

# THE GRAPH DATABASE BUYER'S GUIDE

*Important Considerations  
For Choosing the Right  
Graph Technology*



# Table of Contents



<b>Introduction</b>	<b>1</b>
Make Way for the Graph Database	1
Industry Leaders Are Driving Graph Adoption	2
Relational Databases Are Rigid and Brittle	3
NoSQL Sounded the Early Warning	3
Graph Databases Are Built for the Modern World	4
Graph Models Are Based on Data Connections	5
Benefits of Graph Databases	6
<b>Selecting the Right Graph Database</b>	<b>7</b>
Native Graph Storage and Processing	7
Property Graphs	8
Graph Query Languages	8
Data Ingestion and Integration	9
Development Tools	9
Graph Visualization	10
Graph Data Science	11
OLTP, OLAP, and HTAP Application Support	11
ACID Compliance, Durability, and Consistency	12
High Availability	12
Scalability and Performance	12
Enterprise Security and Privacy	13
Deployment Flexibility	13
Open Source Foundation and Community	13
Business and Technology Partners	14
Vendor Credibility and Resilience	14
Neo4j: The Right Graph Database for the Right Reasons	15
For More Information	15



# Introduction

## Make Way for the Graph Database

Graph technology is one of the hottest sectors in modern data management, rapidly taking its share of application development in the database world. In the short run, Gartner® anticipates that the application of graph technology will grow at 100 percent annually through 2022.<sup>1</sup> As the growth continues, Gartner® predicts that 80 percent of data and analytics innovation will be made using graph technology by 2025.<sup>2</sup>

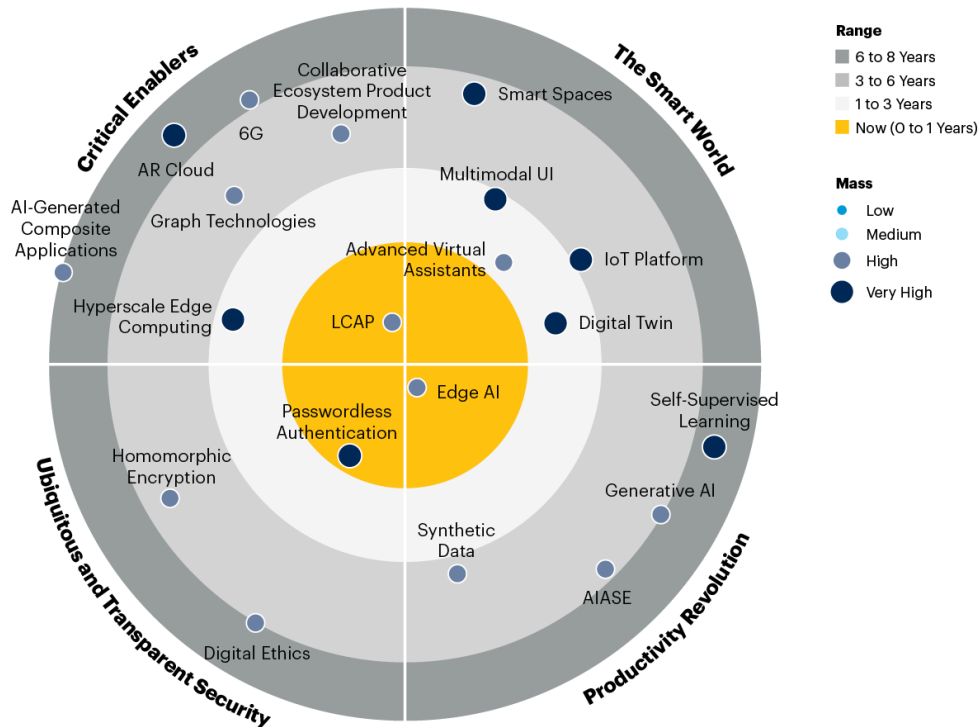
Adoption of graph technology is accelerating because of its unique ability to connect data across diverse sources. Gartner® frames its recommendation this way: "Add business value to your solution by using emerging technologies and trends such as graph technologies to store, manipulate, and analyze relationships between entities."<sup>3</sup>



- 
- 1 Gartner, Predicts 2022: Data and Analytics Strategies Build Trust and Accelerate Decision Making, 2 December 2021, By Jorgen Heizenberg Et Al.
  - 2 Gartner, Gartner Identifies Top 10 Data and Analytics Technology Trends for 2021 (press release), 16 March 2021, By Rita Sallam Et Al.
  - 3 Gartner, Emerging Technologies and Trends Impact Radar: 2022, 15 November 2021, By Tuong Nguyen, Danielle Casey Et Al. GARTNER is a registered trademark and service mark of Gartner, Inc. and/or its affiliates in the U.S. and internationally and is used herein with permission. All rights reserved.
- 
- 

In its Impact Radar for 2022, Gartner® categorizes graph technologies as critical enablers (see upper left of graphic).

### Impact Radar for 2022



Source: Gartner  
749318\_C

Gartner

This graphic was published by Gartner, Inc. as part of a larger research document and should be evaluated in the context of the entire document. The Gartner® document is available upon request from <https://neo4j.com/whitepapers/gartner-emerging-tech-and-trends-impact-radar-2022/>.

Gartner® does not endorse any vendor, product, or service depicted in its research publications and does not advise technology users to select only those vendors with the highest ratings or other designation. Gartner® research publications consist of the opinions of Gartner's research organization and should not be construed as statements of fact. Gartner® disclaims all warranties, expressed or implied, with respect to this research, including any warranties of merchantability or fitness for a particular purpose.

GARTNER is a registered trademark and service mark of Gartner, Inc. and/or its affiliates in the U.S. and internationally and is used herein with permission. All rights reserved.

### Industry Leaders Are Driving Graph Adoption

Several years ago, industry leaders recognized the power of graph technology as the driver of their businesses. Early adopters included Google, Facebook, LinkedIn, eBay, NASA, Microsoft, Airbnb, and Uber, who use graph systems to break new ground and rise above competitors. Before long, Walmart and other established vendors began using graph databases to improve customer experiences and establish their online presence.

**Today, graph-powered applications are used by:**

- Over 75% of the Fortune 500
- All of North America's top 20 banks
- 8 of the top 10 insurers
- 7 of the world's top 10 retailers
- 3 of the top 5 aircraft makers
- 8 of the top 10 automakers
- 3 of the world's top 5 hotels
- 7 of the top 10 telcos

If you plan to join these forward-thinking industry leaders, you must select a graph database from the confusing array of technologies claiming to have graph functionality.

*This guide is meant to help you make informed decisions about selecting a graph database for your organization. After an overview of graph concepts, you'll learn about key graph database functions, as well as issues you should investigate with vendors as a part of your selection process.*

**Relational Databases Are Rigid and Brittle**

The modern world is a whirlwind in which data is being created at exponential growth rates and data connections appear and dissolve dynamically. Relational databases were built for simpler times when data was stable and structured. Their rigid data models often don't meet the new data demands of today's organizations.

Relational systems manage data in rows and columns, but that's not how information exists in the real world. Modern datasets contain rich objects and context that includes their relationships with other objects. The value from that kind of deeply connected data is lost when relational approaches put data into columns and rows, which strips away nearly all contextual meaning.

Relational databases also depend on predefined relationships that don't easily adapt – they are simply too rigid to handle fast, unpredictable changes.

Each small change and addition to a relational database – whether triggered by internal or external events – causes time-consuming and costly schema redesigns. Those disruptions translate directly into lost revenues, higher costs, slow reaction times, and missed opportunities.

Does all this mean that relational databases are dead? No, they do serve certain functions well. But for deep, dynamic dives into your data you need a state-of-the-art system so your organization is better-equipped to adapt to today's ever-changing business environment.

**NoSQL Sounded the Early Warning**

The early adoption of NoSQL database technology was the first sign that systems professionals recognized the limitations of SQL-based, relational approaches. Developers turned to NoSQL systems to address the requirements of basic store-and-retrieve applications by distributing information across data volumes and clusters. The document, wide-column, and key-value data models of traditional NoSQL databases helped developers build simple, scalable applications.

However, the simplicity of traditional NoSQL technology ignores data relationships and the integrity and consistency of connected data. As a result, NoSQL databases fell short of addressing the needs of applications based on contextually rich datasets. The stage was set for a new technology to emerge, one able to power intelligent applications via data relationships.

2017+

## Graph Database

**Flexible, scale-out store/retrieve + analytics**

Optimized for data connections, real-time transactions and analytics

**Natively store & query relationships**

Queries run 1000x faster at scale, up to 1000+ hops

First database language standard in 40 years

**GQL**

2010+

## NoSQL Database

**Scale-out simple model store/retrieve**

Optimized for application-driven, volume throughput

**App Based**

1980+

## Relational Database

**Structured, schema-based store/retrieve**

Optimized for real-time transactional processing/reporting

**SQL**

The evolution of database platforms to meet changing requirements.

### Graph Databases Are Built for the Modern World

[Graph databases](#) easily capture and leverage the dynamic patterns and connections that drive modern business. And they're built for today's datasets, which are in constant motion as companies add and combine new information sources and experiment with different data models.

Companies are adopting graph technology to access real-time transaction analytics and AI/ML required to discover the drivers of their business. Graph technology brings accurate, reliable intelligence to everyday business decision-making to improve business workflows and increase opportunities.

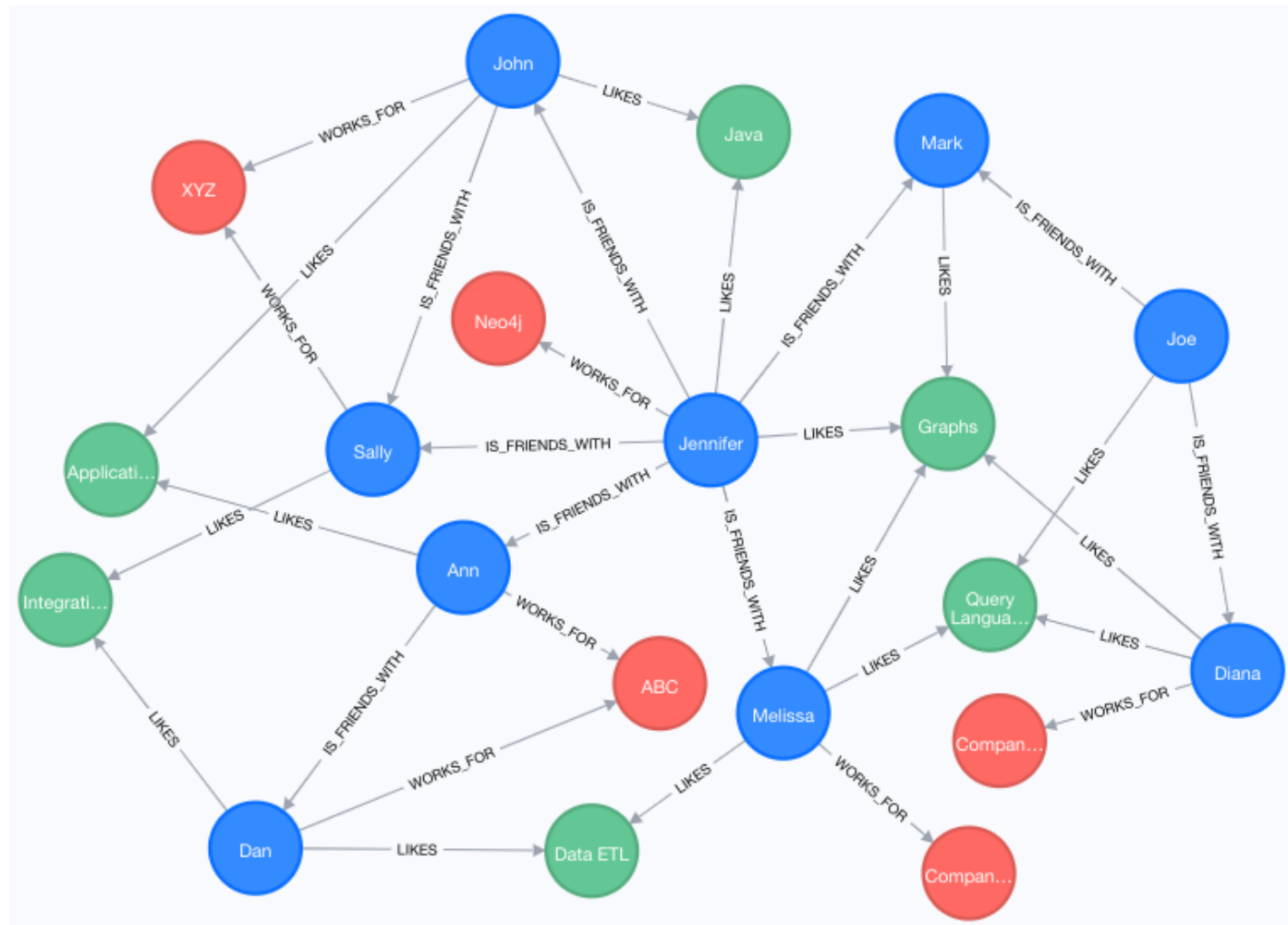
Graph technology's ability to handle unstructured, changing data makes it ideal for computing the complexities and challenges of a [variety of applications](#), including:

- Real-Time Recommendations
- Social Network Publishing
- Artificial Intelligence/Machine Learning
- Knowledge Management
- Customer 360
- Fraud Detection and Analytics
- Supply Chain Management
- Network Systems Operations
- Cybersecurity
- Digital Transformation
- Internet of Things
- Identity and Access Management
- Master Data Management
- Metadata Management

The broad applicability of graph databases like Neo4j and their ability to handle the demands of modern, connected datasets have earned them a crucial role in the enterprise technology stack.

### Graph Models Are Based on Data Connections

Graph data models are based on data elements and their **associated connections**. Those models are represented in **property graphs**.



Graph data models capture connections.

Data elements can be entities of all kinds – people, products, suppliers, objects, buildings, vehicles, subassemblies, and more. In a graph model, those elements are connected by data relationships, which explain how the elements are interrelated. The connections provide invaluable context such as friendship, ownership, membership, hierarchy, location, preference, and a world of other relationships.

Property graphs are simple representations that have significant advantages over traditional database models, namely that they work in the same way a human mind thinks – in relationships.

**With property graphs, you can:**

- Draw and use them directly as graph data models
- Program them quickly using graph languages that understand relationships
- Easily visualize and navigate them by following their relationship paths
- Update data models easily since they avoid the complex overhead of traditional database schemas

Also, and most importantly, property graphs in native graph database engines deliver real-time performance at scale.

**Benefits of Graph Databases**

Based on the very latest in data science and software engineering breakthroughs, graph databases offer significant advantages that include:

- Simple, natural data models that mirror your business reality
- Adding as many business entities, relationships, and rules as you need
- Flexibility for evolving data structures that support agile development
- Real-time updating while users simultaneously access graph data
- Better querying and analytics that provide faster, richer insights



## Selecting the Right Graph Database

Selecting the right graph technology for your organization can be daunting. The remainder of this document describes what to look for in graph databases, related tools, and the vendors who sell and support them.

Each vendor you consider should be able to explain how they address all these areas because they are crucial to your success with graph database technology.

### Native Graph Storage and Processing

**Native graph storage** is fundamental to the integrity and performance of graph databases. It ensures that real-world relationships that connect graph nodes are stored as primary, persistent data elements.

Native graph data models map directly to the way the business works. They capture product data models required by business applications, for example. As a result, application development becomes more straightforward and intuitive.

Without native graph storage, relationship information can be lost, disconnected, or abandoned, causing data corruption that breaks the central navigation system of the database. It's like using a GPS that leads you down a one-way street whose bridge has washed out, leaving you stranded.

Non-native databases imitate graph functionality by transforming graph data and requests into their own native column or document paradigm. Those extra code layers cripple query and application performance. They hide nuances and errors in data transformations, and quietly allow or even cause graph data corruption. Even after all this extra effort to mimic graph functionality, non-native graph databases often require complex queries and joins to produce the required results.

As part of your evaluation, find out whether the graph database performs **native graph processing** or if non-native transformations are hidden in its software.

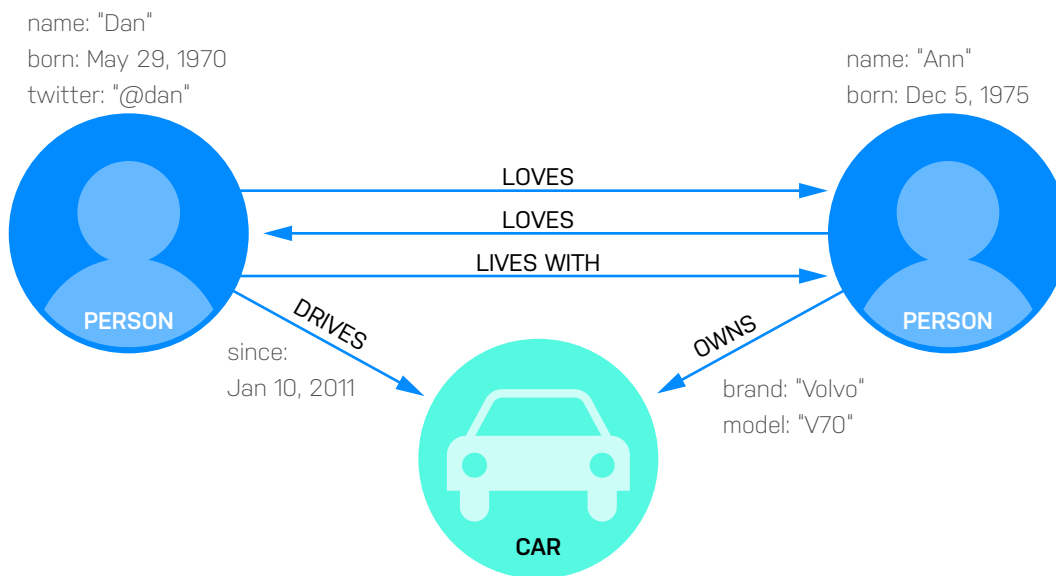
**Index-free adjacency** enables lightning-fast traversals across complex graph datasets. The database stores pointers from one node to the next. Such high-speed, predictable performance is achievable only in native graph environments. In sharp contrast, non-native graph approaches degrade the performance, scalability, and reliability of graph applications.

*Neo4j has 100% native graph storage and processing with index-free adjacency. No graph database does more to preserve the integrity of your data while maximizing query performance and scalability. Neo4j's native graph storage and processing provide an agile foundation for breakthrough applications that reduce TCO and deliver bottom-line returns faster and at lower risk.*

## Property Graphs

**Property graphs** power graph databases. The data elements and relationships (nodes and connections) in property graphs are both treated as first-class, primary data elements; the relationships are every bit as important as the data elements. Together, they provide a semantic foundation of connected data that models how a business works – who knows whom, what they like, where they shop, who supplies which materials, which parts are used in which products, and so on.

Data elements and associated relationships have properties that provide detail and context to support rich queries.



A simple property graph data model.

When an application queries a graph model, the database traverses the paths (the relationships) that connect the nodes (the elements) until it completes the query and returns matching results to the application.

*Neo4j provides flexible native property graphs that enable you to prototype new data models easily and adapt them as business needs change. Other graph databases use fixed property graphs or traditional Resource Description Framework (RDF) formats that often require schema rewrites, complex query coding, and long development cycles for even the slightest changes.*

## Graph Query Languages

SQL is the standard relational query language, but it's designed for row-and-column queries. It's simply too inefficient and complicated to adapt to the graph paradigm. Though several proprietary languages have evolved for graph database queries, two have come to the forefront and are used in most graph database vendors' offerings: Cypher and Gremlin.

Nearly all graph database vendors directly support Cypher, Gremlin, or both. And the ones that only support Gremlin can use Apache's Cypher for Gremlin project to run Gremlin code against

databases and tools that support Cypher. As a result, there is a concerted effort in the ISO community to recognize a **Cypher-based standard language** as a sibling language to SQL.

**Declarative languages** such as SQL and Cypher – in contrast to imperative languages like Java – are easier to learn, write, read, and debug. They are also easier to use since developers simply tell the database what to retrieve and not how to do so. Without Cypher, developers must be intimately familiar with their database's graph schema and specify how to execute each graph traversal, which increases their learning curve and the complexity of their code.

Some query languages must be compiled into binary packages prior to their execution. While compilation can optimize performance, it makes writing, building, and executing queries more complex and eliminates the ability to debug and change queries on the fly.

*As the creator of the [Cypher query language](#) (openCypher), Neo4j uses declarative language in all its graph technologies. Your developers and users get the power, productivity, and ease of declarative queries in all their graph projects.*

## Data Ingestion and Integration

The integrity and usefulness of a graph database depends on datasets that are tightly integrated with your organization's operational systems. Therefore, powerful data ingestion and integration are crucial parts of any graph application platform.

While most databases import CSV files, make sure your chosen graph technology can also perform **high-speed and bulk ingestion**. It's also vital for a graph database to rapidly and dependably create relationships as the data loads.

Bulk ingestion processes should allow you to defer index building and consistency checking until all data is ingested, thereby speeding the loading and indexing processes.

Be cautious about high-performance ingestion claims from vendors since they often achieve such speeds by ignoring data integrity and causing data holes and lumps that skew analytics.

Since datasets change rapidly, verify that your chosen database can keep pace and **change, add, and remove data** and related relationships as needed.

To support **regular streaming of updates**, consider a graph database that supports open source tools such as Apache Hop or Kafka and commercial tools from Informatica, TIBCO, or Trifacta.

Have you invested in **big data technologies** like Hadoop? If so, make sure your graph database integrates easily with them, ideally using already deployed data processing technologies such as [Apache Spark](#).

If you use **data lakes** to feed data warehouses, or if you use the **Apache Spark** analytic processing engine, you should make sure your graph database also integrates easily with that source data.

*As today's most mature and widely used graph database, Neo4j supports all these advanced ingestion technologies and methods. You can integrate Neo4j into your infrastructure quickly and dependably while maximizing productivity of developers, DBAs, and users.*

## Development Tools

Given today's scarcity of developers, your chosen graph data platform should include a variety of tools that enable developers to create powerful graph applications quickly and dependably.

Such a strategy begins with **visual development environments** that enable software engineers to explore graphs, write clear queries, and understand their results. To maximize developer productivity, the environment should include:

- Keyword color-coding and auto-completed values
- A graph drawing tool to help visualize result sets
- A storyboard tool for teaching and sharing instructions

To build extensible, custom solutions, graph developers need **drivers and APIs** that enable them to load data into the graph database, integrate graph results into enterprise apps, and even embed graph databases into their applications.

Neo4j offers commercial support for Java, JavaScript, .NET, Python, and Go. The Neo4j community offers drivers for even more languages. In addition, Neo4j enables OEMs to embed graph apps complete with application logic, UI, and cloud containers in their offerings.

An ideal experience integrates all [developer tools](#) in a **developer launchpad** that provides a consistent, powerful place for developers to create, modify, and maintain their graph database systems.

*The Neo4j Desktop is an integrated launchpad for graph development efforts. It provides access to the Neo4j database, an extensive procedure library, and graph application examples.*

## Graph Visualization

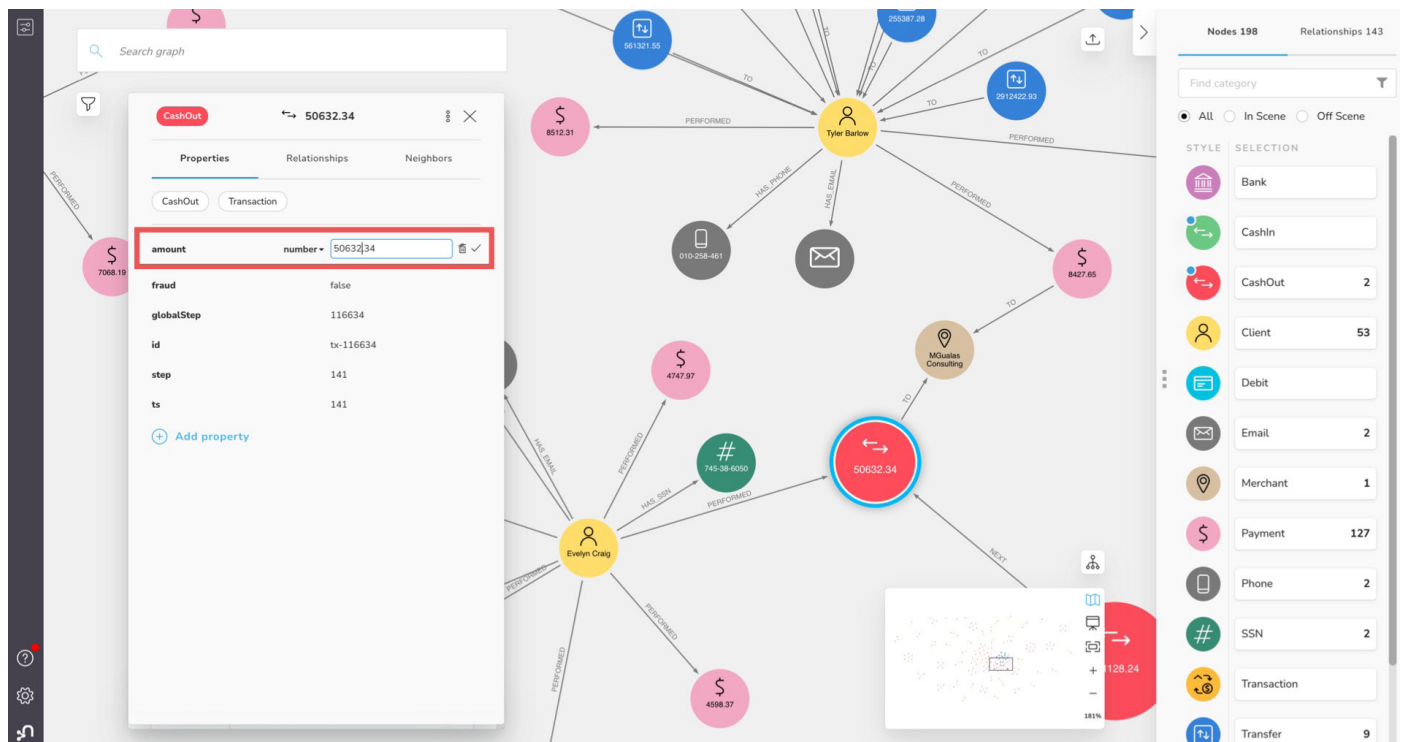
In addition to tools that support the software engineering lifecycle, graph platforms must address the needs of a diverse user base of developers, data scientists, DBAs, IT operations staff, and business users.

**Graph data visualization tools** enable users to quickly explore connections in graph data in a variety of ways, such as filtering nodes and relationships, lassoing parts of the graph, editing properties, highlighting paths, and customizing nodes.

Graph visualization tools should also enable users to:

- Dig deeper into the meaning behind graph patterns
- Share graph views and explore graph datasets
- Present and print graph models and content
- Support **Google-like, natural-language search** familiar to business users
- Learn graphs by offering built-in query suggestions and auto-completion

*Neo4j offers Neo4j Browser, a visual, web-based graph browser for developers, and [Neo4j Bloom](#), a codeless graph visualization tool that enables both graph technology novices and experts to explore their data. There are also other powerful Neo4j visualization tools offered by third-party enterprise software vendors and the graph software community.*



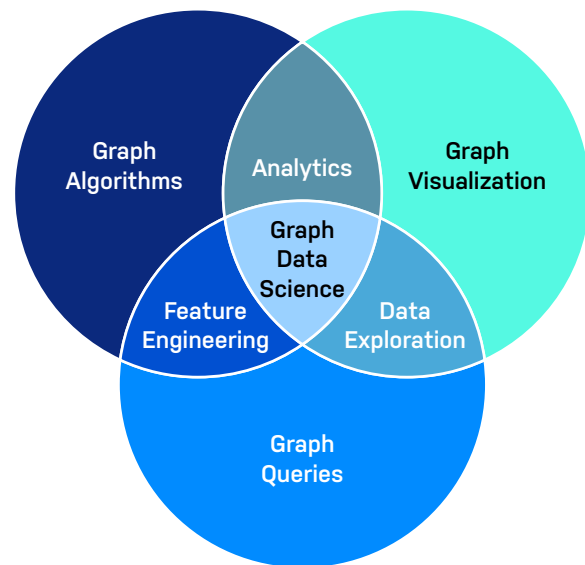
## Graph Data Science

Graph data science plays a primary role in the meteoric rise of graph database technology. Based on the real-world relationships that underlie all graph data, graph data science employs advanced graph analytics and algorithms as well as AI/ML to discover underlying drivers of your business that traditional analytics and statistics miss entirely.

**Graph algorithms** help users readily identify patterns, paths, clusters, and similarities in graph data to provide business insights and to support AI applications. Graph vendors should offer a full spectrum of graph algorithms and provide documentation and training on their use.

Modern organizations are using graph data science to cash in on their graph database investments with better, faster predictions and decisions that drive profits, product development, and customer satisfaction.

*Neo4j Graph Data Science includes powerful graph analytics, the world's most extensive library of graph algorithms, and the industry's only commercial graph embeddings. In contrast with cobbled-together solutions from others, [Neo4j Graph Data Science](#) is tightly integrated to maximize its power and effectiveness in production environments at unmatched scales of up to tens of billions of nodes.*



## OLTP, OLAP, and HTAP Application Support

Traditional database applications are generally divided into two functional categories:

- **Online Transaction Processing (OLTP)** applications that perform transactions for running a business
- **Online Analytical Processing (OLAP)** applications including data warehouses, business intelligence, data mining, and analytics

Graph databases are often used as **Hybrid Transactional-Analytic Platforms (HTAP)** that integrate transactional and analytical functions. To do so, graph platforms must support a wide spectrum of use cases, data interfaces, deployment environments, programming languages, and user skill sets.

When selecting a graph platform, it's crucial to discover the breadth of the technologies and applications with which it integrates. More importantly, you should know how richly it integrates with the applications you use to run your business.

*Neo4j integrates with a universe of transactional and analytic applications, so it easily becomes a vital part of your enterprise application strategy.*

### ACID Compliance, Durability, and Consistency

Due to the connectedness of graph databases, it is crucial that they complete all parts of every transaction without failure. This quality is known as **ACID compliance**, an acronym for Atomicity, Consistency, Isolation, and Durability.

When only part of a transaction completes but other parts fail, a graph database can be left in a corrupted state in which dangling relationships point nowhere, graph entities reference properties that don't exist, or graph nodes can be reached only from one direction. Subsequent graph updates and changes can easily spread the corruption, making significant portions of the database unusable.

Non-native graph databases such as key-value stores are particularly prone to transactional corruption because they support ACID only at a single-key level in their attempt to deliver high throughput. It's a recipe for disaster.

Incomplete transactions in non-ACID databases can cause graphs to omit relationships or, even worse, to imply relationships where none exist. Graph corruptions go undetected by subsequent queries because results look normal but are as corrupt as the relationships on which they are based. As a result, the most dangerous consequence of graph database corruption is unknowingly basing business plans and decisions on bad data.

*Neo4j is 100% ACID compliant to prevent graph corruption and preserve the integrity of your graph data and your resulting business decisions.*

### High Availability

Graph data platforms must offer a highly available solution that gracefully survives multiple system failures. This most often requires built-in application and data redundancy.

Some graph data platforms offer data redundancy over a single, writable graph. Others use multiple graphs to add **fault tolerance** to applications, networks, and operations. While multiple graphs are superior from a failure perspective, they introduce consistency challenges that stem from writing identical updates to multiple graphs.

Graph databases must **recover gracefully** from hardware failures, power interruptions, and other system stoppages that trigger transaction failures, lose data, and cause graph

inconsistencies and corruptions. Make sure your chosen graph database is durable and can recover fully from unforeseen interruptions of all kinds.

*Neo4j Causal Clusters adjust automatically to system failures to achieve high availability and recover fully to ensure graph integrity and consistency. In the event of disasters, operators can reconfigure running clusters to handle interruptions, maintain availability, and avoid data loss.*

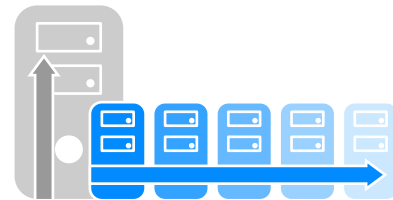
### Scalability and Performance

Scalability is a significant challenge for graph technology vendors. **Scaling graph writes** is a demanding task that requires distributing parts of a graph across servers, so it's easier to dedicate beefier hardware or opt for cloud deployment to maintain write performance.

**Read scaling** is significantly easier for graphs than scaling writes. The most efficient way to support large numbers of read requests is to replicate servers that store the graph.

**In-memory graphs** significantly outperform those that are not cached, providing faster query response for both reads and writes. It's important to understand if you can store all or part of your graph in RAM, as well as how the graph platform reinstates the graph in memory after the system restarts.

Some graph vendors answer all scalability questions with horizontal, scale-out, or cloud approaches, but few offer vertical scaling options as well. This lack of agility and flexibility quickly leads to costly application deployments.



*Neo4j's clustered architecture provides flexible horizontal and vertical scaling solutions and in-memory graph caching to simplify scaling and performance challenges. Neo4j also offers cloud-based DBaaS that frees you from solving those thorny problems on your own. This agile approach enables you to balance performance, cost, and complexity trade-offs to address the unique requirements of each system development project.*

## Enterprise Security and Privacy

Privacy and data theft are on every CIO's list of top concerns, so understanding the security features of your selected graph platform is paramount.

Knowing how a graph data platform performs **user authentication** is a good place to start. Does it integrate with LDAP or Active Directory? Does it support Kerberos for single sign-on? Does it support security groups?

**Data encryption** is crucial to any enterprise database deployment, so dig into the details of encryption support in the graph data platform. Is data encrypted at rest and in transit? Is it encrypted inside server clusters? Is data encrypted during admin functions?

**Privacy** is at the forefront of every modern cybersecurity strategy. Does the graph data platform offer role-based access control to secure sensitive data elements at the node, relationship, and property level? Can it physically store personal data by country to comply with GDPR and other privacy regulations?

**Implementing security** in your applications is just as important. Is data security transparent to developers or must they manage it within their apps? Are security events logged by the database?

*The Neo4j Graph Data Platform offers enterprise-grade security. Its multi-database capabilities ease compliance with privacy and security regulations. It includes role-based security all the way down to individual nodes, relationships, and properties.*

## Deployment Flexibility

One of the most critical factors in selecting a graph database is its deployment flexibility.

Make sure your selected technology supports **multiple graphs**, distributes them across clusters, and directs read and write requests to the correct cluster. As your needs grow, you might need to manage **multiple graph databases** on one server instance, so ask what vendors' plans are for such support.

You must also be able to license graph technology with terms that work for you. Verify that vendors offer a **variety**

of **license models** with published prices that support your deployment preferences.

*Are you interested in **cloud deployment** now or later? Does your selected graph platform run the same in the cloud as on-premises? Can you choose among Amazon, Google, Microsoft, and Alibaba cloud services?*

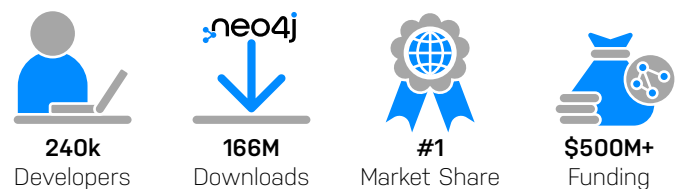


As cloud services continue their explosive growth, **graph database as a service (DBaaS)** grows even faster. Do potential graph vendors have plans for DBaaS? Can their graph platform even support it?

*Neo4j offers fully managed [AuraDB](#) DBaaS as well as on-premises, public cloud, and private cloud deployment, along with flexible licensing so you can choose the deployment method and licensing terms that are right for you.*

## Open Source Foundation and Community

The most successful infrastructure software contains a strong foundation of open source code supported by a global community of users. The graph database market started as an open source project from Neo4j over a decade ago. As with so many other modern technologies, the feedback, contribution, and support of the Neo4j open source community have helped to drive new features, third-party tools and support, a pool of skilled developers, and market share.



*Neo4j has the largest open source community in the graph database world with 166 million downloads and 240,000 developers. [DB-Engines](#) credits Neo4j with half of their popularity score for the total graph database market.*



## Business and Technology Partners

The extensibility, openness, and usability of a database platform is directly driven by **business and technology partners** that add tools, connectors, and services to the platform's ecosystem. Neo4j and its partners provide a variety of application frameworks and blueprint solutions that accelerate time-to-value.

Ask graph database vendors to describe their partnerships and how they work with those companies. The partners should include cloud vendors, app developers, graph consultants, implementation specialists, software OEMs, value-added resellers, regional resellers, and education centers. To learn more, visit the [Neo4j Partners page](#).

*As the first and most battle-tested graph database, Neo4j has a 50 percent share of the graph database market. It has the largest customer base, the largest and most active developer community, and more technology partnerships than any other graph vendor.*

## Vendor Credibility and Resilience

Not all graph database vendors are created equal. As you select a graph data platform, verify that its vendor has the technology, track record, services, and financial stability to play an important role in your enterprise technology stack.

The following are key areas to consider when selecting your graph database vendor.

The **financial stability** of your vendor is essential since it must continuously improve its software to keep pace with technology changes and customer demands. Look for substantial funding and proven financial partners.

The size of the **engineering team** dedicated to graph database and tools development is a direct indicator of the vendor's dedication to the graph market. Beware small

graph engineering teams and teams that are part of larger organizations that aren't dedicated to graph technology.

The vendor should also have a **dedicated services team** that offers comprehensive application development, onboarding, training, education, and support services. Vendors must have the resources to support your graph deployments, around the globe and around the clock. Your success with graph technology depends on it.

Demand **solid customer references** for each of your graph use cases from every vendor. Ask to talk to their customers. If vendors can't share customer references with you, they haven't solved your problem before. Look for customer lists, stories, use cases, and forums on vendors' websites.

**Events and conferences** strengthen your relationship with software vendors, supply a forum for expressing your requirements, and provide opportunities to meet experts, consultants, and prospective developers. They are also an indicator of a vendor's financial commitment to their technology and your success.

*Neo4j is the largest and most stable graph database company in the world, having raised over \$500 million in venture capital funding. Neo4j graph technology is used by more than 800 enterprise customers including 75 percent of the Fortune 100.*

*With more than a decade of production deployments and rigorous testing, Neo4j has thousands of deployments of hundreds of use cases in more than 20 industries – far more than any other graph vendor in the world.*

*Neo4j is backed by an ecosystem of dedicated, enterprise-class partners and customers, making Neo4j an even stronger choice for your graph technology provider.*



## Neo4j: The Right Graph Database for the Right Reasons

Given the vital role that technology analysts see for graph technology over the next decade, selecting the right graph database for your organization is among the most important decisions you can make.

### *By selecting Neo4j, you standardize on an enterprise graph solution that:*

- *Takes milliseconds to execute queries that require minutes or hours on other graph platforms*
- *Adapts to evolving business models and datasets rapidly and without interruption*
- *Scales to meet requirements of the very largest and most complex enterprise applications*
- *Often requires 10x less hardware than other graph databases*
- *Maximizes developer productivity with 10x less code than SQL*
- *Taps into the world's largest community of skilled graph developers*
- *Is already chosen by an overwhelming majority of global enterprises and industry leaders*

Neo4j's native graph architecture provides a foundation for new classes of breakthrough business applications that [deliver significant ROI](#). Its rich, proven graph data platform provides unrivaled business and technical benefits that make it the obvious low-risk, high-reward choice for modern connected data applications.

### For More Information

Ready to take the next step into the future of enterprise data management? Join one of our online conferences at <https://neo4j.com/connections/>. Or send us an email at [sales@neo4j.com](mailto:sales@neo4j.com) and let us know how we can help put the power of Neo4j to work in your organization.

Neo4j is the world's leading graph data platform. We help organizations – including [Comcast](#), [ICIJ](#), [NASA](#), [UBS](#), and [Volvo Cars](#) – capture the rich context of the real world that exists in their data to solve challenges of any size and scale. Our customers transform their industries by curbing financial fraud and cybercrime, optimizing global networks, accelerating breakthrough research, and providing better recommendations. Neo4j delivers real-time transaction processing, advanced AI/ML, intuitive data visualization, and more. Find us at [neo4j.com](#) and follow us at [@Neo4j](#).

Questions about Neo4j? Contact us around the globe:

[info@neo4j.com](mailto:info@neo4j.com)  
[neo4j.com/contact-us](https://neo4j.com/contact-us)