



Intro to Neo4j and Graph Databases

Globant Think Big
Buenos Aires
August 2016

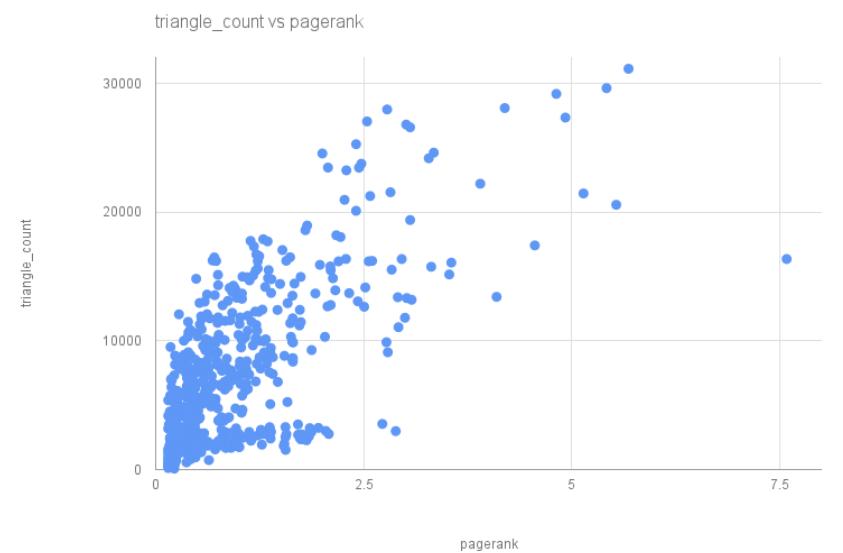
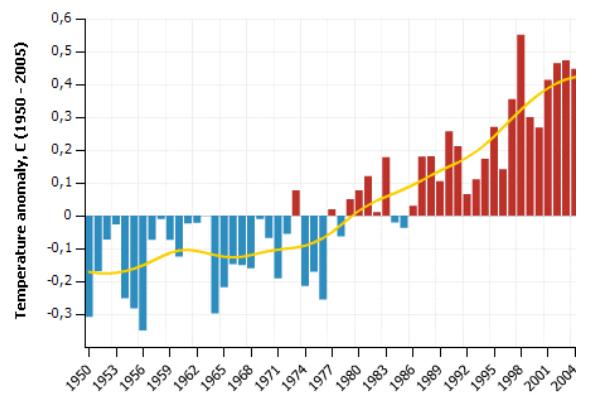
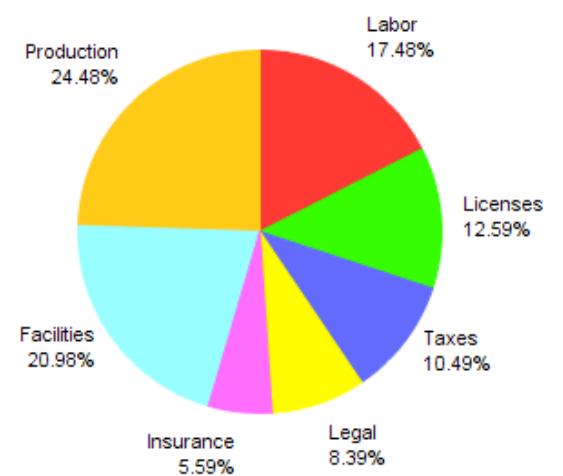
ABOUT ME

- William Lyon
- Developer Relations Engineer @neo4j
- <http://neo4j.com/developer>

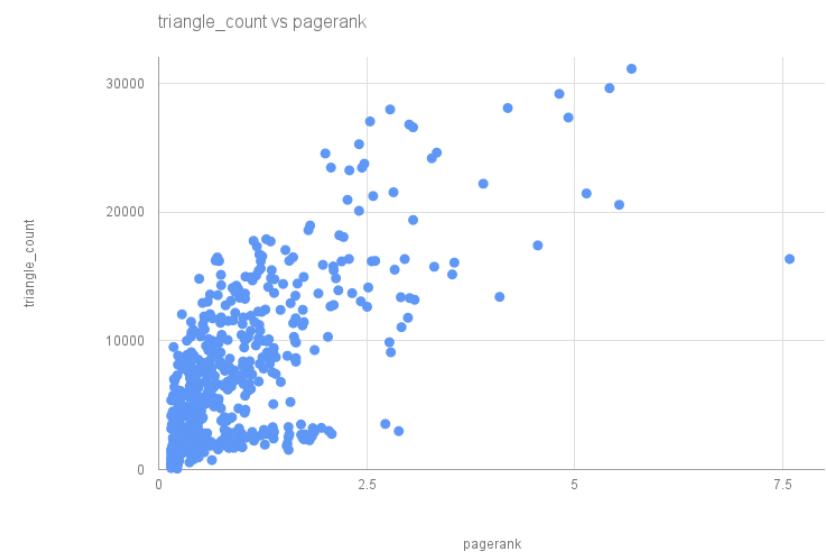
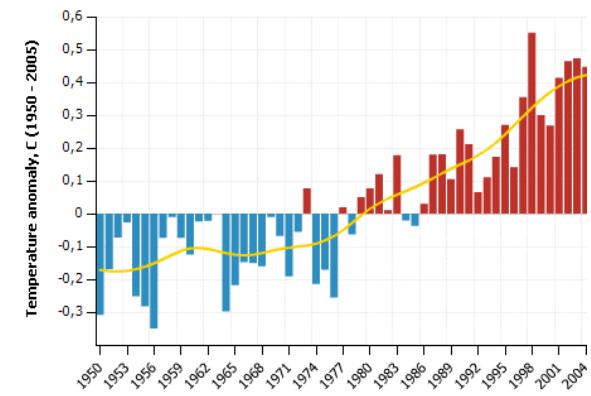
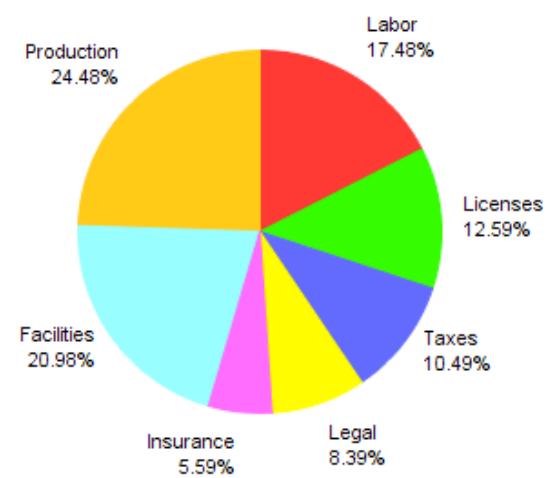


will@neo4j.com
@lyonwj

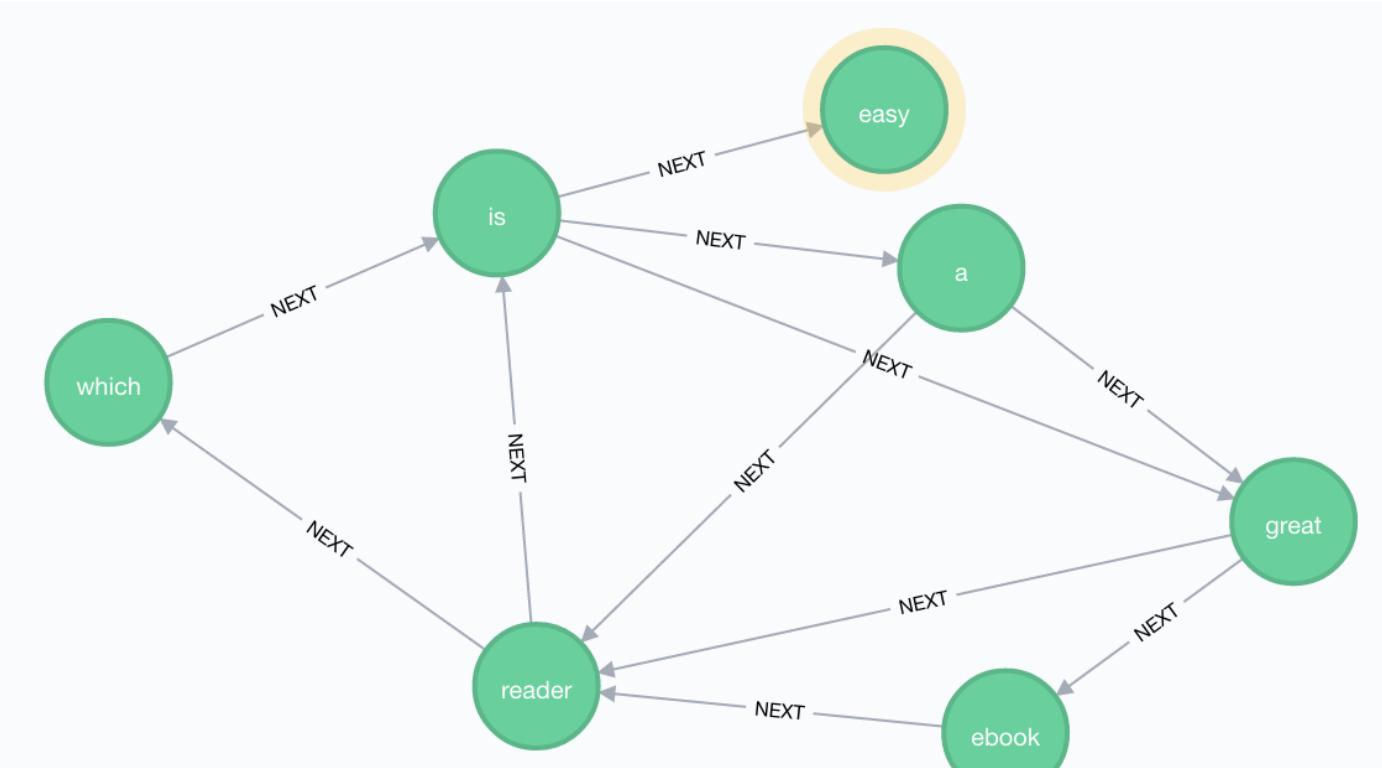
Chart

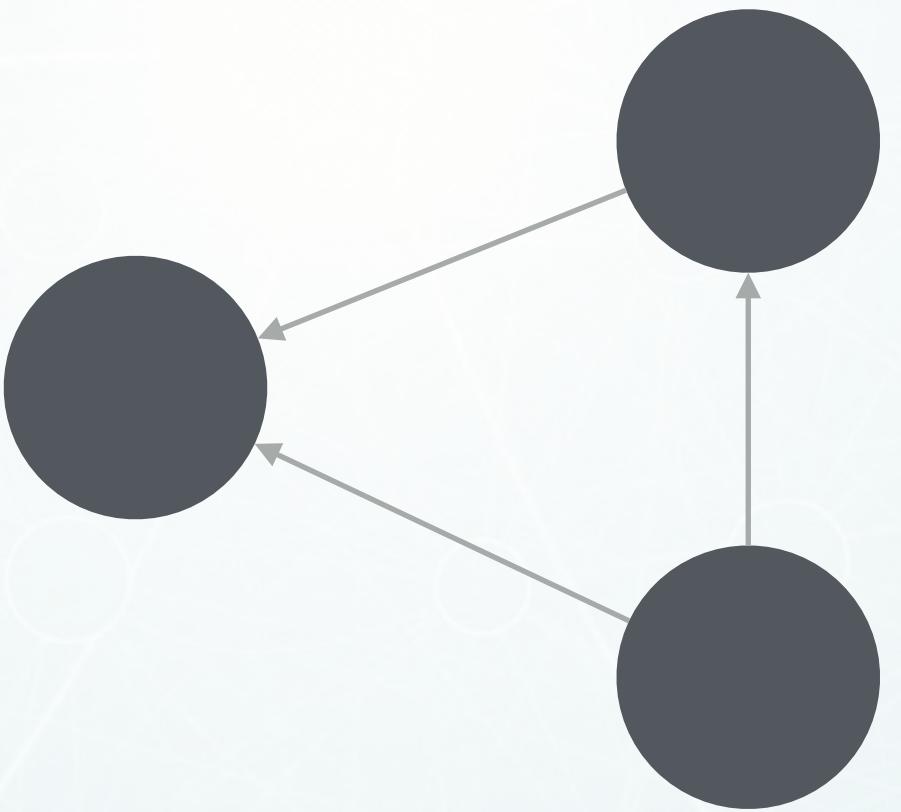


Chart

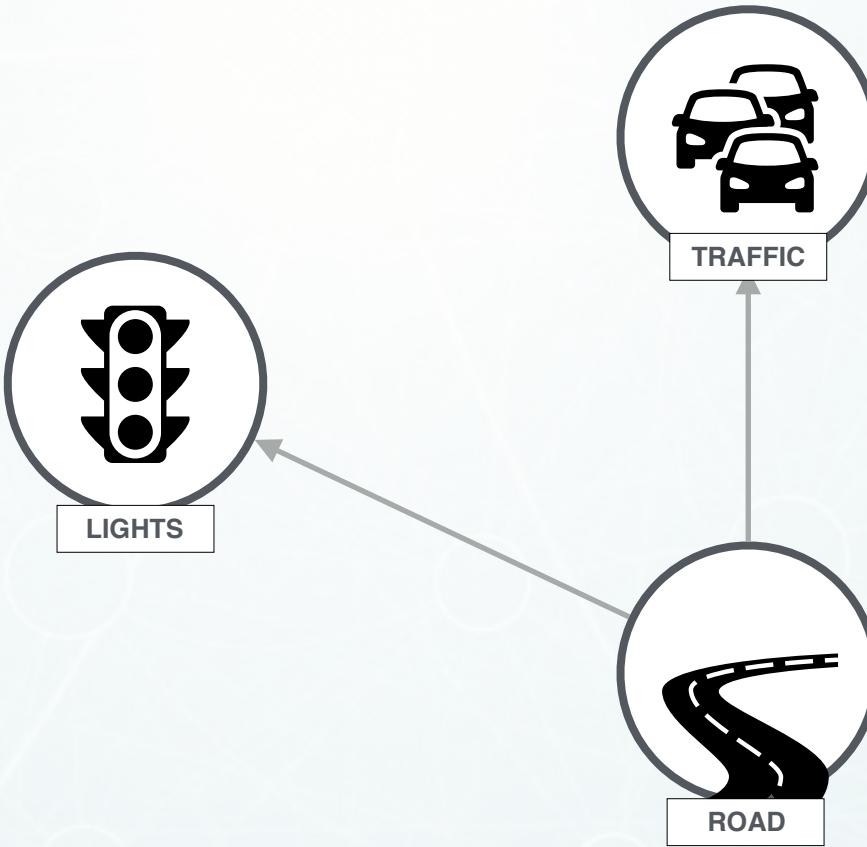


Graph

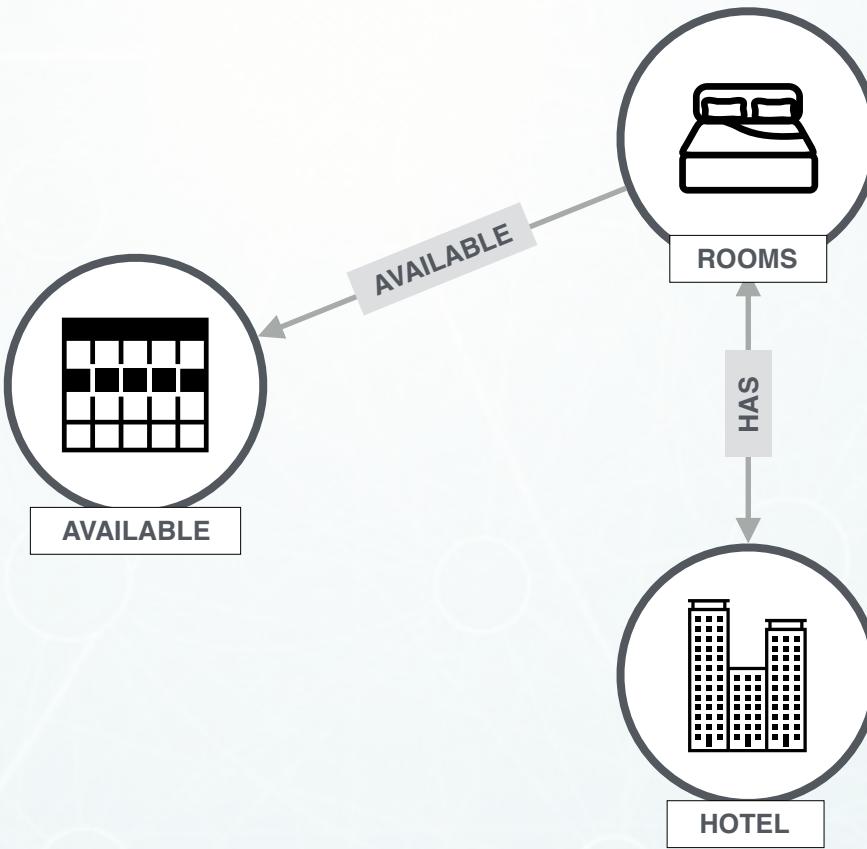




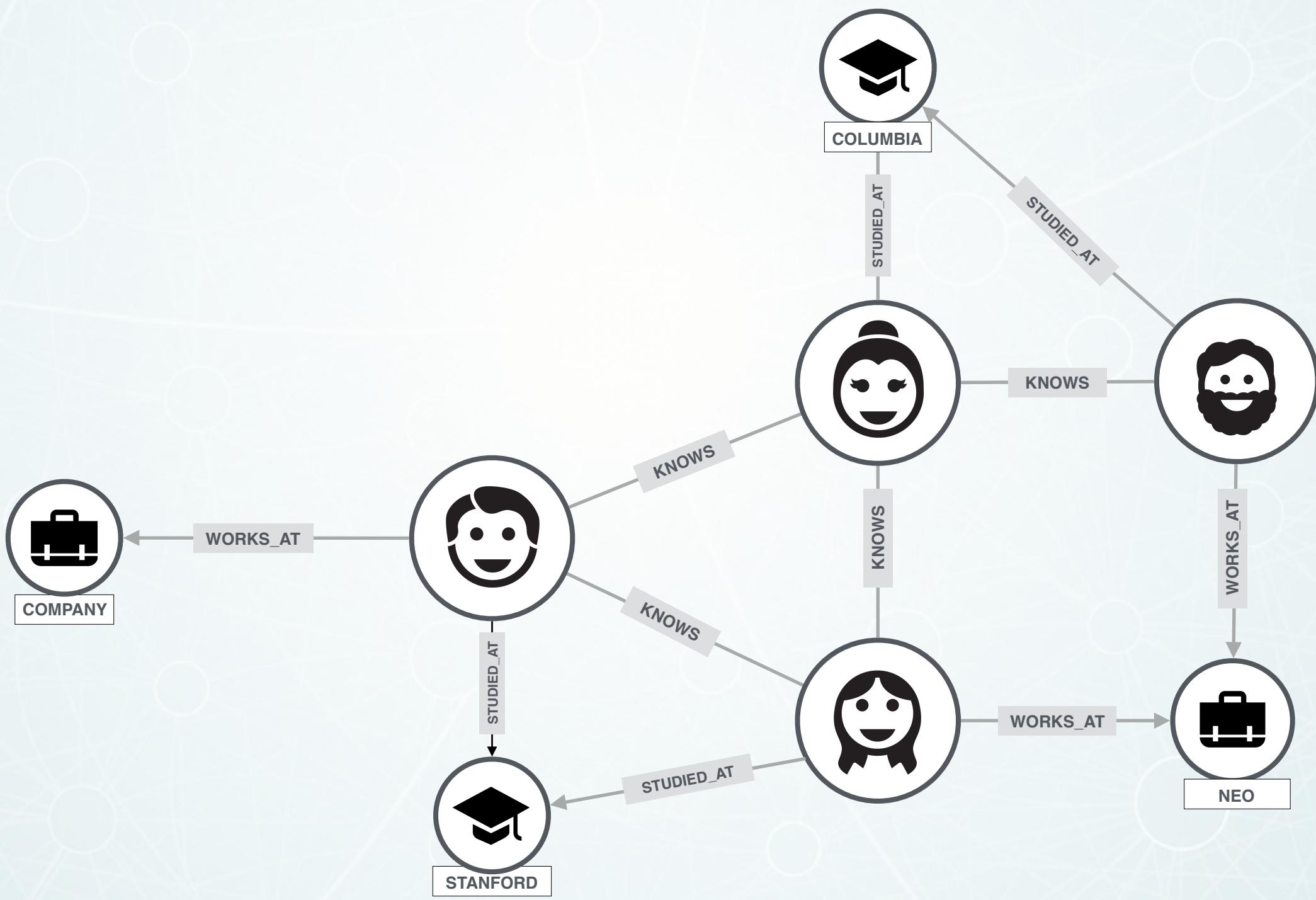
A Graph Is Connected Data



A Graph Is Connected Data



A Graph Is Connected Data

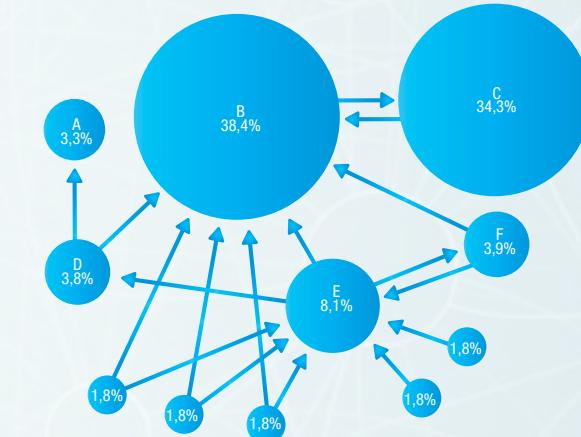


A Graph Is Connected Data

Use of Graphs has created some of the most successful companies in the world

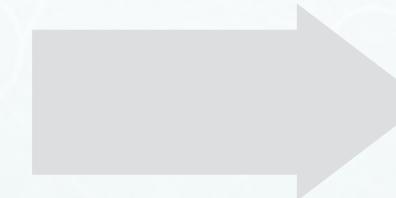


Google

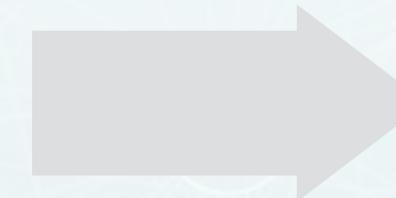
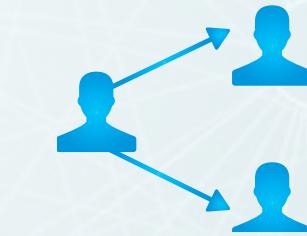


monster

Find better.™



LinkedIn



PayPal





Today we see graph-projects in virtually every industry



Finance



Social networks



HR &
Recruiting



Manufacturing
& Logistics



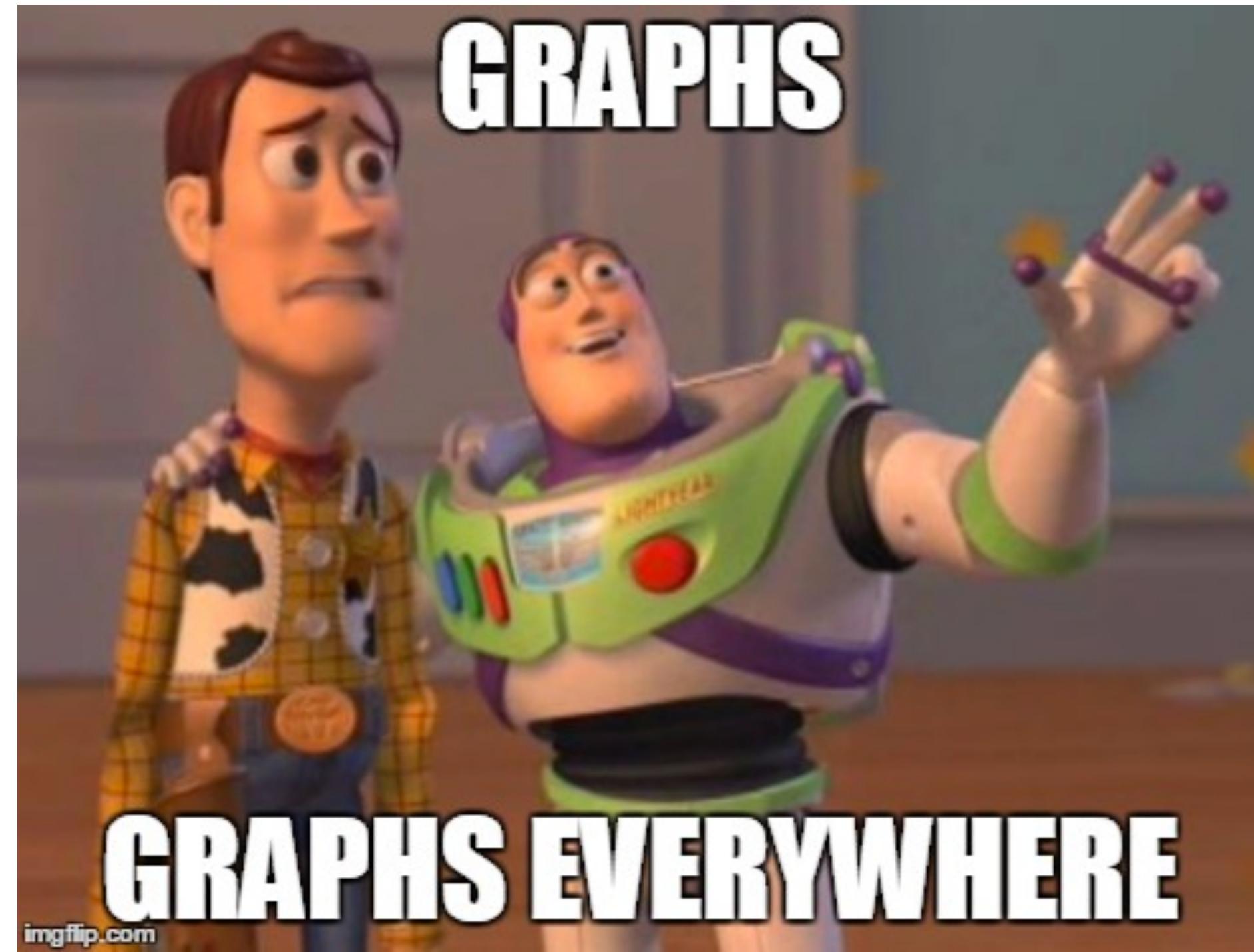
Health Care



Telco



Retail



NEO4j USE CASES

Real Time Recommendations

Master Data Management

Fraud Detection

Graph Based Search

Network & IT-Operations

Identity & Access Management

NEO4j USE CASES

Real Time Recommendations

Master Data Management

Fraud Detection

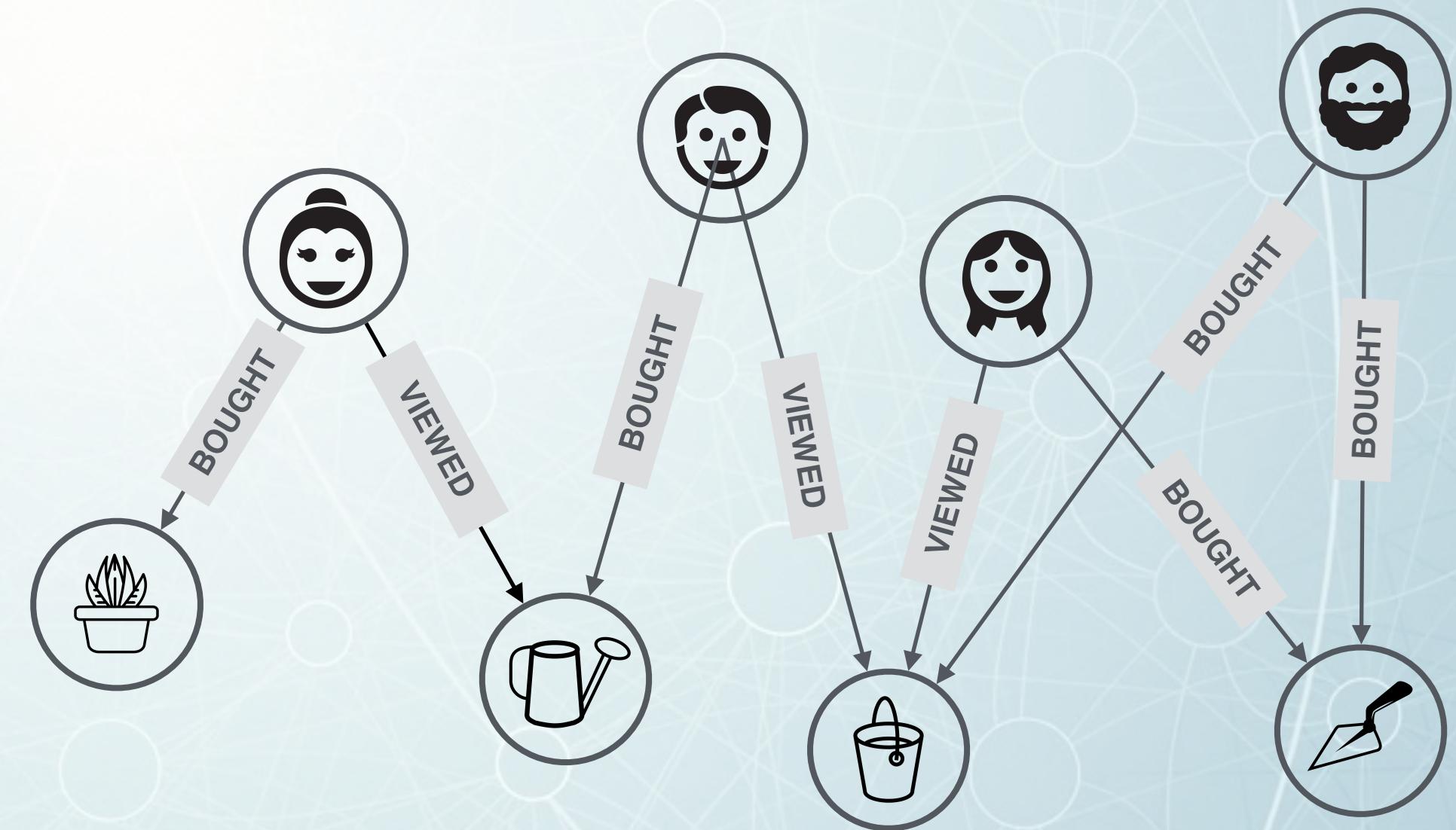
Graph Based Search

Network & IT-Operations

Identity & Access Management



GRAPH THINKING: Real Time Recommendations



NEO4j USE CASES

Real Time Recommendations

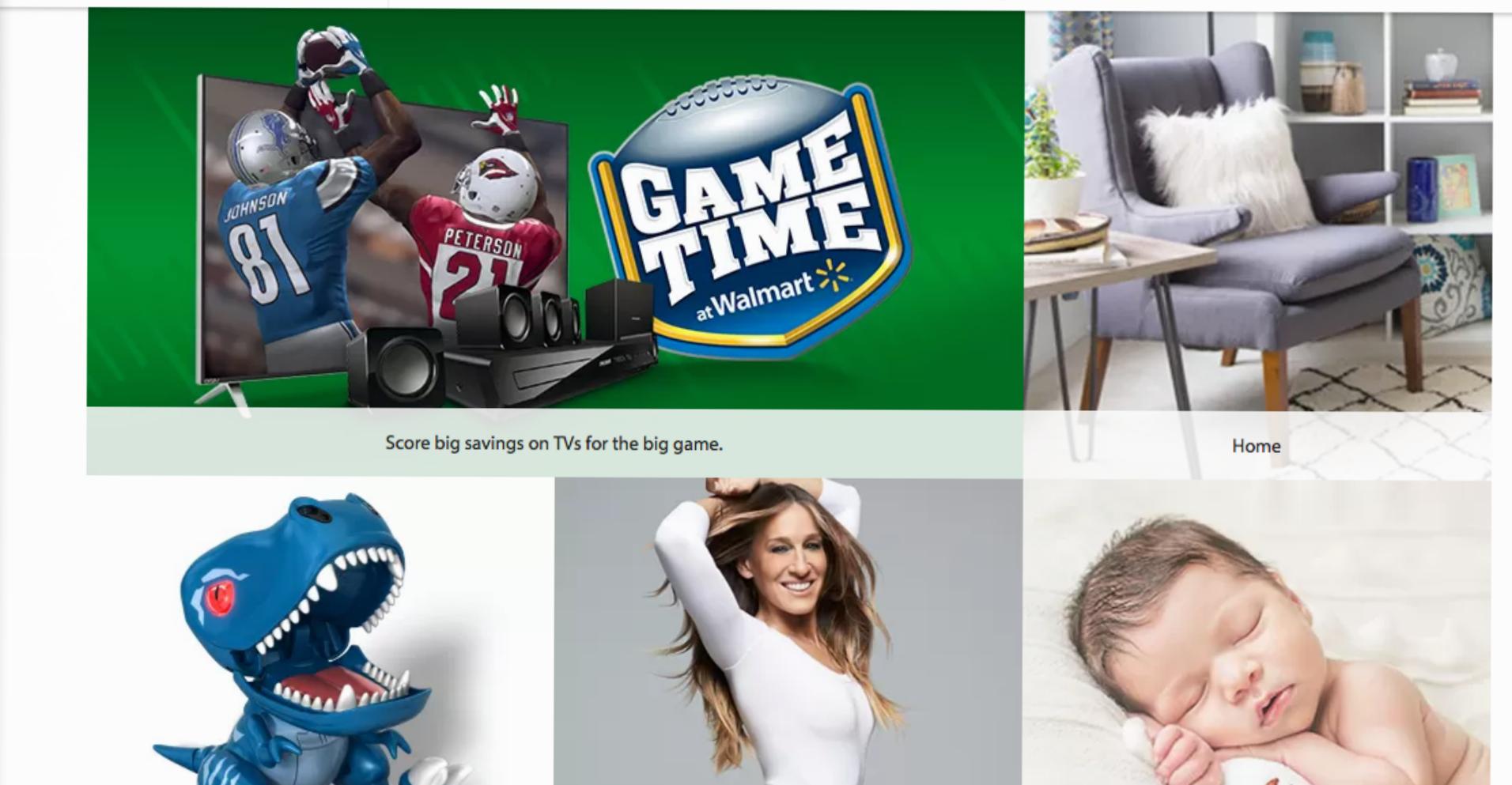
Master Data Management

Fraud Detection

Graph Based Search

Network & IT-Operations

Identity & Access Management



“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

Rollbacks

NEO4j USE CASES

Real Time Recommendations

Master Data Management

Fraud Detection

Graph Based Search

Network & IT-Operations

Identity & Access Management



GRAPH THINKING: Master Data Management



NEO4j USE CASES

Real Time Recommendations

Master Data Management

Fraud Detection

Graph Based Search

Network & IT-Operations

Identity & Access Management



Neo4j is the heart of Cisco HMP: used for governance and single source of truth and a one-stop shop for all of Cisco's hierarchies.

NEO4j USE CASES

Real Time Recommendations

Master Data Management

Fraud Detection

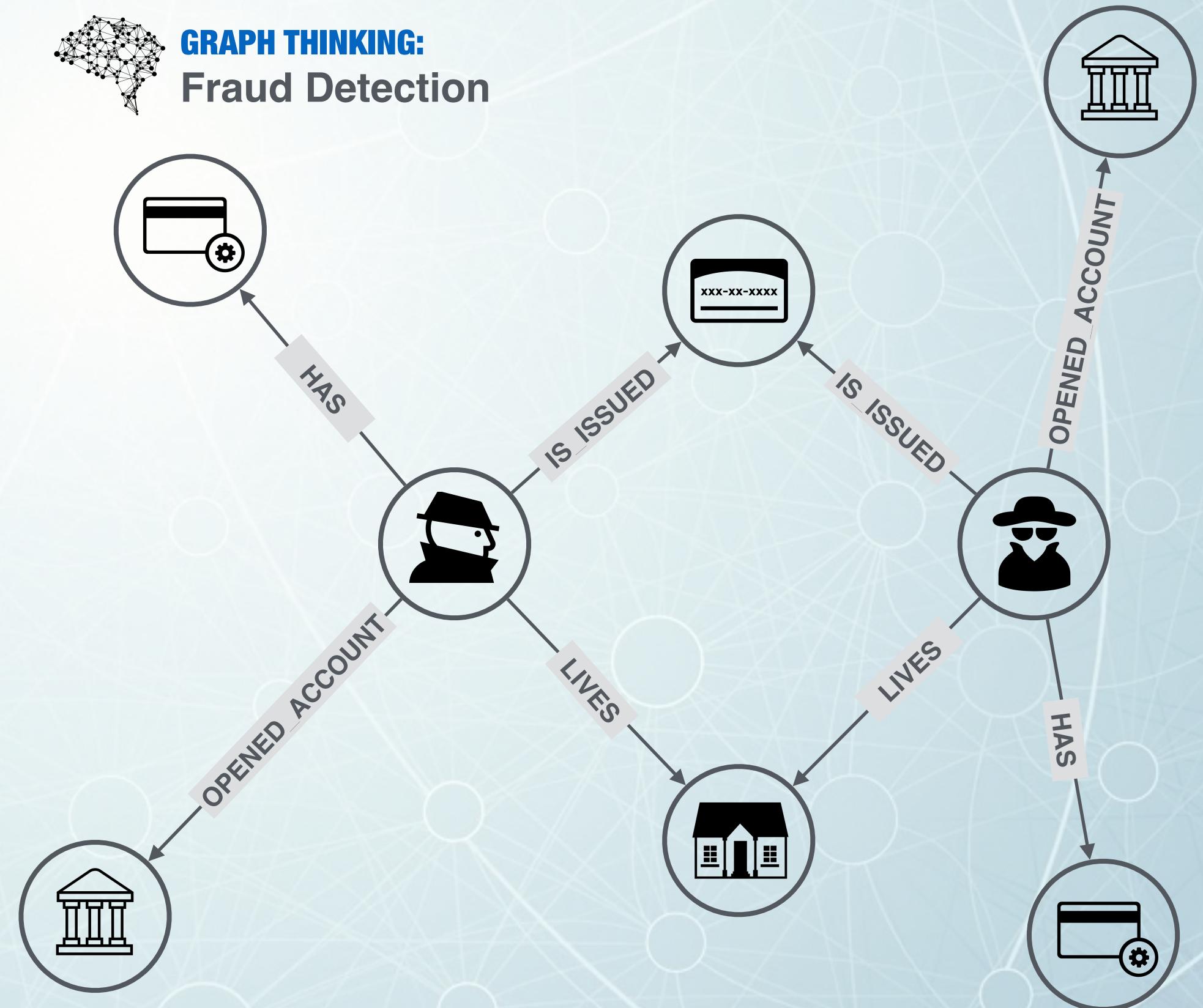
Graph Based Search

Network & IT-Operations

Identity & Access Management



GRAPH THINKING: Fraud Detection



NEO4j USE CASES

Real Time Recommendations

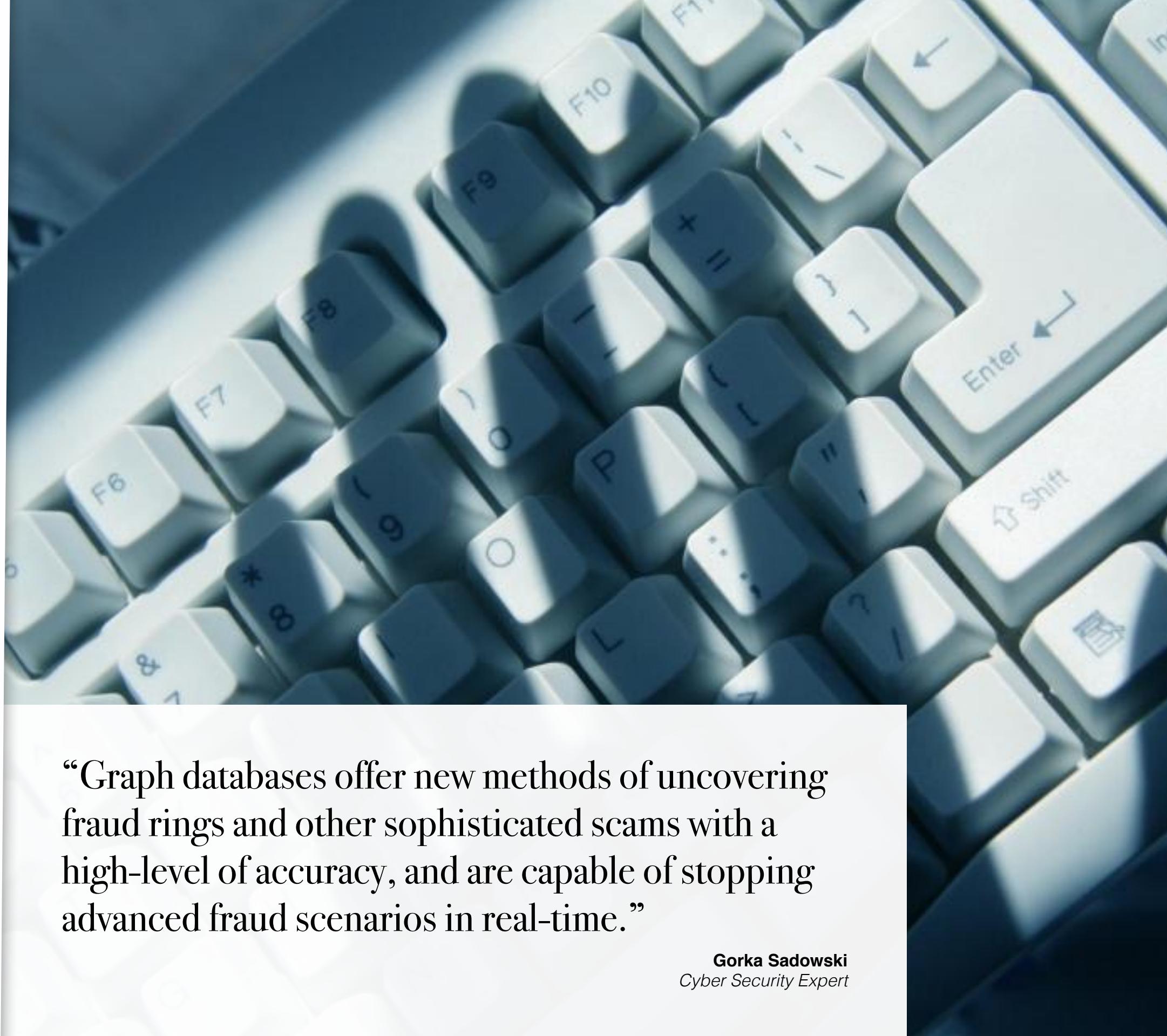
Master Data Management

Fraud Detection

Graph Based Search

Network & IT-Operations

Identity & Access Management



“Graph databases offer new methods of uncovering fraud rings and other sophisticated scams with a high-level of accuracy, and are capable of stopping advanced fraud scenarios in real-time.”

Gorka Sadowski
Cyber Security Expert

NEO4j USE CASES

Real Time Recommendations

Master Data Management

Fraud Detection

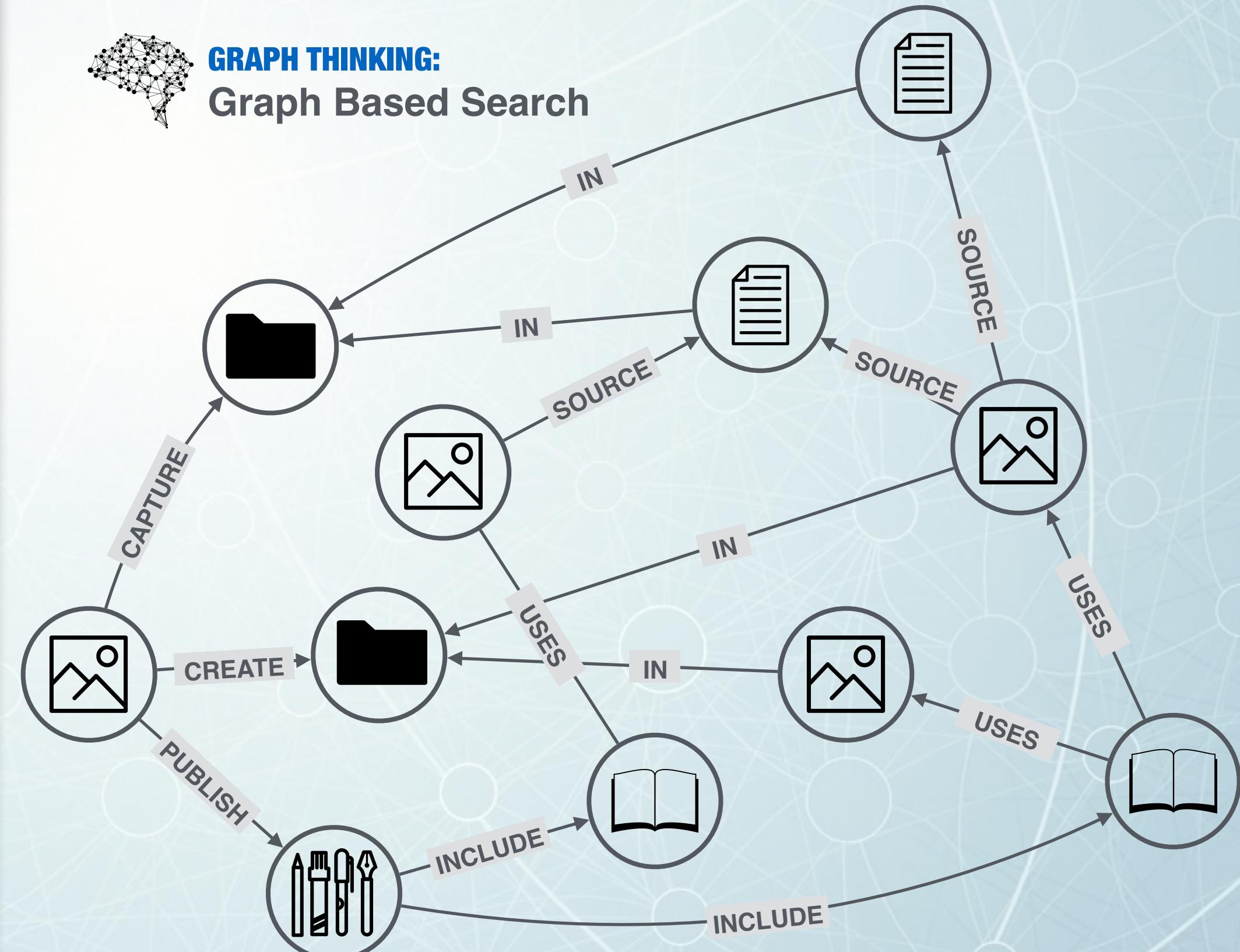
Graph Based Search

Network & IT-Operations

Identity & Access Management



GRAPH THINKING: Graph Based Search



NEO4j USE CASES

Real Time Recommendations

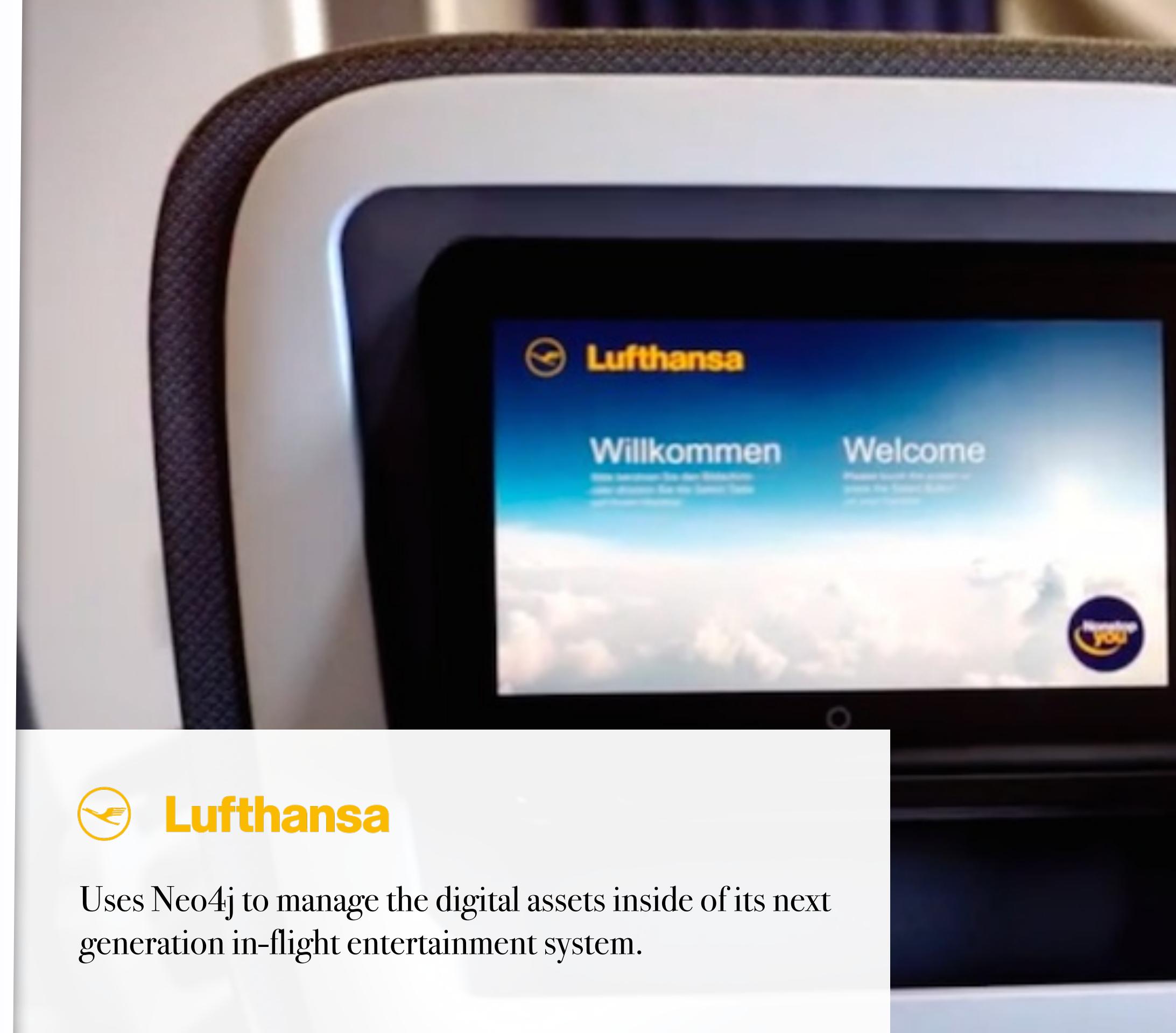
Master Data Management

Fraud Detection

Graph Based Search

Network & IT-Operations

Identity & Access Management



Lufthansa

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

NEO4j USE CASES

Real Time Recommendations

Master Data Management

Fraud Detection

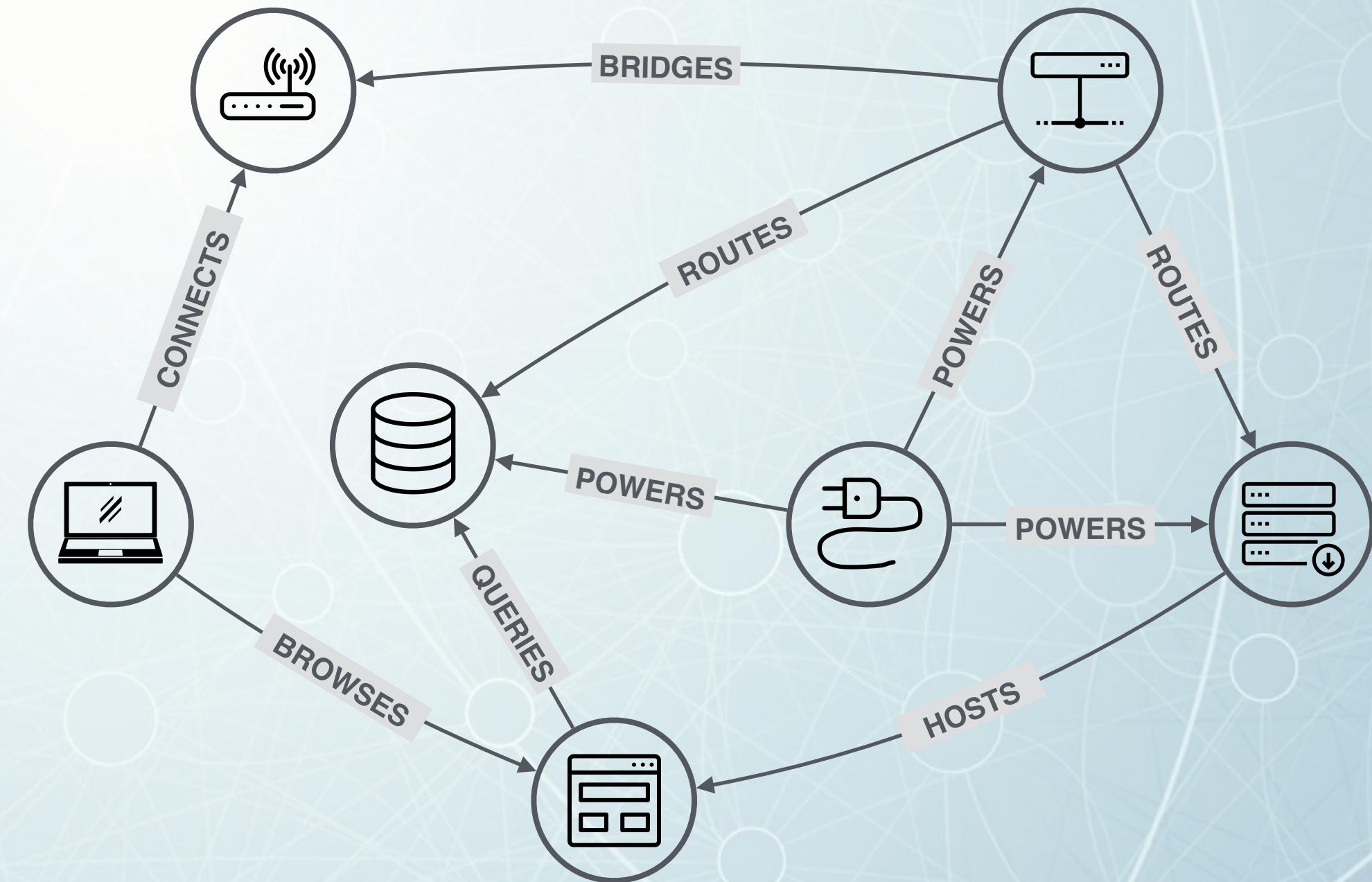
Graph Based Search

Network & IT-Operations

Identity & Access Management



GRAPH THINKING: Network & IT-Operations



NEO4j USE CASES

Real Time Recommendations

Master Data Management

Fraud Detection

Graph Based Search

Network & IT-Operations

Identity & Access Management



Uses Neo4j for network topology analysis
for big telco service providers

NEO4j USE CASES

Real Time Recommendations

Master Data Management

Fraud Detection

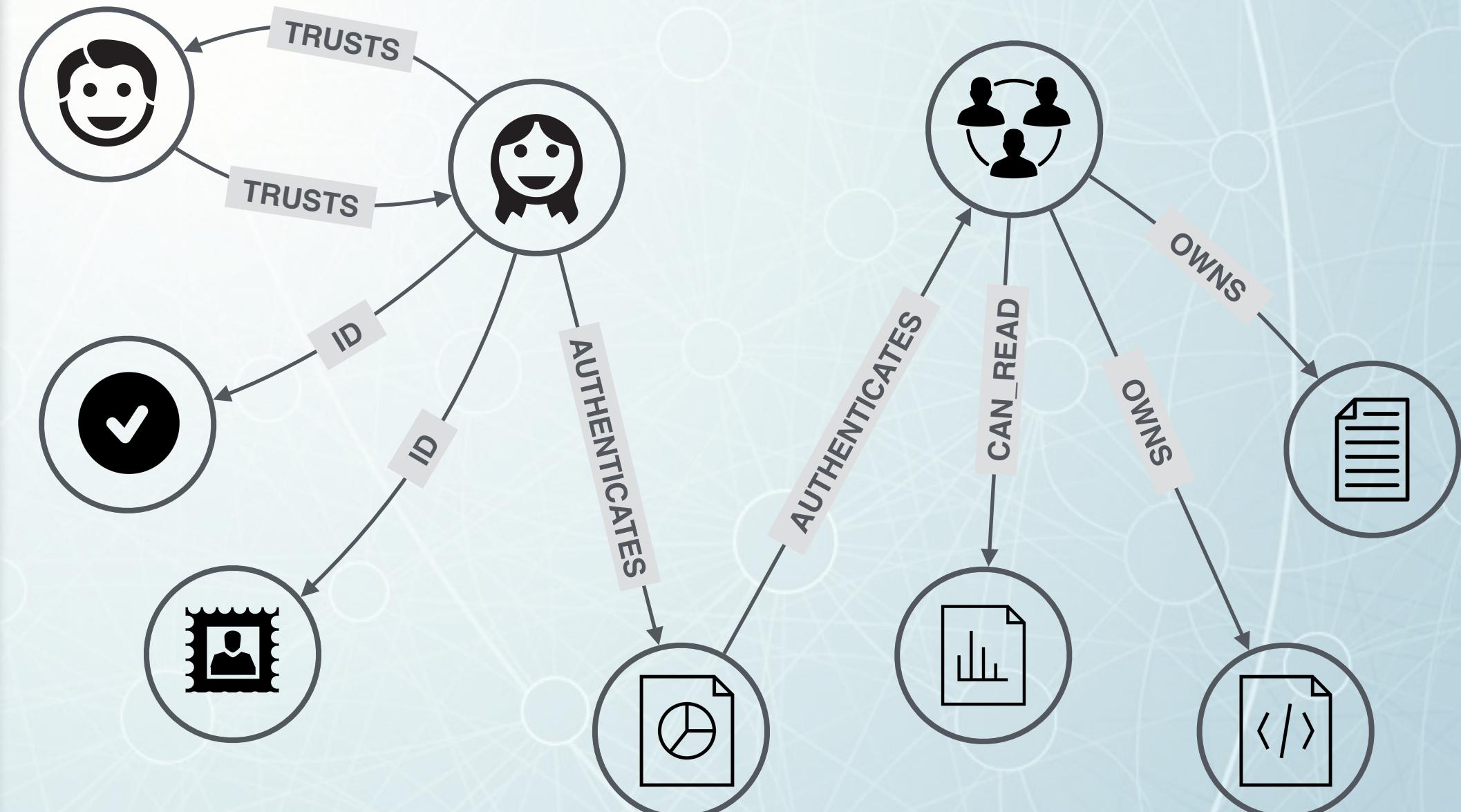
Graph Based Search

Network & IT-Operations

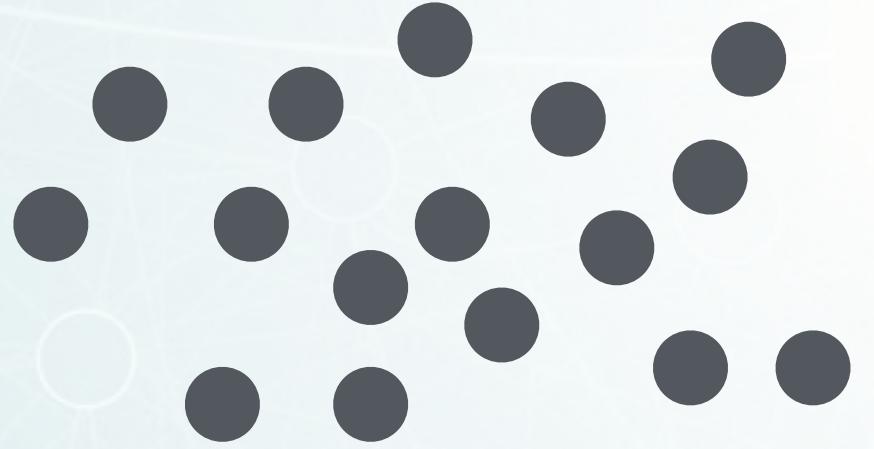
Identity & Access Management



GRAPH THINKING: Identity And Access Management



A way of representing data

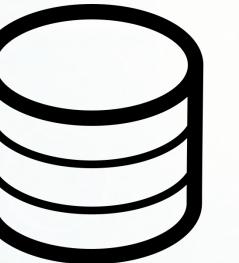
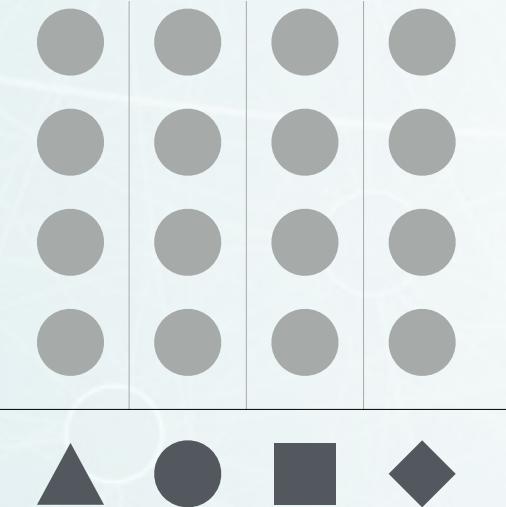


DATA

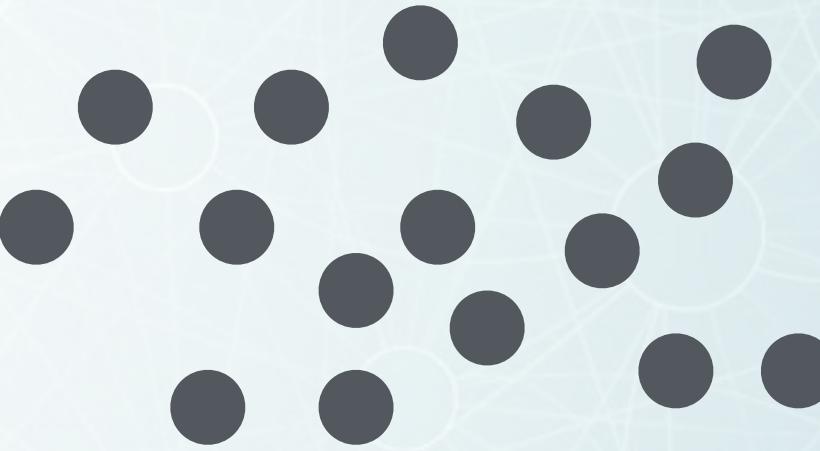


DATA

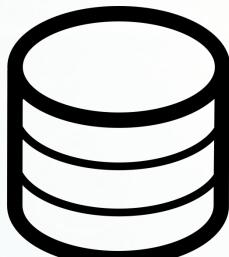
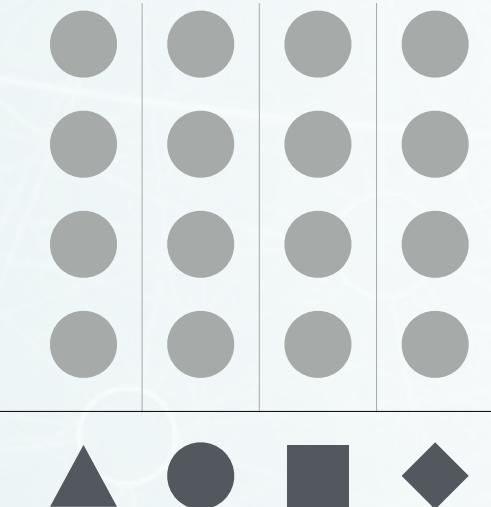
A way of representing data



Relational Database



A way of representing data



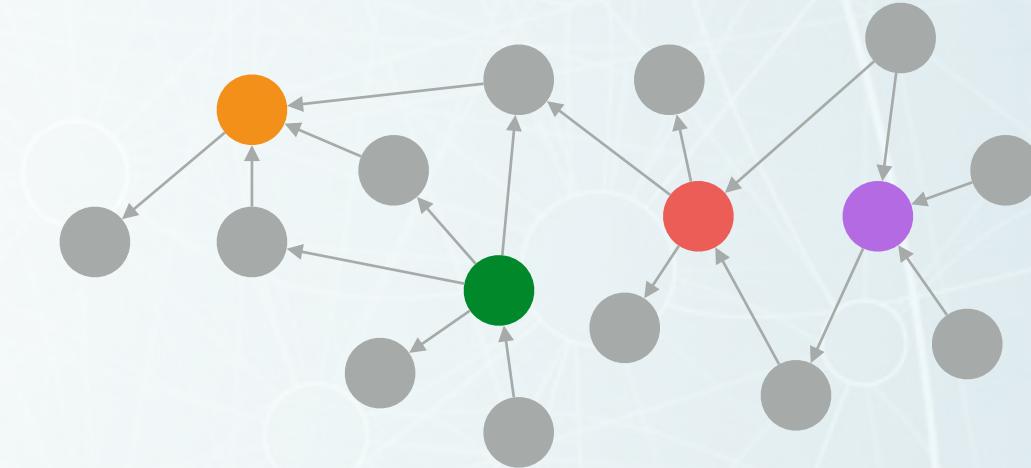
Relational Database

Good for:

- Well-understood data structures that don't change too frequently
- Known problems involving discrete parts of the data, or minimal connectivity



Graph Database



Good for:

- Dynamic systems: where the data topology is difficult to predict
- Dynamic requirements: that evolve with the business
- Problems where the relationships in data contribute meaning & value

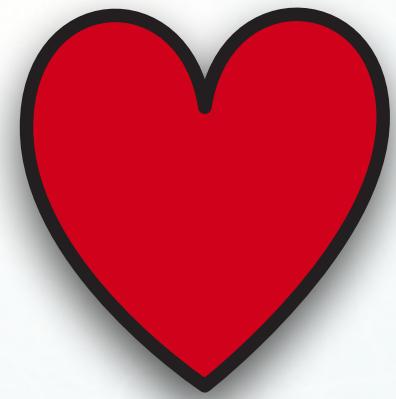
THE PROPERTY GRAPH DATA MODEL



Ann Loves Dan



Ann

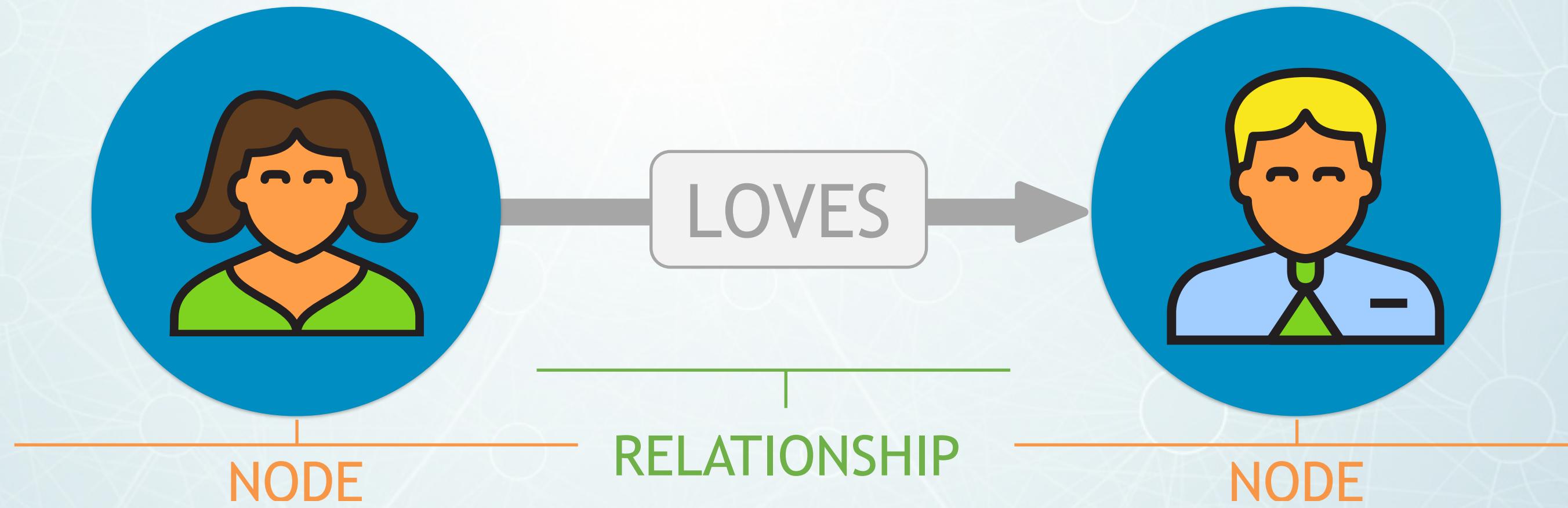


Loves

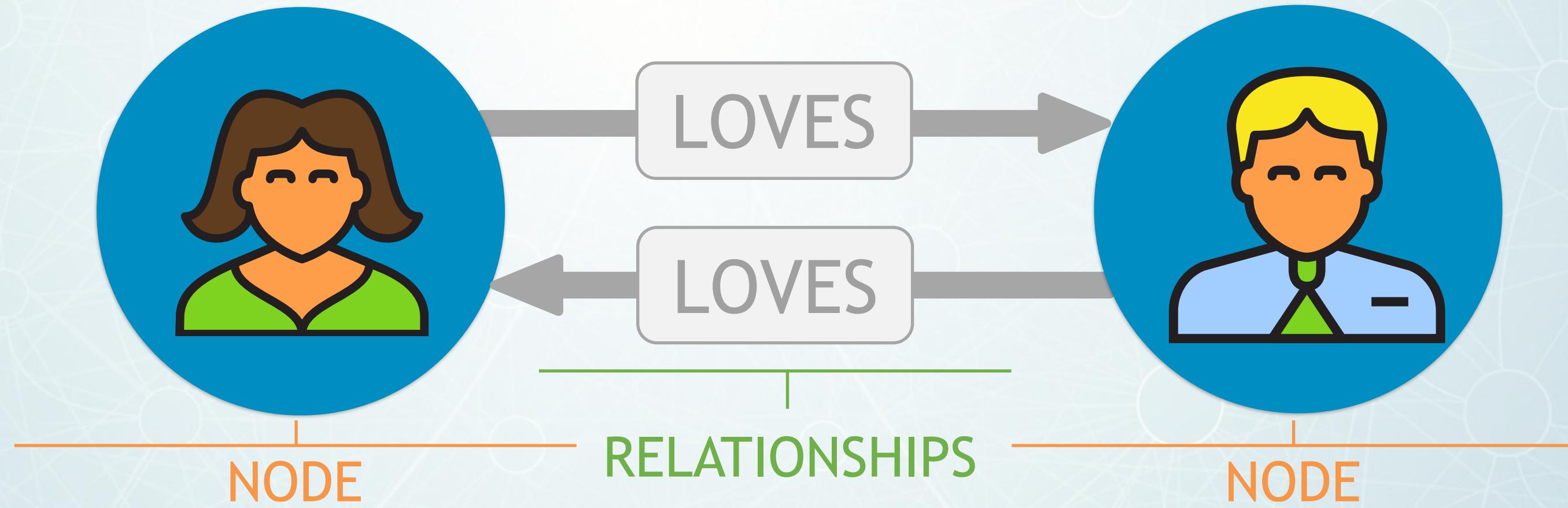


Dan

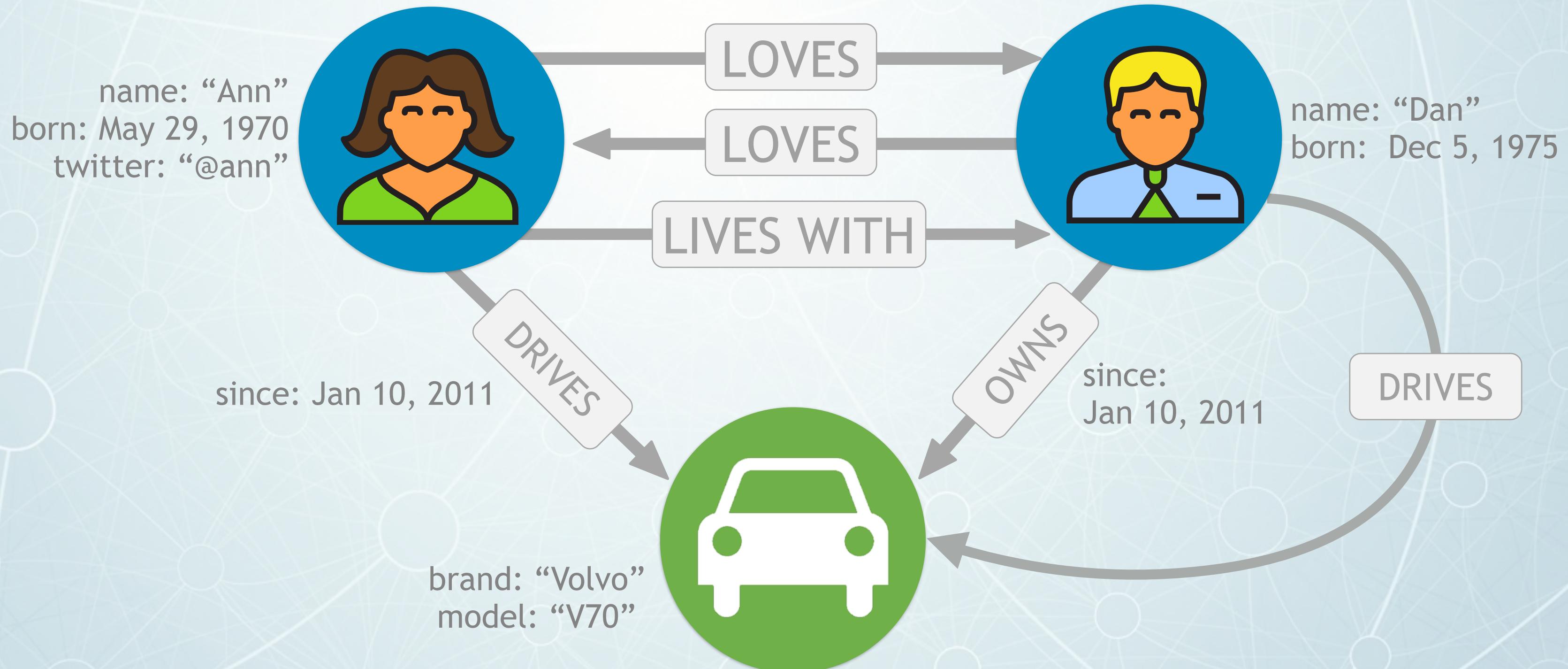
Ann Loves Dan



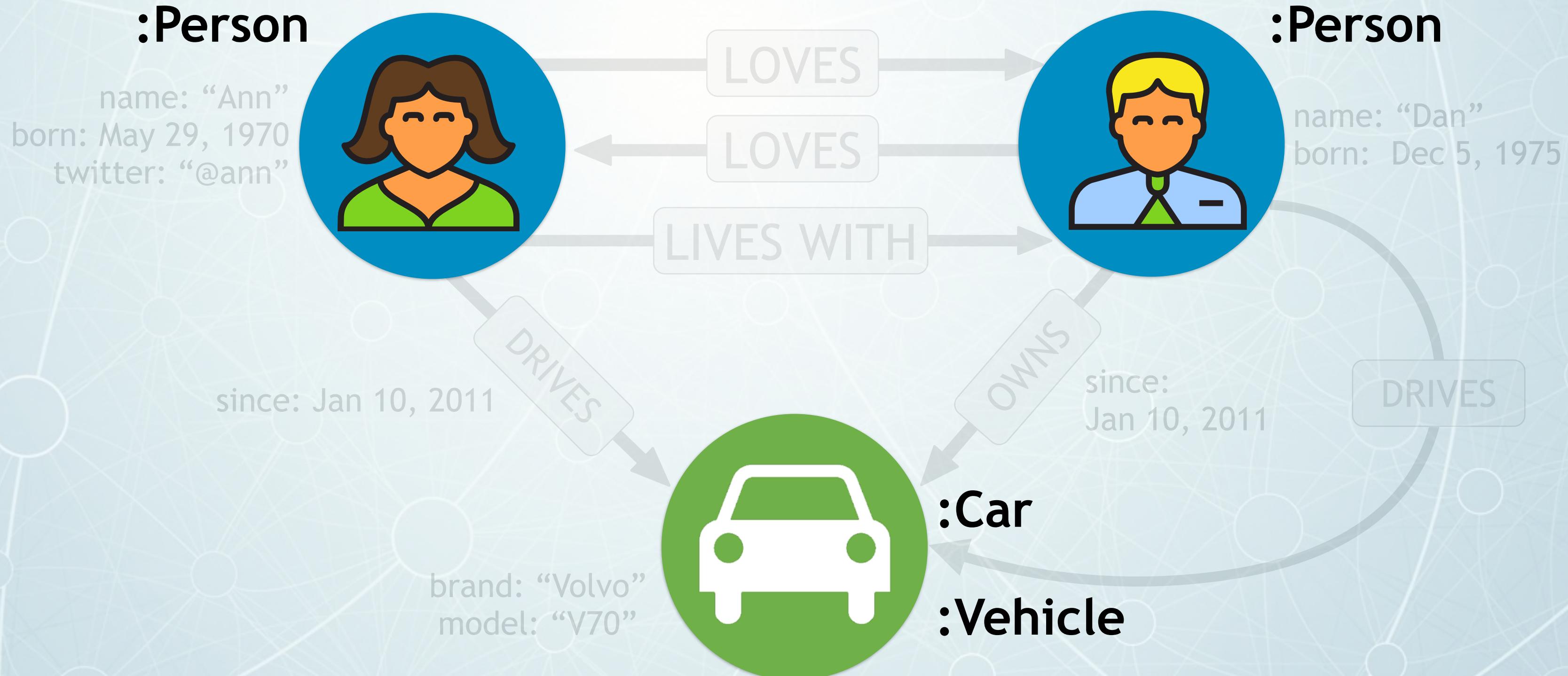
Relationships are Directional



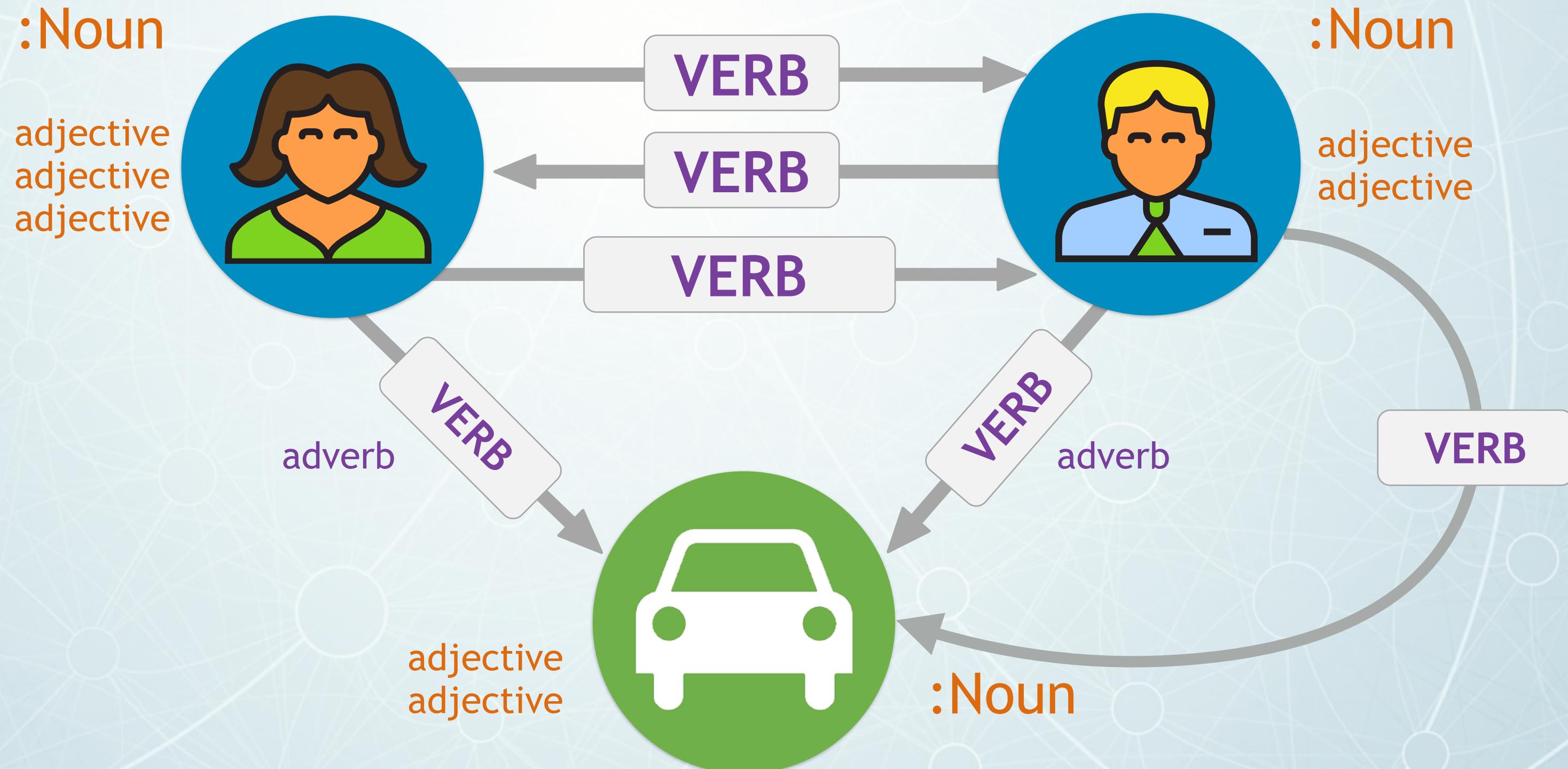
Detailed Property Graph



Labeled Property Graph



Mapping to Languages



WHY GRAPHS?





Intuitivness

Speed

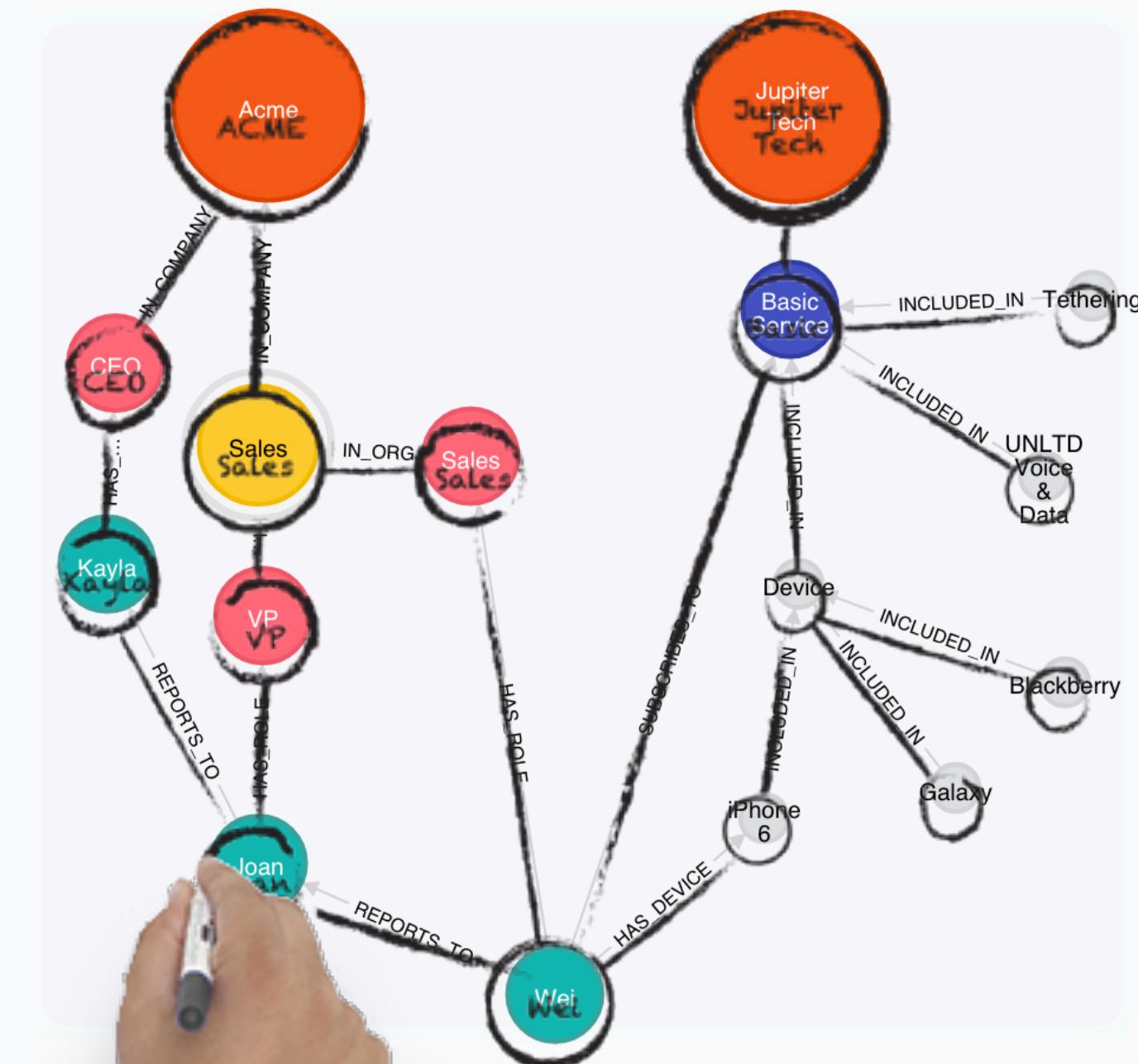
Agility

Intuitiveness

Speed

Agility

Intuitiveness





Intuitivness

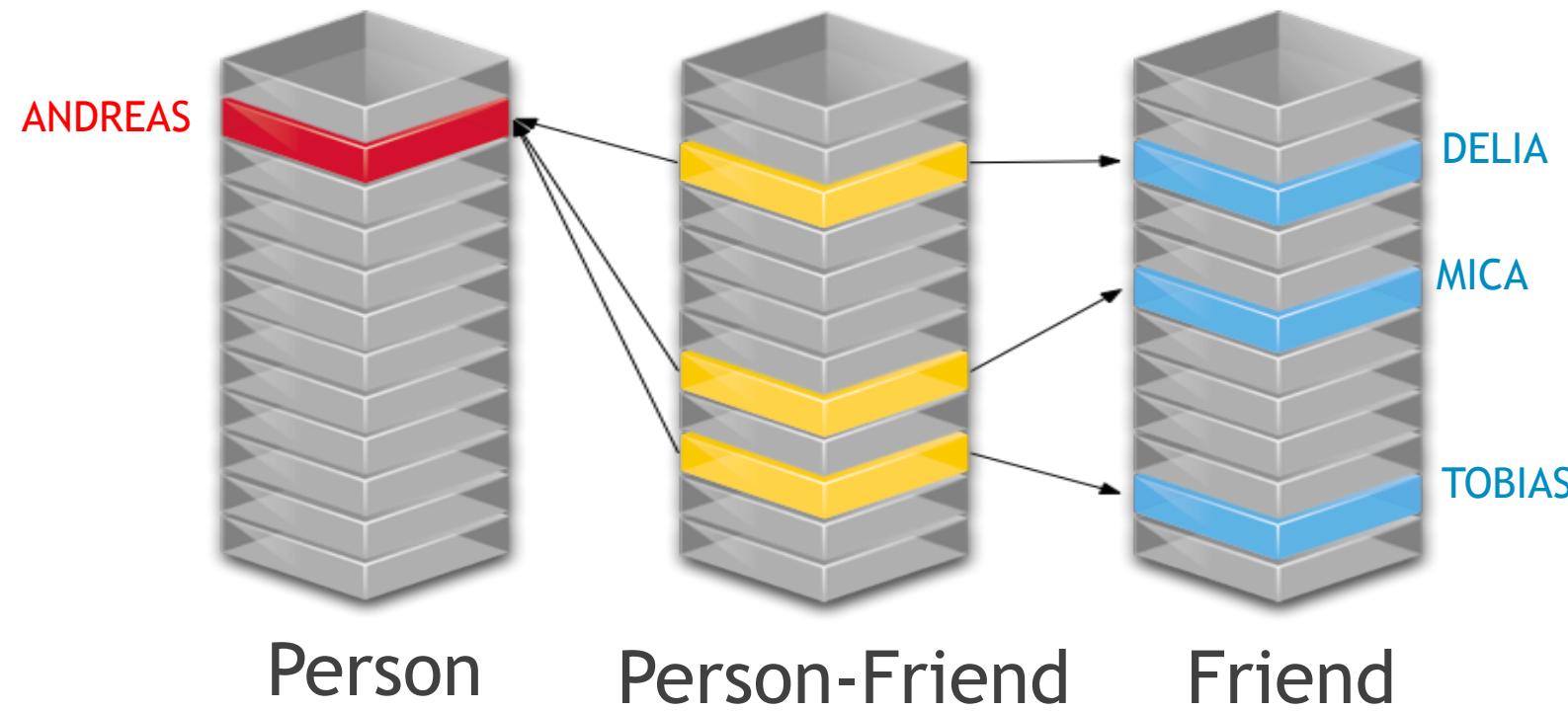
Speed

Agility

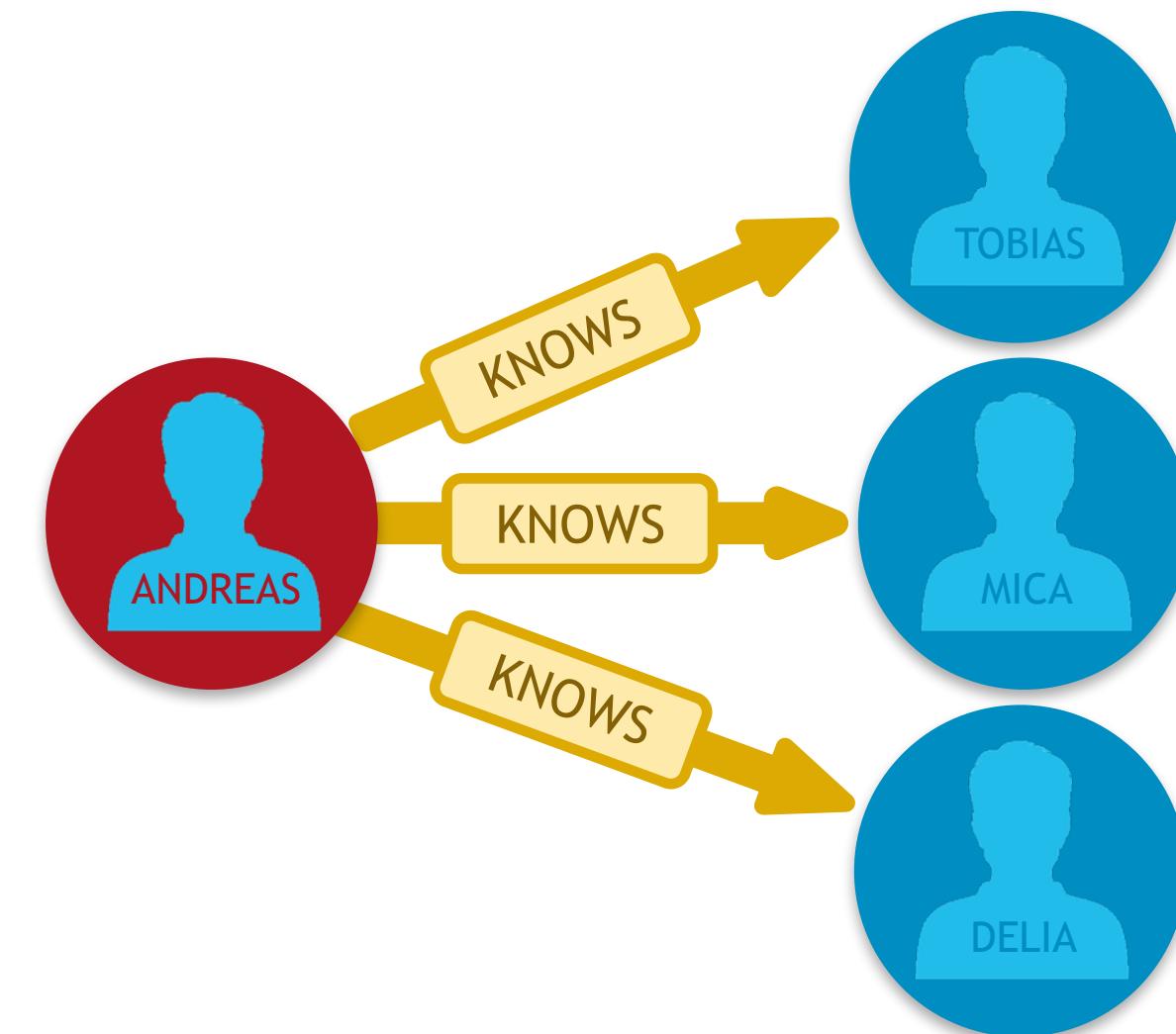
Relational Versus Graph Models



Relational Model



Graph Model



Index free adjacency

Speed



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

- Volker Pacher, Senior Developer

“Minutes to milliseconds” performance
Queries up to 1000x faster than RDBMS or other NoSQL





Intuitivness

Speed

Agility

Agility =

A Naturally Adaptive Model



**A Query Language Designed
for Connectedness**

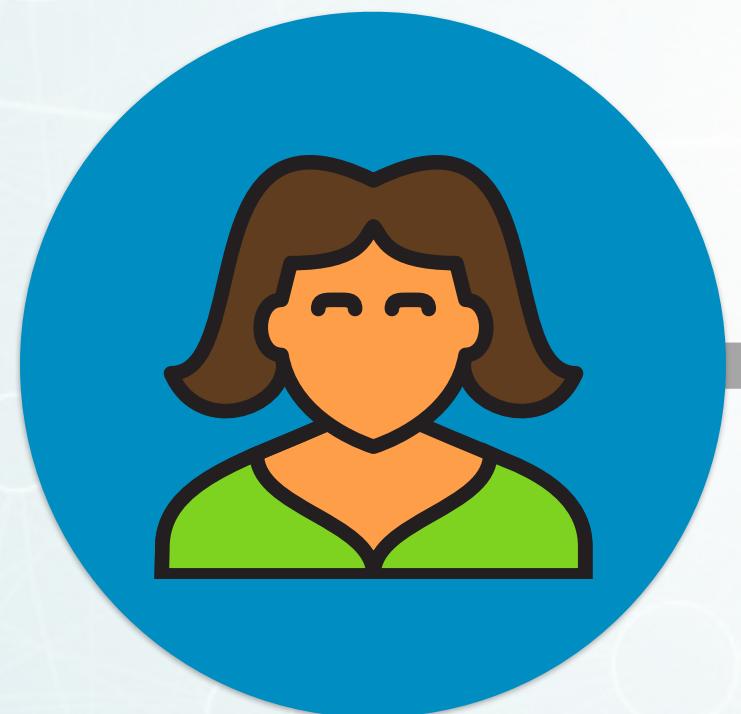
CYPHER

SQL for graphs



(Very) Brief Cypher Tutorial

Creating the Data



NODE



NODE

```
CREATE (:Person { name:“Ann”}) - [:LOVES]-> (:Person { name:“Dan”})
```

—
|
LABEL PROPERTY
—
|
LABEL PROPERTY

Cypher

Typical Complex SQL Join

```
(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
) AS T
GROUP BY directReportees)
```

```
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
) AS T
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")
GROUP BY directReportees
)
```

The Same Query using Cypher

```
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
```

Project Impact

Less time writing queries

- More time understanding the answers
- Leaving time to ask the next question

Less time debugging queries:

- More time writing the next piece of code
- Improved quality of overall code base

Code that's easier to read:

- Faster ramp-up for new project members
- Improved maintainability & troubleshooting

What is openCypher?

openCypher is an open source project to bring a new public implementation of the industry's most widely adopted graph query language: **Cypher**.

Focus on Your Domain

Graphs naturally describe your domain, and Cypher lets you focus on that domain instead of getting lost in the mechanics of data access. Both expressive and efficient, Cypher is intuitive and immediately familiar, without the steep learning curve of learning a new language.

Human Readable

Cypher is a human-readable query language that makes complex things possible. A combination of English prose and intuitive iconography, Cypher is accessible to developers and operations professionals alike.

Complete Open Source Access

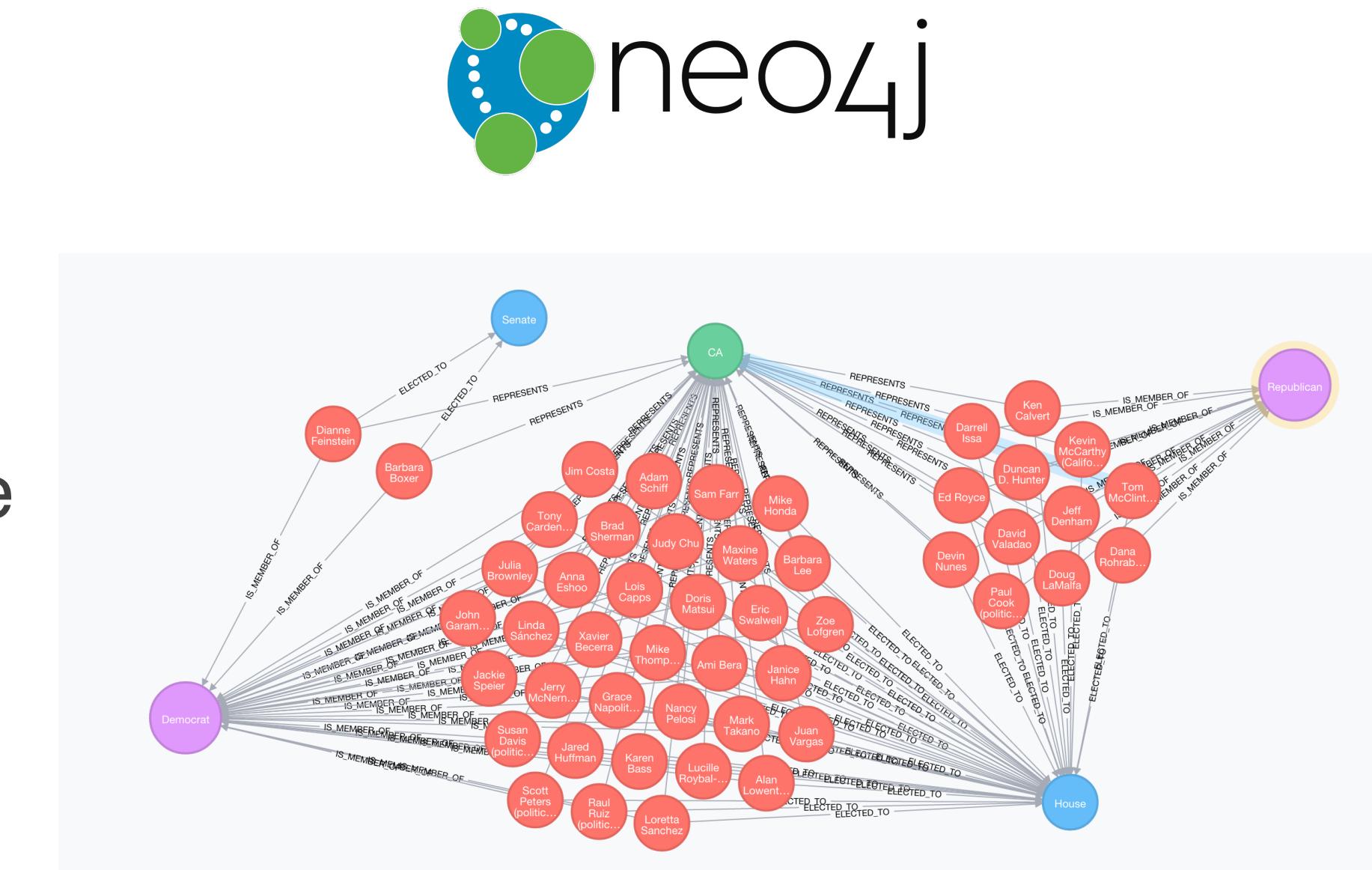
The openCypher project means you can use Cypher as your query language for graph processing capabilities within any product or application. Everyone from Oracle and Apache Spark to Tableau and Structr can (and do) use Cypher's capabilities – now you can too.

```
MATCH (cypher:QueryLanguage)-[:QUERIES]->(graphs)
MATCH (cypher)<-[USES]-(u:User) WHERE u.name IN ['Oracle', 'Apache Spark', 'Tableau', 'Structr']
MATCH (openCypher)-[:MAKES_AVAILABLE]->(cypher)
RETURN cypher.attributes
-----
['awesome',...]
```

<http://www.opencypher.org/>

Graph Database

- Property graph data model
 - Nodes and relationships
- Native graph processing
- (open)Cypher query language



Neo4j - Key Product Features



Native Graph Storage

Ensures data consistency and performance

Native Graph Processing

Millions of hops per second, in real time

“Whiteboard Friendly” Data Modeling

Model data as it naturally occurs

High Data Integrity

Fully ACID transactions

Powerful, Expressive Query Language

Requires 10x to 100x less code than SQL

Scalability and High Availability

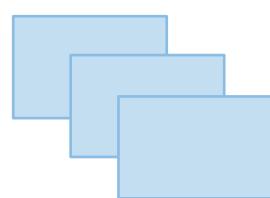
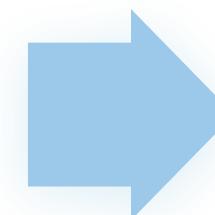
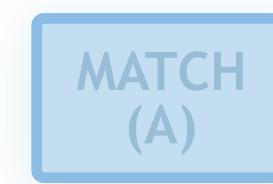
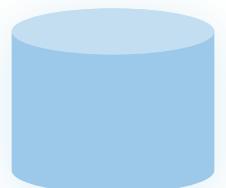
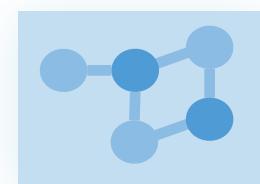
Vertical and horizontal scaling optimized for graphs

Built-in ETL

Seamless import from other databases

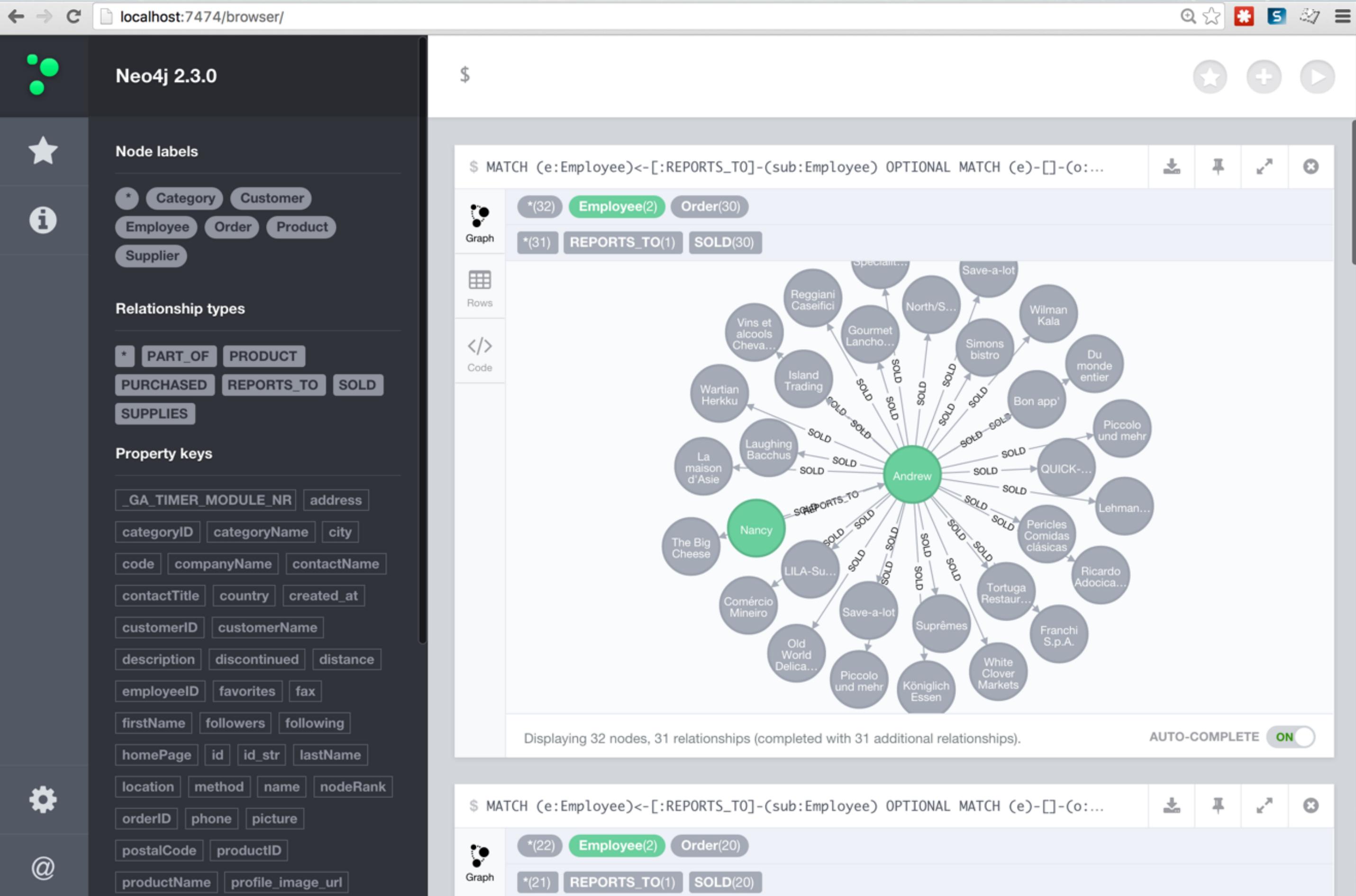
Integration

Drivers and APIs for popular languages



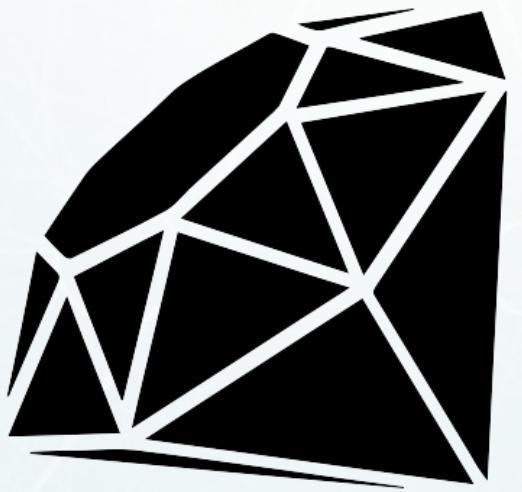
How do you use Neo4j?





Language Drivers

JS



nodejs **php**

Language Drivers

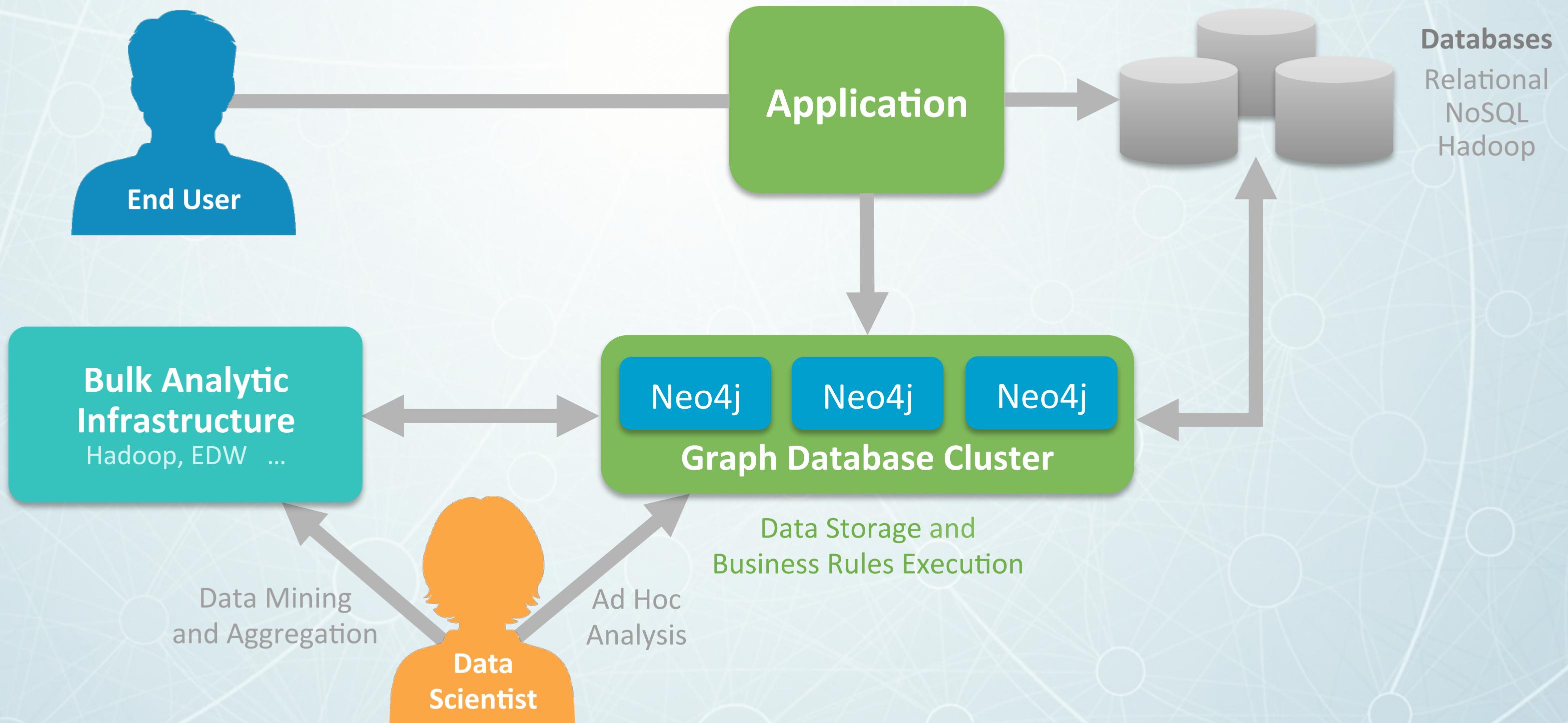
>>=



User Defined Procedures

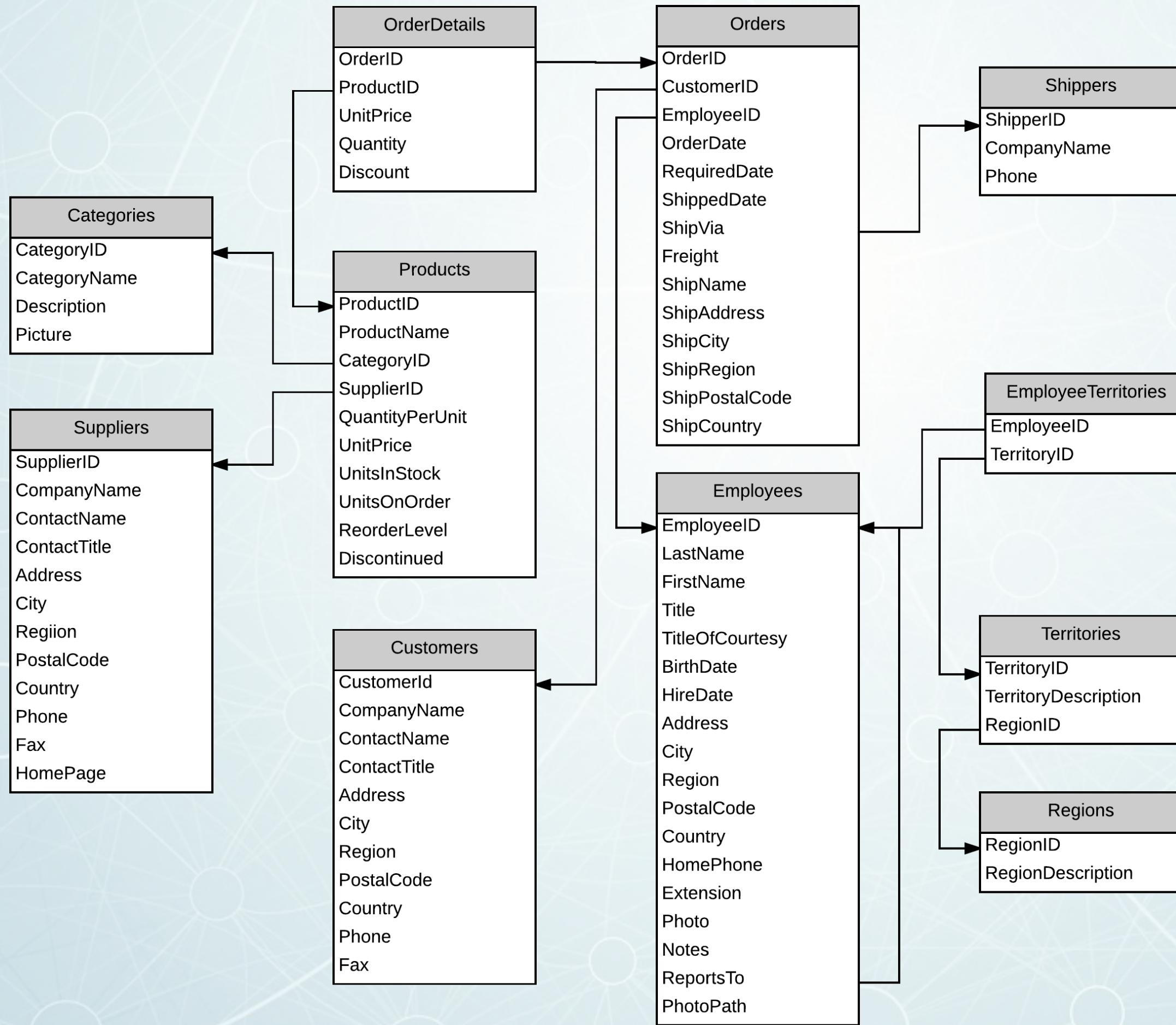


Architectural Options

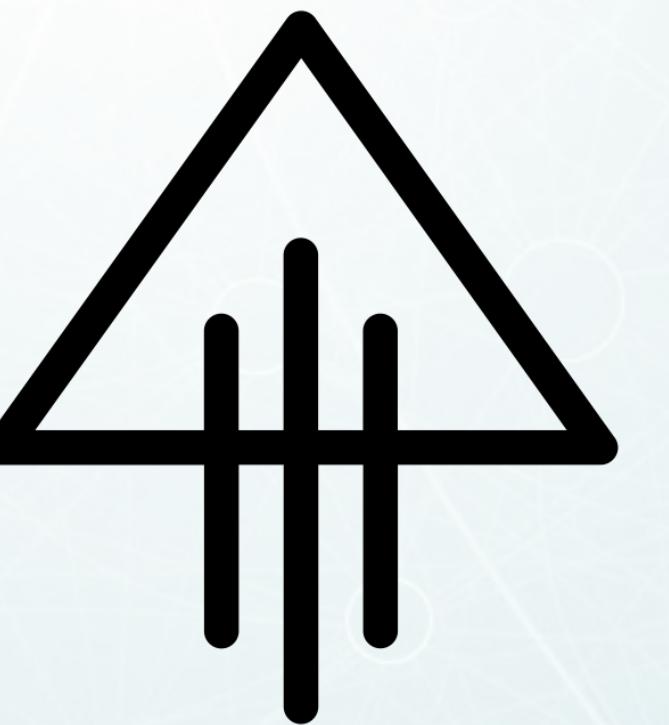


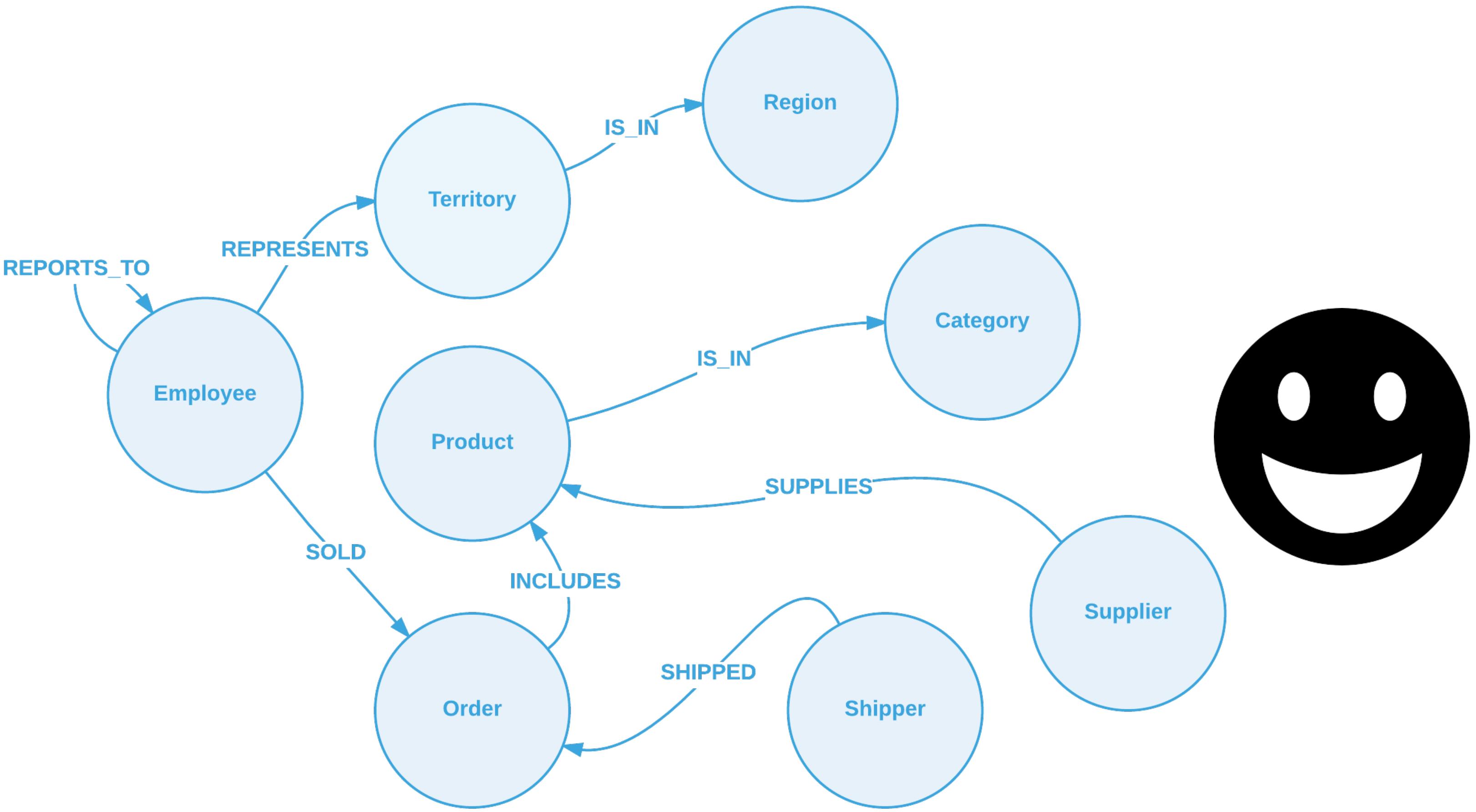
Day in the Life of a RDBMS Developer





Northwind





Property Graph Model



```
CREATE (:Employee{ firstName:"Steven" }) -[:REPORTS_TO]-> (:Employee{ firstName:"Andrew" })
```

LABEL

PROPERTY

LABEL

PROPERTY

Who do people report to?

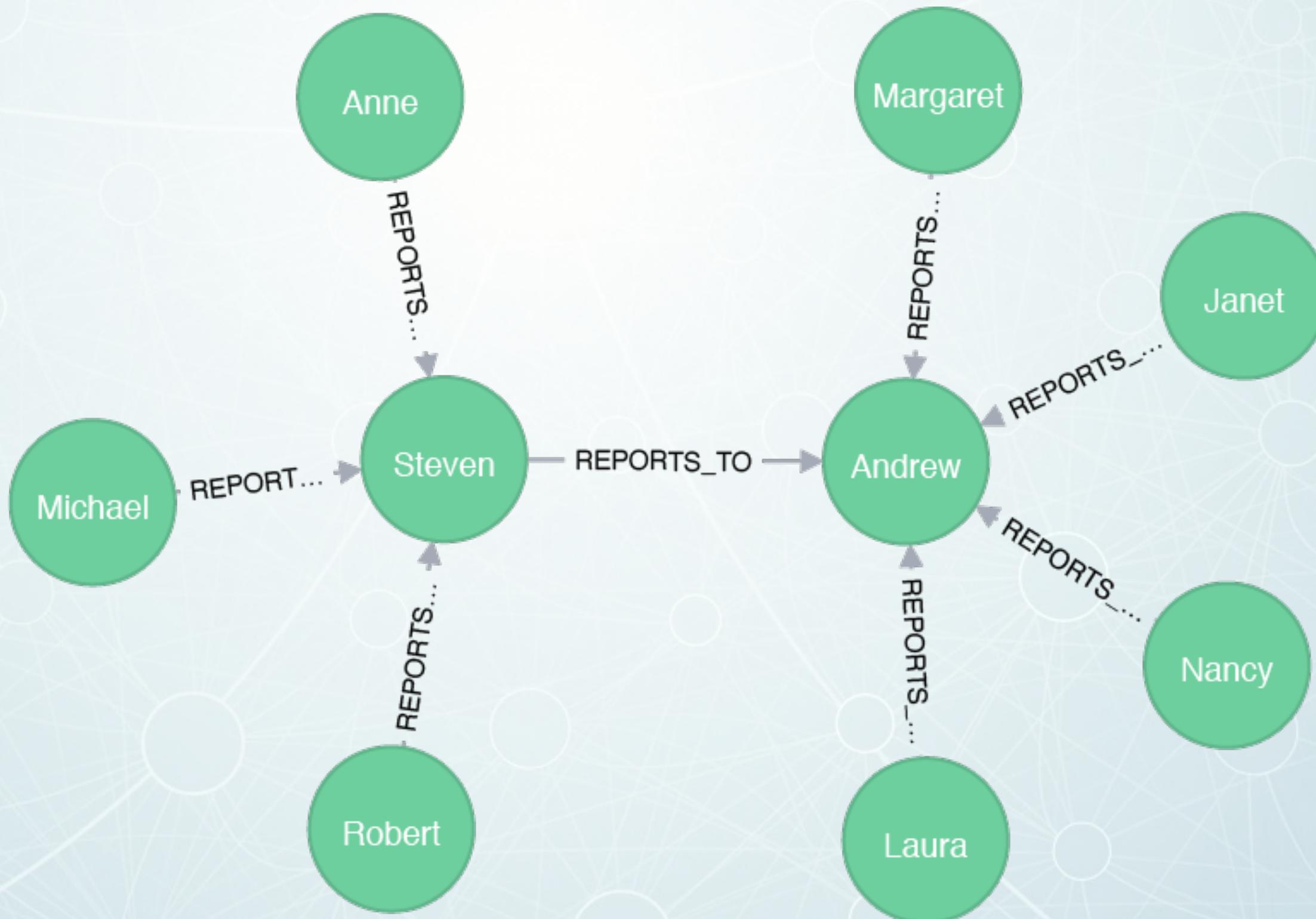
MATCH

```
(e:Employee)<-[ :REPORTS_TO ]-( sub:Employee )
```

RETURN

```
*
```

Who do people report to?



What is Robert's reporting chain?

MATCH

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

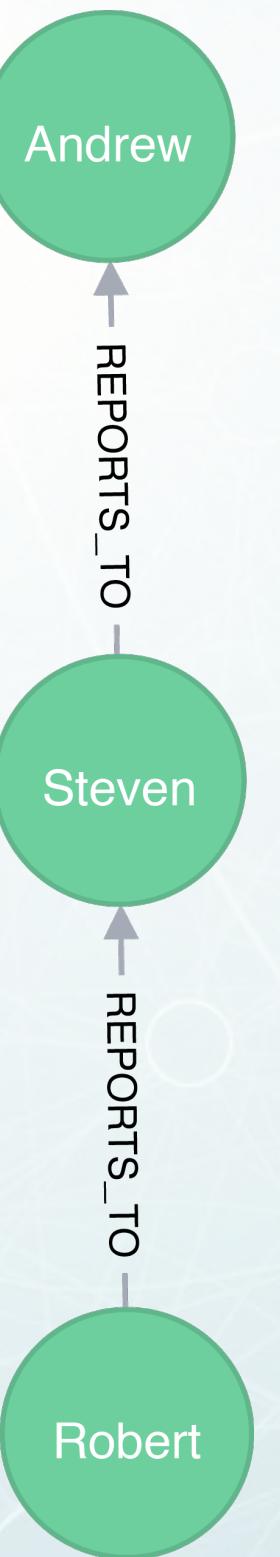
WHERE

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

RETURN

```
p
```

What is Robert's reporting chain?



Product Cross-Selling

MATCH

```
(choc:Product {productName: 'Chocolade' })  
  <- [ :INCLUDES ] - ( :Order ) <- [ :SOLD ] - ( employee ),  
  ( employee ) - [ :SOLD ] -> ( o2 ) - [ :INCLUDES ] -> ( other:Product )
```

RETURN

```
employee.firstName,  
other.productName,  
COUNT( DISTINCT o2 ) as count
```

ORDER BY

```
count DESC
```

```
LIMIT 5 ;
```

Product Cross-Selling

\$ MATCH (choc:Product {productName: 'Chocolade'}) <-[:INCLUDES] - (:Order) <... Download  Refresh  Share  Close 

employee.firstName	other.productName	count
Margaret	Gnocchi di nonna Alice	14
Janet	Gumbär Gummibärchen	12
Nancy	Flotemysost	12
Margaret	Pâté chinois	12
Nancy	Camembert Pierrot	11

Returned 5 rows in 319 ms.

Resources



Install Neo4j and Take it for a Spin

Experience powerful scalability, blazing speed and unparalleled flexibility – download and try Neo4j today.

For Business

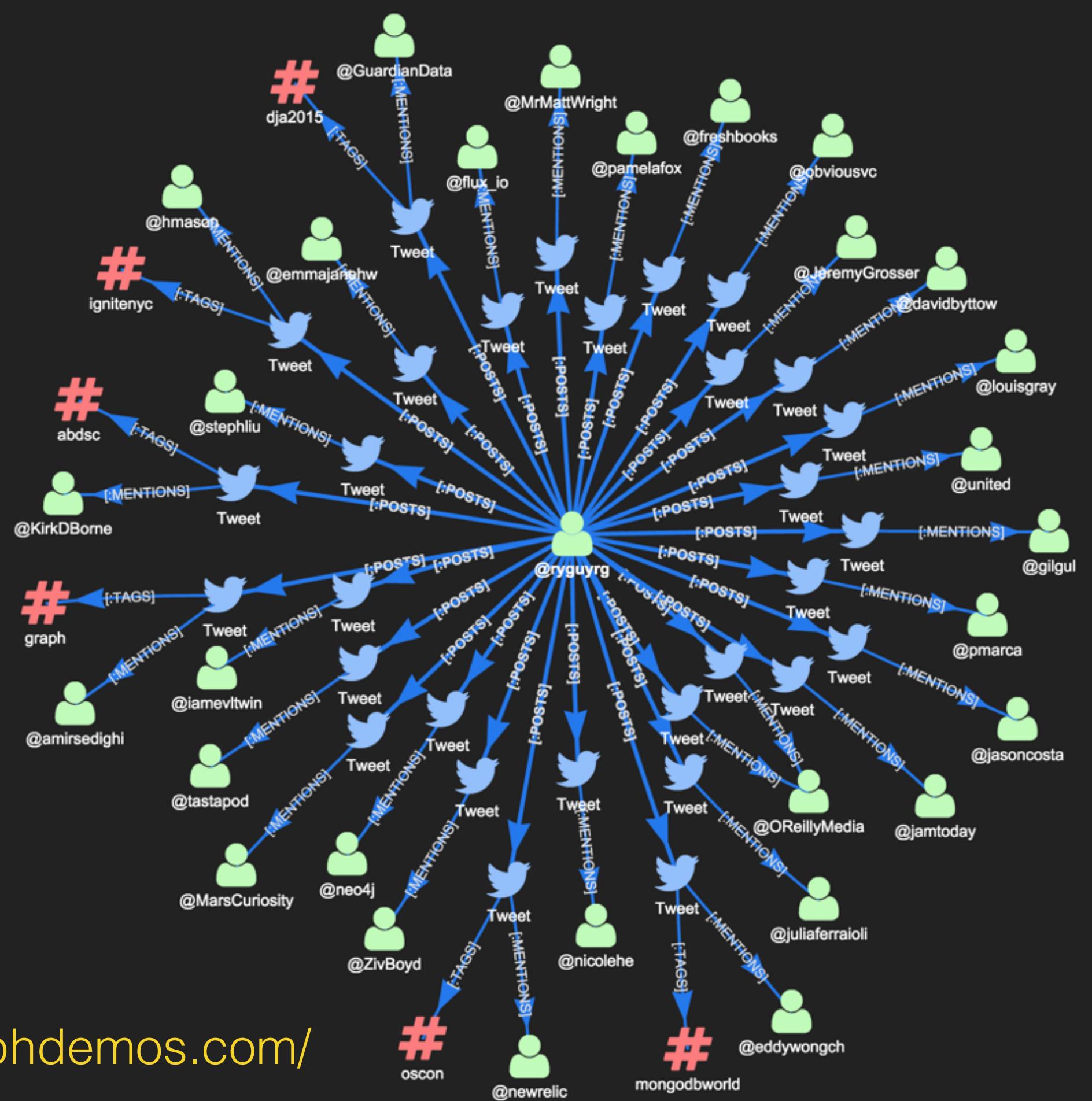
The Neo4j Enterprise Edition offers incredible power and flexibility, with enterprise-grade availability, management and scale-up & scale-out capabilities.

[Download Free Enterprise Trial](#)

For Individuals

Ideal for learning, and smaller do-it-yourself projects that do not require high levels of scaling. Excludes professional services and support.

[Download Community Edition](#)



neo4j.com/developer



Test-Drive Neo4j with Cypher

Social

Network Management

Fraud Detection

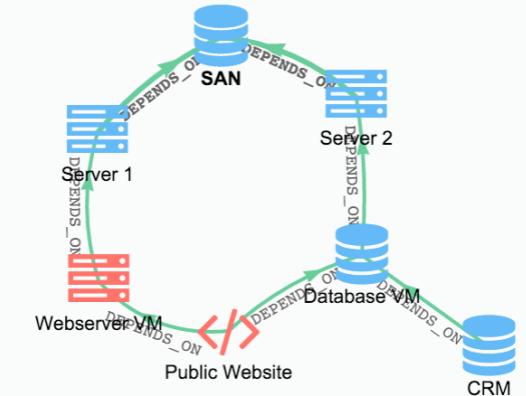
[Impact Analysis](#) | [Dependency Analysis](#) | [Statistics](#)

Impact Analysis

Find all services that depend on Server 1. These would be impacted by an outage of that server.

```
MATCH
  (n:Service)<-[ :DEPENDS_ON* ]-(dependent:Service)
WHERE
  n.name = "Server 1"
RETURN
  dependent
```

Only Webserver VM depends on Server 1. Because we're looking at variable length paths of DEPENDS_ON relationships, we're also able to determine that Public Website would be impacted by an outage of Server 1.



See Code In: [JAVA](#) [PYTHON](#) [RUBY](#) [PHP](#) [C#](#) [NODE.JS](#)

Downloading and Installing Java

- [Download Neo4j JDBC](#)
- Copy and paste code at left into [Social.java](#)
- Run [javac Social.java](#)
- Run [java -cp /path/to/neo4j-jdbc-2.3-SNAPSHOT-jar-with-dependencies.jar:. Social](#)

```
// javac Network.java
// java -cp /path/to/neo4j-jdbc-2.3-SNAPSHOT-jar-with-dependencies.jar:. Network

import java.sql.*;
import static java.util.Arrays.asList;
import java.util.List;

public class Network {

    public static void query(URLConnection con,
                           String query, String[] columns, Object
...params)
        throws SQLException {
        try (PreparedStatement pst = con.prepareStatement(query)) {
            for (int i=0;i<params.length;i++)
                pst.setObject(i + 1, params[i]);
            ResultSet rs = pst.executeQuery();
            int count = 0;
            while (rs.next()) {
                for (int i=0;i<columns.length;i++)
                    System.out.print(rs.getString(columns[i])+"\t"
);
        }
    }
}
```

How we engage the Neo4j community



GitHub

Graph Academy

Learn. Graph. Deploy.



Documentation

Google Groups

BOOKS

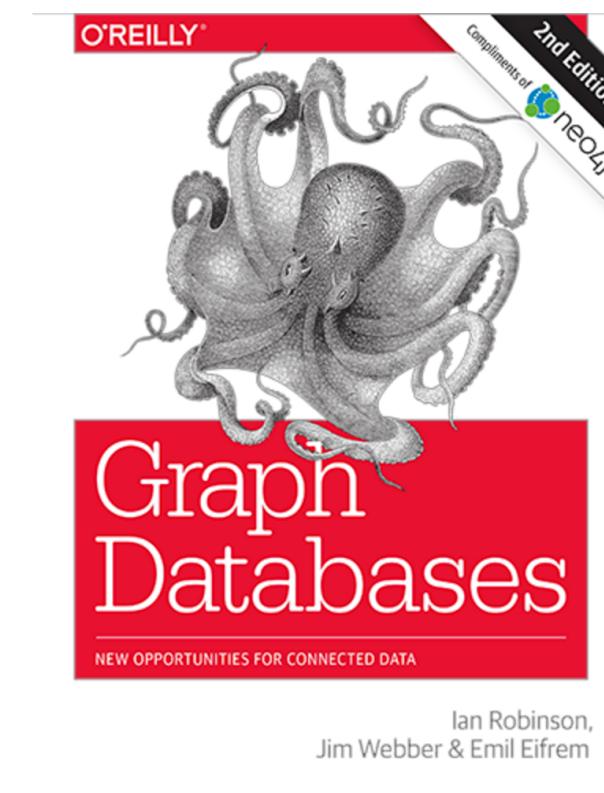
Built-in Guides

On-site
Training

GraphGist

Online Training





UPDATE! 2nd Edition

O'Reilly's *Graph Databases*

The Definitive Book on
Graph Databases.

Official Released Version of O'Reilly's Graph
Databases.

Copyright (c) 2015, Neo Technology, Inc. All rights reserved. The reproduction
or distribution of this copyrighted work is strictly prohibited.

Graph Databases
The Definitive Book on Graph Databases and Introduction to Neo4j

graphdatabases.com



What is a GraphGist?

With Neo4j GraphGists you can describe and model your domain in a simple text file ([AsciiDoc](#)) and render it as a rich, interactive, database-backed page in any browser. It is perfect to document a specific domain, use-case, question or graph problem.

Examples

GraphGists work like any AsciiDoc document, but they allow you to insert special comments to define how data from Neo4j can be displayed and interacted with.

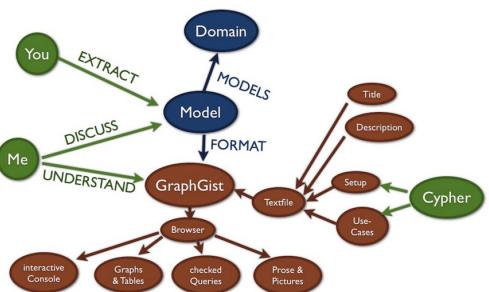
First define a block of [Cypher](#) code to setup the database:

```
//setup
//hide
[source,cypher]
-----
CREATE (:Database {name:'Neo4j'})-[:SUPPORTS]->(:Language {name:'Cypher'})
-----
```

You can then make queries to the data and output them as a table and/or a graph:

```
[source,cypher]
-----
MATCH (db:Database)-[:SUPPORTS]->(language:Language)
RETURN db.name, collect(language.name)
-----
//table
```

```
[source,cypher]
-----
MATCH (db:Database)-[rel:SUPPORTS]->(:Language)
RETURN rel
-----
//graph
```



GraphGists

Insight #4: Similar Repositories

Grouping by #contributors will exclude repositories with commits using a single account - *limitation*

```
Query 5
MATCH (a)-[r1:IS_ACTOR]->(MATCH)-<-[r2:IS_ACTOR]-(b) WHERE a.id > b.id
WITH a,b, collect(DISTINCT MATCH.id) AS connections, collect(DISTINCT type(r1)) AS rel1
WHERE length(connections) >= 1 //set minimum # of connections
RETURN a.id, b.id,length(connections) AS count ORDER BY length(connections) DESC
```

[Test run OK](#)

Show

10

entries

Search:

a.id	b.id	count
hakujin/homebrew	BrewTestBot/homebrew	3
x1022as/docker	oceawu/docker	3
cocos2d/cocos2d-x	andyque/cocos2d-x	3
dingpinglv/cocos2d-x	andyque/cocos2d-x	3
		2
		2
		2
		2
		2

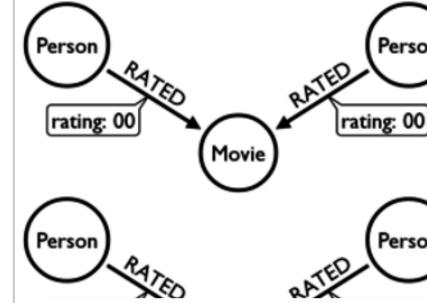
Aardvark: The Anatomy of a ...



Aardvark (2008 - 2011) was a social search service that connected users live with friends or fr...

Author: whatSocks

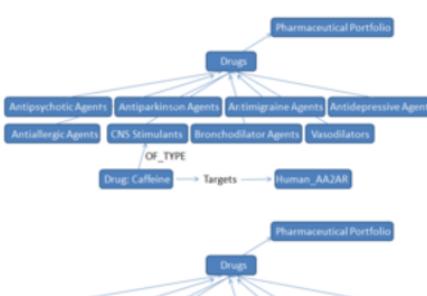
Movie Recommendations wi...



In this Graph Gist, I'm using k-NN with cosine similarity as the similarity metric to calcula...

Author: Nicole White

Pharmaceutical Drugs and t...



This GraphGist explores how to represent a pharmaceutical portfolio in a property graph. A phar...

Author: Josh Kunken

<http://neo4j.com/graphgists/>

THANK YOU!



will@neo4j.com
@lyonwj