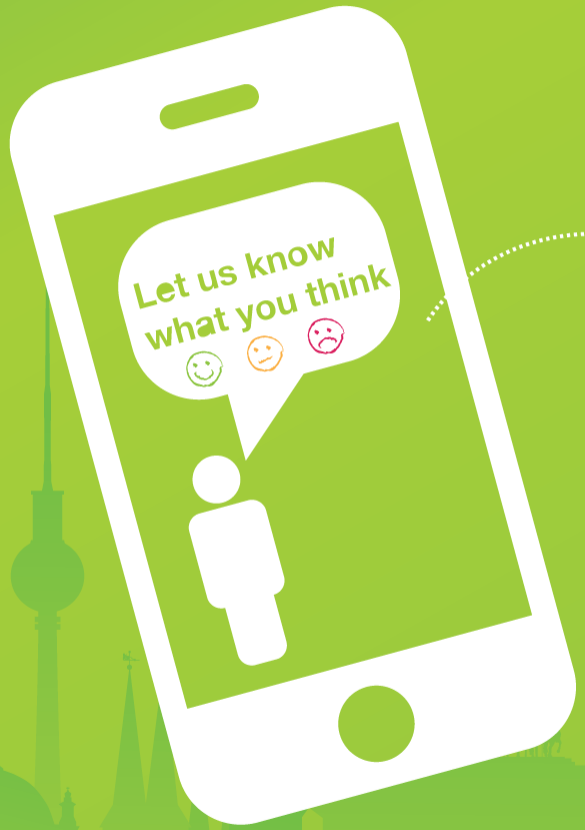
goto; conference



Click ‘engage’
to rate sessions
and ask questions

Follow us on Twitter @GOTOber

www.gotober.com

Discovering the Power of Dark Data

Or: The Value of Relationships

GotoCon Berlin Dec 2015

Michael Hunger

Caretaker, Neo4j Community

Ask me anything

michael@neo4j.com

@mesirii | @neo4j

What is Dark Data?


TED
IDEASWORTHSPREADING

Matt Ridley:

When ideas have sex

TEDGlobal 2010 · 16:26 · **Filmed** Jul 2010

Subtitles available in 33 languages

View interactive transcript

www.ted.com/talks/matt_ridley_when_ideas_have_sex

neo4j

Information wants to have SEX too.


Lives in Silos


Or Warehouses

Data Lake?


Data Stream

The world is a graph – everything is connected

- people, places, events
- companies, markets
- countries, history, politics
- sciences, art, teaching
- technology, networks, machines, applications, users
- software, code, dependencies, architecture, deployments

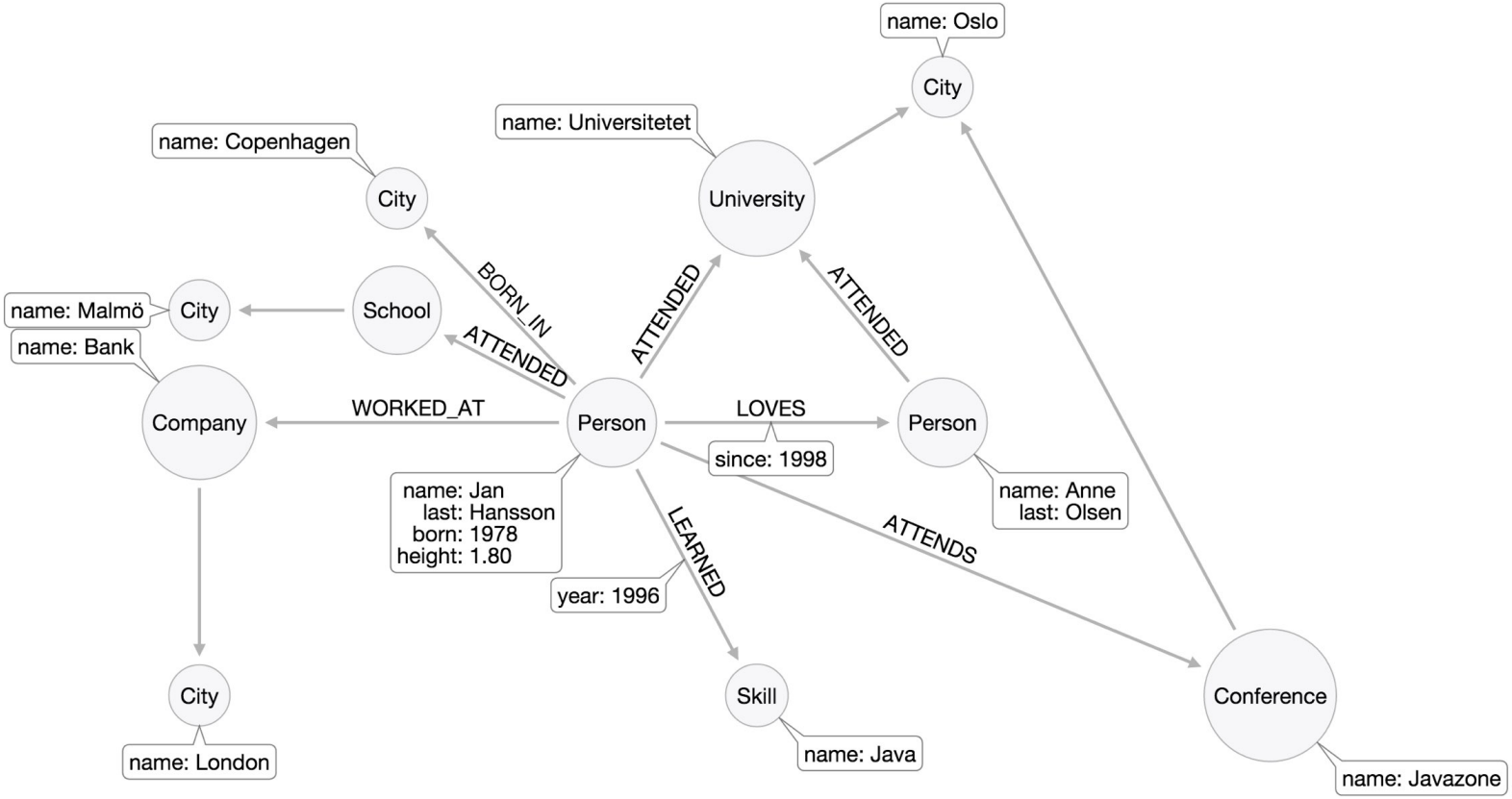
Like a brain

This is Jan

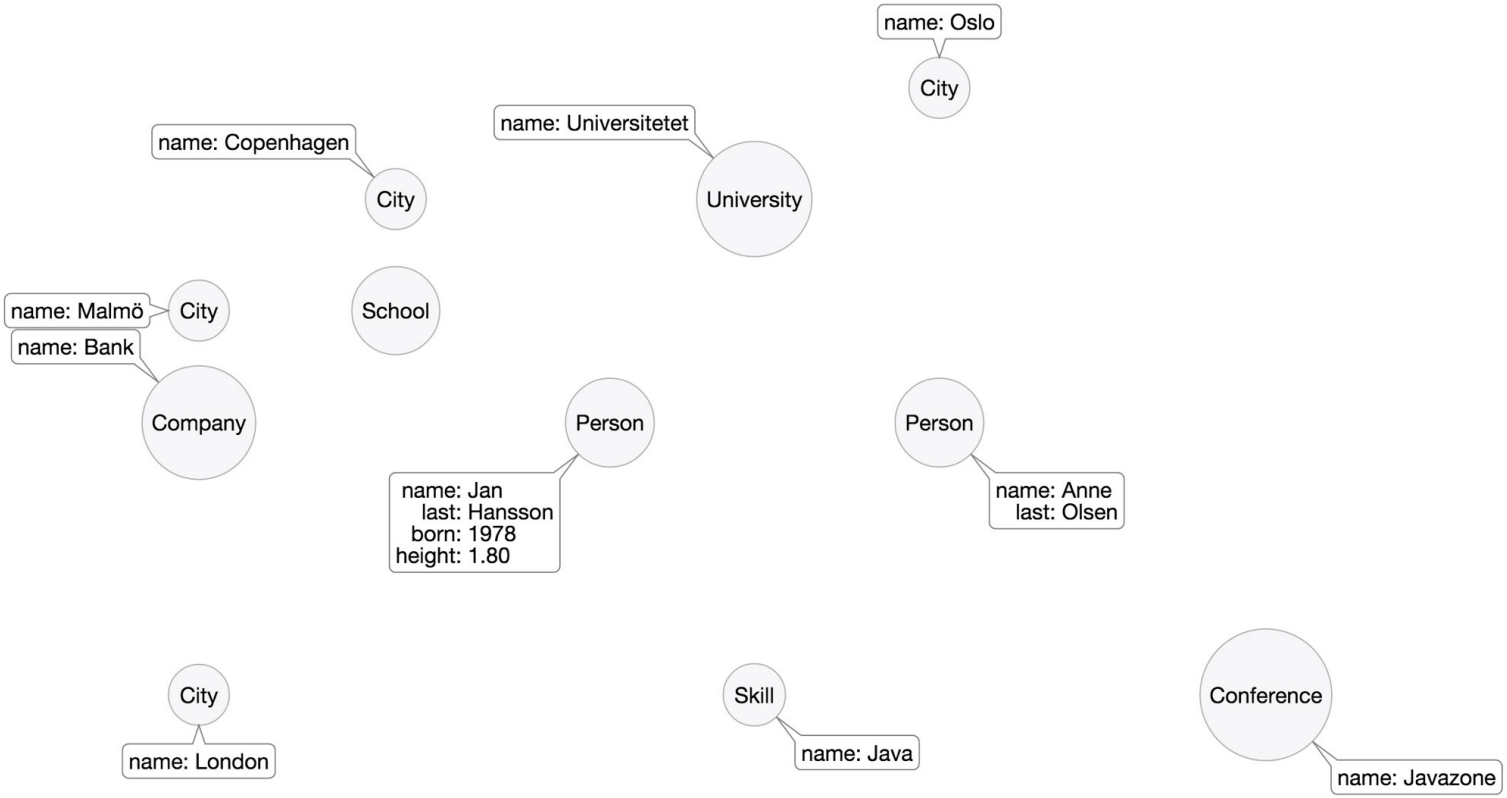
name: Jan
last: Hansson
born: 1978
height: 1.80

This is Jan

Meet Jan


name: Oslo
City
name: Universitetet
name: Copenhagen
University
City
BORN_IN
ATTENDED
ATTENDED
name: Malmö
City
School
name: Bank
ATTENDED
Company
WORKED_AT
Person
LOVES
Person
since: 1998
name: Jan
last: Hansson
born: 1978
height: 1.80
year: 1996
LEARNED
ATTENDS
name: Anne
last: Olsen
City
Skill
Conference
name: London
name: Java
name: Javazone

Context is King

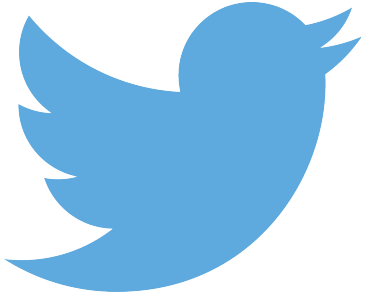


Community Graph

Neo4j


neo4j
?

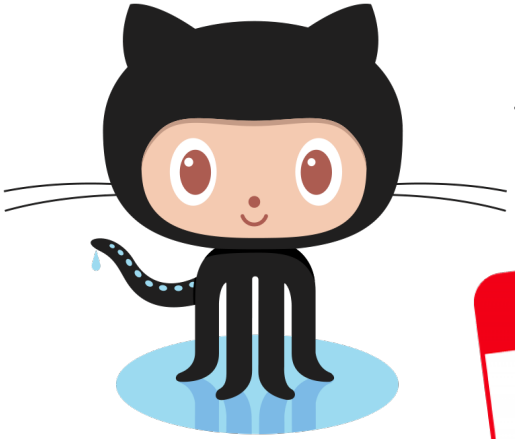
Let‘s have a look.



neo4j

WIKIPEDIA
The Free Encyclopedia

Google
Norway



[√] Seems to be legit

What about the full picture?

We‘re missing something!

A (Data) Model

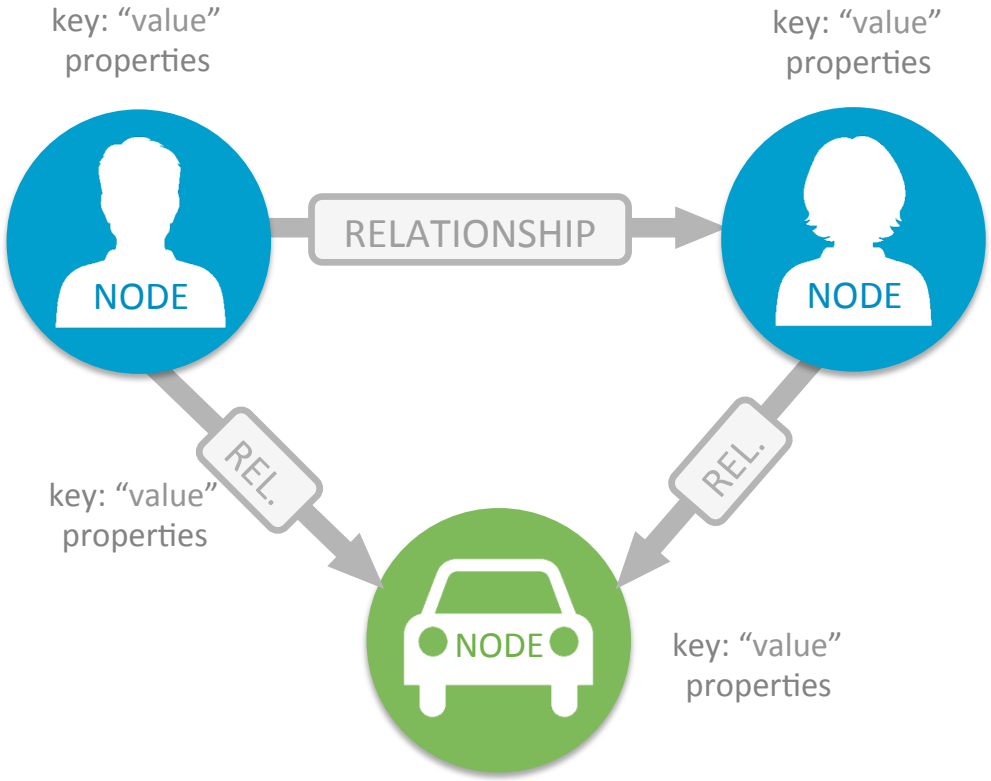
Property Graph Model

Nodes

- The entities in the graph
- Can have name-value *properties*
- Can be *labeled*

Relationships

- Relate nodes by type and direction
- Can have name-value *properties*



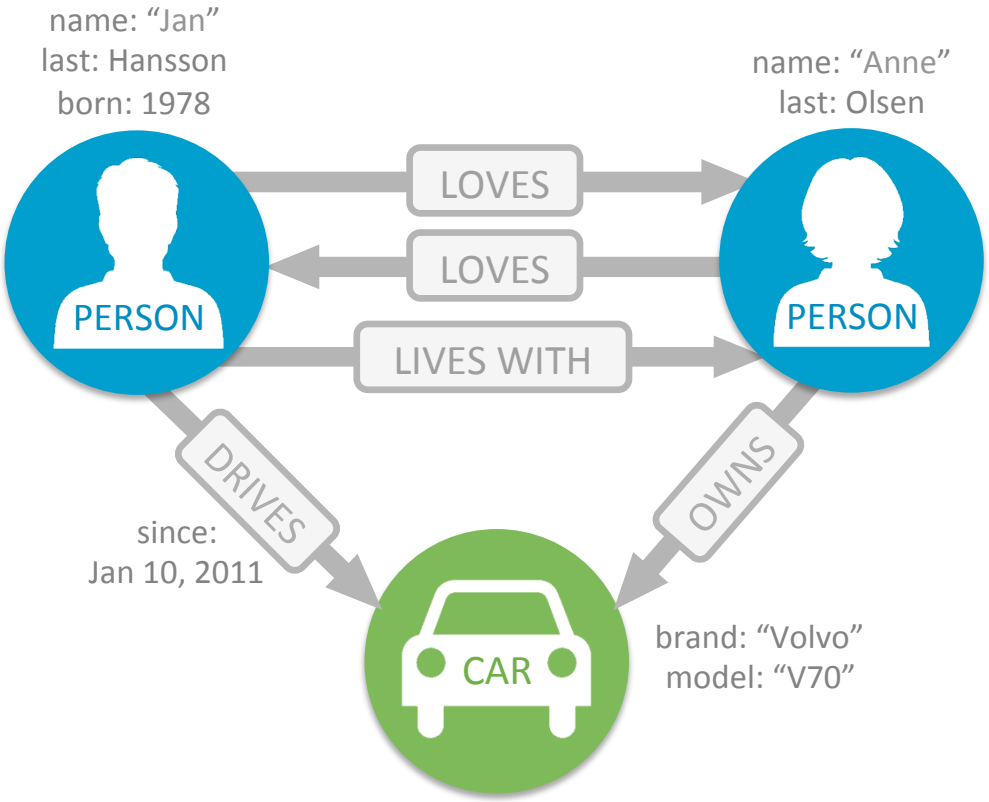
Property Graph Model & Example

Nodes

- The entities in the graph
- Can have name-value *properties*
- Can be *labeled*

Relationships

- Relate nodes by type and direction
- Can have name-value *properties*



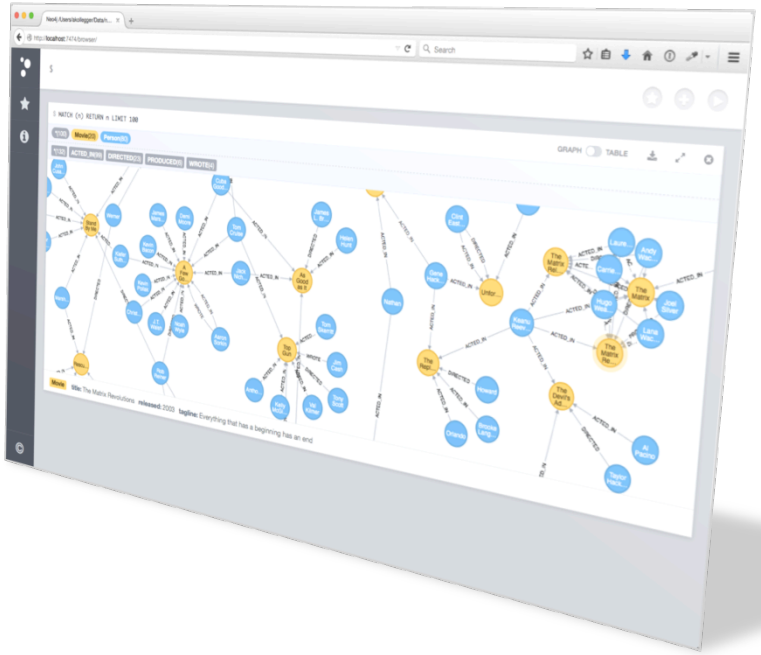
We need to store and query our data!


neo4j

Your friend Neo4j

An *open-source graph database*

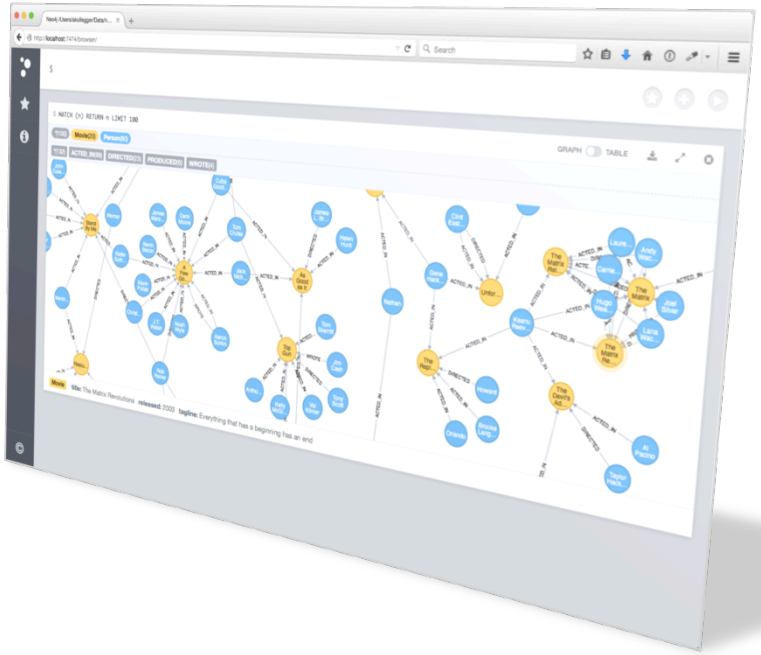
- **Manager and store** your **connected data** as a **graph**
- **Query relationships** easily and quickly
- **Evolve model and applications** to support new requirements and insights
- Built to solve **relational pains**



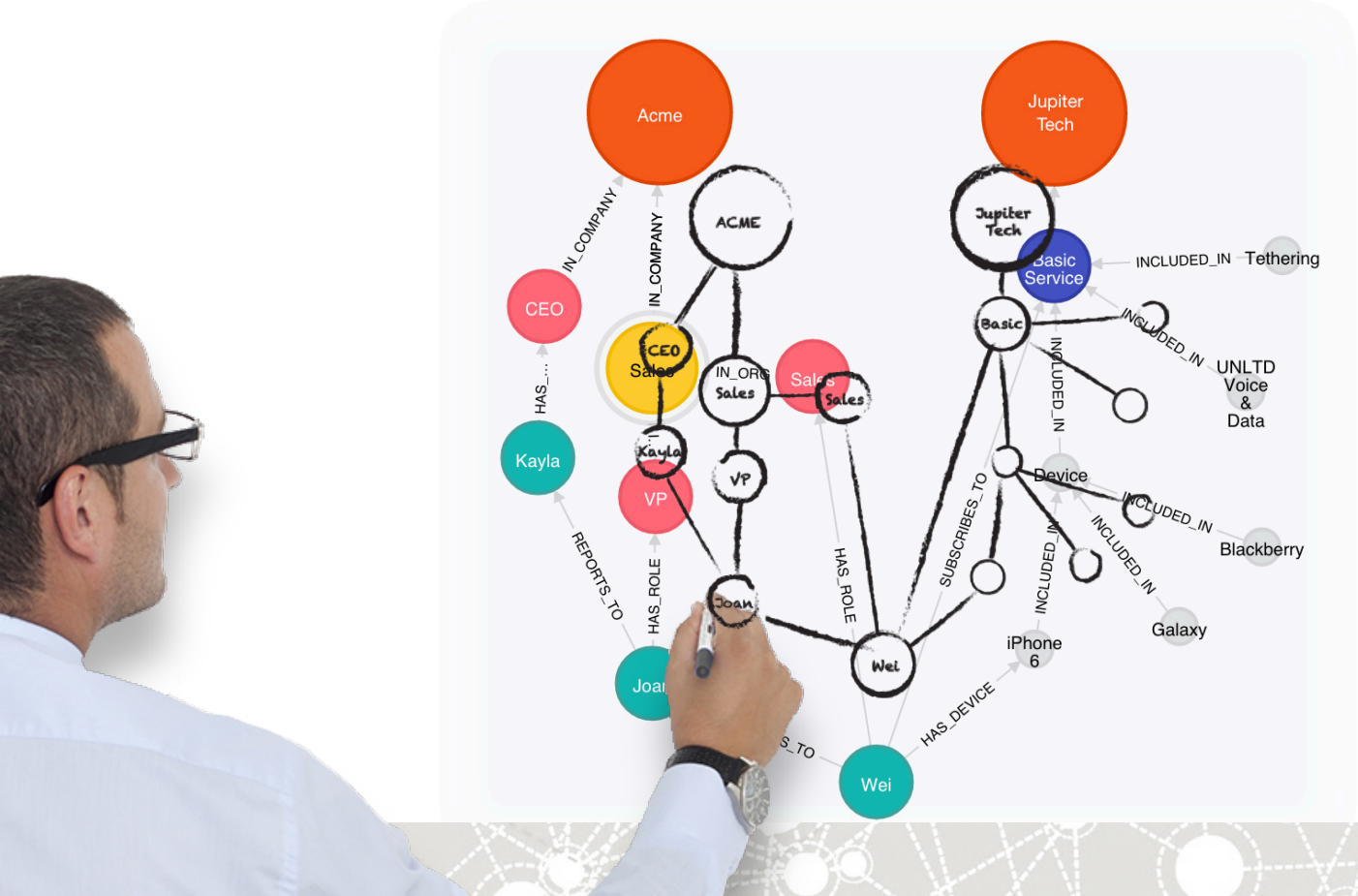
Your friend Neo4j

An *open-source graph database*

- **Built for Relationships**
- **Open Source**
- **Java & Scala**
- **High Performance**
- **ACID-Database**

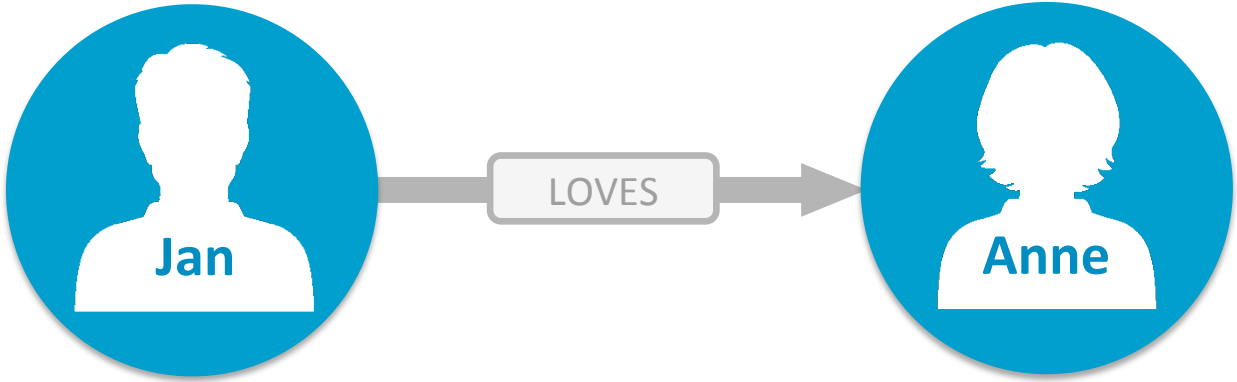


Whiteboard Friendly Graph Modeling



Query your graph data with Cypher

Graph Query Language: Cypher



NODE NODE

MATCH (jan:Person { name:“Jan”}) -[:LOVES]-> (anne:Person { name:“Anne”})

LABEL PROPERTY LABEL PROPERTY

Cypher: Clauses

- **CREATE** pattern
- **MERGE** pattern
- ADD
- DELETE

Cypher: Clauses

- **MATCH** pattern
  WHERE pred
- ORDER BY expr
  SKIP … LIMIT …
- **RETURN** expr AS alias …

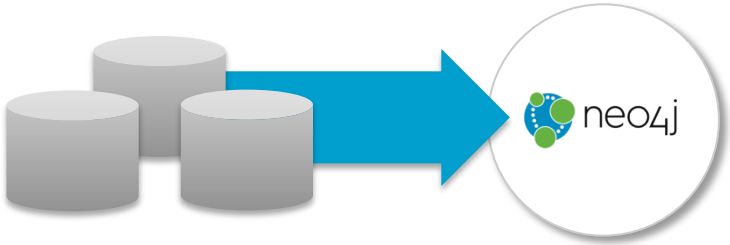
Cypher: Clauses

- WITH expr AS alias, ...
- UNWIND coll AS item
- **LOAD** CSV FROM „url“ AS row

Getting Data into Neo4j

Cypher-Based “LOAD CSV”

- Transactional (ACID) writes
- Initial and incremental loads of up to 10 million nodes and relationships

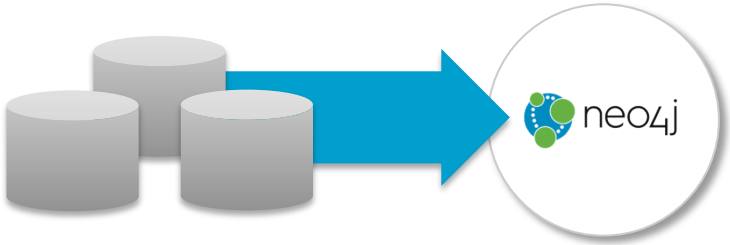



```
LOAD CSV WITH HEADERS FROM "url" AS row
MERGE (:Person {name:row.name,
                age:toInt(row.age)});
```

Getting Data into Neo4j

Load JSON with Cypher

- Send JSON as parameter
- Deconstruct the document
- Into a non-duplicated graph model

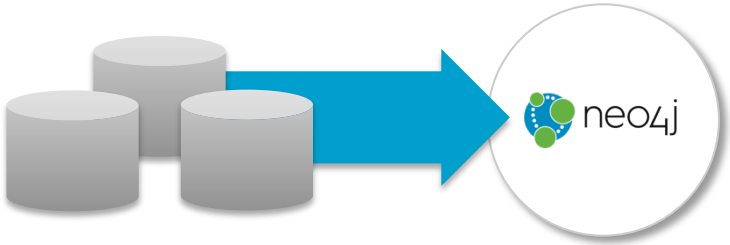



```
UNWIND {json}.items as item
MERGE (:Person {name:item.name,
                age:toInt(item.age)});
```

Getting Data into Neo4j

CSV Bulk Loader *neo4j-import*

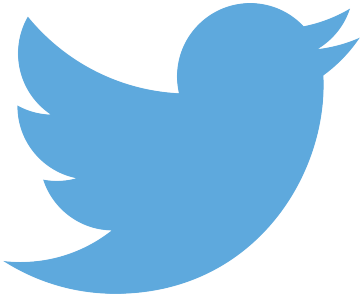
- For initial database population
- For loads with 10B+ records
- Up to 1M records per second



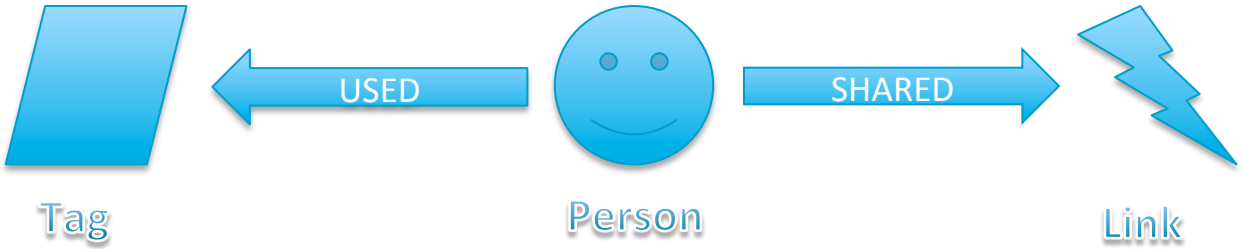

```
bin/neo4j-import –-into people.db
 --nodes:Person people.csv
 --relationship:FRIEND_OF friendship.csv
```

End of Detour
Demo Time

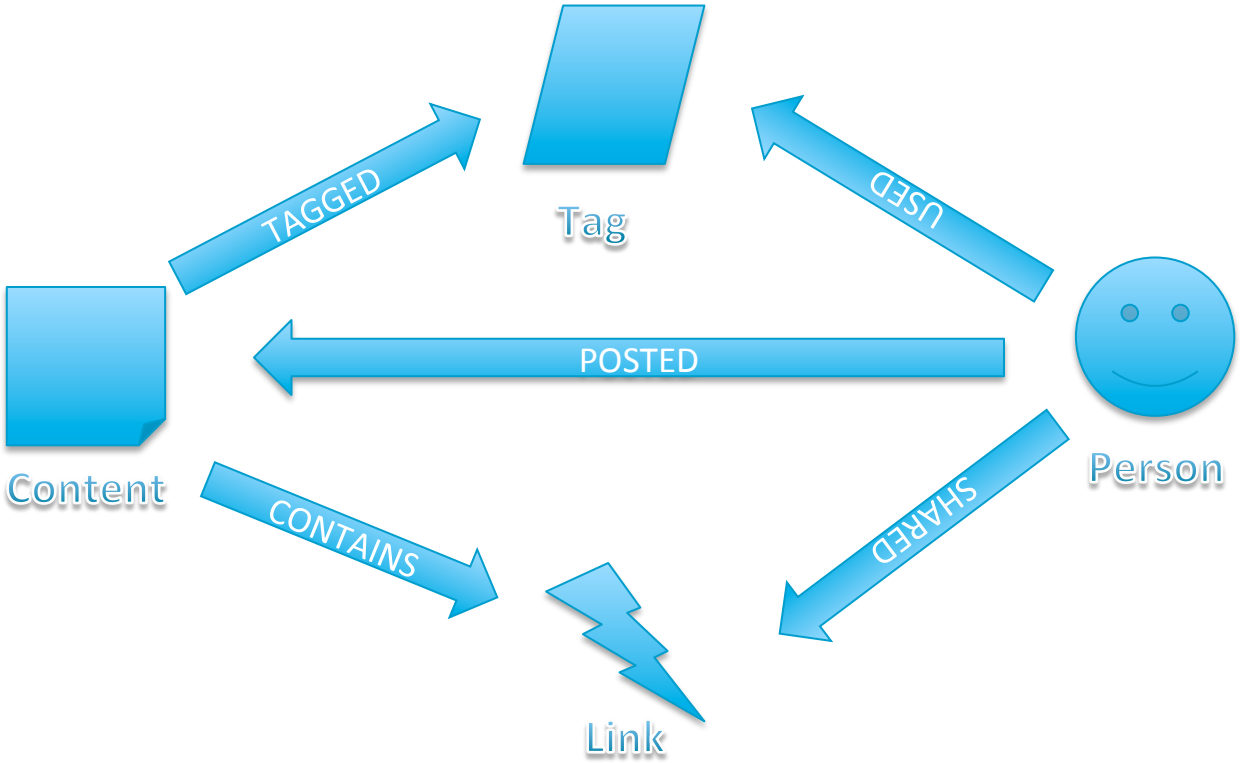
Let's LOAD



Core Model



Full Model

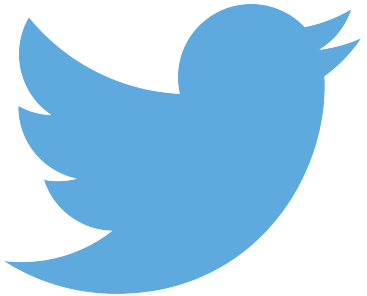

Tag
TAGGED
USED
POSTED
Content
Person
CONTAINS
SHARED
Link

Twitter

- Run a twitter search, exclude Neo4j sources
- „neo4j OR #neo4j OR @neo4j -from:@neo4j -from:@neotechnology“
- Pass resulting JSON directly to Cypher

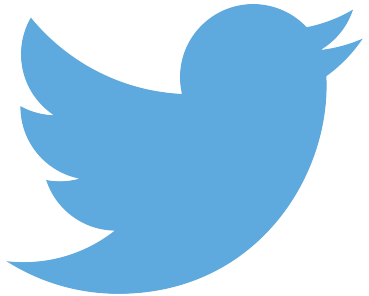
```
(:Person)-[:TWEETED]->(:Tweet:Content)-[:TAGGED]->(:Tag)
(:Tweet)-[:MENTIONS]->(:Person)
(:Tweet)-[:RETWEET]->(:Tweet)
(:Tweet)-[:REPLY]->(:Tweet)
(:Tweet)-[:CONTAINS]->(:Link)
```

Twitter



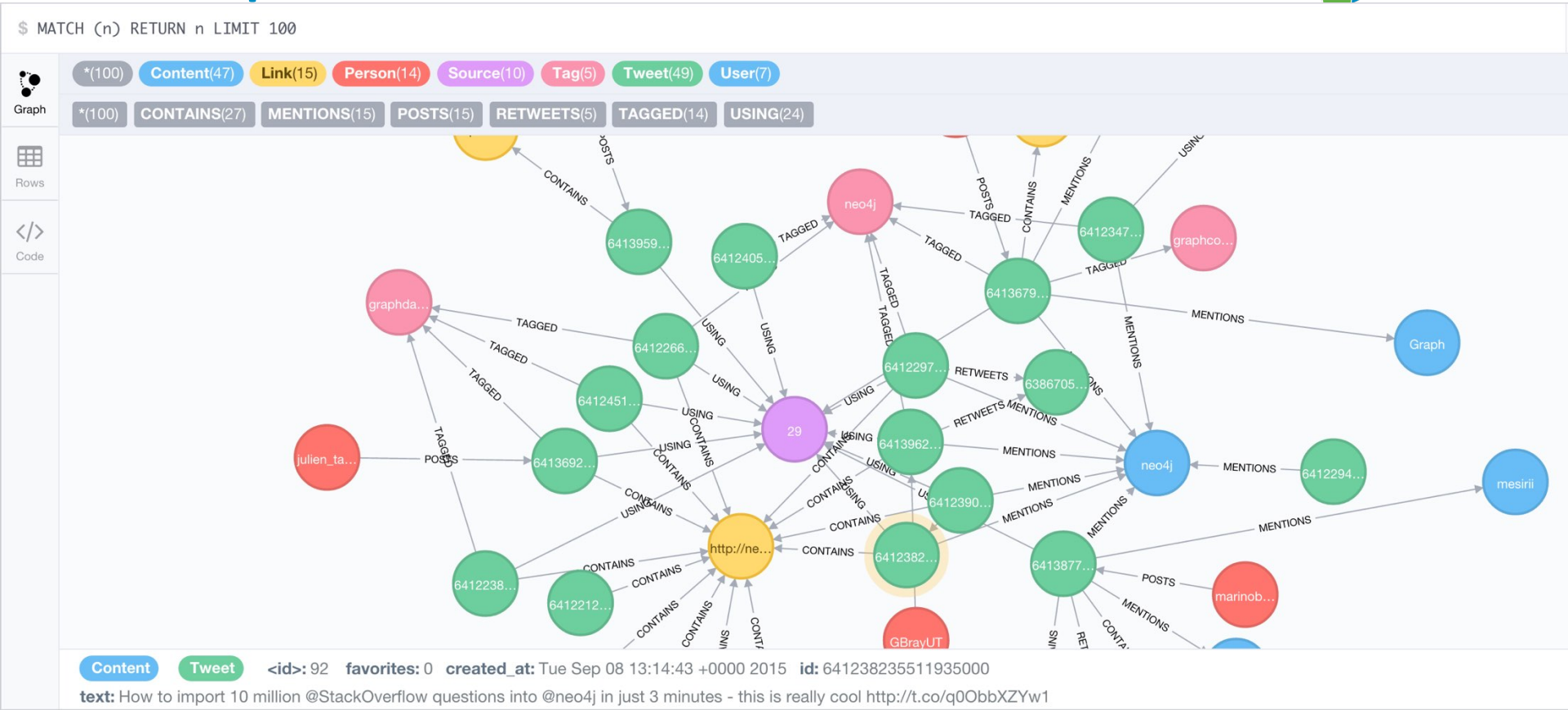
```
UNWIND {tweets} AS t
WITH t, t.entities AS e, t.user AS u

MERGE (tweet:Tweet:Content {id:t.id})
SET tweet.text = t.text, tweet.created_at = t.created_at,...

MERGE (p:Person {name:u.name})
SET p.handle = u.screen_name, p.followers = u.followers_count, ...

MERGE (p)-[:POSTED]->(tweet)

FOREACH (h IN e.hashtags |
  MERGE (tag:Tag {name:toLower(h.text)})
  MERGE (tweet)-[:TAGGED]->(tag))

FOREACH (url IN e.urls |
  MERGE (link:Link {url:u.expanded_url})
  MERGE (tweet)-[:CONTAINS]->(link))
...
```

Twitter



```
UNWIND {tweets} AS t
WITH t, t.entities AS e, t.user AS u

MERGE (tweet:Tweet:Content {id:t.id})
SET tweet.text = t.text, tweet.created_at = t.created_at,...

MERGE (p:Person {name:u.name})
SET p.handle = u.screen_name, p.followers = u.followers_count, ...

MERGE (p)-[:POSTED]->(tweet)

FOREACH (h IN e.hashtags |
  MERGE (tag:Tag {name:toLower(h.text)})
  MERGE (tweet)-[:TAGGED]->(tag))

FOREACH (url IN e.urls |
  MERGE (link:Link {url:u.expanded_url})
  MERGE (tweet)-[:CONTAINS]->(link))
...
```

Data imported



Connect! All the Things!

- twitter handle
- email
- website
- name (non-unique)
- tags
- url

StackOverflow

- SO Dump is CSV wrapped in XML
- Filter entries for Neo4j tag
- Use LOAD CSV
- Try to connect on **tags** and **people's names**

```
(:Person)-[:POSTED]->(:Question:Content)-[:TAGGED]->(:Tag)
(:Question)<-[:ANSWERED]-(:Person)
```

neo4j.com/blog/import-10m-stack-overflow-questions/
neo4j.com/blog/cypher-load-json-from-url/

StackOverflow

```
LOAD CSV WITH HEADERS FROM "file:users.csv" AS row
MERGE (owner:Person {name:toLower(row.display_name)})
  ON CREATE SET owner.so_id = row.user_id, owner:StackOverflow, ...;

LOAD CSV WITH HEADERS FROM "file:posts.csv" AS row
MERGE (question:Question:Content {so_id:row.question_id})
 ON CREATE SET question.title = row.title,
      question.answer_count = toInt(row.answer_count),
      question.score = toInt(row.score),
      question.creation_date = row.creation_date
FOREACH (tagName IN split(row.tags,";") |
  MERGE (t:Tag {name:toLower(tagName)})
  MERGE (question)-[:TAGGED]->(t));

LOAD CSV WITH HEADERS FROM "file:posts.csv" AS row
MATCH  (owner:Person {name:row.user_id})
MATCH  (question:Question {so_id:row.question_id})
CREATE (owner)-[:POSTED]->(question_id);
```

GitHub

- Find Repositories mentioning Neo4j via API
- Import JSON via parameters
- Try to connect on **language tags** and **people's names**
- Potentially Github Events API / Dump

```
(:Person)-[:POSTED]->(:Question:Content)-[:TAGGED]->(:Tag)
(:Question)<-[:ANSWERED]-(:Person)
```

GitHub


```
WITH {json} as data
UNWIND data.items as r
MERGE (repository:Repository:Content {gh_id:r.id})
 ON CREATE SET repository.name = r.name, repository.title = r.description,
  repository.score = r.score, repository.created_at = r.created_at,
  repository.forks = r.forks_count, repository.stars = r.stargazers_count

MERGE (link:Link {link:toLower(r.homepage)})
MERGE (repository)-[:CONTAINS]->(link)

MERGE (lang:Tag {name:toLower(r.language)})
MERGE (repository)-[:TAGGED]->(language)

MERGE (owner:Person {name:toLower(r.owner.login)})
  SET owner.gh_id = r.owner.id, owner.type = r.owner.type, owner:GitHub
MERGE (owner)-[:OWNS]->(repository)
```

Meetup

- Many active Neo4j Meetups around the world
- Mark Needham did extensive analysis of them with Neo4j and Clojure, R, Python, Cypher
- Signed API URLs

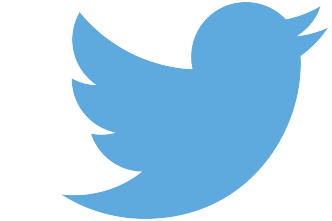
```
(:Person)-[:MEMBER]->(:Group)-[:TAGGED]->(:Tag)
(:Question)<-[:ANSWERED]-(:Person)
```

Meetup


```
WITH {json} as data
UNWIND data.items as r
MERGE (repository:Repository:Content {gh_id:r.id})
 ON CREATE SET repository.name = r.name, repository.title = r.description,
  repository.score = r.score, repository.created_at = r.created_at,
  repository.forks = r.forks_count, repository.stars = r.stargazers_count

MERGE (link:Link {link:toLower(r.homepage)})
MERGE (repository)-[:CONTAINS]->(link)

MERGE (lang:Tag {name:toLower(r.language)})
MERGE (repository)-[:TAGGED]->(language)

MERGE (owner:Person {name:toLower(r.owner.login)})
  SET owner.gh_id = r.owner.id, owner.type = r.owner.type, owner:GitHub
MERGE (owner)-[:OWNS]->(repository)
```

Let‘s answer some questions

- How many people where active on more than 1 platform?
- Which tags were most frequently used
- Which tags were most frequently used **together**



Dark Data?


Dark Forces?


“Information assets that organizations collect, process and store during regular business activities,
but generally fail to use for other purposes.”
Gartner®

Golden Customer

Software Analytics

jqassistant.org

Software Analytics

Software **is** connected information

- Source -> AST
- Inheritance, Composition, Delegation
- Call Trees
- Runtime Memory
- Dependencies
  - Modules, Libraries
- Tests
- ...

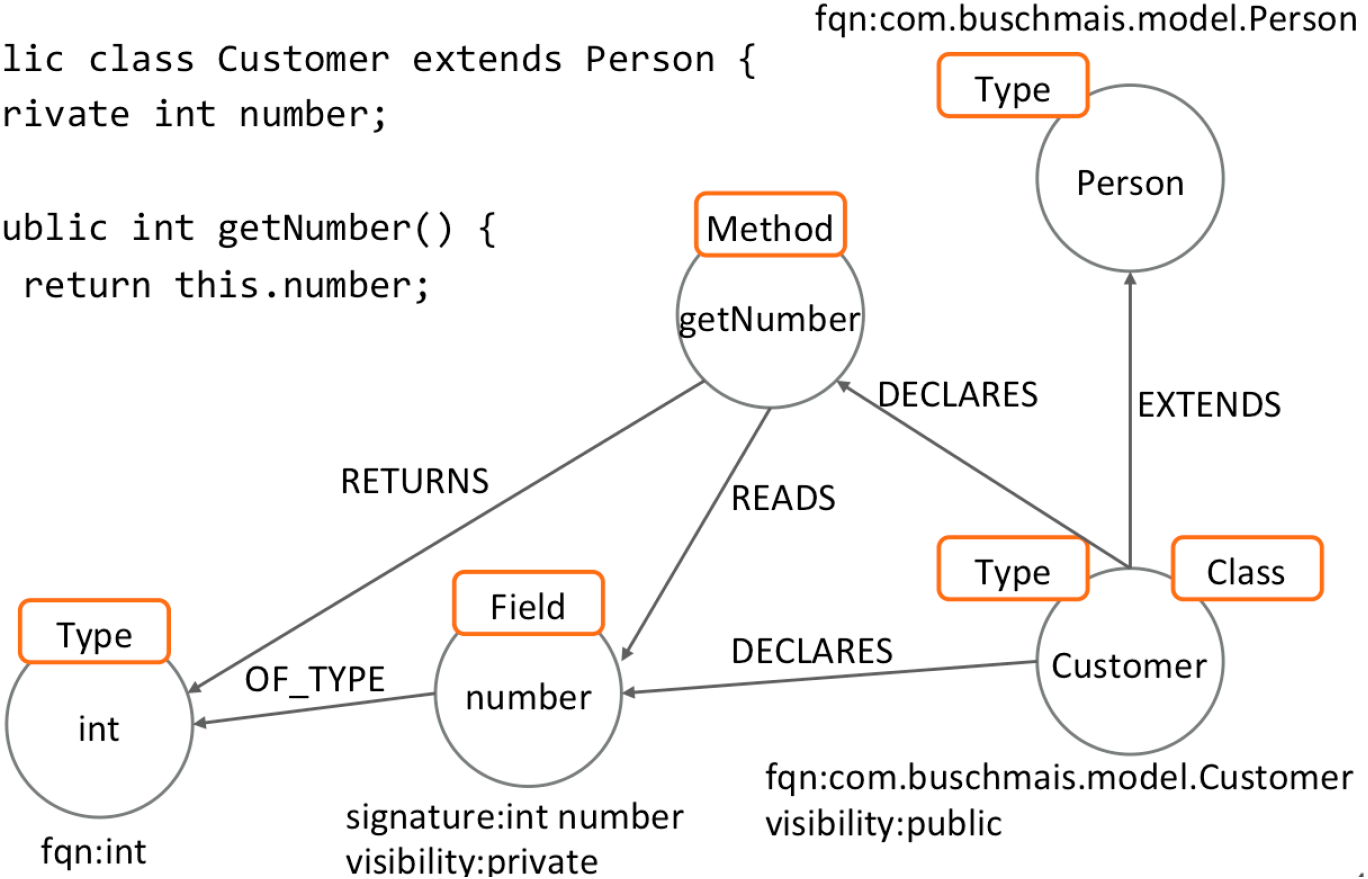
https://jqassistant.org

jQAssistant

- GeekCruise: My first Neo4j project
- **Software deteriorates**
- Develop rules and enforce them
- Commercial Tools too inflexible
- Open Source Software ...
  - Scanner -> Enhancer -> Analyzer
- Enrichment, Concepts and Rulez in Cypher
- Scanner Plugins
- Integrate in Build Process
  - Fail, Generate Reports, ...

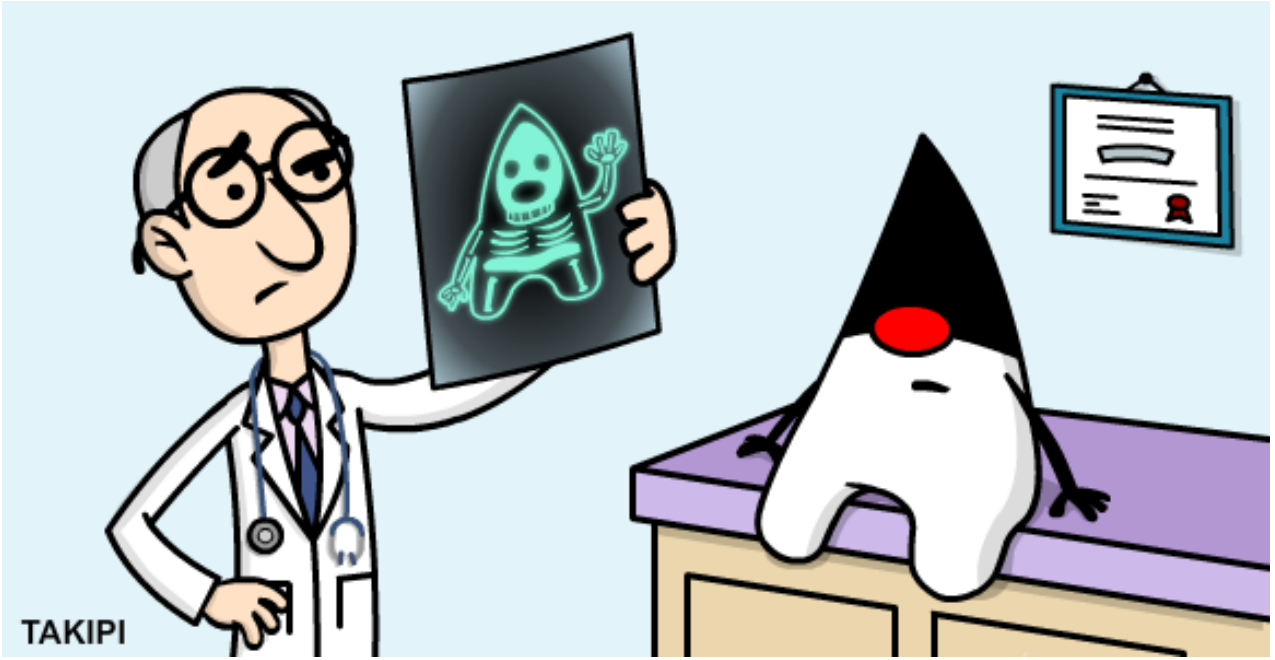
http://jqassistant.org

```
public class Customer extends Person {
  private int number;

  public int getNumber() {
    return this.number;
  }
}
```



You can explore ...

... The JDK



http://jqassistant.org/demo/java8

A tale of french silos

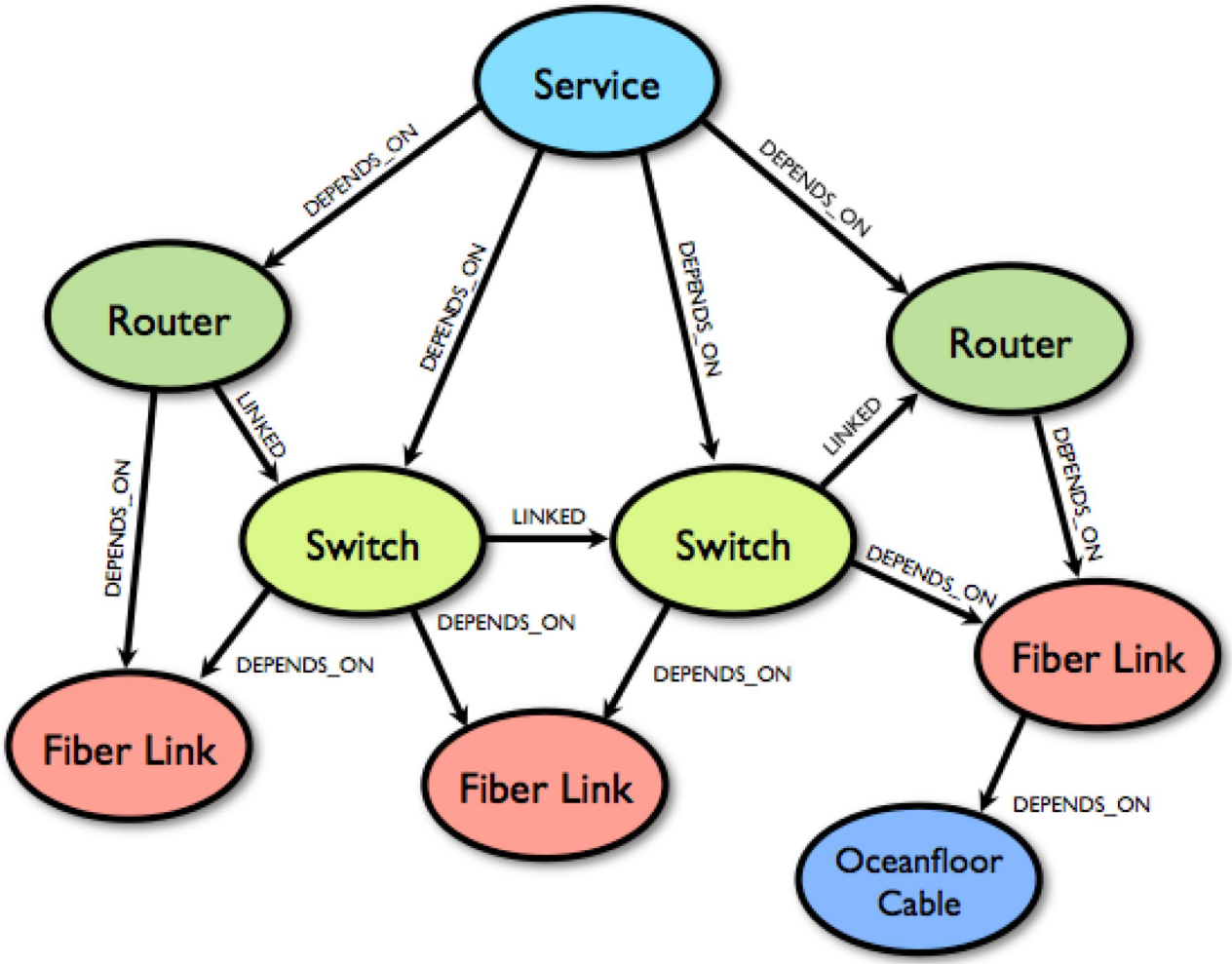
SFR France

Industry: Communications
Use case: Network Management

Paris, France

- Large French communications company
- Tons of various networking infrasctructure
- Has to shut down equipment for maintenance
- **Who is impacted? How to compensate?**
- Data lives in 30+ systems
- Took a week to print, reseach, inform ...
- Until ...

Domain


Service
DEPENDS_ON
DEPENDS_ON
DEPENDS_ON
DEPENDS_ON
Router
Router
LINKED
LINKED
DEPENDS_ON
DEPENDS_ON
Switch
LINKED
Switch
DEPENDS_ON
DEPENDS_ON
DEPENDS_ON
DEPENDS_ON
Fiber Link
Fiber Link
Fiber Link
DEPENDS_ON
Oceanfloor Cable

Industry: Communications
Use case: Network Management

Paris, France

- Second largest communications company in France
- Part of Vivendi Group, partnering with Vodafone

Business Problem

- Infrastructure maintenance took **one full week** to plan, because of the need to model network impacts
- Needed rapid, automated "**what if**" analysis to ensure resilience during unplanned network outages
- Identify weaknesses in the network to uncover the need for additional redundancy
- Network information spread **across > 30 systems**, with daily changes to network infrastructure
- Business needs sometimes changed very rapidly

Solution & Benefits

- **Flexible** network inventory management system, to support modeling, aggregation & troubleshooting
- **Single source of truth** (Neo4j) representing the entire network
- Dynamic system loads data **from 30+ systems**, and allows new applications to access network data
- Modeling efforts greatly reduced because of the **near 1:1 mapping** between the real world and the graph
- **Flexible schema** highly adaptable to changing business requirements

Visual Graph Search

For Non-Developers

Popoto.js

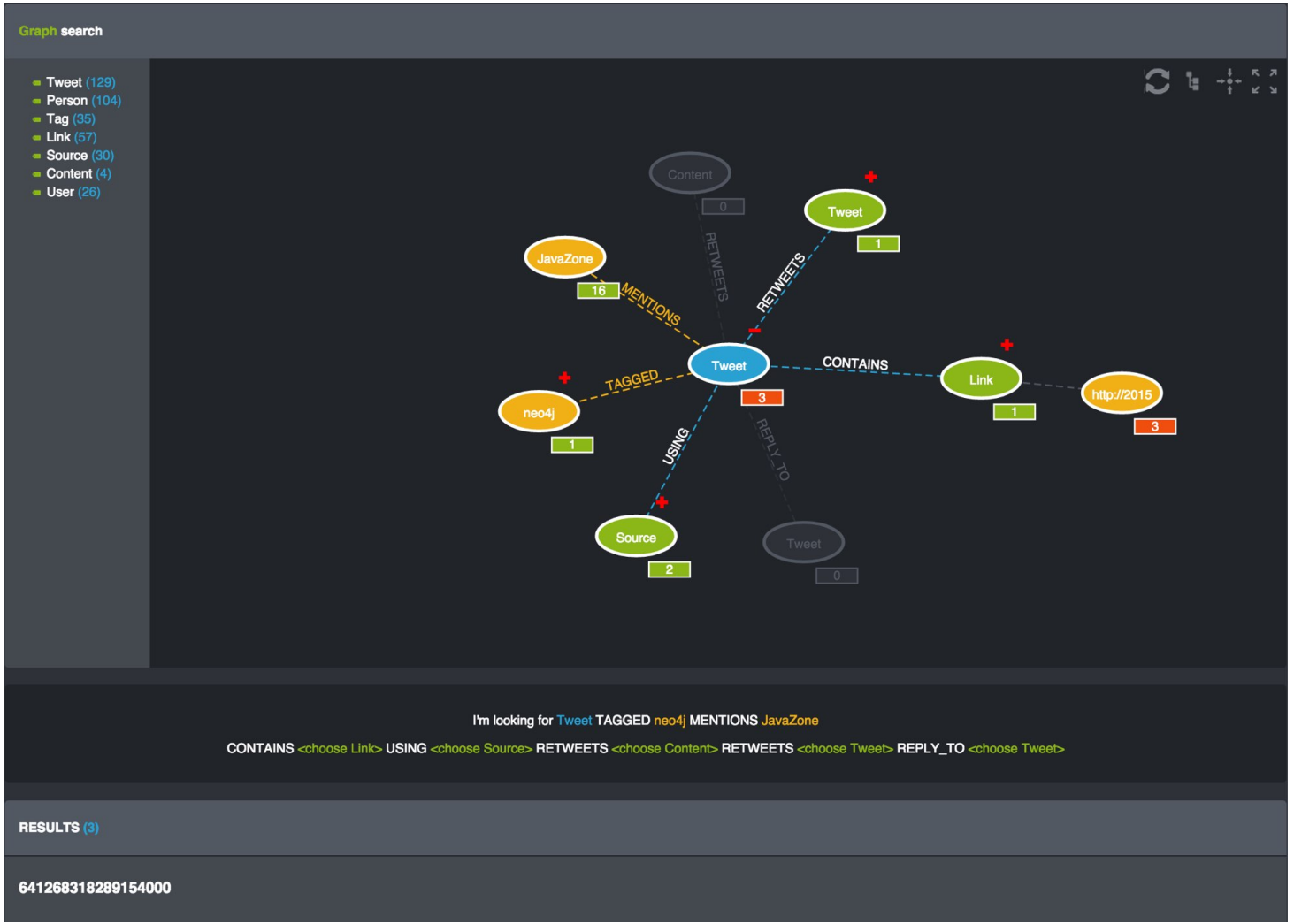
- Neat Javascript library based on **d3.js**
- Uses Graph Metadata to offer visual search
- Categories to filter Instances
- Component based extensions
- Graph
- Zero Config with Web Extension

http://www.popotojs.com/

Popoto.js

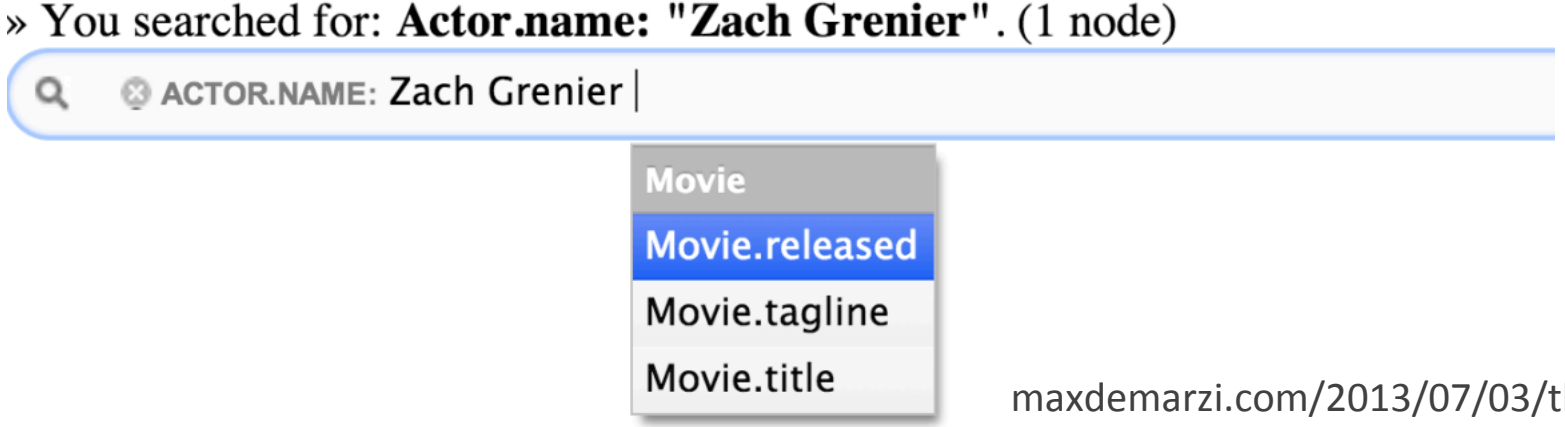

localhost:7474/popoto/
POPOTO JS 0.0-α
Graph search
Product (77)
Category (8)
Supplier (29)
Employee (9)
Customer (91)
Order (830)
Order
295
Tarte au su
48
Pavlova
43
Sir Rodney'
39
Sir Rodney'
16
Category
Confections
8
PART_OF
Product
Product
13
NuNuCa Nuß
18
Teatime Cho
37
Zaanse koek
I'm looking for Order PART_OF Confections
ORDERS <choose Product>
RESULTS (295)
filtered Orders
10481
10334

Popoto.js


Graph search
Tweet (129)
Person (104)
Tag (35)
Link (57)
Source (30)
Content (4)
User (26)
Content
0
Tweet
1
JavaZone
16
MENTIONS
RETWEETS
RETWEETS
Tweet
3
CONTAINS
Link
1
http://2015
3
TAGGED
neo4j
1
USING
REPLY_TO
Source
2
Tweet
0
I'm looking for Tweet TAGGED neo4j MENTIONS JavaZone
CONTAINS <choose Link> USING <choose Source> RETWEETS <choose Content> RETWEETS <choose Tweet> REPLY_TO <choose Tweet>
RESULTS (3)
641268318289154000

Visual Search Bar

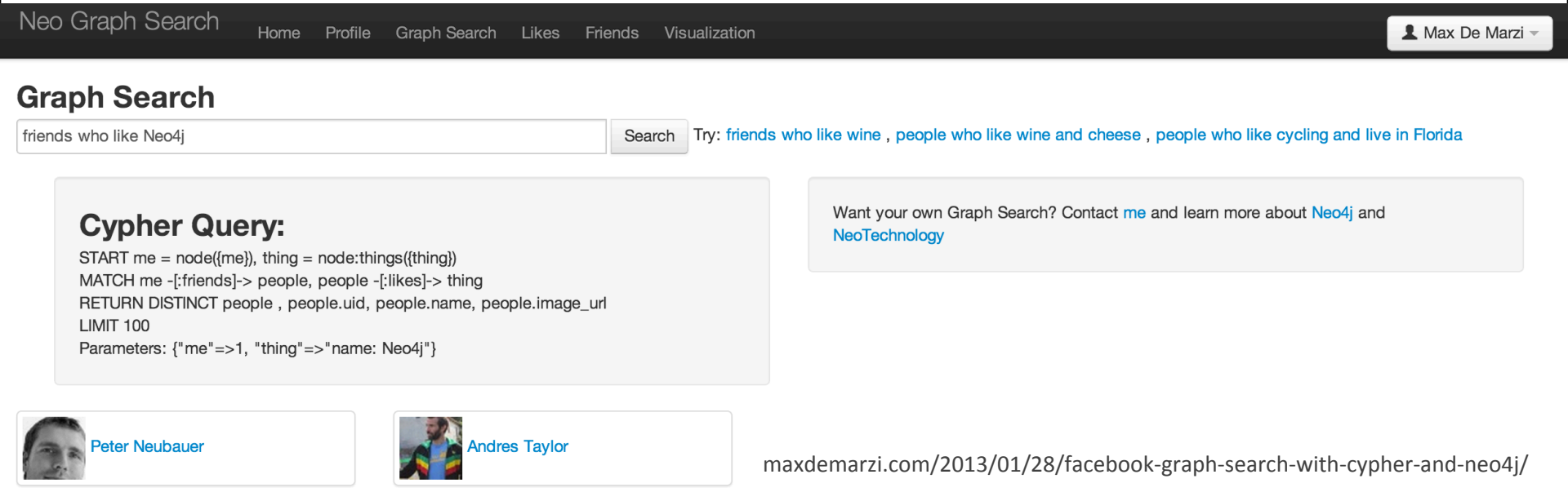
- Based on visualsearch.js
- Uses graph metadata for parametrization
- Limit results by selected items
- By Max de Marzi



maxdemarzi.com/2013/07/03/the-last-mile/

Facebook Graph Search

- Natural Language to Cypher
- Ruby TreeTop Gem for NLP
- Convert phrases to Cypher Fragments



maxdemarzi.com/2013/01/28/facebook-graph-search-with-cypher-and-neo4j/

Users Love Neo4j

Performance

"The Neo4j graph database gives us drastically improved performance and a simple language to query our connected data”

- Sebastian Verheugher, Telenor

Scale and Availability

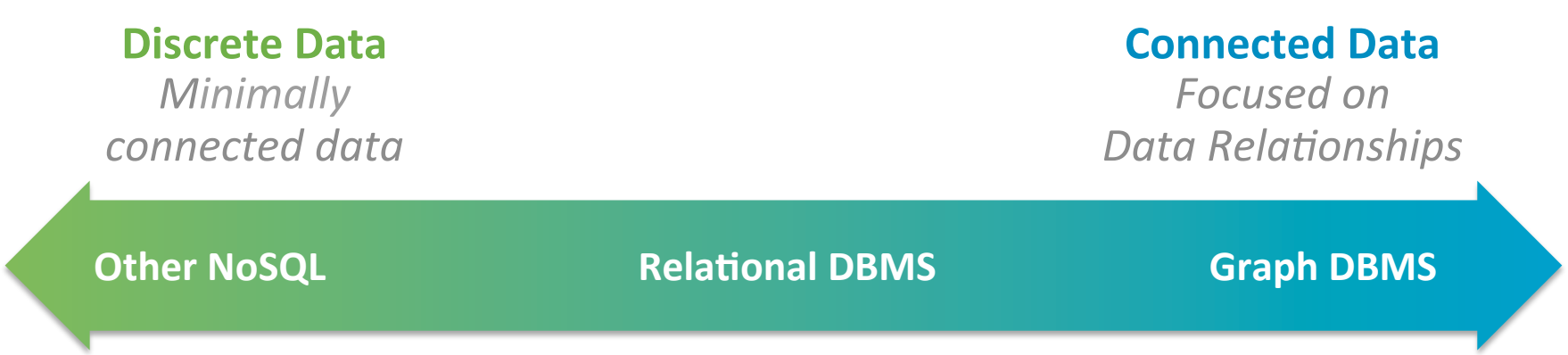
"As the current market leader in graph databases, and with enterprise features for scalability and availability, Neo4j is the right choice to meet our demands.”

- Marcos Wada, Walmart

“We found Neo4j to be literally **thousands of times faster** than our prior MySQL solution, with queries that require **10 to 100 times less code**. Today, Neo4j provides eBay with functionality that was **previously impossible**.”

Volker Pacher
Senior Developer

Summary - Use the Right Database for the Job



Graph Databases are designed for data relationships

Development Benefits
Easy model maintenance
Easy query

Deployment Benefits
Ultra high performance
Minimal resource usage

There Are Lots of Ways to Easily Learn Neo4j

community

Graph Academy
Learn. Graph. Deploy.

On-site Training

BOOKS

Documentation

GraphGist

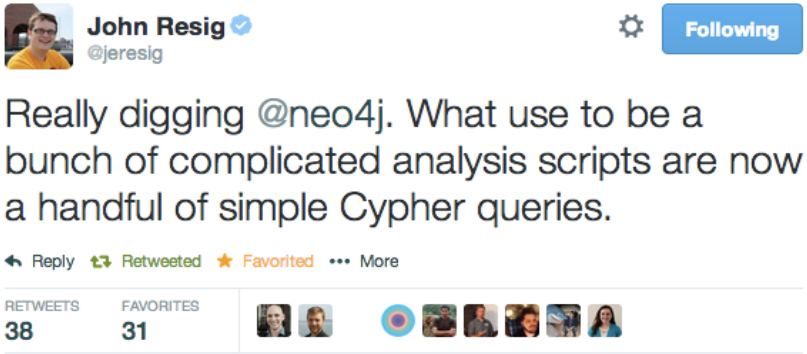
Built-in Guides

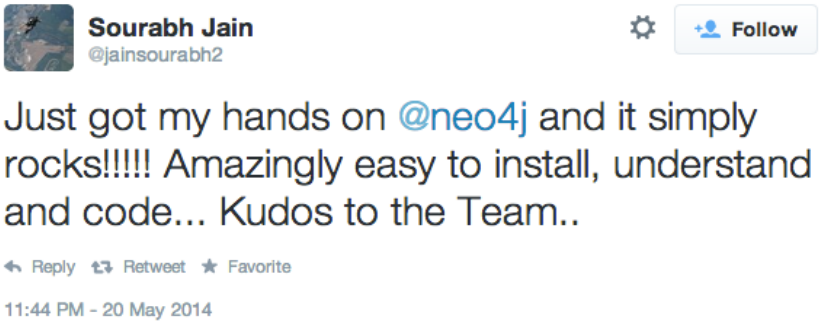
Google Groups

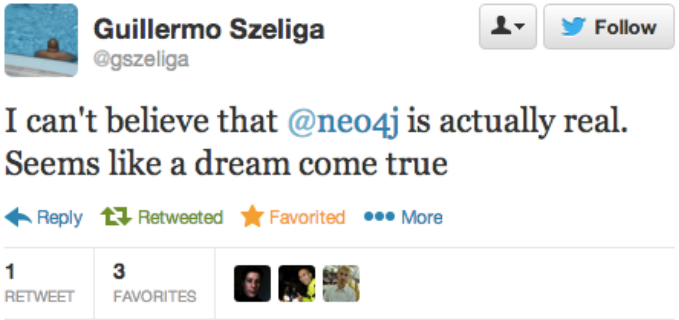
Online Training

neo4j.com/developer

Users Love Neo4j – Will you too?







Thanks! Questions?

Free O'Reilly Book:
graphdatabases.com
Find Me: @neo4j
Grab some stickers!

goto; conference

Please

Remember to rate session

Thank you!

Why should I care?

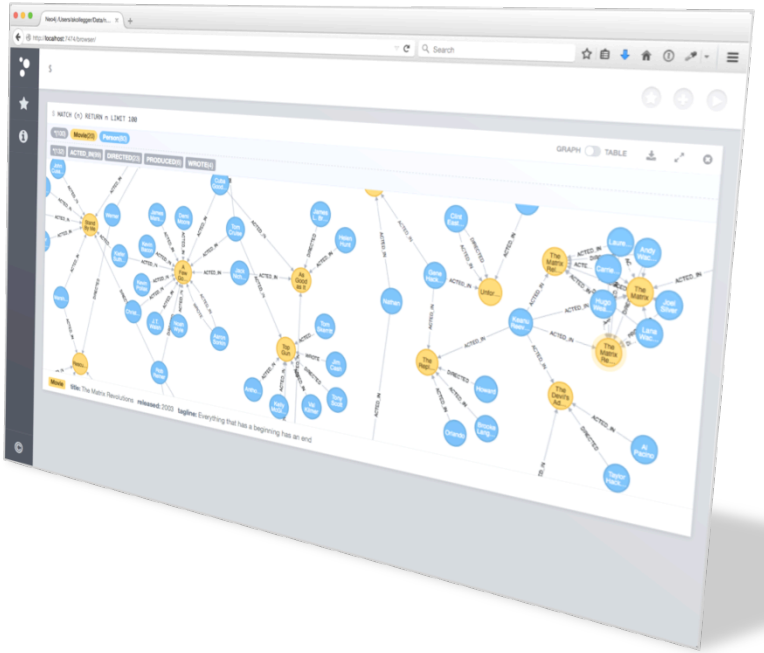
Because Relationships Matter

What is it with Relationships?

- World is full of connected people, events, things
- There is “Value in Relationships” !
- What about Data Relationships?
- How do you store your object model?
- How do you explain
  JOIN tables to your boss?

Neo4j – allows you to connect the dots

- Was built to efficiently
  - ***store***,
  - ***query*** and
  - ***manage*** highly connected data
- Transactional, ACID
- Real-time OLTP
- Open source
- Highly scalable already on few machines



Value from Data Relationships

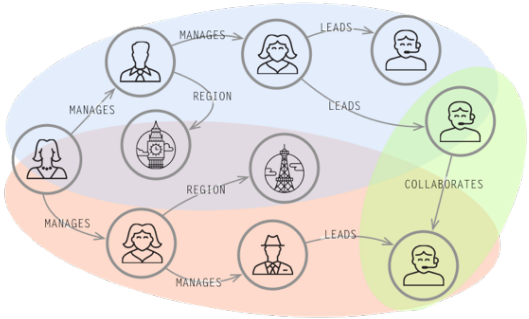
Common Use Cases

Internal Applications

Master Data Management

Network and
IT Operations

Fraud Detection

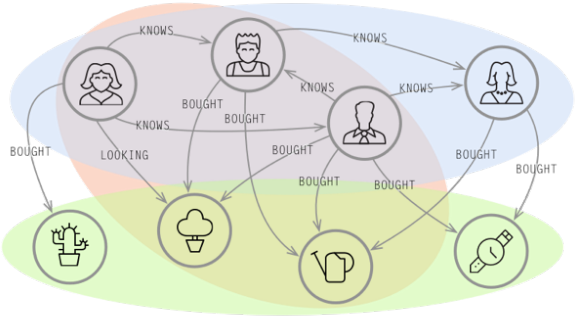


Customer-Facing Applications

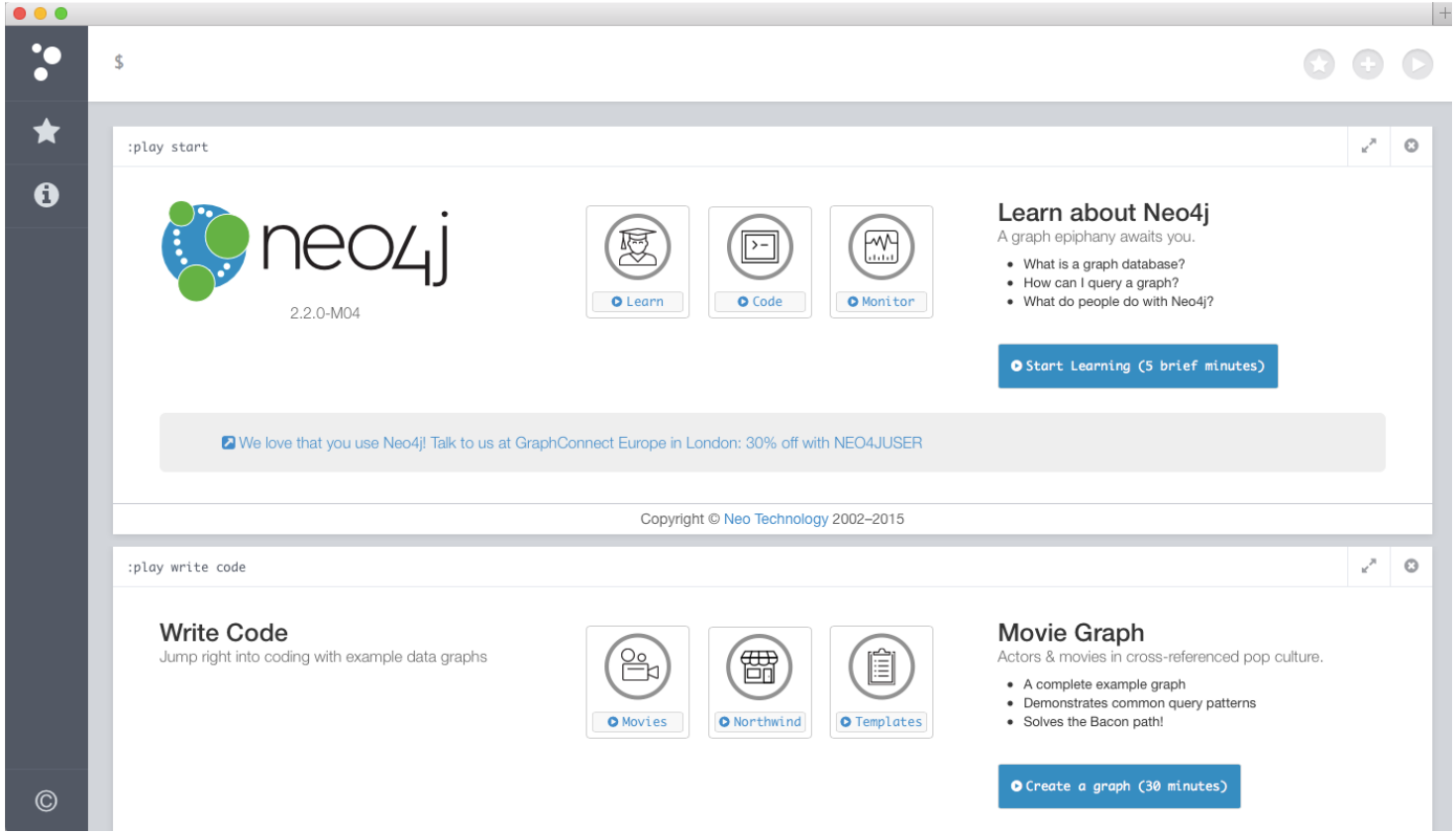
Real-Time Recommendations

Graph-Based Search

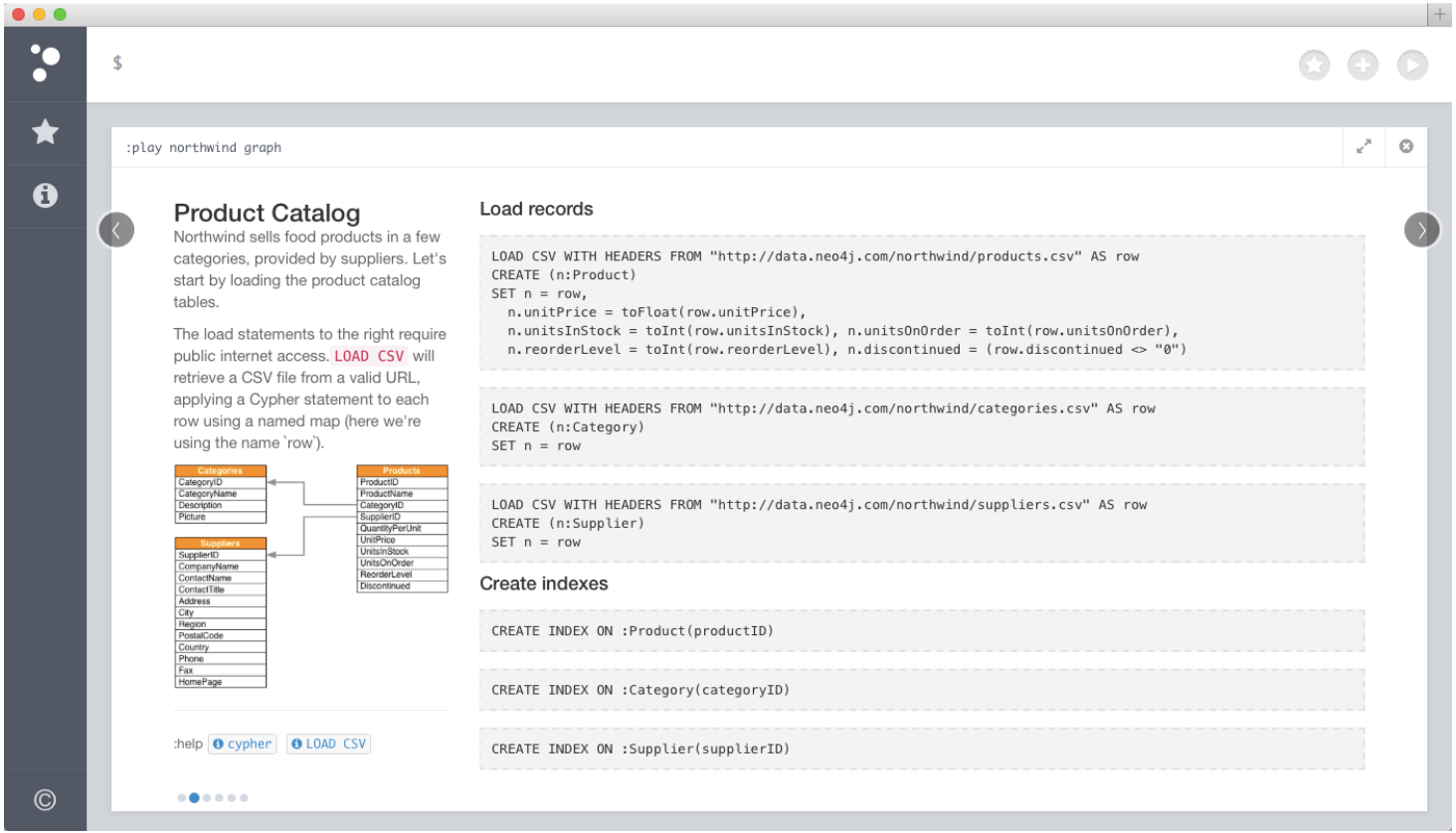
Identity and
Access Management



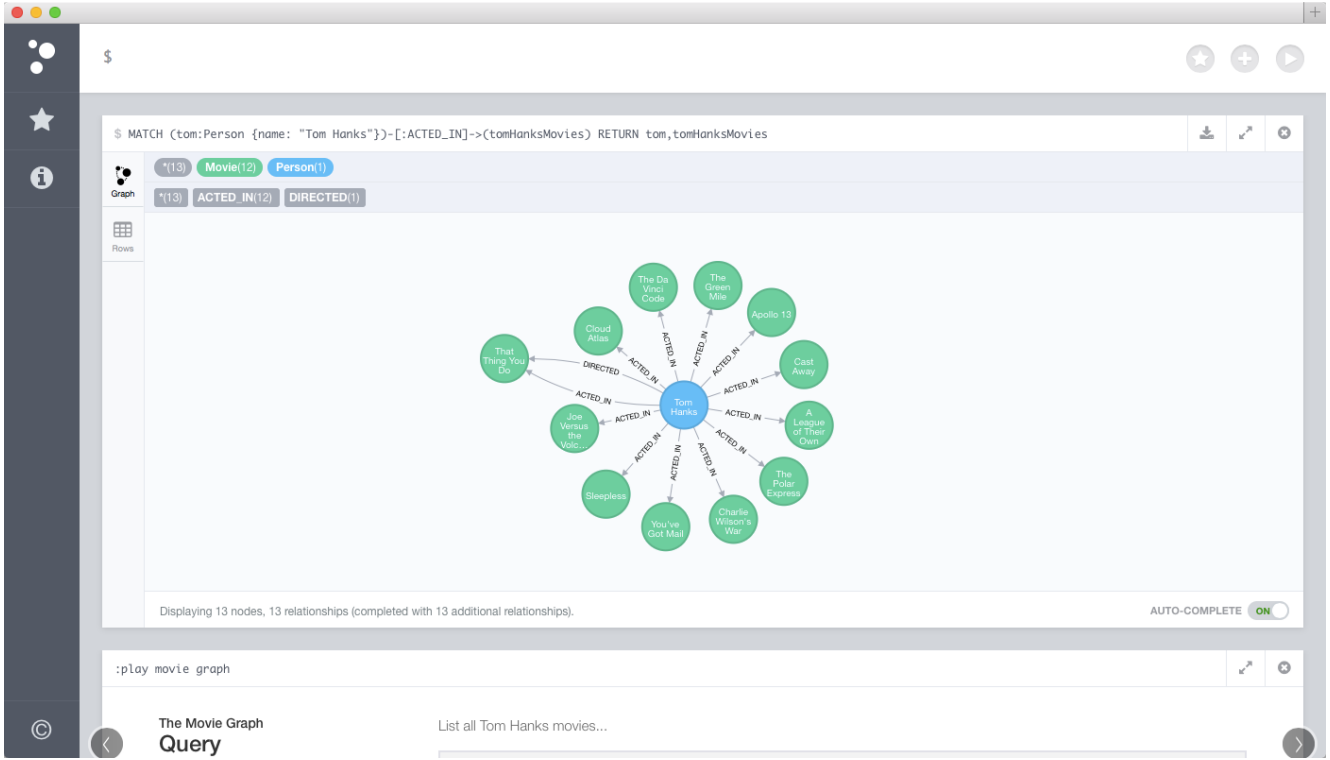
Neo4j Browser – Built-in Learning



RDBMS to Graph – Familiar Examples



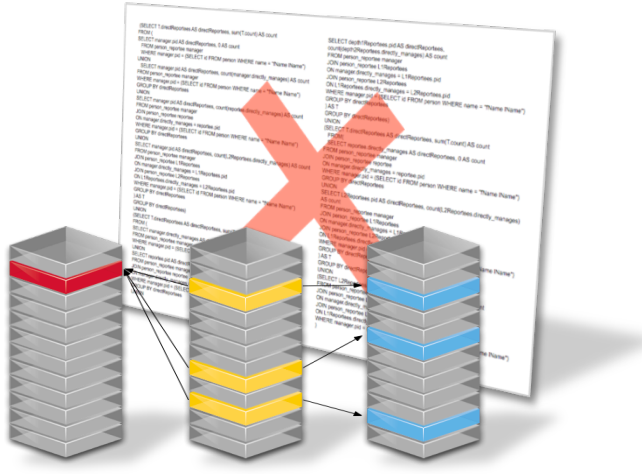
Neo4j Browser – Visualization



Relational DBs Can't Handle Data Relationships Well

- ***Cannot model or store data and relationships*** without complexity
- ***Performance degrades*** with number and levels of relationships, and database size
  - ***Query complexity grows*** with need for JOINs
- ***Adding new types of data and relationships*** requires schema redesign, increasing time to market

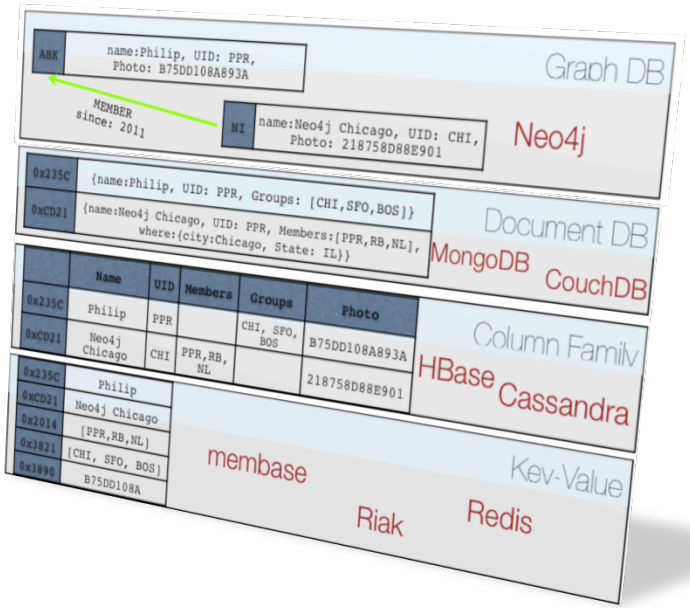
… making traditional databases **inappropriate** when data relationships are valuable in **real-time**



Slow development
Poor performance
Low scalability
Hard to maintain

NoSQL Databases *Don't* Handle Data Relationships

- ***No data structures*** to model or store relationships
- ***No query constructs*** to support data relationships
- ***Relating data requires "JOIN logic"*** in the application
- Additionally, ***no ACID support*** for transactions



… making NoSQL databases **inappropriate** when data relationships are valuable in **real-time**

Graph Database Fundamentals

Express Complex Relationship Queries Easily

Find all reports and how many people they manage, up to 3 levels down

Cypher Query

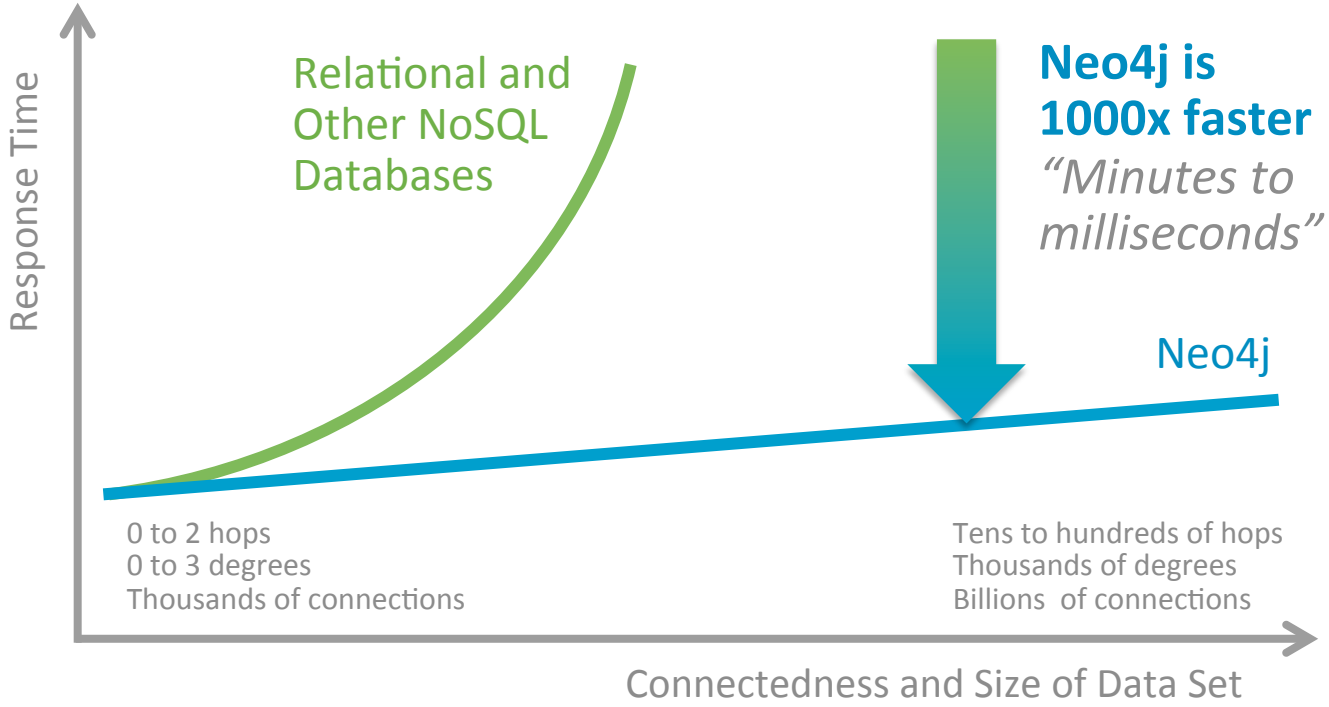
```
MATCH (boss)-[:MANAGES*0..3]->(sub),
      (sub)-[:MANAGES*1..3]->(report)
WHERE boss.name = “John Doe”
RETURN sub.name AS Subordinate,
  count(report) AS Total
```

SQL Query

```
(SELECT T.directReportees AS directReportees, sum(T.count) AS count
FROM (
SELECT manager.pid AS directReportees, 0 AS count
  FROM person_reportee manager
  WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
UNION
  SELECT manager.pid AS directReportees, count(manager.directly_manages) AS count
FROM person_reportee manager
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
UNION
SELECT manager.pid AS directReportees, count(reportee.directly_manages) AS count
FROM person_reportee manager
JOIN person_reportee reportee
ON manager.directly_manages = reportee.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
UNION
SELECT manager.pid AS directReportees, count(L2Reportees.directly_manages) AS count
FROM person_reportee manager
JOIN person_reportee L1Reportees
ON manager.directly_manages = L1Reportees.pid
JOIN person_reportee L2Reportees
ON L1Reportees.directly_manages = L2Reportees.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
) AS T
GROUP BY directReportees)
UNION
(SELECT T.directReportees AS directReportees, sum(T.count) AS count
FROM (
SELECT manager.directly_manages AS directReportees, 0 AS count
FROM person_reportee manager
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
UNION
SELECT reportee.pid AS directReportees, count(reportee.directly_manages) AS count
FROM person_reportee manager
JOIN person_reportee reportee
ON manager.directly_manages = reportee.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
UNION
SELECT depth1Reportees.pid AS directReportees,
count(depth2Reportees.directly_manages) AS count
FROM person_reportee manager
JOIN person_reportee L1Reportees
ON manager.directly_manages = L1Reportees.pid
JOIN person_reportee L2Reportees
ON L1Reportees.directly_manages = L2Reportees.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
) AS T
GROUP BY directReportees)
UNION
(SELECT T.directReportees AS directReportees, sum(T.count) AS count
  FROM(
  SELECT reportee.directly_manages AS directReportees, 0 AS count
FROM person_reportee manager
JOIN person_reportee reportee
ON manager.directly_manages = reportee.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
UNION
SELECT L2Reportees.pid AS directReportees, count(L2Reportees.directly_manages)
AS count
FROM person_reportee manager
JOIN person_reportee L1Reportees
ON manager.directly_manages = L1Reportees.pid
JOIN person_reportee L2Reportees
ON L1Reportees.directly_manages = L2Reportees.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
) AS T
GROUP BY directReportees)
UNION
(SELECT L2Reportees.directly_manages AS directReportees, 0 AS count
FROM person_reportee manager
JOIN person_reportee L1Reportees
ON manager.directly_manages = L1Reportees.pid
JOIN person_reportee L2Reportees
ON L1Reportees.directly_manages = L2Reportees.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
)
```

Real-Time Query Performance

Graph Versus Relational and Other NoSQL Databases



Value from Data Relationships

Common Use Cases

Internal Applications

Master Data Management

Network and
IT Operations

Fraud Detection

Customer-Facing Applications

Real-Time Recommendations

Graph-Based Search

Identity and
Access Management

There Are Lots of Ways to Easily Learn

community

Graph Academy
Learn. Graph. Deploy.

On-site
Training

BOOKS

Documentation

GraphGist

Built-in
Guides

Online Training

Slide Bucket

Web Example: CAPTCHA

Largest Ecosystem of Graph Enthusiasts

- 1,000,000+ downloads
- 27,000+ education registrants
- 25,000+ Meetup members in 29 countries
- 100+ technology and service partners
- 170+ enterprise subscription customers including 50+ Global 2000 companies

Value from Data Relationships

Common Use Cases

Internal Applications

Master Data Management

Network and
IT Operations

Fraud Detection

Customer-Facing Applications

Real-Time Recommendations

Graph-Based Search

Identity and
Access Management