Graphs are Everywhere

Yogesh Haribhau Kulkarni

YHK

Outline

YHK

About Me

Yogesh Haribhau Kulkarni

Bio:

- 20+ years in CAD/Engineering software development
- Got Bachelors, Masters and Doctoral degrees in Mechanical Engineering (specialization: Geometric Modeling Algorithms).
- Currently doing Coaching in fields such as Data Science, Artificial Intelligence Machine-Deep Learning (ML/DL) and Natural Language Processing (NLP).
- Feel free to follow me at:
  - Github (github.com/yogeshhk)
  - LinkedIn (www.linkedin.com/in/yogeshkulkarni/)
  - Medium (yogeshharibhaukulkarni.medium.com)
  - Send email to yogeshkulkarni at yahoo dot com

Office Hours: Saturdays, 2 to 5pm (IST); Free-Open to all; email for appointment.

YHK

Introduction

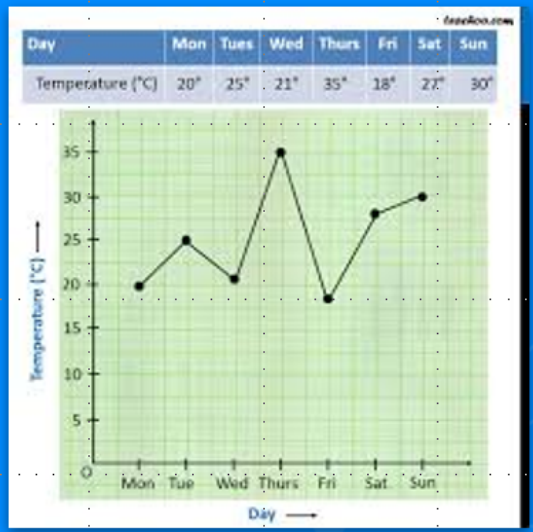
Welcome

- Graphs are a powerful tool for modeling and analyzing complex relationships.
- In this presentation, we will explore the ubiquitous nature of graphs and how Neo4j's graph database can unlock their potential.
- Let's dive into the world of graphs!

YHK

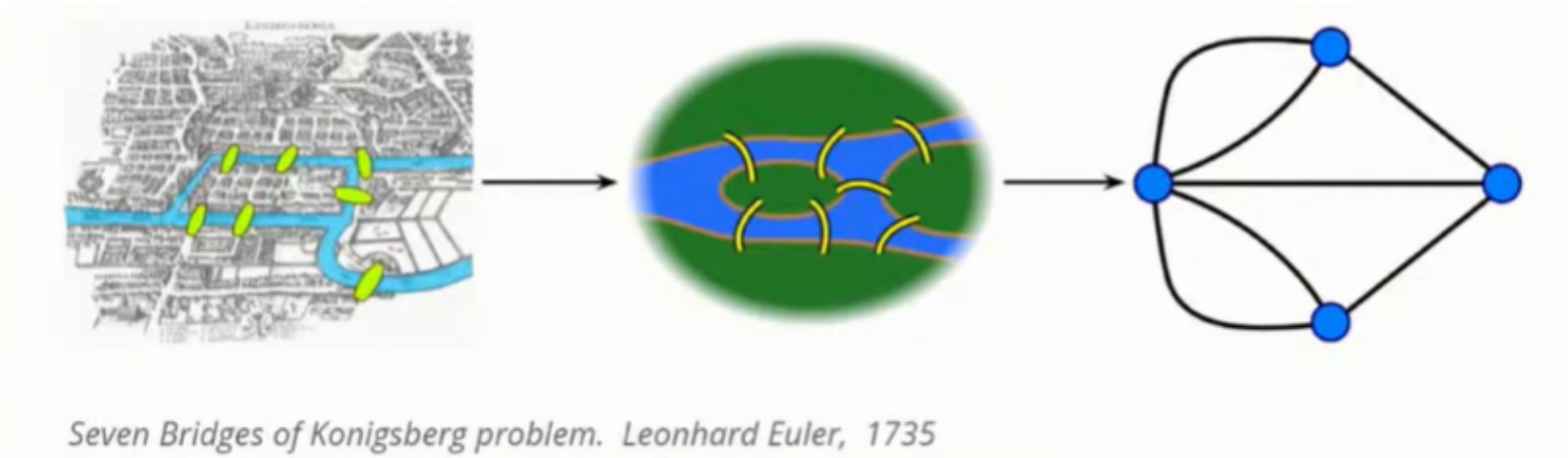
Graphs? Which one?

versus



A graph is

. . . a set of discrete objects, each of which has some set of relationships with the other objects
Euler: Can we take a walk to all 4 islands, without crossing any of the bridge twice?
Abstraction (Does size of islands matter?):

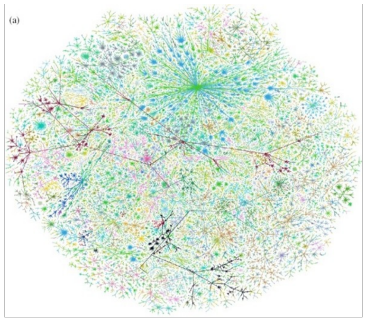


Solution: No way!! What's the rule? [Ans: Homework!!]

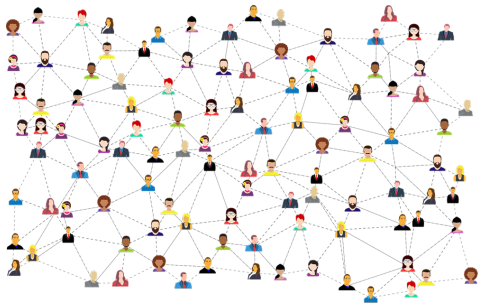
(Ref: Introduction to Neo4j - a hands-on crash course - neo4j)

YHK

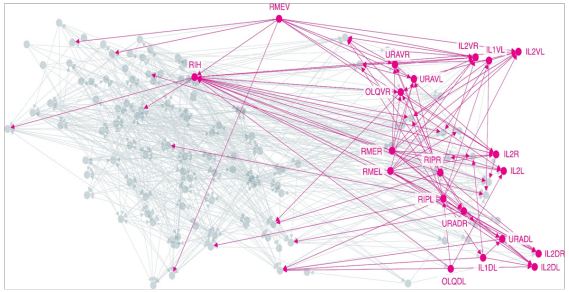
Graph-structured Data Are Ubiquitous



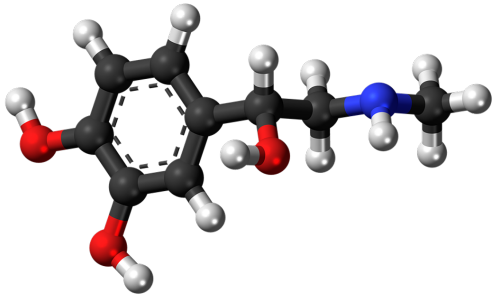
Internet



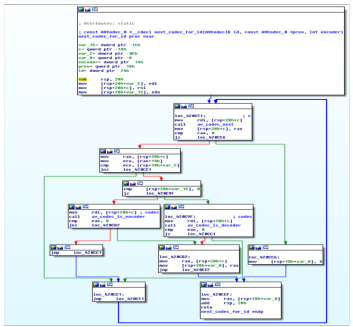
Social networks



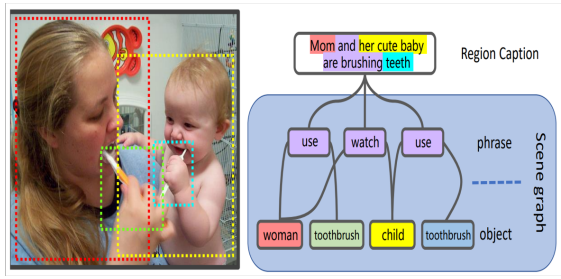
Networks of neurons



Biomedical graphs



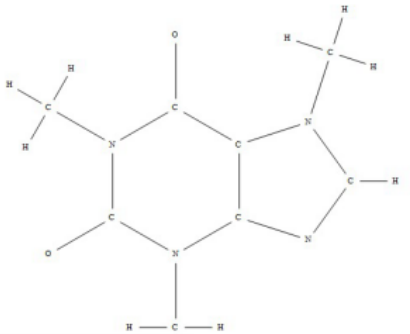
Program graphs



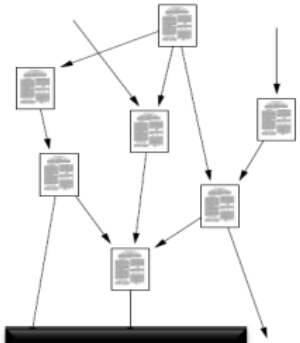
Scene graphs

YHK

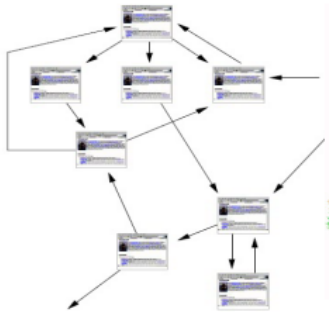
Networks around us!



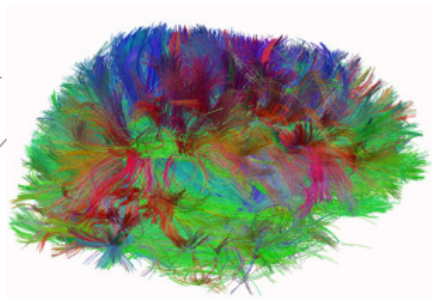
Molecules



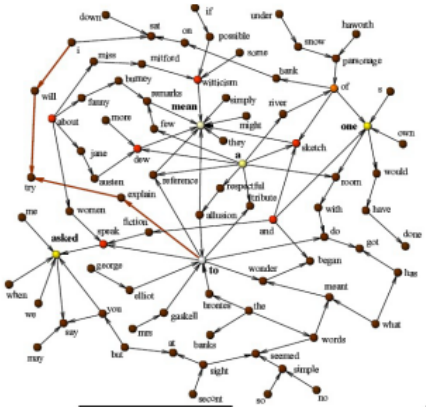
Knowledge



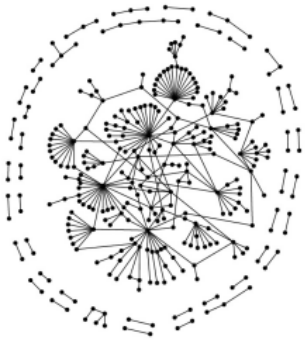
Information



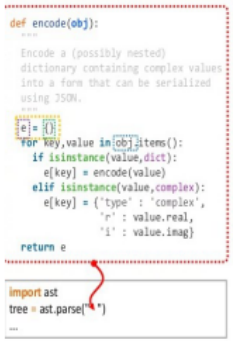
Brain/neurons

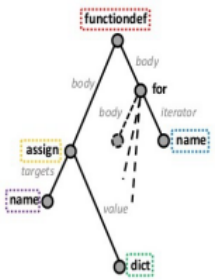


Genes



Communication

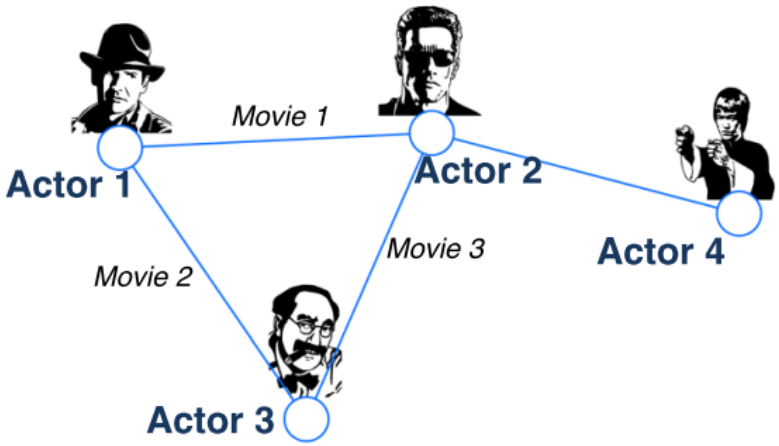


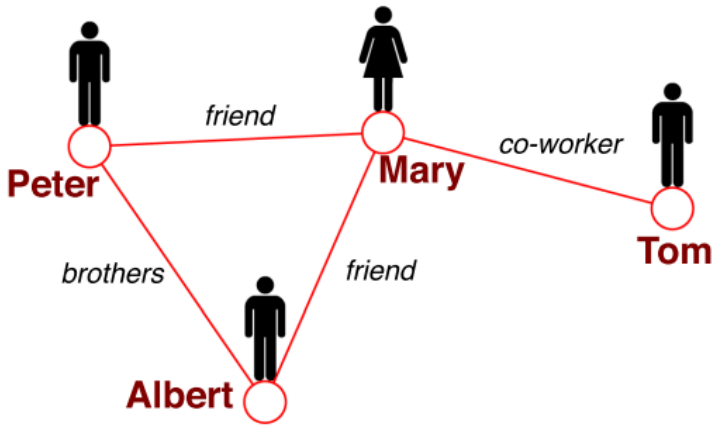


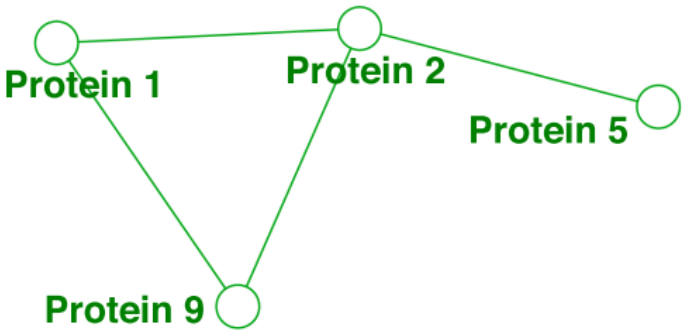
Software

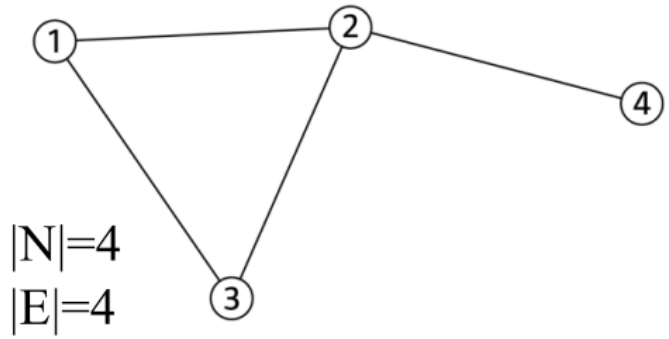
Social

Graphs: Common Language



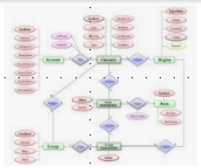




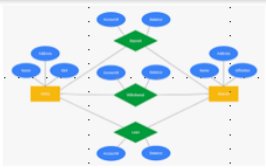


YHK

Graphs are everywhere !


Entity–relationship model - Wikipedia
en.wikipedia.org

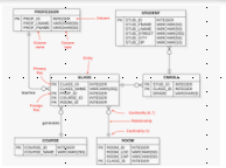

Entity Relationship Diagram (ERD) | ...
creately.com


Entity Relationship Diagram (ERD ...
smartdraw.com


ER Diagram (ERD) - Definition ...
lucidchart.com


Entity Relationship (ER) Diagram M...
guru99.com

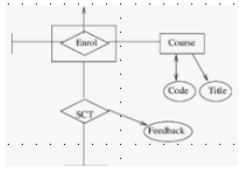

Entity Relationship Diagram (ERD ...
medium.com


Entity-Relatibnship Diagram - StarUML ...
docs.staruml.io

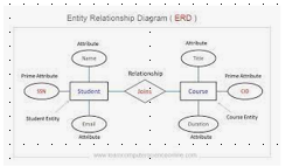

What is Entity Relationship Diagram (ERD)?
visual-paradigm.com


Create ER Diagram for SQL Server ...
blog.devart.com

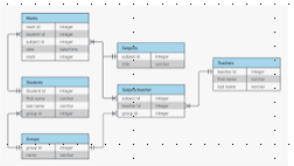

Entity Relationship Diagram (ERD ...
smartdraw.com

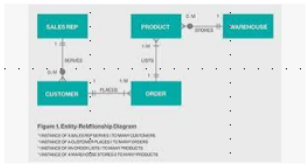


Entity-Relationship Diagram for Student ...
researchgate.net



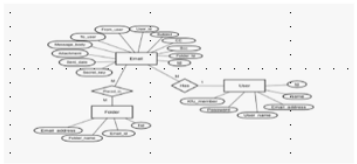
Entity Relationship Diagram (ERD ...
learncomputerscienceonline.com

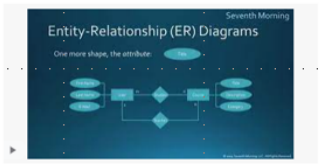

ER Diagram (ERD) - Definition ...
lucidchart.com



What is Entity Relationship Diagram ...
techtarget.com

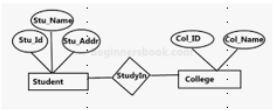

Entity Relationship Diagram (ERD ...
trevor.io

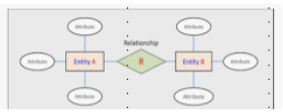

Entity Relationship Diagram |Download ...
researchgate.net



Entity-Relationship Diagrams - YouTube .
youtube.com


Entity Relationship Diagram Tutorial ...
edrawsoft.com





neo4j

YHK

Why do graphs matter?

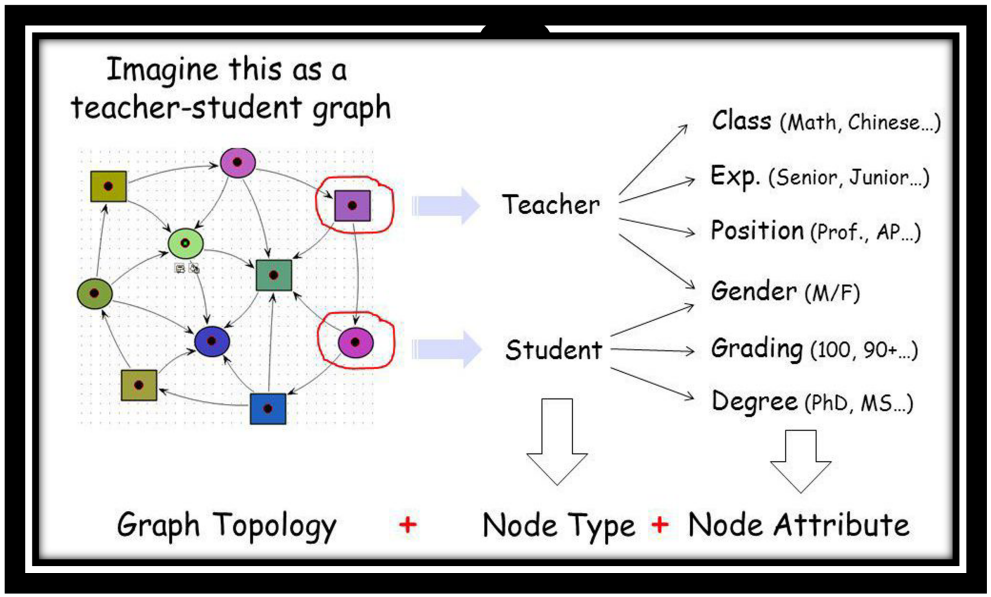
Across an organization, every department can benefit from graphs to answer questions like who or what is important, what should I do next, and what's unusual about this?

(Ref: 5 Graph Data Science Basics Everyone Should Know - neo4j)

YHK

Graphs: A Universal Language

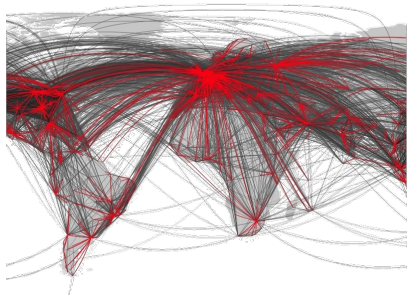
Graphs are a general language for describing and modeling complex systems



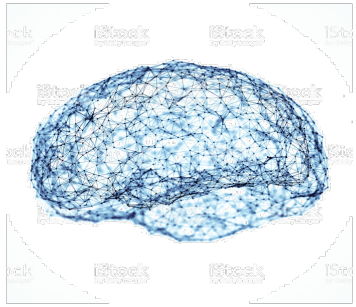
Graph!

Data as Graphs - Explicit

Social Graphs

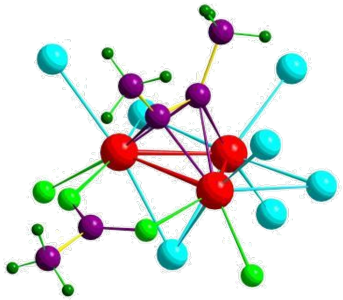


Transportation Graphs

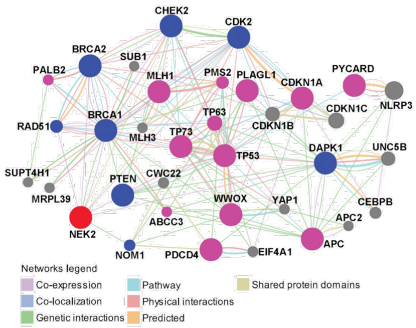


Brain Graphs

Web Graphs



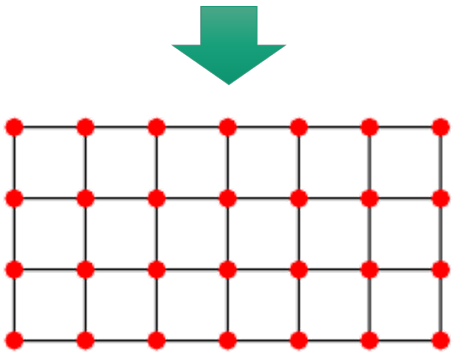
Molecular Graphs

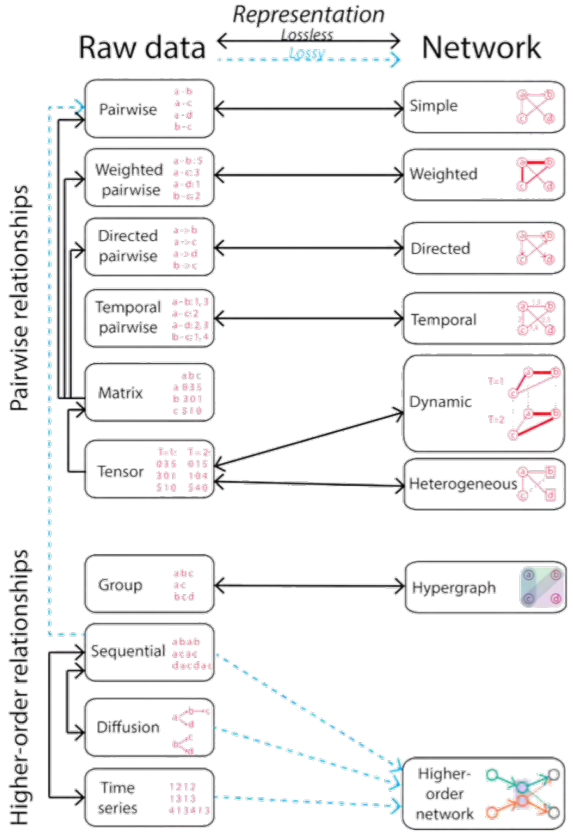


Gene Graphs

YHK

Data as Graphs - Implicit





Jian Xu. Representing Big Data as Networks. PhD Dissertation, University of Notre Dame

YHK

Graph4NLP

Homogeneous vs Multi-relational vs Heterogeneous Graphs

Graph types	Homogeneous	Multi-relational	Heterogeneous
# of node types	1	1	> 1
# of edge types	1	> 1	>= 1

64

YHK

Graph Theory Basics

Graph Definition

A graph G is an ordered pair $G = (V, E)$, where V is the set of vertices and E is the set of edges connecting the vertices.

- Directed vs. undirected graphs
- Nodes (vertices) and relationships (edges)
- Properties and labels

YHK

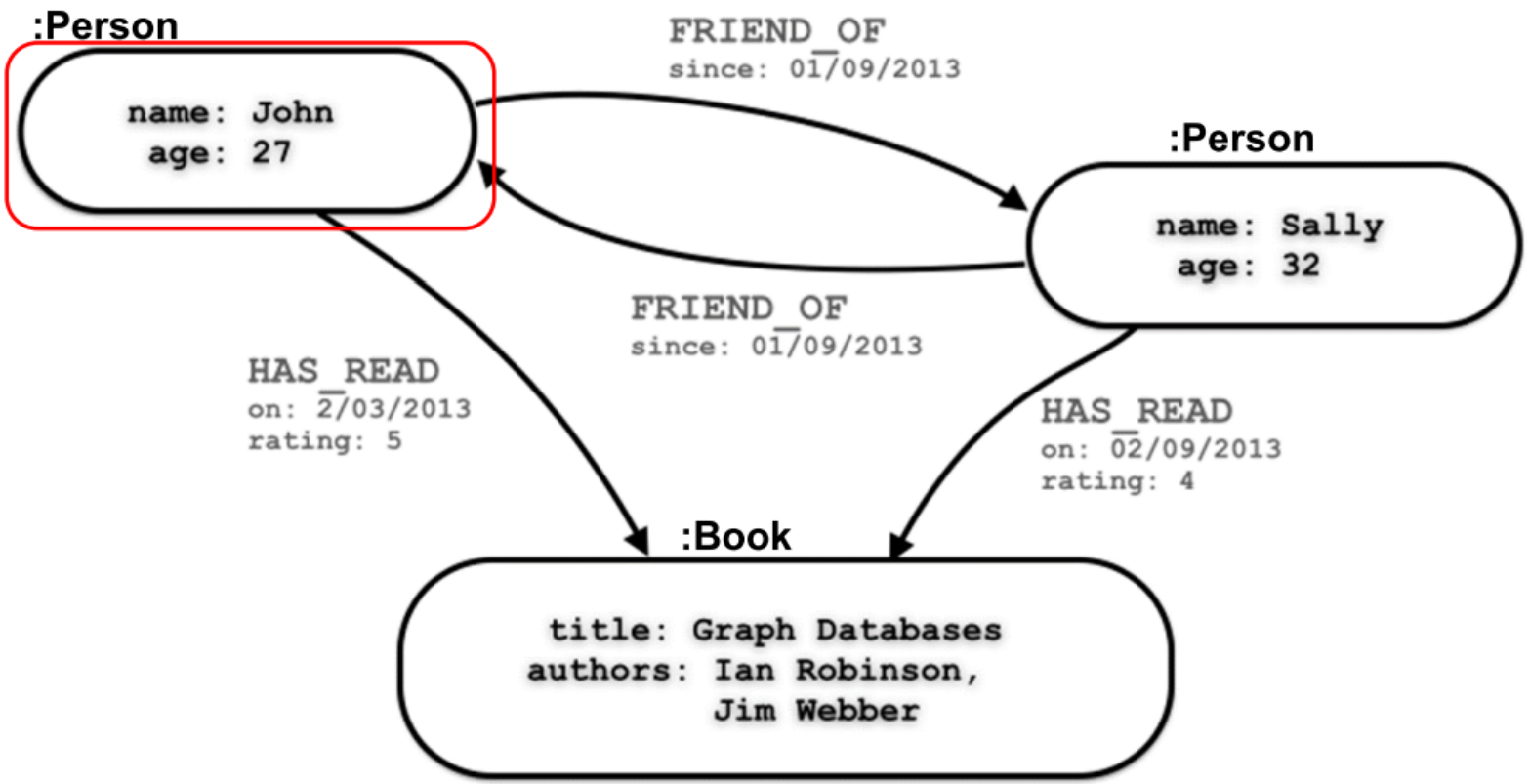
Graph Components

- Node (Vertex): A must data element for constructing a graph
- Relationship (Edge) : Link between two nodes, can have direction and type.
- Label: Node category/type such as PERSON, ORG, etc. One node can have many types.
- Properties: Attributes or fields in Nodes or Edges, eg. A node can have Label PERSON and Property such as “name: Jane”

(Ref: Introduction to Neo4j - a hands-on crash course - neo4j)

YHK

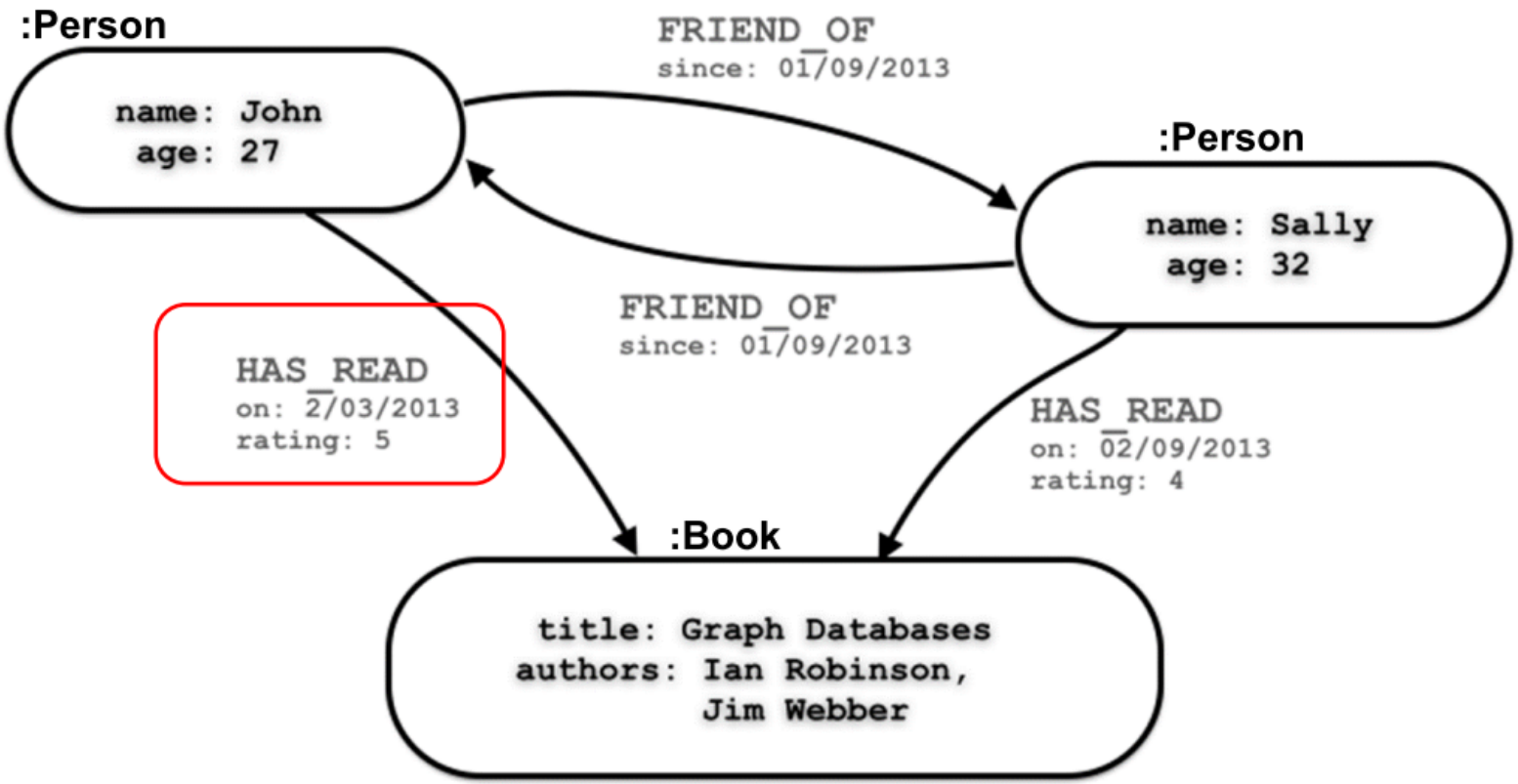
Nodes



(Ref: CIS 6930 - Advanced Databases - Neo4j)

YHK

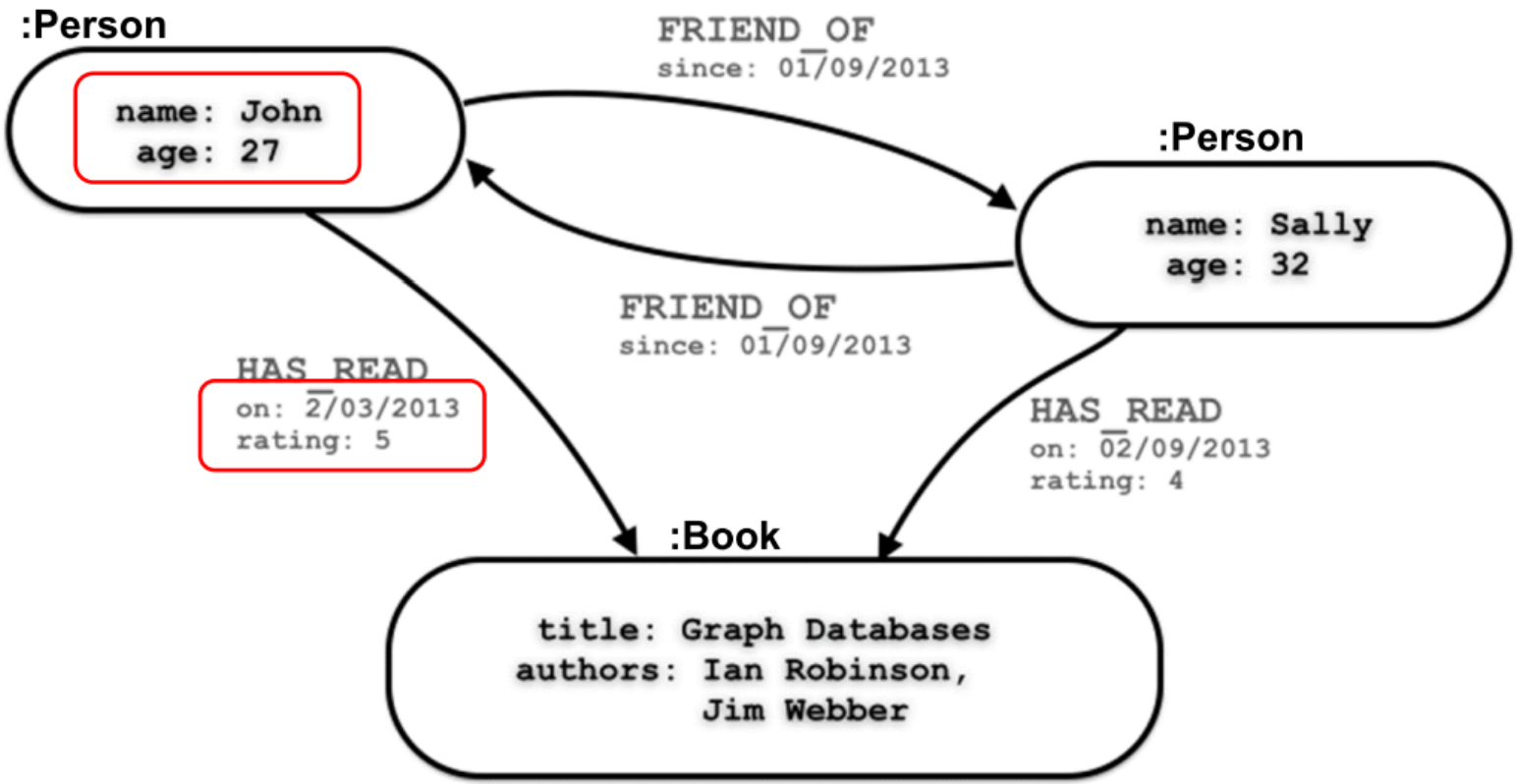
Relationships



(Ref: CIS 6930 - Advanced Databases - Neo4j)

YHK

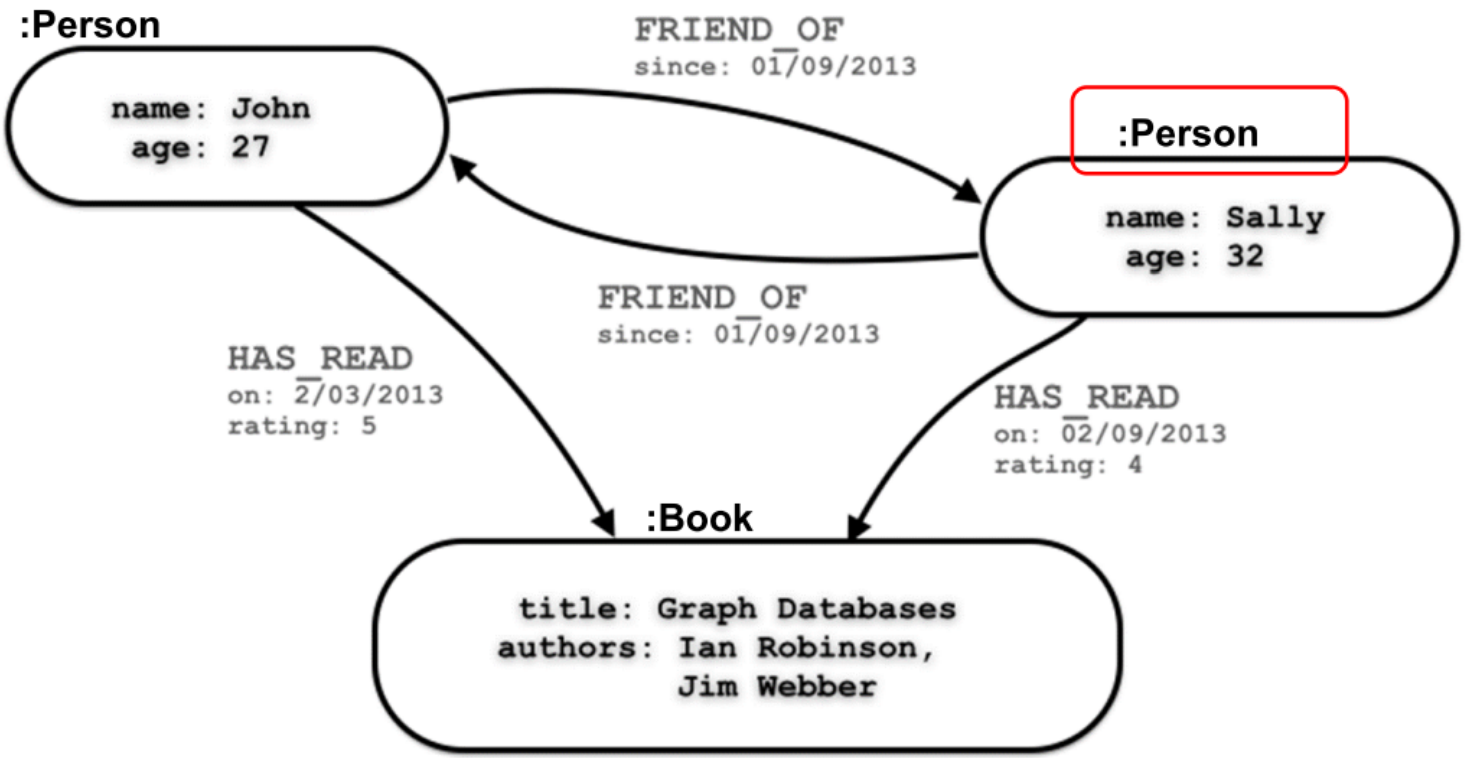
Properties



(Ref: CIS 6930 - Advanced Databases - Neo4j)

YHK

Labels



(Ref: CIS 6930 - Advanced Databases - Neo4j)

YHK

Graph-Edge Types

- Undirected graph: edges are bidirectional, e.g. Michale is brother of John, necessarily means that John is brother of Michale.
- Directed graph: edges have one direction, e.g. A likes B does not necessarily mean that B likes A.
- Weighted graph: edges have weights, e.g. connection from city A to city B via road R1 will have wright, say, 8, due to high traffic, but via road R2, may have weight 2, due to lower traffic.

YHK

Graph Algorithms

- Graph algorithms are a set of computational techniques for analyzing and processing graphs.
- Some popular graph algorithms include:
  - Breadth-First Search (BFS)
  - Depth-First Search (DFS)
  - Shortest Path Algorithms (Dijkstra's, Bellman-Ford)
  - Clustering Algorithms (Louvain, Label Propagation)
  - PageRank, Betweenness Centrality, and more.
- These algorithms are fundamental building blocks for solving a wide range of problems.

YHK

Why Graphs? Why Now?

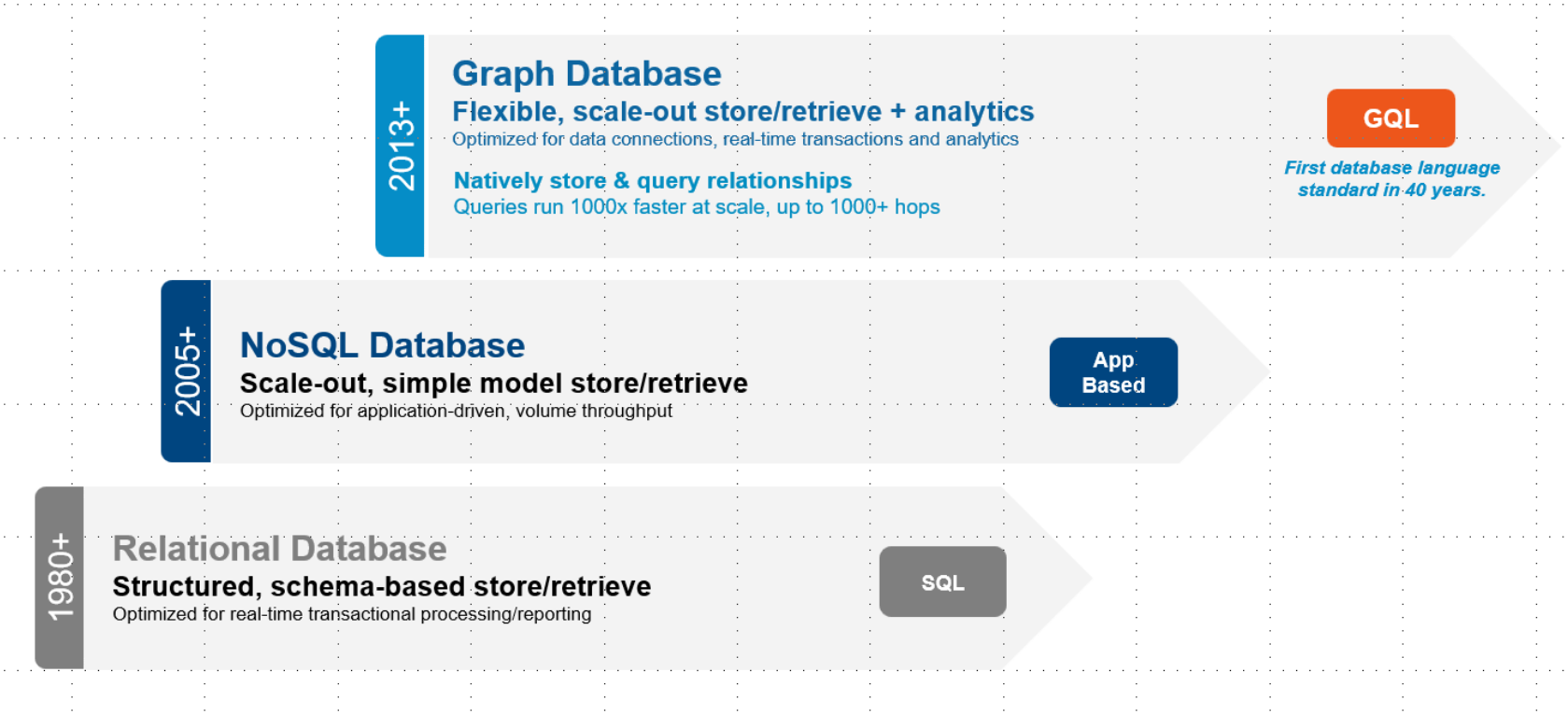
- Universal language for describing complex data: Networks/graphs from science, nature, and technology are more similar than one would expect
- Shared vocabulary between fields: Computer Science, Social science, Physics, Biology, Economics
- Data availability (+ computational challenges): Social/Internet, text, logic, program, bio, health, and medical
- Impact: Social networking, Social media, Drug design, Event detection, Natural language processing, Computer vision, and Logic reasoning

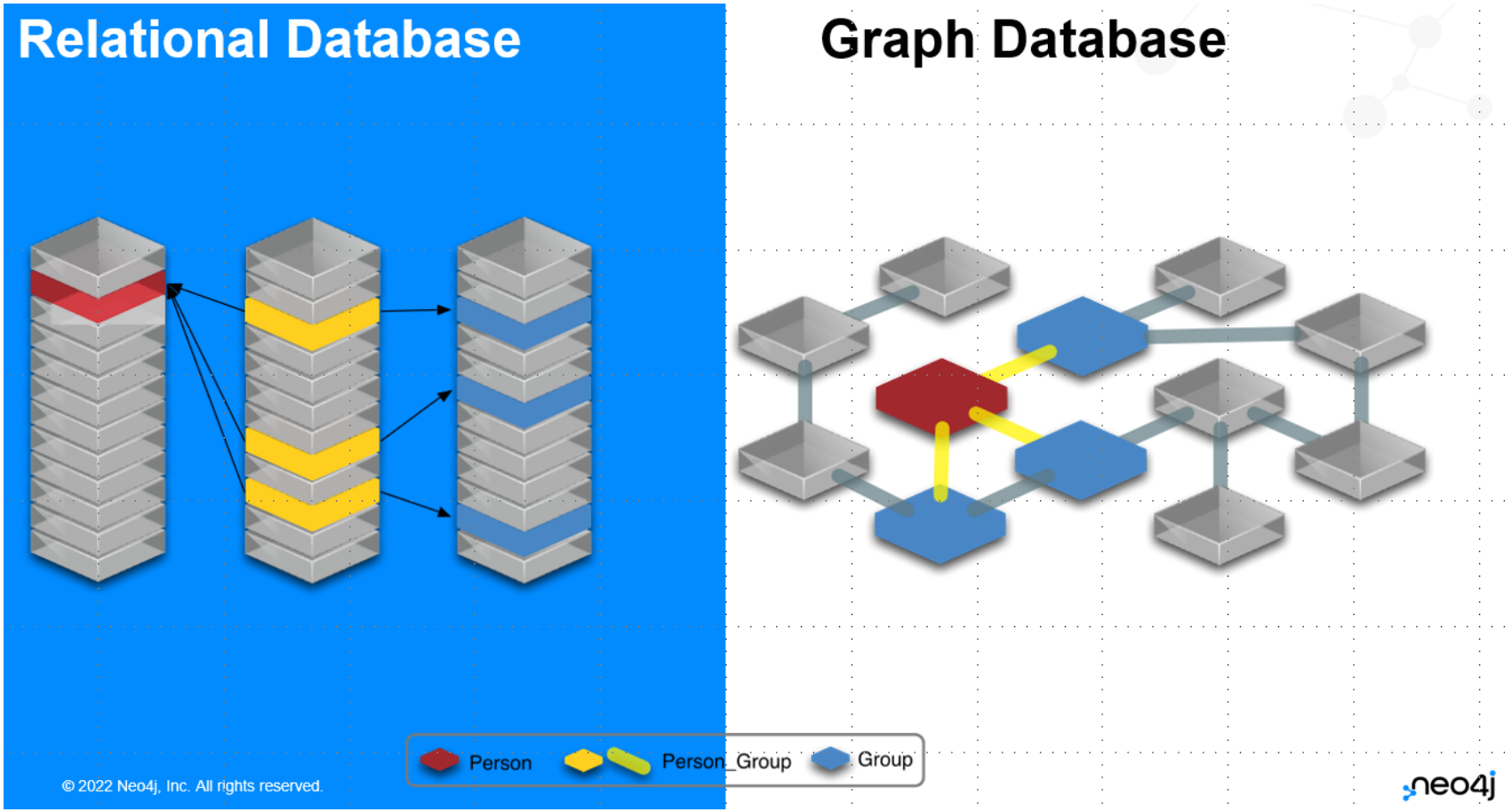
YHK

Graph Database

Why?

Graph databases keep data in its natural, connected state.




Relational Database
Graph Database
Person
Person_Group
Group
© 2022 Neo4j, Inc. All rights reserved.
neo4j

Neo4j: A Graph Database

- Neo4j is a leading graph database that allows you to model, store, and query highly connected data.
- Features:
  - Native graph storage and processing
  - Cypher query language
  - ACID transactions
  - High scalability and performance
- Neo4j provides a flexible and efficient way to work with graph data.

YHK

What is Neo4j?

- Idea conceived on a flight to Mumbai
- Product & Company started in 2007
- Open-Core Model:
  - Free Open-Source Community Edition
  - Closed-Source Enterprise Edition
- Free for local use with Neo4j Desktop / Docker
- Current Version: 5.0 (Released in Nov 2022)

YHK

Graph Database Use Cases

- Fraud detection and prevention
- Social network analysis
- Recommendation engines
- Network and IT operations
- Knowledge graph management
- Impact analysis and risk assessment
- Master data management
- And many more!

YHK

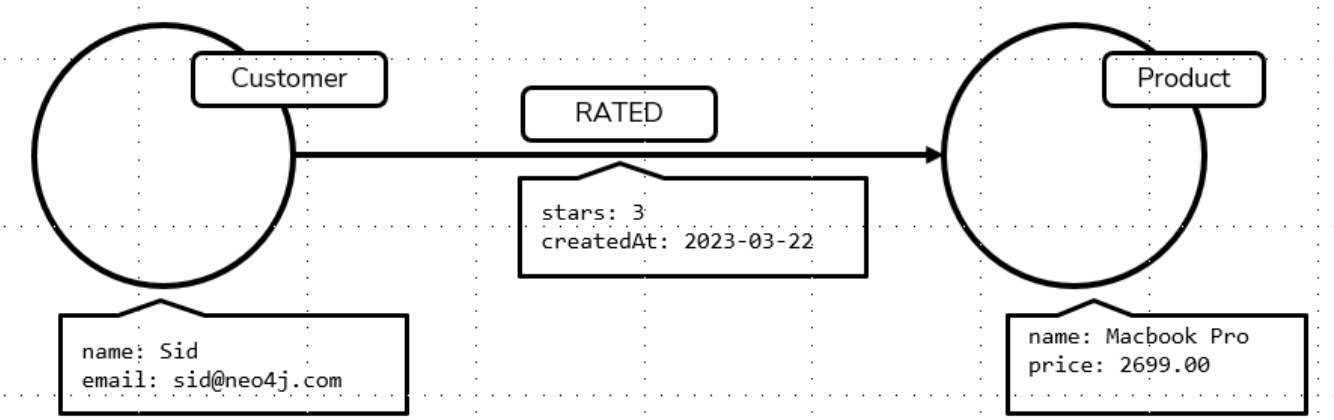
Graph Modeling in Neo4j

- Neo4j follows a property graph model.
- Nodes represent entities, while relationships capture the connections between them.
- Nodes and relationships can have properties to store additional information.
- Labels and relationship types provide meaningful categorization.

YHK

Cypher Query Language

- Cypher is Neo4j's query language designed specifically for working with graph data.
- It is a declarative language that allows you to express complex graph patterns and operations.
- Cypher provides a human-readable syntax for querying and manipulating graph data.
- Examples of Cypher queries: CREATE, MATCH, WHERE, RETURN, etc.




```
MATCH (c:Customer)-[r:RATED]->(p:Product)
RETURN c.name AS customer, p.name AS product, r.stars
AS rating
```

YHK

Nodes and relationships at a glance

Description	Node	Relationship
Generic	`()`	`--` `-->` `-[]-`
With a reference	`(n)`	`-[r]-`
With a node label or rel type	`(:Person)`	`-[:ACTED_IN]-`
With a label/type and an inline property	`(:Person {name: 'Bob'})`	`-[:ACTED_IN {role: 'Dave'}]-`
With a variable, label/type and an inline property	`(p:Person {name: 'Bob'})`	`-[r:ACTED_IN {role: 'Rob'}]-`

neo4j

YHK

Graph Data Import

- To work with graph data in Neo4j, you need to import it into the graph database.
- Neo4j provides various methods for data import, including:
  - CSV import: Bulk import data from CSV files.
  - Integration with ETL (Extract, Transform, Load) tools.
  - Integration with programming languages (e.g., Java, Python) using Neo4j drivers.
- Efficient data import is crucial for graph analytics workflows.

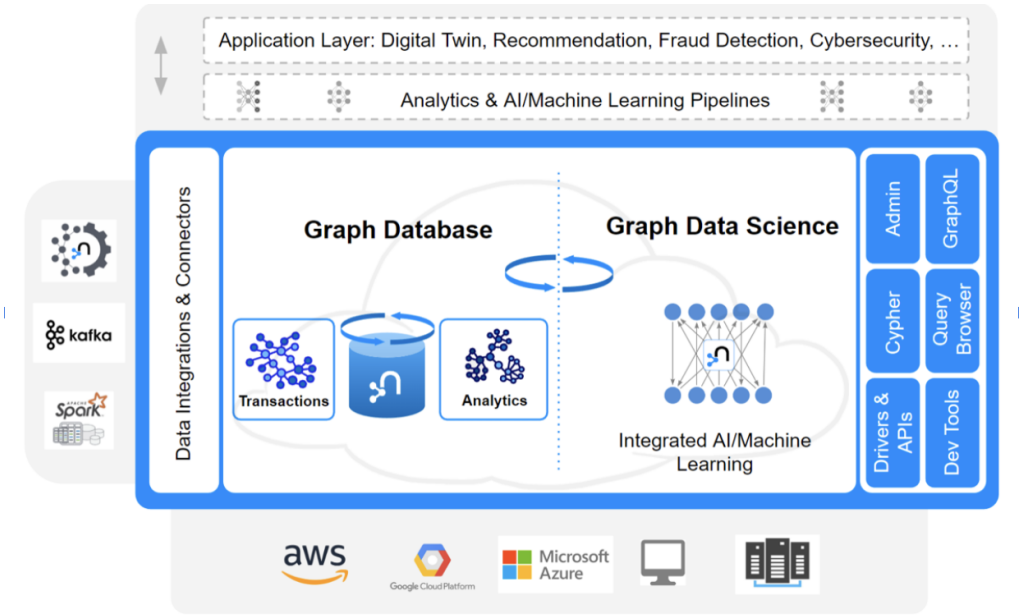
YHK

Graph Analytics Workflow

- Define the problem: Identify the business problem that can be solved using graph analytics.
- Data modeling: Create a graph data model that represents the problem domain.
- Data import: Import data into the graph database.
- Algorithm selection: Choose the appropriate graph algorithm(s) for the problem at hand.
- Algorithm execution: Run the selected algorithm(s) on the graph data.
- Result interpretation: Analyze and interpret the algorithm output to gain insights.

YHK

Neo4j Architecture



(Ref: https://neo4j.com/blog/neo4j-integrates-data-ecosystem-connectors/)

- Neo4j follows a client-server architecture.
- Core components: Graph engine, transaction manager, and query processing.
- Highly scalable and fault-tolerant architecture.

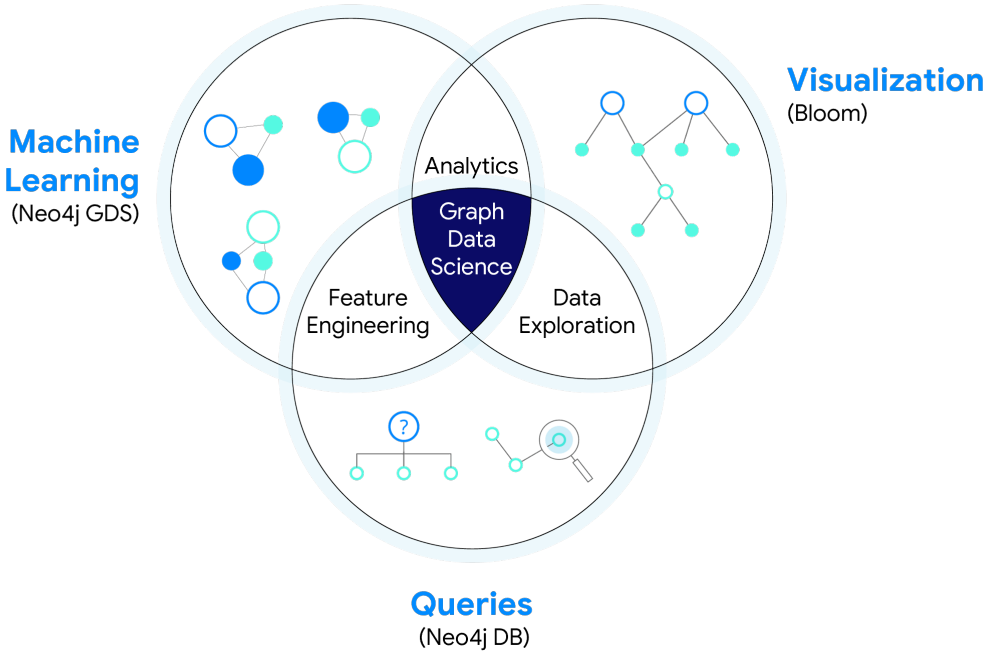
YHK

Scalability and Performance

- Neo4j is designed to handle large-scale graph datasets.
- It provides horizontal scalability through clustering and sharding techniques.
- Queries are optimized for efficient graph traversal and pattern matching.
- Neo4j's query optimizer maximizes query performance.

YHK

Graph Data Science



Queries
Find the patterns you know exist

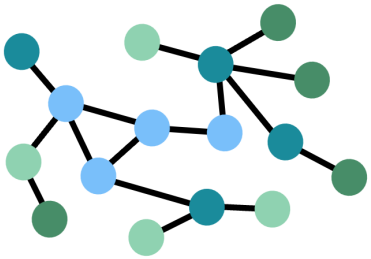
Machine Learning
Uncover trends and make predictions

Visualization
Explore, collaborate, and explain

YHK

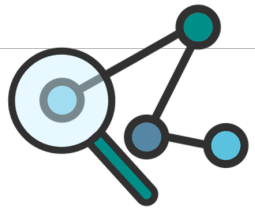
Graph and Data Science

Knowledge Graphs



Find the patterns you're looking for in connected data

Graph Algorithms



Use unsupervised machine learning techniques to identify associations, anomalies, and trends.

Graph Native Machine Learning

Use embeddings to learn the features in your graph that you don't even know are important yet.

Train in-graph supervised ML models to predict links, labels, and missing data.

YHK

Graph Data Science (GDS) Family of Algorithms

Pathfinding & Search

Centrality & Importance

Community Detection

Supervised Machine Learning

Heuristic Link Prediction

Similarity

Graph Embeddings

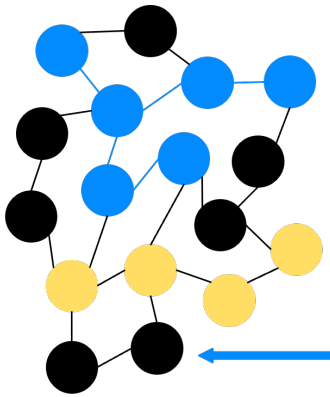
…and more

76

YHK

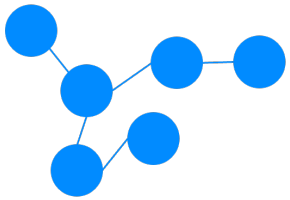
GDS Process in Neo4j

Build Knowledge Graph



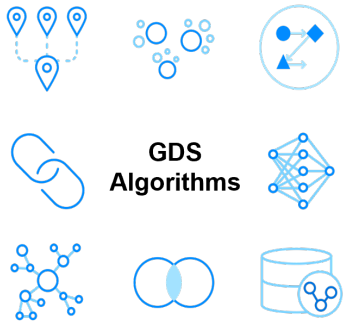
Ingest and model data into the Knowledge Graph. Then, it will be stored into the **disk**.

Graph Projection



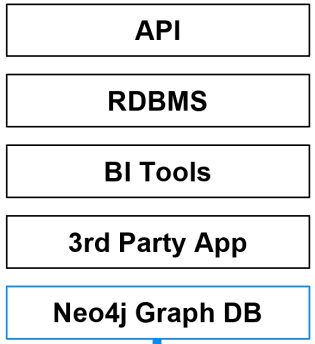
Specific portion of the **graph will be projected in-memory** depending on what the algorithm requires e.g. monopartite, bipartite or multipartite graph.

Execute Algorithm



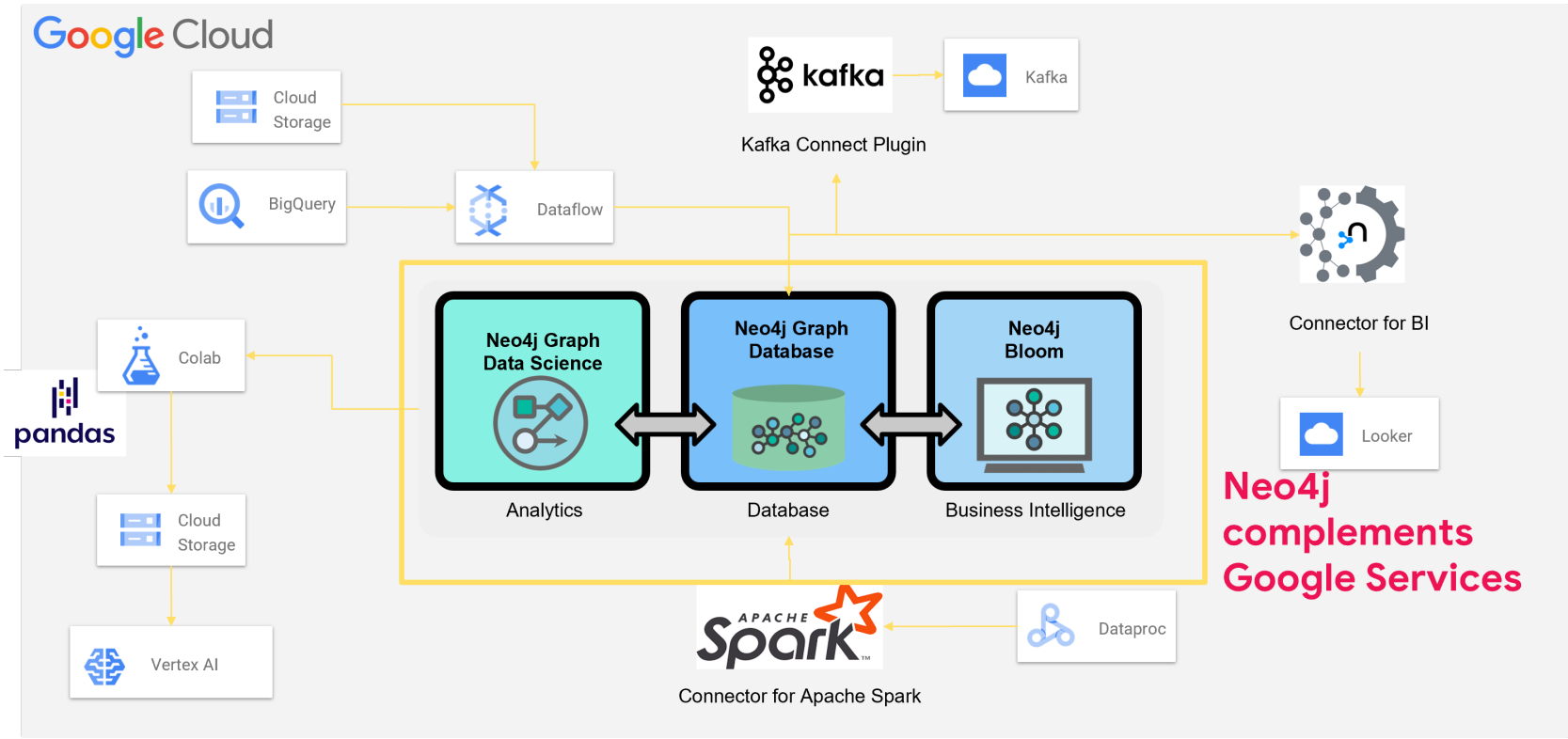
Graph algorithm and machine learning will be executed within Neo4j.

Use the Graph Insights



Outputs of the GDS algorithms and ML is **feed back** into **Neo4j** or any of the **downstream systems**.

Neo4j in the Google Cloud Ecosystem



Use Cases for Graph Databases

- Fraud detection and prevention: Uncover patterns of fraudulent activities in financial transactions.
- Social network analysis: Identify communities, influencers, and relationships within a social graph.
- Recommendation engines: Power personalized recommendations based on user preferences and graph connections.
- Network and IT operations: Analyze and optimize network infrastructure, detect anomalies, and troubleshoot issues.
- Knowledge graph management: Organize and link diverse knowledge sources for semantic search and data integration.
- Impact analysis and risk assessment: Analyze the impact of changes or events on interconnected systems and assess associated risks.
- Master data management: Manage and integrate complex relationships within master data domains.
- Recommendation engines: Power personalized recommendations based on user preferences and graph connections.
- Network and IT operations: Analyze and optimize network infrastructure, detect anomalies, and troubleshoot issues.

YHK

Real-World Applications

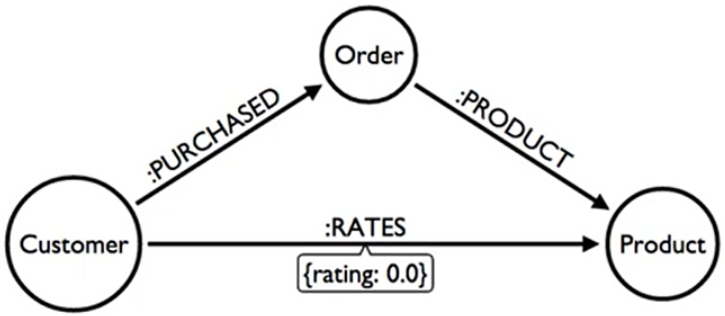
- Healthcare: Disease networks, drug discovery
- E-commerce: Personalized recommendations, fraud detection
- Transportation: Route optimization, logistics planning
- Social media: Influencer analysis, community detection
- Finance: Anti-money laundering, fraud detection
- IoT: Sensor networks, anomaly detection

YHK

Common Use cases: E-Commerce Recommendations

Easy in graph databases: those who bought A also bought B.

```cypher
MATCH (c:Category)-[:HAS_CHILD|HAS_PRODUCT*1..3]→(p:Product)
RETURN p.id, p.title, collect(c.name) AS categories
```

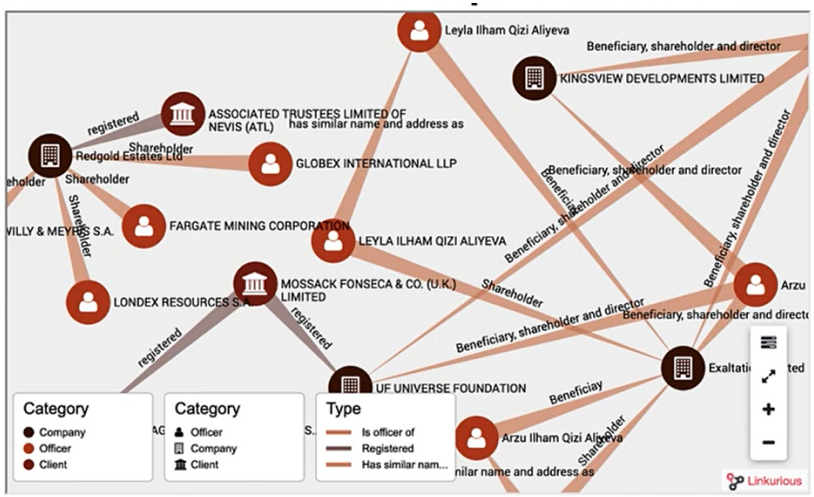


(Ref: Introduction to Neo4j - a hands-on crash course - neo4j)

YHK

Common Use cases: Investigative Journalism

Panama papers: Identify corruption based on relationships between people/companies/financial-institutions.

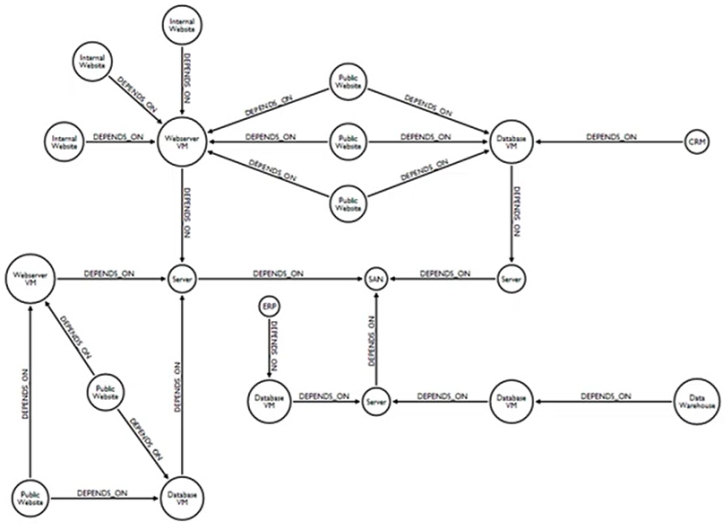


- What families with the name that contains the string 'aliye' are Officers of Companies?
- How is the family with the name that contains the string 'aliye' related to Companies?
- How are Officers related to each other?
- What are the connections between multiple companies and a family?

(Ref: Introduction to Neo4j - a hands-on crash course - neo4j)

YHK

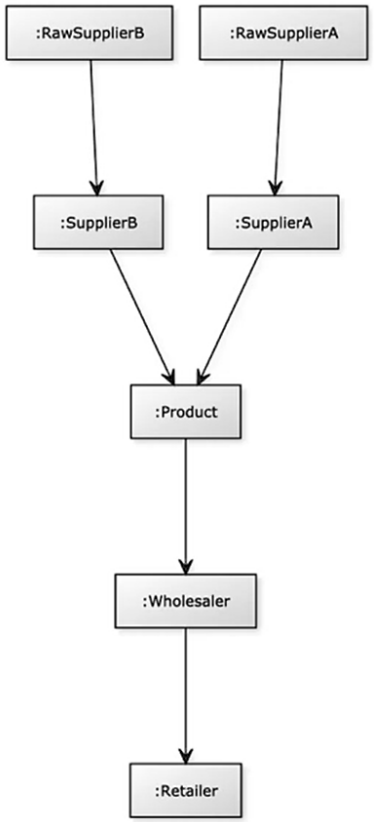
Common Use cases: Network Dependencies



- What are the direct dependencies of public websites?
- What are the direct dependencies of internal websites?
- What is the most depended-upon component?
- Find the dependency chain for a business critical component.
- What is the impact of removing a server?

(Ref: Introduction to Neo4j - a hands-on crash course - neo4j)

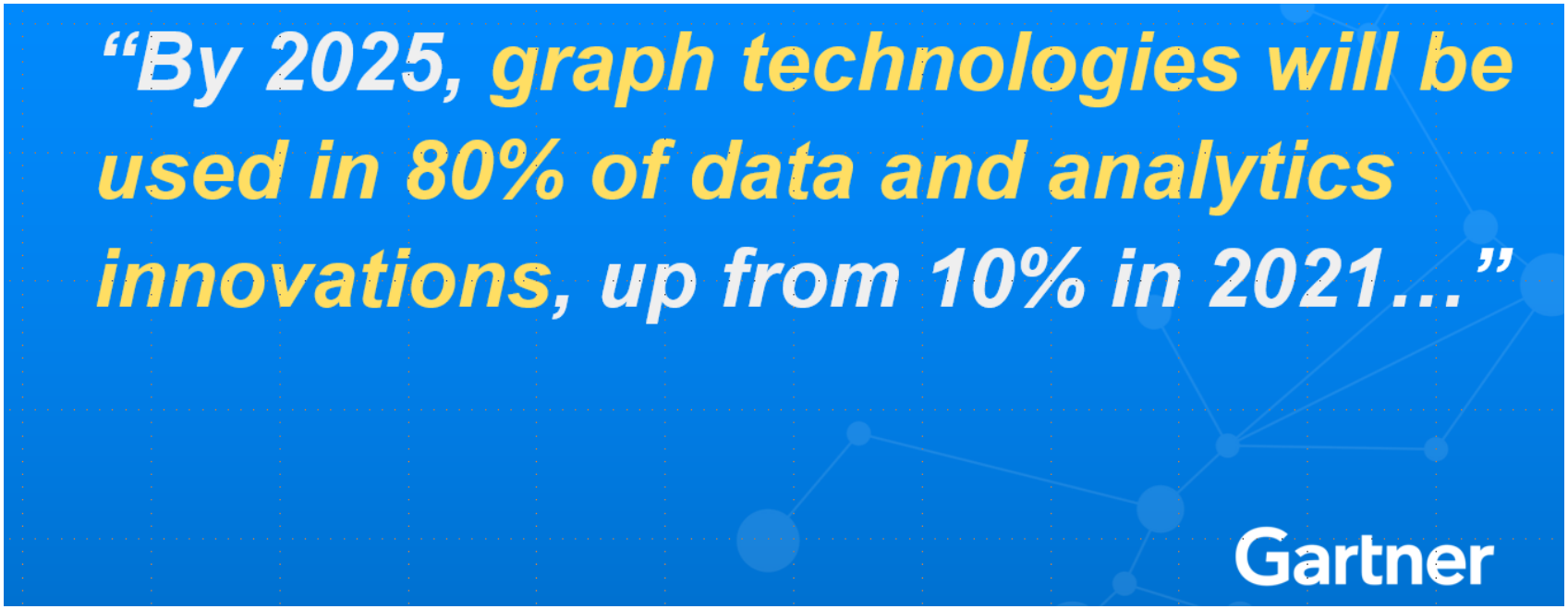
Common Use cases: Supply Chain



- Who is the best wholesaler for each retailer based upon distance?
- Which raw supplier will give a particular retailer the freshest products?
- Which retailer provides locally grown products?
- How can we rate each supply chain?

(Ref: Introduction to Neo4j - a hands-on crash course - neo4j)

YHK


"By 2025, graph technologies will be used in 80% of data and analytics innovations, up from 10% in 2021…"
Gartner

Continue your graph journey with Graph Academy

What is Graph Academy?

Free, Self-Paced, Hands-on Online Training to help you learn how to build, optimize and launch your Neo4j project, all from the Neo4j experts.

What's more?

2 free certifications designed to test you on your overall knowledge of Neo4j:

- Neo4j Graph Data Science Certification
- Neo4j Certified Professional

Interested? For more information visit:
www.graphacademy.neo4j.com

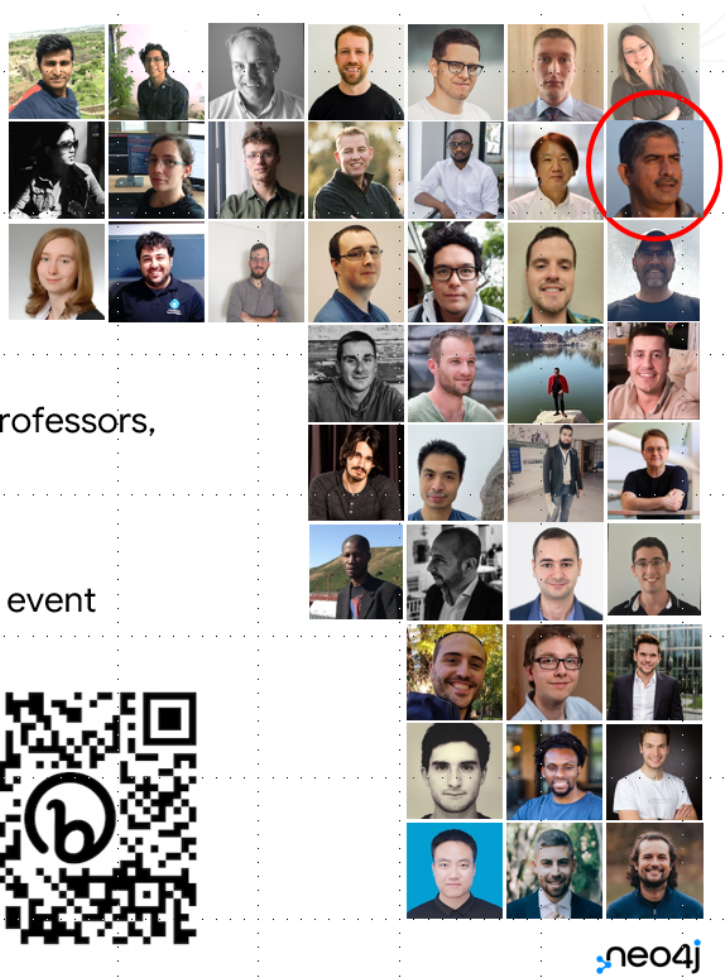
neo4j

YHK

Community

Meet the Neo4j Ninjas

Masters of Graphs



Ninjas are:

Active graph bloggers, presenters, GitHub contributors, professors, user group leaders, and researchers

Benefits:

Ninjas benefit from exclusive access to Neo4j experts, VIP event experience, special giveaways and much more

Interested? For more information visit:
https://bit.ly/Neo4jNinjas

neo4j

YHK

Key Takeaways

- Graphs are a natural way to represent and analyze complex relationships in various domains.
- Neo4j provides a powerful and efficient graph database solution for working with highly connected data.
- Graph algorithms enable powerful insights and solutions to a wide range of problems.

YHK

References

- Neo4j website: `https://neo4j.com`
- "Graph Algorithms" book by Mark Needham and Amy E. Hodler
- "Graph Databases" book by Ian Robinson, Jim Webber, and Emil Eifrem
- Neo4j documentation and developer guides
- "an introduction to neo4j (graph database tutorial for beginners)" - Chris Hay
- CIS 6930 - Advanced Databases - Neo4j
- "Learning Neo4j" - Wabri/LearningNeo4j - Github
- Neo4J Certification Sample Questions - https://wiki.glitchdata.com/index.php/Neo4J_Certification
- Neo4j Certified Professional: Exam Practice Tests - By Cristian Scutaru
- "Graph Data Science with Neo4j Graph Algorithms - Will Lyon" Youtube
- "Mark Needham - Intro to Graph Data Science with Neo4j" Youtube
- "THE POWER OF "GRAPH DATA SCIENCE" using Neo4j" Youtube
- "Using Neo4j Graph Data Science in Python to Improve Machine Learning Models" - Tomaz Bratanic
- "Graph Data Science for Supply Chains – Part 1" - Zach Blumenfeld

YHK

Thanks ...

- Search **"Yogesh Haribhau Kulkarni"** on Google and follow me on LinkedIn and Medium
- Office Hours: Saturdays, 2 to 5pm (IST); Free-Open to all; email for appointment.
- Email: yogeshkulkarni at yahoo dot com

(Generated by Hugging Face QR-code-AI-art-generator, with prompt as "Follow me")

YHK