

Market Guide for Graph Database Management Systems

Published 30 August 2022 - ID G00753429 - 16 min read

By Merv Adrian, Afraz Jaffri

Graph is one of many new DBMS choices targeted at analytical, operational, data science and AI use cases. This research helps data and analytics leaders navigate the market for commercially available graph database management systems.

Overview

Key Findings

- Based on the increasing usage, popularity and maturity of graph, Gartner estimates that the market for graph technologies, including graph database management systems (DBMSs), will grow to \$3.2 billion by 2025 with a CAGR of 28.1%.
- Vendors in the graph DBMS market are expanding their stacks into platforms for enterprise knowledge graphs or graph artificial intelligence (AI), with associated product ecosystems.
- Data and analytics professionals often struggle to distinguish between different graph implementation models. However, the traditional split between RDF and property graph support is becoming less important for graph DBMS selection than features that support enterprise readiness.
- Native and multimodel graph databases can scale to meet enterprise production requirements, including stability, availability and security. Selection of either type requires balancing the potential increase in query speed against the increasing complexity of the data model.
- Managed as-a-service offerings significantly lower the barrier to entry for organizations that do not have graph expertise.

Recommendations

Data and analytics leaders responsible for implementing and evolving data management solutions, operational infrastructure and analytical infrastructure should:

- Develop a graph use case before choosing a graph DBMS by defining business value, sample queries, and transactional versus analytical processing requirements.
- Distinguish between RDF and property graph models only where a clear need exists by evaluating data exchange, ontology reuse, data science support and reasoning requirements.
- Assess enterprise requirements for graph DBMS product functionality by comparing security models, API support, ease of integration, configuration and optimization, and end-user tools for visualization, query and schema building.
- Test whether existing multimodel DBMS offerings can deliver required analytics, scalability and performance before adding another vendor and product combination.
- Deploy managed services where available to minimize the overhead of managing and maintaining graph DBMS, which may be unfamiliar to existing team members.

Strategic Planning Assumptions

By 2025, graph technologies will be used in 80% of data and analytics innovations, up from 10% in 2021, facilitating rapid decision making across the enterprise.

Market Definition

This document was revised on 31 August 2022. The document you are viewing is the corrected version. For more information, see the [Corrections](#) page on gartner.com.

Graph DBMSs enhance support for graph data models by providing data loading, data conversion, consistency, security and maintenance, along with the ability to provision clusters that scale up and scale out. Graph DBMSs may be either:

- **Transactional:** Optimized for retrieving small parts of one or multiple graphs
- **Analytical:** Optimized to perform operations on each node and edge in a graph, and aggregate results

Large traditional DBMS and platform vendors, as well as startups, are targeting these opportunities, because interest and momentum are demonstrably expanding (see Note 1). Data and analytics leaders in all vertical industries are supporting graph use cases, such as 360-degree views of customers, recommendation engines and fraud detection. (See Note 2 for some use-case examples.)

Market Description

Graph DBMSs may be classified by the way graphs are structured and queried:

- **Property graphs** (aka labeled property graphs [LPGs]) allow pairs of tuples (properties) to be added to any node or edge in a graph. Nodes can be grouped together using labels.

- **Resource Description Framework (RDF)** graphs explicitly model every node and edge combination as a set of triples of the form <subject> <predicate> <object>.

Use cases for each type of graph overlap, and the difference between the two types is becoming less important. RDF is widely implemented by vendors, overcoming some of the modeling complexity that previously was a barrier to adoption. Property graph vendors are adding support for partitioning graphs, adding rules and exporting graphs. Some products allow both graph formats to be used interchangeably.

Languages used for graph DBMSs differ from the SQL-based standard. For example:

- Apache TinkerPop provides a graph computing framework for transactional and analytical use cases. The Gremlin graph traversal language is developed under the auspices of TinkerPop.
- Cypher is an alternative to Gremlin. It is a declarative graph query language designed for use with property graphs.
- RDF-based graph vendors offer standard support for the SPARQL language.

Different vendors support different combinations of these and other languages — and all need to align with more commonly used tools to grow their market. The much-anticipated ISO/IEC Graph Query Language (GQL) and associated SQL to property graph (SQL/PGQ) standard — the first such standard in decades — will further enhance interoperability (see Note 3). Many vendors participate in this work, including the large enterprise DBMS vendors, which will use it as a way to capture more market share.

Graph DBMSs can be categorized as either:

- **Native DBMSs:** Native solutions may be more scalable and more applicable for resource-heavy processing involving real-time calculations, multihop queries and machine learning (ML). Product architecture can impact these attributes. That is, embedded, in-memory, disk-based and cluster-aware DBMSs will have different deployment and performance profiles.
- **Multimodel DBMSs:** These are typically offered as part of broadly deployed products that support graph as one data structure among others. The perceived advantages of multimodel are ease of integration with existing data, support for use cases that utilize different data types and the elimination of another vendor to manage.

Both native and multimodel graph databases can scale to meet enterprise production requirements, including stability, availability and security. However, native graph DBMSs offer performance gains when dealing with queries over very large graphs (typically billions of nodes).

The ability of marketplace offerings to support all these different attributes is increasing. Pure graph DBMSs are expanding their technology stacks to handle two main types of use cases:

- Data integration and knowledge management as enterprise knowledge graph platforms (EKGP)
- Graph analytics, data science and AI as graph AI platforms (GAIP)

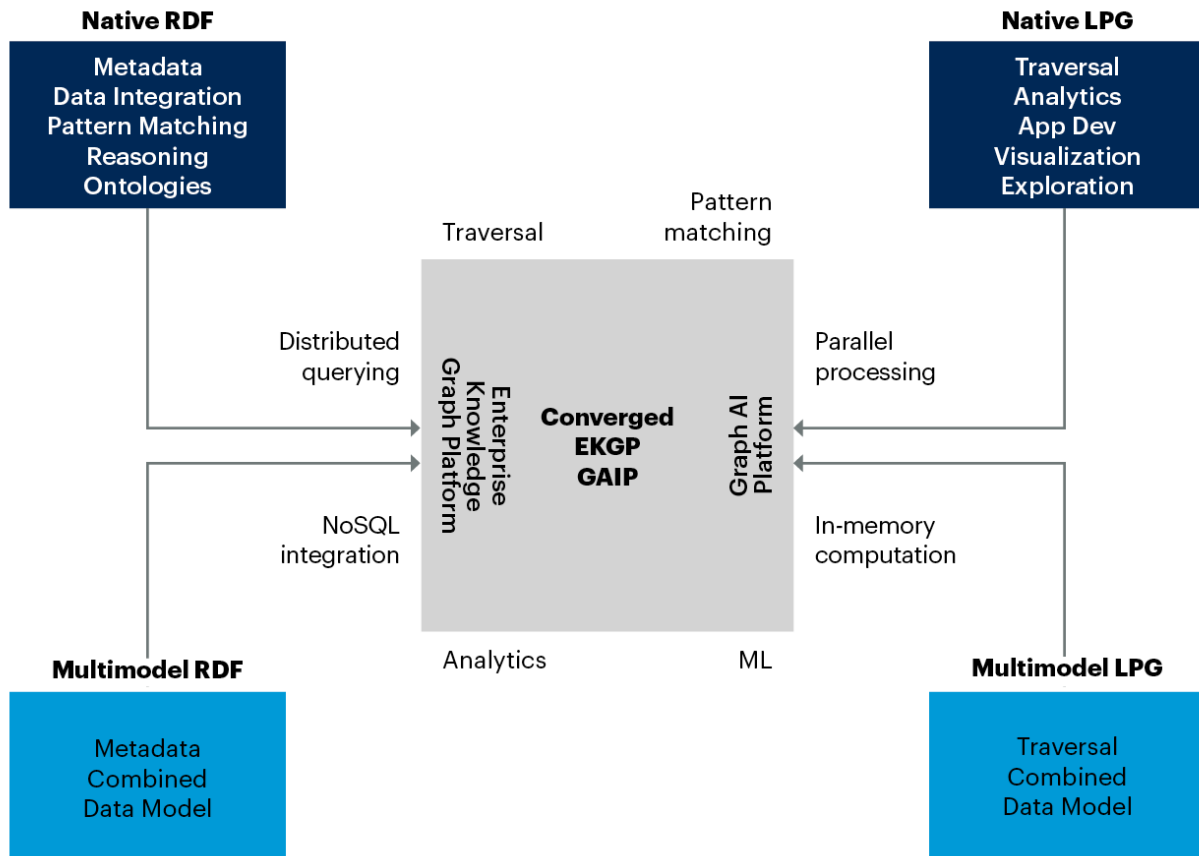
Finally, as is the case elsewhere in the DBMS market, some graph DBMS offerings are cloud-only. That is, they are purely as-a-service offerings, as opposed to hybrid on-premises DBMSs.

Not all products listed here will support all types of use cases or architectural choices equally well. Use Figure 1 to identify which features map best to the use cases you have in mind.

Figure 1: Convergence of Capabilities in the Graph DBMS Landscape



Convergence of Capabilities in the Graph DBMS Landscape



Source: Gartner
737853_C

Gartner

Market Direction

The following constituencies are driving the growth of graph DBMSs:

- Application developers are moving toward graph for an increasing number of customer- and internally facing projects, utilizing graph databases as the storage and execution back end. For an example, see [Case Study: Data and Analytics Monetization With Knowledge Graphs and AI \(Turku City Data\)](#).
- Data architects are designing knowledge-graph-based solutions for content management, personalization and semantic data interoperability. For an example, see [Case Study: Entity-Event Knowledge Graph for Powering AI Solutions \(Montefiore\)](#).
- Data scientists are needing to perform higher-order exploration into connections and relationships between data points for better insights. For an example, see [Case Study: Answering Critical Business Questions With Graph Analytics \(Jaguar Land Rover\)](#).
- Database designers and specialists are struggling to support users with relationship-based analytical requests, or are simply seeking better solutions to address growing volumes of semistructured data. See [Understanding When Graph Analytics Are Best for Your Business Use Case](#).
- Business owners and domain experts are seeking purpose-built tools for use cases best served by graph technology, such as decision intelligence, expert systems, 360-degree views of customers, fraud detection and investigative intelligence. For an example, see [Case Study: Augment Domain-Expert Decisions With Knowledge Graphs \(BDO UK\)](#).

All of these groups will benefit from the accelerating innovation in the graph DBMS space. Increasingly, end users are looking to reuse existing data, as migration tools for data from relational sources, object stores or even rival graph vendors are becoming common. Not surprisingly, as cloud platform providers enter the market, products like Amazon Neptune will drive more of this activity to attract users to the cloud as an ideal platform for experimentation. Hardware acceleration and hardware-based graphs will also dramatically improve performance. Support for these options will more likely occur on cloud platforms than at user sites.

The longtime competition among languages has been a challenge. However, it has also resulted in both significant enhancements to product capabilities and continuing efforts to promote standards such as GQL. Vendors are supporting GraphQL as an API so that developers can access the graph data model without learning SPARQL or Gremlin to build

queries. Moreover, vendors of RDF graph DBMSs are adding support for openCypher or Gremlin to support graph traversal. As acknowledged in the [Hype Cycle for Data Management, 2022](#), we expect these efforts to continue driving graph DBMSs through the Trough of Disillusionment to the next level of adoption over the next three to five years.

Market Analysis

Interest in and use of graph have been increasing steadily. In 2020 and 2021, inquiries to Gartner data management analysts about graph rose substantially. Among vendors mentioned, five of the top eight were not graph DBMS specialists, although the top two were. This demonstrates a continuing lack of awareness of the specialist vendors listed in this Market Guide, even among users interested in applying graph analysis. However, graph DBMS vendors are reporting increasing interest and visibility as their revenue and pipelines continue to grow, and rising cloud deployments are accelerating the trend further. Cloud service providers (CSPs), like other DBMS market leaders, are adding graph capabilities to their own DBMSs and making competitive products available on their platforms. The revenue leader among graph DBMS specialists, Neo4j, crossed the \$100 million revenue line in 2021.

Cloud plays dominate new DBMS deployments overall. Not all of these dominant vendors are competing aggressively in the graph DBMS space yet, but that status will change as the market grows. Gartner expects the percentage of revenue attributable to cloud in the overall DBMS market to exceed 50% by 2023 (see [Forecast: Public Cloud Services, Worldwide, 2020-2026, 2Q22 Update](#) and [Forecast: Enterprise Infrastructure Software, Worldwide, 2020-2026, 2Q22 Update](#)). Trends that have become inherent to the overall DBMS market are manifesting themselves in the cloud DBMS market as well. In both cases, at least 80% of the revenue is attributable to a handful of dominant vendors (see Figure 2).

Figure 2: DBMS Market Dynamics



DBMS Market Dynamics Overall and Cloud

Overall DBMS Revenue (Millions)				Cloud DBMS Revenue (Millions)				
		2021	2020			2021	2020	
82%	Microsoft	19,306.0	15,973.6	AWS	19,245.3	13,518.5	80%	
	AWS	19,245.3	13,518.5	Microsoft	6,870.2	4,711.9		
	Oracle	16,532.2	15,629.0	Google Cloud	5,211.1	3,177.4		
	Google Cloud	5,211.1	3,177.4	Alibaba Cloud	1,833.9	1,380.9		
	IBM	4,484.3	4,481.6	Oracle	1,183.2	876.8		
	SAP	4,018.6	3,797.0	SAP	841.5	655.8		
	Alibaba Cloud	1,833.9	1,380.9	Huawei	705.7	439.7		
	Teradata	1,072.3	920.5	Tencent	683.0	431.3		
				Snowflake	608.2	326.9		

Source: Gartner
753429_C



As the incumbent CSP services begin to take market share, they will be a formidable barrier for small vendors that will have to both partner with and compete with them. As with other specialty markets, the small vendors' agility and concentration on specific requirements, such as enterprise knowledge graphs, domain solutions, or analytics and AI, will help them stay ahead. However, the small vendors will also have to focus on integration with other systems and prove enterprise-level readiness.

Graph DBMS specialist offerings will need to demonstrate differentiation, leaning on hybrid deployment capabilities for cases where data gravity keeps some data on-premises. They will need to clearly articulate why a new and different technology should be a part of your data and analytics portfolio. Choosing to add an additional product to your toolkit should begin with an assessment of the capabilities of your existing products, which may be suitable for smaller, first-use graph cases.

Further advancements will occur in the following areas:

- Scale (via parallel computation and GPU utilization)
- Low-/no-code querying and schema development
- Platform capabilities for application development
- Data transformation and ingestion
- Graph visualization
- Graph data science and machine learning
- GraphOps for graph data models

Developing these capabilities, bringing them to market and supporting them will require commercial success and investment. Data and analytics leaders must recognize that not all the vendors listed here will achieve sufficient scale and operational effectiveness to succeed. Moreover, at this still-early stage of market development, vendors have formidable barriers to overcome, including:

- Identifying use cases/raising user awareness
- Addressing concerns about vendor lock-in
- Finding a pricing model and level that prospects will accept
- Building a partner and service provider ecosystem
- Supporting and extending production deployments

Representative Vendors

The vendors listed in this Market Guide do not imply an exhaustive list. This section is intended to provide more understanding of the market and its offerings.

Market Introduction

All the vendors listed in Table 1 offer commercial graph DBMS products. We have not included open-source offerings without commercial support.

Table 1: Representative Vendors in Graph DBMS			
Vendor ↓	Product Name ↓	Native or Multimodel ↓	Query Language ↓
Amazon Web Services	Amazon Neptune	Native	Gremlin, SPARQL, Cypher
Ant Group	TuGraph	Native	Cypher, Gremlin
ArangoDB	ArangoDB	Native	ArangoDB Query Language (AQL)
Bitnine	AgensGraph	Multimodel	Cypher, SQL
BlackSwan Technologies	ELEMENT	Native	SPARQL
CreateLink Technology Co.	Galaxybase	Native	Cypher, Gremlin
Cambridge Semantics	Anzo	Native	Cypher, SPARQL
DataStax	DataStax Enterprise Graph	Multimodel	Gremlin, GraphQL, gRPC, CQL, Spark SQL
Dgraph	Dgraph	Native	DQL, GraphQL
FactNexus	GraphBase	Native	Bounds

<i>Vendor</i> ↓	<i>Product Name</i> ↓	<i>Native or Multimodel</i> ↓	<i>Query Language</i> ↓
Franz	AllegroGraph	Multimodel	SPARQL, SPARQL*, GraphQL, Prolog/Datalog, Lisp, JIG/Gremlin, domain-specific languages
Haizhi Stargraph Technology	AtlasGraph DB	Native	openCypher
IBM	IBM Db2	Multimodel	Gremlin
MarkLogic	MarkLogic Server	Multimodel	JavaScript, Optic, Search, SPARQL, XQuery
Memgraph	Memgraph	Native	Cypher
Microsoft	SQL Server, Azure Cosmos DB	Multimodel	Gremlin
Neo4j	Neo4j	Native	Cypher, Gremlin, GraphQL, SQL
Objectivity	InfiniteGraph	Native	DO
Ontotext	Ontotext GraphDB	Native	SPARQL, SPARQL*, GraphQL, SQL
OpenLink Software	Virtuoso Universal Server	Multimodel	SQL, SPARQL, SPASQL, XPath, XQuery
Oracle	Oracle Autonomous Database	Multimodel	PGQL, SPARQL
Oxford Semantic Technologies	RDFox	Native	SPARQL
Profium	Profium Sense	Native	SPARQL
Redis	RedisGraph	Native	Cypher
Stardog	Stardog Enterprise Knowledge Graph Platform	Native	GraphQL, SPARQL, SQL
TIBCO Software	TIBCO Graph Database	Native	Gremlin
TigerGraph	TigerGraph DB	Native	GSQL
Transwarp	Transwarp StellarDB	Native	Transwarp Extended - OpenCypher
Ultipa	Ultipa Graph	Native	UQL
Vaticle	TypeDB	Native	TypeQL
VelocityDB	VelocityGraph	Native	LINQ

<i>Vendor</i> ↓	<i>Product Name</i> ↓	<i>Native or Multimodel</i> ↓	<i>Query Language</i> ↓
Vesoft	Nebula Graph	Native	nGQL
CQL = Cassandra Query Language; DQL = Dgraph Query Language; nGQL = Nebula Graph Query Language; PGQL = Property Graph Query Language; SPASQL = SPARQL within SQL; UQL = Ultipa Query Language			

Source: Gartner (August 2022)

Market Recommendations

- Differentiate graph use cases between those that require transactional processing and those that require analytical processing by defining query structures for partial graph traversal and entire graph processing. Choose the DBMS accordingly.
- Determine if there is a clear requirement for using RDF as the graph data model by looking at schema separation, ontology reuse, inference and universal identifiers. Without these characteristics, the choice of graph data model is arbitrary.
- Assess usability aspects of graph DBMS tooling, including ease of visual schema building and querying, system integration, and management.
- Test whether existing multimodel offerings can deliver required analytics, scalability and performance before adding another vendor and product combination.
- Deploy managed services, where available, to minimize the overhead of managing and maintaining graph DBMS, which may be unfamiliar to existing team members.

Acronym Key and Glossary Terms

Edge	An edge is a connection from one node to another node in the same graph structure. This connection type could be either implied (without a label) when all edges represent the same connection (e.g., messages passed on a network, friends in a social network) or explicitly defined when there is more than one type (e.g., different message types, connections between people and content). In addition, edges can carry weights representing the strength of the connection or some other measure. Edges can also have direction.
EKGP	An enterprise knowledge graph platform is a suite of tools to analyze data from structured and unstructured data sources, and construct schemas and ontologies; automate the population of knowledge graphs according to those schemas; and provide querying capability and services for application development.
GAIP	A graph AI platform provides a suite of tools for constructing a graph data model; running in-database graph algorithms; and constructing ML models, including graph-based feature engineering and graph representation learning.
GQL	GQL is a collaboration between vendors, specialists, commercial organizations and academics to create a standard query language for property graph implementations. ¹
GraphQL	GraphQL is a query language for APIs that uses a syntax describing how to ask for data. It can be offered by many different databases and services, not just graph DBMSs. ²
Hop	A single hop takes a starting node and follows all edges from that node to adjacent nodes that are directly connected. Multihop paths are those that take subsequent hops from the adjacent nodes to nodes that are further away from the starting node.
Node	A node is a single data point in a graph data structure.
Ontology	An ontology describes a domain of interest at a conceptual level. It can be thought of as a richer form of data schema that can include rules, restrictions and complex relationships. An ontology can be modeled as a graph using the World Wide Web Consortium (W3C) standards for OWL 2, which include a number of subsets implemented by graph DBMS vendors. ³ Using an ontology to infer and add nodes and edges (triples) to a graph based on the defined rules is known as “inference.”
RDF	Resource Description Framework is a W3C standard for knowledge representation using “<subject> <predicate> <object>” triples that can be represented as a graph and serialized in various formats. ⁴
Semantics	The concepts, rules and relationships described in the graph for a particular domain are collectively referred to as semantics.
Traversal	Traversal refers to taking a number of hops from a starting node to an end node or set of nodes, by specifying either the number of hops or the end node. A common computing framework used for graph traversal is Apache TinkerPop and the associated Gremlin language. ⁵

Evidence

Forecast Analysis: Artificial Intelligence Software, Worldwide.

¹ See [GQL Standard](https://gqlstandards.org), gqlstandards.org.

² See [A Query Language for Your API](https://graphql.org), GraphQL Foundation.

³ See [OWL 2 Web Ontology Language Primer \(Second Edition\)](https://www.w3.org/TR/owl2-primer/), W3C.

⁴ See [RDF 1.1 Primer](https://www.w3.org/TR/rdf1.1-primer/), W3C.

⁵ See [Apache TinkerPop](https://tinkerpop.apache.org), tinkerpop.apache.org.

Note 1: Representative Vendor Selection

The 32 vendors named in this Market Guide guide were selected to represent the types of solutions as discussed — native and multimodel, property graph and RDF, and on-premises and cloud — offered as commercial products with support.

Note 2: Some Use-Case Examples

Transactional use cases include:

- Complex join removal
- Recursive querying
- 360-degree view of the customer
- Recommendation
- Compliance monitoring
- Semantic search
- Data fabric
- Content management

Product features and architecture enable these use cases by supporting appropriate languages, which can vary by the architectural model chosen for the DBMS:

- PG-native or multimodel — using TinkerPop with Gremlin and vendor-specific query languages
- Triplestores native or multimodel — using RDF and SPARQL

Analytical use cases include:

- Route optimization
- Community detection
- Link prediction
- Anti-money-laundering
- Influencer analysis
- Threat detection (cybersecurity)
- Network analysis
- Supply chain optimization
- Graph DBMS (native or multimodel) with built-in functions/algorithms
- Graph DBMS (native or multimodel) that integrates with compute framework/BI tool
- Stand-alone compute framework

Note 3: Forthcoming ISO/IEC Standards

- [ISO/IEC CD 39075: Information Technology — Database Languages — GQL](#), International Organization for Standardization
- [ISO/IEC DIS 9075-16: Information Technology — Database Languages SQL — Part 16: Property Graph Queries \(SQL/PGQ\)](#), International Organization for Standardization

© 2022 Gartner, Inc. and/or its affiliates. All rights reserved. Gartner is a registered trademark of Gartner, Inc. and its affiliates. This publication may not be reproduced or distributed in any form without Gartner's prior written permission. It consists of the opinions of Gartner's research organization, which should not be construed as statements of fact. While the information contained in this publication has been obtained from sources believed to be reliable, Gartner disclaims all warranties as to the accuracy, completeness or adequacy of such information. Although Gartner research may address legal and financial issues, Gartner does not provide legal or investment advice and its research should not be construed or used as such. Your access and use of this publication are governed by [Gartner's Usage Policy](#). Gartner prides itself on its reputation for independence and objectivity. Its research is produced independently by its research organization without input or influence from any third party. For further information, see "[Guiding Principles on Independence and Objectivity](#)."