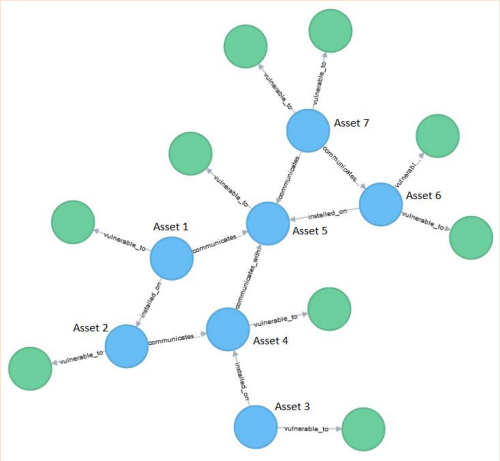
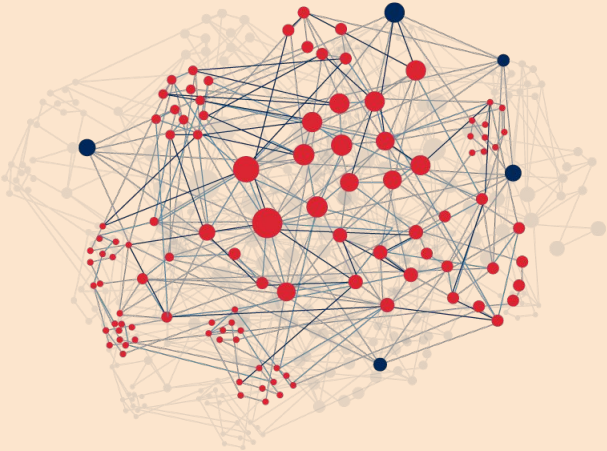
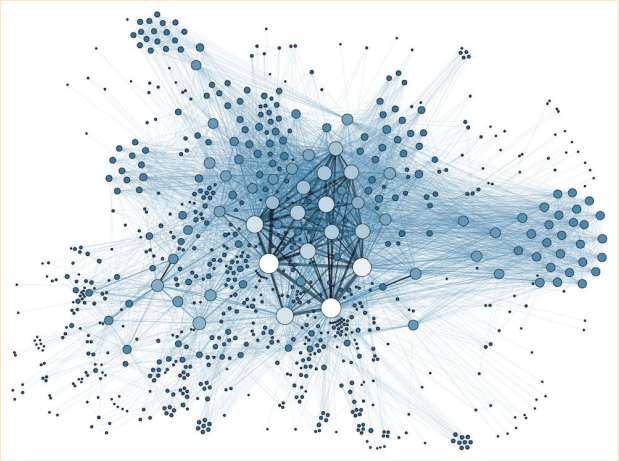
GRAPH DATABASES

"Graph Databases are rapidly gaining traction in the market as an effective method for deciphering meaning … allowing you to manage your data more efficiently. … Your unstructured content is now discoverable allowing all types of users to quickly find the exact information for which they are searching."

- Forbes: "The Hype Around Graph Databases And Why It Matters"





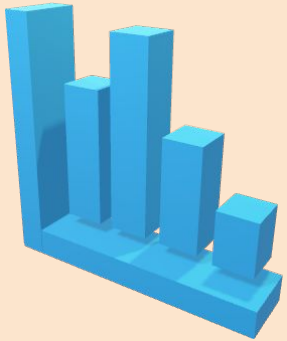


THE MISSING LINK

THE DATA

THE BUSINESS

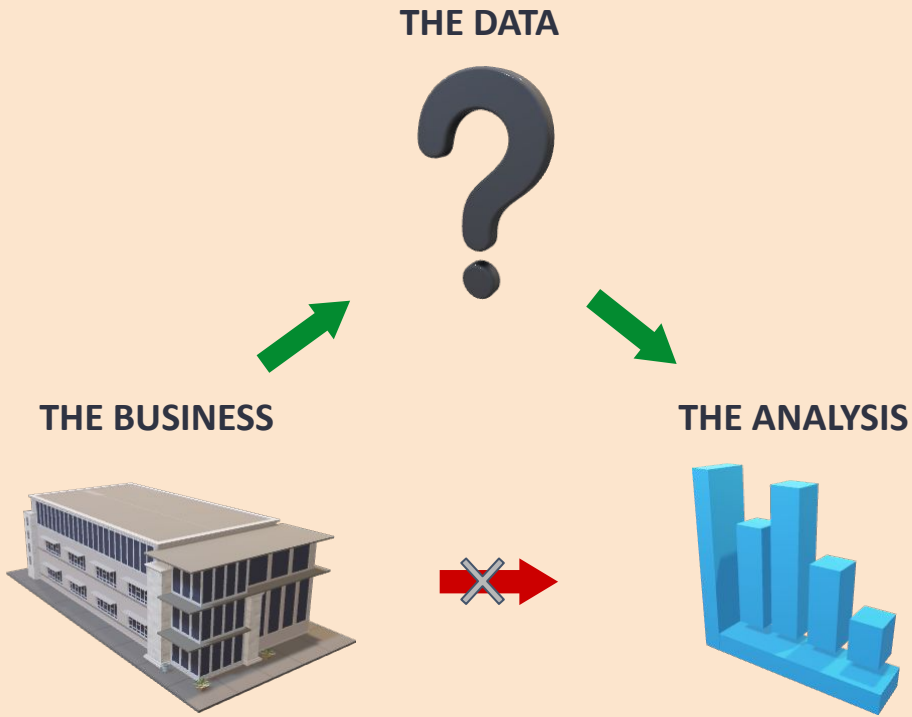
THE ANALYSIS

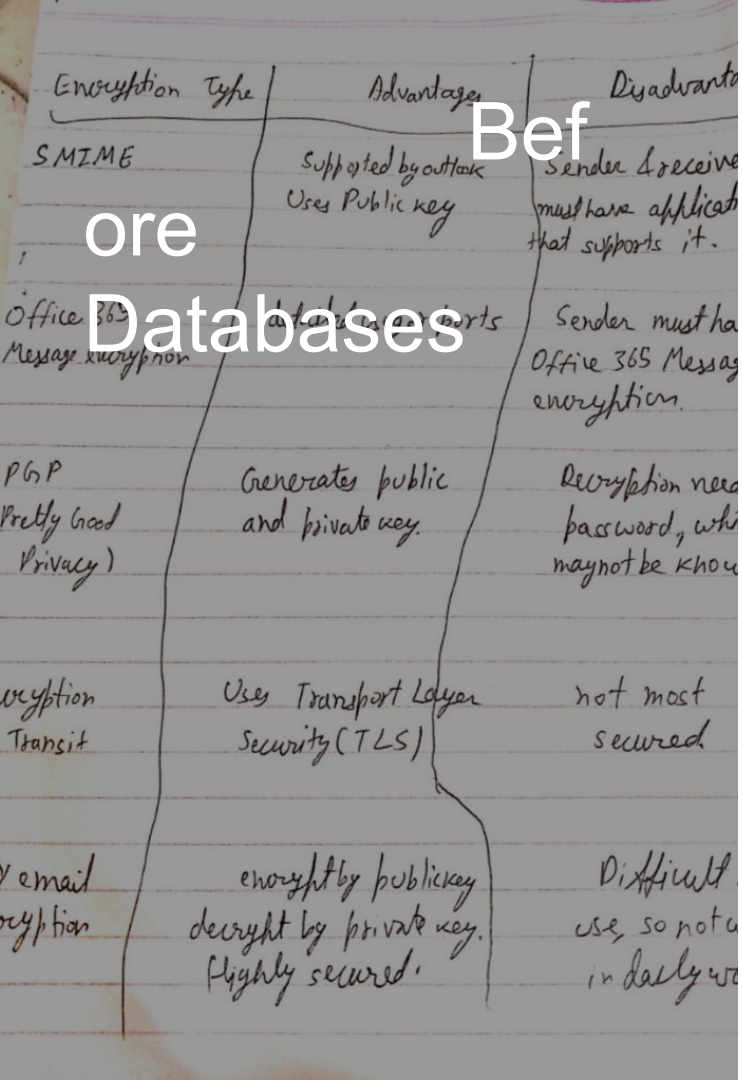


THE IMPORTANCE OF QUALITY DATA

What data do we have?

- Do we have quality structured data?
- What is the relationship between the different assets?
- Do the connected assets interact with each other properly?
- How strong is the relationship between the different assets?
- Do we have missing data and anomalies?





Bef

ore

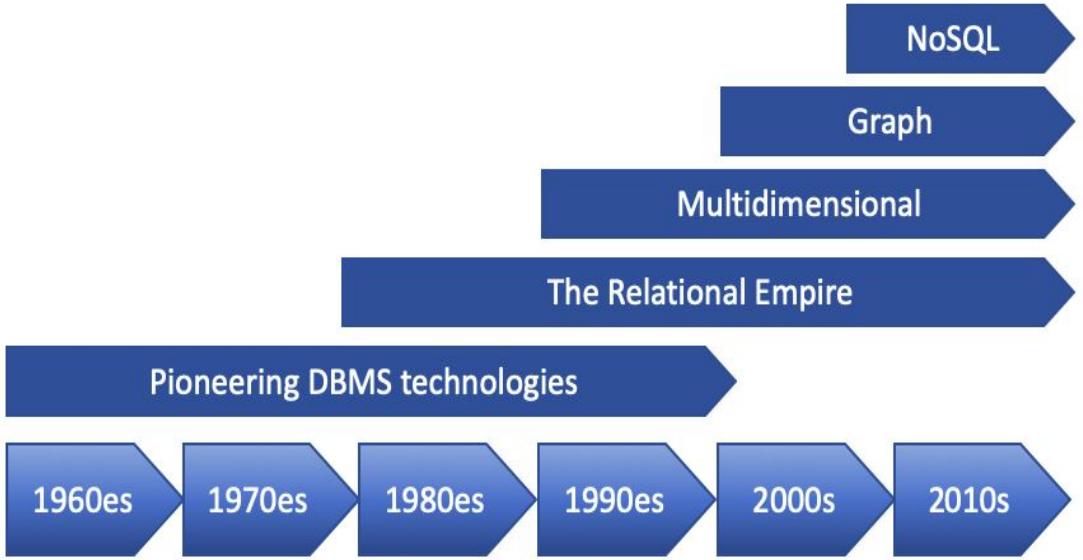
Databases

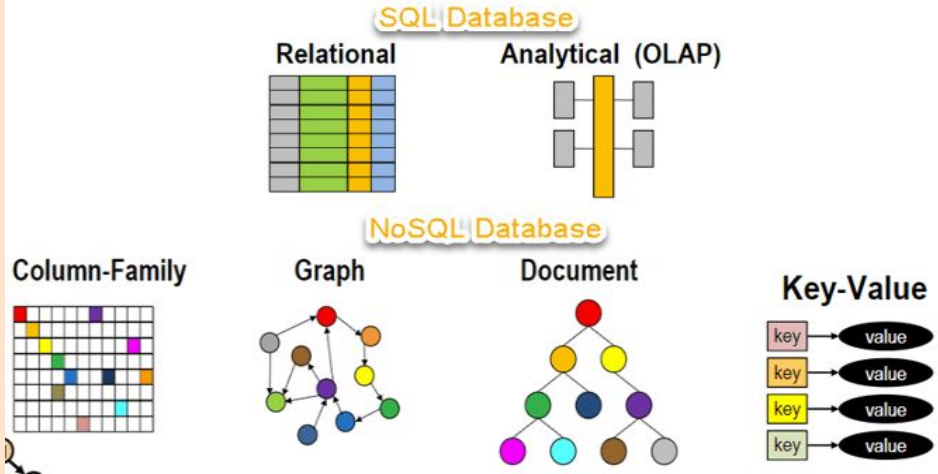
Limitations of paper-based storage

- Lists, ledgers with thousands or millions of
  records in filing cabinets
- Accessing and physically obtaining such records
  was slow and laborious task
- Misplaced records
- Fires that wiped entire archives
- Destroyed history of societies, organizations
  and governments

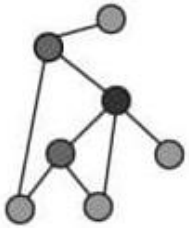
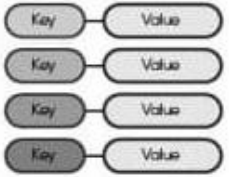
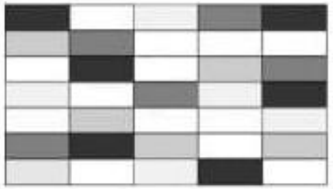
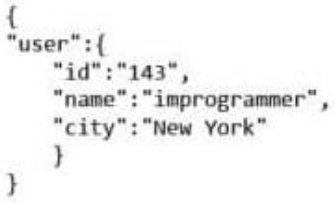
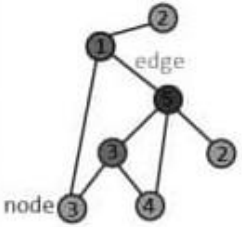
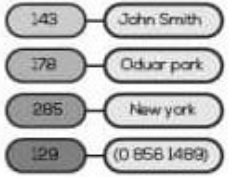
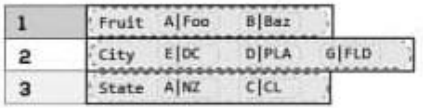
Evolution of DBs

- Hierarchical and Network DBMS in the 1960's
- Relational empire period: began in the 1970's and started to loose users in 2008 with the rise of NoSQL and Big Data
- Graph DBs period started in 1999 with Semantic Web and 2008 with Property Graph DBs

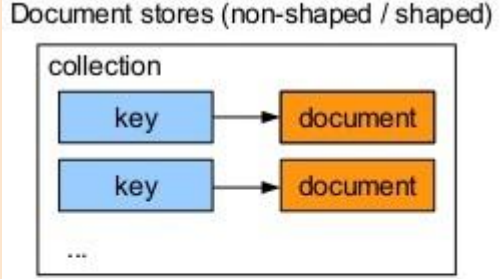


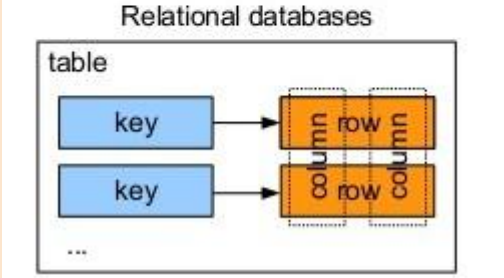


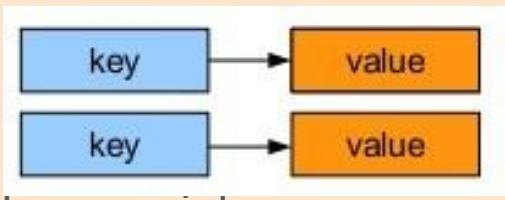
NoSQL DBs

Document	Graph	Key-Value	Wide-Column
			
			
			

NoSQL DBs







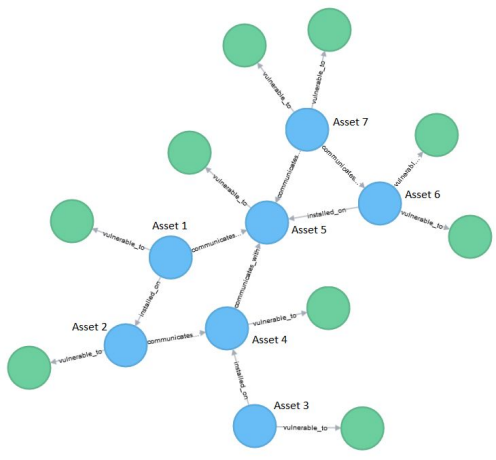
- **Document** DBs (Flexibility): usually pair each key value to a complex object called document; Documents are stored in **collections** which are similar to tables in RDBM and have the structure of BTrees
- **Key-value DBs** (Simplicity) :
  Every item is stored as a key-value pair, such as a dictionary and a map. Key-value pair stores are the most simple NoSQL DBs
- **Graph DBs:** store information about networks, graphs such as social connections, road maps, transport links
- **Columnar** (Column-oriented DBs): optimized for queries over large datasets and instead of rows they store columns of data

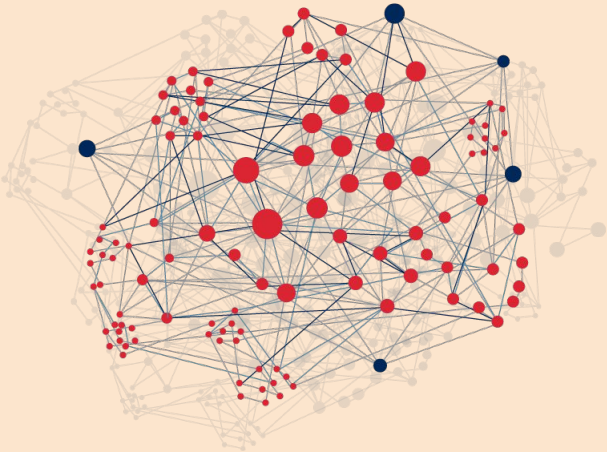
Feature	NoSQL Databases	Relational Databases
Performance	High	Low
Reliability	Poor	Good
Availability	Good	Good
Consistency	Poor	Good
Data Storage	Optimized for huge data	Medium sized to large
Scalability	High	High (but more expensive)

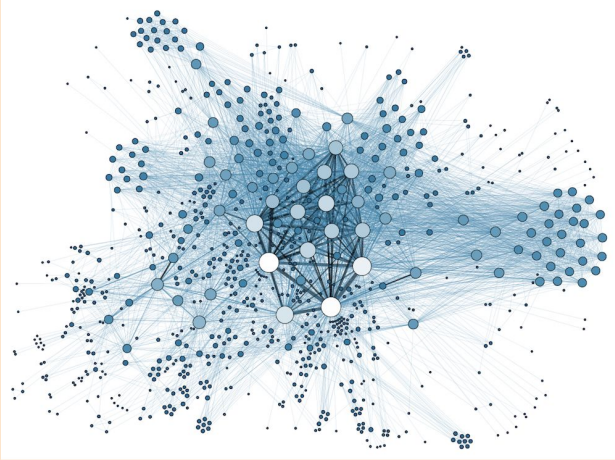
WHAT IS GRAPH DATABASE

"Graph Databases are rapidly gaining traction in the market as an effective method for deciphering meaning … allowing you to manage your data more efficiently. ... Your unstructured content is now discoverable allowing all types of users to quickly find the exact information for which they are searching."

- Forbes: "The Hype Around Graph Databases And Why It Matters"







VALUE FOR BUSINESS WITH GDB

SAVE TIME

- Visually comprehend data structure in seconds
- Save time from easily figuring out complicated relationships
- Quickly identify data glitches

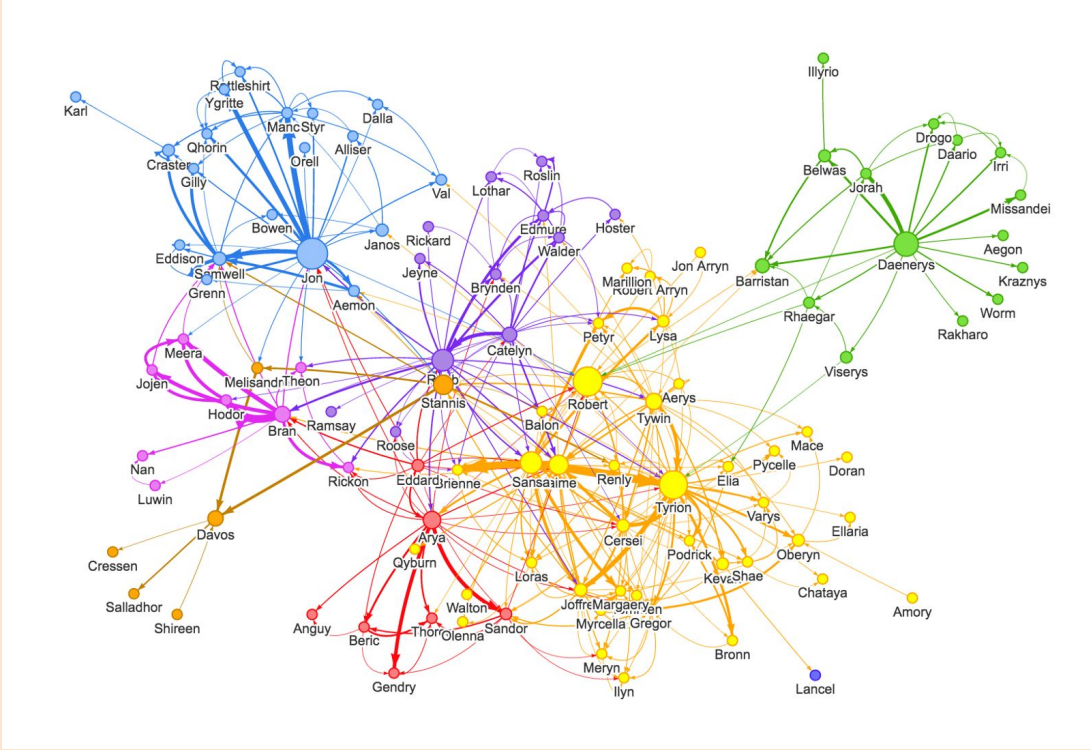
SAVE MONEY

- Invest in the right database systems
- Hire the right data experts
- Reduce costs from low-quality analysis based on poorly structured data
- Provide flexibility and scalability

HOW GDB ACHIEVES VALUE FOR BUSINESS

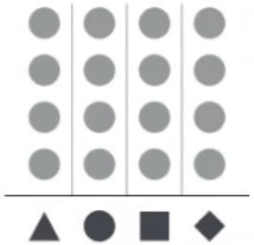
DATA POINTS RELATIONSHIP

- Understand the data network structure
- Monitor data network structure changes
- Identify closely interacting clusters of data
- Easy-to-read hierarchical relationships, i.e. tree-based structures



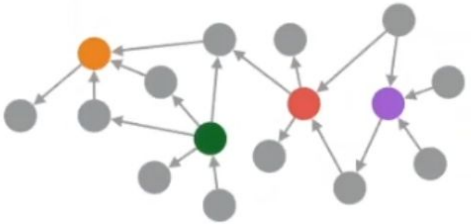
GRAPH DATABASE vs. RELATIONAL DATABASE

A way of representing data



Relational Database

Graph Database

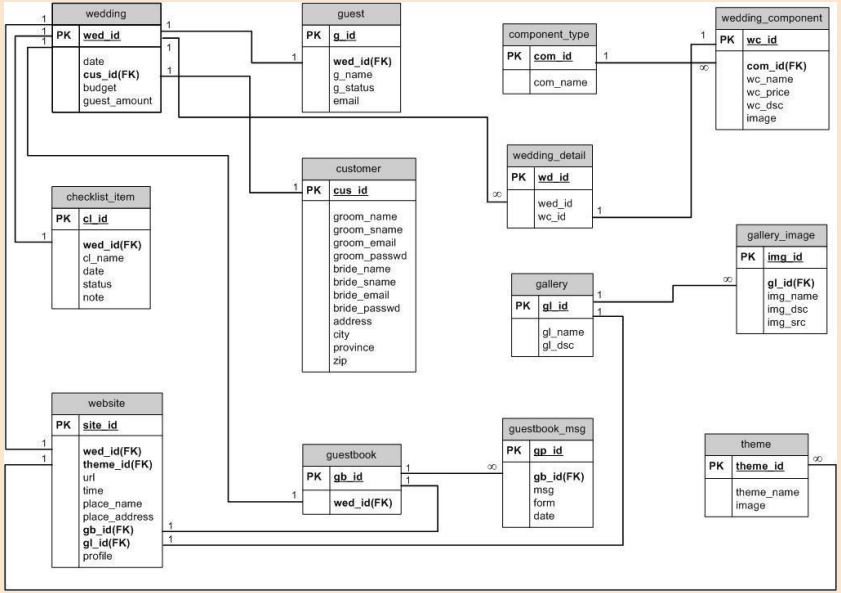


Structured
Pre-computed
Based on rigid rules

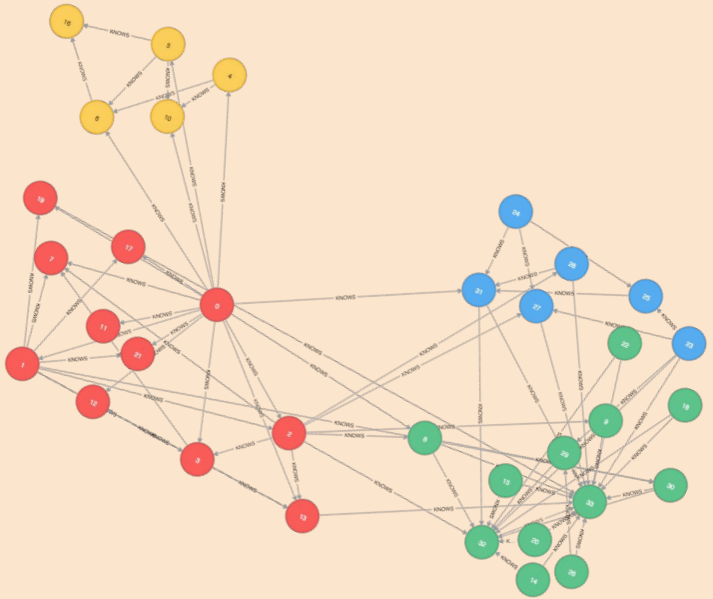
Highly Flexible
Real-Time Queries
Highly Contextual

GRAPH DATABASE vs. RELATIONAL DATABASE

RELATIONAL DATABASE



GRAPH DATABASE



GRAPH DATABASE vs. RELATIONAL DATABASE

RELATIONAL DATABASE

Object 1

PK		
CusID	(variable)	(variable)
CustID 1	…	…
CustID 2	…	…
CustID 3	…	…

Object 2

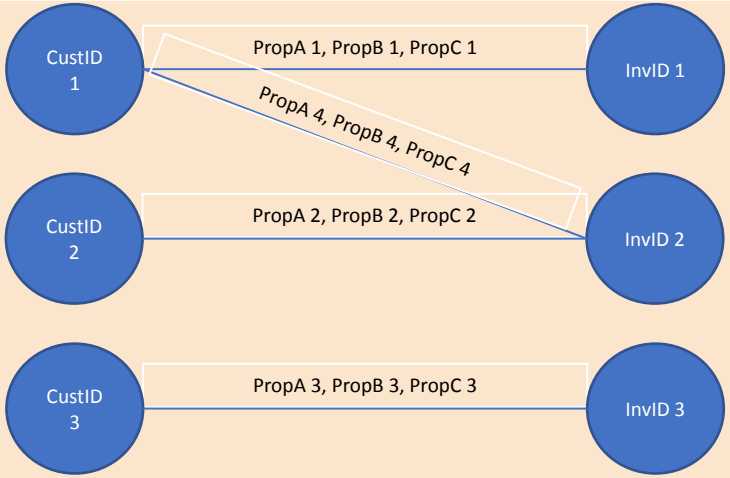
PK		
InvID	(variable)	(variable)
InvID 1	…	…
InvID 2	…	…
InvID 3	…	…

3rd table

FK	FK			
CustID	InvID	PropA	PropB	PropC
CustID 1	InvID 1	PropA 1	PropB 1	PropC 1
CustID 2	InvID 2	PropA 2	PropB 2	PropC 2
CustID 3	InvID 3	PropA 3	PropB 3	PropC 3
CustID1	InvID 2	PropA 4	PropB 4	PropC 4

- Additional tables to connect objects by properties
- Extract and visualize to understand the data
- Not easy to detect anomalies, missing data

GRAPH DATABASE



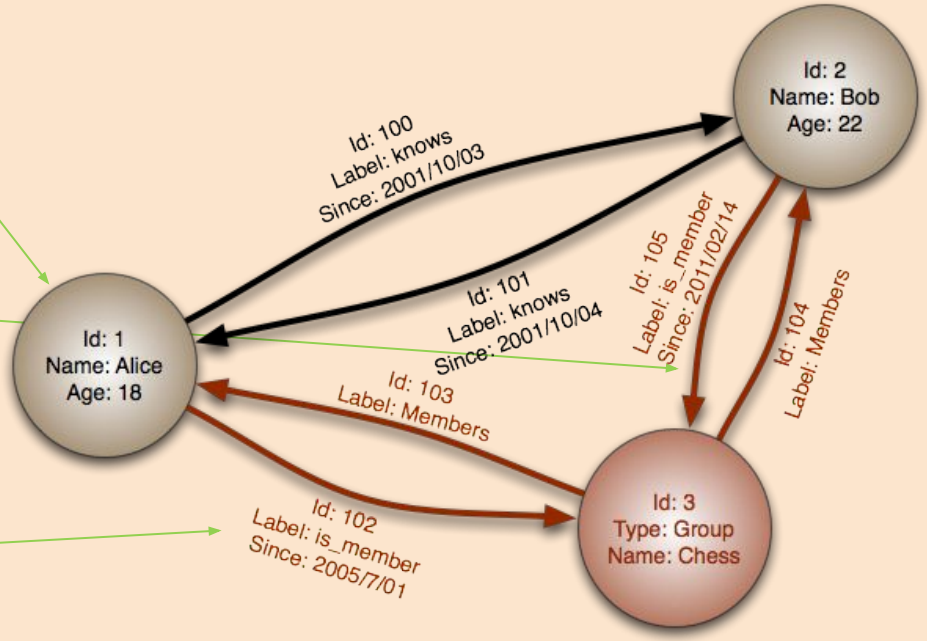
- GDB directly connects objects by their properties
- One glance at the data gives all the insights
- GDB directly shows anomalies, missing data

GDB MODULE STRUCTURE

Nodes
The equivalent of the ***record*** or ***row*** in a relational database.

Edges
The **lines that connect** nodes to other nodes, representing the relationship between them.

Properties
The **relevant information** related to the nodes or the relations between them.



Labeled Property Graphs vs RDF

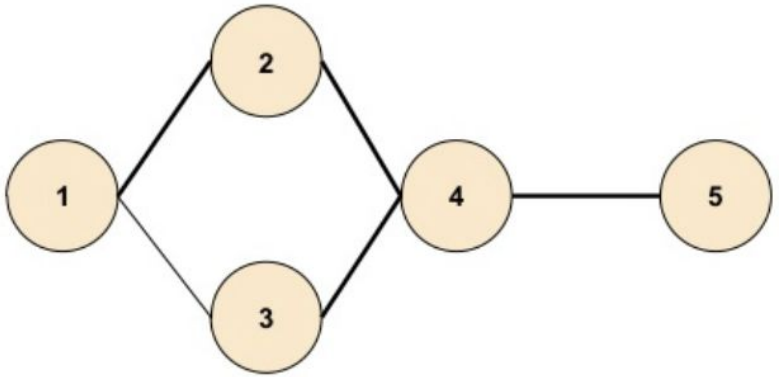
- LPG - General graph representation, plenty of graph analytics and ML libraries.

https://neo4j.com/blog/rdf-triple-store-vs-labeled-property-graph-difference/

- RDF - Resource Description Framework: W3C standard for data exchange in the Web. Publishing and linking data with formal semantics and no central control.

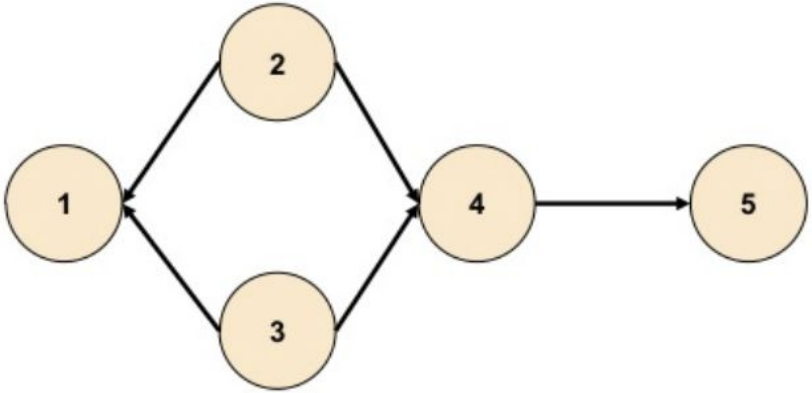
https://www.ontotext.com/knowledgehub/fundamentals/rdf-vs-property-graphs/

Undirected Graph



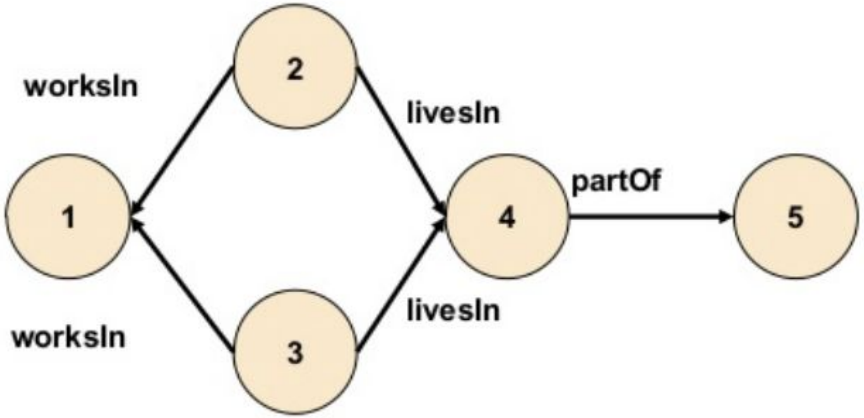
Source: www.ontotext.com

Directed Graph



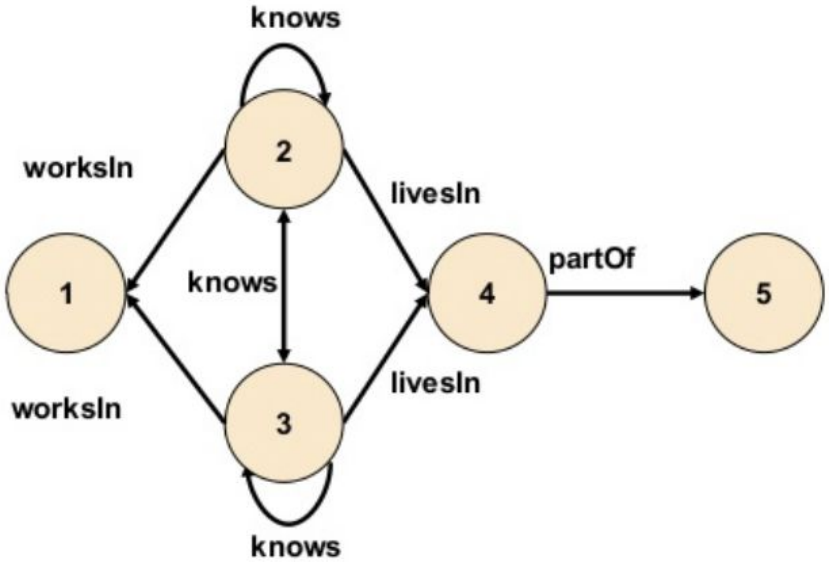
Source: www.ontotext.com

Directed Labeled Graph



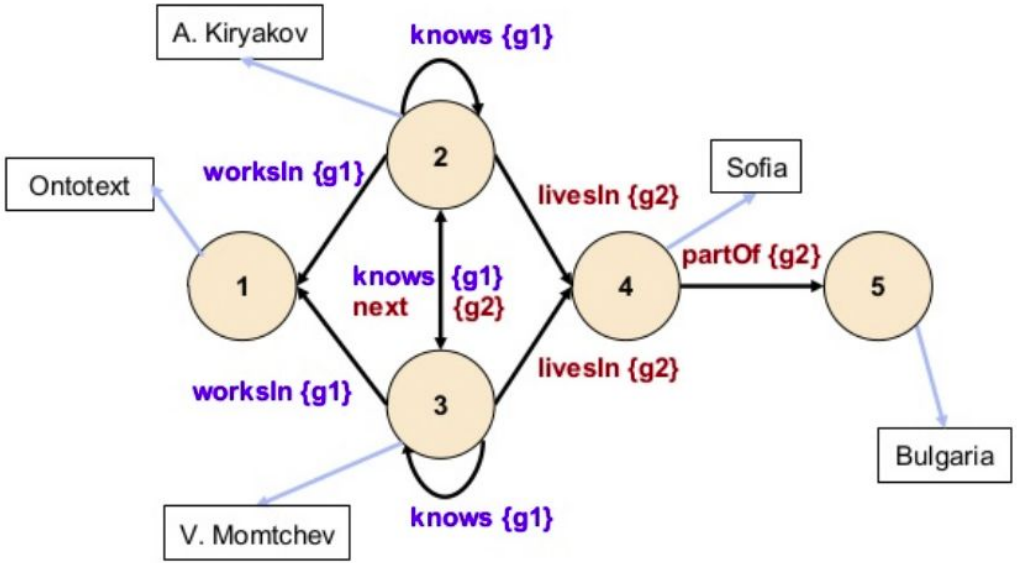
Source: www.ontotext.com

Directed Labeled Cyclic Graph



Source: www.ontotext.com

Directed Labeled Cyclic Multigraph with Node Labels

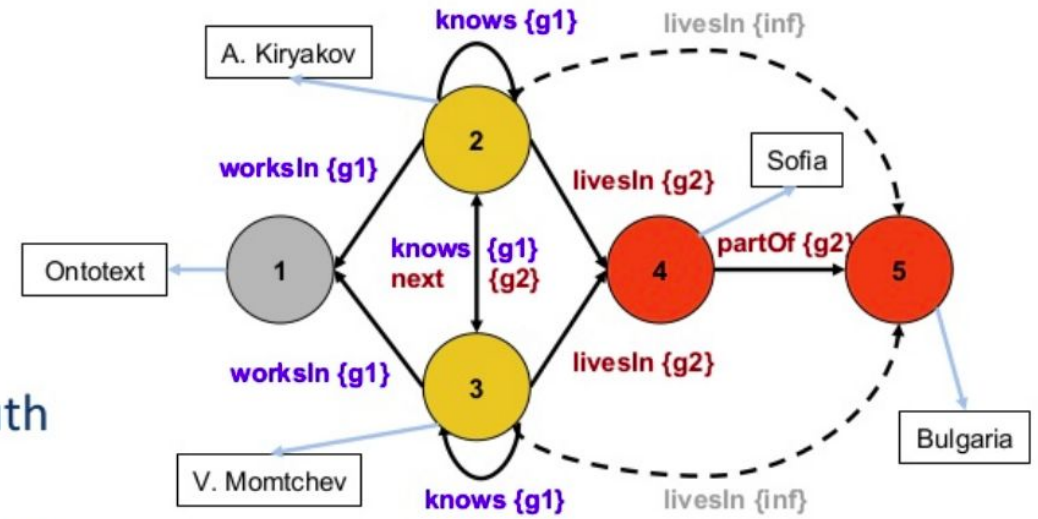


ontotext

Source: www.ontotext.com

RDF: Directed Labeled Cyclic Multigraph with Labels, Types, Logic and Semantics

Edges IDs:
1 - http://ontotext.com
2 - https://www.linkedin.com/in/atanas-kiryakov
3 - https://www.linkedin.com/in/vassil-momtchev
4 - https://en.wikipedia.org/wiki/Sofia
5 - https://en.wikipedia.org/wiki/Bulgaria



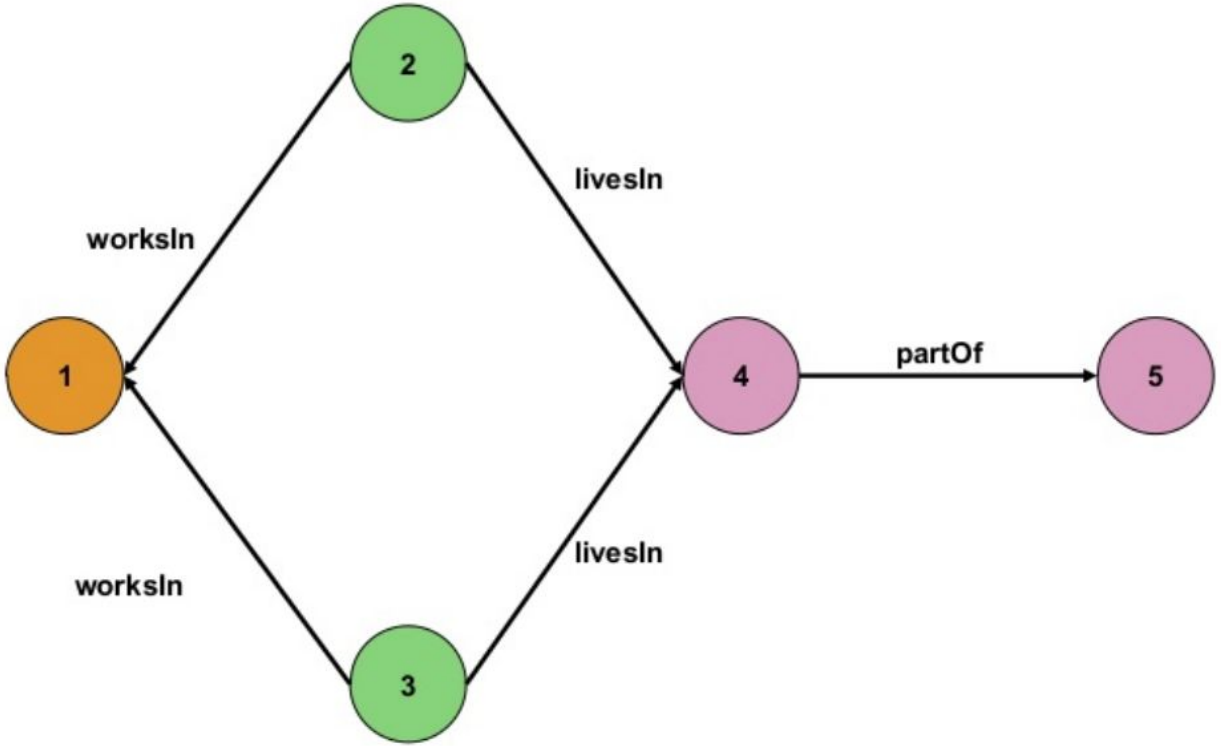
Optimized for:

- Flexible web model
- Multiple versions of the truth
- Global identifiers
- Information schema language
- Logic inference and data quality

ontotext

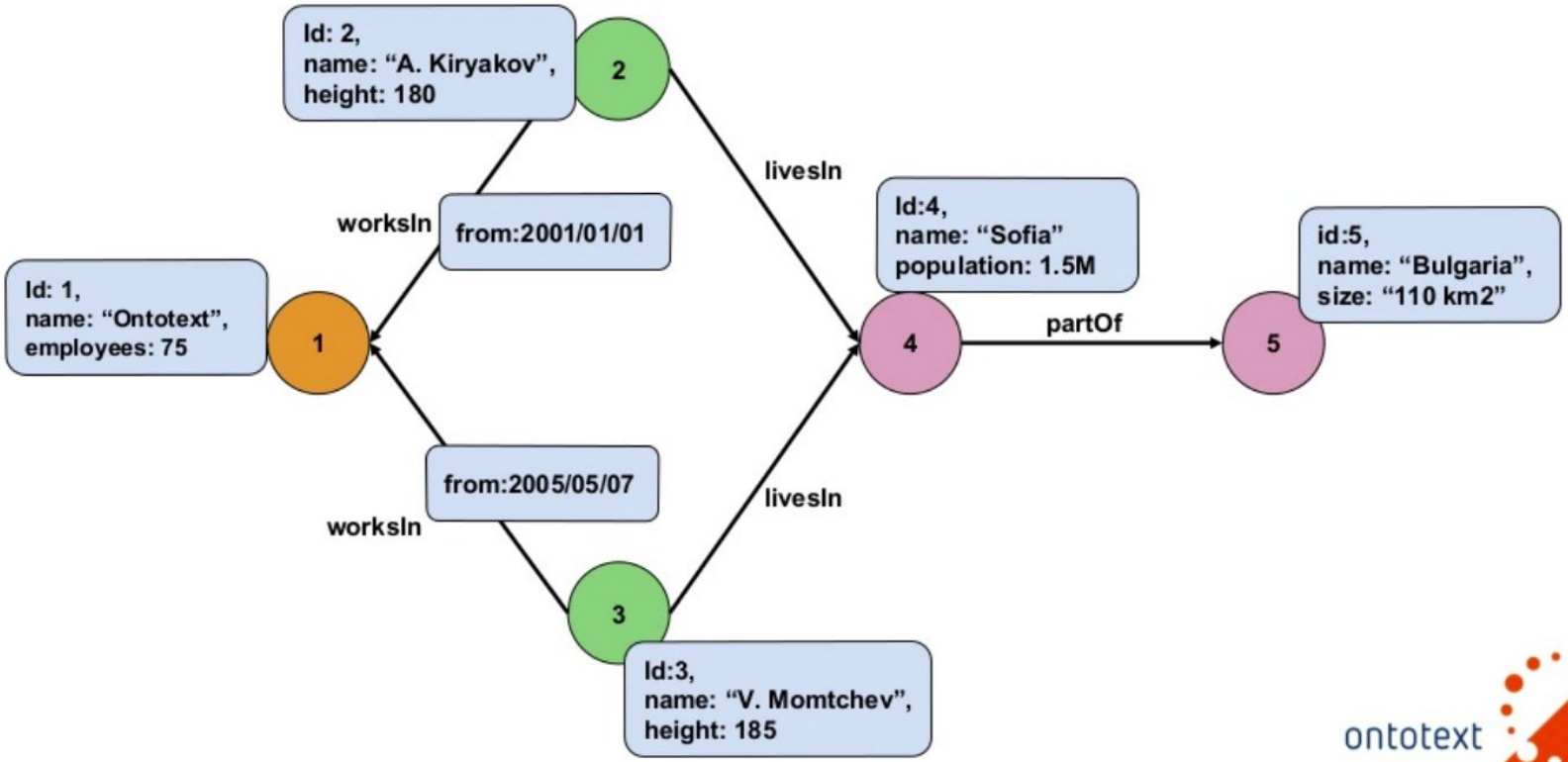
Source: www.ontotext.com

Directed Labeled Graph with Types



Source: www.ontotext.com

Directed Property Labeled Graph with Types



Source: www.ontotext.com

Feature	RDF	Property Graph
Expressivity	Arbitrary complex descriptions via links to other nodes; no properties on edges With RDF* the model gets much more expressive than PG	Limited expressivity, beyond the basic directed cyclic labeled graph Properties (key-value pairs) for nodes and edges balance between complexity and utility
Formal semantics	Yes, standard schema and model semantics foster data reuse and inference	No formal model representation
Standardization	Driven by W3C working groups and standardization processes	Different competing vendors
Query language	SPARQL specifications: Query Language, Updates, Federation, Protocol (end-point)...	Cypher, PGQL, GCore, GQL (no standard)
Serialization format	Multiple serialization formats	No serialization format
Schema language	RDFS, OWL, Shapes	None

Source: www.ontotext.com

Feature	RDF	Property Graph
Designed for	Linked Open Data (Semantic Web): Publishing and linking data with formal semantics and no central control	Graph representation for analytics
Processing Strengths	Set analysis operations (as in SQL, but with schema abstraction and flexibility)	Graph traversal Plenty of graph analytics and ML libraries
Data Management Strengths	Interoperability via global identifiers Interoperability via a standard: schema language, protocol for federation, reasoning semantics Data validation, data type support, multilinguality	Compact serialization, shorter learning curve, functional graph traversal language (Gremlin)
Main use cases	Data-driven architecture Master/reference data sharing in enterprises Knowledge representation	Graph analytics and path search

Source: www.ontotext.com

USE CASE: GDB FOR SMART CITIES

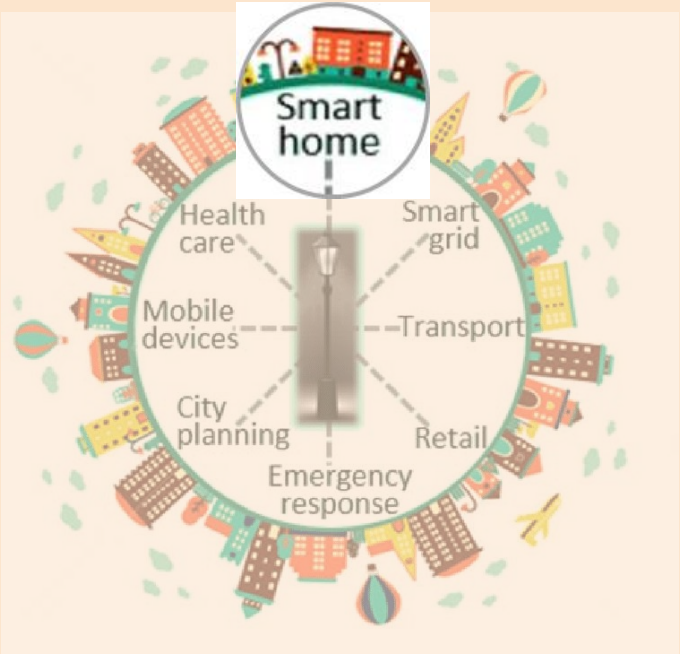
GDB in smart cities allows for:

- Same spatial domain
- Multiple networks
- Network interaction

GDB way of visualization enables:

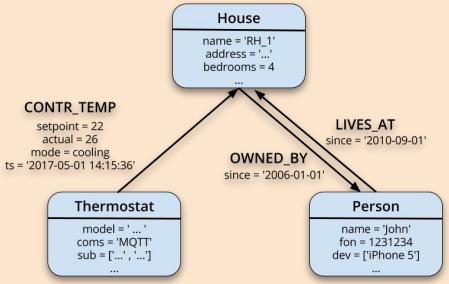
- Planning for “smart hubs”
- Estimating congestion patterns
- Resources distribution optimisation

USE CASE: GDB FOR SMART CITIES: SMART HOME



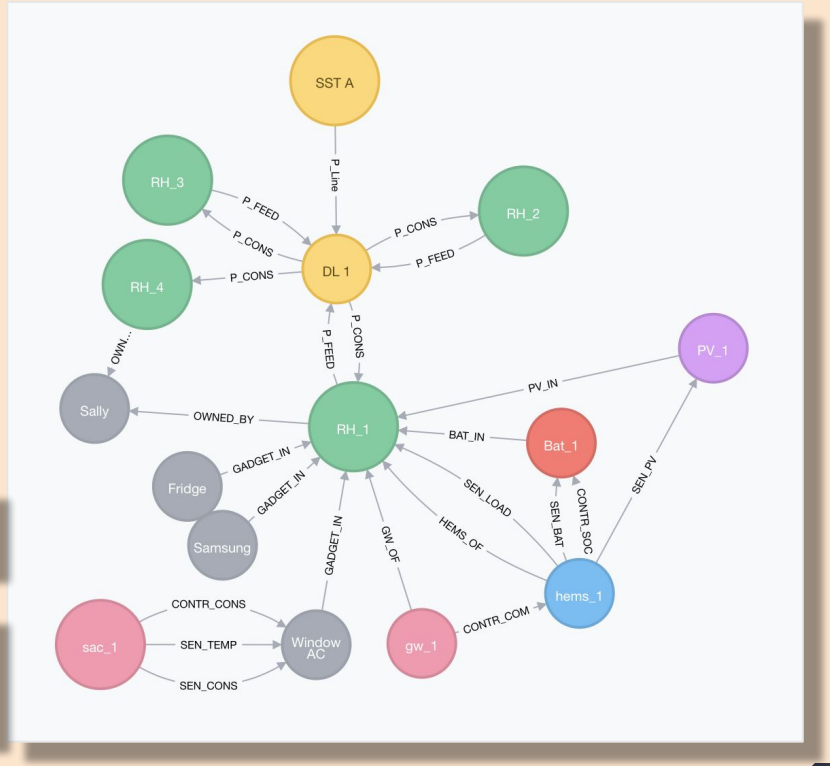
USE CASE: GDB FOR SMART CITIES: SMART HOME

- SST_A - electricity distribution company
- DL_1 - two-way smart substation
- Monitoring saved energy from the community

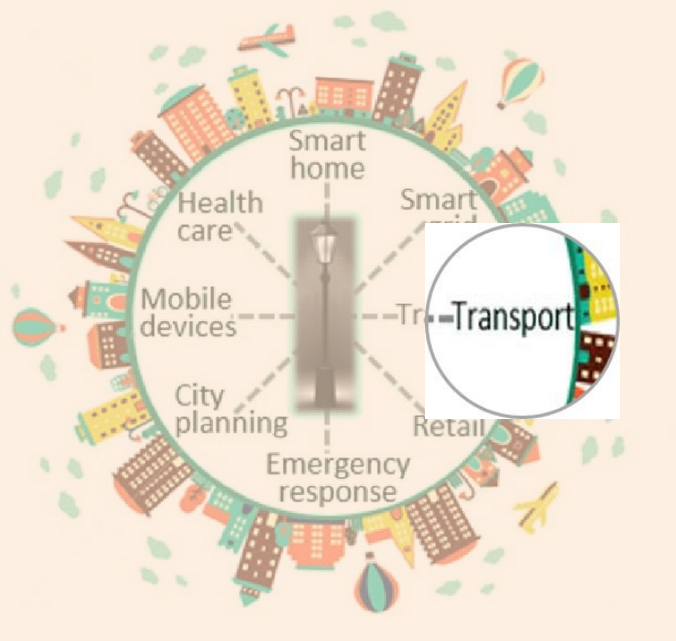



```
MATCH (t:Thermostat) - [:CONTR_TEMP] -> () -
    [:OWNED_BY] -> (p) RETURN p.name, p.fon
```

```
MATCH (:Res_Home) - [p:P_FEED] -> () <-
   [:P_Line] - (:Substation {name: 'SST A'})
WHERE p.ts >= timestamp()-600000
RETURN SUM(p.wattage)
```

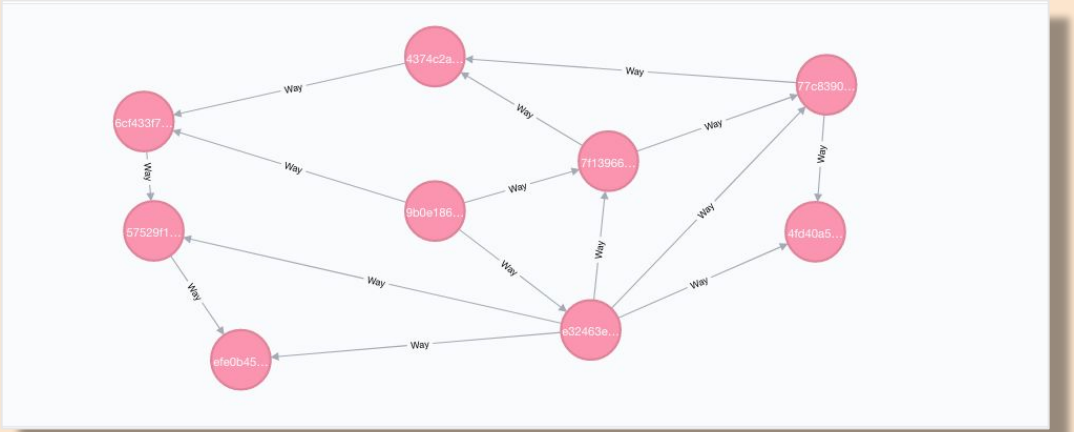


USE CASE: GDB FOR SMART CITIES: TRANSPORT



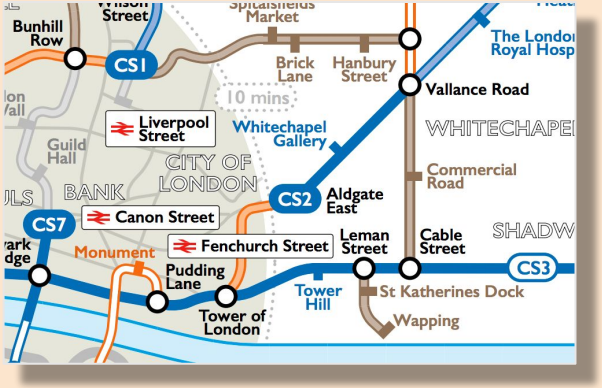
USE CASE: GDB FOR SMART CITIES: TRANSPORT

GDB transportation network in raw format



GDB transportation networks visualized in production

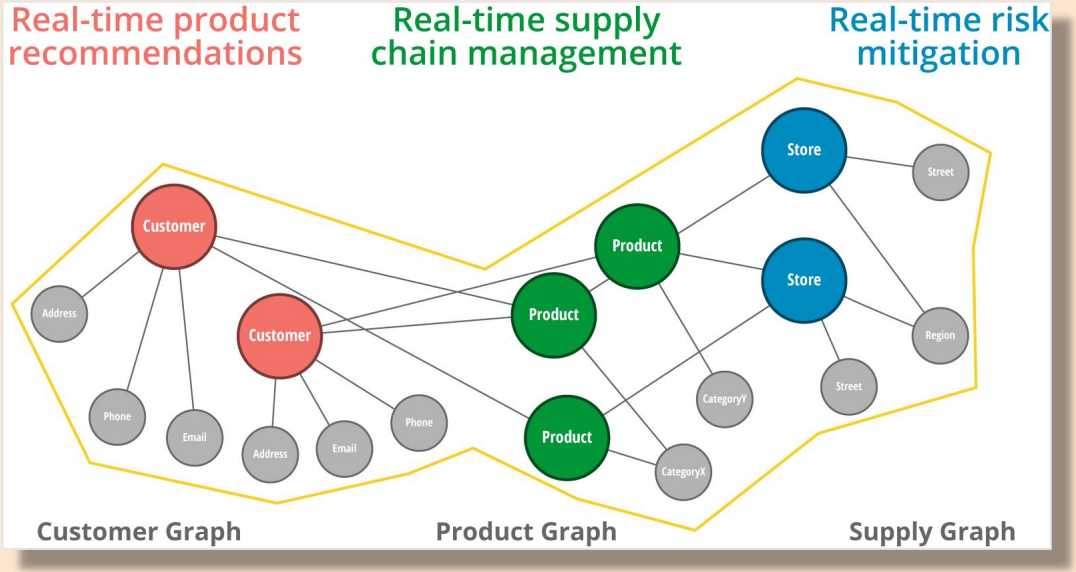
- Same spatial domain
- Multiple transportation networks
- Interaction between them



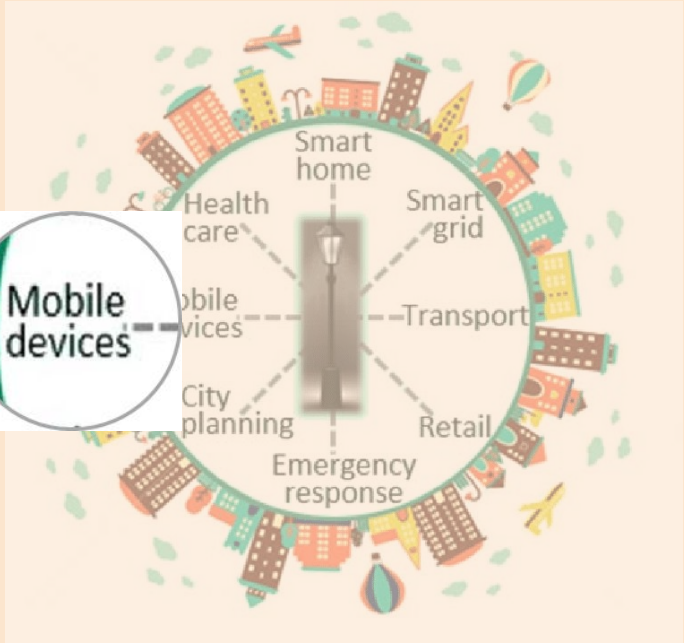
USE CASE: GDB FOR SMART CITIES: RETAIL

USE CASE: GDB FOR SMART CITIES: RETAIL

GDB allows for real time management and monitoring

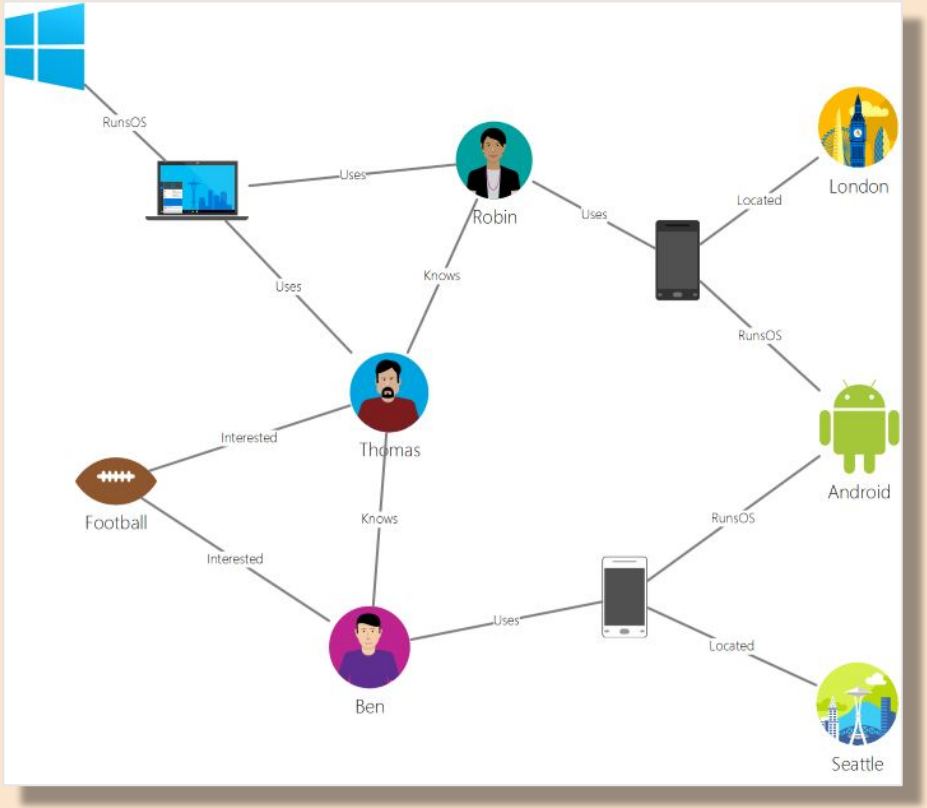


USE CASE: GDB FOR SMART CITIES: MOBILE DEVICES



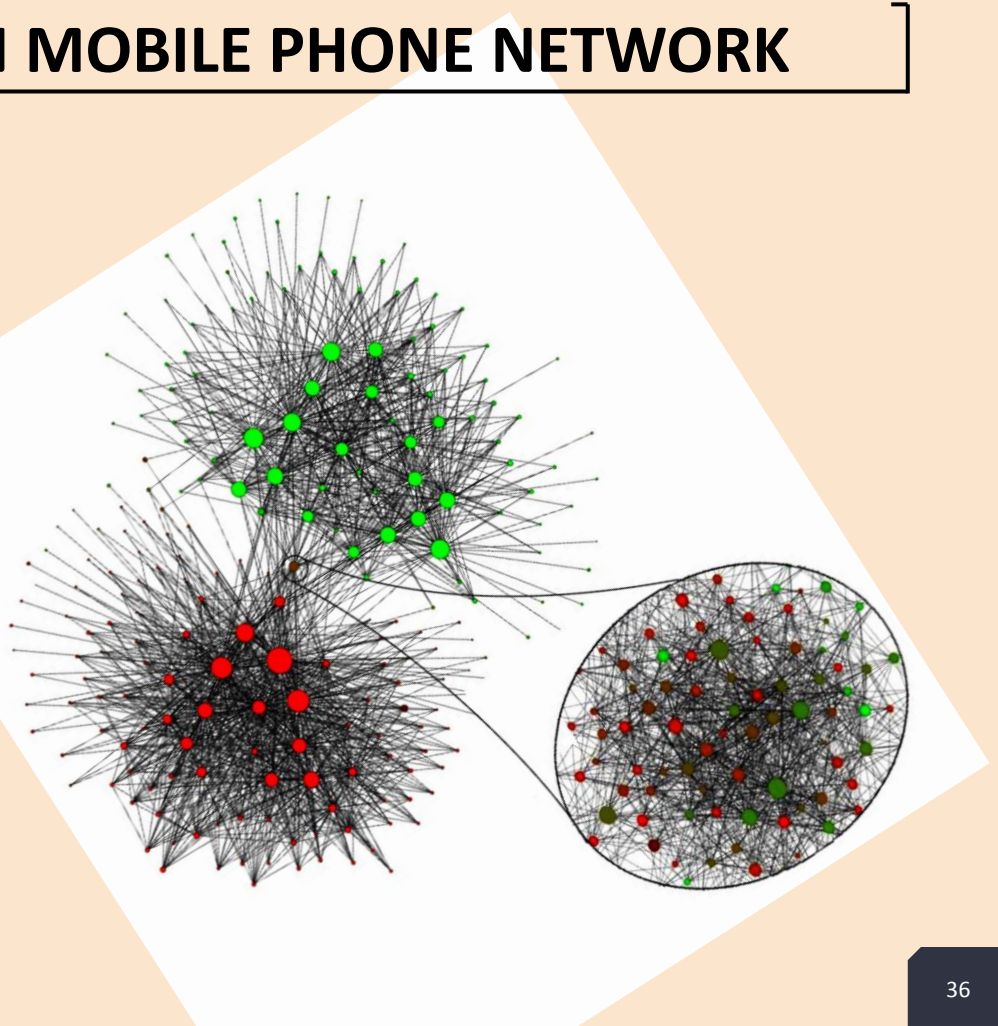
USE CASE: GDB FOR SMART CITIES: MOBILE DEVICES

- Enables mapping services provided by Google Maps, Citymapper, Moovit, etc.
- Mobile devices allow for traffic monitoring and congestion forecasting
- The scalability of GDB allows for support of millions of users in real time



USE CASE: GDB FOR BELGIAN MOBILE PHONE NETWORK

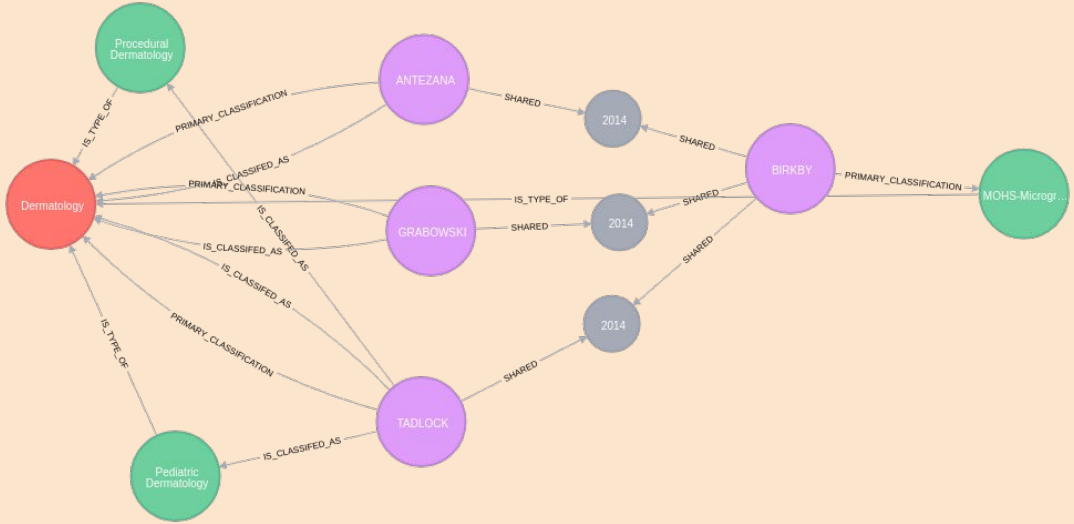
- About 2 million customers on the network
- Node size is proportional to the number of individuals in the community
- Colour on a red-green (French-Dutch)scale represents main language spoken
- In-between GDB recognizes a cluster (communication nodes) - community where both languages are spoken



USE CASE: GDB FOR SMART CITIES: HEALTH CARE

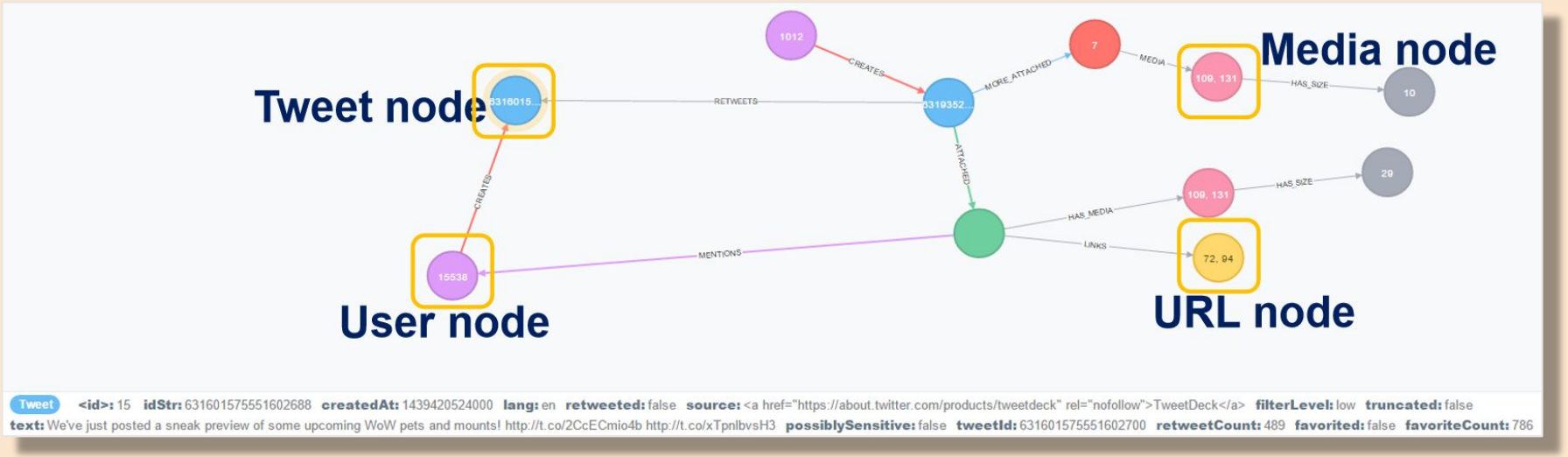
USE CASE: GDB FOR SMART CITIES: HEALTH CARE

- Recommendation system in health care
- Easily allows for querying useful patient-doctor relationship information




```
MATCH pth= (pc1:Provider_Taxonomy)<--(mohs_surgeon:Provider)-[ps1:SHARED]->(sp:Shared_Patient)<-[ps2:SHARED]-(refer
ring:Provider)-[ica:IS_CLASSIFED_AS]->(pc2:Provider_Taxonomy) where mohs_surgeon.provider_last_name_legal_name = "B
IRKBY" and mohs_surgeon.provider_first_name = "CRAIG" return pth
```

USE CASE: GDB FOR A TWEET

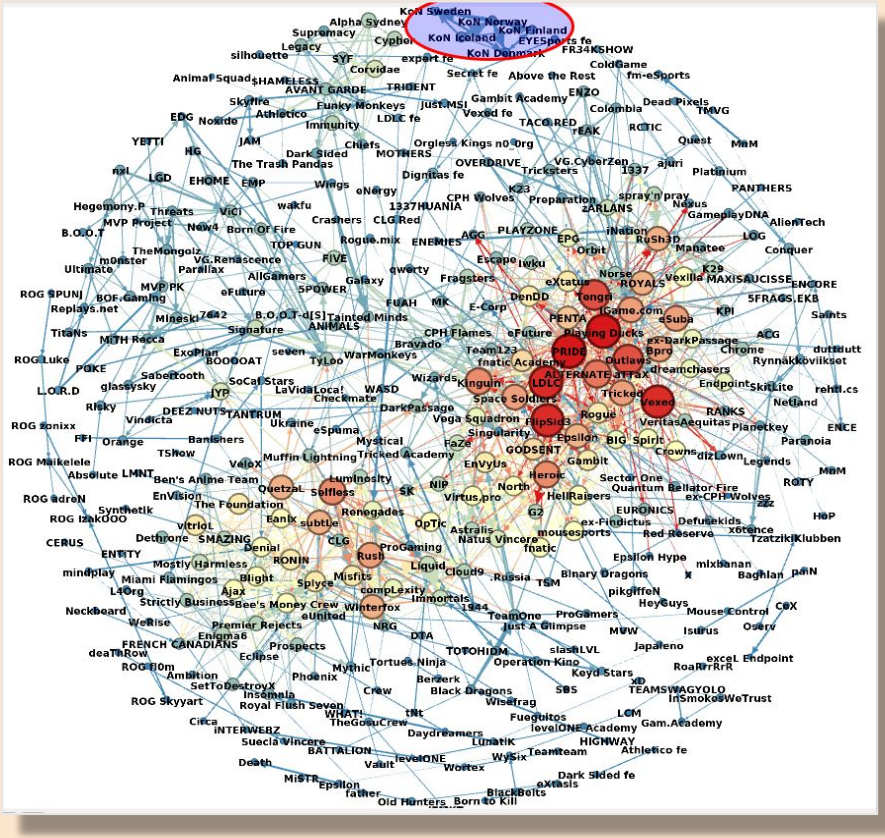


Use Case:

The example above shows the graph database structure of a single tweet. The GDB visualization includes the nodes Tweet, User, Media, and URL, as well as how they link with each other, and based on what properties.

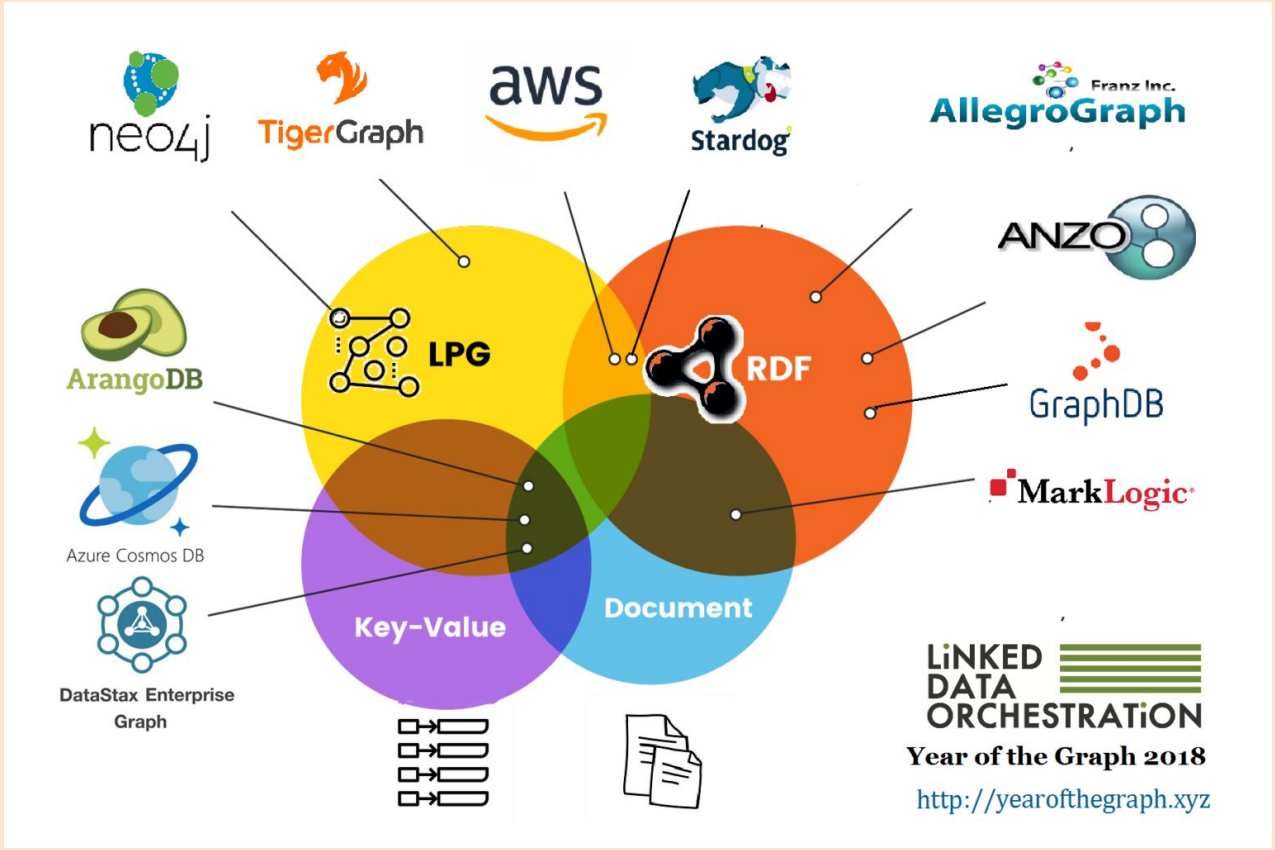
USE CASE: GDB FOR CS: GO GAME TEAMS RANKING

- About 3,000 teams on the network
- Node size is proportional to the strength of the team
- Colour on a red-orange-yellow-green-blue scale represents the strength of the team
- The thickness of the link represents the difference of the strengths of the corresponding teams



Neo4j Licensing

- Neo4j Community Edition: fully opensource, licensed and distributed under GPL v3.
- Commercial licensing options with free options for academic, startup and development uses.
- Neo4j Enterprise: Google Compute Platform, Microsoft Azure, Amazon EC2.
- Neo4j Aurora: the simplest way to run Neo4j on the cloud.
  Subscription-based product, completely automated and fully managed.


neo4j
TigerGraph
aws
Stardog
Franz Inc.
AllegroGraph
ANZO
ArangoDB
LPG
RDF
GraphDB
MarkLogic
Azure Cosmos DB
Key-Value
Document
DataStax Enterprise Graph
LINKED DATA ORCHESTRATION
Year of the Graph 2018
http://yearofthegraph.xyz

Thank you for your attention!

Evgeniy Ivanov Marinov:

GATE Institute

Email: evgeniy.iv.marinov@gmail.com; evgeniy.marinov@gate-ai.eu