Bancos de Dados de Grafos

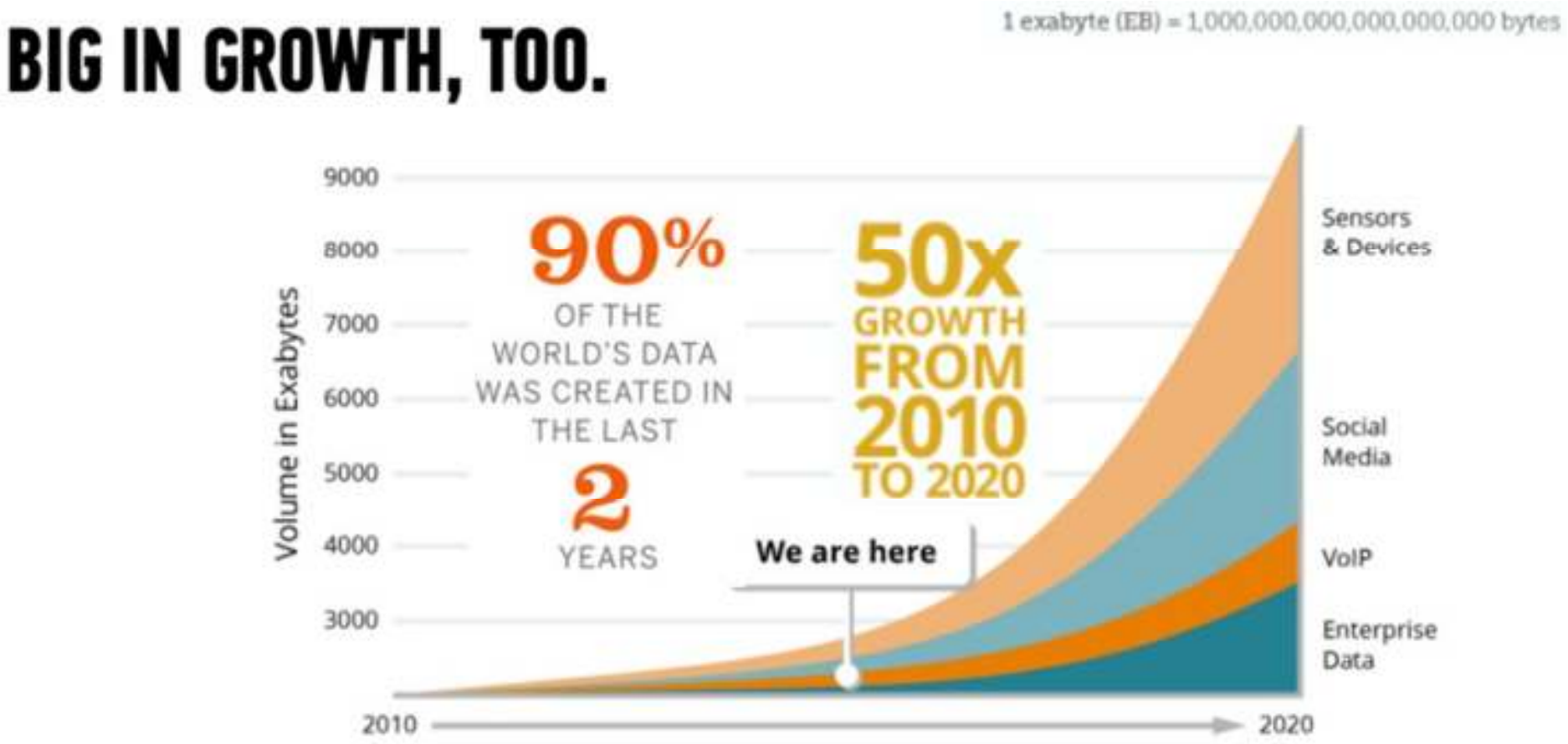
Jaudete Daltio

MC536

Motivações para novos SGBDs

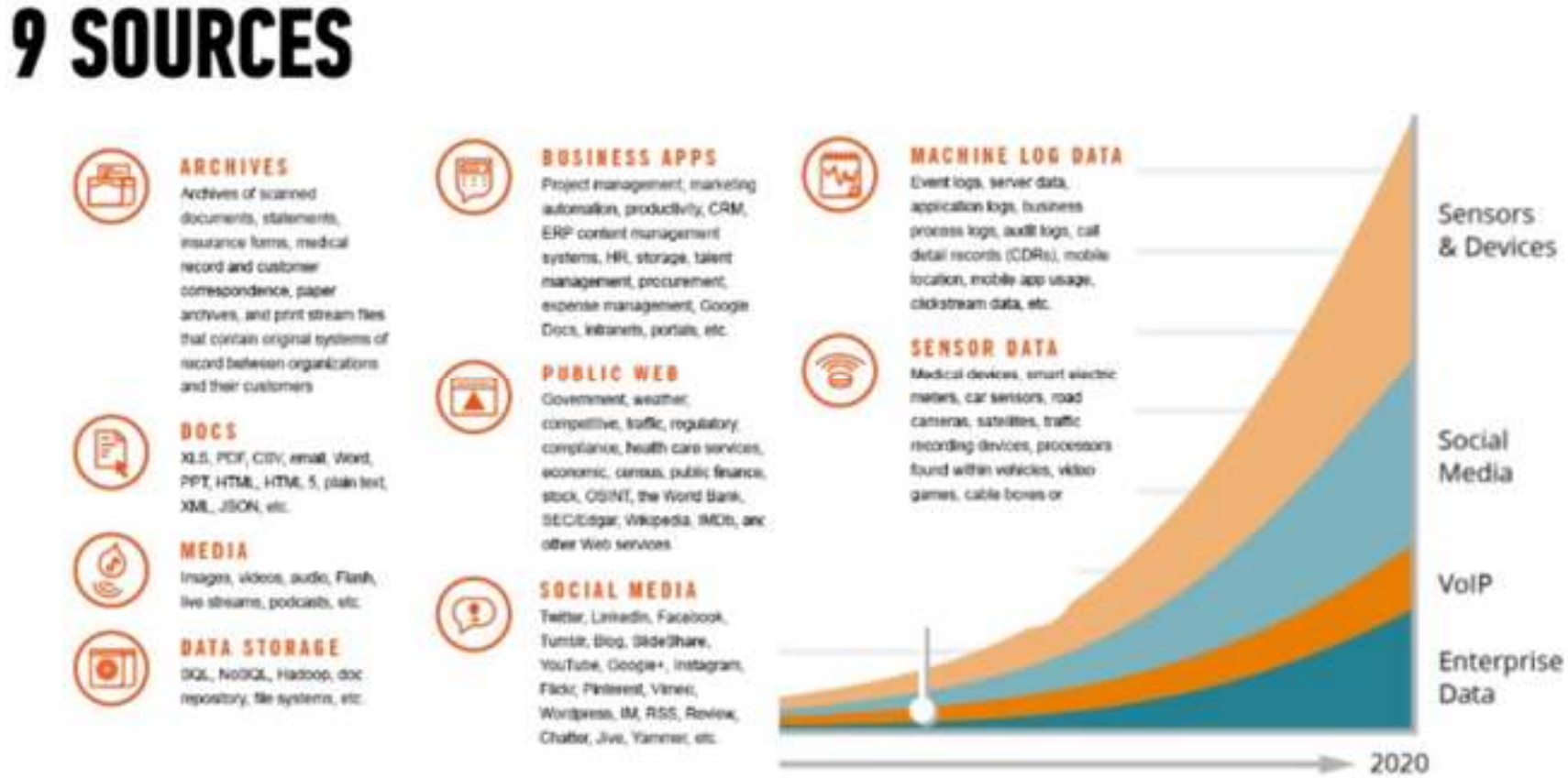
- Volume de dados crescent
- Distribuídos, heterogêneos e interligados
- Questões sobre armazenamento e processamento descentralizado, desempenho e semântica

Tendência (1): volume



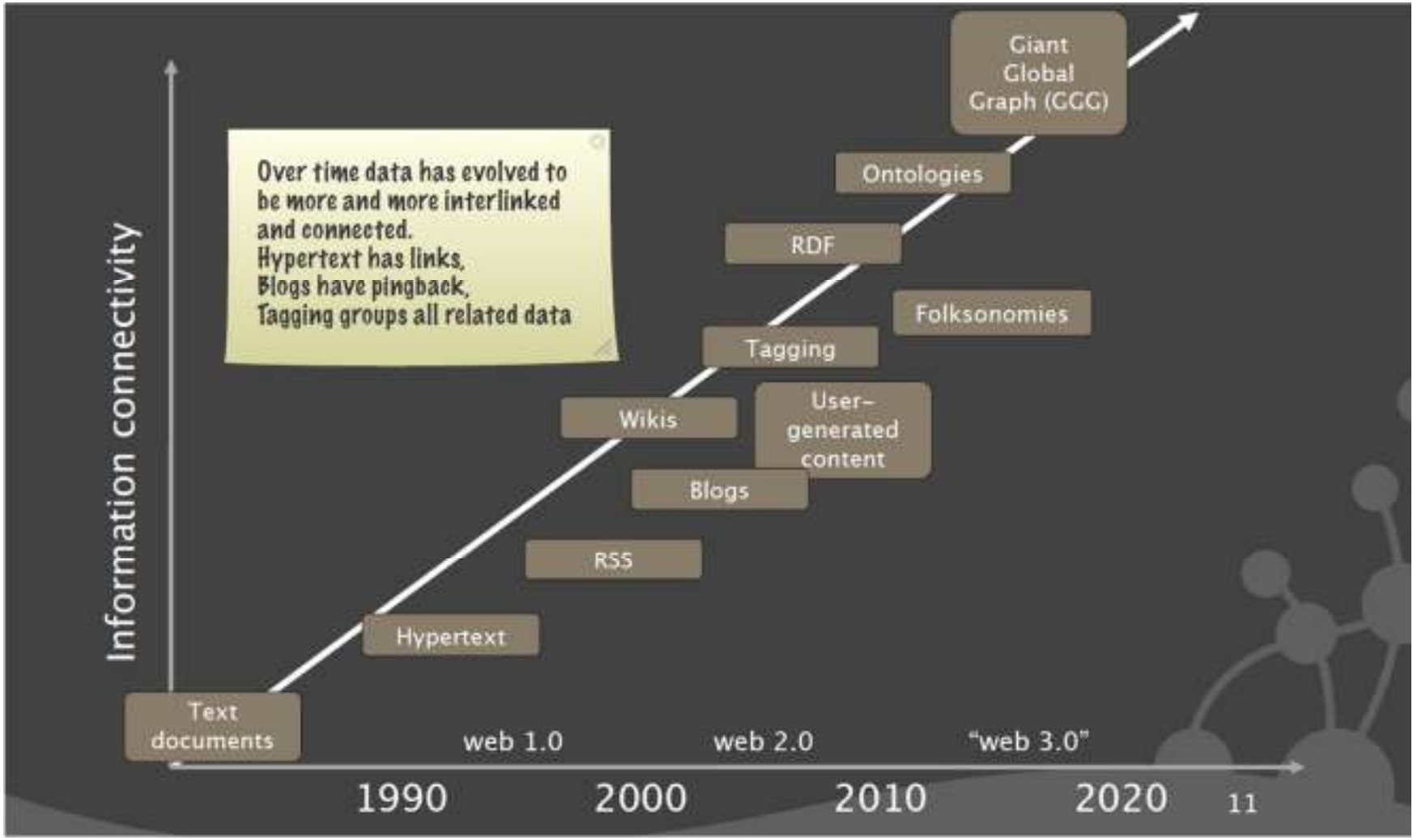
https://pt.slideshare.net/sfamilian/working-with-big-data-jan-2016-part-1/6-CONTEXT_WHATS_BIG_DATAHOW_BIG

Tendência (1): volume



https://pt.slideshare.net/sfamilian/working-with-big-data-jan-2016-part-1/6-CONTEXT_WHATS_BIG_DATAHOW_BIG

Tendência (2): conectividade



Tendência (3): heterogeneidade

NO APIs

Data that has no standard Web service and requires alternative methods of integration

SOME APIs

Data that has a standard Web service

STRUCTURED

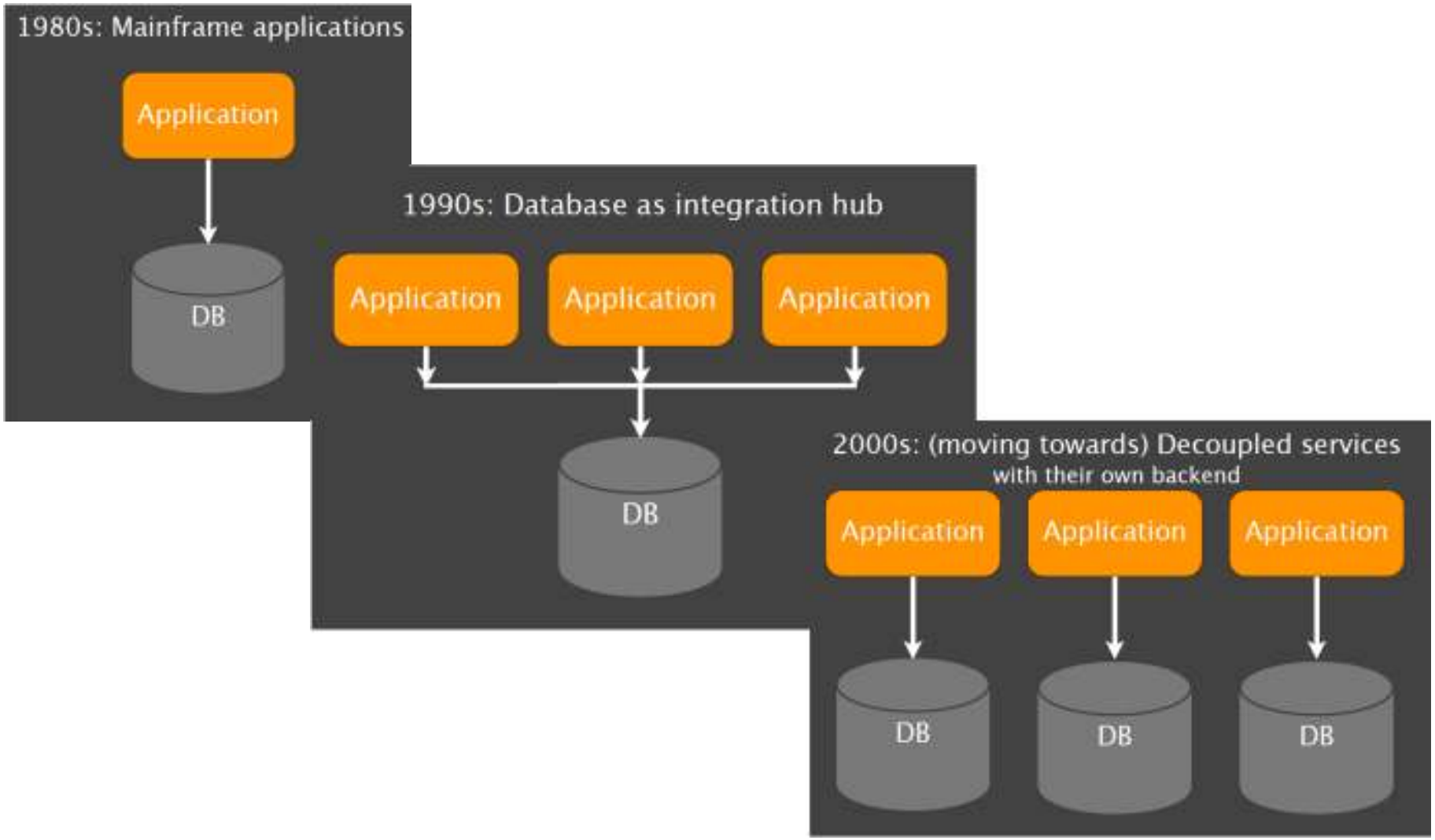
Data that resides in a fixed field within a record or file

UNSTRUCTURED

Data that does not have a pre-defined data model or is not organized in a pre-defined manner

- Individualization of content
  - In the salary lists of the 1970s, all elements had exactly one job
  - In the salary lists of the 2000s, we need 5 job columns! Or 8? Or 15?
- All encompassing "entire world views"
  - Store more data about each entity

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)

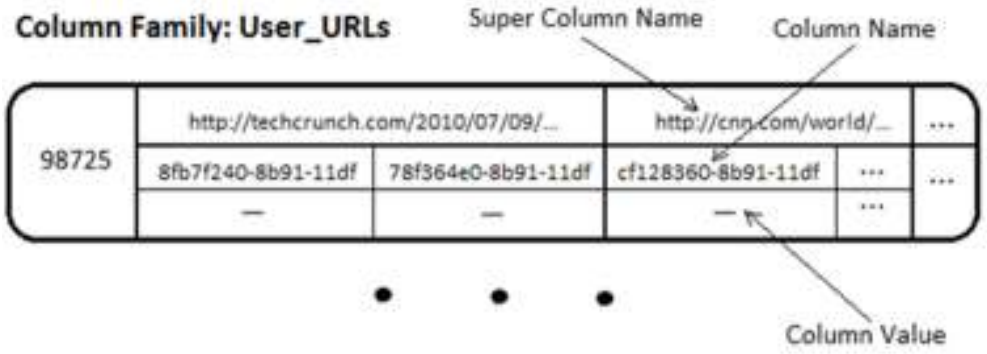
- Noção de esquema é fuzzy
  - Em um mesmo BD, duas instancias de uma mesma entidade podem ter atributos diferentes

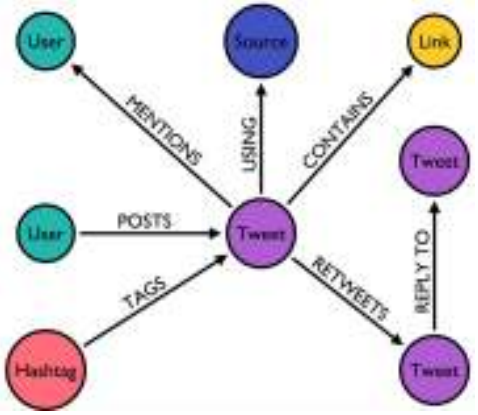
- Diferentes modelos de dados

Modelo de Dados (Codd, 1980)

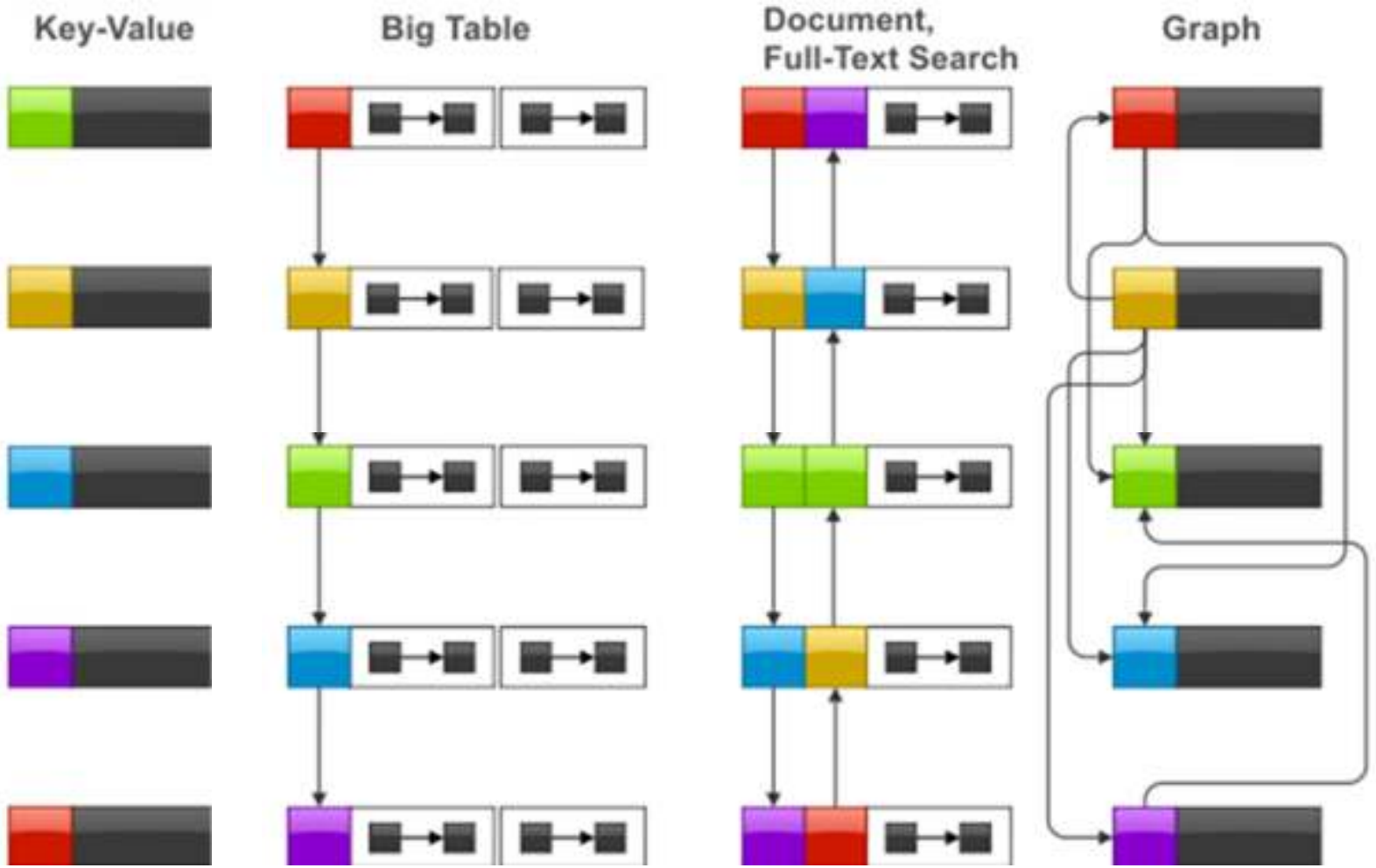
- Coleção de elementos para representar dados e expressar detalhes semânticos
- Componentes
  - Tipos de estruturas de dados
  - Restrições de integridade para definir estados consistentes do banco de dados
  - Operadores para recuperar dados

Categorias: Modelos de Dados

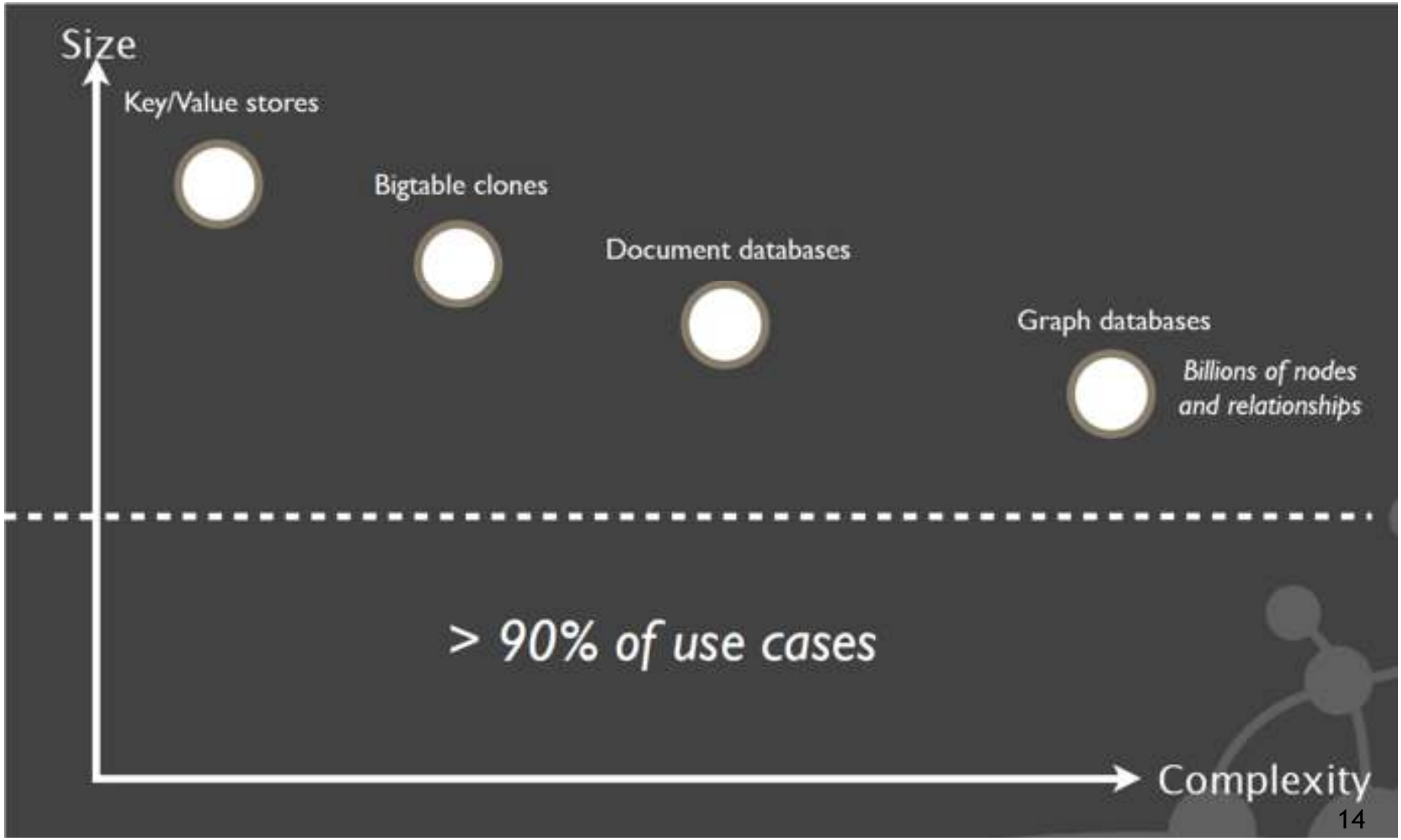




NOSQL Databases

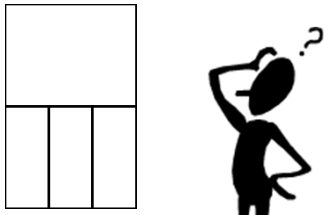


Tamanho X Complexidade



Lápide

- ER
- Modelagem
- Mapeamento
- Especificação
- Normalização

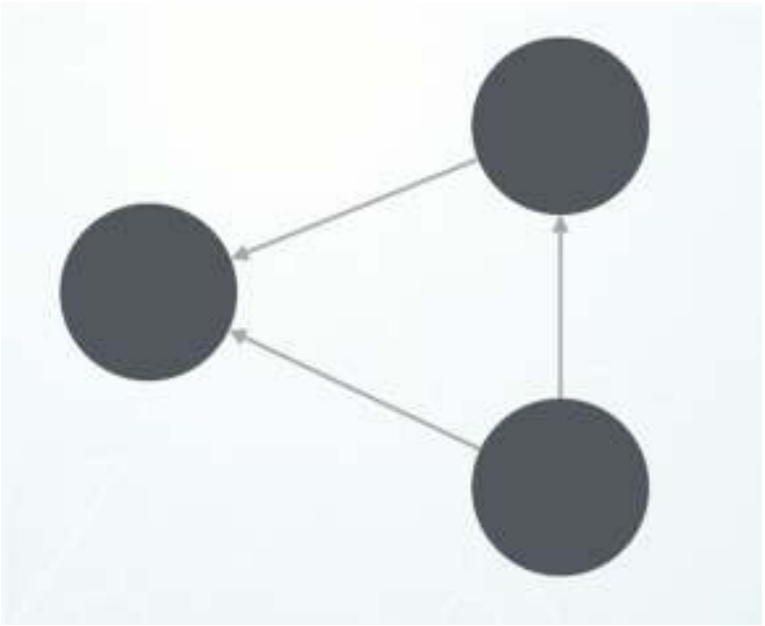


Bancos de Dados de Grafos

O que é um BD grafos ?

- BD NOSQL
- Utiliza **modelos de dados** baseados em grafos

Um grafo é

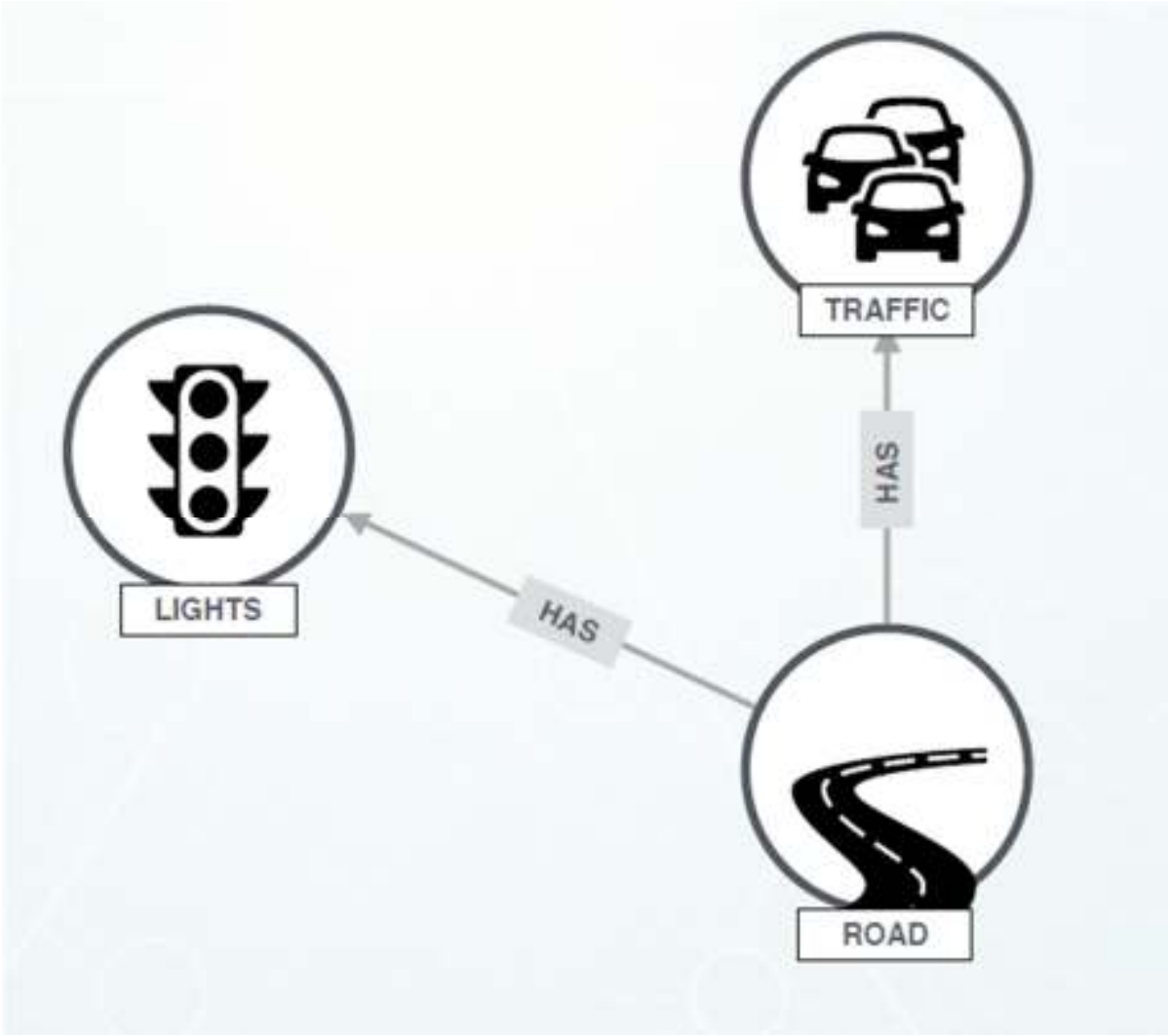


- um conjunto V de vértices
  (vértice = nó)
- um conjunto E de arestas: pares ordenados v e w ∈ V
  (aresta = arco)

Por que grafos ?

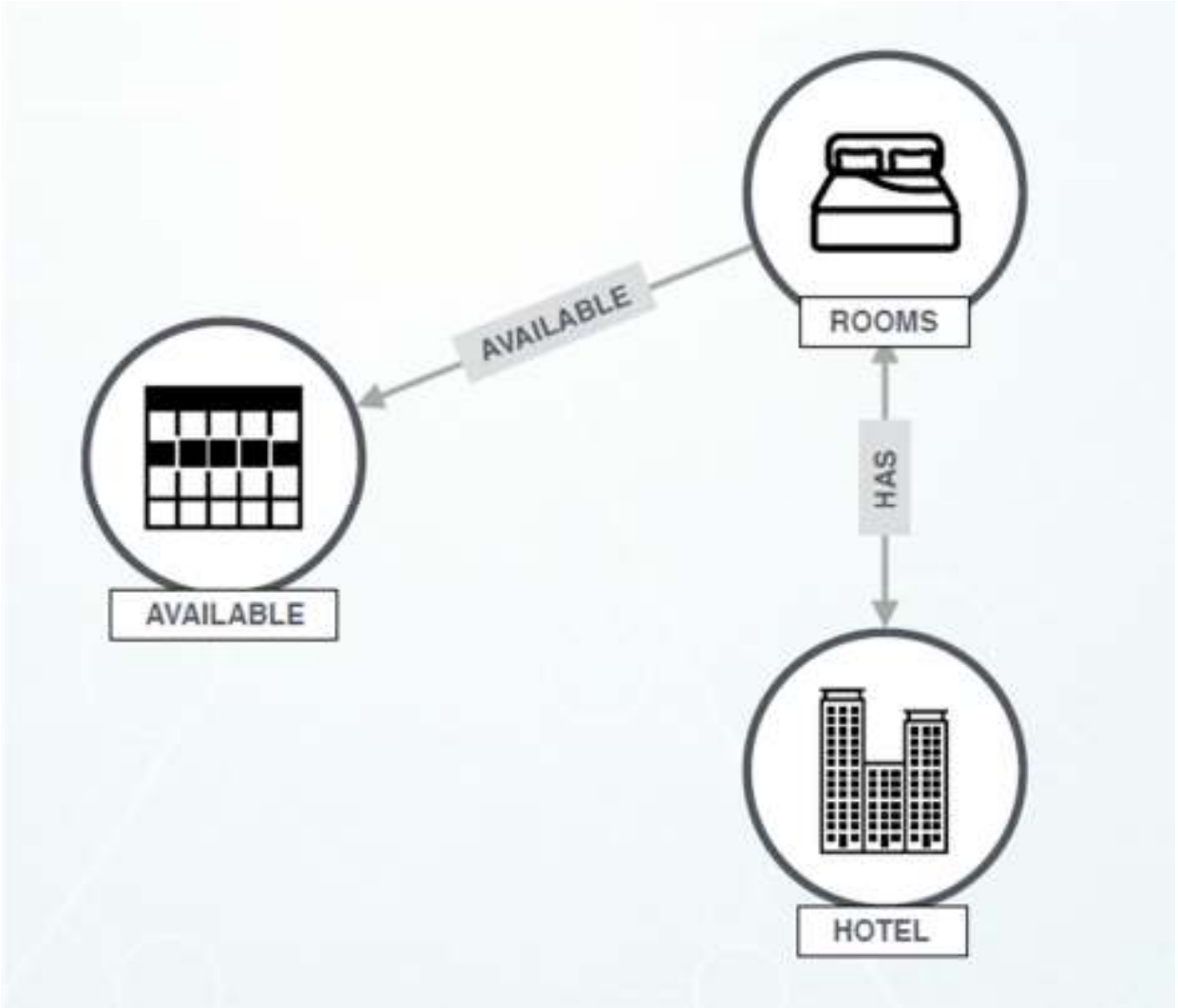
- Vários cenários e problemas do mundo real podem ser mapeados como grafos

Tráfego



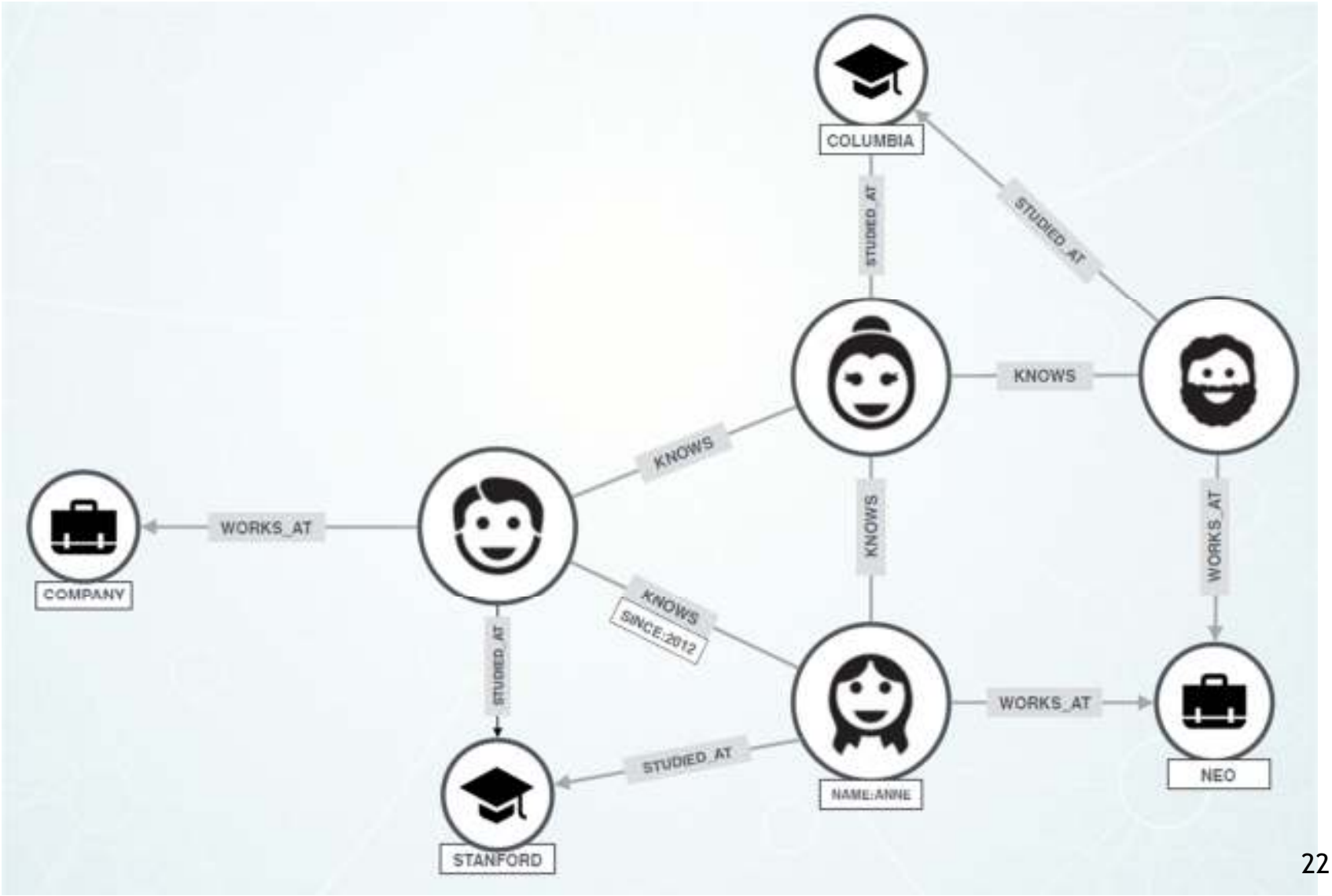
Fonte: slides Neo4j GraphDays, 2016

Hotéis



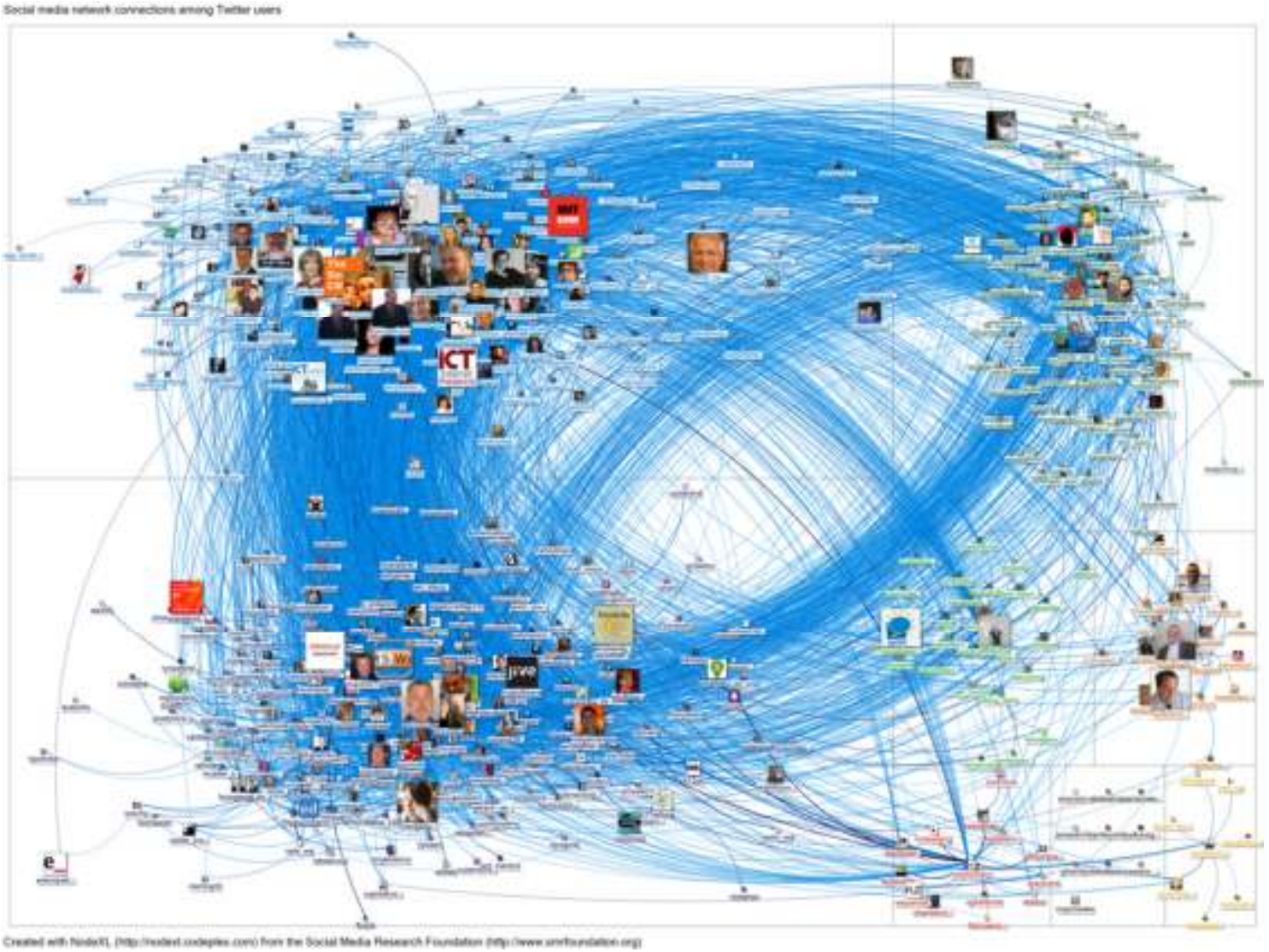
Fonte: slides Neo4j GraphDays, 2016

Relacionamentos

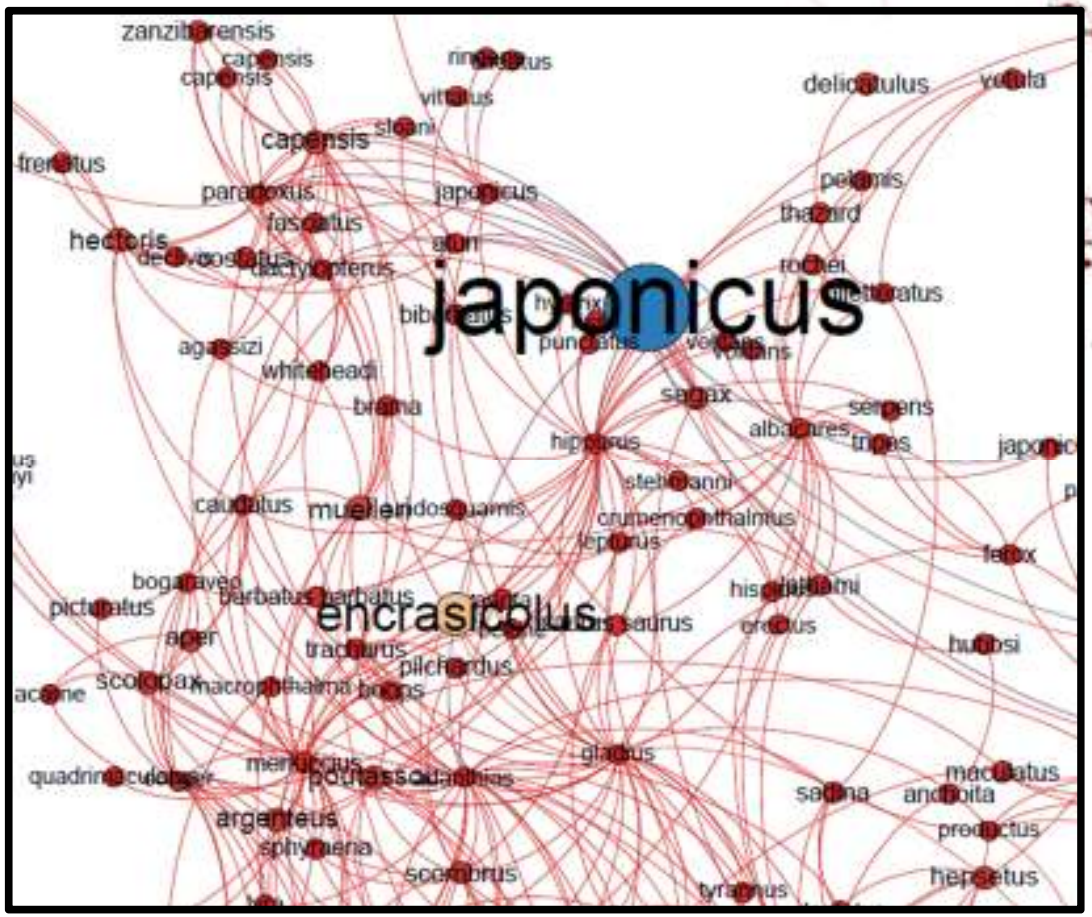


Fonte: slides Neo4j GraphDays, 2016

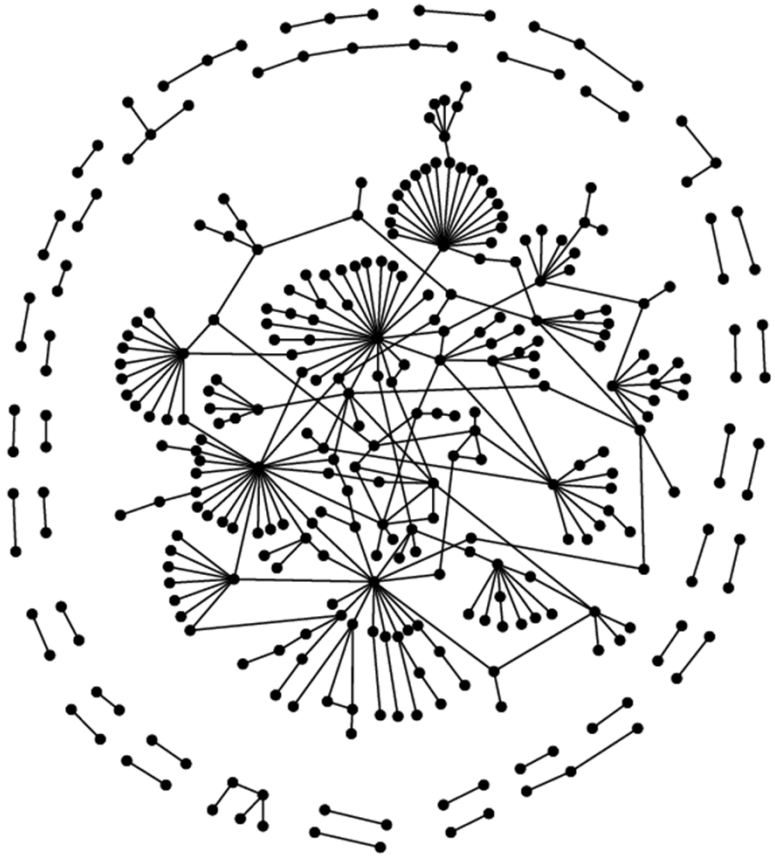
Twitter (Influência)



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

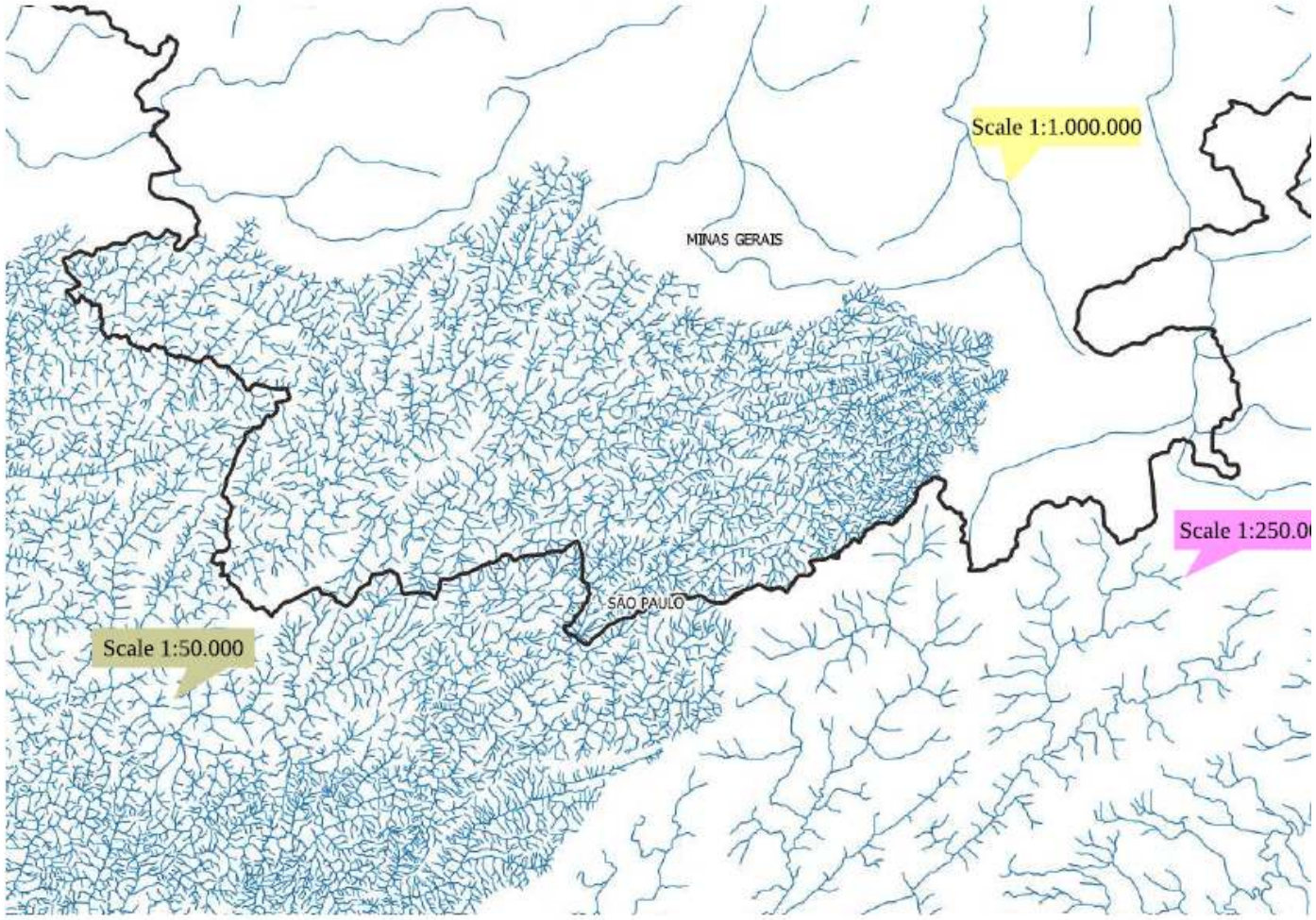
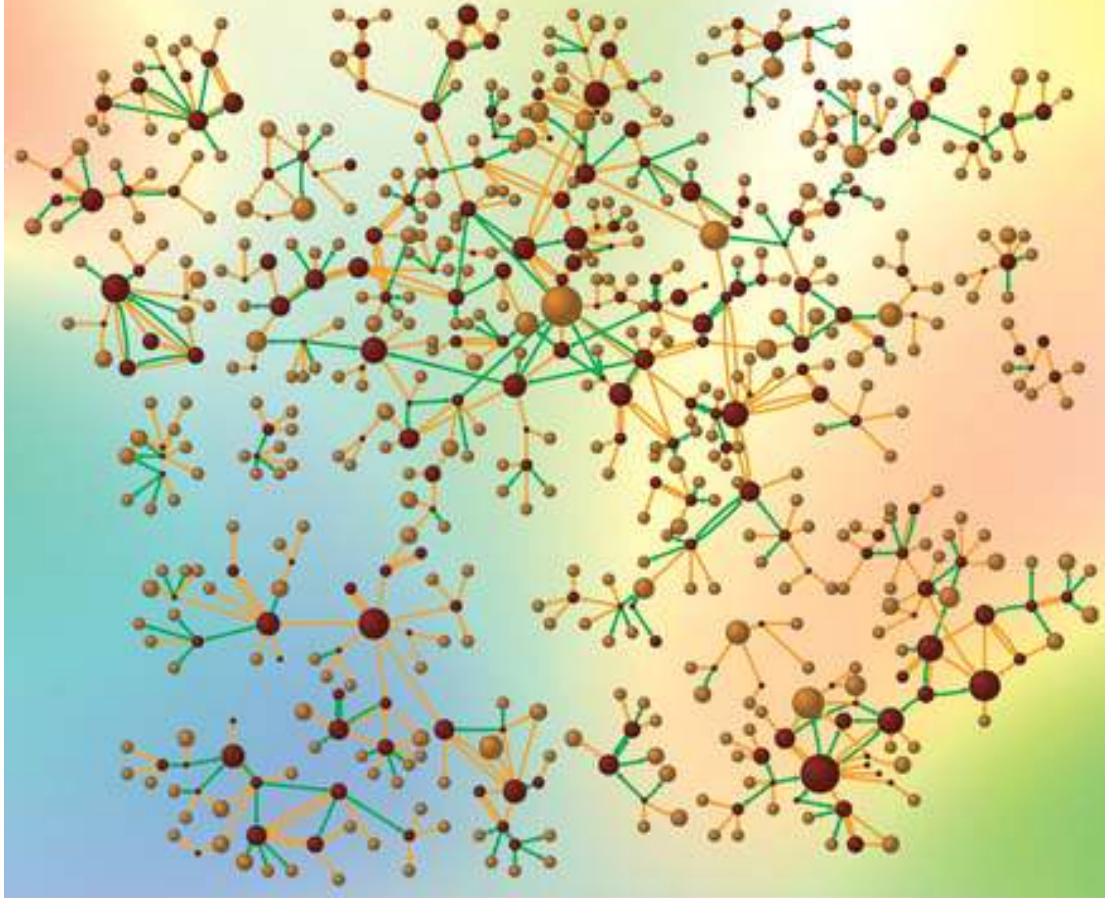


Fig. 2 Different Drainage Stretch Scales in Drainage Network

NETWORK SCIENCE



NATIONAL RESEARCH COUNCIL
OF THE NATIONAL ACADEMIES

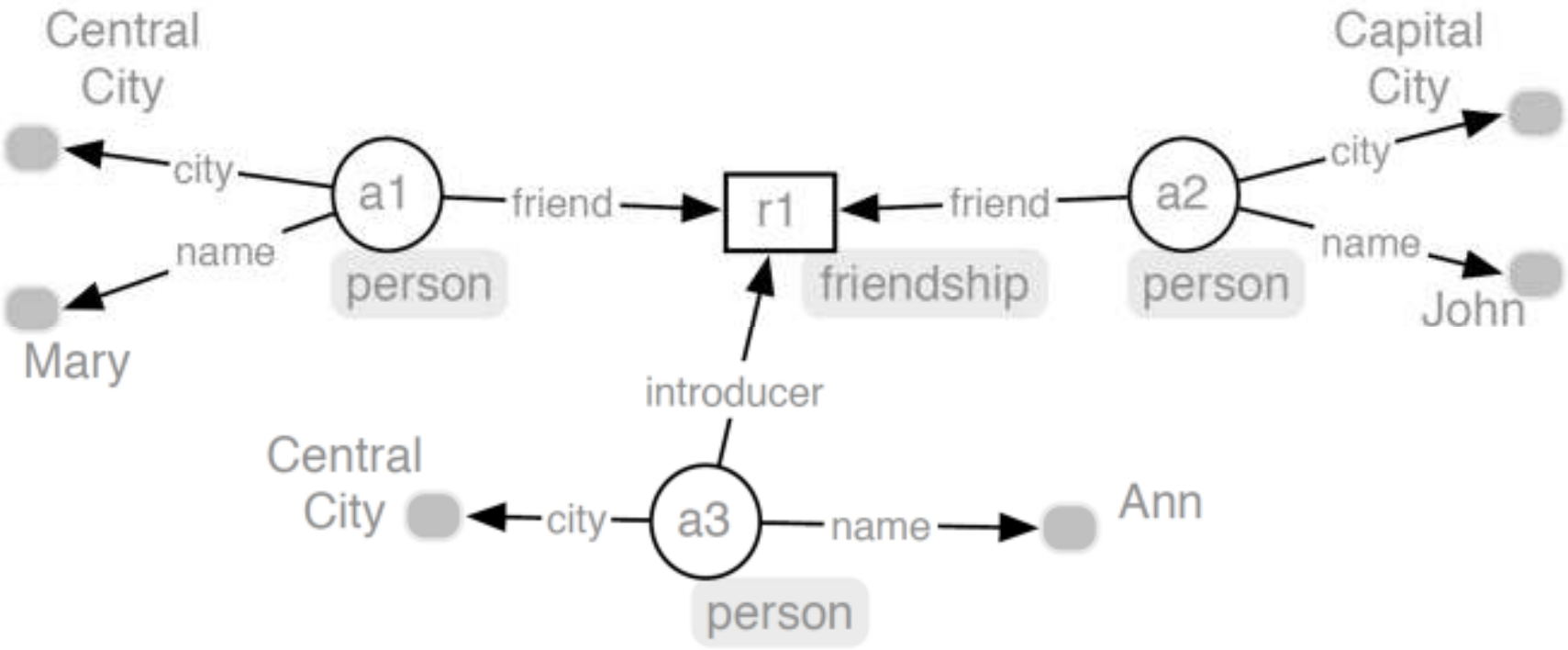
Modelo de Dados (Codd, 1980)

- 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

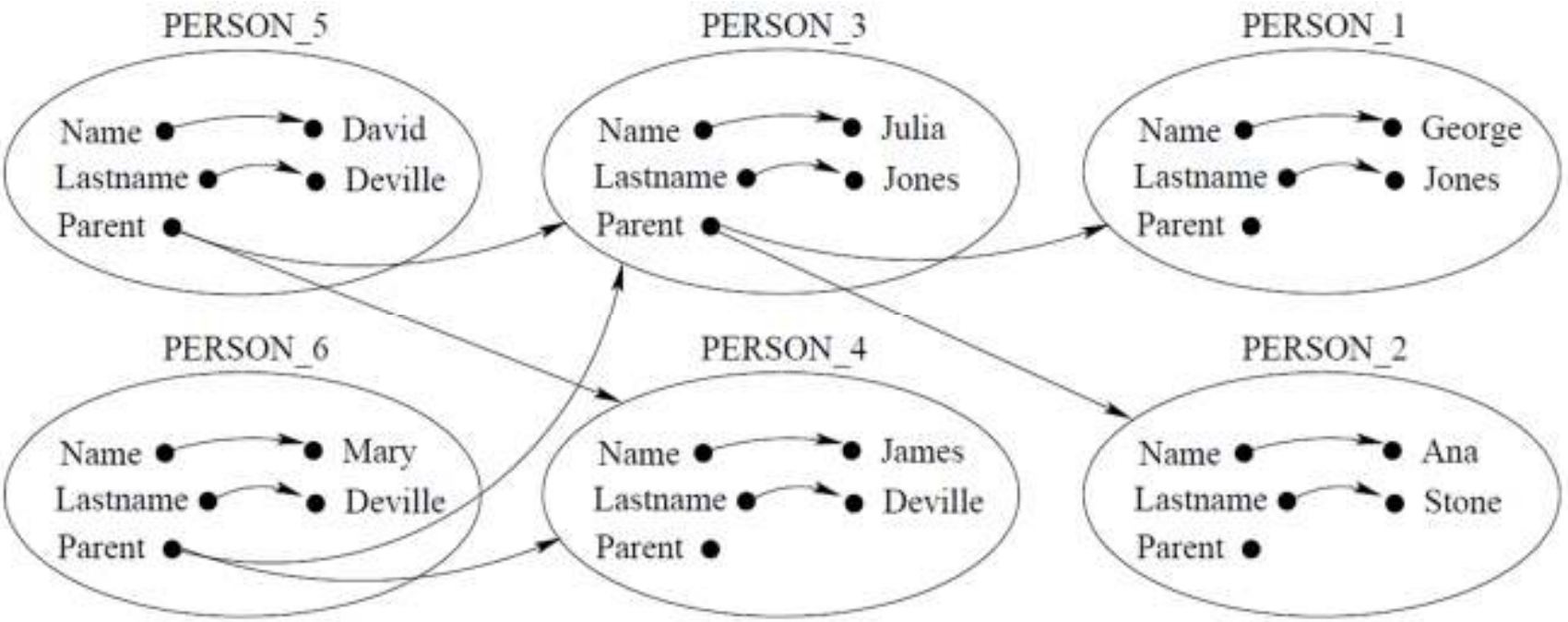
Estrutura de Dados

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

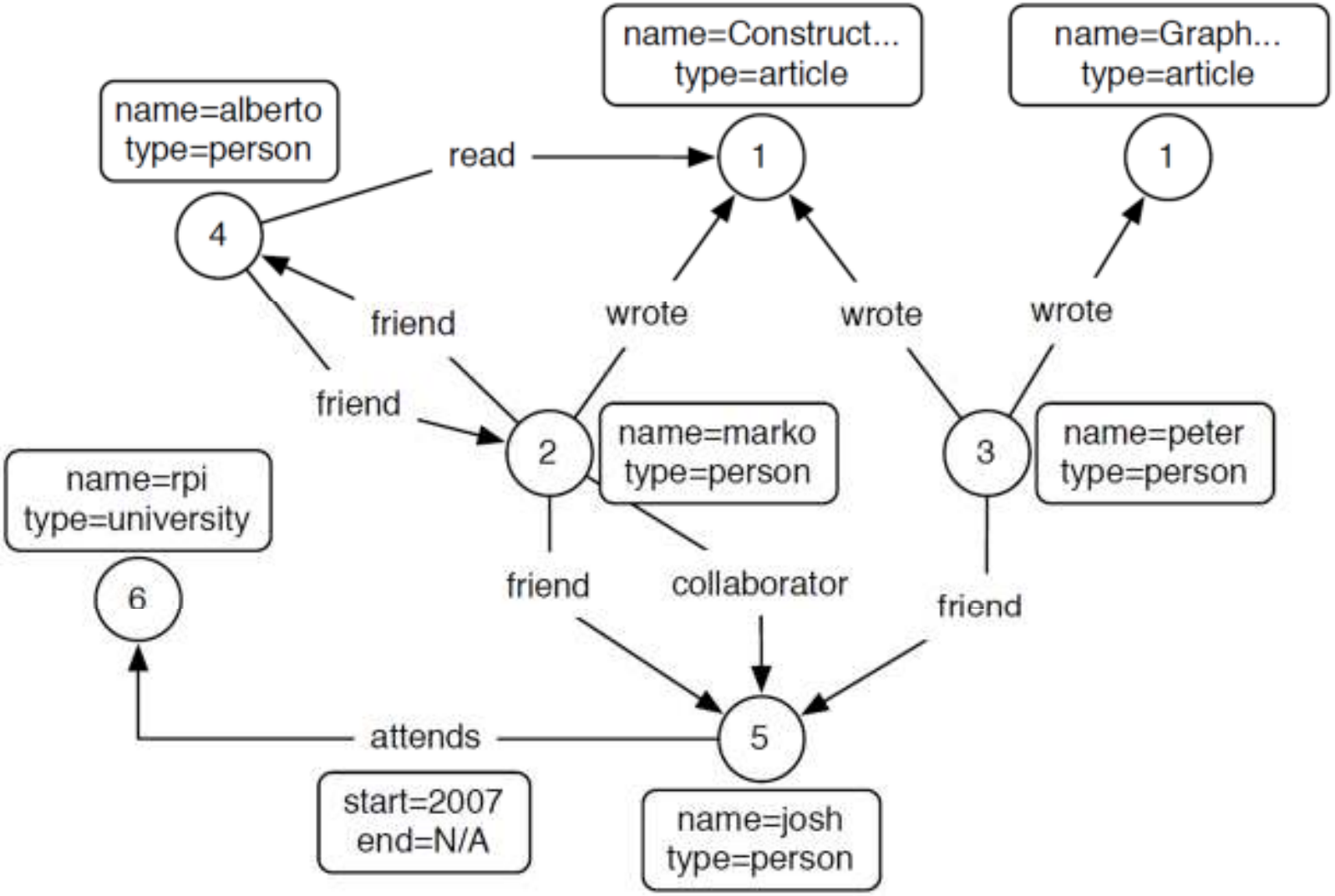
Grafo RDF


Central City
city
a1
friend
r1
friend
a2
Capital City
city
name
name
person
friendship
person
Mary
John
introducer
Central City
city
a3
name
Ann
person

Hipergrafo


PERSON_5
Name
David
Lastname
Deville
Parent
PERSON_3
Name
Julia
Lastname
Jones
Parent
PERSON_1
Name
George
Lastname
Jones
Parent
PERSON_6
Name
Mary
Lastname
Deville
Parent
PERSON_4
Name
James
Lastname
Deville
Parent
PERSON_2
Name
Ana
Lastname
Stone
Parent

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 May 2018	Rank Apr 2018	Rank May 2017	DBMS	Database Model	Score May 2018	Score Apr 2018	Score 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

342 sistemas

30 de grafos

Neo4J – 22

Giraph - 124

DB Grafos

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

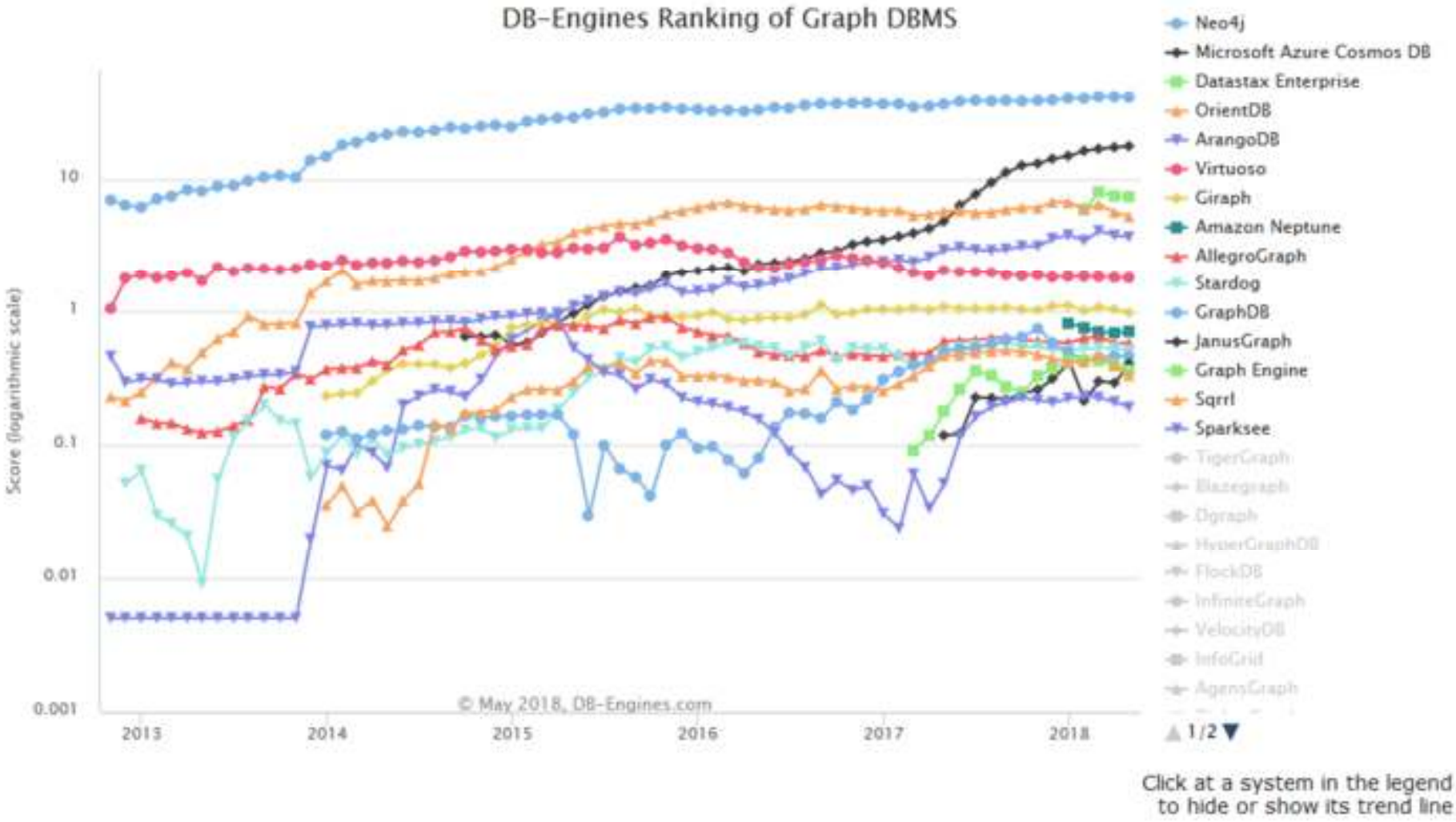
30 systems in ranking, May 2018

Rank May 2018	Rank Apr 2018	Rank May 2017	DBMS	Database Model	Score May 2018	Score Apr 2018	Score 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	-0.06	-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

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

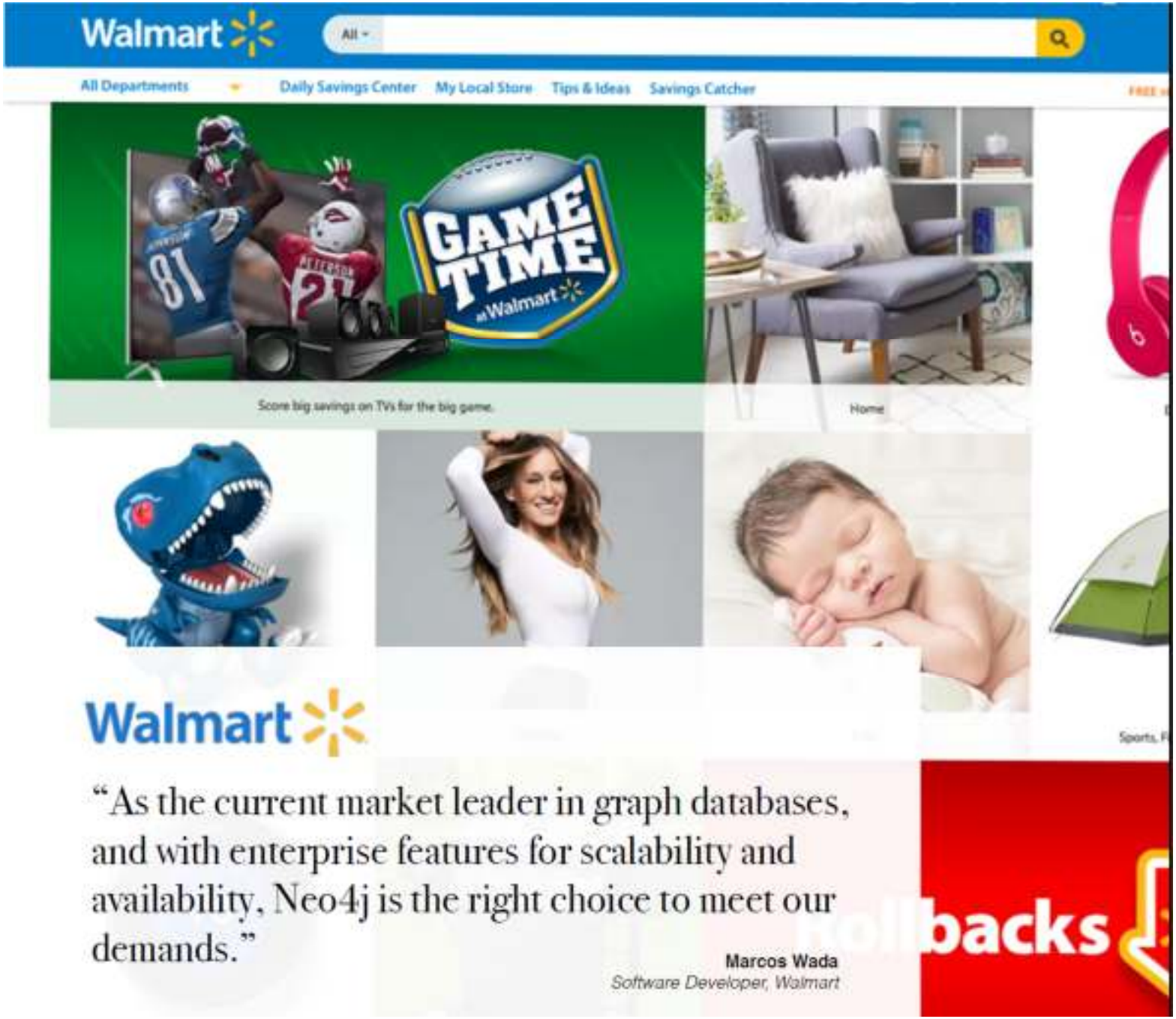
Fonte: slides Michael Hunger, Relational to Graph, 2015

Empresas que adotam

Fonte: slides Michael Hunger, Relational to Graph, 2015

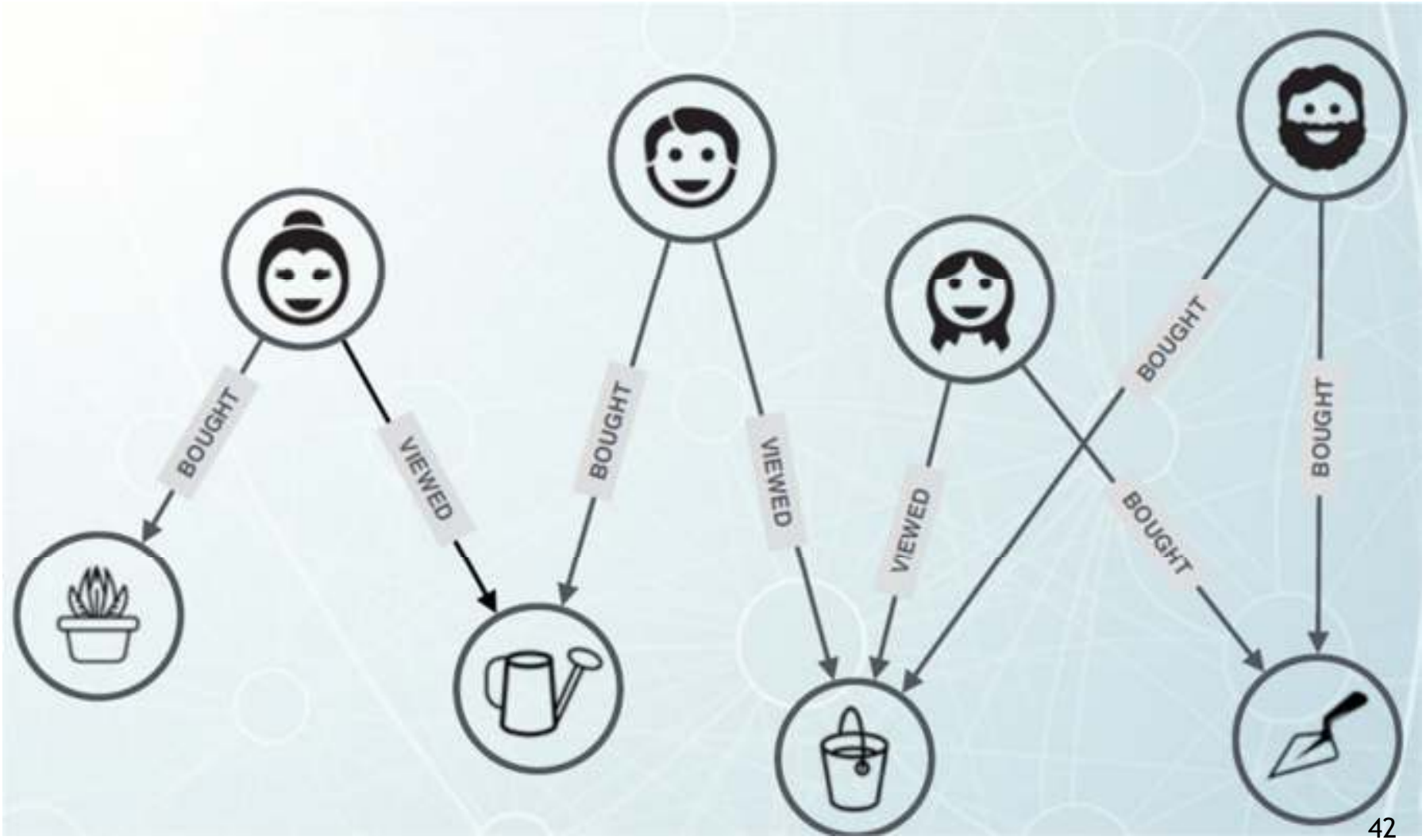
Como adotam

Fonte: slides Michael Hunger, Relational to Graph, 2015

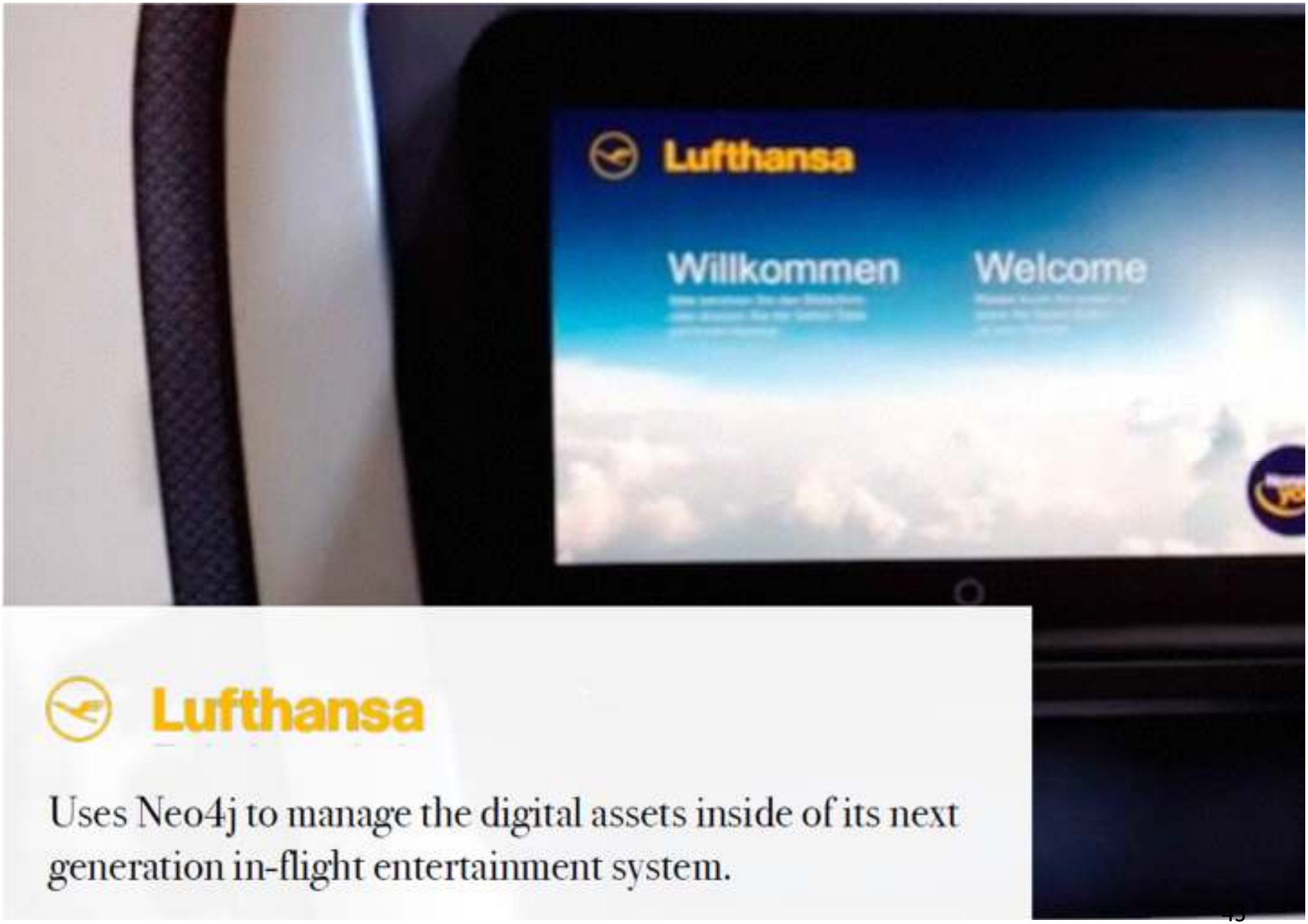


Fonte: slides Neo4j GraphDays, 2016

Estudos de Caso - Recomendação

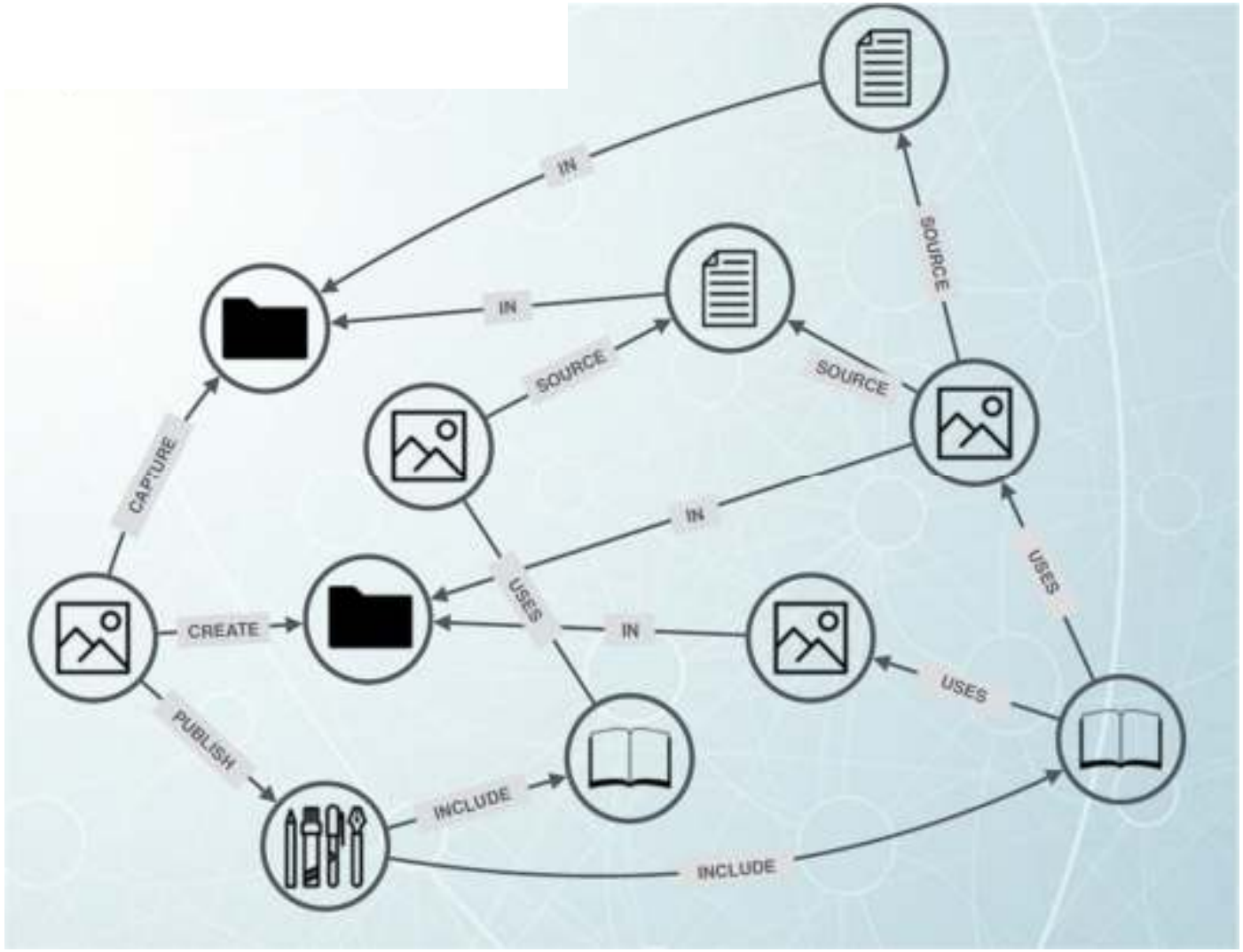


Fonte: slides Neo4j GraphDays, 2016



Fonte: slides Neo4j GraphDays, 2016

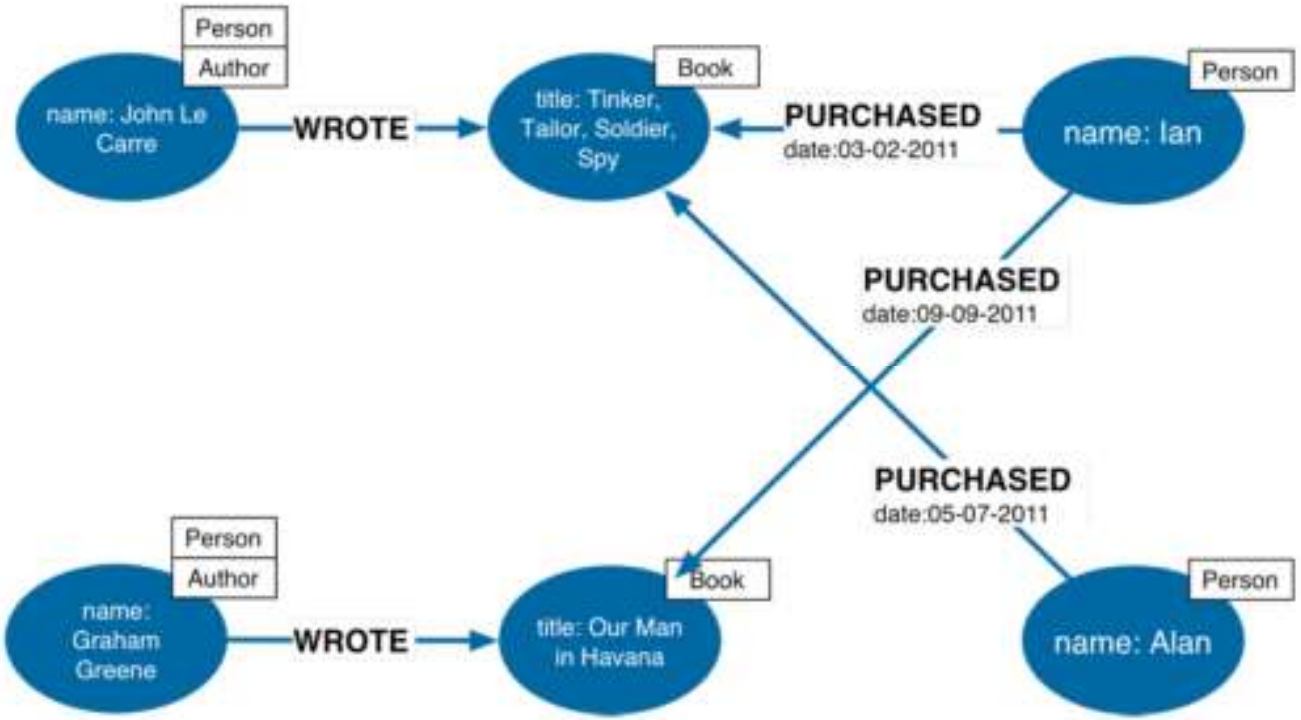
Estudos de Caso - Buscas



Fonte: slides Neo4j GraphDays, 2016

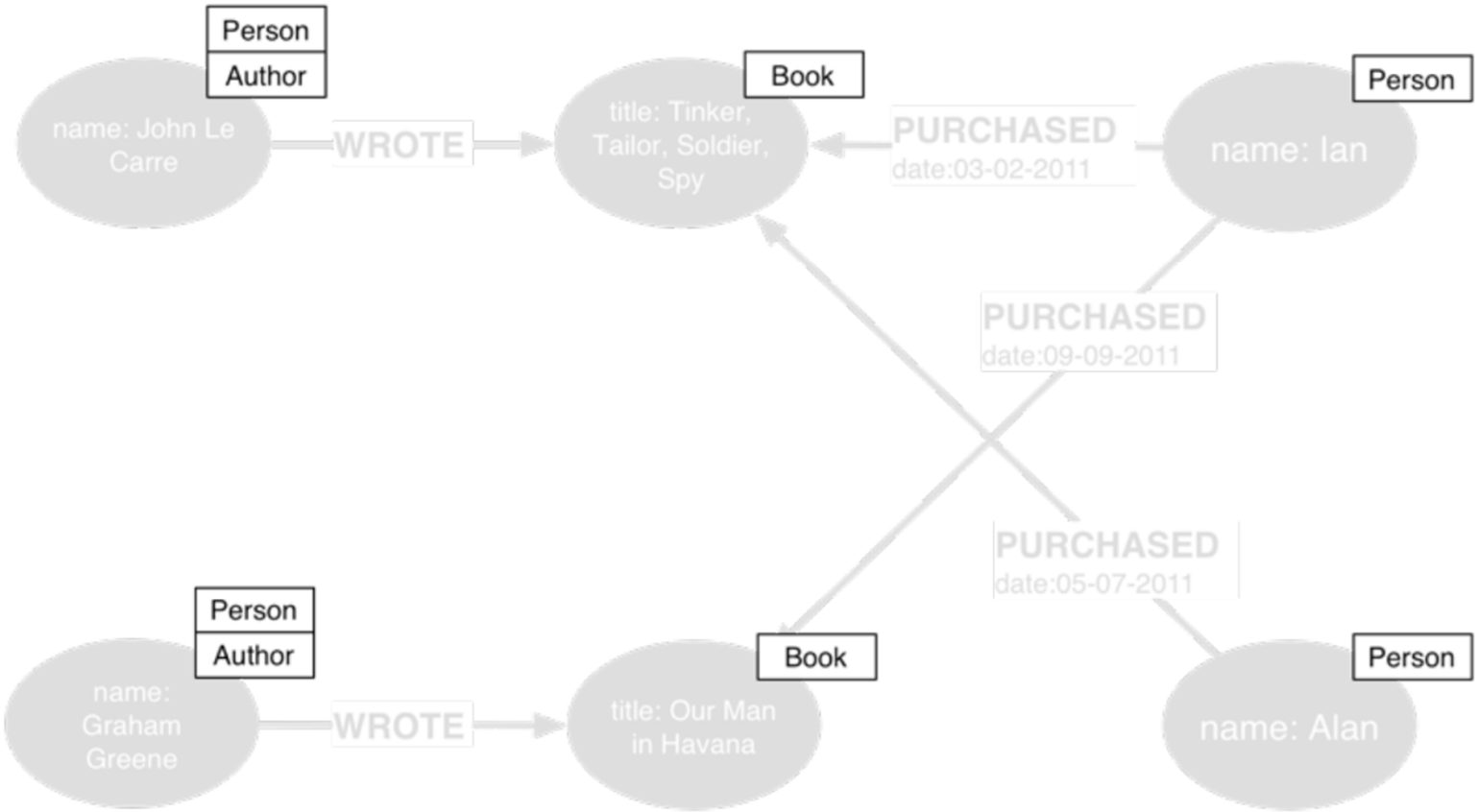
Conceitos basicos – modelo de dados do Neo4j

Grafo de Propriedades



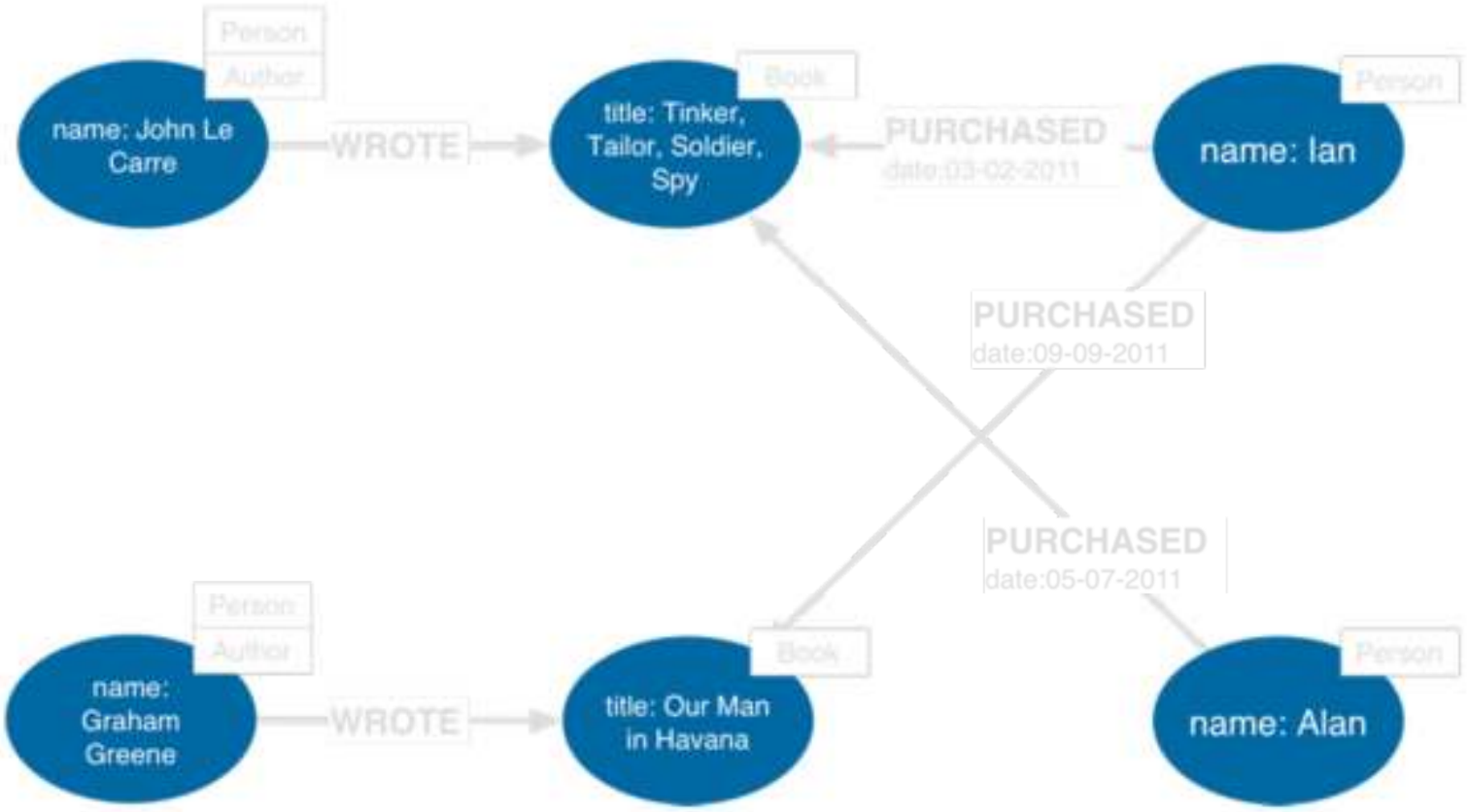
Fonte: slides Michael Hunger Graph Database Introduction, 2014

Rótulos (Labels)



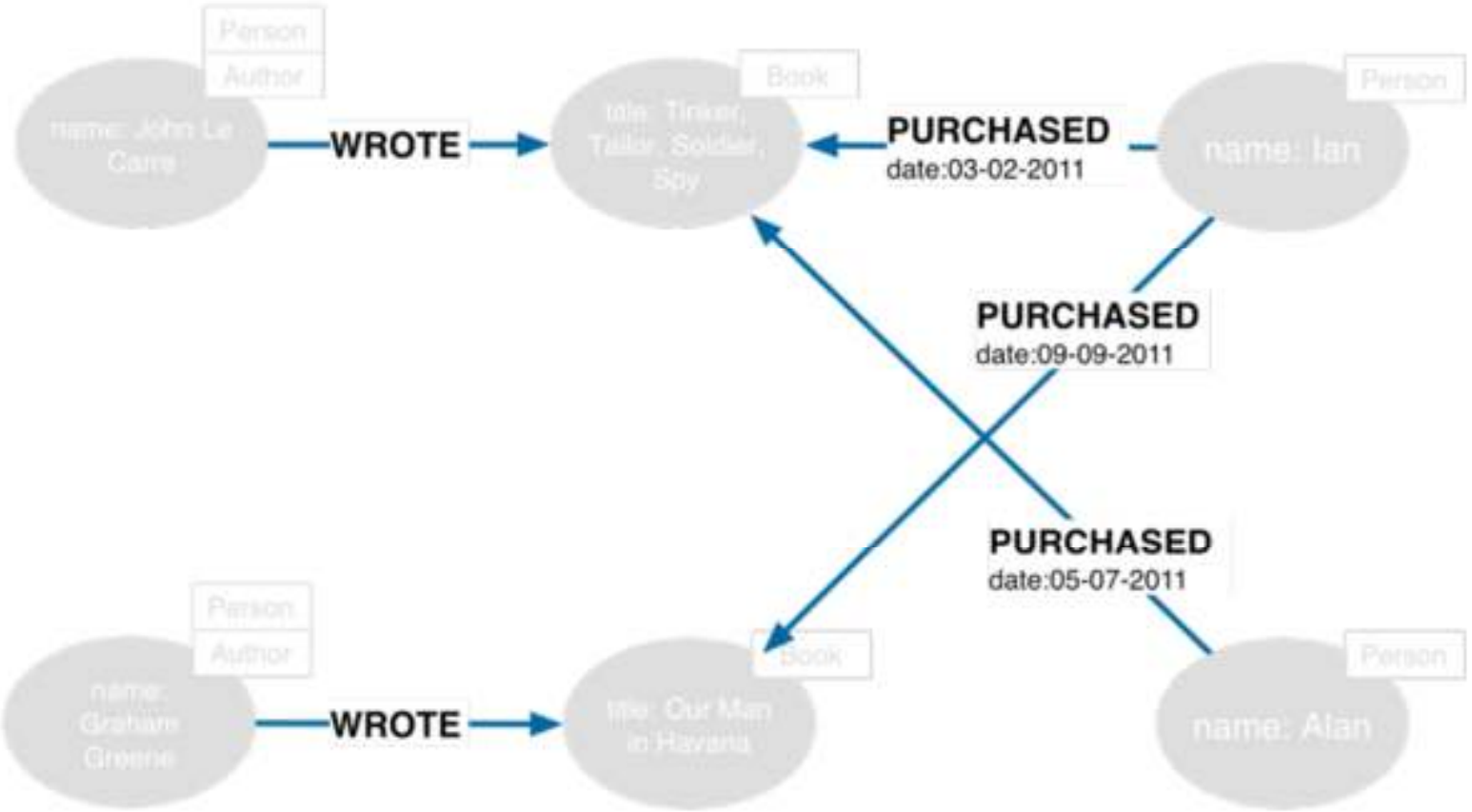
Fonte: slides Michael Hunger Graph Database Introduction, 2014

Atributos



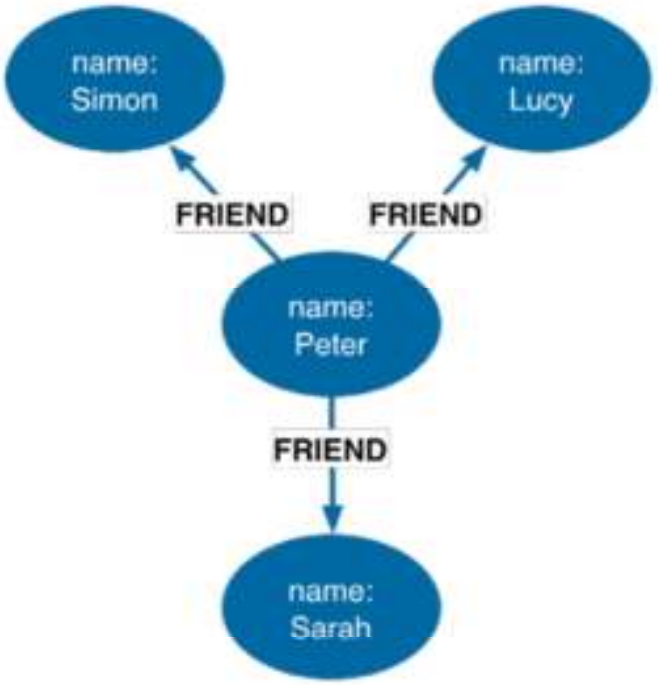
Fonte: slides Michael Hunger Graph Database Introduction, 2014

Arestas

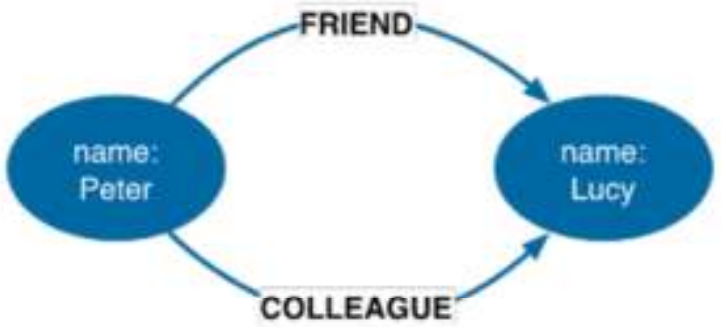


Fonte: slides Michael Hunger Graph Database Introduction, 2014

Tipos de Arestas



Nodes can have more than one relationship

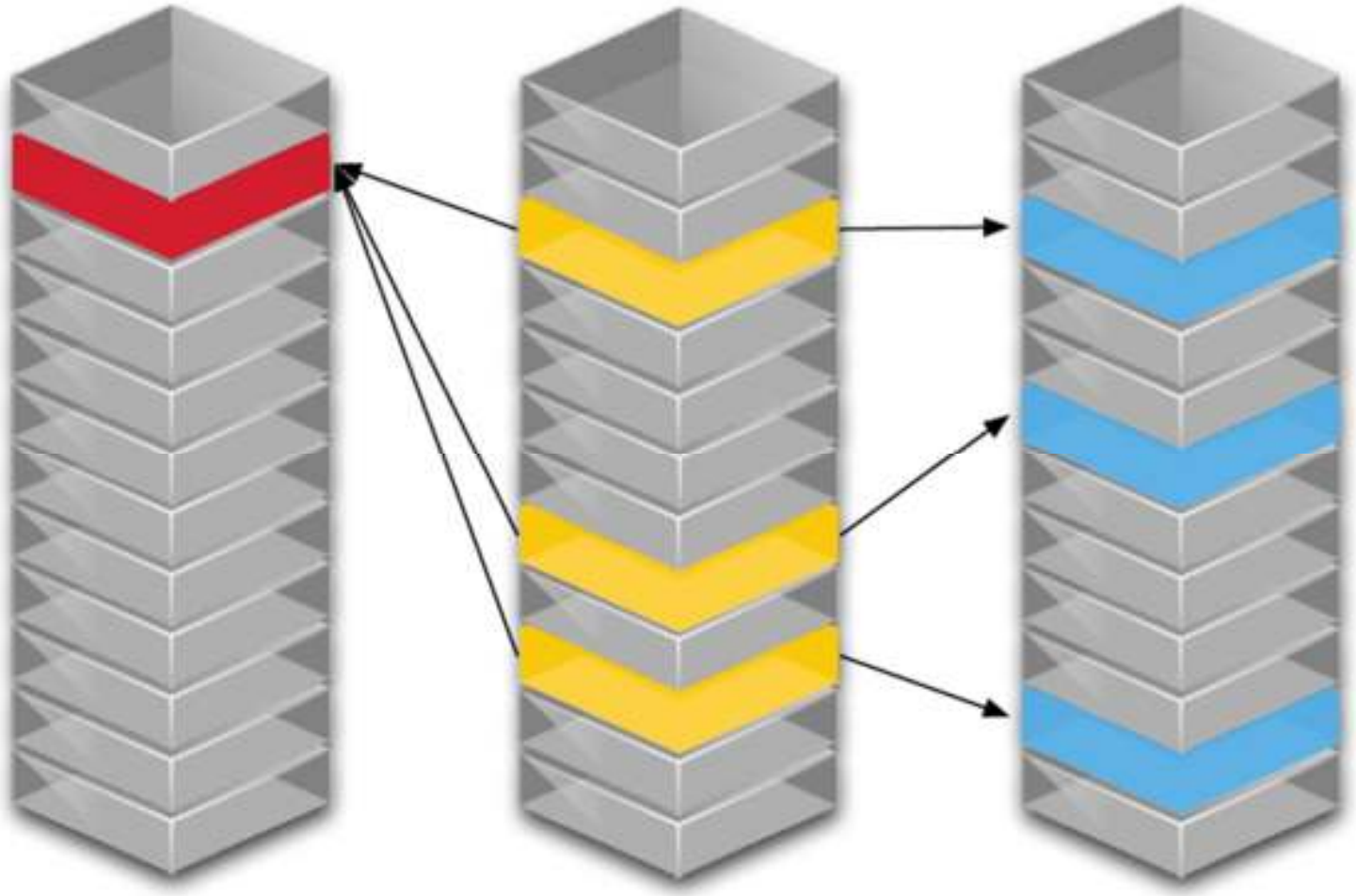


Nodes can be connected by more than one relationship

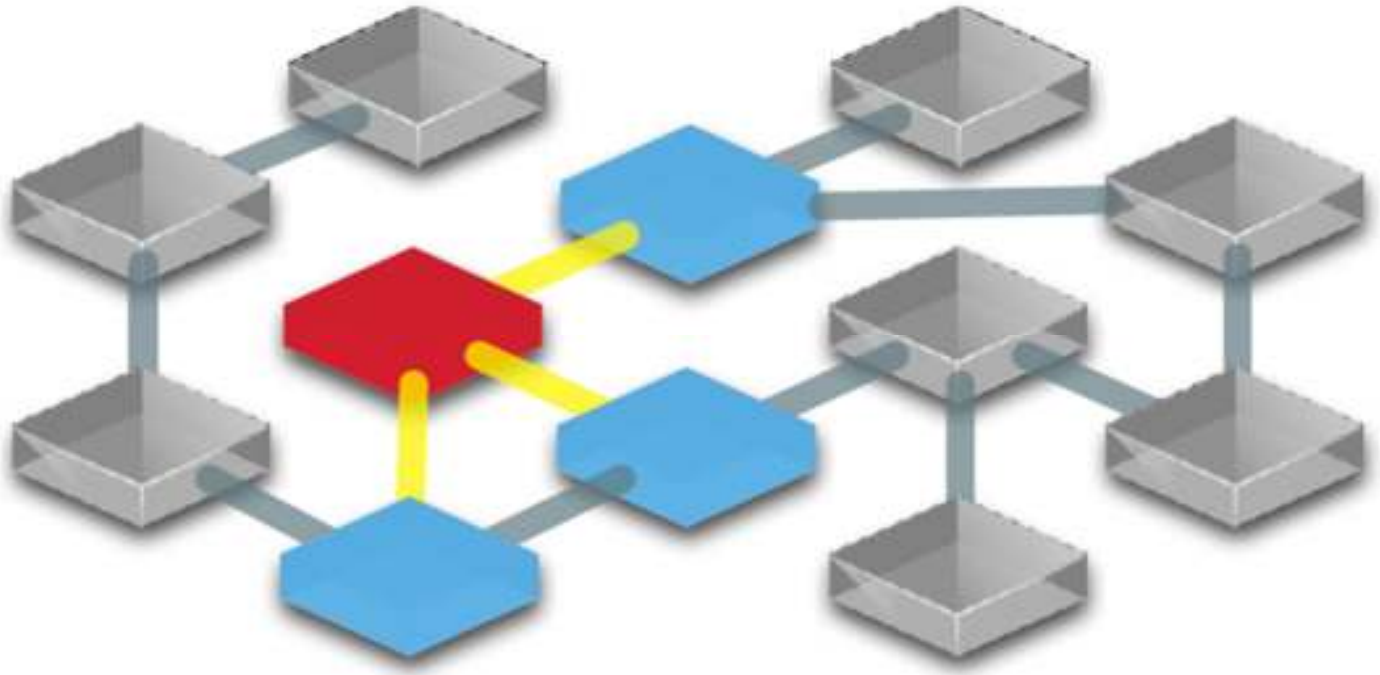
Self relationships are allowed

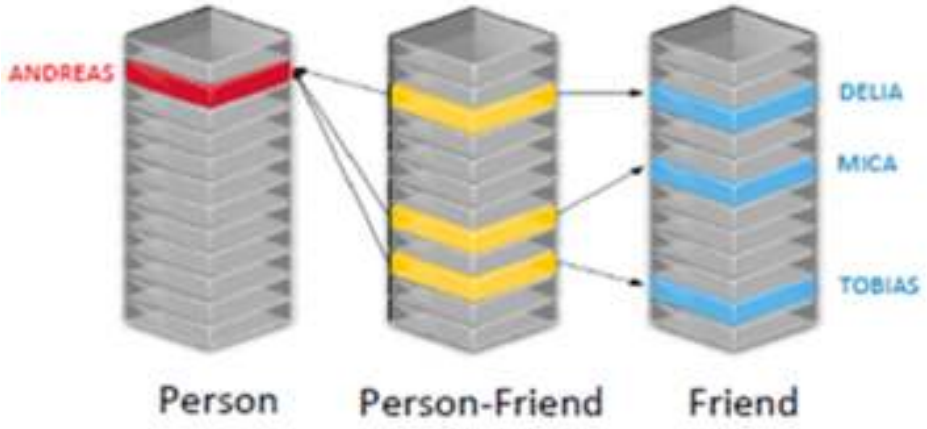
Fonte: slides Michael Hunger Graph Database Introduction, 2014

Mapeamento dos Relacionamentos



Mapeamento dos Relacionamentos





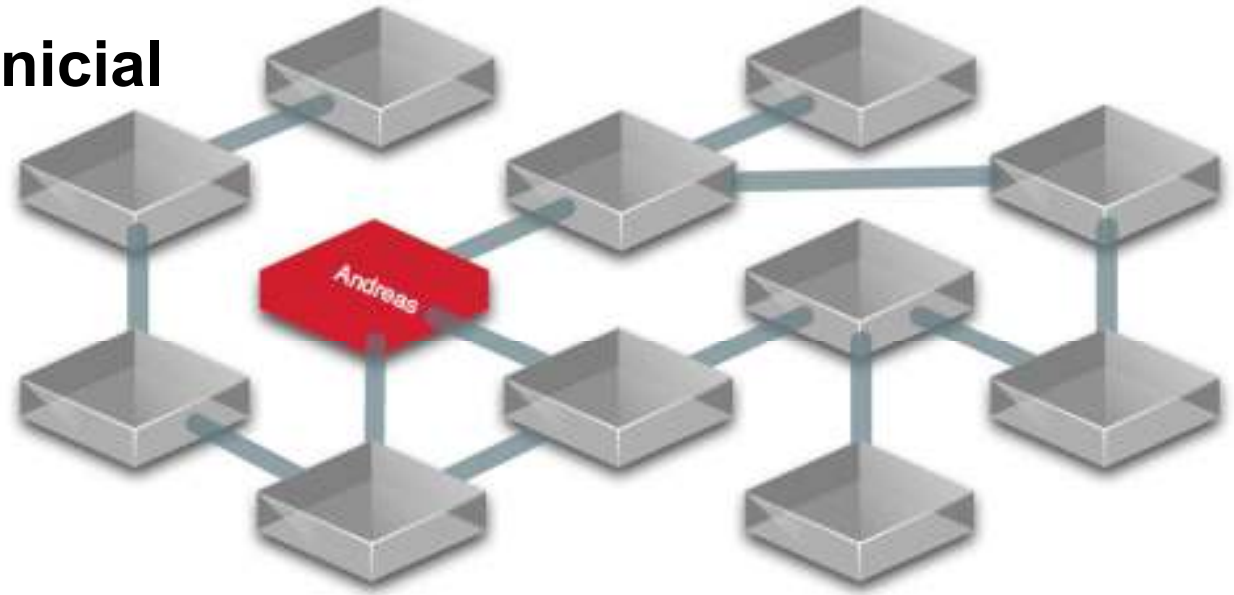
Cypher

- Declarativo
- Busca Padrões

Consultas em Grafos – "Percurso" Em Neo4j - Cypher

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

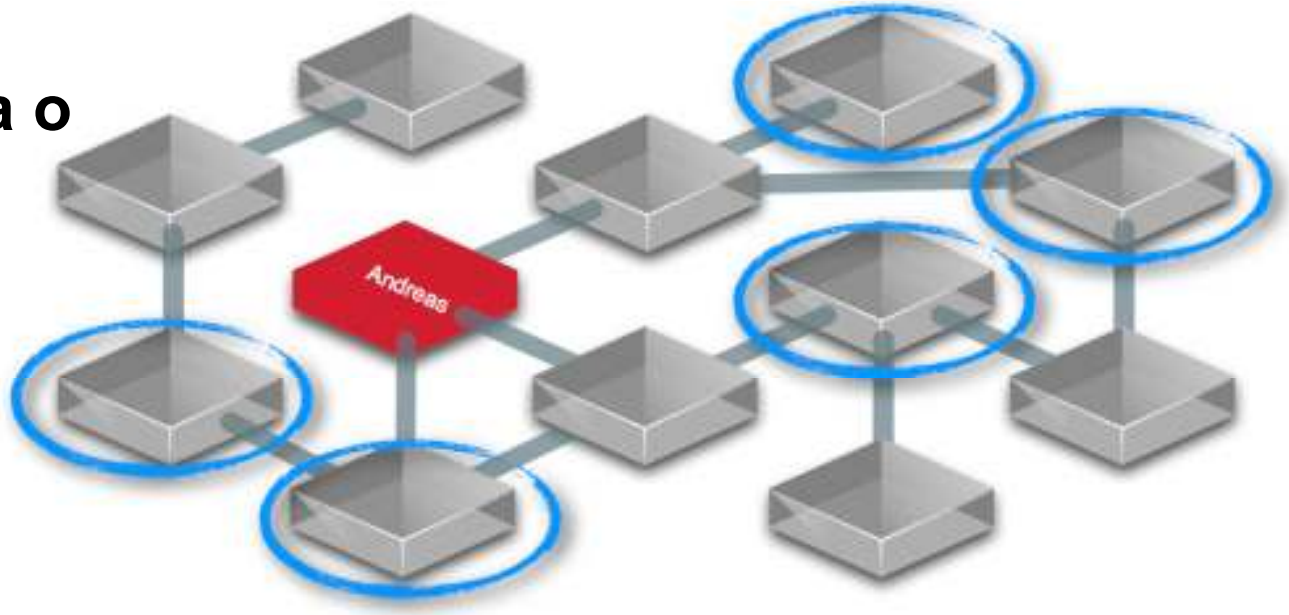
1. Ache nó inicial



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 ?

Fonte: slides Michael Hunger, Relational to Graph, 2015

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)
```


Fonte: slides Michael Hunger, Relational to Graph, 2015

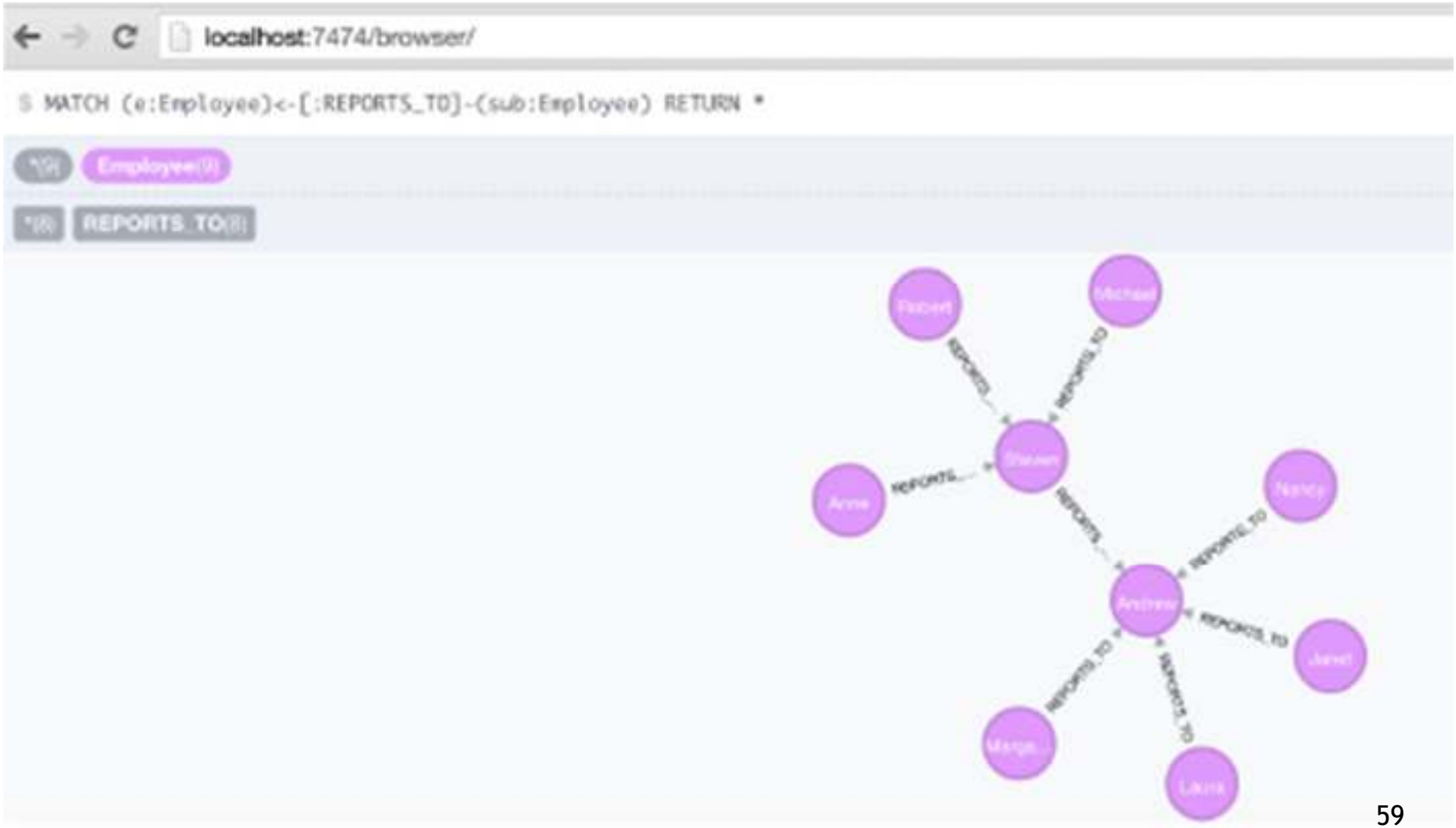
Consulta

```
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
  *
```


Fonte: slides Michael Hunger, Relational to Graph, 2015

Quem se reporta a quem ?



Fonte: slides Michael Hunger, Relational to Graph, 2015

Quem se reporta a quem ?

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

Graph

Rows

managerID	managerName	employeeID	employeeName
2	Andrew	5	Steven
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

Fonte: slides Michael Hunger, Relational to Graph, 2015

Pessoas a quem o Robert se reporta

```
MATCH
 path=(e:Employee)<-[:REPORTS_TO*]-(sub:Employee)
WHERE
   sub.firstName = 'Robert'
RETURN
   path;
```

Fonte: slides Michael Hunger, Relational to Graph, 2015

Quem é o chefe ?

```
MATCH
  (e:Employee)
WHERE
  NOT (e)-[:REPORTS_TO]->()
RETURN
   e.firstName as bigBoss;
```


Fonte: slides Michael Hunger, Relational to Graph, 2015

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
```


Fonte: slides Michael Hunger Graph Database Introduction, 2014

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
```


Fonte: slides Michael Hunger Graph Database Introduction, 2014

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
```


Fonte: slides Michael Hunger Graph Database Introduction, 2014

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
```


Fonte: slides Michael Hunger Graph Database Introduction, 2014

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
```


Fonte: slides Michael Hunger Graph Database Introduction, 2014

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
```


Fonte: slides Michael Hunger Graph Database Introduction, 2014

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
```


Fonte: slides Michael Hunger Graph Database Introduction, 2014

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)
)
```


Fonte: slides Michael Hunger Graph Database Introduction, 2014

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 ?