grew quickly. However, the platform Shutl built to support same-day delivery couldn't keep up with the exponential growth.

The service platform needed a revamp in order to support the explosive growth in data and new features. The MySQL queries being used created a code base that was too slow and too complex to maintain. The queries used to select the best courier were simply taking too long, and Shutl needed a solution to maintain a competitive service. The development team believed a graph database could be added to the existing Service-Oriented Architecture (SOA) to solve the performance and scalability challenges.

eBay selected Neo4j for its flexibility, speed and ease of use. Its property graph model harmonized with the domain being modeled, and the schema-flexible nature of the database allowed easy extensibility, speeding up development. In addition, it overcame the speed and scalability limitations of the previous solution.

"Our Neo4j solution is literally thousands of times faster than the prior MySQL solution, with queries that require 10-100 times less code. At the same time, Neo4j allowed us to add functionality that was previously not possible," said Volker Pacher, Senior Developer for eBay.

The Cypher graph query language allowed queries to be expressed in a very compact and intuitive form, speeding development. The team was also able to take advantage of existing code, using a Ruby library for Neo4j that also supports Cypher.

Implementation was completed on schedule in just a year. Queries are now easy and fast. The result is a scalable platform that supports expansion of the business, including the growth it is now experiencing as the platform behind eBay Now.

Supply Chain Visibility

Supply chains are vast and complex. Products are often composed of different ingredients or parts that move through different vendors, and each of those parts may be composed of subparts, and the subparts may come from still other subparts and other vendors from various parts of the world. Because of this complexity, retailers tend to know only their direct suppliers, which can be a problem when it comes to risk and compliance.

Retailers need transparency across the entire supply chain to detect fraud, contamination, high-risk sites, and unknown product sources. If a specific raw material is compromised in some way, for example, companies must be able to rapidly identify every product impacted. This requires managing and searching large volumes of data without delay or other performance issues. Transparency is also important for identifying weak points in the supply chain or other single points of failure. If a part or ingredient was previously available from three suppliers but is now only available from one, the retailer needs to know how that might affect future output.

Achieving visibility across the supply chain requires deep connections. A relational database is simply not built to handle a lot of recursive queries or JOINs, and as a result performance suffers. A graph database, however, is designed to search and analyze connected data. The architecture is built around data relationships first and foremost. This enables retailers and manufacturers to manage and search large volumes of data with no performance issues and achieve the supply chain visibility they need.

Retailers need transparency across the entire supply chain to detect fraud, contamination, high-risk sites, and unknown product sources

CASE STUDY:

Transparency-One

Recognizing the inherent risks of the supply chain, Transparency-One sought to build a platform that allows manufacturers and brand owners to learn about, monitor, analyze and search their supply chain, and to share significant data about production sites and products.

Transparency-One initially considered building the platform on a classic SQL database-type solution. However, the company quickly realized that the volume and structure of information to be processed would have a significant impact on performance and cause considerable problems. So, Transparency-One began looking at graph databases.

Neo4j was the only graph database that could meet Transparency-One's requirements, including the capacity to manage large volumes of data. Neo4j is also the most widely used graph database in the world, both by large companies and startups.

"We tested Neo4j with dummy data for several thousand products, and there were no performance issues," said Chris Morrison, CEO of Transparency-One. "As for the search response time, we didn't have to worry about taking special measures, since we got back results within seconds that we would not have been able to calculate without this solution."

Using Neo4j, Transparency-One got up and running and built a prototype in less than three months. Since then, the company has extended Neo4j with new modules and the platform is currently deployed by several companies.

A graph database can help retailers address revenue management while delivering the scale and performance necessary for a real-time pricing engine.

Graphs for Revenue Management

It's never been easier for customers to comparison shop. In a matter of minutes, customers can compare prices for a specific product across a dozen stores — and all from the comfort of home. They can even compare prices and purchase from a competitor while shopping at a different retailer's brick-and-mortar storefront. To compete on prices and optimize profitability, retailers need to deliver competitive prices in real-time.

Competitive pricing is based on a variety of factors, such as inventory, location, season, consumer demand, and more. These factors are very fluid and change quickly. For example, if a hotel plans pricing based on the basketball championships, which goes to seven games, then those cities where the games are hosted will have low inventory, and should be priced accordingly. But if the championship is over in five games, then there will be more inventory for what would've been the last two games, and the pricing should change appropriately.

What's more, each retail location may have a different price based on the market. The more retailers are able to understand their micro-markets and optimize product pricing to match availability, the more options there are to improve margins and sales in the right proportion. However, a relational database can't keep up with these data points, and poor performance makes it impossible to deliver real-time pricing updates across multiple locations.

A graph database can help retailers address revenue management while delivering the scale and performance necessary for a real-time pricing engine. The interdependencies between the many variables can be represented as a graph, which gives retailers an effective way to determine and efficiently calculate prices even as dependencies change rapidly.

CASE STUDY:

Marriott International – one of the world's largest hospitality companies – needed a new pricing engine to drive both revenue and competitive differentiation. The older pricing engine was being stretched by complex pricing rules that resulted in long pricing update cycles despite spending on massive amounts of hardware and tuning the legacy application. Marriott needed to provide a global system for on-property managers to review pricing recommendations and maintain pricing strategies for a 365-day horizon.

The hospitality company's previous revenue management system was a manual-based mainframe green screen system. Prices were changed relatively infrequently due to the complexity of doing so. Publishing performance was slow — new prices didn't show up for minutes or hours — and users avoided the system. But as the number of properties increased, so too did the number of pricing strategies and the complexity of those strategies.

The system used a highly normalized data model consisting of about ten levels of data in a relational database management system with foreign key constructs. The company decided to build an application aware of the data model and relationships of the data. However, some rate programs required over 30,000 lines of SQL queries to process, which took too long. With a business requirement to publish in less than 60 seconds, the company decided a rewrite was necessary.

The team decided to try Neo4j to achieve scale and performance for its 4,500 independent graphs with related data. In just eight weeks they built a prototype using Neo4j. Although it wasn't functionally complete, the team built a projection model that could process its most complex property in less than 34 seconds compared to 240 seconds (four minutes). The prototype also demonstrated that it could process properties concurrently compared to serially.

In five months, the company deployed the global system based on Neo4j for its 4,500 properties. As a result, the company found a 10-fold increase in publishing volumes, a 96% reduction in average publishing times, and a 50% reduction in server capacity and infrastructure costs.

With Neo4j, Marriott International found a 10-fold increase in publishing volumes, a 96% reduction in average publishing times, and a 50% reduction in server capacity and infrastructure costs.

Systems Administrators: Graphs for Network and IT Operations

Retail IT organizations also benefit from graph databases. Oftentimes, companies have complex networks, and they increasingly have components that are in the cloud (or multiple clouds) as well as on-premises data centers. It can be difficult to represent every IT asset and understand how they're interconnected in most traditional configuration management databases (CMDB).

Consider, for example, a physical server that's running multiple virtual machines (VMs). The VMs may be hosting containers that are running different processes and connected to different subnets. A graph database can be used to see how all these components interconnect.

System administrators can also use a graph database to maintain a map of all the different network assets. This map can be used to better secure the network and to detect vulnerabilities or to limit the spread of an intrusion.

Penetration testers and security admins use Neo4j when they use Bloodhound, an open source pen-testing tool. Bloodhound tracks Active Directory permissions and is used by both blue and red teams to uncover problems in Active Directory security as well as potential attacks. Because of Neo4j's index-free adjacency, users get predictable query response times regardless of the size of the database.

Despite the variety of challenges faced by the retail industry and the fact that they touch different parts of the organization, these challenges share an underlying commonality: The need to understand connected data relationships.

Neo4j: Benefits and Key Features

Despite the variety of challenges faced by the retail industry and the fact that they touch different parts of the organization, these challenges share an underlying commonality: The need to understand connected data relationships. A graph database such as Neo4j does exactly that.

Neo4j is the number one database for connected data, thanks to its emphasis on ***native graph storage*** and ***real-time query processing***. Its real-time performance when traversing data, ACID-compliant transactions for data reliability, flexibility of data modelling and a wide array of developer productivity tools means retail organizations deliver powerful new insights from data connections with greater agility and at lower cost than previously possible.

Specific benefits of Neo4j include:

- **Minutes-to-milliseconds query performance of traversals.** Neo4j can traverse any level of data in real-time due to its native graph architecture. RDBMS and other NoSQL databases typically see a significant performance degradation when traversing data beyond three levels of depth.
- **Data Integrity.** Neo4j is a fully ACID transactional database for storing your critical data. Neo4j's fully ACID transactions and enterprise-grade Causal Clustering ensures data integrity of the graph at all times, making it the ideal database for any kind of mission-critical application.
- **Flexibility.** With Neo4j, your development team doesn't need to struggle to fit your data into predefined tables, and when business requirements change, the data model can flexibly adapt in minutes – not months.
- **Developer Productivity.** Neo4j delivers unparalleled developer productivity tools to build graph applications. With tools including Cypher – the industry's most intuitive and expressive graph query language – along with official language drivers, stored procedures and native Java APIs, developers can build and maintain graph applications with greater efficiency.

Specific technological innovations that make Neo4j so powerful include:

- Enterprise-grade Causal Clustering
- Index-free adjacency enabling real-time queries evaluating millions of relationships
- Advanced security architecture

Neo4j can be used either as a central database platform or alongside an existing database to add graph capabilities to new or existing applications.

The Neo4j Ecosystem: Accelerating Speed to Market

When companies choose Neo4j, they become part of a passionate, thriving community of hundreds of thousands of users and developers who inspire and support each other, sharing skills and accelerating delivery. Neo4j has more than 200 customers that include UBS, Cisco, HP and Lufthansa, in addition to a whole range of exciting startups.

Neo4j, the world's leading graph database, enables retailers to address the broad range of challenges described in this white paper. Existing systems represent data in tables and columns, making it hard to trace connections across datasets. Queries that take hundreds of lines of SQL can be reduced to less than 10 lines of Neo4j's intuitive Cypher query language. Better still, answers to those queries are returned in milliseconds, not hours.

Using a graph database such as Neo4j in conjunction with or in place of a relational database management system or NoSQL database can help retailers benefit from a sustainable competitive advantage.

Using a graph database such as Neo4j in conjunction with or in place of a relational database management system or NoSQL database can help retailers benefit from a sustainable competitive advantage.

Conclusion

Retailers face a number of challenges, largely from tech-savvy, online retailers. From delivering real-time product recommendations to dynamic pricing and optimized delivery routing, retailers must overcome these challenges quickly in order to remain viable — and achieve their own competitive advantage.

Retailers must also achieve greater agility in order to respond to changing consumer and technology trends ahead of their competitors. As the world's leading graph database, Neo4j enables retailers to understand connected data and leverage relationships at a greater depth than traditional techniques and technologies to serve real-time product recommendations, optimize delivery routing, and much more.

To learn more about adding Neo4j to your environment, contact a Neo4j expert or download a 30-day free trial of Neo4j Enterprise Edition.

Questions about Neo4j?

Contact us around the globe:
info@neo4j.com
neo4j.com/contact-us