Build Your Applications on a Fully Managed Graph Database

Unlocking analytics and GenAI starts with connecting the data.

Why Graph Database?

Businesses increasingly need flexible databases to adapt to changing market conditions and requirements without significant downtime or restructuring. Companies today deal with a multitude of complex relationships that span various aspects of their operations. For instance, understanding customer interactions involves tracking and analyzing behaviors, preferences, and purchasing patterns across multiple channels such as online platforms, physical stores, and social media.

The in-depth analysis allows companies to tailor marketing strategies, improve customer service, and enhance overall customer experience. Relationships within supply chain networks are similarly complex. Supply chains comprise manufacturers, suppliers, distributors, and retailers, each with its own processes and data systems. Effective management of these relationships requires a comprehensive understanding of the flow of goods, inventory levels, transportation logistics, and potential bottlenecks.

Graph databases are engineered to handle interconnected data, making it easier to map and analyze relationships between discrete entities.

To navigate these intricate webs of relationships, businesses need robust data analytics tools that provide deep insights and accurate analysis. Graph databases excel in this regard because they are designed to handle interconnected data, making it easier to map and analyze the relationships between various entities. This capability is crucial for identifying patterns, predicting trends, and making informed business decisions. For example, graph databases can help optimize supply chain routes, detect anomalies that indicate fraud, and identify key influencers within a customer network.

The Limitations of Traditional Databases

Traditional approaches fail to uncover crucial connections within data. Relational databases, with their row-and-column organization, can't capture the complexity of real-world interactions — they only reveal partial insights. This lack of visibility makes it difficult or impossible to understand how individual data points are connected. The rigid schemas of relational databases create bottlenecks, leading to slow and complicated queries when exploring hierarchical, recursive, and many-to-many relationships. NoSQL databases, meanwhile, store disjointed documents and columns, missing the relationships entirely.

Traditional databases struggle to handle complex relationships for three principal reasons:

- **Complex joins:** Relational databases do not store relationships natively; they use joins to connect data. As data and the connections within it grow, maintaining joins becomes cumbersome.
- **Performance limitations:** As data volumes increase and query complexity grows, traditional databases struggle to maintain optimal performance.
- **Rigid architecture:** Relational databases are difficult to evolve. Bringing in new datasets typically requires refactoring the existing schema, which means that queries and application code need to be refactored as well.

Neo4j AuraDB Advantages

Power your Applications with Neo4j AuraDB

- Native Graph Model
- Flexible Schema
- Cypher Query Language
- Query Planner
- Productivity Tools

Model, Store and Query with Flexibility and Agility

- Match Patterns in Data
- Fast Traversals (Index-free)
- Diverse Workloads
- Enrich Predictions & Insights
- Visualize Graph Data

Uncover Hidden Patterns and Relationships

- Scalable Multi-Node Cluster
- 99.5% Uptime SLA
- Data Protection
- Tools Ecosystem
- Event-Driven Architecture

Highly Available with Zero Administration

neo4j auraDB

aws Google Cloud Azure

Neo4j AuraDB is a fully managed cloud database deployed on a scalable, secure, fault-tolerant, and highly available architecture, backed by a 99.95% uptime guarantee and point-in-time recovery. Uncover hidden patterns in data quickly — without joins — using Neo4j's Cypher query language and graph algorithms. Simplify development with pre-tuned algorithms, a flexible schema, and native integrations.

Design a data model that mirrors whiteboard conceptual sketches, with a native property graph model, a developer-friendly schema, and the expressive Cypher query language.

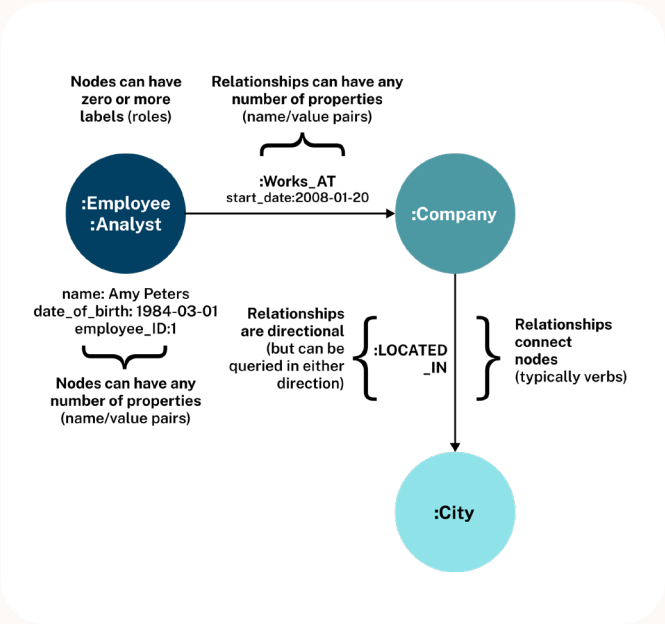
Build Applications With Ease

Native Graph Model

The Neo4j native graph model efficiently stores data and relationships in a format optimized for querying highly connected information. This model mirrors real-world data structures and conceptual designs, simplifying the transition from data design to implementation with a native graph data model enabling faster and more intuitive development.

Flexible Schema

Neo4j's flexible schema adapts to changing data and application requirements without necessitating database refactoring or rewriting application code. Unlike rigid schemas in traditional databases, Neo4j's schema can evolve as the data model grows, seamlessly accommodating new nodes and relationship types. The ability to handle schema changes dynamically ensures that Neo4j can adapt to evolving business needs and data landscapes with minimal disruption.



Cypher Query Language

Neo4j's Cypher query language is a powerful, intuitive tool designed specifically for querying graph data. It allows developers to easily express complex patterns and relationships within the data, using a syntax that is both readable and concise. Cypher's pattern-matching syntax visually represents nodes and relationships, making it straightforward to specify the exact connections in the query. **Cypher is shorter than the equivalent SQL queries or programmatic code.**

SQL

```
and   a7.account_id = t7.source_account_id
and   t7.recipient_account_id = a8.account_id
and   a8.account_id = t8.source_account_id
and   t8.recipient_account_id = a9.account_id
and   a9.account_id = t9.source_account_id
and   t9.recipient_account_id = a10.account_id
and   a10.account_id = t10.source_account_id
```

Cypher

```
--Find Fraud Ring 3-10 txns

MATCH ring=(a:Account)
(()-[:PERFORMS]→()- [:BENEFITS_TO]-()){3,10}(a)
RETURN ring;
```

The Neo4j Query Planner will help you tune your Cypher queries and identify the execution plan with the lowest execution cost.

Streamline Development with AuraDB's Unified Tools

Developers have a comprehensive set of tools to build mission-critical applications faster on Neo4j AuraDB. Get started with the language-specific drivers and tools.

Aura console, featuring GenAI Copilot, is a unified platform for developers and administrators integrates Neo4j Bloom, Neo4j Browser, Neo4j Importer, and Neo4j OpsManager. This single interface simplifies data administration, management, ingestion, modeling, and visualization, enhancing usability and creating a streamlined user experience. GenAI Copilot supports natural language interaction, assisting in writing and refining Cypher queries and visually exploring graph data.

Aura Console Offers:

- Improved Workflows: A single hub for all data management tasks
- Consistent User Experience: Unified UX across Neo4j tools
- Easier Collaboration: Share resources and collaborate on projects
- Secure Data Access: Expanded roles and new access controls
- Enhanced Productivity: GenAI Copilot helps write and improve Cypher queries

Uncover patterns in data with Cypher and advanced graph algorithms, visualize with Bloom, run queries 1000x faster, and leverage ACID compliance.

The Department for Education (DfE) is a department is responsible for child protection, education, apprenticeships, and wider skills in England. Luke Slowen, Enterprise Architect at DfE, shared his reasons for selecting Neo4j AuraDB.

"The Department for Education delivers critical services across the education sector, for which we depend on a complex IT infrastructure composed of multiple business systems, services, applications, and components," said Slowen. "Our Neo4j graph creates a shared view of our service landscape, which we use to find ways to reuse capability, improve usability, and rationalize, delivering better-value services for our users. Neo4j AuraDB helps us visualize our data and the opportunities in one place, so we can focus our efforts on streamlining our service delivery and operations."

Department for Education

Uncover Patterns in Data Effortlessly

Find Patterns in Data to Gain Deeper Insights

With Neo4j, you can match patterns in data and relationships to uncover complex or recursive structures efficiently. Using the expressive Cypher query language and advanced graph algorithms eliminates the need for joins, making it easier to analyze intricate connections and gain deeper insights into your data.

Run Diverse Workloads (Analytical and Transactional) on the Same Database

Parallel runtime dramatically reduces the response times of analytical queries, which are often "graph-global," meaning they need to traverse large parts of the graph to uncover complex or hidden patterns. Neo4j has dipped into academics to find an innovative solution based on morsel-based parallelism. Parallel runtime splits queries into multiple software threads, which run concurrently across multiple CPU cores — increasing query speeds by up to 100x.

Visualize and Explore Graph Data

Neo4j Bloom is a powerful visualization tool designed to help organizations explore and interact with graph data intuitively. Its user-friendly interface allows both technical and non-technical users to visualize complex data relationships, patterns, and insights. Bloom users can explore graph data with natural language queries, so they don't need a technical understanding of Cypher or graph databases.

Neo4j now offers enterprise-level support for NeoDash, a low-code dashboard builder that empowers users to easily create interactive visualizations of their graph data. Seamlessly integrated with Neo4j, NeoDash works natively with Cypher queries to build maps, charts, tables, and more, making data insights accessible to everyone. With NeoDash, users can create visualizations using an intuitive drag-and-drop interface, build dashboards directly from Cypher queries, customize and add interactivity to their visualizations, and securely publish dashboards for read-only sharing with stakeholders.

Faster Traversals With Index-Free Adjacency

With Neo4j's index-free adjacency, each node directly references its adjacent (neighboring) nodes, meaning that accessing relationships and related data is simply a pointer lookup. This means native graph processing time increases in proportion to the amount of data processed, not exponentially with the number of relationships traversed and hops navigated.

Without index-free adjacency, a large graph dataset will be crushed under its own weight because queries take longer and longer as the dataset grows. On the flip side, native graph queries perform at a constant rate based on the amount of data they touch, no matter the total volume.

Deploy on a fully managed, fault-tolerant cloud infrastructure on a multi-node cluster with a 99.95% uptime guarantee while maintaining data integrity and security through RBAC, encryption, and compliance with industry standards.

Dun & Bradstreet provides commercial data, analytics, and insights for businesses spanning various sectors. Matthew Voss, Identity Insights Leader, explained how his team supports business strategy using Neo4j AuraDB. "Neo4j helps us to answer questions that span connected data in real-time, including Ultimate Beneficial Ownership (UBO) information," said Voss. "Neo4j AuraDB aligns well with Dun & Bradstreet's cloud strategy and helps us to deliver the deep knowledge our clients need to take intelligent actions based on timely compliance decisions."

dun & bradstreet

Highly Available, Zero Administration

Scalable Multi-Node Architecture With 99.95% Uptime SLA

Achieve high availability with a multi-node cluster architecture provisioned in different availability zones. Ensure redundancy and fault tolerance by maintaining multiple replicas of the database distributed across these nodes. Benefit from a 99.95% uptime guarantee ensured by the self-healing cluster architecture. Manage datasets of varying sizes while tackling memory-intensive workloads by leveraging AuraDB instances with up to 512GB of RAM.

For read-intensive use cases, customers can now spin up "secondaries," which are analogous to a horizontal scaling of existing AuraDB instances. For example, secondary instances can be used to run certain queries on a sub-graph.

Protect Data With Standard Regulations and Security Features

Protect data through comprehensive access controls, leveraging role-based access control (RBAC) to manage read/write permissions across different parts of the graph. Integrate seamlessly with single sign-on (SSO) identity providers like Microsoft Azure Active Directory (AAD) and Okta to streamline access management. Ensure data security at rest and in transit by employing cloud providers' encryption mechanisms to prevent unauthorized access and data theft. Utilize Customer Managed Keys (CMK) to restrict access from service providers and isolate communications to databases from the public internet via cloud providers' private links, mitigating the risk of data hacks. Comply with stringent data governance and industry regulations such as ISO 27001, ISO 20243, GDPR, CCPA, SOC2 Type II, and HIPAA, ensuring adherence to best practices and safeguarding health data and patient confidentiality with HIPAA compliance. Refer to trust.neo4j.com for more information.

Facilitate Event-Driven Architecture With Change Data Capture

Leverage Neo4j change data capture (CDC) to facilitate event-driven architecture. This feature enables triggering actions or workflows in response to changes in the database, enhancing real-time data processing and integration capabilities.

Caterpillar Inc. is a leading global manufacturer of construction and mining equipment, engines, and industrial turbines. Known for its durable and innovative products, the company plays a key role in building infrastructure and advancing industries worldwide

"We wanted to create a system that would allow someone to ask any type of question as long as it was in the domain," said Ryan Chandler, Chief Data Scientist at Caterpillar. "This meant creating a dialog system to test the use of a graph, demonstrate an open-ended user interface capable of answering questions, and develop a capability to create a spoken human-machine interface."

CATERPILLAR

Seamless Data Integration With Tools Ecosystem

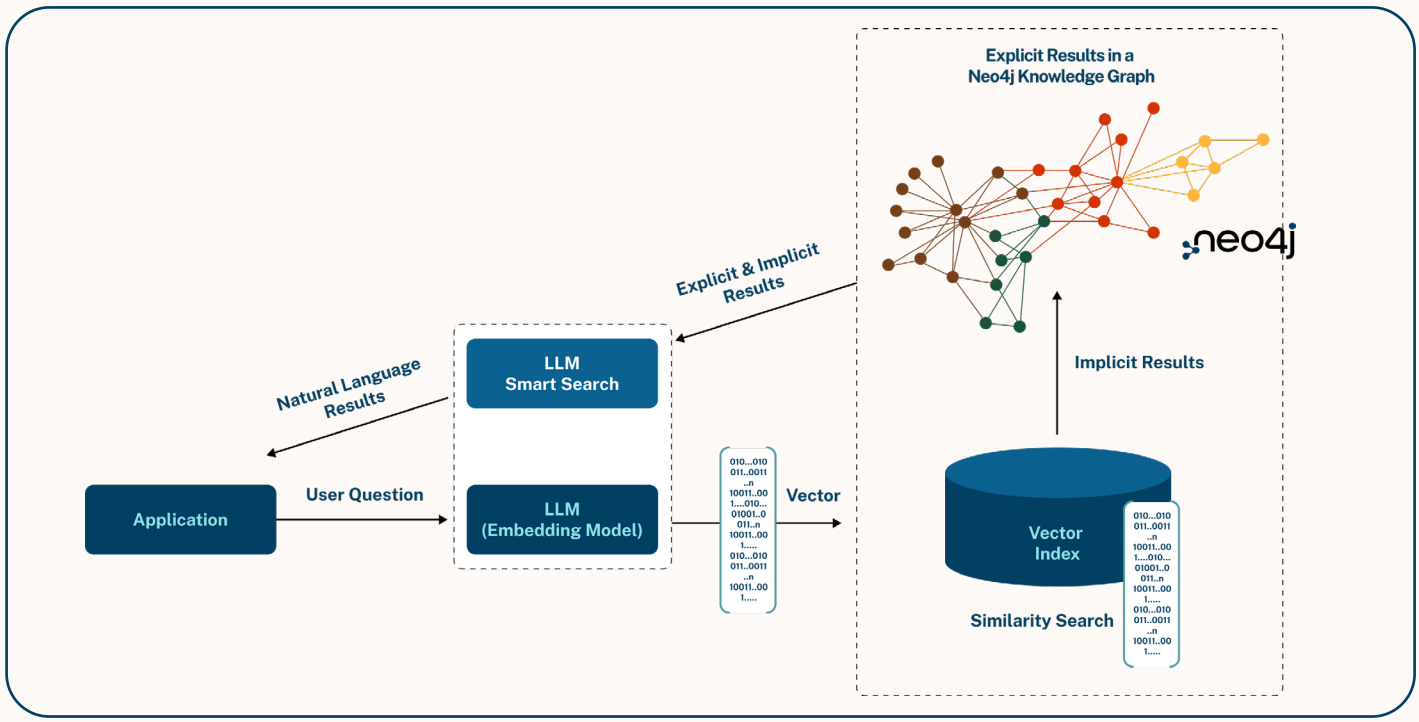
Neo4j integrates seamlessly with a wide range of technologies and platforms, including programming languages like Python, Java, and JavaScript, as well as frameworks like Apache Spark and Apache Kafka. This enables users to leverage existing tools and infrastructure while incorporating graph data into their workflows.

Neo4j AuraDB's tools ecosystem enhances the developer experience, offering a fully managed database with zero admin work, so you can focus on building impactful apps. With extensive connectors, drivers, and integrations, it's easy to integrate into data pipelines for real-time and offline use cases. Neo4j is the top choice for adding graph features to your ML and feature store pipeline. Plus, developers can start right away with a free database, perfect for building, innovating, and scaling in the Cloud.

Include Neo4j AuraDB in Your GenAI Applications for Better Accuracy and Explainability

To realize their immense potential, GenAI applications need to be able to retrieve accurate, richly contextual information from large volumes of text. The traditional approach to this, vector similarity search, often fails to capture critical context, limiting the usefulness of GenAI responses.

Neo4j knowledge graphs — graph databases enriched with a semantic layer — excel at providing this context and improving LLM accuracy, explainability, and transparency. Through a technique called GraphRAG, LLMs retrieve relevant information from knowledge graphs using vector search and then augment their responses with the domain-specific contextual data in the knowledge graph:



It's hard to overstate the importance of knowledge graphs in GenAI development. Gartner considers knowledge graphs essential to the development of GenAI, and has urged data leaders to "leverage the power of LLMs with the robustness of knowledge graphs to build fault-tolerant AI applications."

For more information on knowledge graphs and other GenAI resources, refer to:
neo4j.com/generativeai/#resources

How to Get Started With AuraDB

To get started with AuraDB, sign up on the AuraDB website and create a database by selecting a name, region, and configuration options. After creation, use the provided connection details (URI, username, password) to connect via Neo4j Browser, Neo4j Desktop, or supported drivers. Load your data using Cypher commands, CSV files, or import tools, and explore it with Cypher queries and Neo4j Bloom for visualization.

For additional information or guidance, see the AuraDB documentation page or Neo4j's free online training resources.