

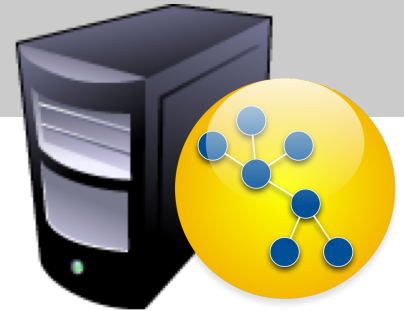
# Getting Started with Graph Databases

[rik@neotechnology.com](mailto:rik@neotechnology.com)

# Agenda

- Introduction
  - NO-SQL context
  - What is Neo4j?
  - When/why should I use it?
- Graph Queries
  - Cypher query language
  - Create and query data
- Graph Visualisations
- Case Studies
- Q&A

# Introduction



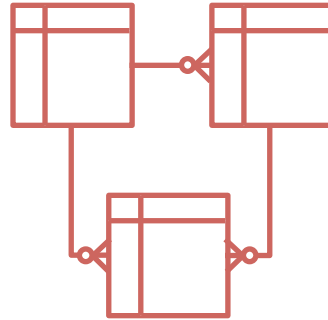
NOSQL is simply...

Not Only SQL



# What's so bad about

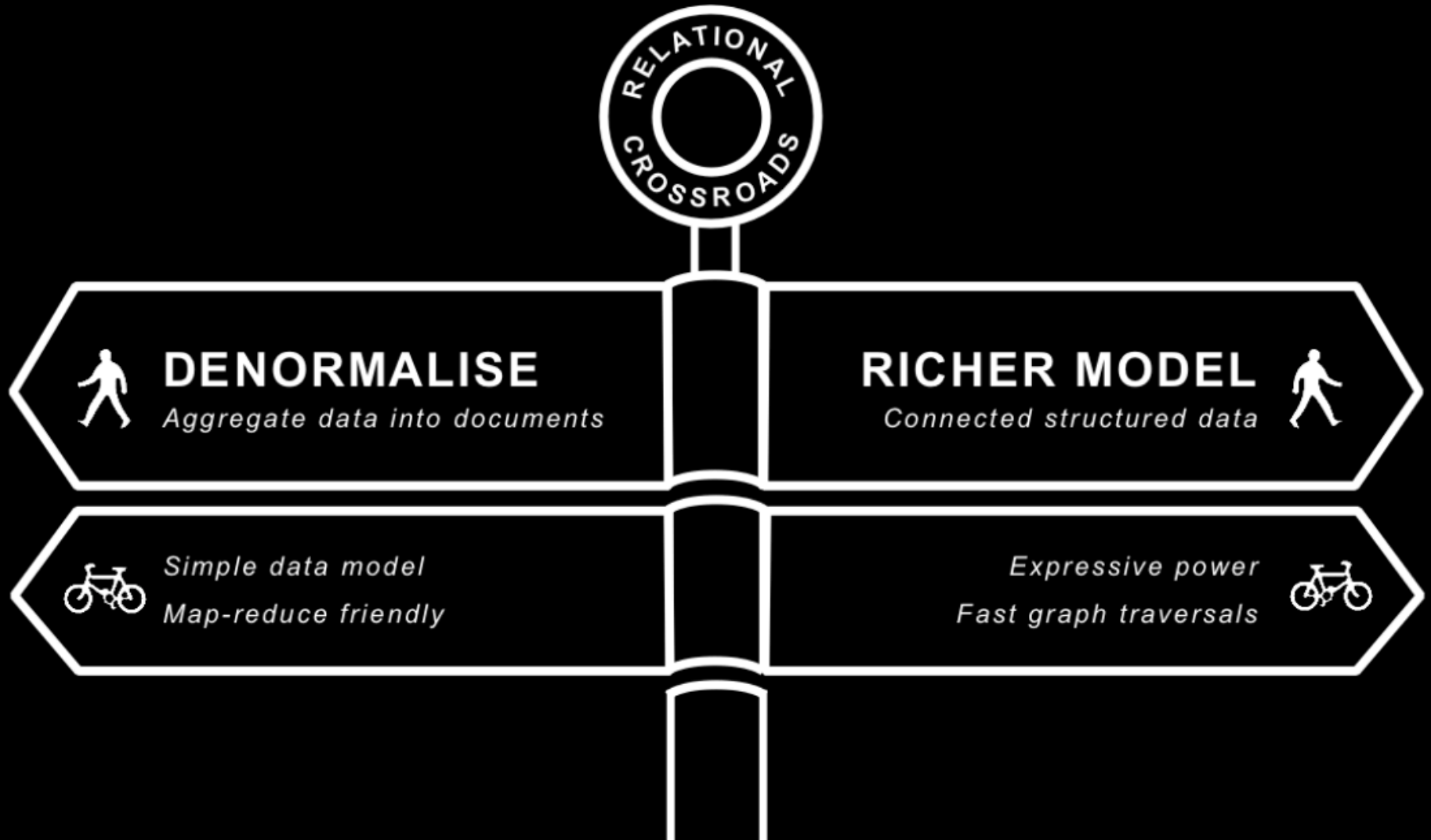
## Relational

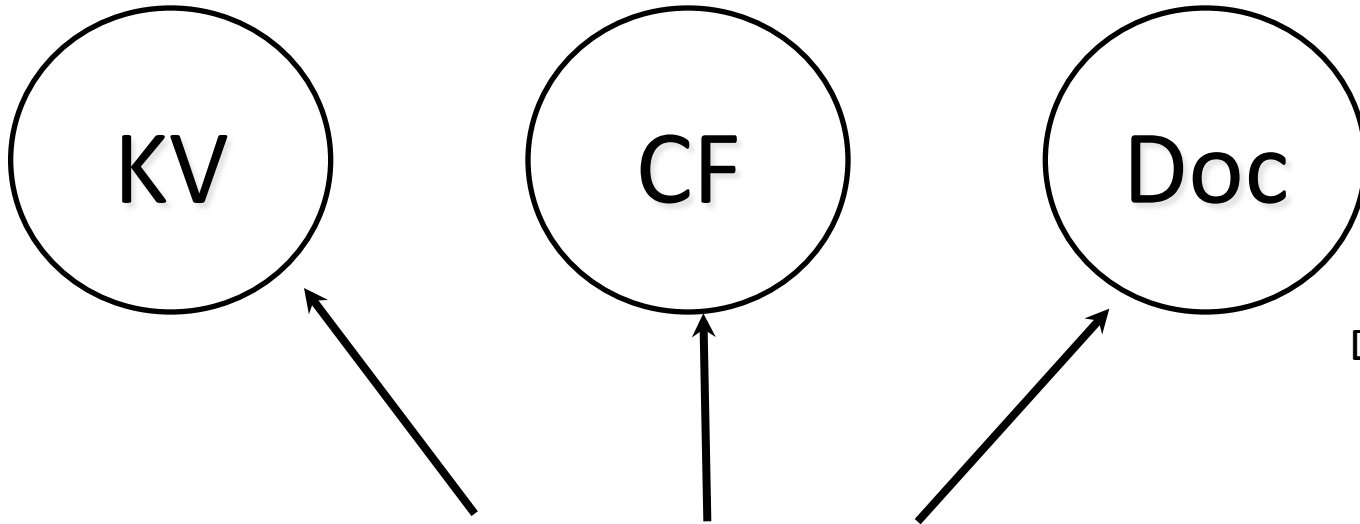


**VOLUME**

**Complexity**

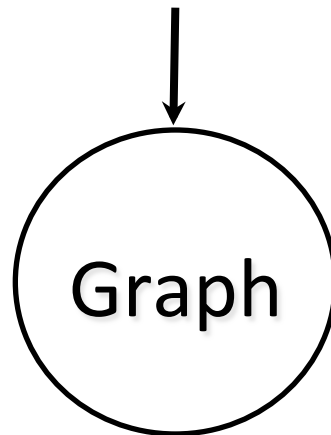
# The Relational Crossroads





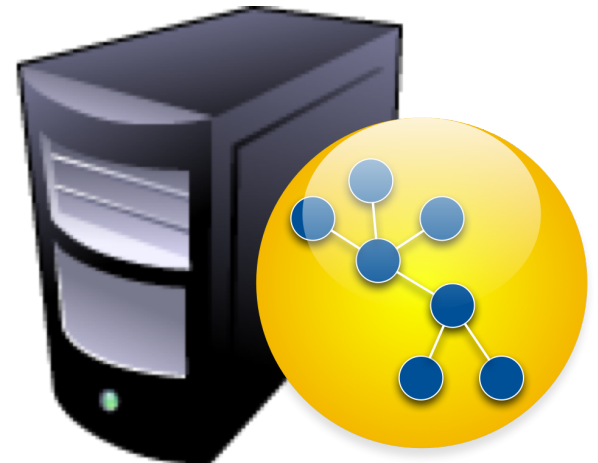
Four NOSQL Categories  
arising from the “relational crossroads”

Normalise



# So what is a graph database?

- OLTP database
  - “end-user” transactions
- Model, store, manage data as a graph



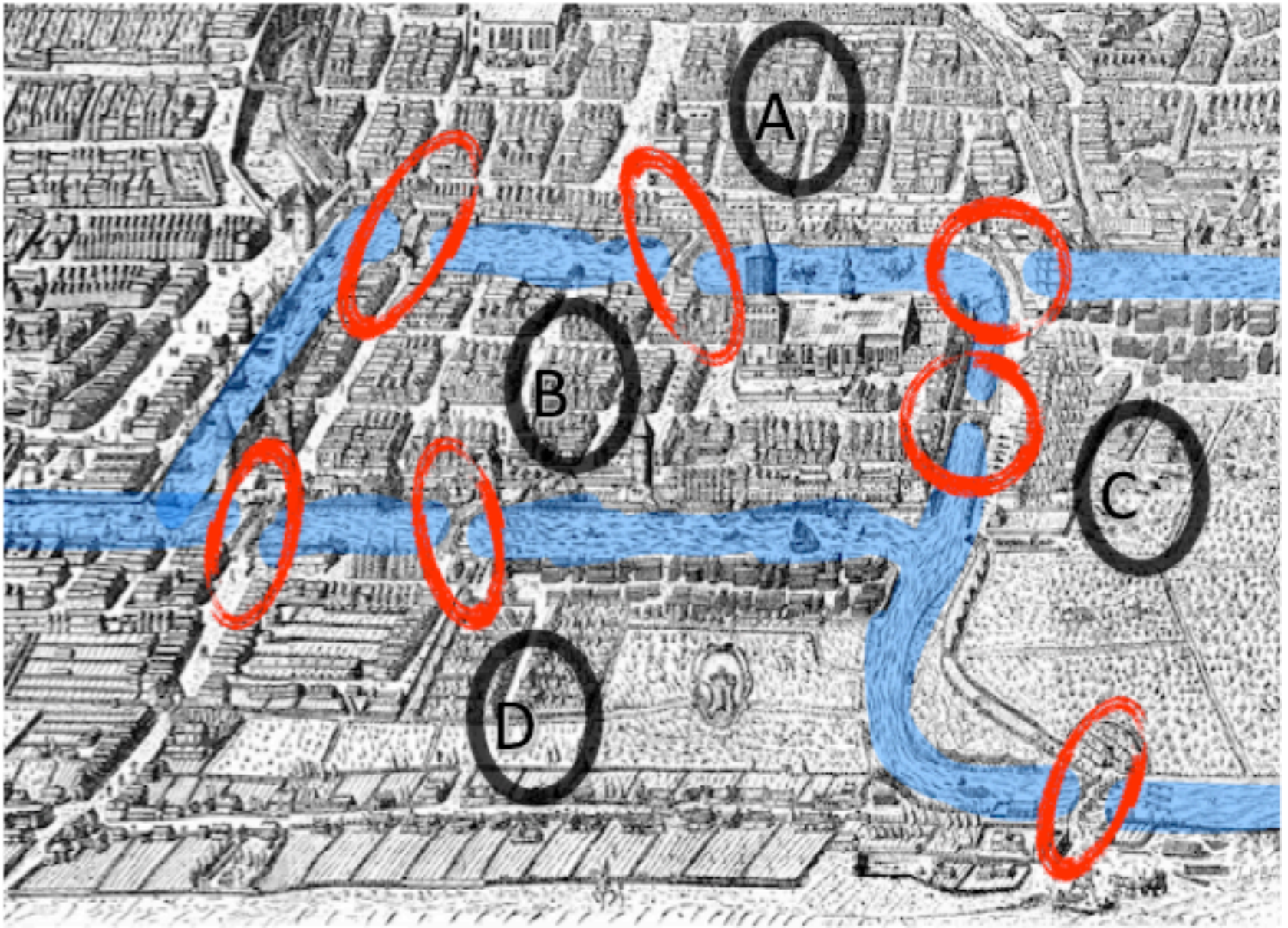


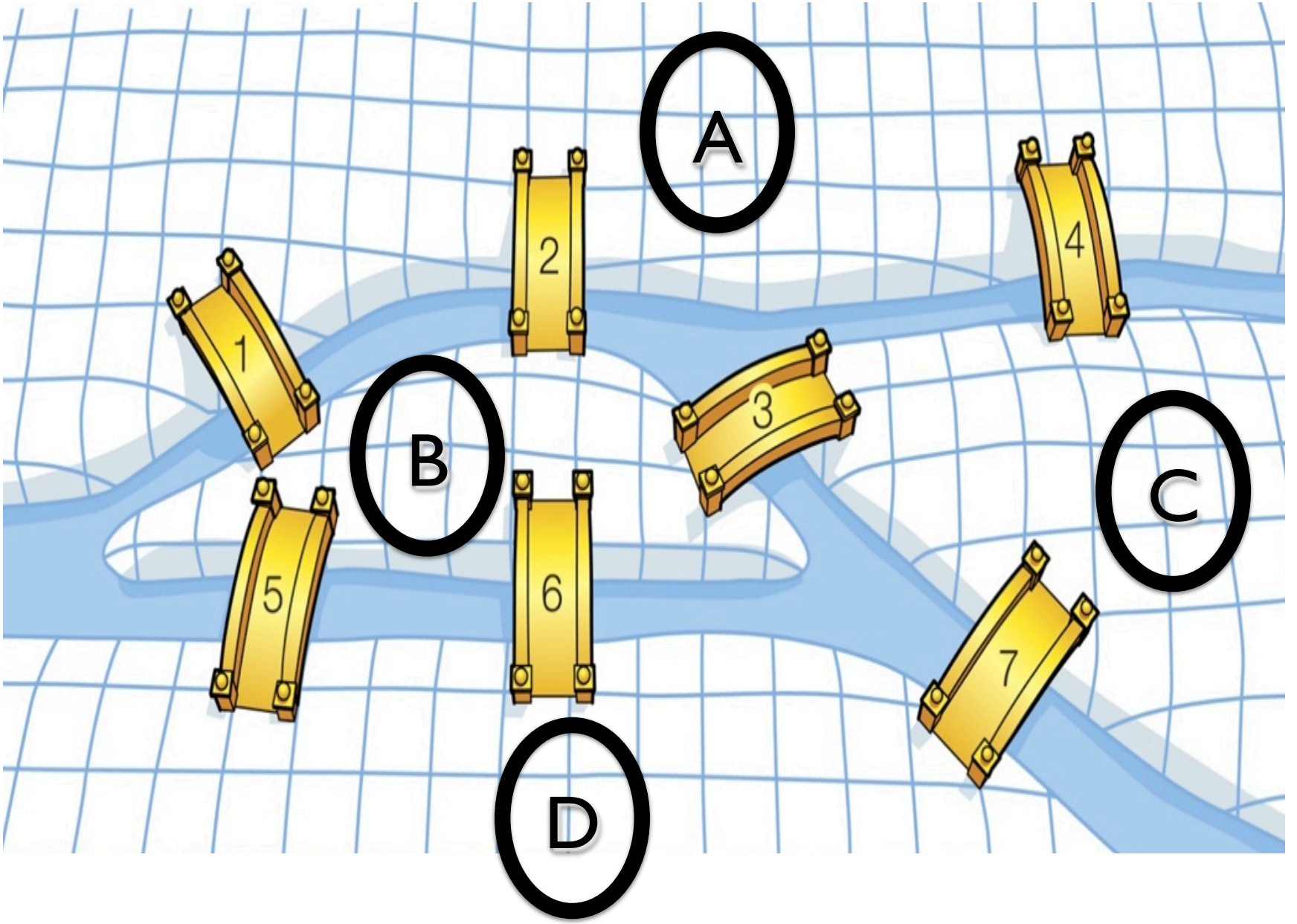
## Meet Leonhard Euler

- Swiss mathematician
- Inventor of Graph Theory (1736)

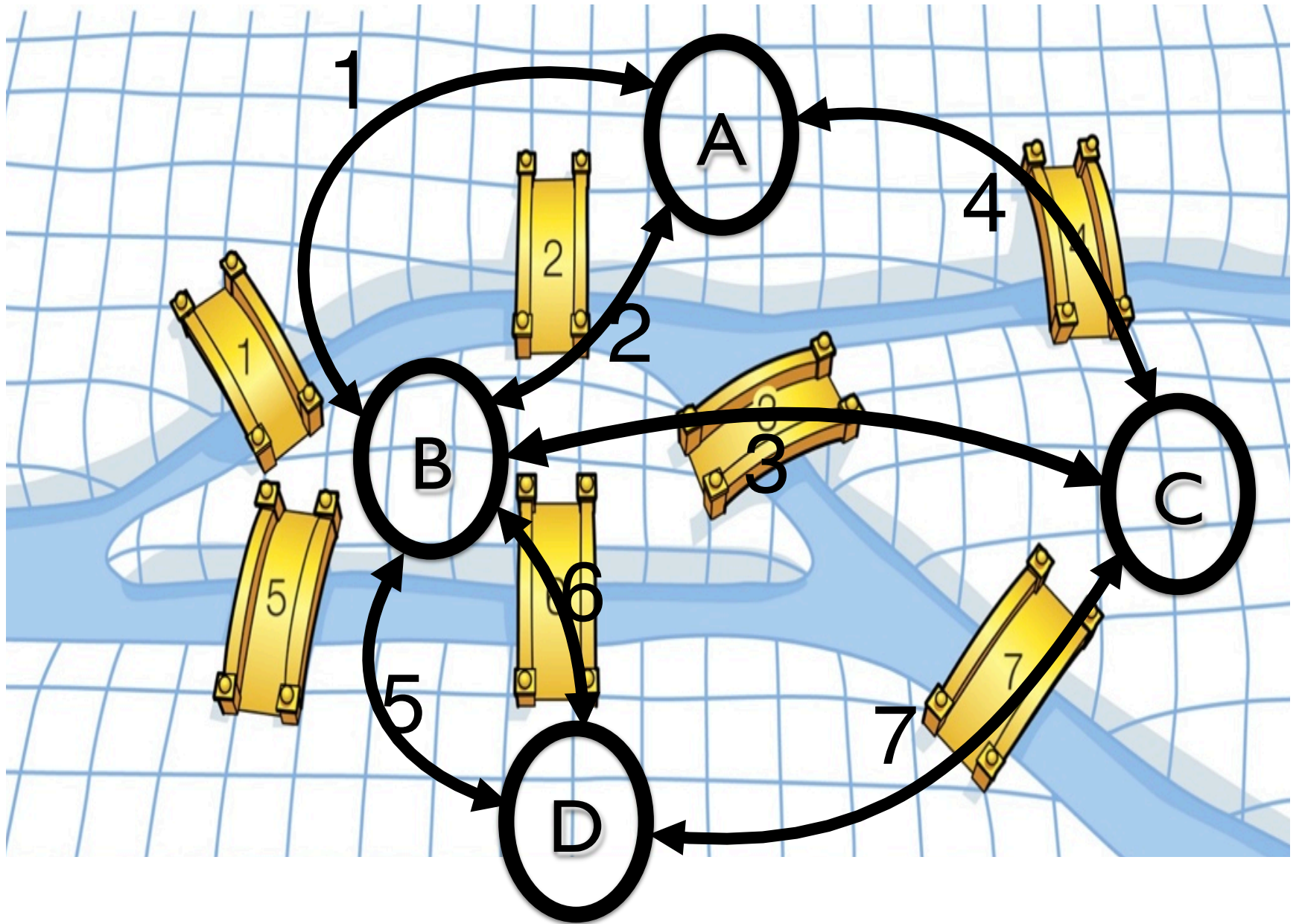


# Königsberg (Prussia) - 1736





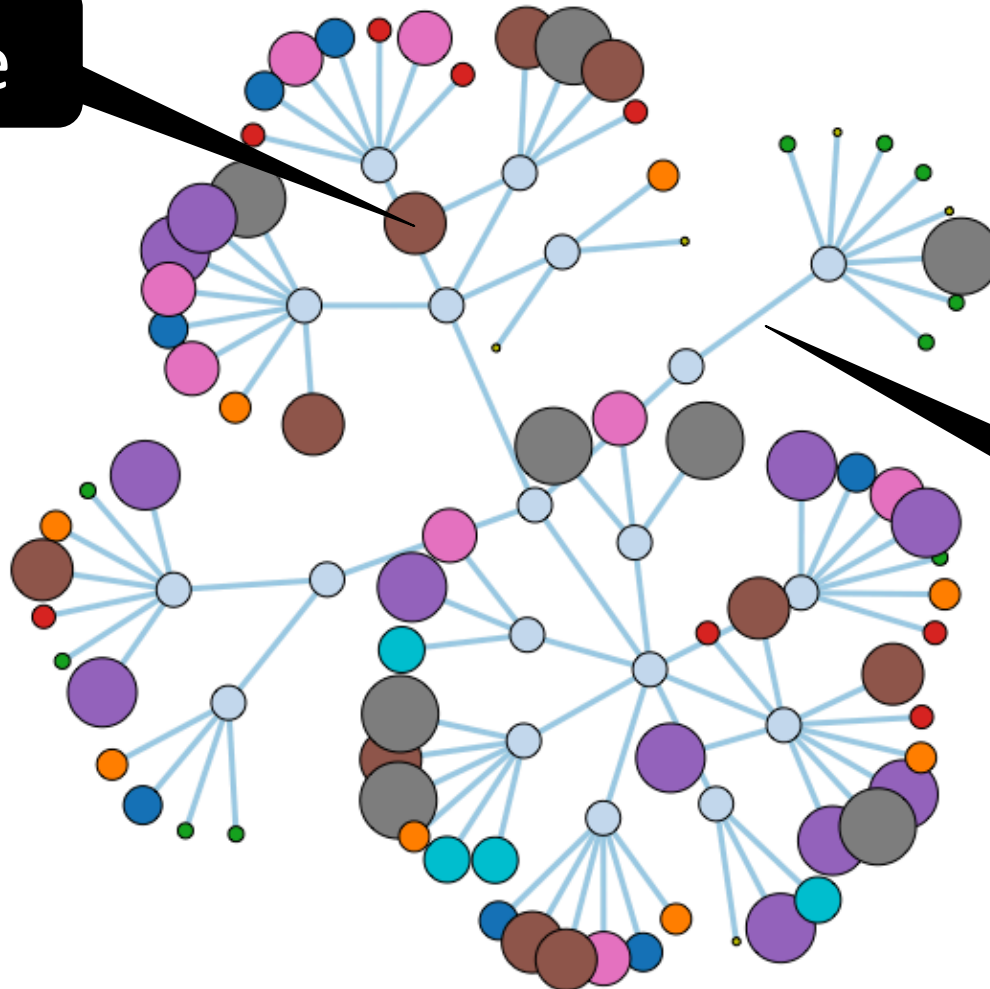






# What is a graph?

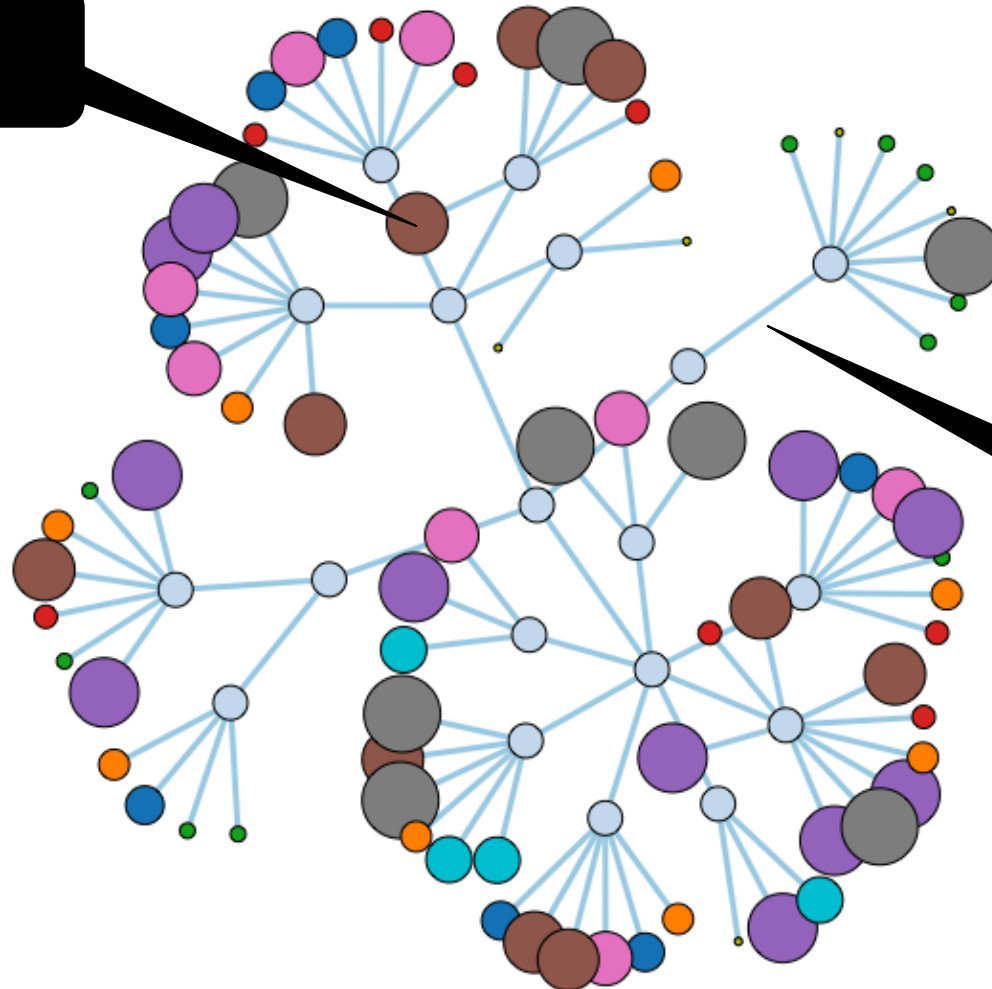
Vertice



Edge

# What is a graph?

Node



Relationship

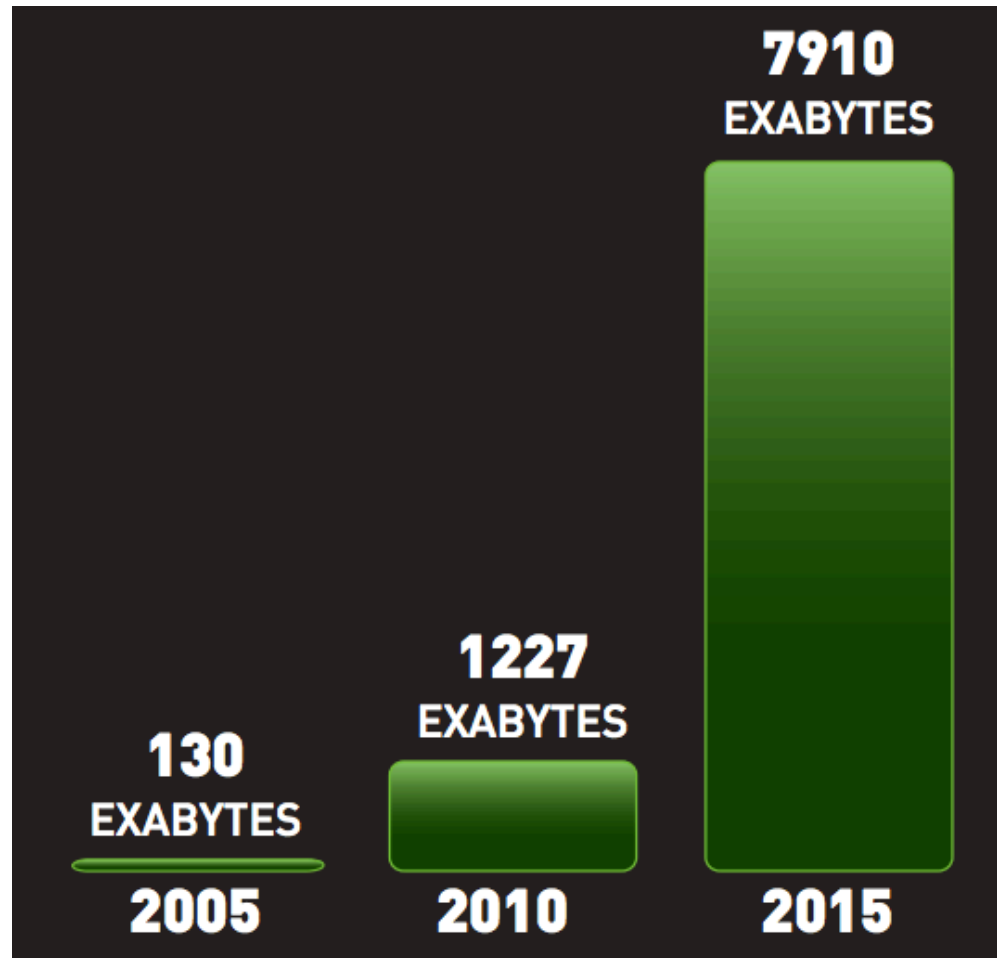
What are graphs good for?

Complexity

# Data Complexity

*complexity = f(size, semi-structure, connectedness)*

# Size

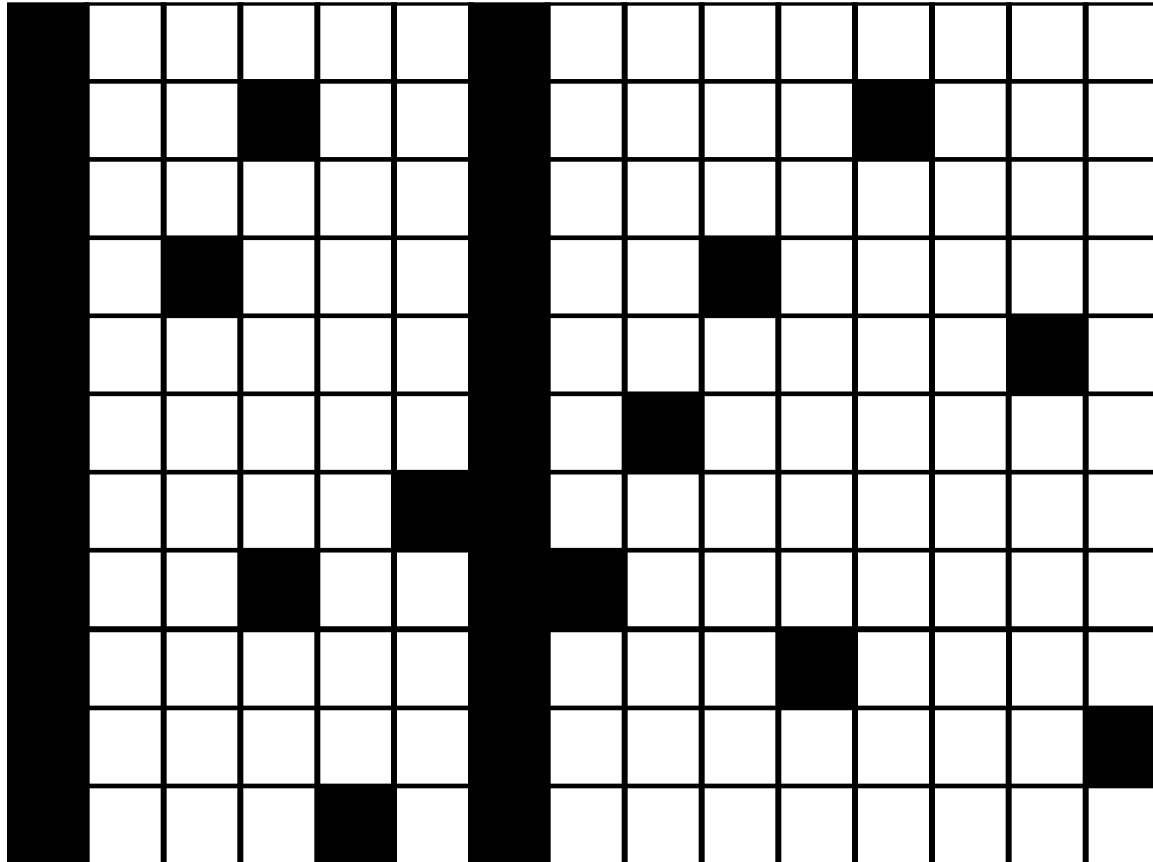


Name (Symbol)	Value
kilobyte (kB)	$10^3$
megabyte (MB)	$10^6$
gigabyte (GB)	$10^9$
terabyte (TB)	$10^{12}$
petabyte (PB)	$10^{15}$
exabyte (EB)	$10^{18}$
zettabyte (ZB)	$10^{21}$
yottabyte (YB)	$10^{24}$

# The Real Complexity

*complexity = f(size, **semi-structure**, **connectedness**)*

# Semi-Structure



# Semi-Structure

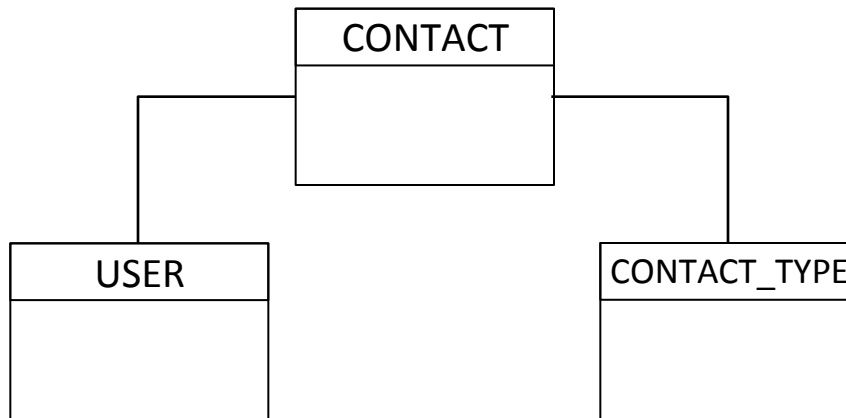
USER_ID	FIRST_NAME	LAST_NAME	EMAIL_1	EMAIL_2	FACEBOOK	TWITTER	SKYPE
315	Rik	Van Bruggen	rik@neotechnology.com	rik@vanbruggen.be	NULL	@rvanbruggen	rvanbruggen

Email: rik@neotechnology.com

Email: rik@vanbruggen.be

Twitter: @rvanbruggen

Skype: rvanbruggen

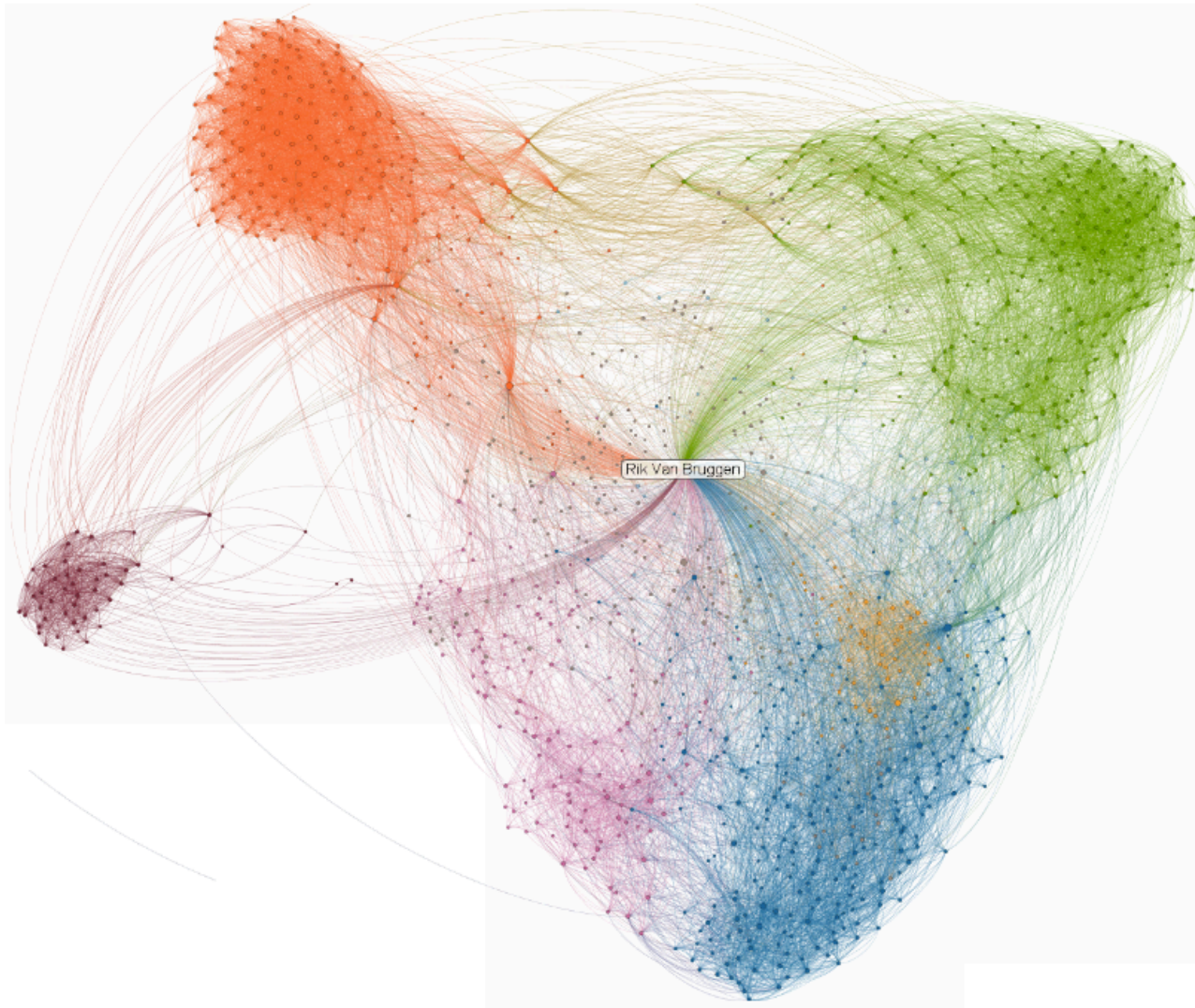


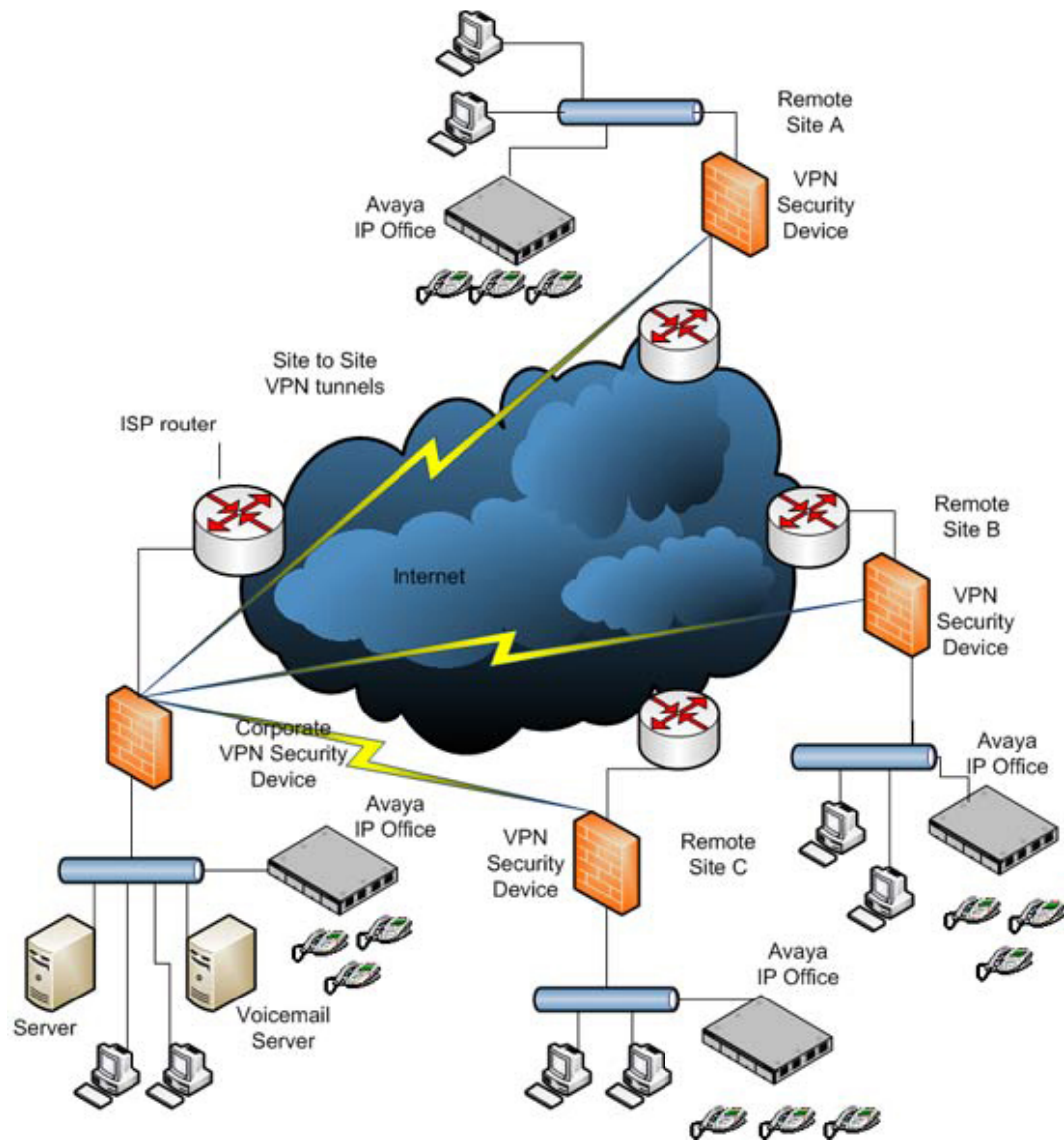


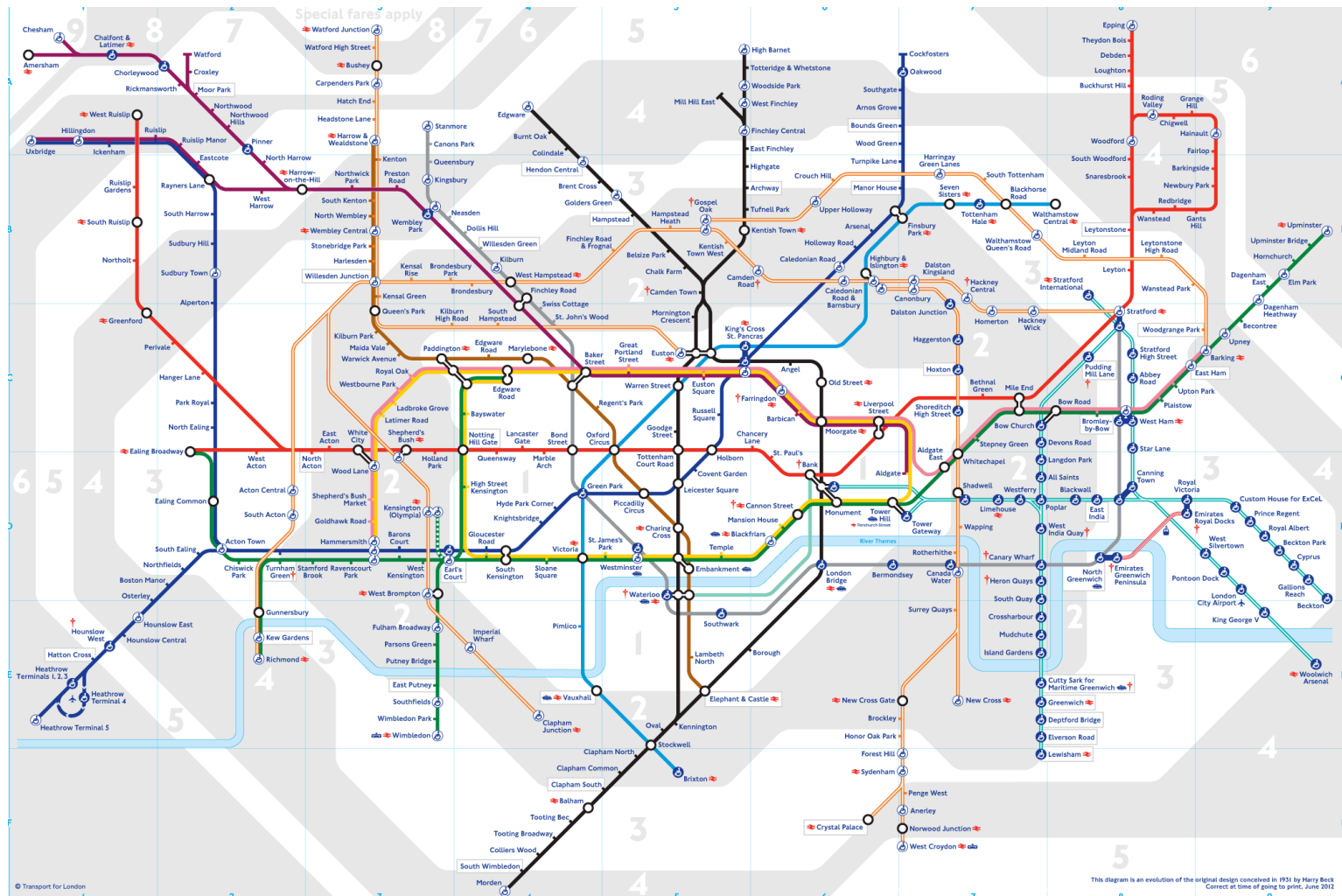
# The Real Complexity

*complexity = f(size, semi-structure, **connectedness**)*

# Examples of Connectedness







## Frequently Bought Together



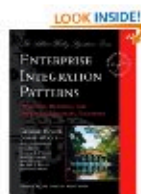
Price For All Three: **£69.50**

 Add all three to Basket

[Show availability and delivery details](#)

- ✓ **This item:** Patterns of Enterprise Application Architecture (The Addison-Wesley Signature Series) by Martin Fowler Hardcover **£24.00**
- ✓ Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions (Addison-Wesley Signature Series (Fowler) Addison-Wesley Sign) by Gregor Hohpe Hardcover **£22.00**
- ✓ Domain-driven Design: Tackling Complexity in the Heart of Software by Eric Evans Hardcover **£23.50**

## Customers Who Bought This Item Also Bought



Enterprise Integration  
Patterns: Designing, ...

Gregor Hohpe

★★★★★ (16)

Hardcover

£22.00



Service Design Patterns:  
Fundamental Design ...

Robert Daigneau

★★★★★ (2)

Hardcover

£19.97



Design patterns : elements of  
reusable ...

Erich Gamma

★★★★★ (51)

Hardcover

£21.00



Domain-driven Design:  
Tackling Complexity in ...

Eric Evans

★★★★☆ (13)

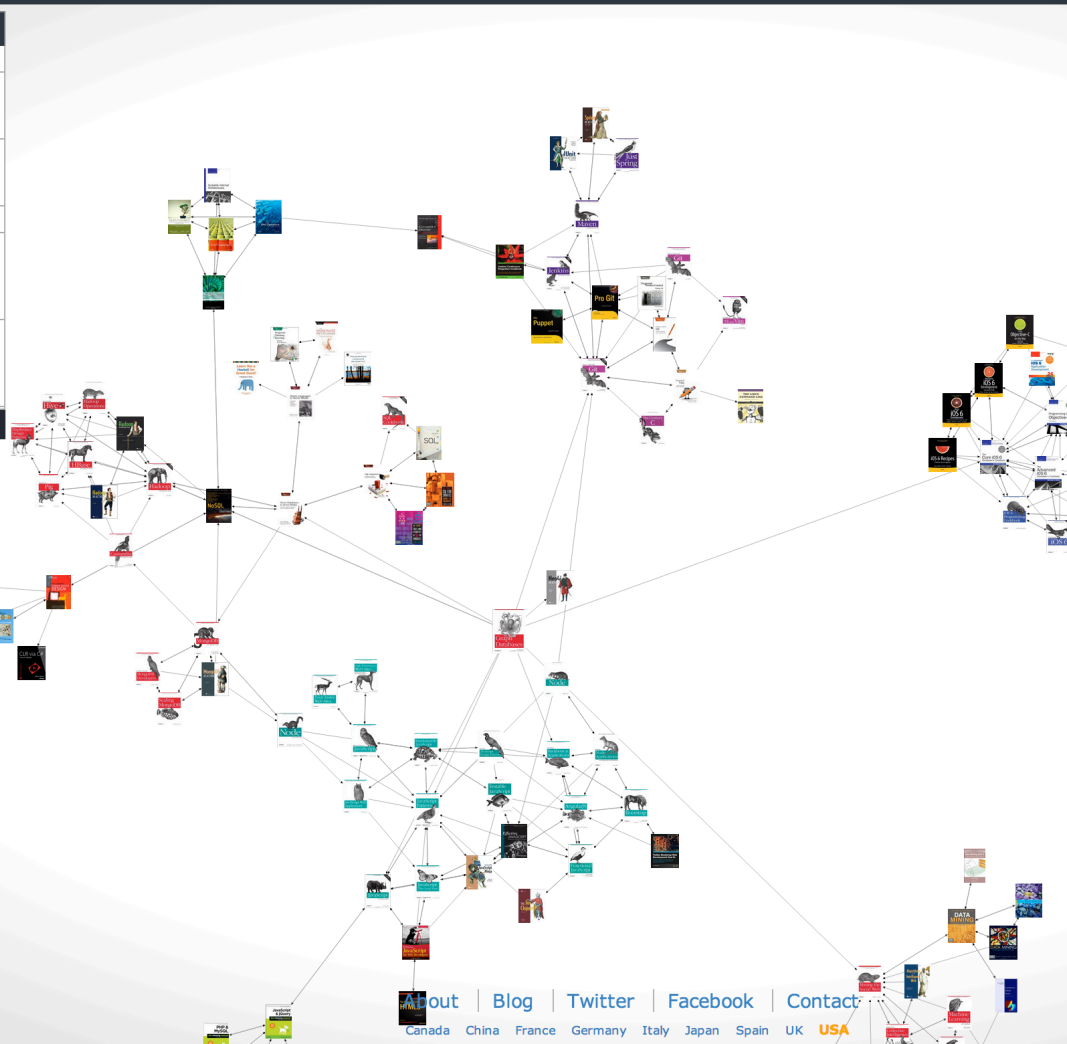
Hardcover

£23.50

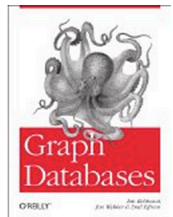


Found Products (115)	Hide
JavaScript Patterns	\$19.99
The Core iOS 6 Developer's Cookbook (4th Edition) (Developer's Library)	\$25.35
Programming in Objective-C (5th Edition) (Developer's Library)	\$29.47
Hadoop: The Definitive Guide	\$27.89
Mining the Social Web: Analyzing Data from Facebook, Twitter, LinkedIn, and Other Social Media Sites	\$26.88
Version Control with Git: Powerful tools and techniques for collaborative software development	\$21.33

Sort by: Popularity



### Graph Databases



Price: \$24.11  
 By: Emil Eifrem  
 Published: 2013-06-17  
 Pages: 224



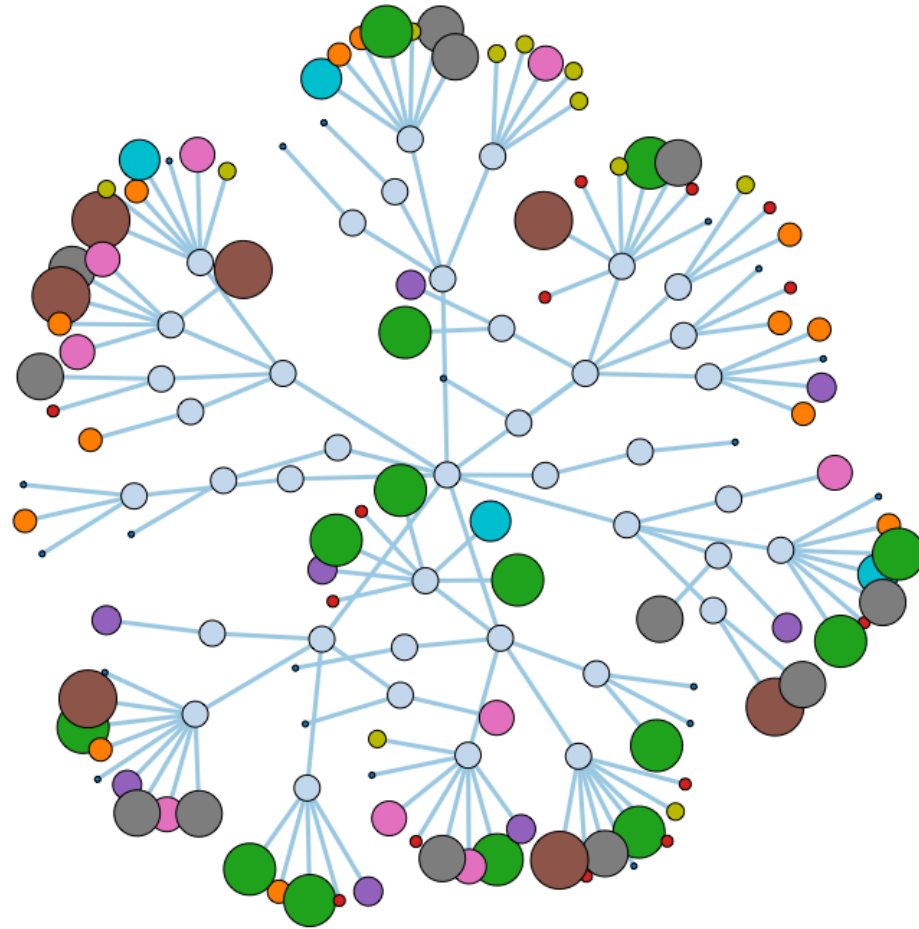
[Product Description](#)   [Customer Reviews](#)

Discover how graph databases can help you manage and query highly connected data. With this practical book, you'll learn how to design and implement a graph database that brings the power of graphs to bear on a broad range of problem domains. Whether you want to speed up your response to user queries or build a database that can adapt as your business evolves, this book shows you how to apply the schema-free graph model to real-world problems.

Learn how different organizations are using graph databases to outperform their competitors. With this book's data modeling, query, and code examples, you'll quickly be able to implement your own solution.

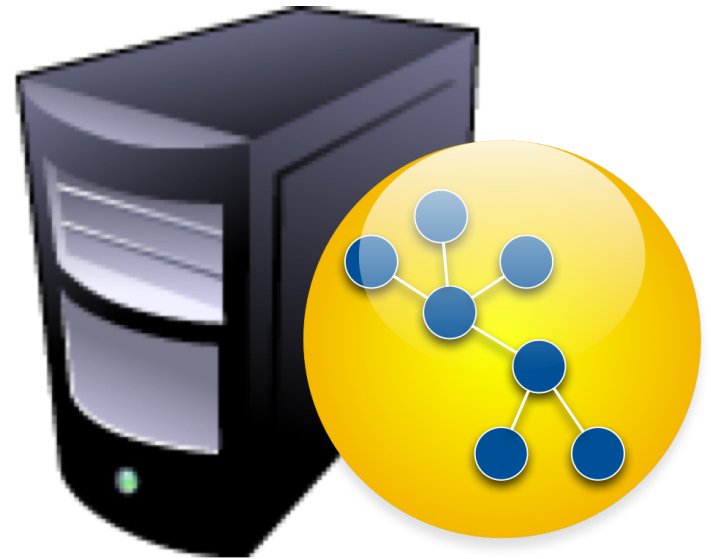
- Model data with the Cypher query language and property graph model
- Learn best practices and common pitfalls when modeling with graphs
- Plan and implement a graph database solution in test-driven fashion
- Explore real-world examples to learn how and why organizations use a graph database
- Understand common patterns and components of graph database architecture
- Use analytical techniques and algorithms to mine graph database information

# Graphs Are Everywhere



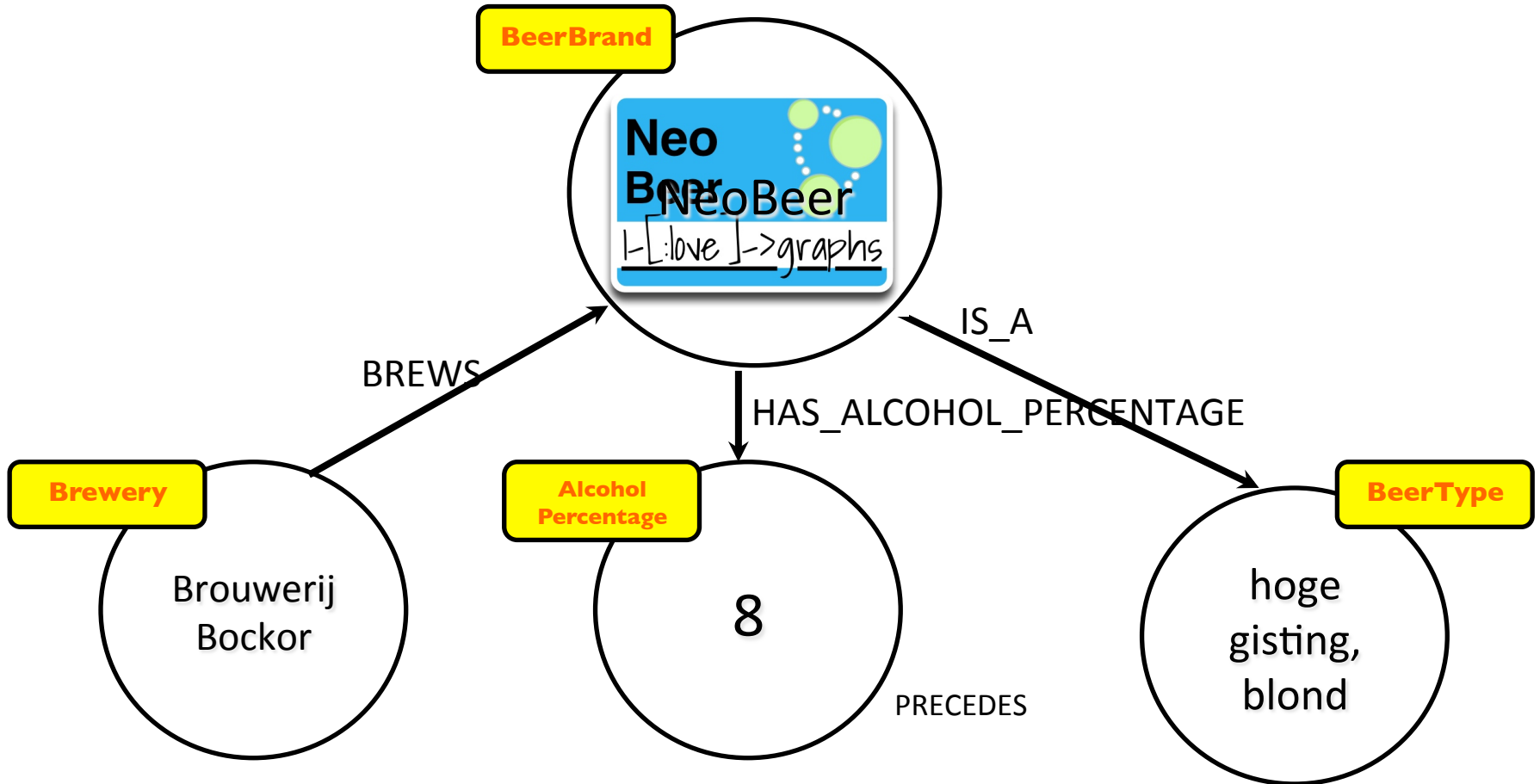
# Neo4j is a Graph Database

- JVM based
- ACID transactions
- Query language
- Rich Java APIs
- Using the Property Graph model

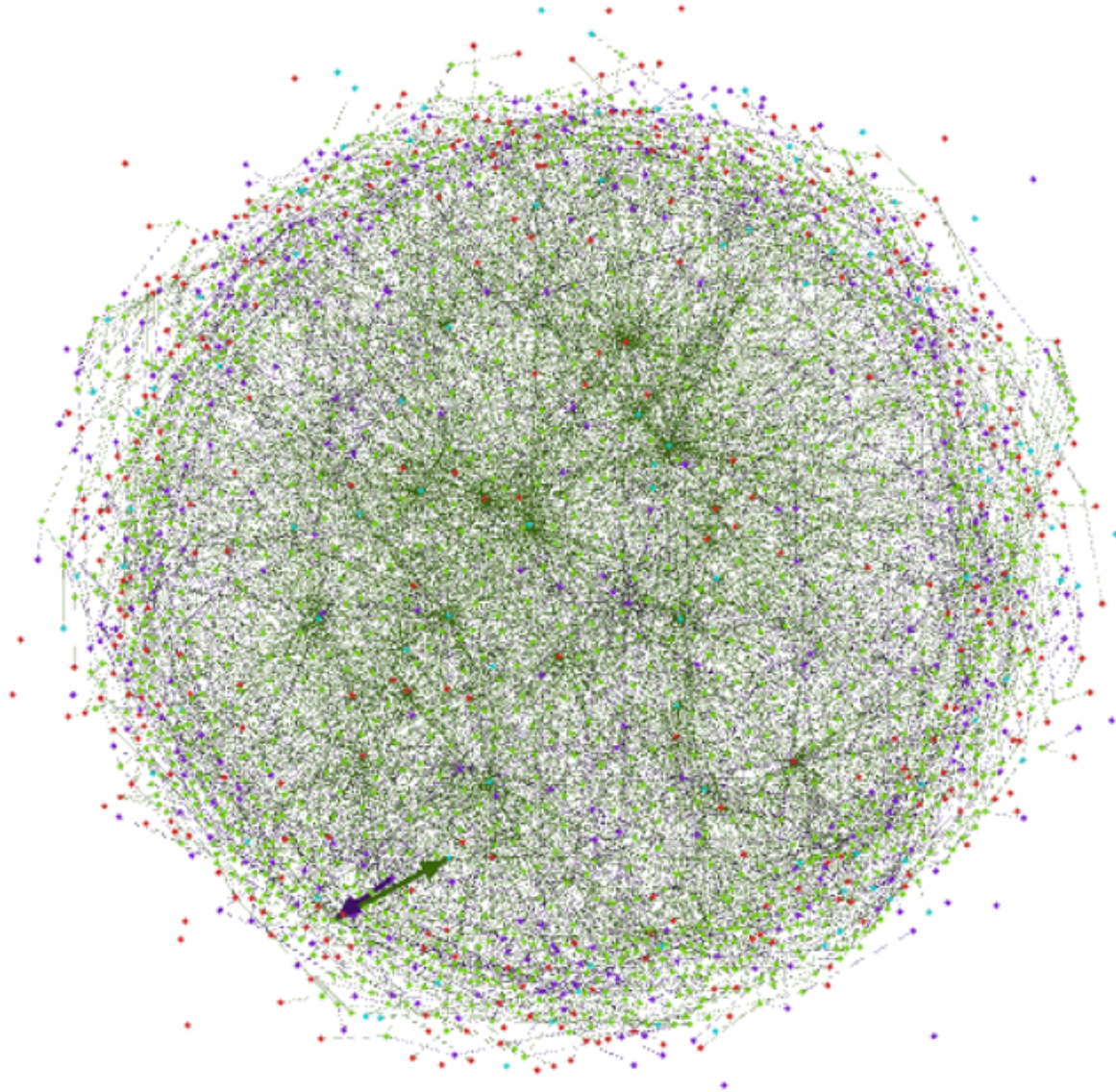




# Describing Graphs

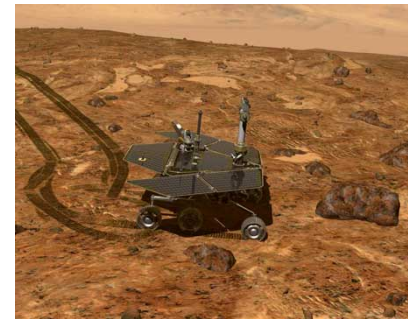
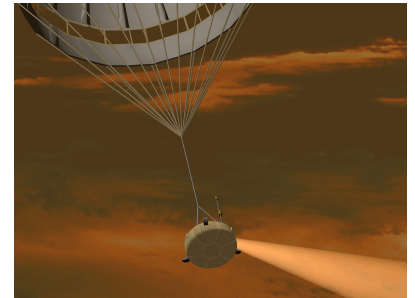


# A Hairball!

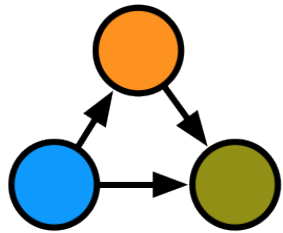


# Querying a Graph

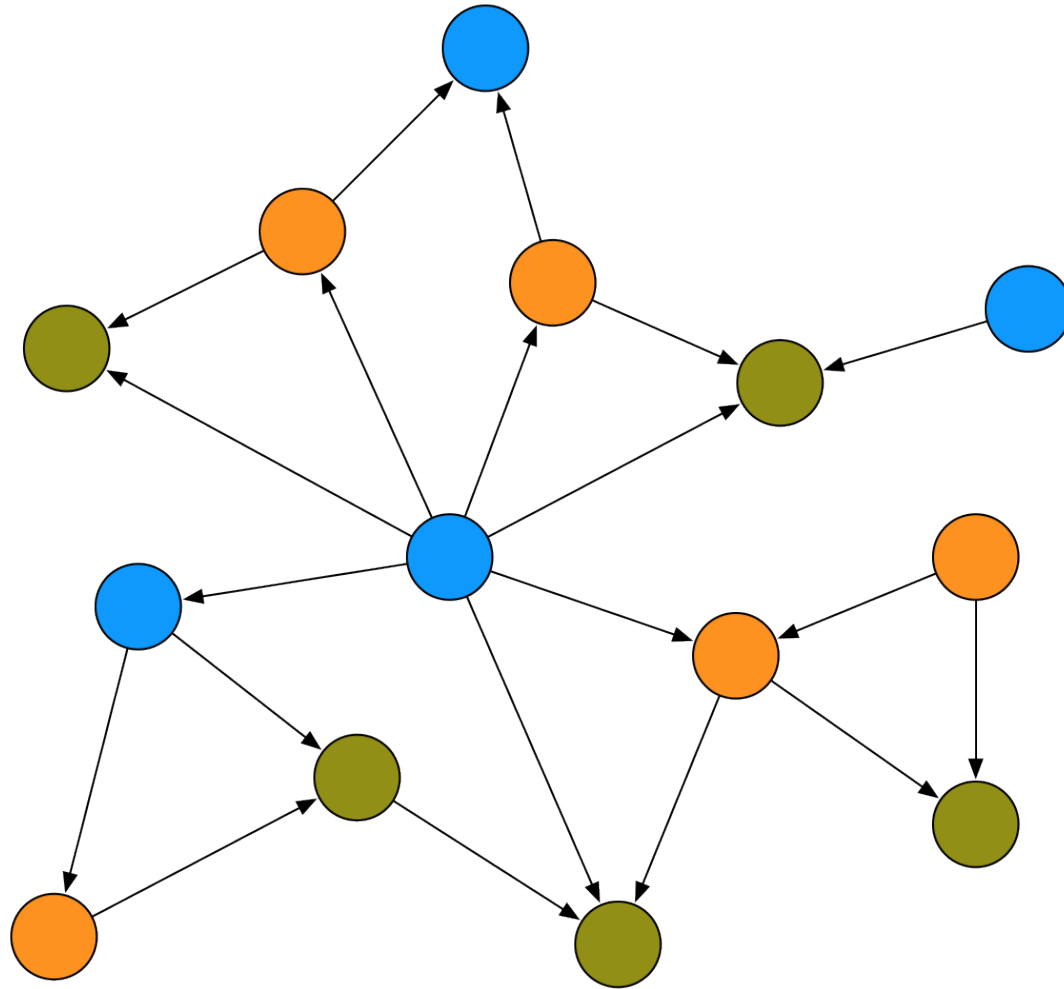
- Graph local
  - Contextualized “ego-centric” queries
- “Parachute” into graph
  - Start node(s)
    - Found through Index lookups
- Crawl the surrounding graph
  - 2 million+ joins per second
    - No more Index lookups:  
Index-free adjacency



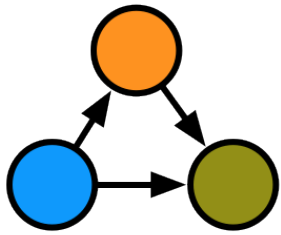
# Queries: Pattern Matching



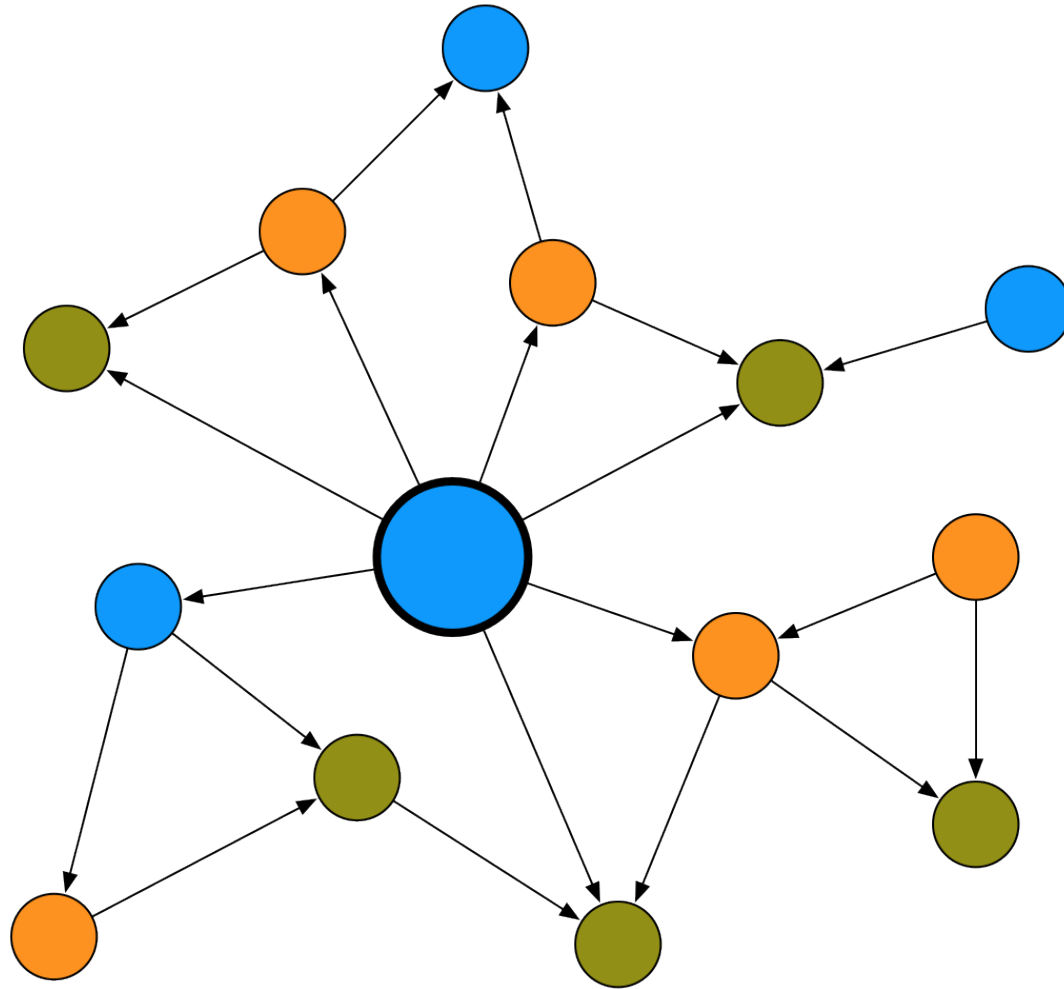
Pattern



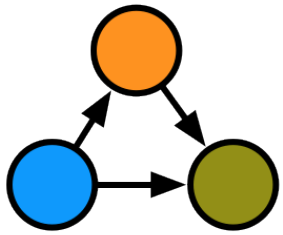
*Don't worry (as much) about the*  
**Start Node**



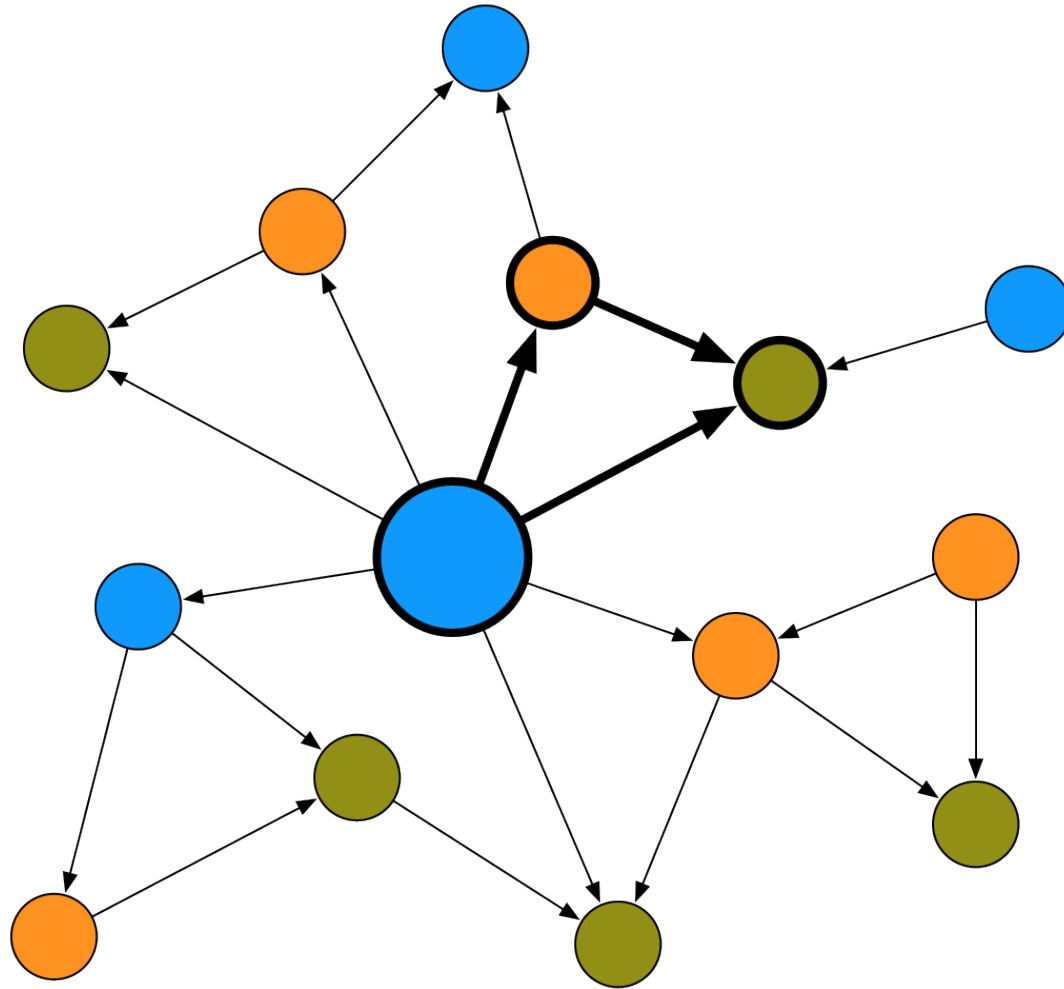
Pattern



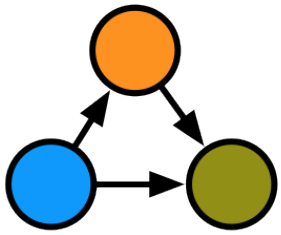
# Match



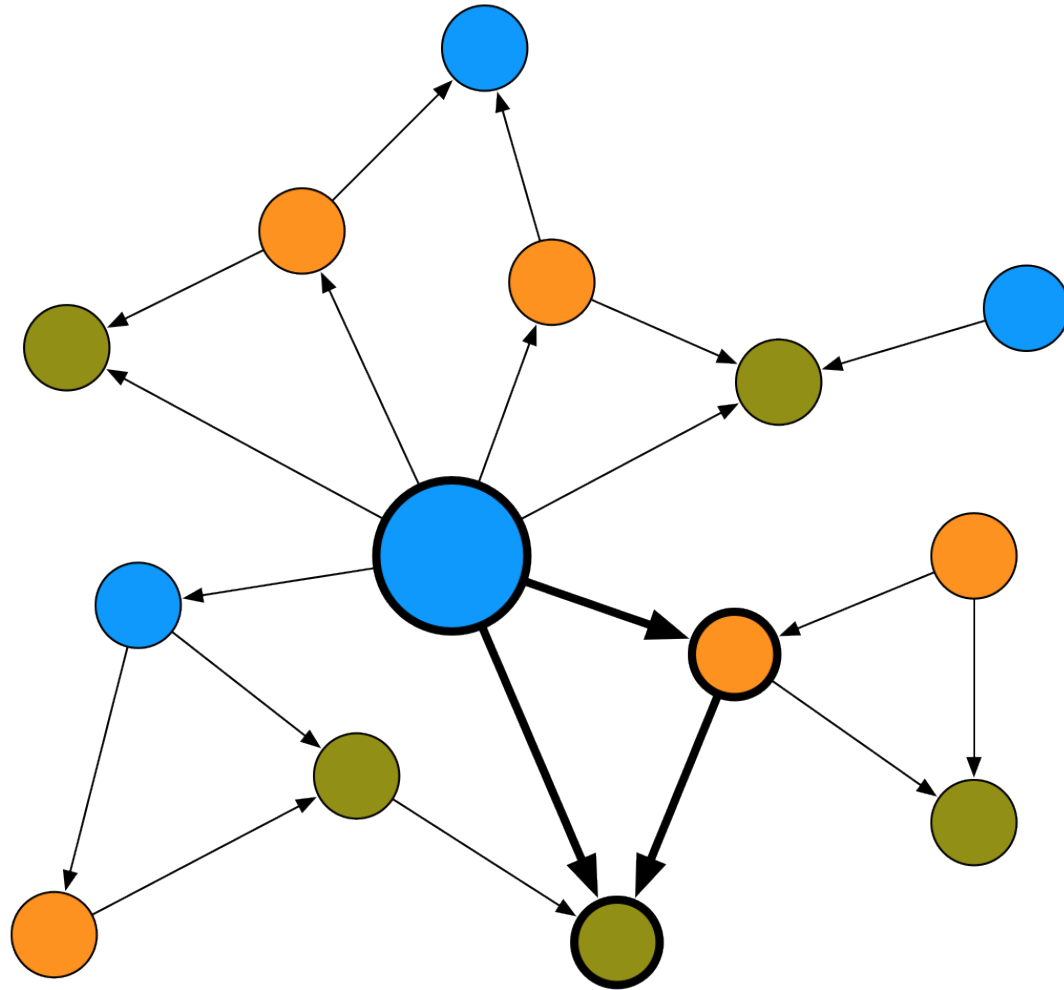
Pattern



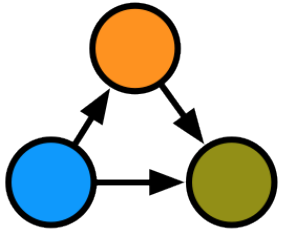
# Match



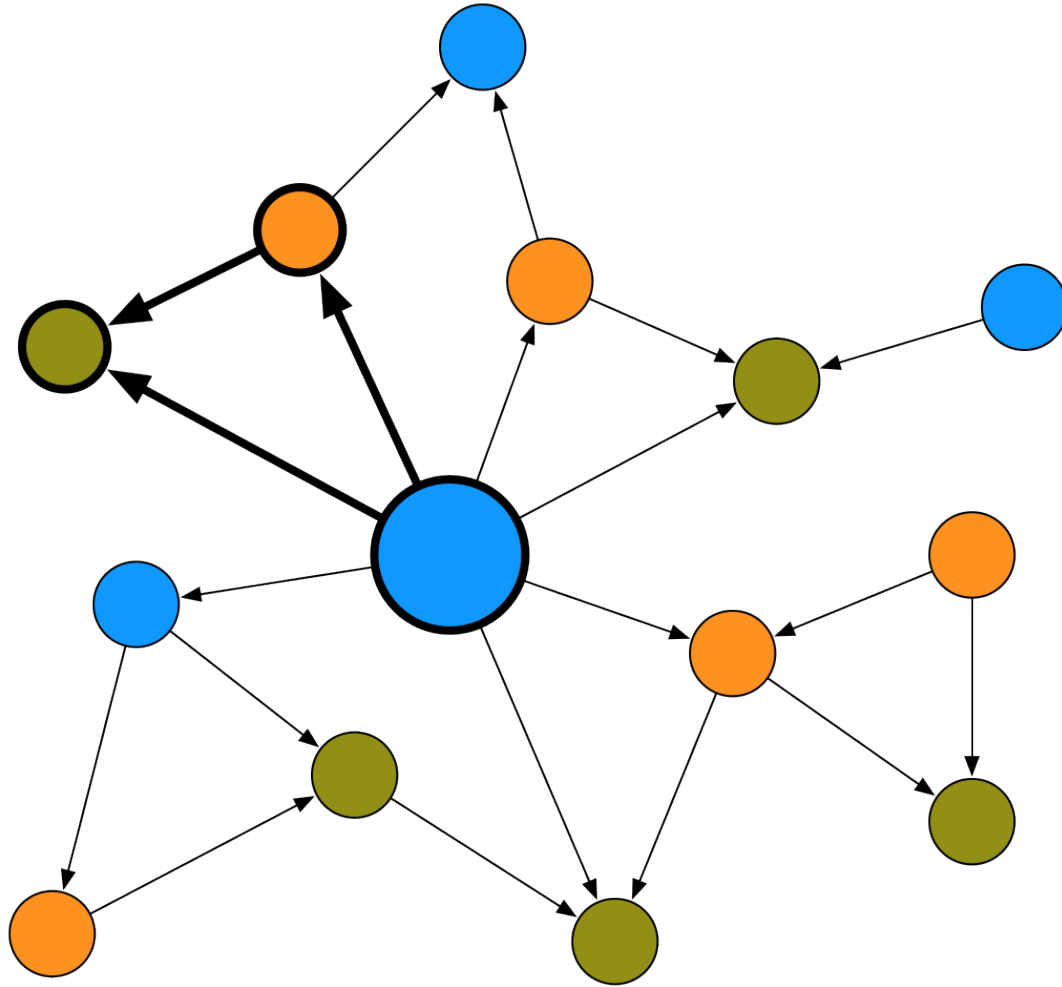
Pattern



# Match

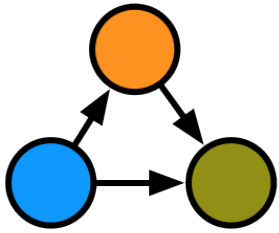


Pattern

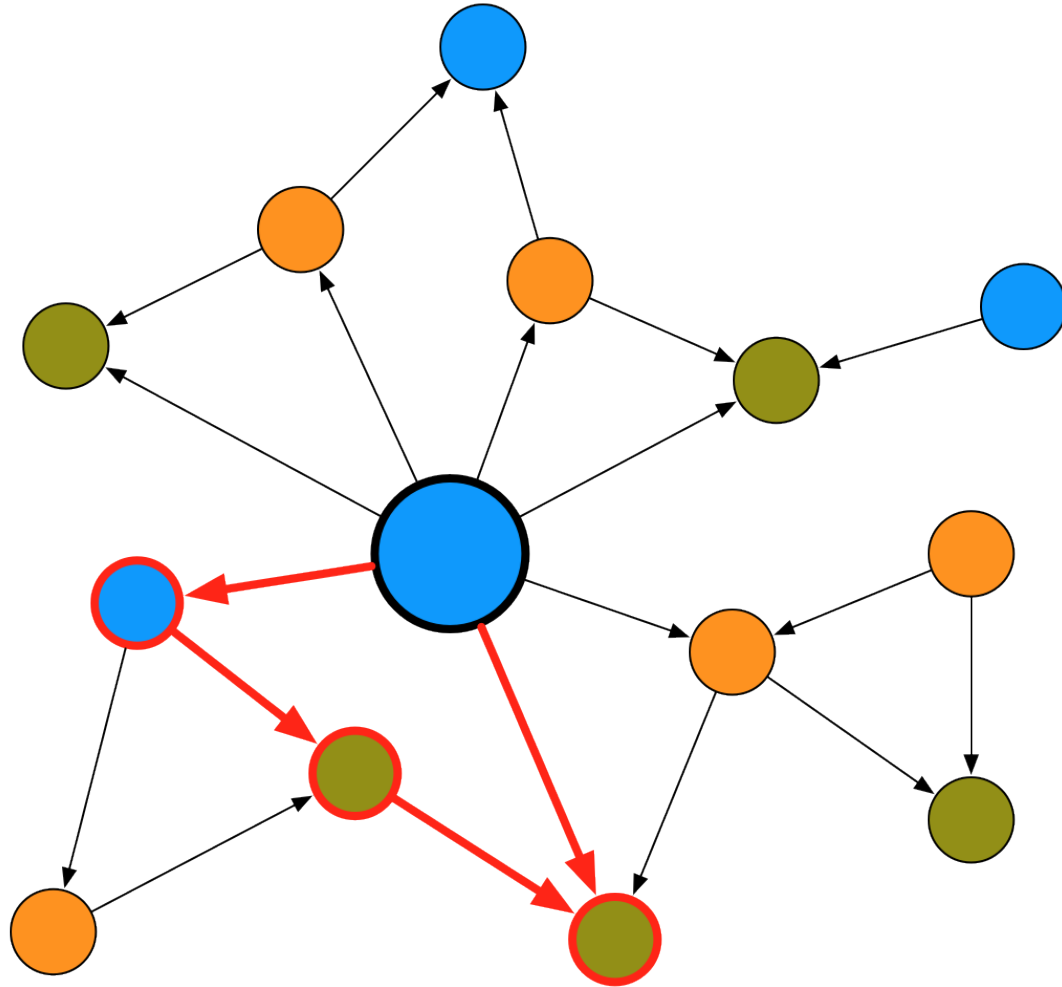




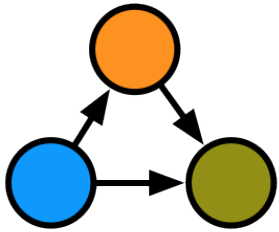
# Non-Match



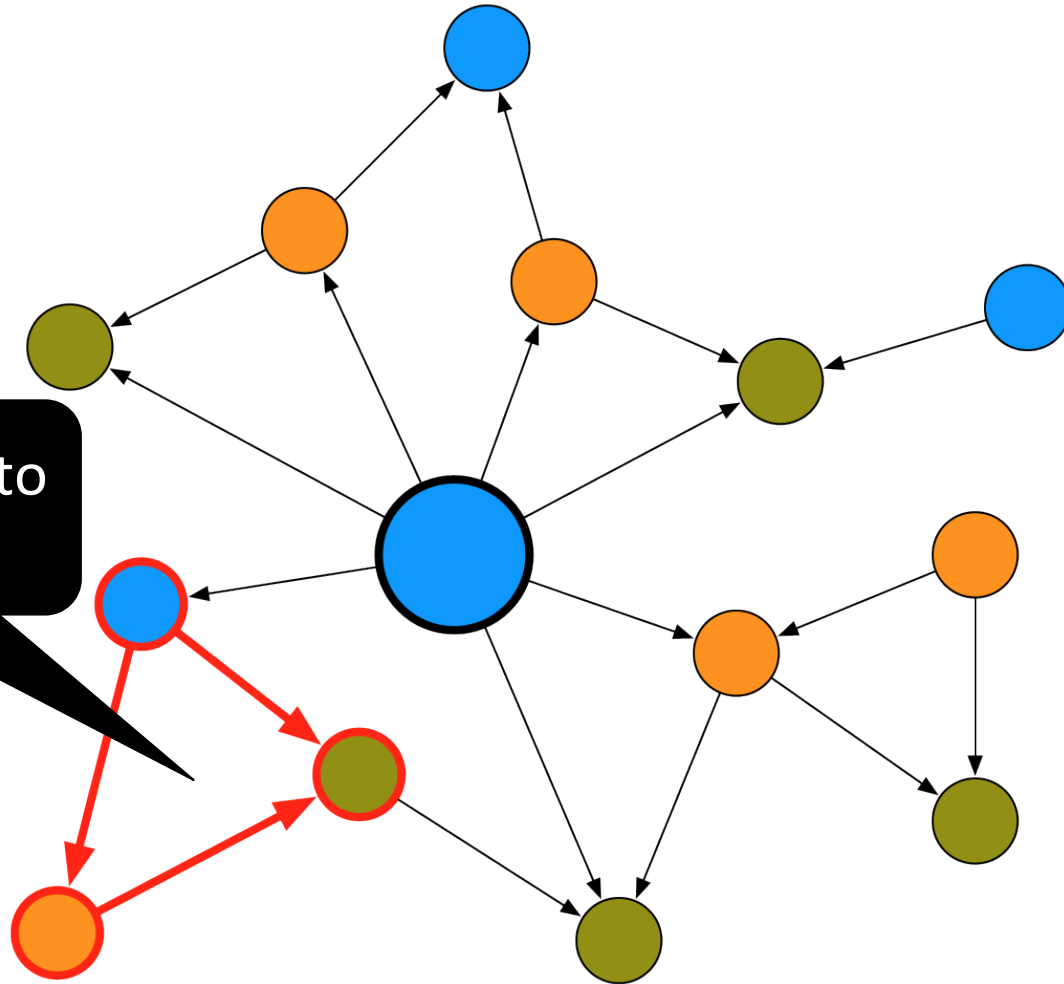
Pattern



# Non-Match

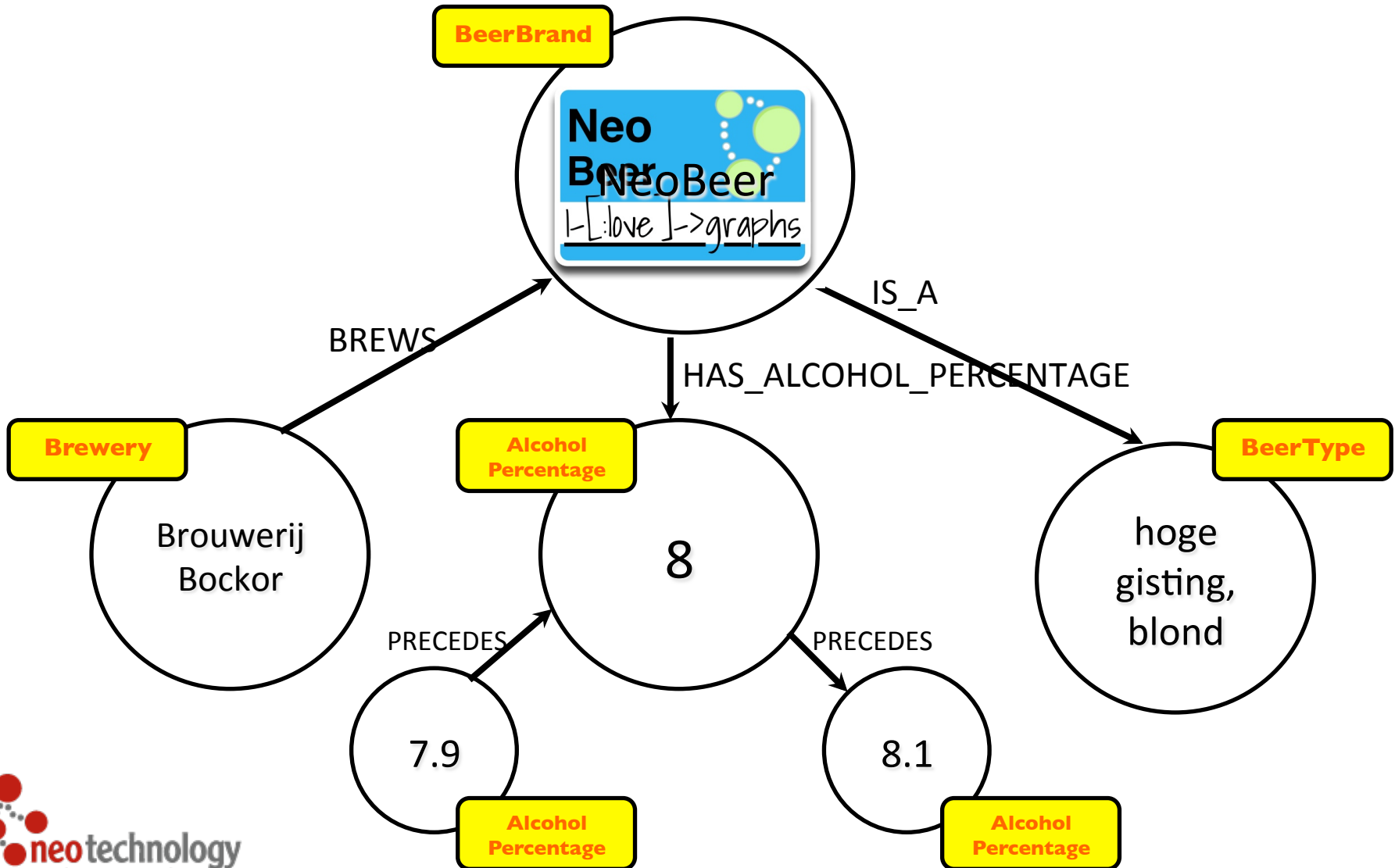


Pattern

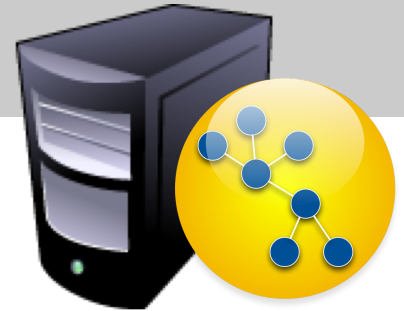


Not anchored to start node

# Describing Graphs



# Graph Visualisations



- Web Admin
- Neography / Neovigator
- Neoclipse
- Gephi, Prefuse, d3.js, etc....

# Neo4j Web Admin

Neo4j

Overview Dashboard | Explore and edit Data browser | Power tool Console | Add and remove Indexes | Details Server info | Documentation

rels:17

Returned 1 row. Query took 59ms

Style | Re-layout | Clear

```
graph TD; A("Abdij Notre-Dame d'Orval") -- owns --> B("Orval"); B -- isa --> C("trappist"); B -- HasAlcoholPercentage --> D("6.20"); C -- isa --> E("Westvleteren Zes"); D -- HasAlcoholPercentage --> F("10 nodes"); E -- HasAlcoholPercentage --> F;
```

The graph visualization shows a network of nodes and relationships. The largest node is "Abdij Notre-Dame d'Orval". It is connected to "Orval" via a relationship labeled "owns". "Orval" is connected to "trappist" via a relationship labeled "isa". "Orval" is also connected to a node labeled "6.20" via a relationship labeled "HasAlcoholPercentage". "trappist" is connected to "Westvleteren Zes" via a relationship labeled "isa". The node "6.20" is connected to a node labeled "10 nodes" via a relationship labeled "HasAlcoholPercentage". "Westvleteren Zes" is also connected to "10 nodes" via a relationship labeled "HasAlcoholPercentage".

# Neoclipse

The screenshot displays the Neoclipse Neo4j Graph view interface. The main window shows a graph with four nodes and three relationships:

- Node 10014:** name: trappist, type: BeerType
- Node 17:** name: Orval, type: BeerBrand
- Node 100012:** name: 6.20, type: AlcoholPercentage
- Node 50005:** name: Abdij Notre-Dame d'Orval, type: Brewery

Relationships:

- isa:** Connects Node 17 to Node 10014.
- HasAlcoholPercentage:** Connects Node 17 to Node 100012.
- Brews:** Connects Node 50005 to Node 17.

The interface includes a 'Connections' panel on the left, a 'Cypher Editor' at the top, a 'Help' panel on the right, and two data tables at the bottom.

**Properties Table:**

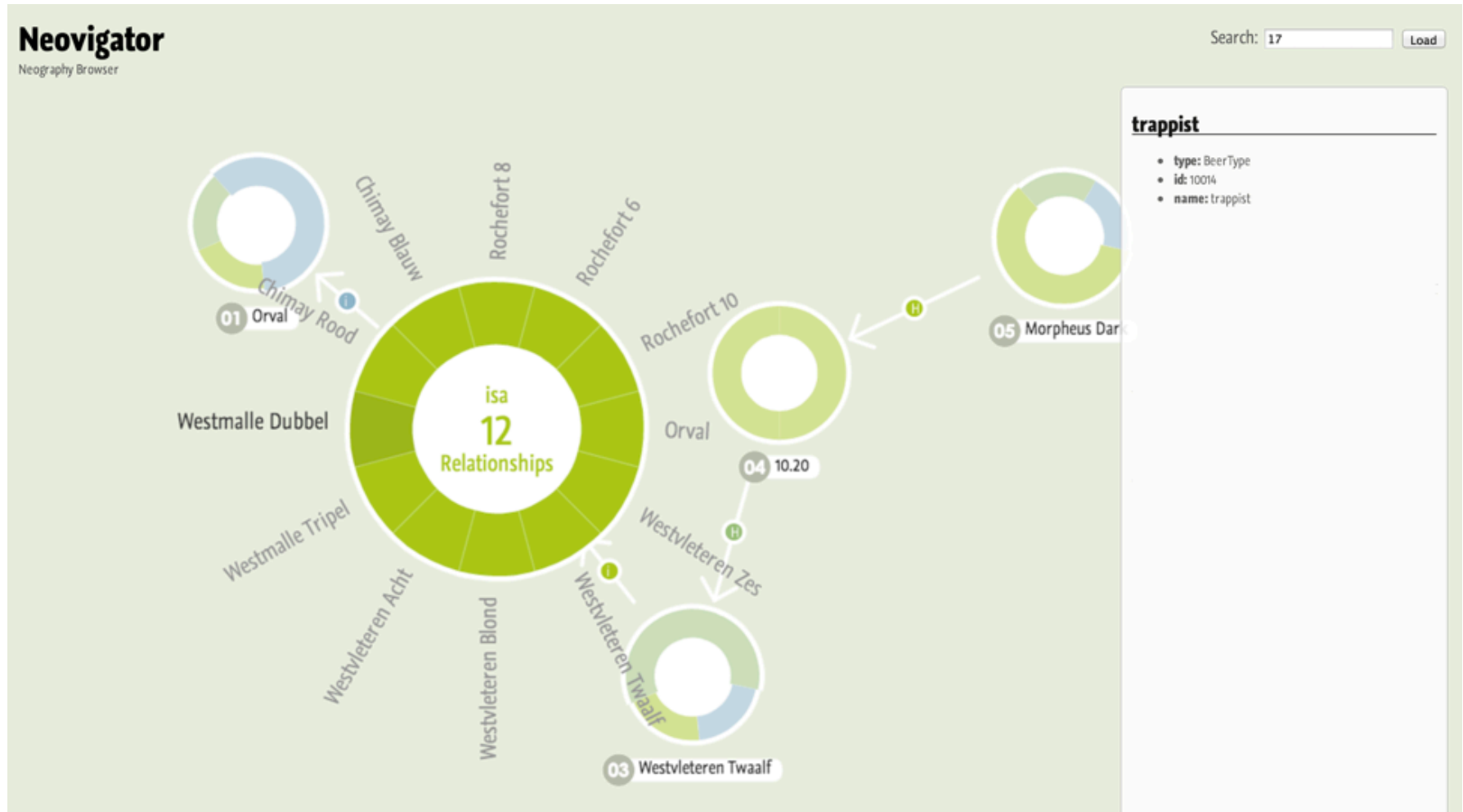
Property	Value
Node	
Id	17
Properties	
id	17
name	Orval
type	BeerBrand

**Relationship types Table:**

Relationship type	In	Out
Brews	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
HasAlcoholPercentage	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
isa	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

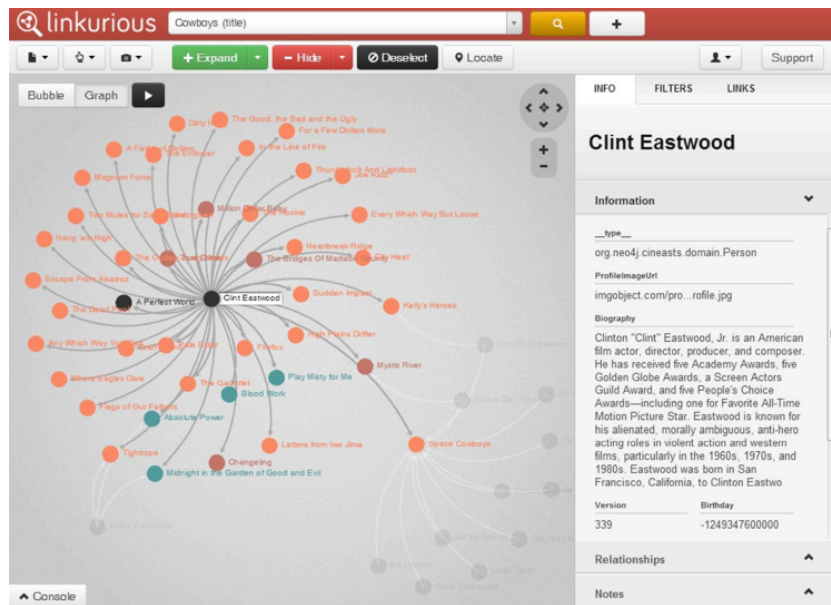
Using Neo4j Graph view - /org\_neo4j\_neoclipse.doc/html/tasks/graphview.html

# Neovigator

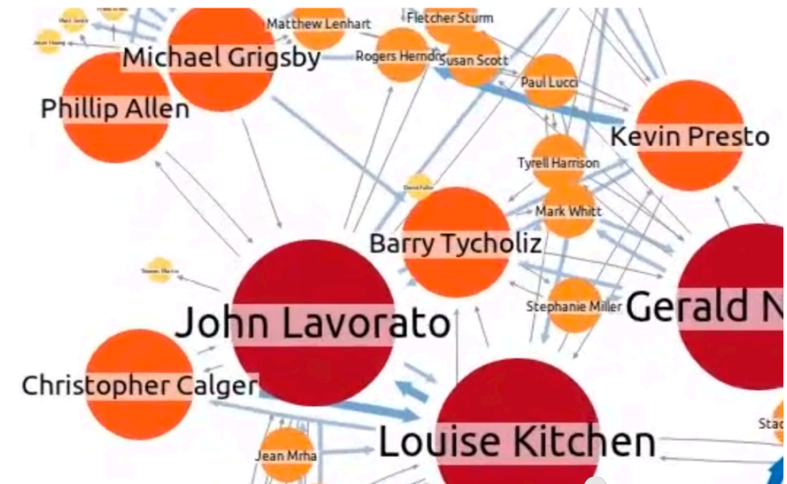


# Other Visualisation options:

- Linkurio.us



- KeyLines





# Case Studies

Core Industries & Use Cases:	Software	Financial Services	Telecommunications	Health Care & Life Sciences	Web Social, HR & Recruiting	Media & Publishing	Energy, Services, Automotive, Gov't, Logistics, Education, Gaming, Other
Network & Data Center Management	Zenöss, Junisphere, NetApp, SERENA gen, VIRTUAL INSTRUMENTS		hp, SFR				
MDM / System of Record			CISCO, Deutsche Telekom, maaii	ZEPHYR HEALTH INC, HealthUnlocked	Wooz, EQUILAR, viadeo, glassdoor	LIFECHURCH.TV, SQUIDOO	Juice PLUS+, onefinestay, teachscape
Social	Glowbl, ICE			SharePractice			LAUREATE INTERNATIONAL UNIVERSITIES
Geo	DingLicom		Justdial		classmates.com	indiatimes	gamesy, LAUREATE INTERNATIONAL UNIVERSITIES
Recommendations	AXON ACTIVE, kitedesk				Dshini, careerbuilder, InfoJobs, moviepilot	Perigee, CHIP, zeebox	shurtl, research now, compete
Identity & Access Mgmt	LIQUID COMMON, aikux.com, entropy	Global500 Finance	telenor			LifeWay	
Content Management	springcm, Adobe			SevenBridges genomics	SRM SOCIETY FOR HUMAN RESOURCE MANAGEMENT, hinge	decibel, <fuseworks/>	DOSB NEW MEDIA GMBH, DEUTSCHER OLYMPISCHER SPORT BUND
BI, CRM, Impact Analysis, Fraud Detection, Resource Optimization, etc.	SODIFRANCE, idMISSION, Humaninvest.co	DRW TRADING GROUP, NexLP	Global500 Telcommunication	janssen			DRAKER, Impact Technologies, LOCKHEE MARTIN, Global500 Energy, Global500 Aerospace

# Contact Us

- Neo Technology UK  
5-11 Lavington Street  
London  
SE1 0NZ  
ENGLAND  
+44 808 189 0493
- [rik@neotechnology.com](mailto:rik@neotechnology.com) or
- +32 478 686800

