

Bancos de Dados de Grafos

Jaudete Daltio

MC536

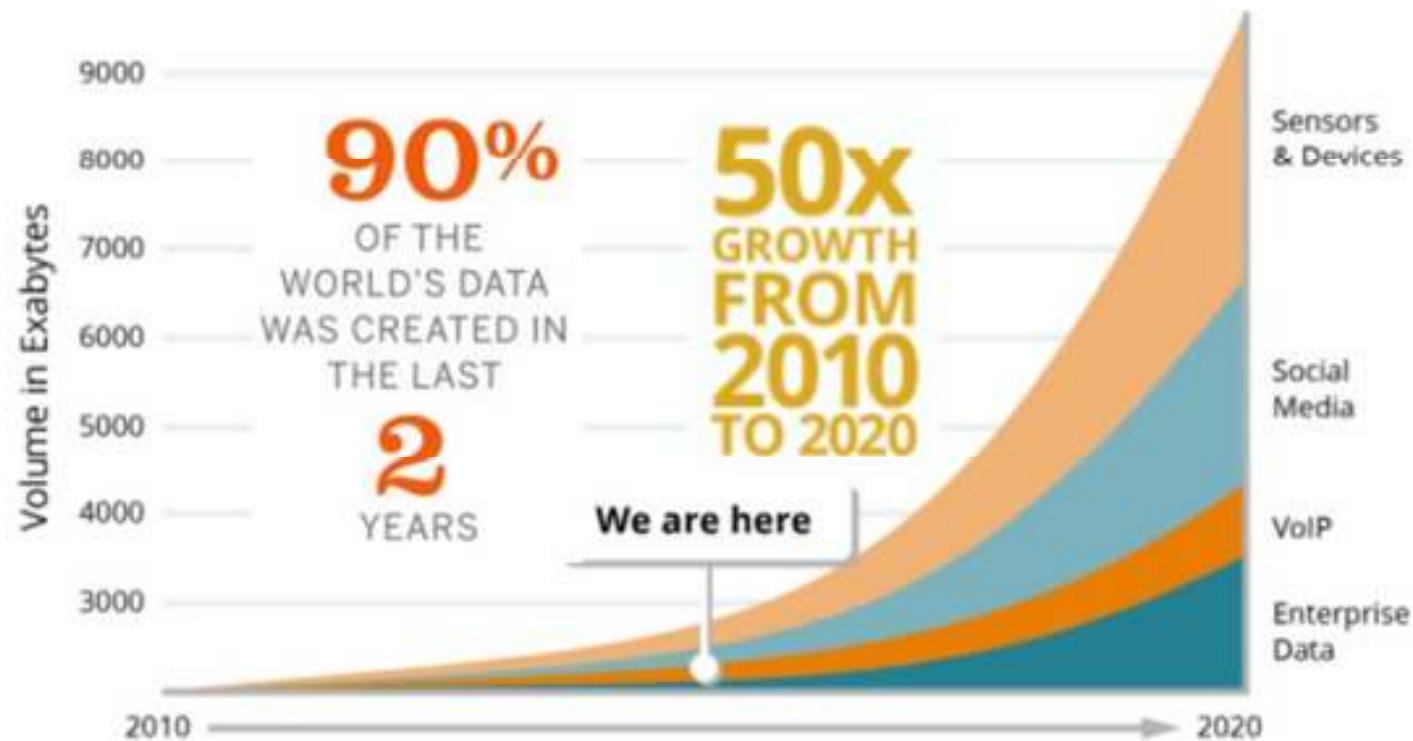
Motivações para novos SGBDs

- Volume de dados crescent
- Distribuídos, heterogêneos e interligados
- Questões sobre armazenamento e processamento descentralizado, desempenho e semântica

Tendência (1): volume

BIG IN GROWTH, TOO.

1 exabyte (EB) = 1,000,000,000,000,000 bytes

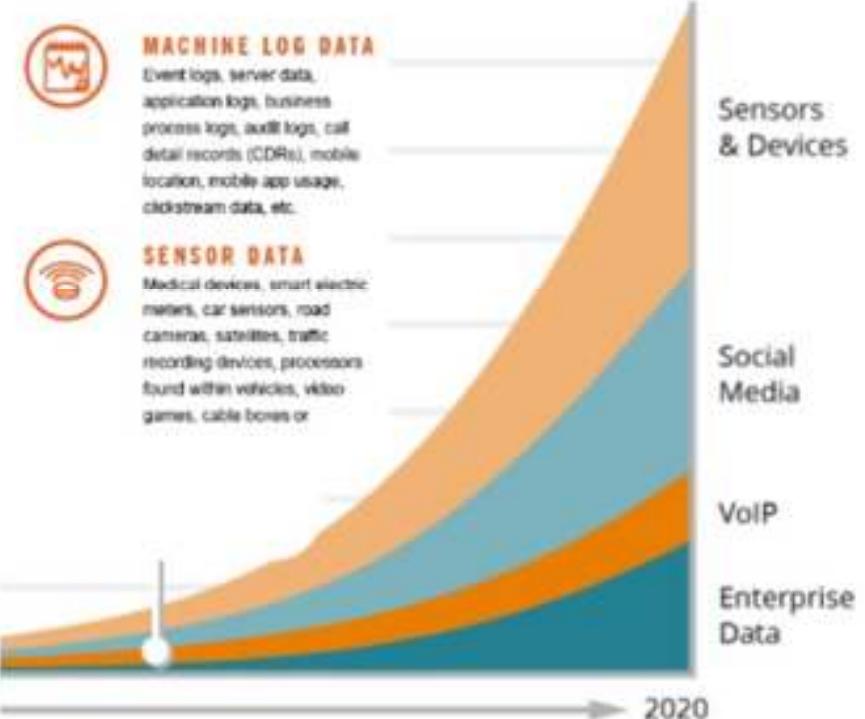


Tendência (1): volume

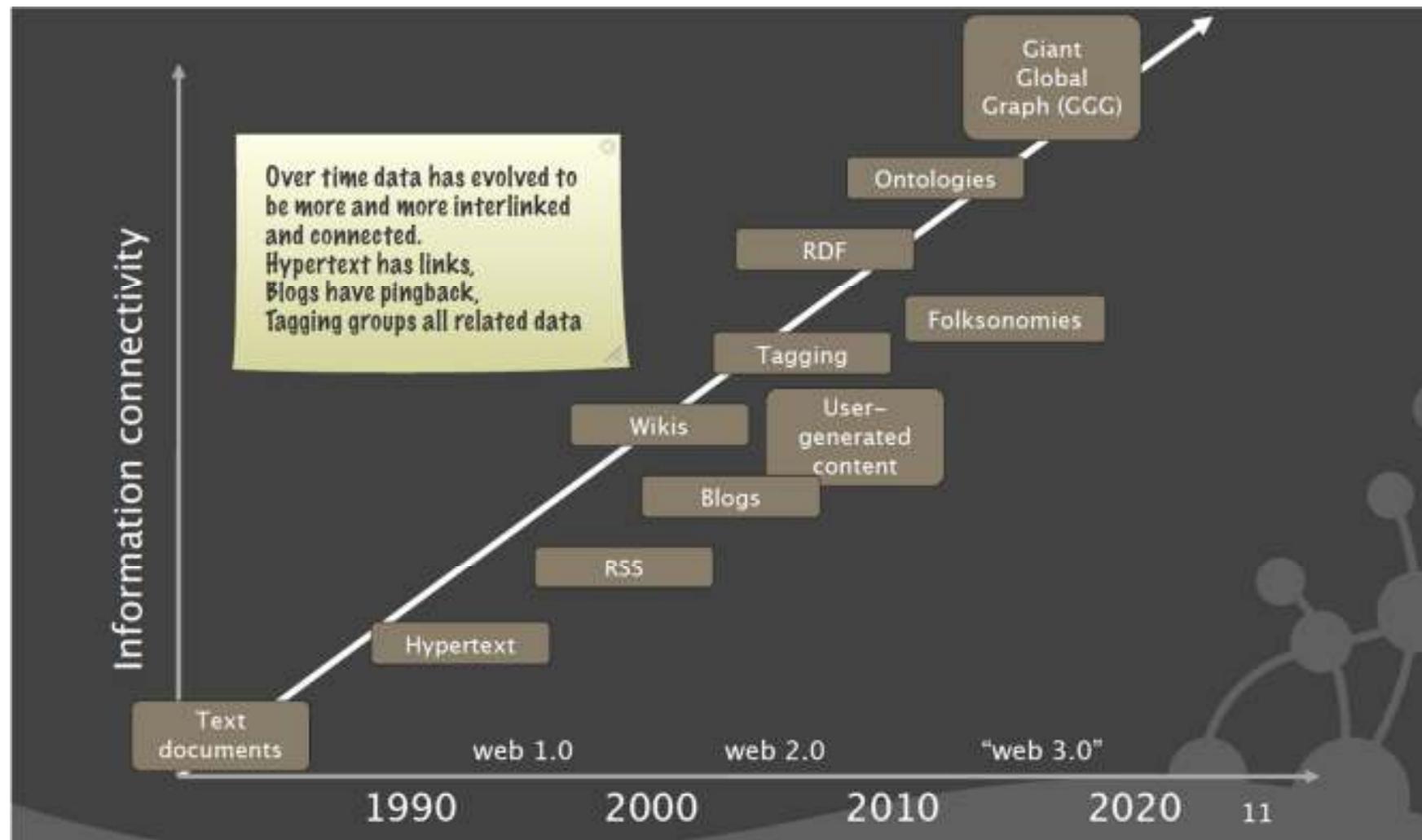
9 SOURCES

- ARCHIVES**
Archives of scanned documents, statements, insurance forms, medical record and customer correspondence, paper archives, and print stream files that capture original systems of record between organizations and their customers
- DOCS**
XLS, PDF, CSV, email, Word, PPT, HTML, HTM, S, plain text, XML, JSON, etc.
- MEDIA**
Images, videos, audio, Flash, live streams, podcasts, etc.
- DATA STORAGE**
SQL, NoSQL, Hadoop, doc repository, file system, etc.

- BUSINESS APPS**
Project management, marketing automation, productivity, CRM, ERP, content management systems, HR, storage, talent management, procurement, expense management, Google Docs, Intranets, portals, etc.
- PUBLIC WEB**
Government, weather, competitive, traffic, regulatory, compliance, health care services, economic, census, public finance, stock, OSINT, the World Bank, SEC Edgar, Wikipedia, IMDB, and other Web services.
- SOCIAL MEDIA**
Twitter, LinkedIn, Facebook, Tumblr, Blog, Storify, YouTube, Google+, Instagram, Flickr, Pinterest, Vimeo, WordPress, IM, RSS, Review, Chatter, Jive, Yammer, etc.



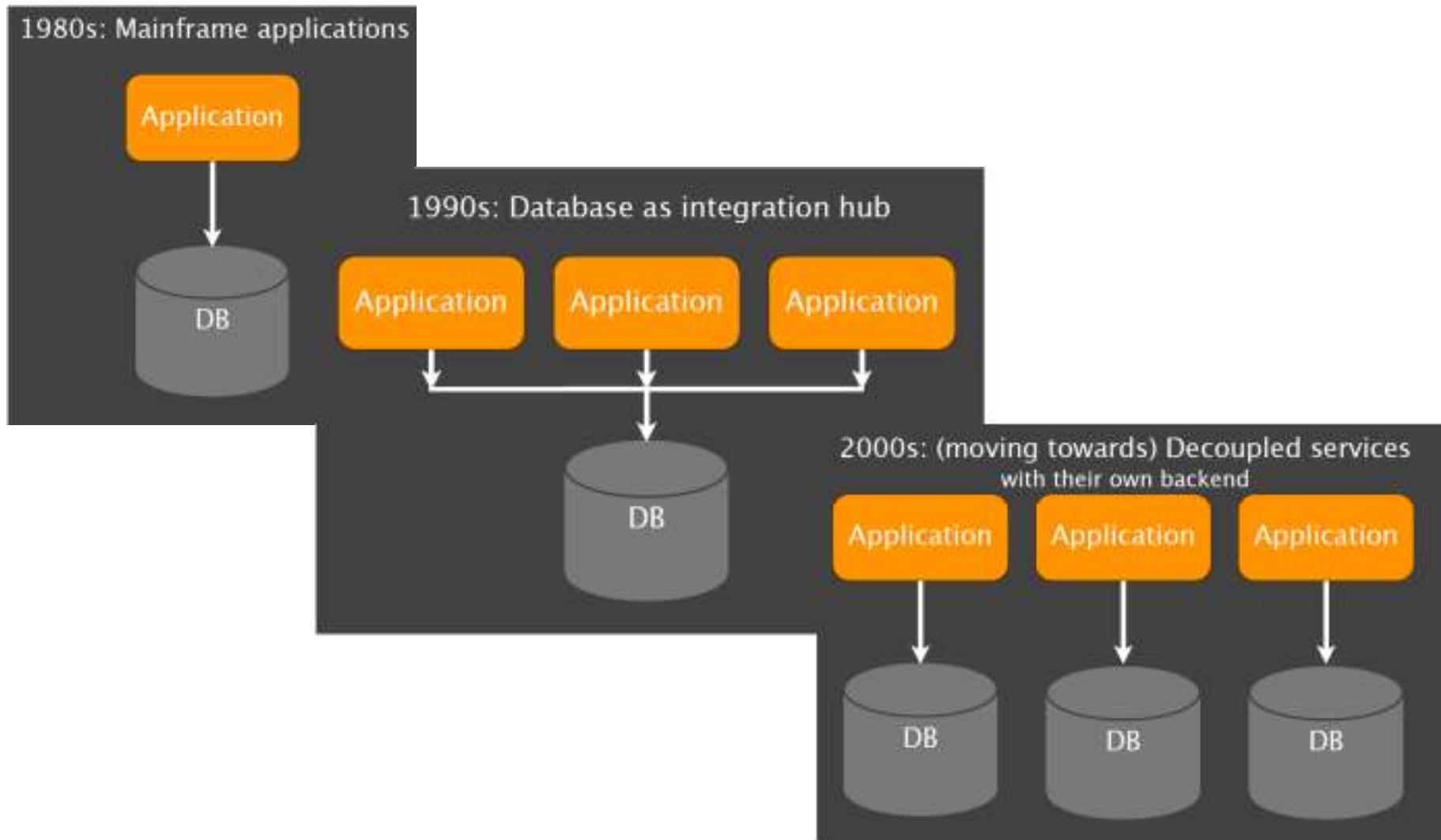
Tendência (2): conectividade



Tendência (3): heterogeneidade



Tendência (4): arquiteturas



NOSQL

No to SQL

Never SQL



Reconhecer que para alguns problemas
podem existir soluções de persistência
customizadas com **melhor desempenho**
OU mais amigaveis

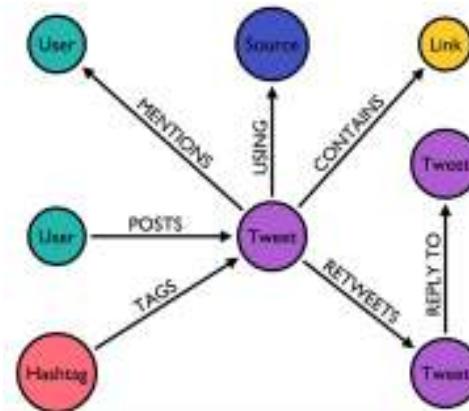
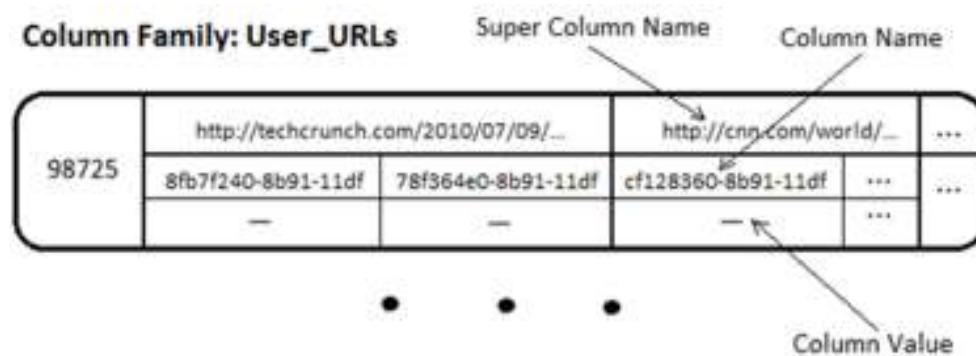
NOSQL “SGBDs”

- Não possuem todas as propriedades de um SGBD relacional (p ex, consistência)
- Noção de esquema é fuzzy
 - Em um mesmo BD, duas instâncias de uma mesma entidade podem ter atributos diferentes
- Diferentes modelos de dados

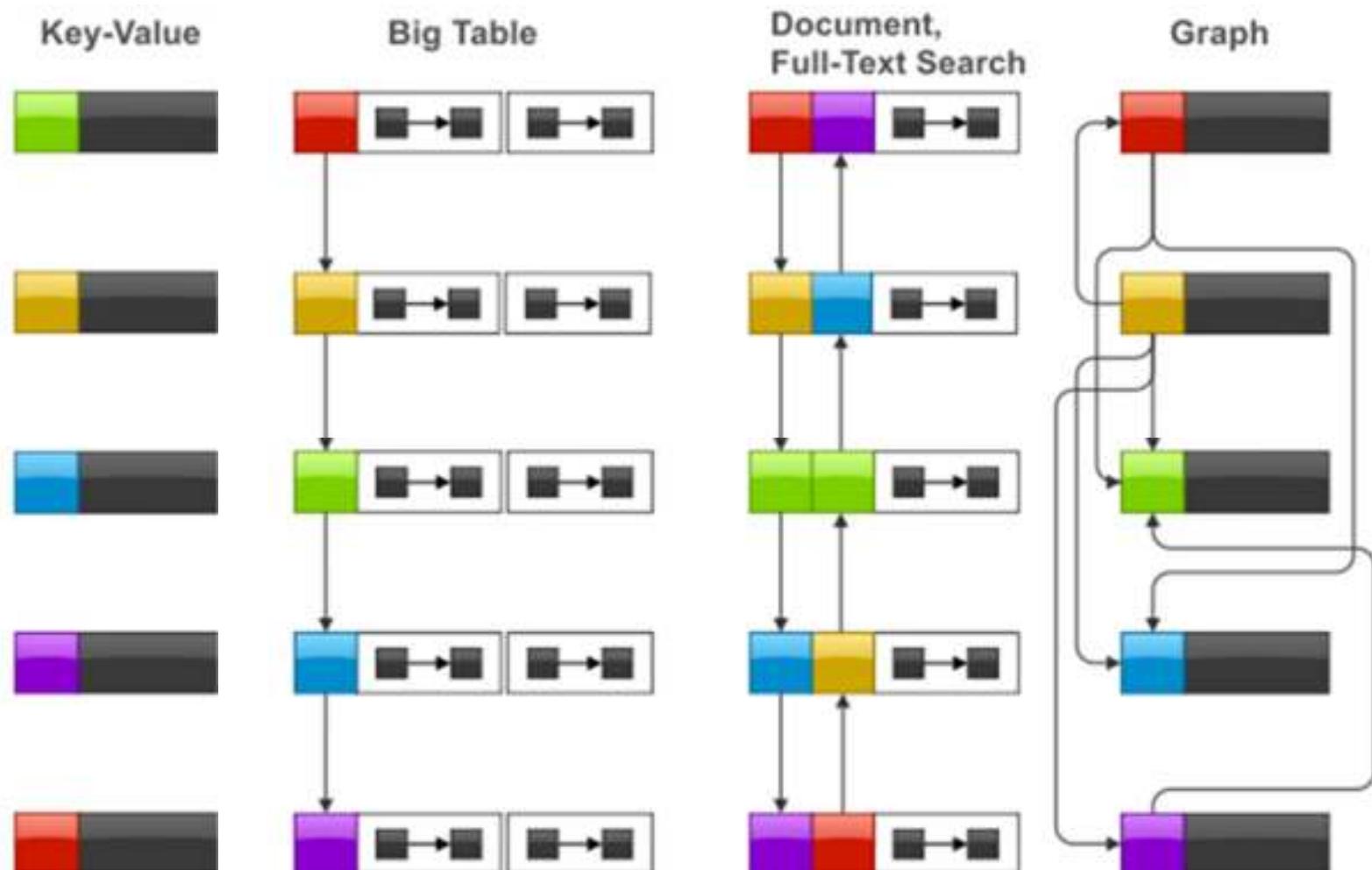
Modelo de Dados (Codd, 1980)

- Coleção de elementos para representar dados e expressar detalhes semânticos
- Componentes
 - Tipos de estruturas de dados
 - Restrições de integridade para definir estados consistentes do banco de dados
 - Operadores para recuperar dados

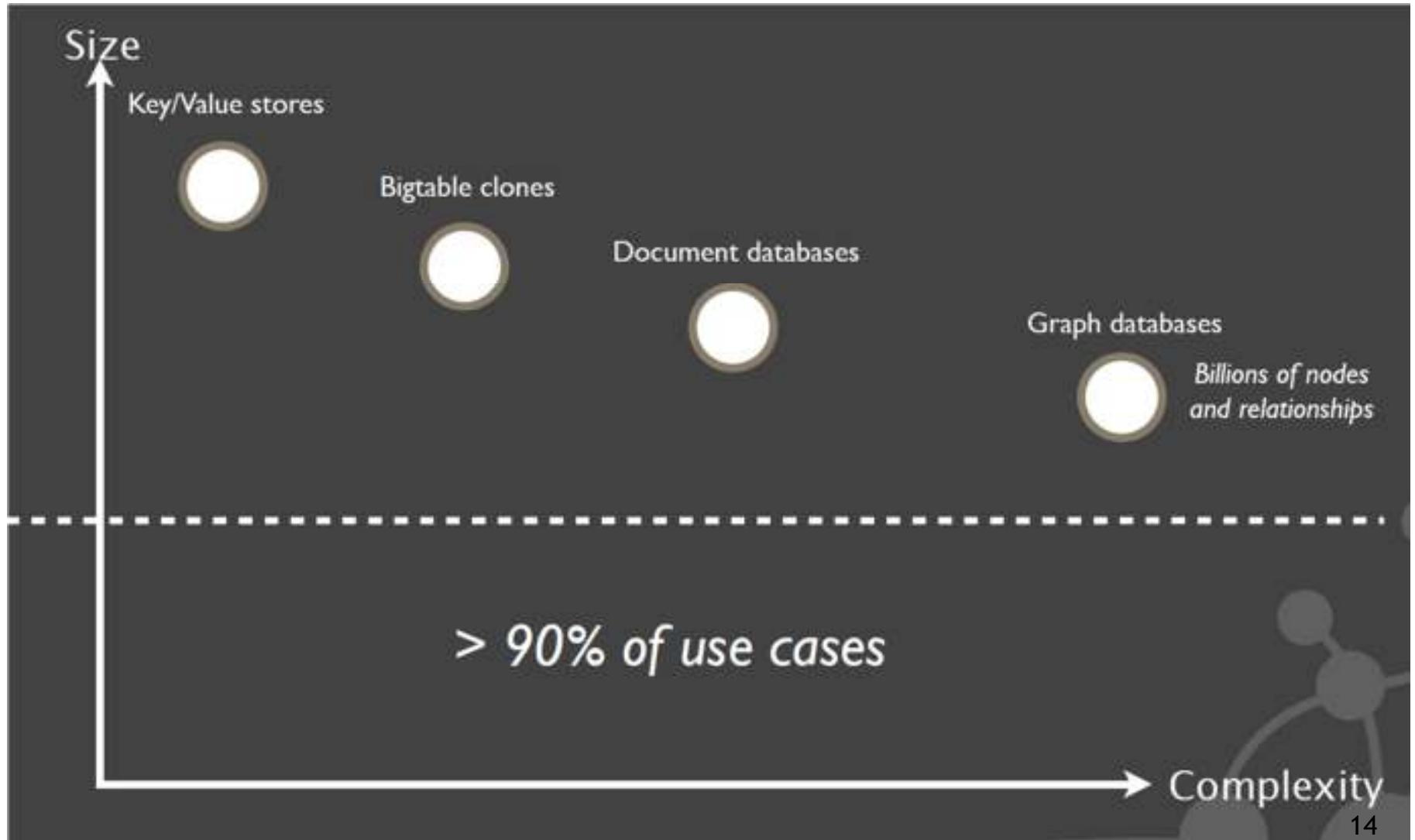
Categorias: Modelos de Dados



NOSQL Databases

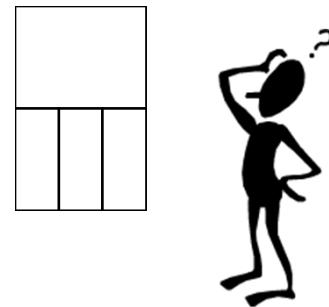


Tamanho X Complexidade



Lápide

- ER
- Modelagem
- Mapeamento
- Especificação
- Normalização

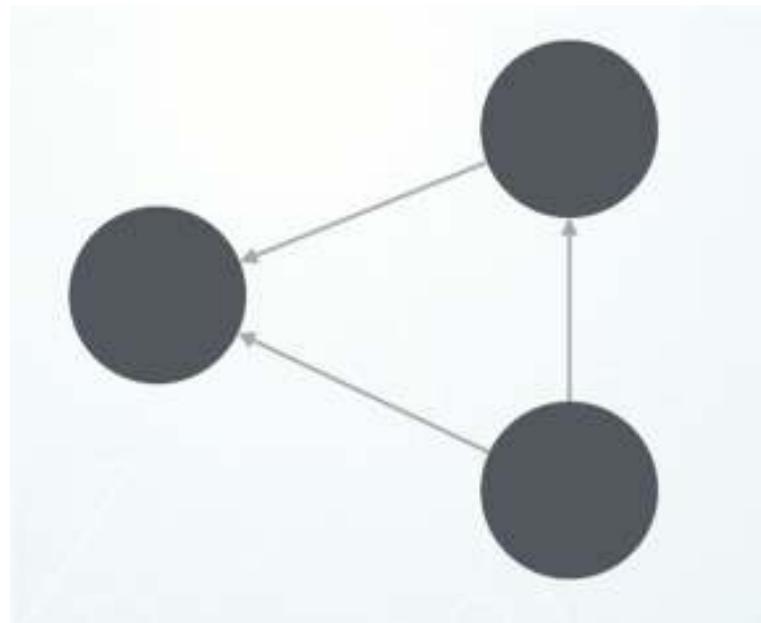


Bancos de Dados de Grafos

O que é um BD grafos ?

- BD NOSQL
- Utiliza modelos de dados baseados em grafos

Um grafo é

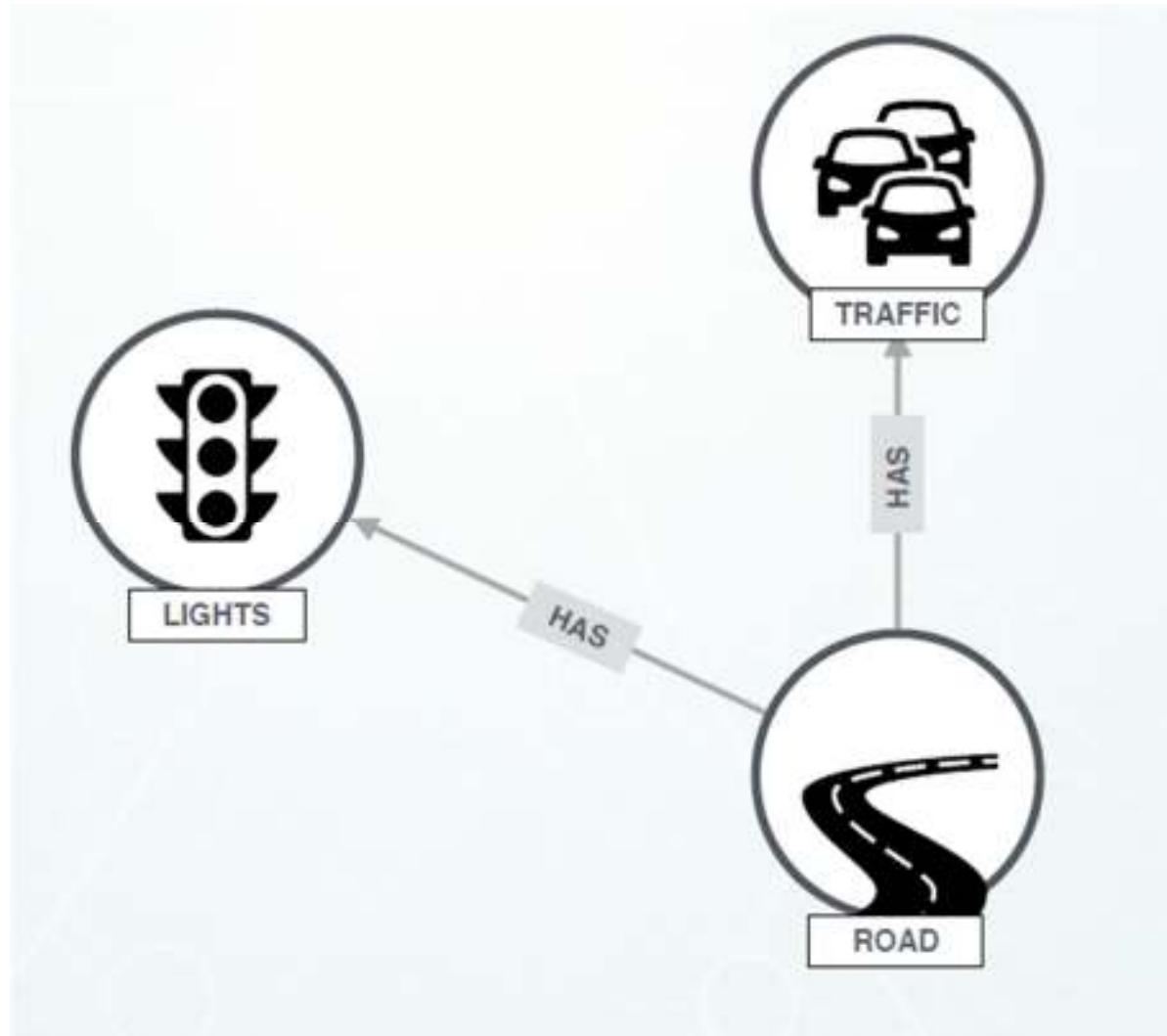


- um conjunto V de vértices (vértice = nó)
- um conjunto E de arestas: pares ordenados $v \in w \in V$ (aresta = arco)

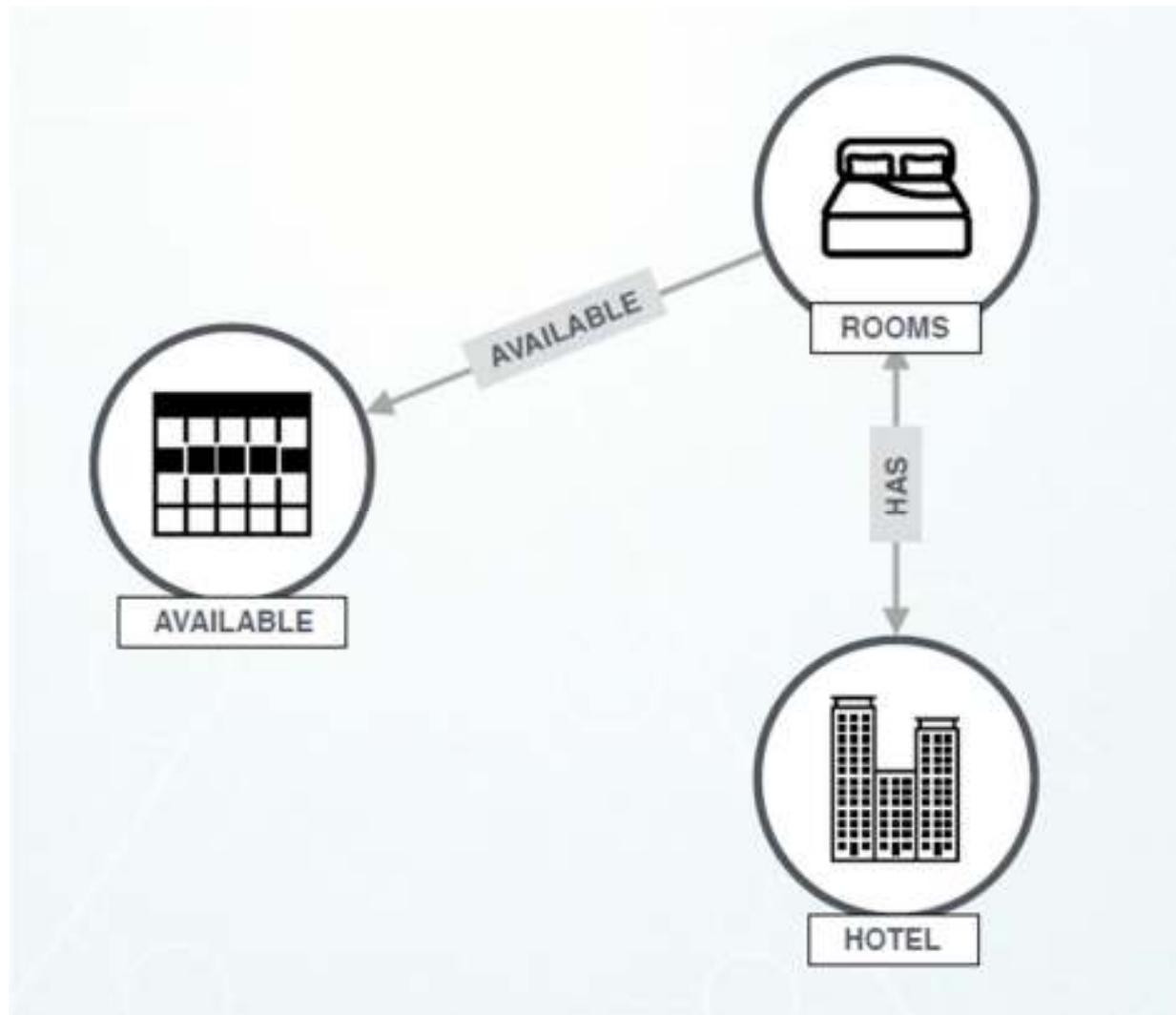
Por que grafos ?

- Vários cenários e problemas do mundo real podem ser mapeados como grafos

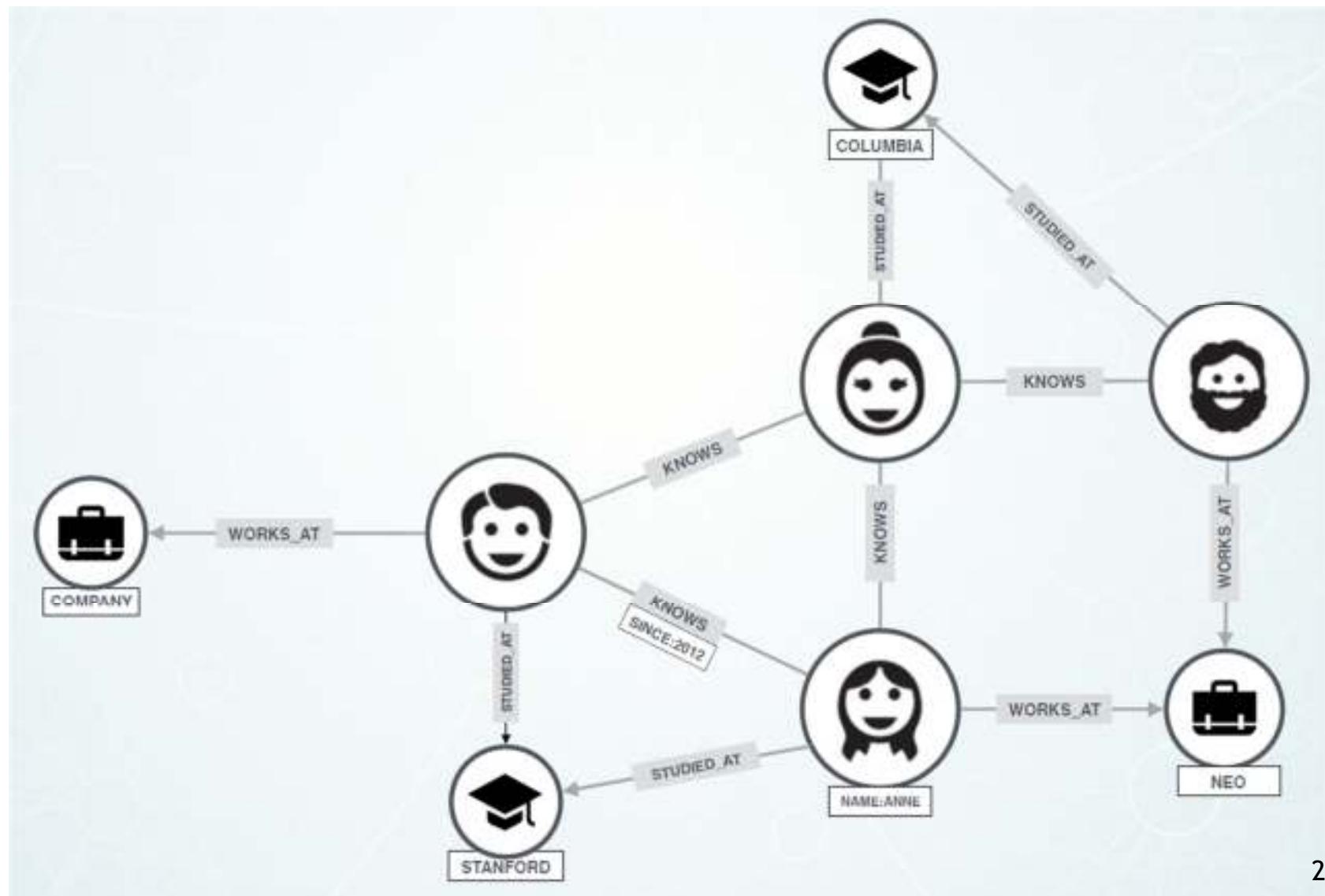
Tráfego



Hotéis



Relacionamentos

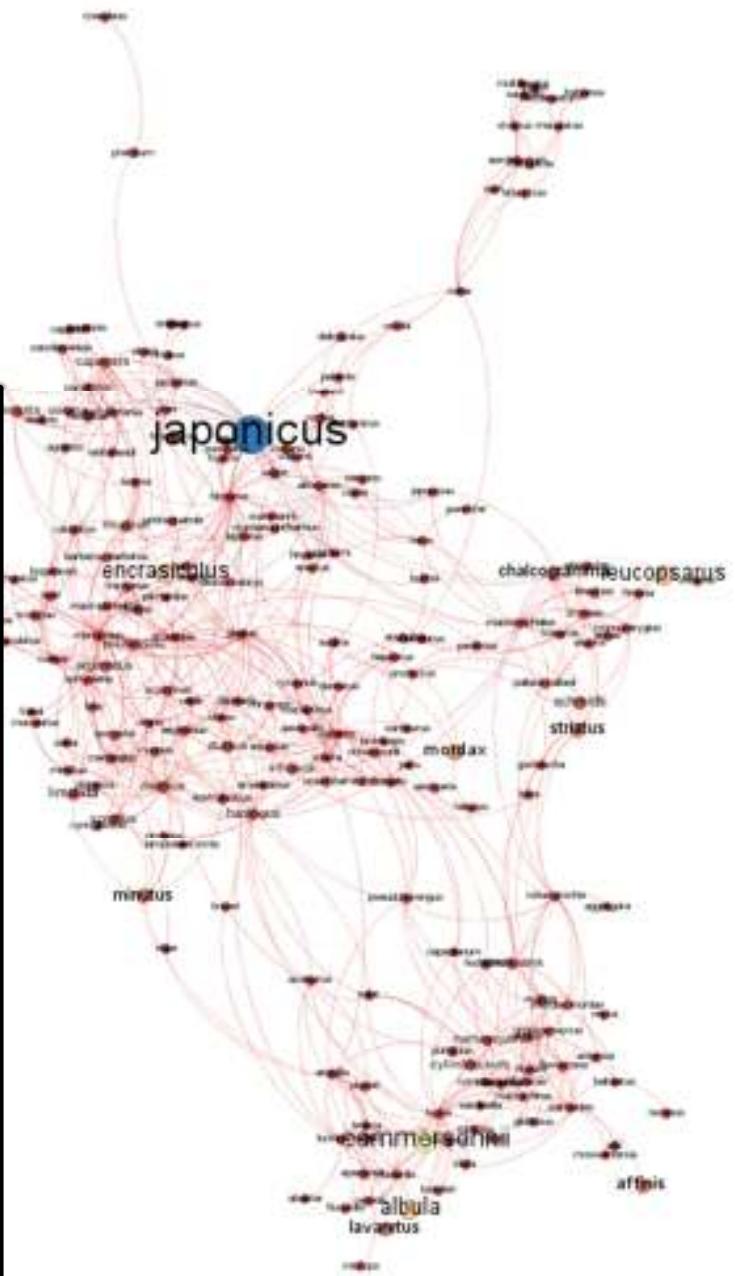
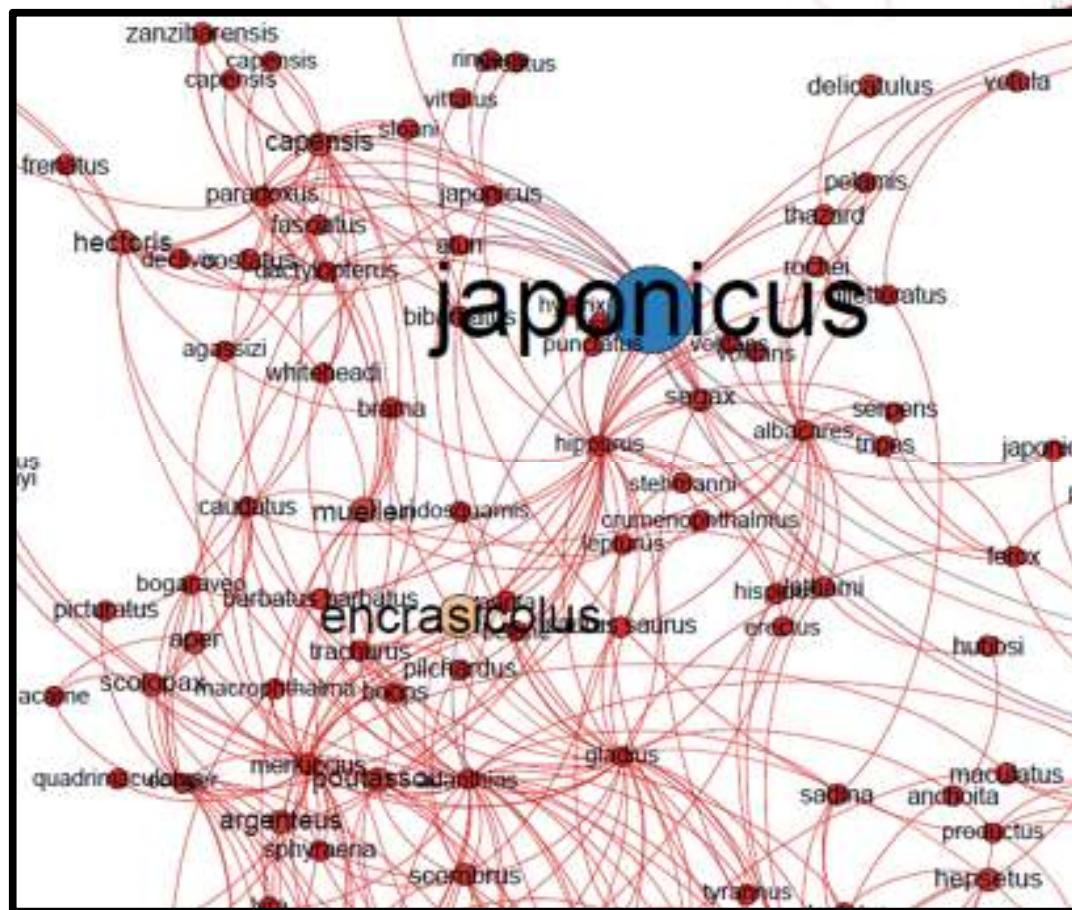


Twitter (Influência)

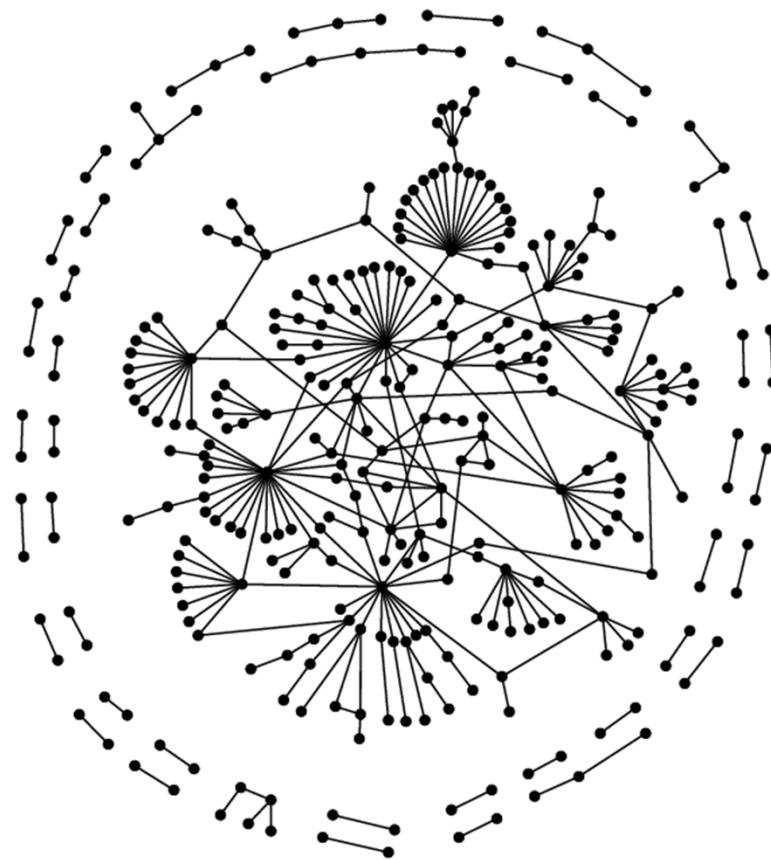


Created with NodeXL (<http://nodeXL.codeplex.com>) from the Social Media Research Foundation (<http://www.smrfoundation.org>)

Cadeia Alimentar FishBase



Proteínas da Levedura



Yeast proteins: Sergei Maslov and Kim Sneppen,
[Specificity and stability in topology of protein networks](#),
Science 296, 910-913 (2002).

Rede hidrográfica Brasil - 620K nós

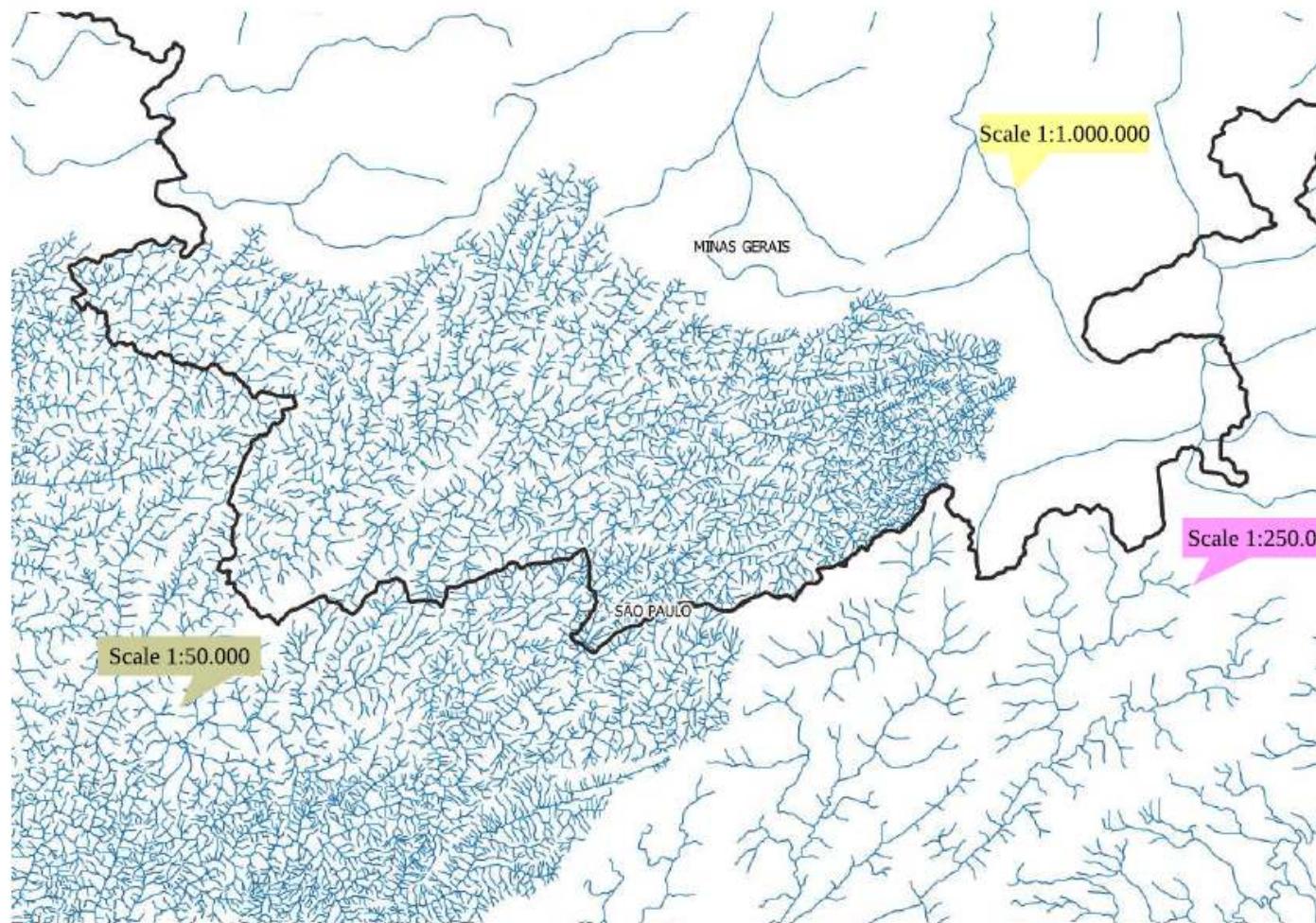
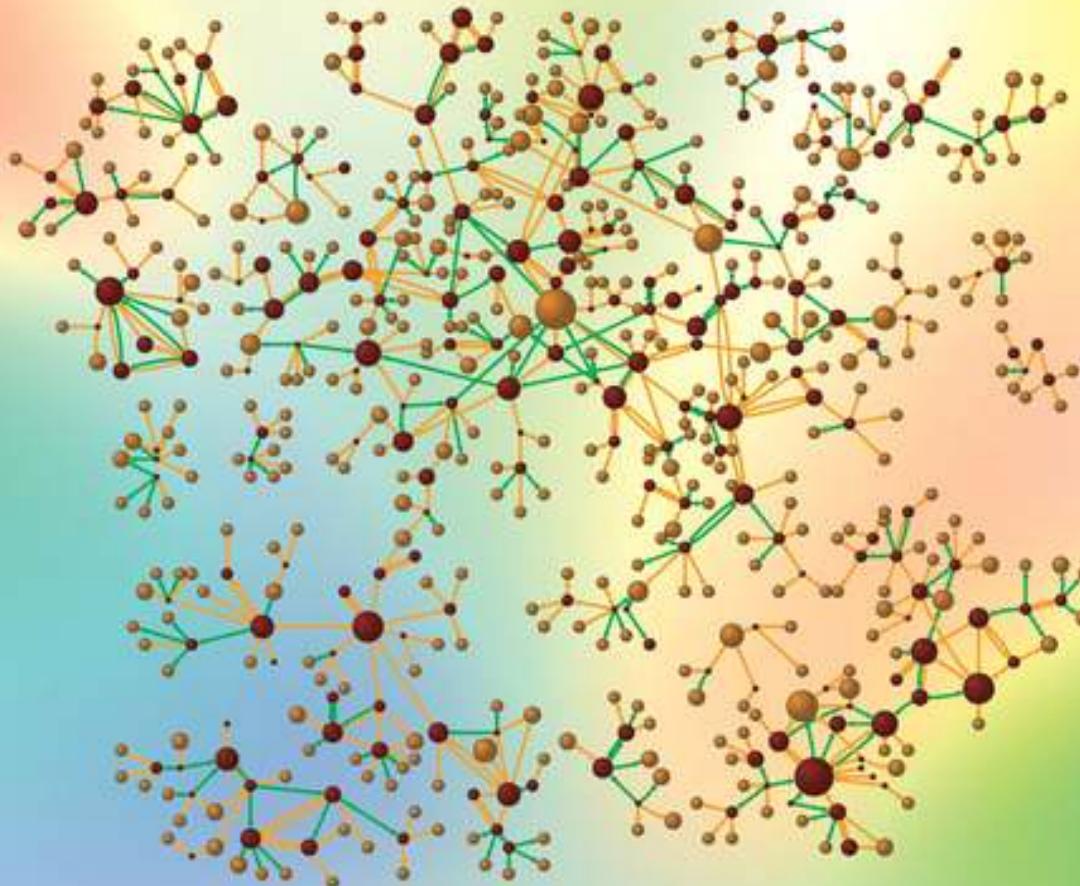


Fig. 2 Different Drainage Stretch Scales in Drainage Network

NETWORK SCIENCE



NATIONAL RESEARCH COUNCIL
OF THE NATIONAL ACADEMIES

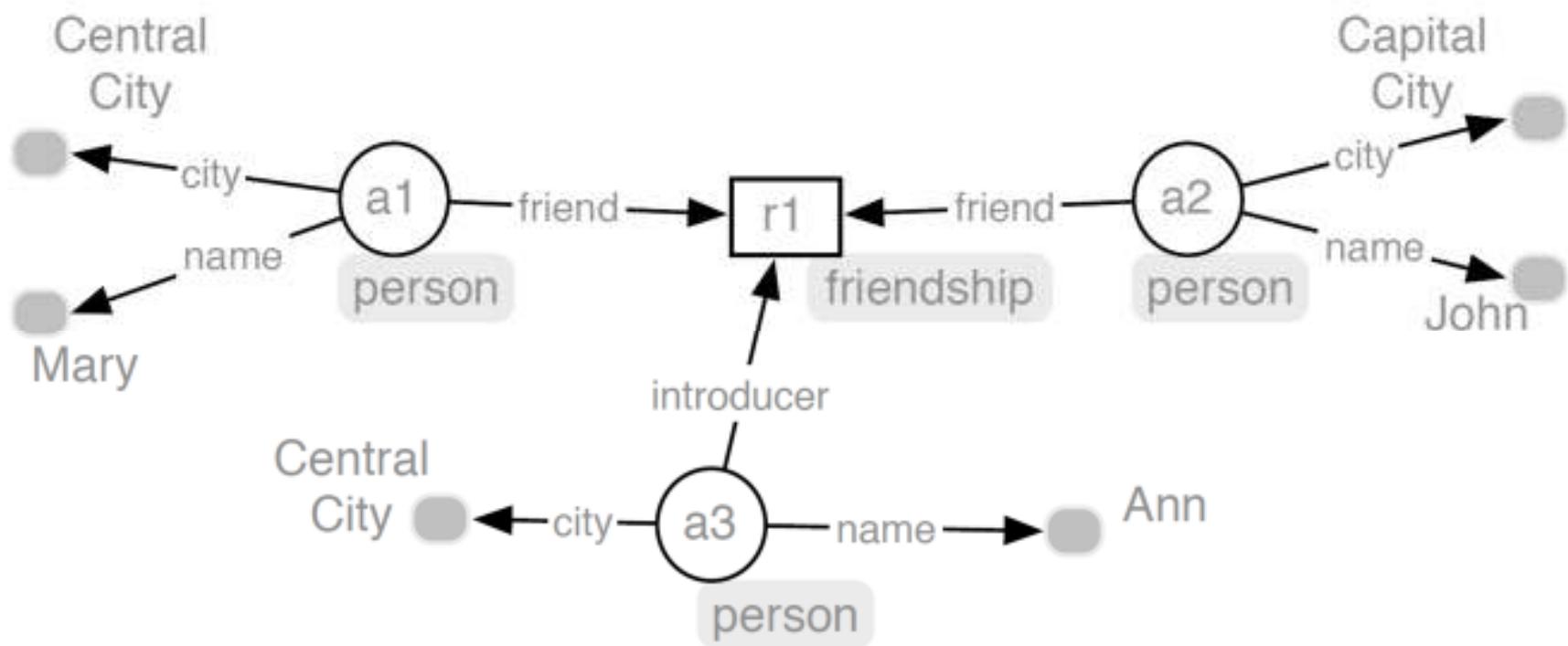
Modelo de Dados (Codd, 1980)

- Coleção de elementos para representar dados e expressar detalhes semânticos
- Componentes
 - Tipos de estruturas de dados
 - Restrições de integridade para definir estados consistentes do banco de dados
 - Operadores para recuperar dados

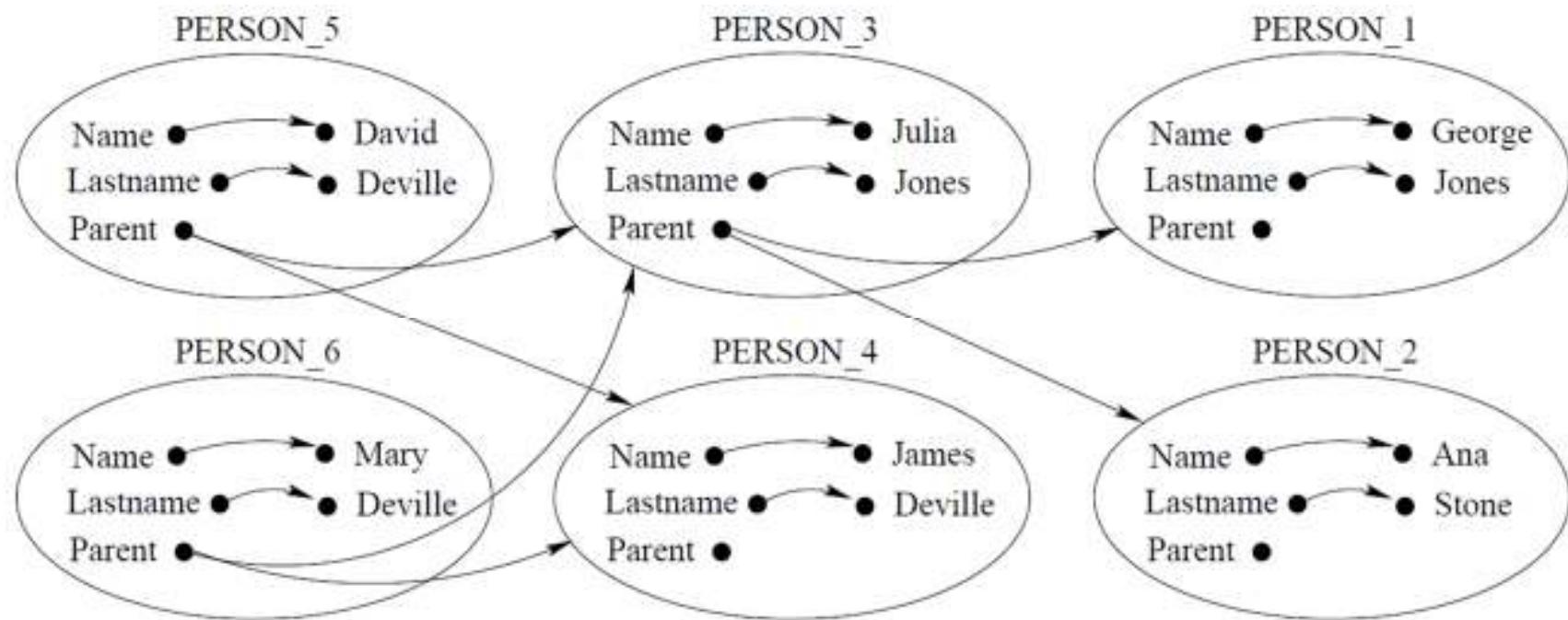
Estrutura de Dados

- Variações sob a definição básica
- Aumentar expressividade
- Representar cenários específicos de forma menos ambígua

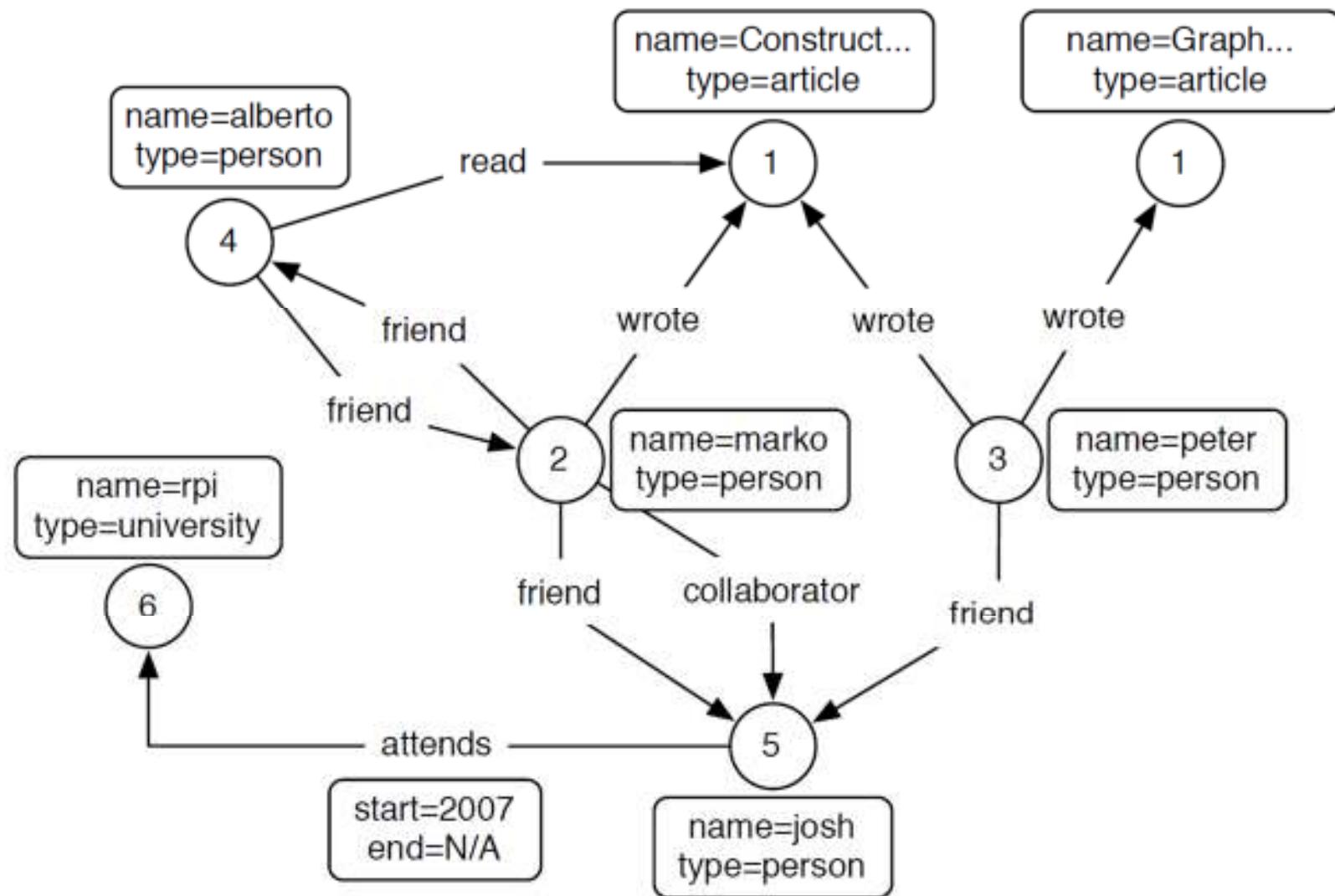
Grafo RDF



Hipergrafo



Grafo de Propriedades



Modelo de Dados (Codd, 1980)

- Coleção de elementos para representar dados e expressar detalhes semânticos
- Componentes
 - Tipos de estruturas de dados
 - Restrições de integridade para definir estados consistentes do banco de dados
 - Operadores para recuperar dados

DB Grafos

<http://db-engines.com/en/ranking/graph+dbms>

30 systems in ranking, May 2018									
Rank	DBMS			Database Model	Score				
	May 2018	Apr 2018	May 2017		May 2018	Apr 2018	May 2017		
1.	1.	1.	Neo4j 	Graph DBMS	40.58	-0.32	+4.44		
2.	2.	+	Microsoft Azure Cosmos DB 	Multi-model 	17.54	+0.35	+12.70		
3.	3.		Datastax Enterprise 	Multi-model 	7.38	-0.09			
4.	4.	↓ 2.	OrientDB 	Multi-model 	5.25	-0.39	-0.49		
5.	5.	5.	ArangoDB	Multi-model 	3.70	-0.10	+0.75		
6.	6.	6.	Virtuoso	Multi-model 	1.79	-0.01	-0.27		
7.	7.	7.	Giraph	Graph DBMS	0.98	-0.06	-0.11		
8.	8.		Amazon Neptune	Multi-model 	0.71	+0.02			
9.	9.	↓ 8.	AllegroGraph 	Multi-model 	0.58	+0.00	-0.02		
10.	10.	↓ 9.	Stardog	Multi-model 	0.51	-0.02	+0.00		
11.	11.	↓ 10.	GraphDB 	Multi-model 	0.46	-0.00	-0.04		
12.	+	14. + 19.	JanusGraph	Graph DBMS	0.41	+0.12	+0.29		
13.	↓ 12.	↑ 16.	Graph Engine	Multi-model 	0.36	-0.04	+0.18		
14.	↓ 13.	↓ 11.	Sqrrl	Multi-model 	0.33	-0.09	-0.13		
15.	15.	↑ 21.	Sparksee	Graph DBMS	0.19	-0.02	+0.14		
16.	16.		TigerGraph 	Graph DBMS	0.17	-0.01			
17.	+	20. ↓ 14.	Blazegraph	Multi-model 	0.14	+0.01	-0.13		
18.	18.	↓ 12.	Dgraph	Graph DBMS	0.14	+0.00	-0.15		
19.	↓ 17.	↓ 17.	HyperGraphDB	Graph DBMS	0.14	-0.01	-0.03		
20.	↓ 19.	↓ 15.	FlockDB	Graph DBMS	0.13	+0.00	-0.06		
21.	+	22. ↓ 13.	InfiniteGraph	Graph DBMS	0.13	+0.02	-0.15		
22.	+	23. 22.	VelocityDB	Multi-model 	0.10	+0.02	+0.06		
23.	↓ 21.	↓ 18.	InfoGrid	Graph DBMS	0.10	-0.02	-0.03		
24.	+	25. 24.	AgensGraph 	Multi-model 	0.04	+0.01	+0.03		
25.	↓ 24.		TinkerGraph	Graph DBMS	0.04	-0.00			
26.	+	29.	HGraphDB	Graph DBMS	0.01	-0.01			
27.	↓ 26.	↓ 23.	GraphBase	Graph DBMS	0.01	-0.01	-0.03		
28.			AnzoGraph	Graph DBMS	0.01				
29.	↓ 28.	↓ 20.	GlobalsDB	Multi-model 	0.00	-0.02	-0.06		
30.	↓ 27.	↓ 25.	GRAKN.AI 	Multi-model 	0.00	-0.02	+0.00		

342 sistemas

30 de grafos

Neo4J – 22

Giraph - 124

DB Grafos

<http://db-engines.com/en/ranking/graph+dbms>

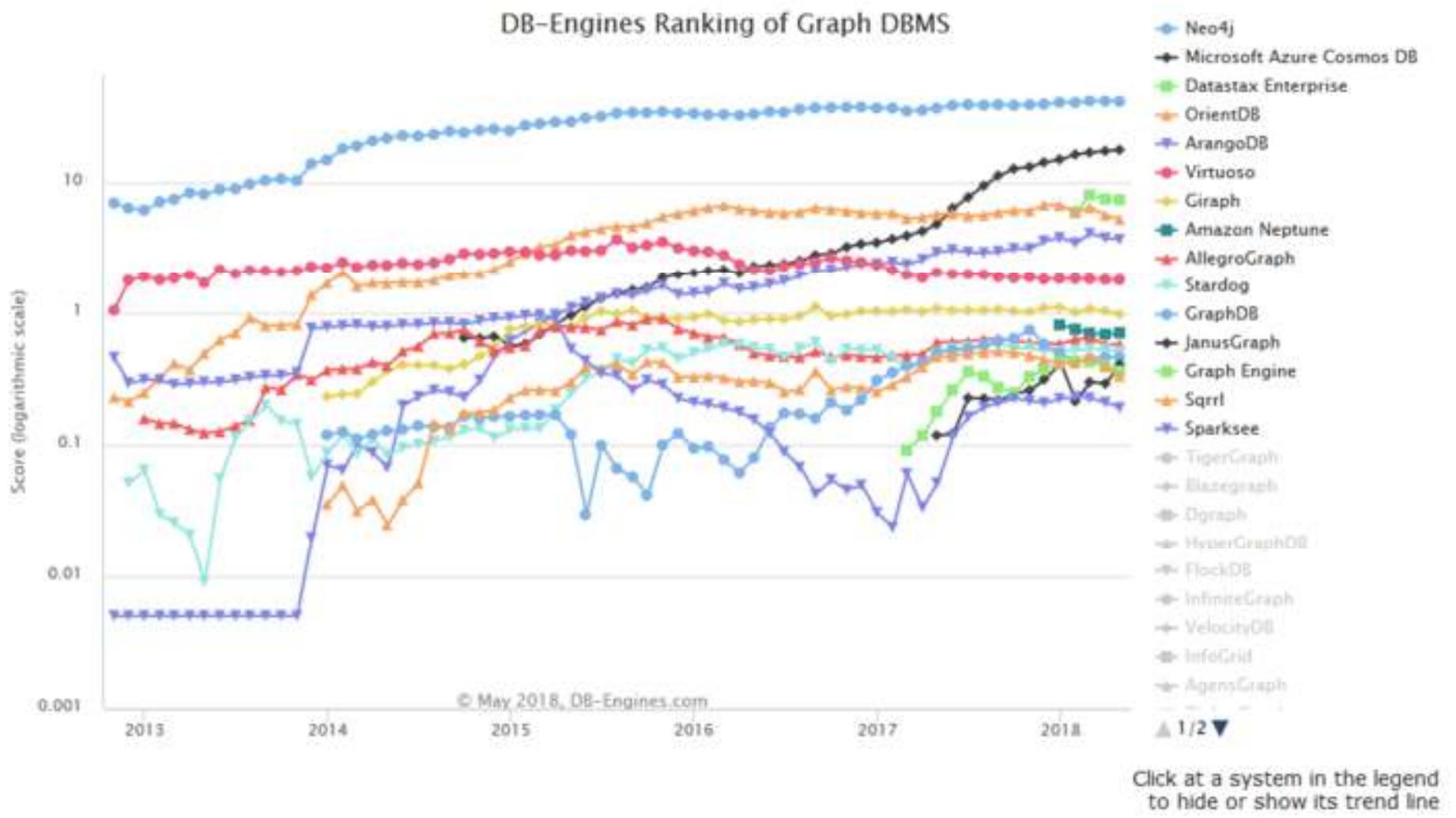
30 systems in ranking, May 2018

Rank	DBMS			Database Model	Score		
	May 2018	Apr. 2018	May 2017		May 2018	Apr. 2018	May 2017
1.	1.	1.	Neo4j	Graph DBMS	40.58	-0.32	+4.44
2.	2.	↑ 4.	Microsoft Azure Cosmos DB	Multi-model	17.54	+0.35	+12.70
3.	3.		Datastax Enterprise	Multi-model	7.38	-0.09	
4.	4.	↓ 2.	OrientDB	Multi-model	5.25	-0.39	-0.49
5.	5.	5.	ArangoDB	Multi-model	3.70	-0.10	+0.75
6.	6.	6.	Virtuoso	Multi-model	1.79	-0.01	-0.27
7.	7.	7.	Giraph	Graph DBMS	0.98	-0.06	-0.11
8.	8.		Amazon Neptune	Multi-model	0.71	+0.02	
9.	9.	↓ 8.	AllegroGraph	Multi-model	0.58	+0.00	-0.02
10.	10.	↓ 9.	Stardog	Multi-model	0.51	-0.02	+0.00
11.	11.	↓ 10.	GraphDB	Multi-model	0.46	-0.00	-0.04
12.	↑ 14.	↑ 19.	JanusGraph	Graph DBMS	0.41	+0.12	+0.29
13.	↓ 12.	↑ 16.	Graph Engine	Multi-model	0.36	-0.04	+0.18
14.	↓ 13.	↓ 11.	Sqrrl	Multi-model	0.33	-0.09	-0.13
15.	15.	↑ 21.	Sparksee	Graph DBMS	0.19	-0.02	+0.14
16.	16.		TigerGraph	Graph DBMS	0.17	-0.01	
17.	↑ 20.	↓ 14.	Blazegraph	Multi-model	0.14	+0.01	-0.13
18.	18.	↓ 12.	Dgraph	Graph DBMS	0.14	+0.00	-0.15
19.	↓ 17.	↓ 17.	HyperGraphDB	Graph DBMS	0.14	-0.01	-0.02
20.	↓ 19.	↓ 15.	FlockDB	Graph DBMS	0.13	+0.00	-0.06
21.	↑ 22.	↓ 13.	InfiniteGraph	Graph DBMS	0.13	+0.02	-0.15
22.	↑ 23.	22.	VelocityDB	Multi-model	0.10	+0.02	+0.06
23.	↓ 21.	↓ 18.	InfoGrid	Graph DBMS	0.10	-0.02	-0.03
24.	↑ 25.	24.	AgensGraph	Multi-model	0.04	+0.01	+0.03
25.	↓ 24.		TinkerGraph	Graph DBMS	0.04	-0.00	
26.	↑ 29.		HGraphDB	Graph DBMS	0.01	-0.01	
27.	↓ 26.	↓ 23.	GraphBase	Graph DBMS	0.01	-0.01	-0.03
28.			AnzoGraph	Graph DBMS	0.01		
29.	↓ 28.	↓ 20.	GlobalsDB	Multi-model	0.00	-0.02	-0.06
30.	↓ 27.	↓ 25.	GRAKN.AI	Multi-model	0.00	-0.02	+0.00

MULTI?

DB Grafos

<http://db-engines.com/en/ranking/graph+dbms>



Neo4j

- Grafo rotulado de propriedades tipadas
 - Arestas possuem tipos
 - É possível mais de uma aresta, de tipos diferentes, entre dois vértices
 - Vértices podem possuir rótulos
 - É possível múltiplos rótulos em um mesmo vértice



37

Fonte: slides Michael Hunger, Relational to Graph, 2015

Em 2015

Company

- Neo Technology, Creator of Neo4j
- 80 employees with HQ in Silicon Valley, London, Munich, Paris and Malmö
- \$45M in funding from Fidelity, Sunstone, Conor, Creandum, Dawn Capital

Product

- Neo4j - World's leading graph database
- 1M+ downloads, adding 50k+ per month
- 150+ enterprise subscription customers including over 50 of the Global 2000



Empresas que adotam

Financial Services	Communications	Health & Life Sciences	HR & Recruiting	Media & Publishing	Social Web	Industry & Logistics
UBS First Data. ADP ADVENT NOMURA veda die Bayerische ice	cisco hp SFR EarthLink maail Let's connect EarthLink	GoodStart Genetics janssen ZEPHYR HEALTHCARE acuraspan™ doximity Care.com	careerbuilder InfoJobs EQUILAR	NATIONAL GEOGRAPHIC LIFECHURCH.TV codex livestation Perigee® TechCrunch	classmates eHarmony Hinge meetic SNAP	ebay now noble group SNCF KiwiRail ConocoPhillips Impact Technologies
Entertainment	Consumer Retail	Business Services	Information Services			
Bally bwin.party	gamesys YELAGO	Walmart Juice PLUS+ icobrain	compete Cerved Group	scribeestar research now	LOCKHEED MARTIN Lufthansa Systems	

Como adotam

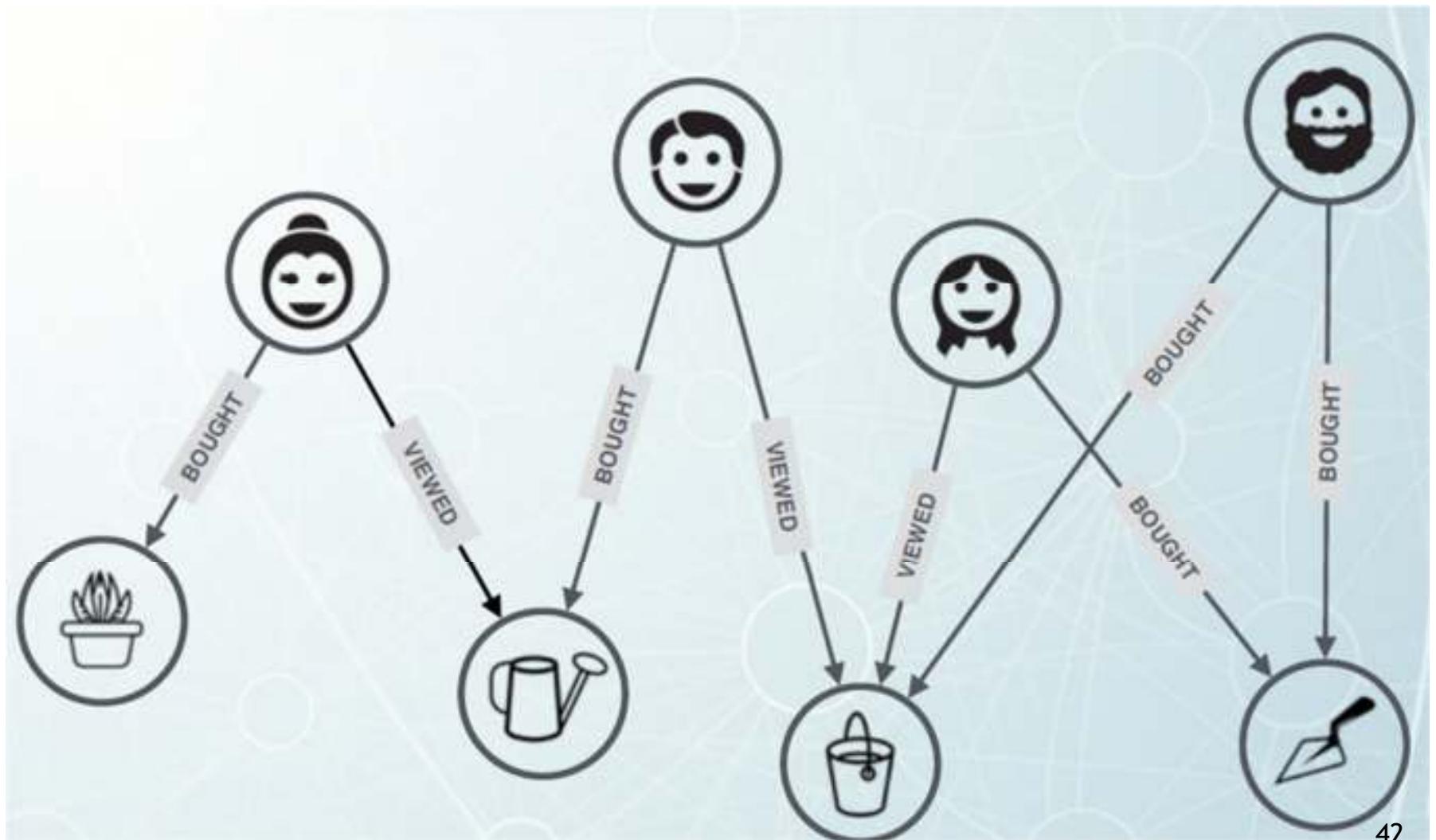
Social	Recom-mendations	Search & Discovery	Network & Data Center	Master Data Management	Identity & Access	GEO
eHarmony	careerbuilder classmates- SNAP	Walmart careerbuilder InfoJobs	IEEE	hp	cisco	TOMTOM
meetic	viadeo	koobcode	First Data. janssen	SFR	Pitney Bowes ADVENT	UBS telenor ebay now SNCF
Hinge	NATIONAL GEOGRAPHIC	LIQUID COMMON	e-Spirit	NetApp	die Bayerische	classmates-
DOWN	CloudB	Acuraspan™	codex	Zenoss	TechCrunch	KiwiRail
maiii Let's connect	Zonefinestay	NOMURA	TRIBAL technologies	gen	veda	DingLi.com Perigee®
ZEPHYR HEALTH INC	gamesys	n research now*	Lufthansa	uni sphere	ZEPHYR HEALTH INC	DOWN
mallowstreet	ice	compete	Didacti	VIRTUAL INSTRUMENTS	Didacti	CHRONOTRACK
Care.com™	bwin.party	cobrain	scribester	EarthLink	aikux.com	YELAGO
LIFECHURCH.TV	doximity	livestation	springcm	Juice PLUS+	EQUILAR	
		shutl kitedesk	noble group			

The screenshot shows the Walmart homepage. At the top, the Walmart logo is on the left, followed by a search bar with a magnifying glass icon and a 'FREE' button. Below the search bar are links for 'All Departments', 'Daily Savings Center', 'My Local Store', 'Tips & Ideas', and 'Savings Catcher'. A large green banner on the left promotes 'GAME TIME' at Walmart, showing two football players on a screen. Below the banner is the text 'Score big savings on TVs for the big game.' To the right of the banner are several product categories: 'Home' (showing a living room), 'Baby' (showing a baby sleeping), and 'Sports, Fit' (showing a tent). The bottom section features a testimonial from Marcos Wada, a Software Developer at Walmart, who says: "As the current market leader in graph databases, and with enterprise features for scalability and availability, Neo4j is the right choice to meet our demands." The Walmart logo is also present at the bottom of this section.

“As the current market leader in graph databases, and with enterprise features for scalability and availability, Neo4j is the right choice to meet our demands.”

Marcos Wada
Software Developer, Walmart

Estudos de Caso - Recomendação



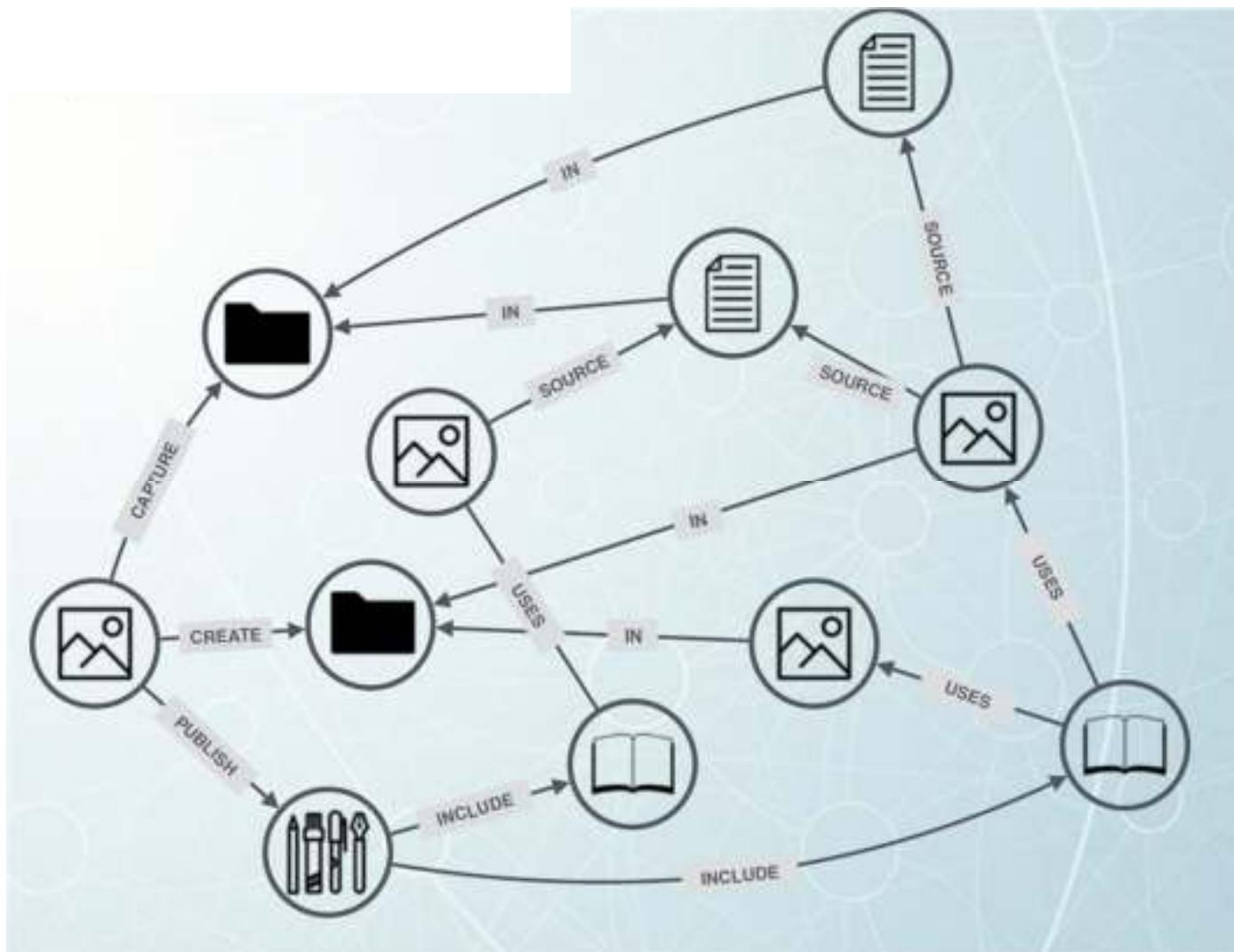
42

Fonte: slides Neo4j GraphDays, 2016



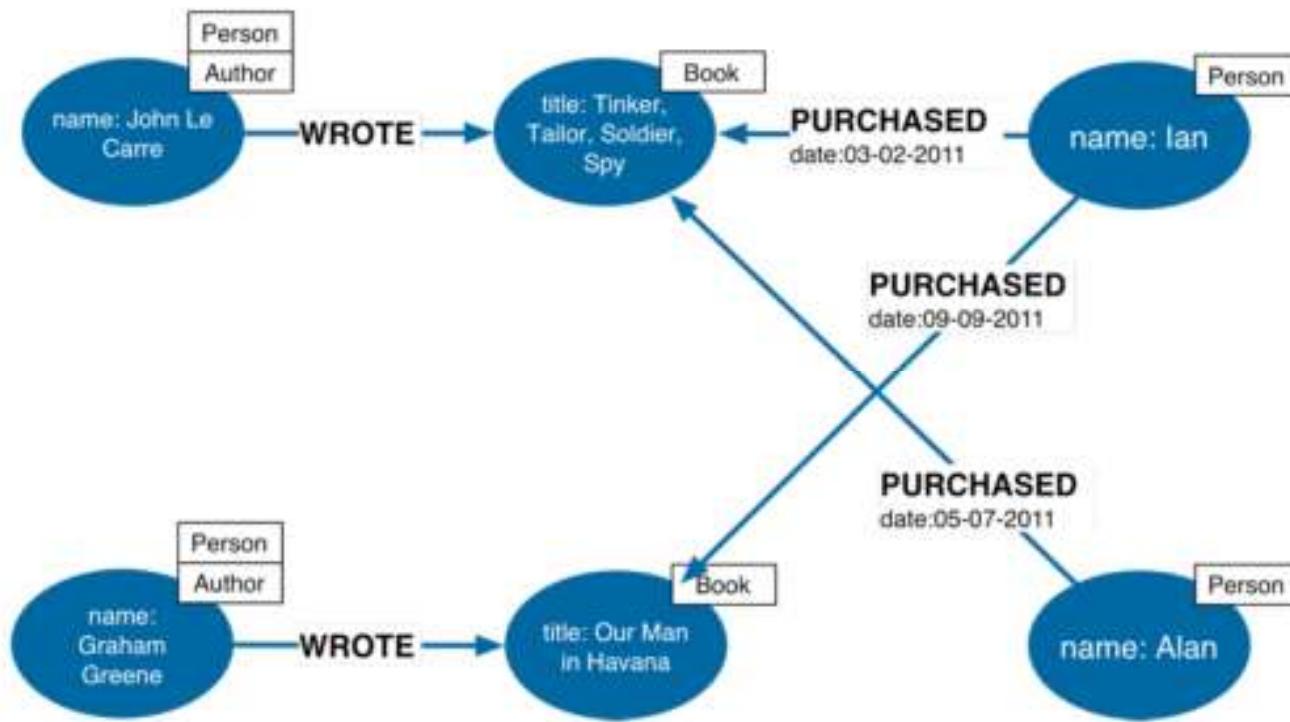
Uses Neo4j to manage the digital assets inside of its next generation in-flight entertainment system.

Estudos de Caso - Buscas

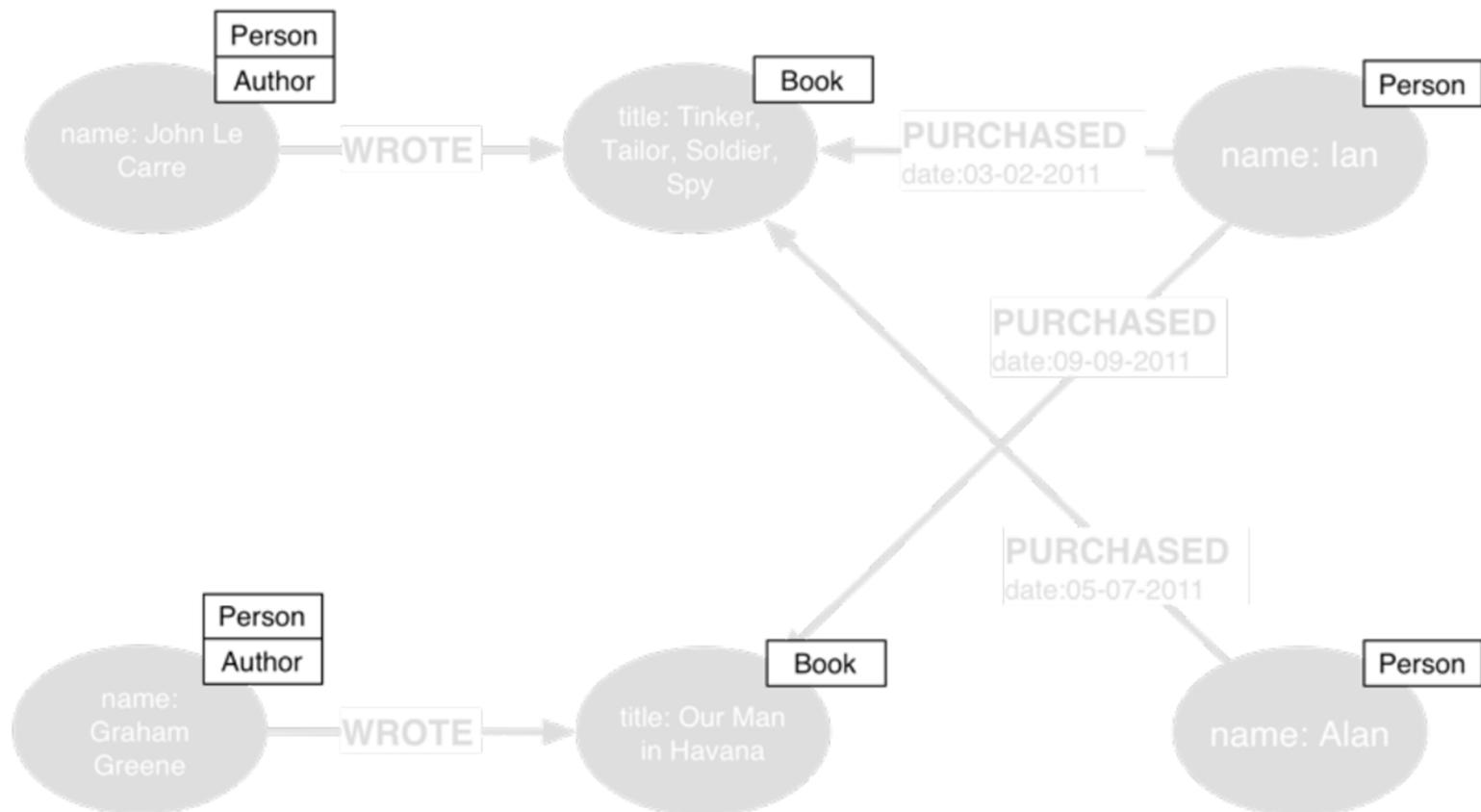


Conceitos basicos - modelo de dados do Neo4j

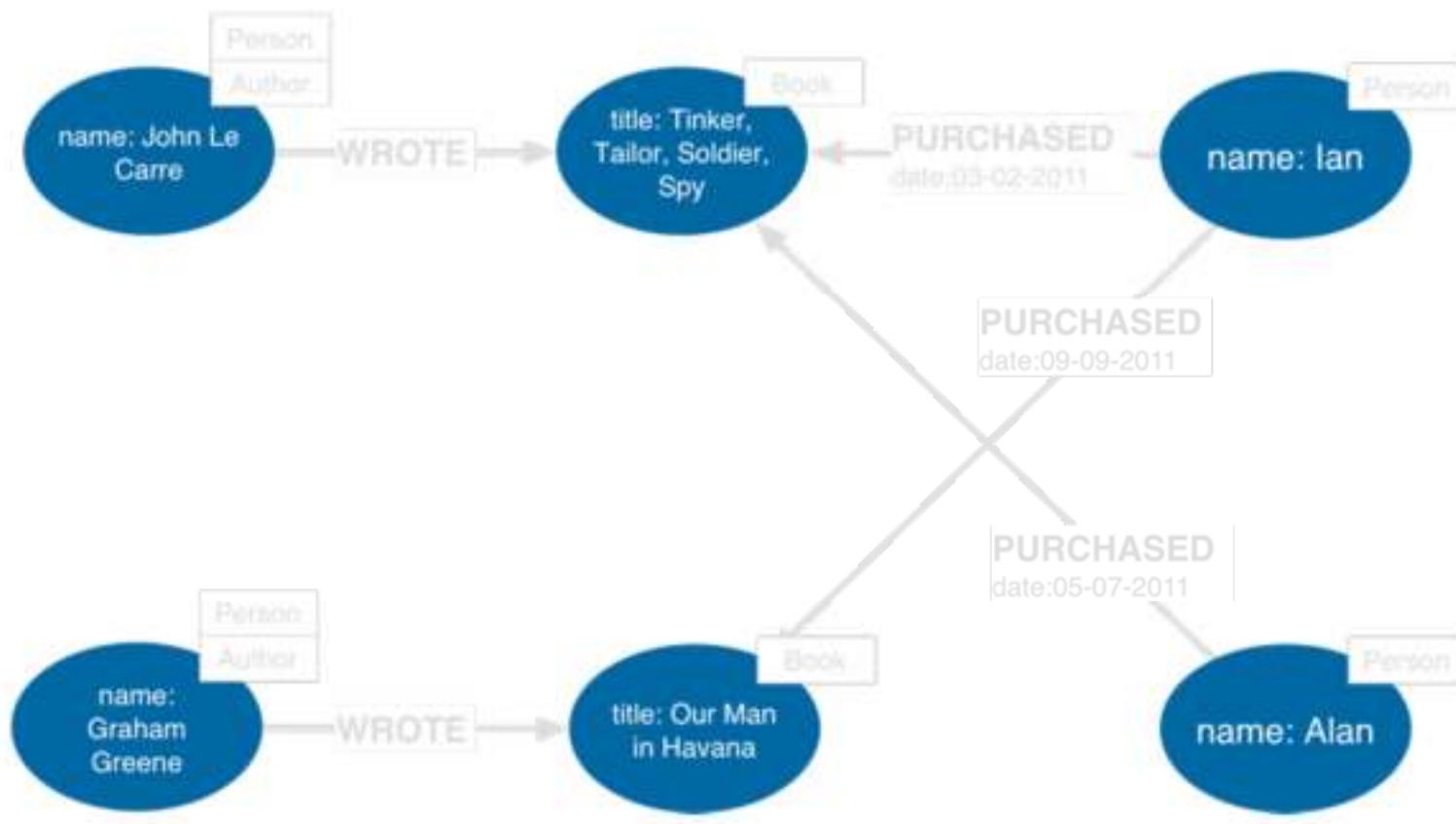
Grafo de Propriedades



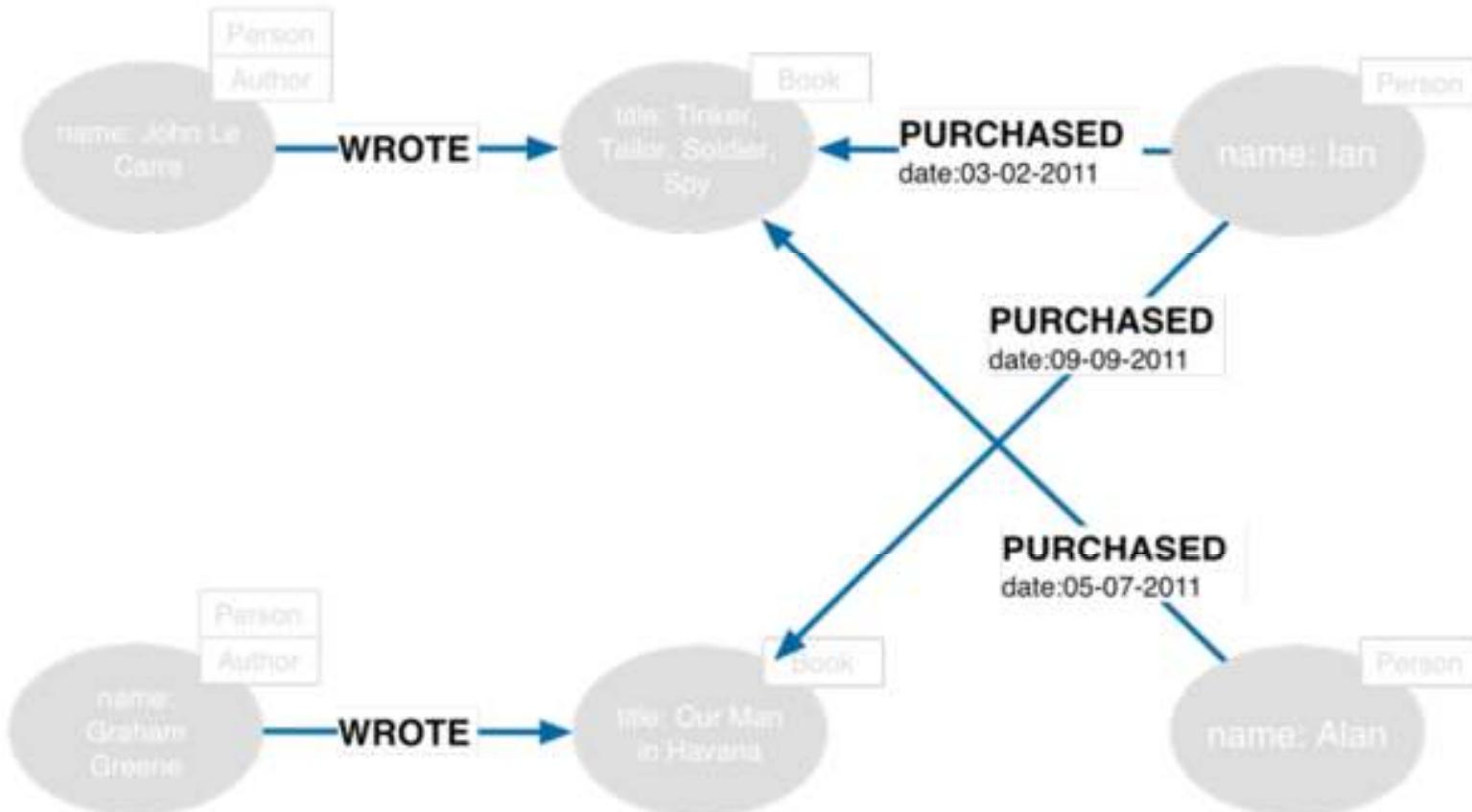
Rótulos (Labels)



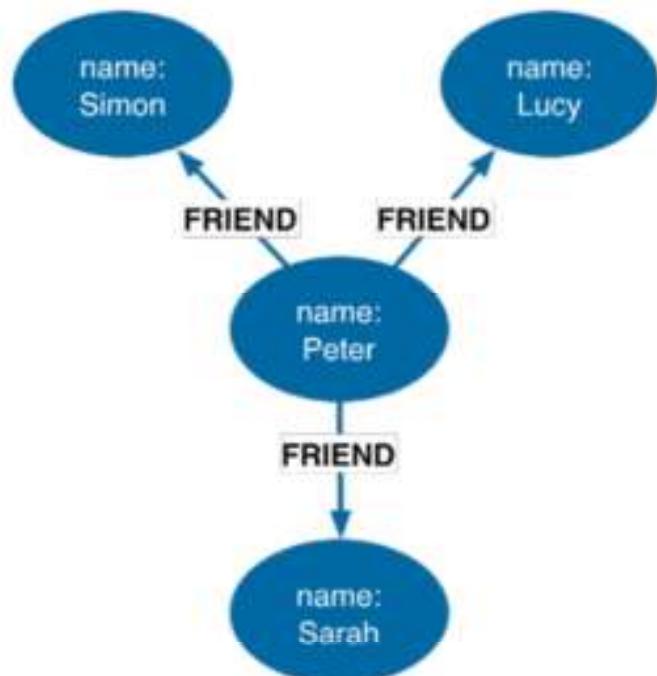
Atributos



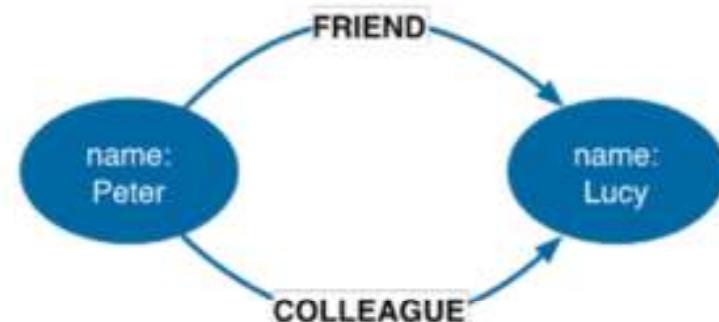
Arestas



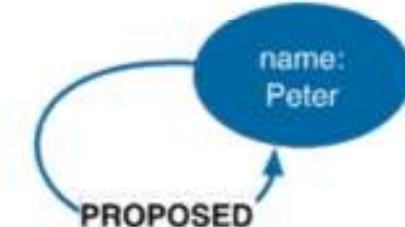
Tipos de Arestas



Nodes can have more than one relationship

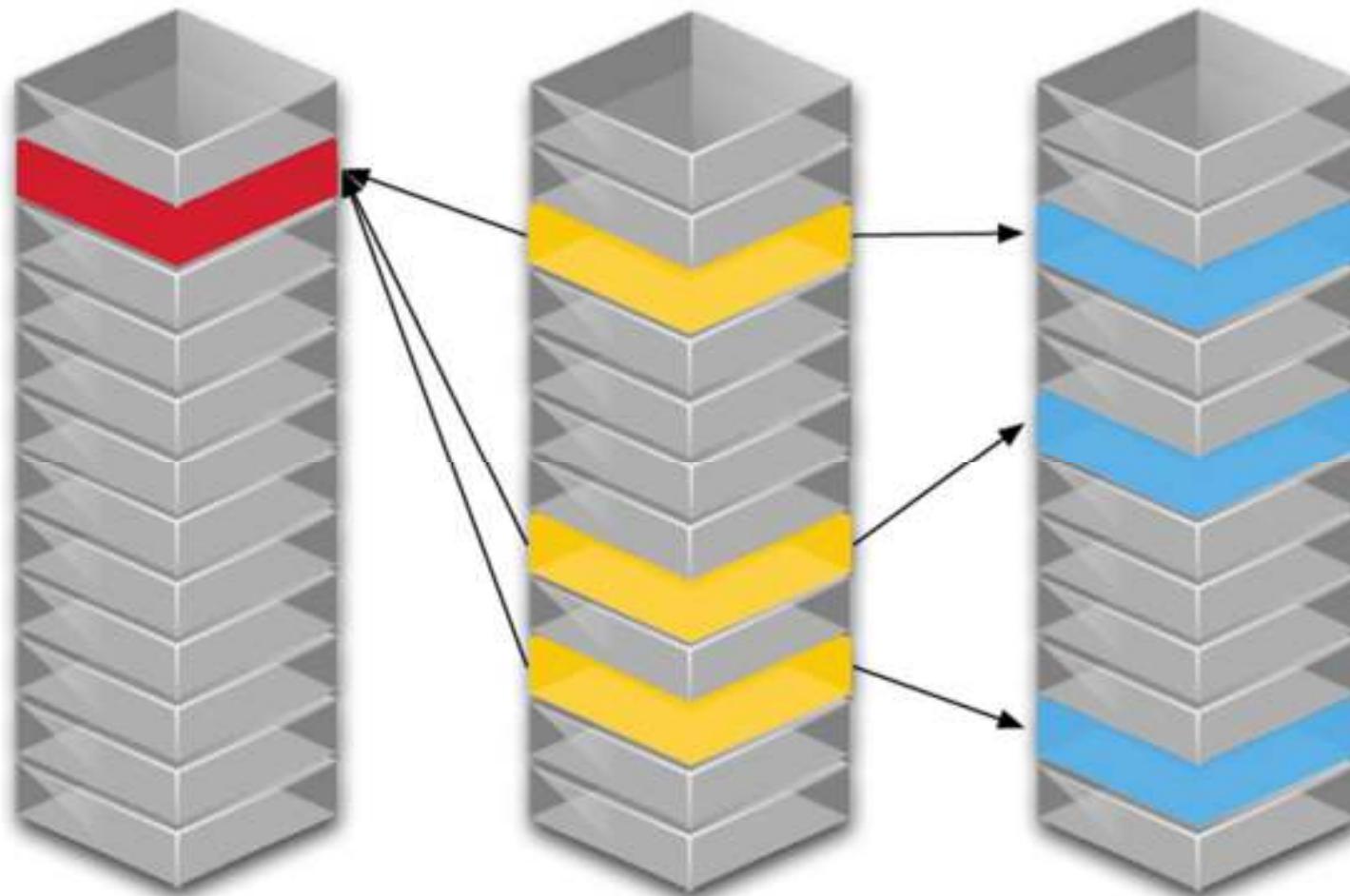


Nodes can be connected by more than one relationship

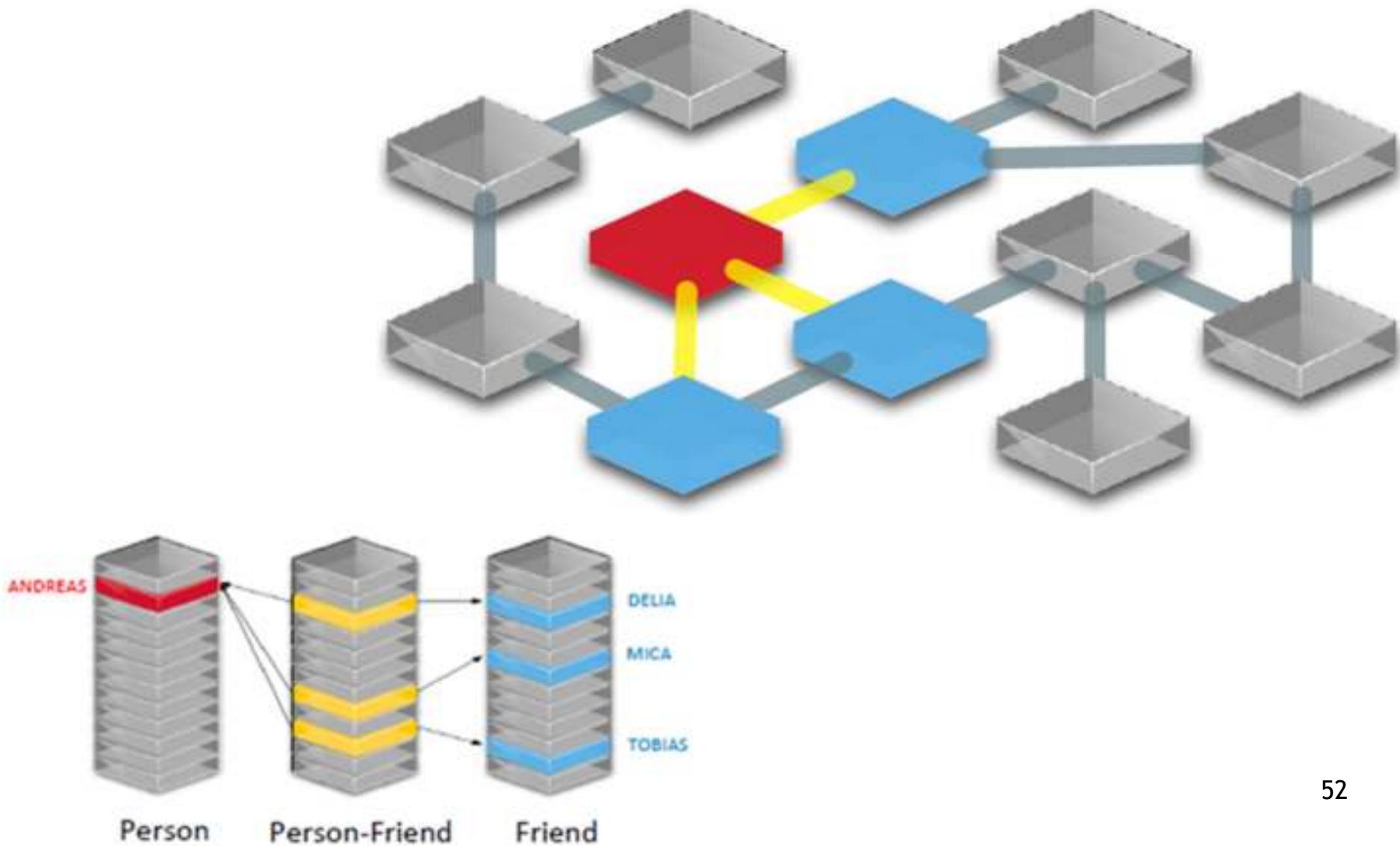


Self relationships are allowed

Mapeamento dos Relacionamentos



Mapeamento dos Relacionamentos



Cypher

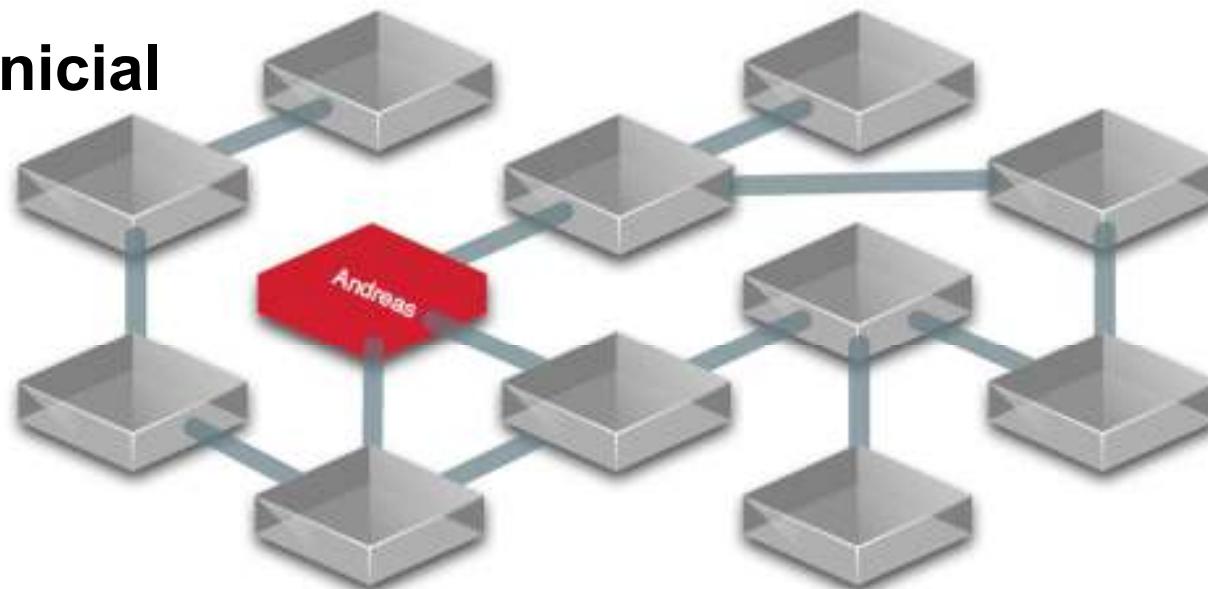
- Declarativo
- Busca Padrões

Consultas em Grafos - “Percurso”

Em Neo4j - Cypher

```
// find starting nodes  
MATCH (me:Person {name:'Andreas'})
```

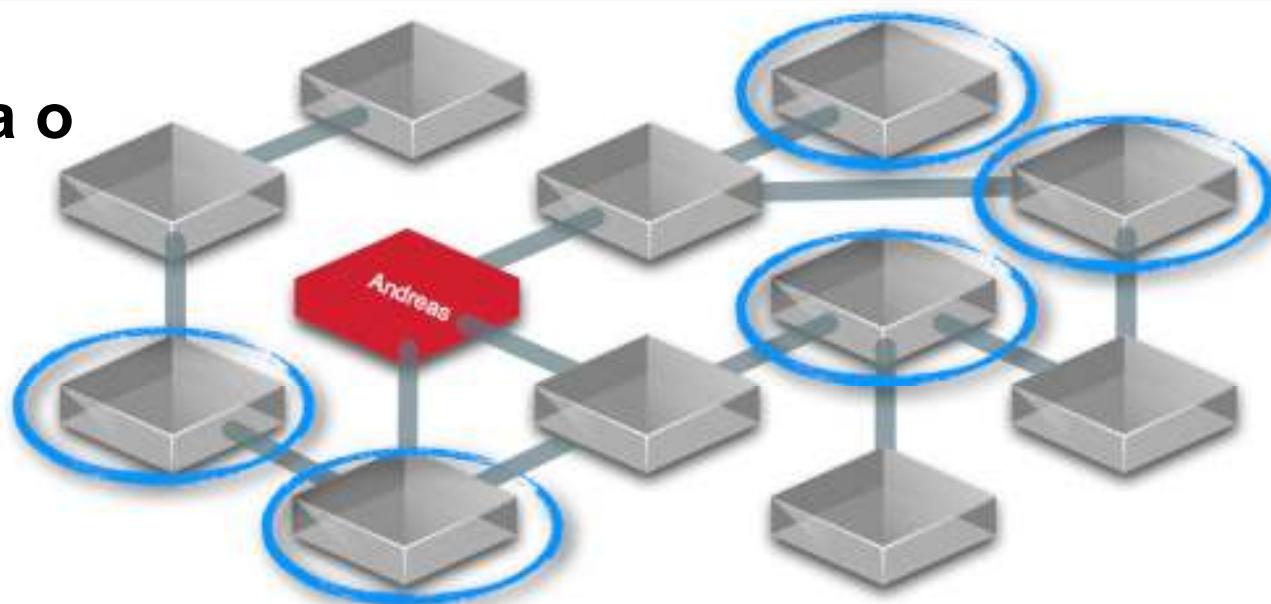
1. Ache nó inicial



Consultas em Grafos - “Percurso”

```
// then traverse the relationships  
MATCH (me:Person {name:'Andreas'})-[:FRIEND]-(friend)  
          - [:FRIEND]-(friend2)  
RETURN friend2
```

2. Percorra o grafo



Exemplo

Quem se reporta a quem ?



Consulta

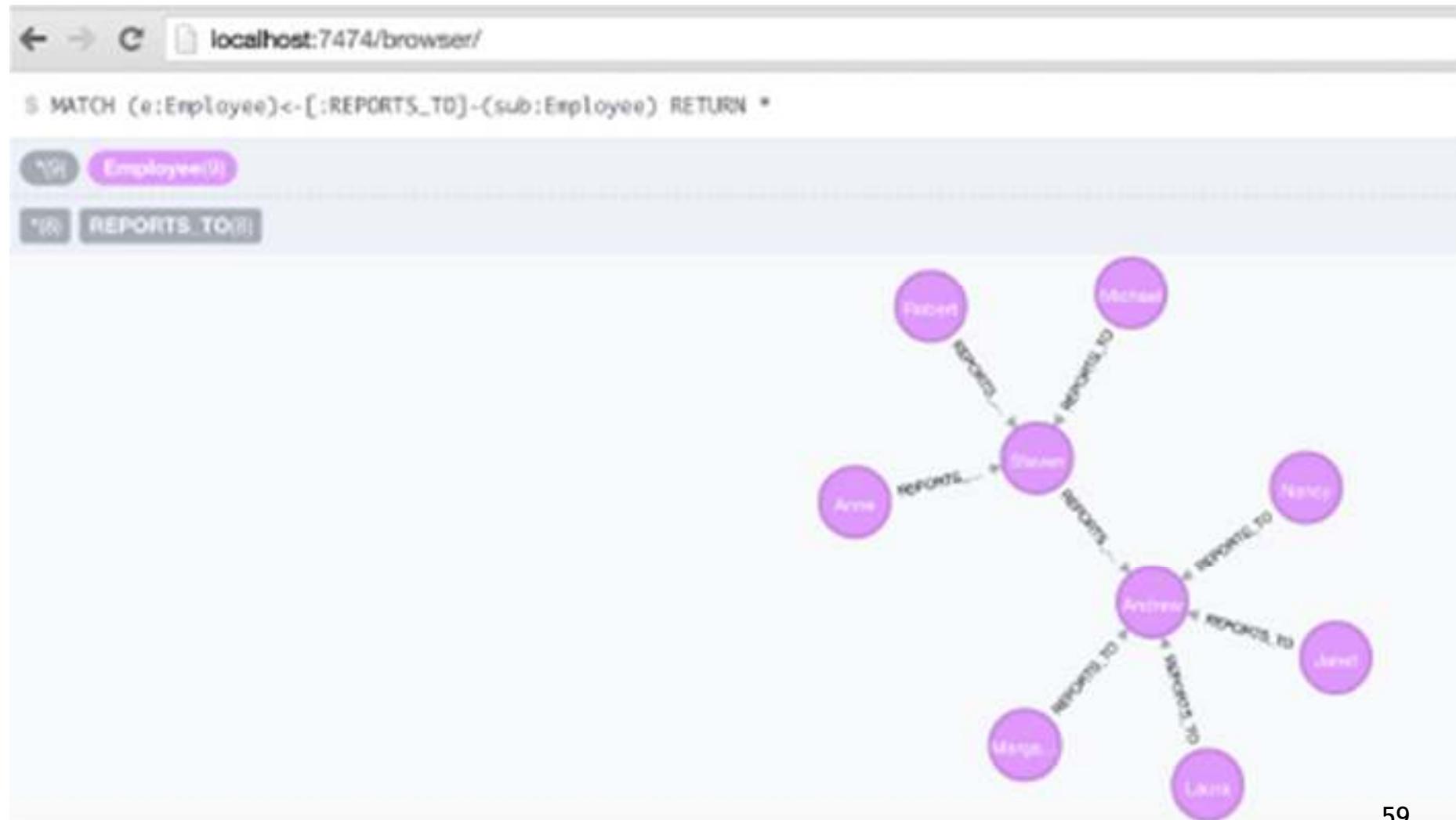
```
SELECT *
FROM Employee as e
JOIN Employee_Report AS er ON (e.id = er.manager_id)
JOIN Employee AS sub ON (er.sub_id = sub.id)
```

Consulta

```
SELECT *
FROM Employee as e
JOIN Employee_Report AS er ON (e.id = er.manager_id)
JOIN Employee AS sub ON (er.sub_id = sub.id)
```

```
MATCH
(e:Employee)-[:REPORTS_TO]-(sub:Employee)
RETURN
*
```

Quem se reporta a quem ?



Quem se reporta a quem ?

```
S MATCH path = (e:Employee)<-[ :REPORTS_TO ]-(sub) RETURN e.employeeID AS man...
```

Graph	managerID	managerName	employeeID	employeeName
	2	Andrew	5	Steven
Rows	2	Andrew	4	Margaret
	2	Andrew	3	Janet
	2	Andrew	1	Nancy
	2	Andrew	8	Laura
	5	Steven	6	Michael
	5	Steven	9	Anne
	5	Steven	7	Robert

Pessoas a quem o Robert se reporta

MATCH

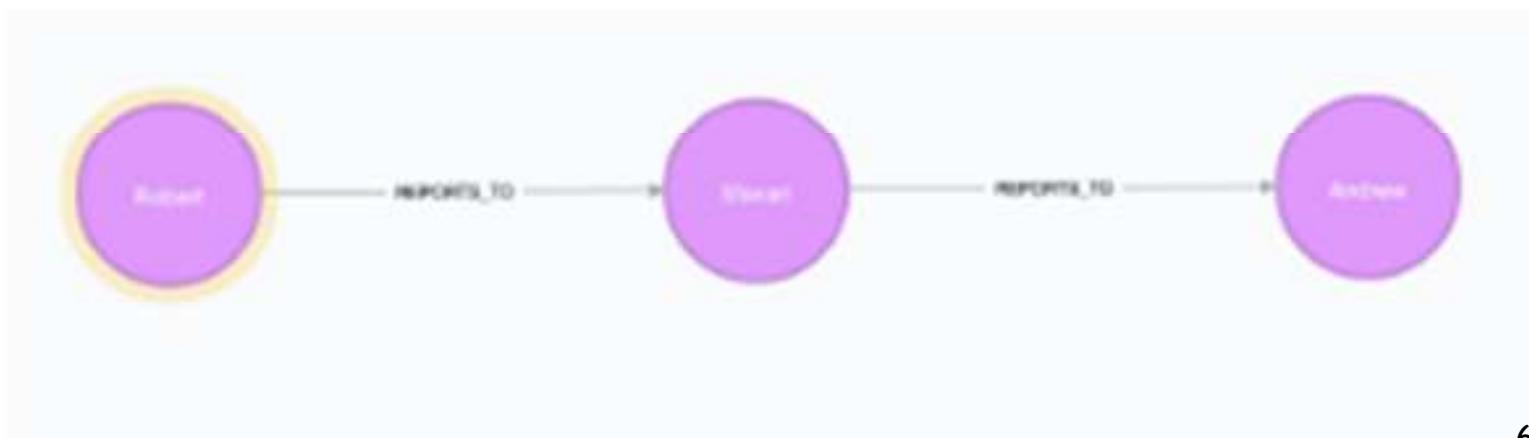
```
path=(e:Employee)-[:REPORTS_TO*]-(sub:Employee)
```

WHERE

```
sub.firstName = 'Robert'
```

RETURN

```
path;
```



Quem é o chefe ?

MATCH

```
( e :Employee )
```

WHERE

```
NOT ( e ) - [ :REPORTS_TO ] -> ( )
```

RETURN

```
e.firstName as bigBoss;
```

Estrutura de uma Consulta

```
MATCH (n:Label)-[:REL]->(m:Label)
WHERE n.prop < 42
WITH n, count(m) as cnt,
     collect(m.attr) as attrs
WHERE cnt > 12
RETURN n.prop,
       extract(a2 in
                  filter(a1 in attrs
                                 WHERE a1 =~ "...-.*")
                  | substr(a2,4,size(a2)-1))
          AS ids
ORDER BY length(ids) DESC
LIMIT 10
```

MATCH - Padrão

```
MATCH (n:Label)-[:REL]->(m:Label)
WHERE n.prop < 42
WITH n, count(m) as cnt,
      collect(m.attr) as attrs
WHERE cnt > 12
RETURN n.prop,
       extract(a2 in
                  filter(a1 in attrs
                                 WHERE a1 =~ "...-.*")
                  | substr(a2,4,size(a2)-1))
       AS ids
ORDER BY length(ids) DESC
SKIP 5 LIMIT 10
```

WHERE - Seleção (Filtro)

```
MATCH (n:Label)-[:REL]-(m:Label)
WHERE n.prop < 42
WITH n, count(m) as cnt,
     collect(m.attr) as attrs
WHERE cnt > 12
RETURN n.prop,
       extract(a2 in
                  filter(a1 in attrs
                                 WHERE a1 =~ "....-.*")
                  | substr(a2,4,size(a2)-1))
          AS ids
ORDER BY length(ids) DESC
SKIP 5 LIMIT 10
```

RETURN - Projeção

```
MATCH (n:Label)-[:REL]-(m:Label)
WHERE n.prop < 42
WITH n, count(m) as cnt,
     collect(m.attr) as attrs
WHERE cnt > 12
RETURN n.prop,
       extract(a2 in
                  filter(a1 in attrs
                                 WHERE a1 =~ "...-.*")
                  | substr(a2,4,size(a2)-1))
       AS ids
ORDER BY length(ids) DESC
SKIP 5 LIMIT 10
```

ORDER BY

```
MATCH (n:Label)-[:REL]-(m:Label)
WHERE n.prop < 42
WITH n, count(m) as cnt,
     collect(m.attr) as attrs
WHERE cnt > 12
RETURN n.prop,
       extract(a2 in
                  filter(a1 in attrs
                                 WHERE a1 =~ "...-.*")
                  | substr(a2,4,size(a2)-1))
          AS ids
ORDER BY length(ids) DESC
SKIP 5 LIMIT 10
```

WITH + WHERE = HAVING (SQL)

```
MATCH (n:Label)-[:REL]->(m:Label)
WHERE n.prop < 42
WITH n, count(m) as cnt,
      collect(m.attr) as attrs
WHERE cnt > 12
RETURN n.prop,
       extract(a2 in
                  filter(a1 in attrs
                                 WHERE a1 =~ "...-.*")
                  | substr(a2,4,size(a2)-1))
          AS ids
ORDER BY length(ids) DESC
SKIP 5 LIMIT 10
```

Collections

```
MATCH (n:Label)-[:REL]->(m:Label)
WHERE n.prop < 42
WITH n, count(m) as cnt,
    collect(m.attr) as attrs
WHERE cnt > 12
RETURN n.prop,
       extract(a2 in
           filter(a1 in attrs
                   WHERE a1 =~ "...-.*")
           | substr(a2,4,size(a2)-1))
       AS ids
ORDER BY length(ids) DESC
LIMIT 10
```

FOREACH | CREATE | MERGE

```
CREATE (y:Year {year:2014})
FOREACH (m IN range(1,12) |
  CREATE
    (:Month {month:m})-[:IN]->(y)
)
```

```
MERGE (y:Year {year:2014})
ON CREATE
  SET y.created = timestamp()
FOREACH (m IN range(1,12) |
  MERGE
    (:Month {month:m})-[:IN]->(y)
)
```

Prática - Neo4j

- www.ic.unicamp.br/~cmbm/MC536/
- > neo4j console
- //localhost:7474

Em resumo ...

- Novas tendências e demandas levaram a emergir novas frentes em BD
- NOSQL
- Vários modelos de dados - requisitos específicos
- BD Grafos: grafos como modelo de dados
- Neo4j
 - Grafo de Propriedades - tipado e rotulado
 - Linguagem de consulta Cypher

Dúvidas ?