Graph Databases and the #PanamaPapers

BED-con 2016

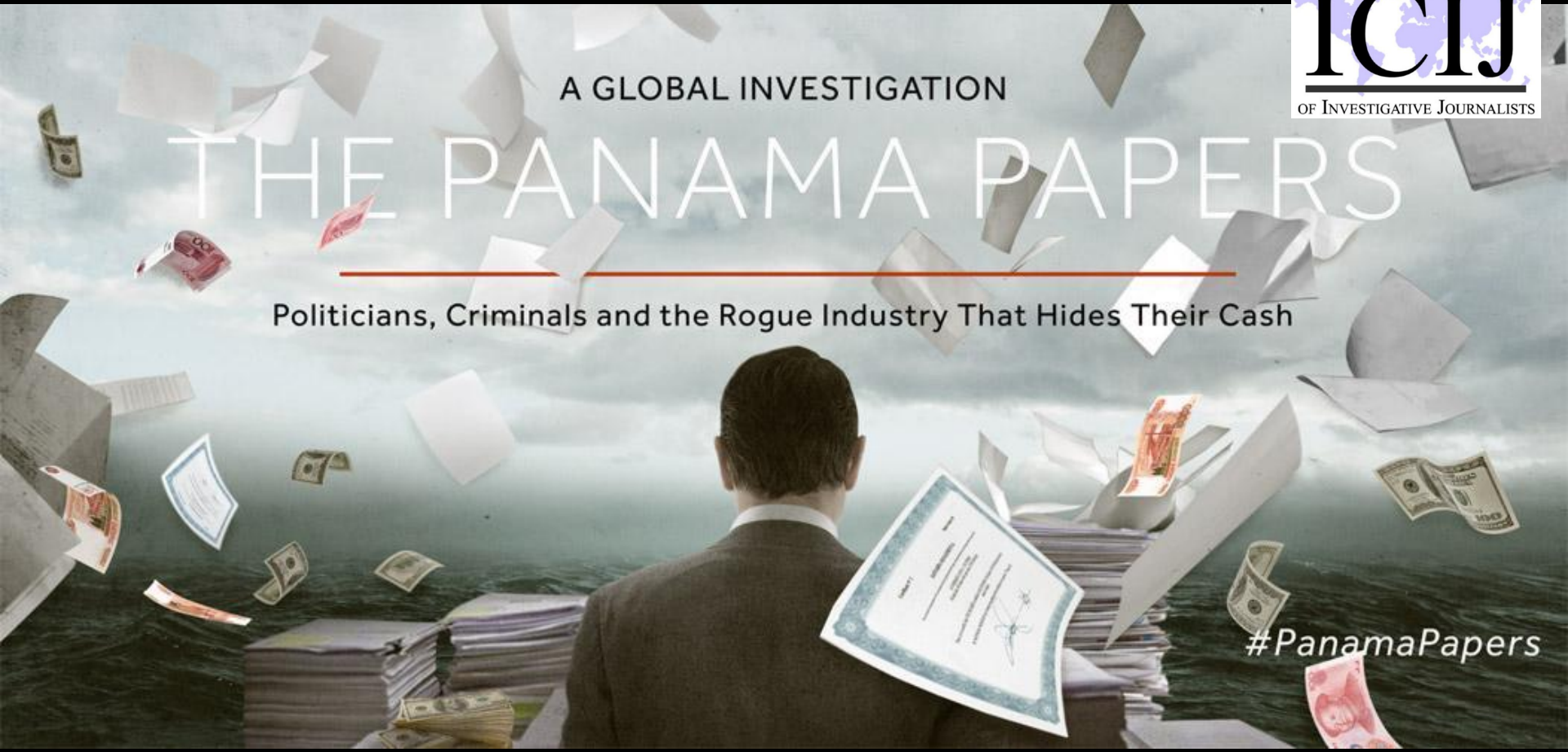
Stefan Armbruster

(Stefan)-[:WORKS_FOR]->(Neo4j)

stefan@neo4j.com | @darthvader42

github.com/sarmbruster | blog.armbruster-it.de

Stefan Armbruster - Field Engineer @Neo4j



INTERNATIONAL CONSORTIUM
ICIJ
OF INVESTIGATIVE JOURNALISTS

Source Material

Kudos to Michael Hunger @mesiiri
source material taken from

- the ICIJ presentation
- the Reddit AMA
- online publications (SZ, Guardian, TNW et.al.)
- the ICIJ website
  - https://panamapapers.icij.org/
  - The Power Players
  - Key Numbers & Figures

MOSSACK FONSECA

INTERNATIONAL CONSORTIUM
ICIJ
OF INVESTIGATIVE JOURNALISTS

[John Doe]
Hello. This is John Doe.
Interested in data?

[Süddeutsche Zeitung]
We're very interested.

[John Doe]
There are a couple of conditions. My life is in danger.
We will only chat over encrypted files.
No meeting, ever.
The choice of stories is obviously up to you.

[Süddeutsche Zeitung]
Why are you doing this?

[John Doe]
I want to make these crimes public.

Amount of data compared to previous leaks

Quelle: Süddeutsche Zeitung

© DW

International Consortium of Investigative Journalists

25 years The Center for Public Integrity

International Consortium ICIJ of Investigative Journalists

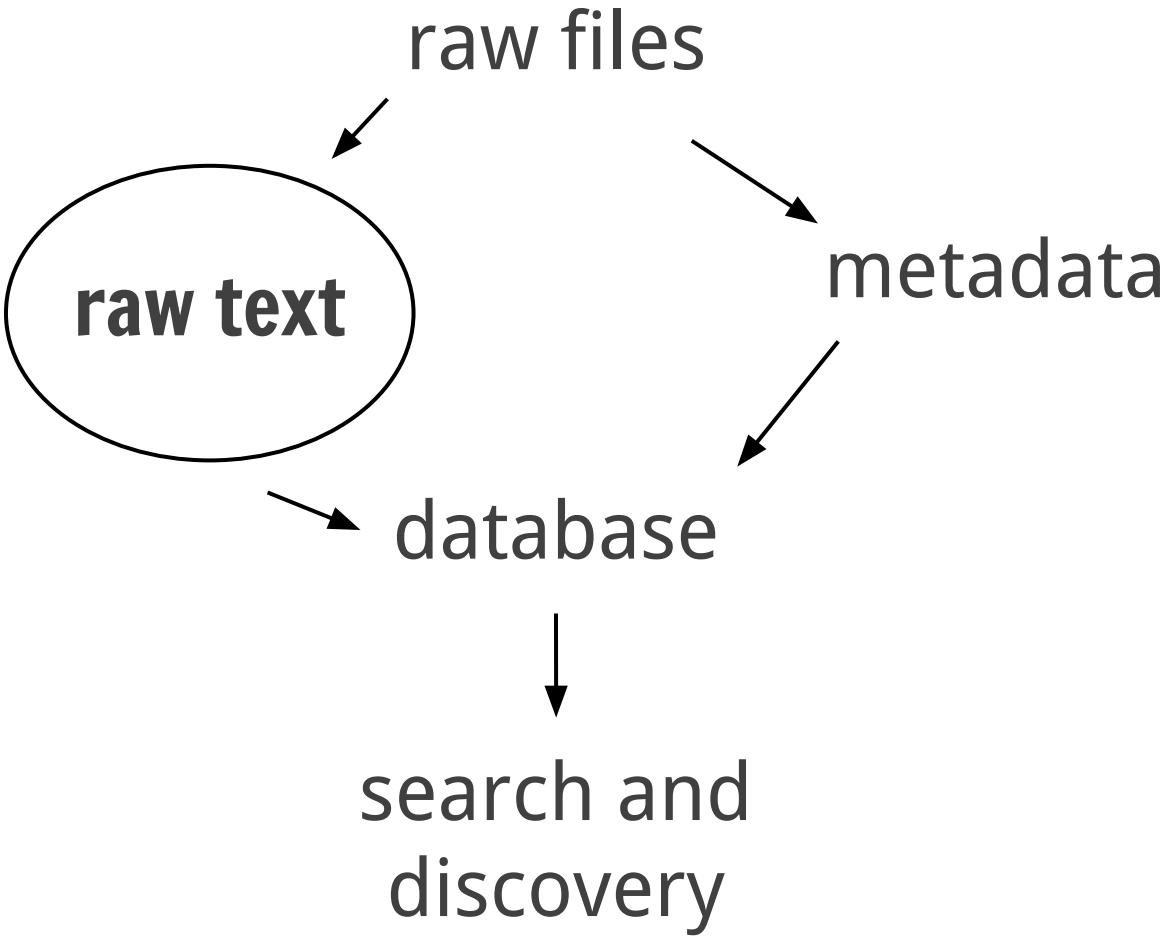
The World's Best Cross-Border Investigative Team

ICIJ The International Consortium of Investigative Journalists

+190 journalists in more than 65 countries

12 staff members (USA, Costa Rica, Venezuela, Germany, France, Spain)

50% of the team = Data & Research Unit



INTERNATIONAL CONSORTIUM
ICIJ
OF INVESTIGATIVE JOURNALISTS

INTERNATIONAL CONSORTIUM ICIJ OF INVESTIGATIVE JOURNALISTS

Investigators used **Nuix's** optical character recognition to make millions of scanned documents text-searchable. They used Nuix's named entity extraction and other analytical tools to identify and cross-reference the names of Mossack Fonseca clients through millions of

blacklight

Bookmarks (4) | Saved Searches | History | Log Out | mcaruana@icij.org | Users | Forum

International Consortium of Investigative Journalists

Everything | "joaquin loera"~2 | Search

Limit your results

- Data update
- Path
- Year created or sent
- Type

You searched for: "joaquin loera"~2

Start Over

1 - 4 of 4

Sort by Relevance | 10 per page

1. [redacted]

Bookmark

Text: [redacted] of Joaquin Guzmán Loera [redacted] Joaquin Guzmán Loera [redacted]

Subject: [redacted]

Date: [redacted]

Creator: [redacted]

400 users

Lucene syntax queries with proximity matching!

Solr

Stack

Unstructured data extraction

- Nuix professional OCR service
- ICIJ Extract (open source, Java: https://github.com/ICIJ/extract), leverages Apache Tika, Tesseract OCR and JBIG2-ImageIO.

Structured data extraction

- A bunch of Python

Database

- Apache Solr (open source, Java)
- Redis (open source, C)
- **Neo4j (open source, Java)**

App

- Blacklight (open source, Rails)
- Linkurious (closed source, JS)

Menu | Find | Edit data | Geo map

1 SELECTED NODES

SANDS INVEST & FINANCE LIMITED
#246682

pinned | Company

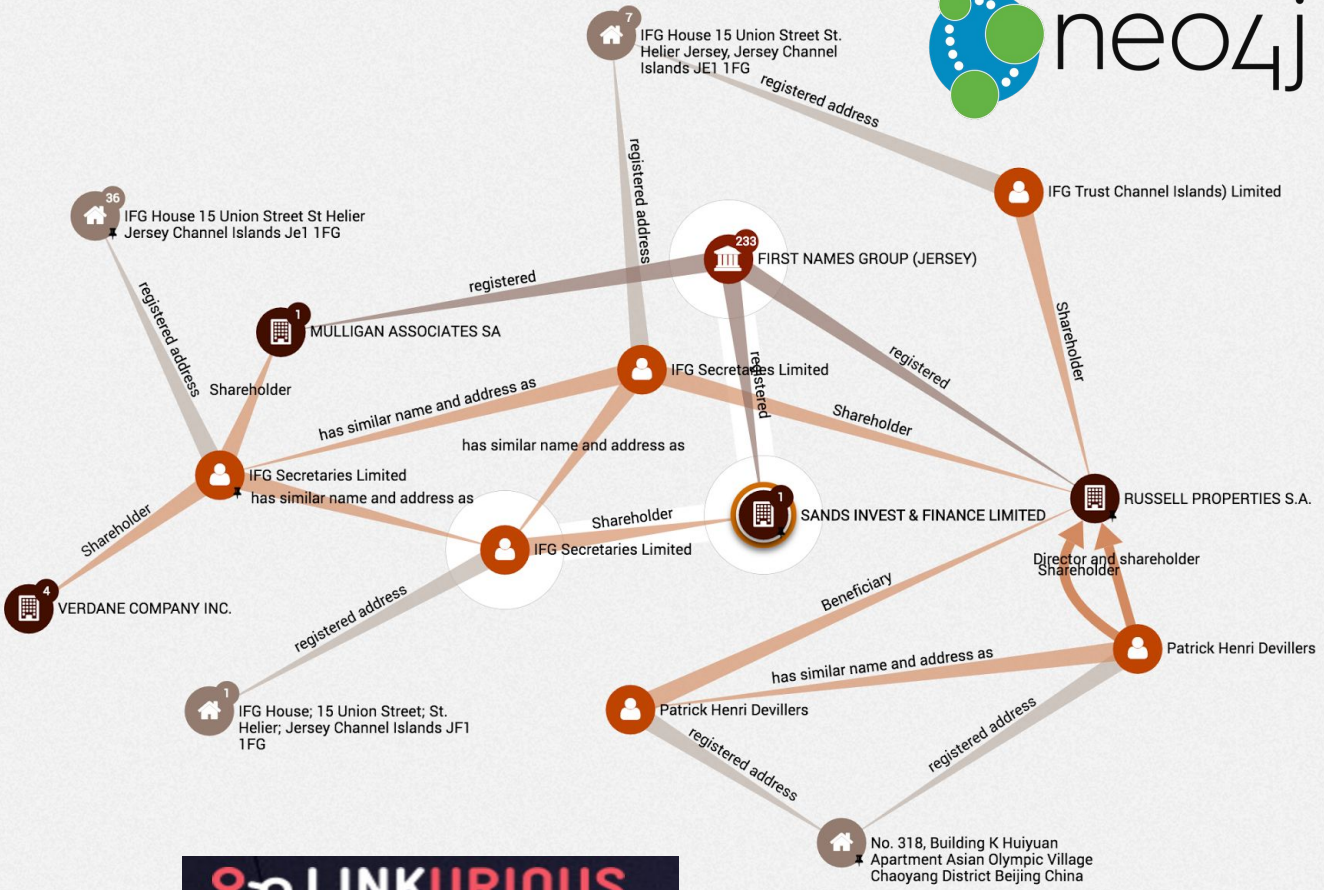
Select... | Deselect | Hide | Expand...

PROPERTIES

Find a property...

file_number	6015821
inactivationDate	18-FEB-2013
jurisdiction	BVI
name	SANDS INVEST & FINANCE LIMITED
registrationDate	20-APR-1998
status	DIS
struck_off_date	31-OCT-2013

2 EDGES / 3 IN DATABASE



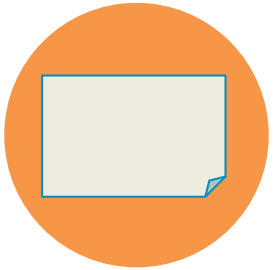
neo4j

LINKURIOUS
VISUALIZE GRAPH DATA EASILY

INTERNATIONAL CONSORTIUM ICIJ OF INVESTIGATIVE JOURNALISTS

Disconnected Documents

Context is King



name: “John”
last: „Miller“
role: „Negotiator“

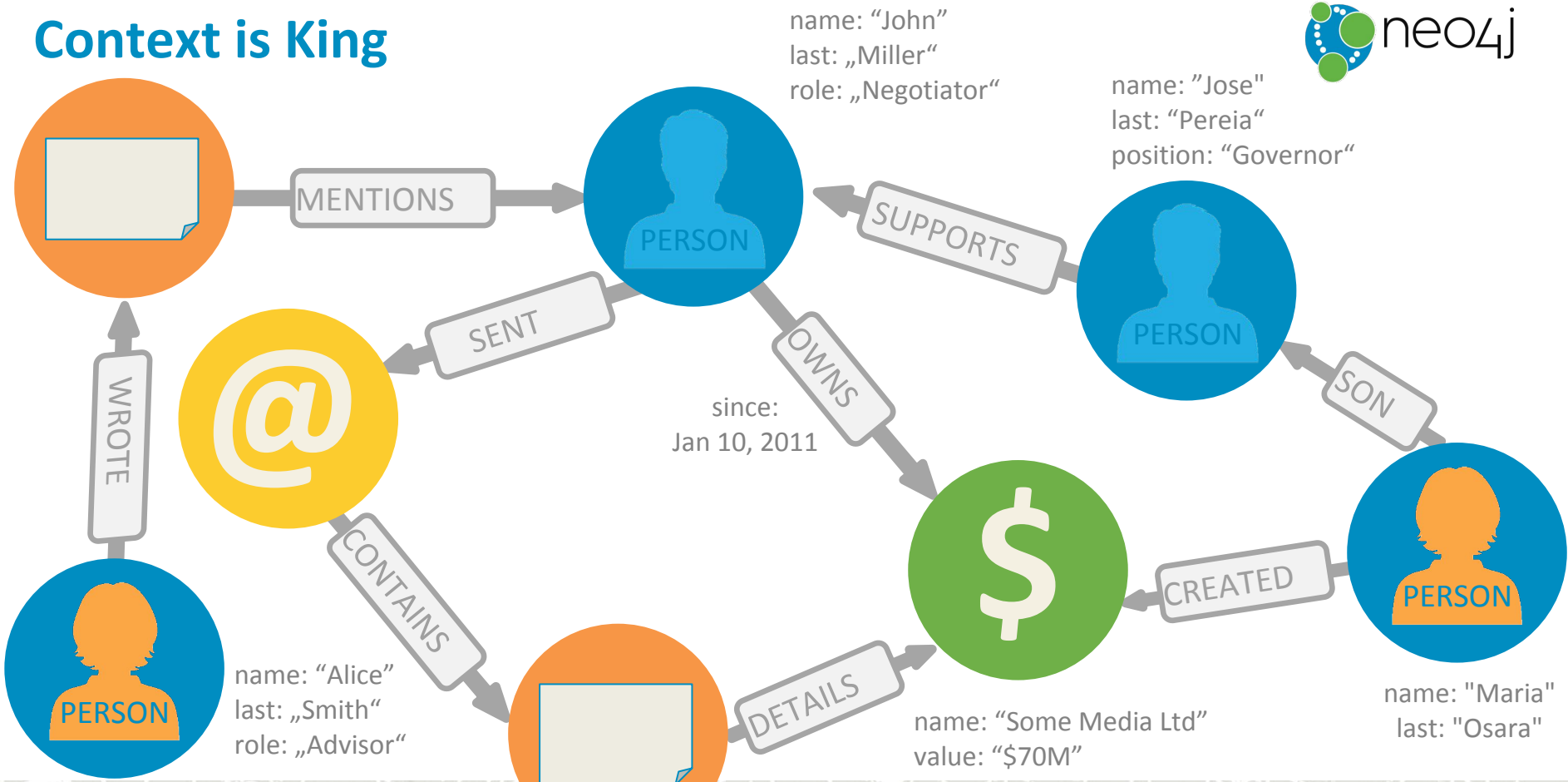
name: ”Jose"
last: “Pereia“
position: “Governor“

name: "Maria"
last: "Osara"

name: “Alice”
last: „Smith“
role: „Advisor“

name: “Some Media Ltd”
value: “$70M”

Context is King

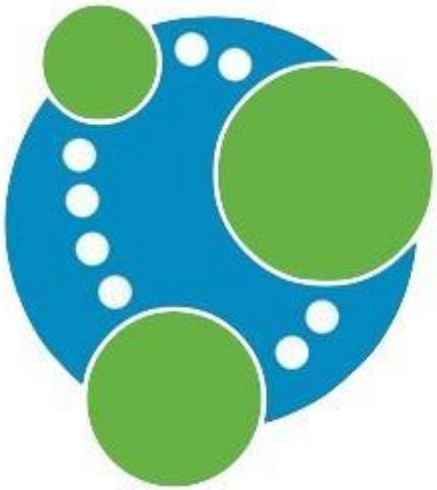


The world is a graph – everything is connected

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

We need to store and query our meta-data!

Real, inferred and integrated



neo4j

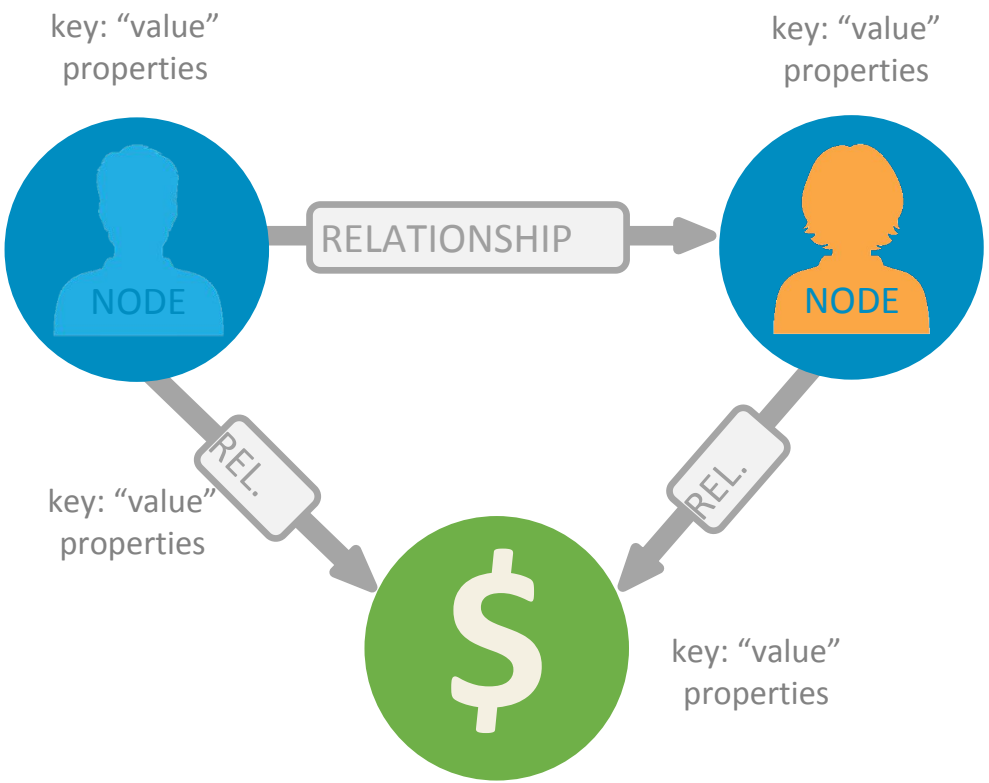
Property Graph Model

Nodes

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

Relationships

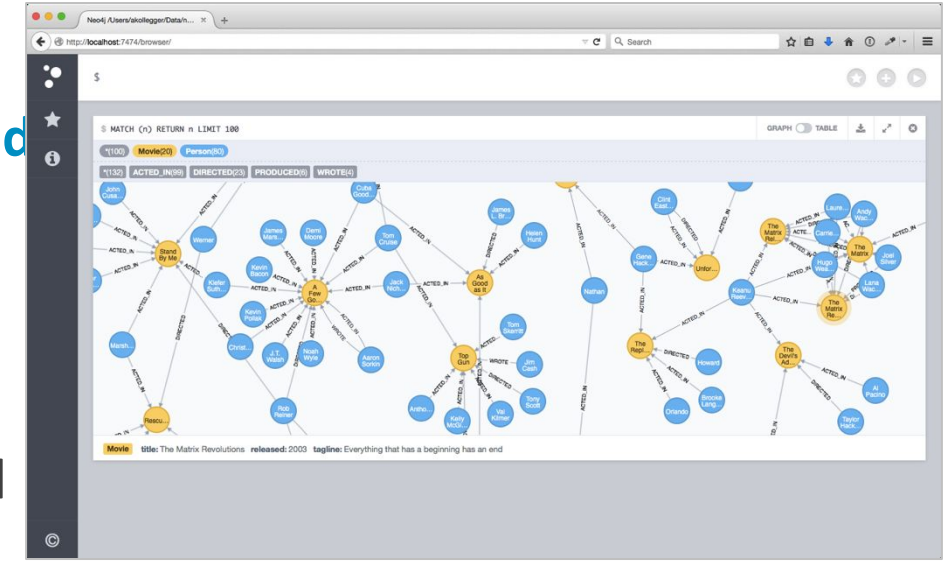
- Relate nodes by type and direction
- Can have name-value *properties*



Your friend Neo4j

An *open-source graph database*

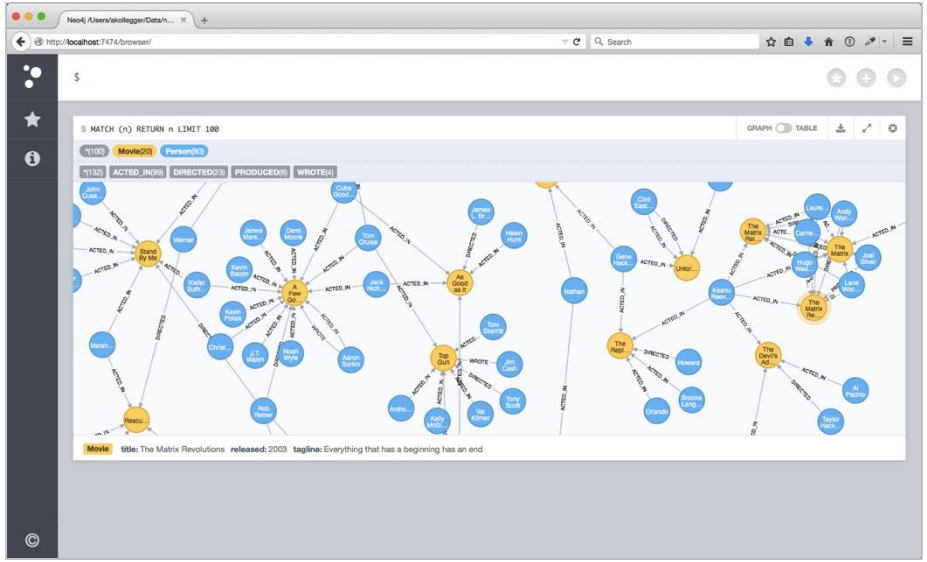
- **Manage and store** your **connected data** as a **graph**
- **Query relationships** easily and quickly
- **Evolve model and applications** to support new requirements and insights
- Built to solve **relational pains**



Your friend Neo4j

An *open-source graph database*

- **Built for Connected Data**
- **Easy to use**
- **Optional Schema**
- **Highly Scalable Performance**
- **Transactional ACID-Database**



Value from Data Relationships

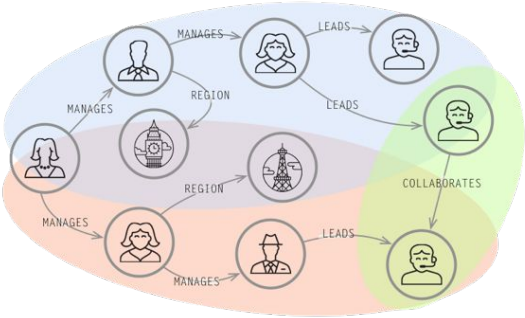
Common Use Cases

Internal Applications

Master Data Management

Network and
IT Operations

Fraud Detection

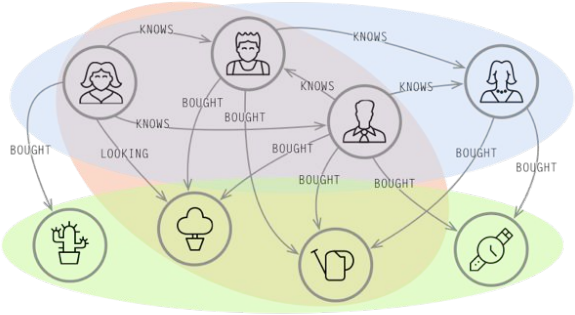


Customer-Facing Applications

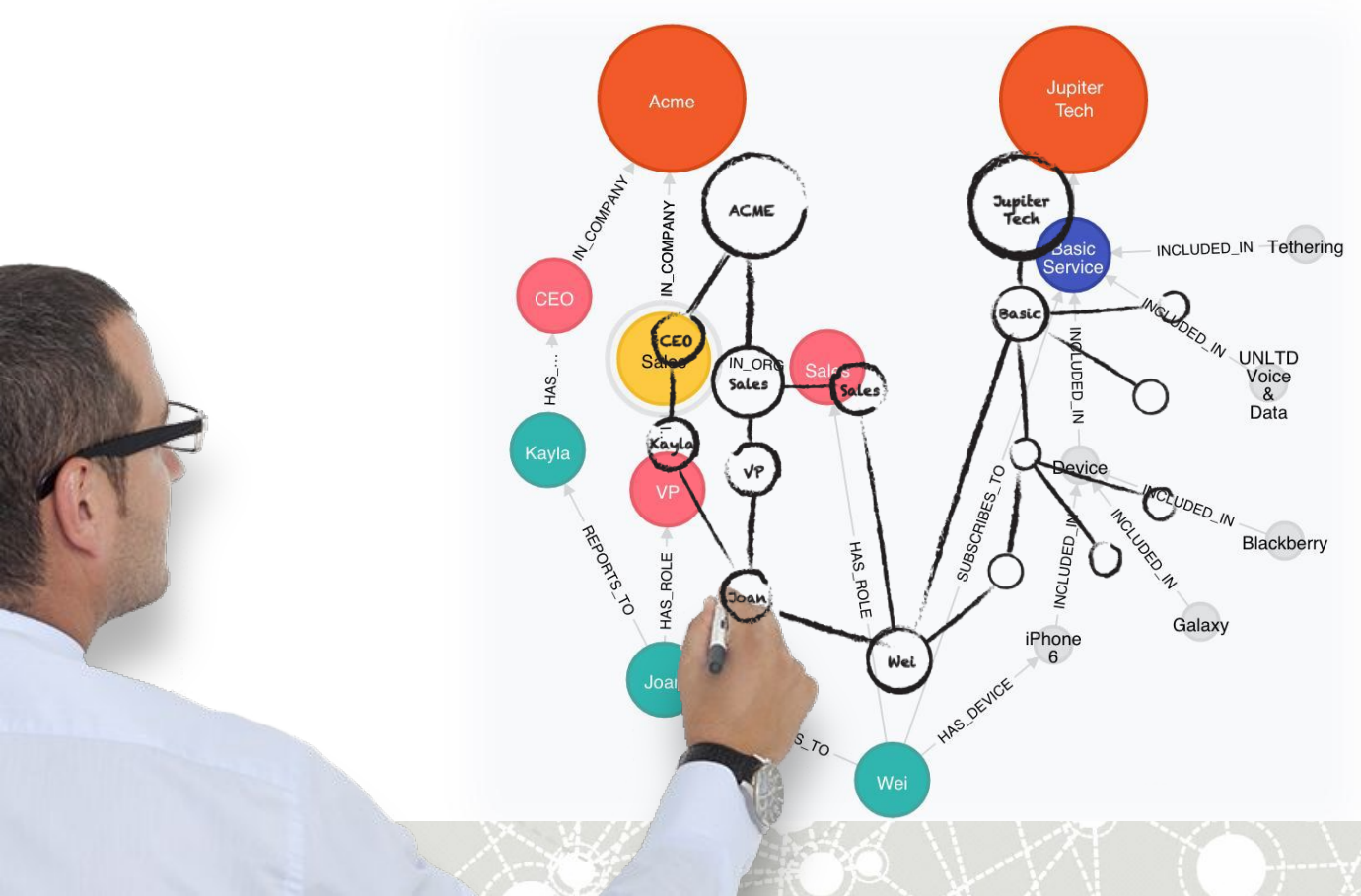
Real-Time Recommendations

Graph-Based Search

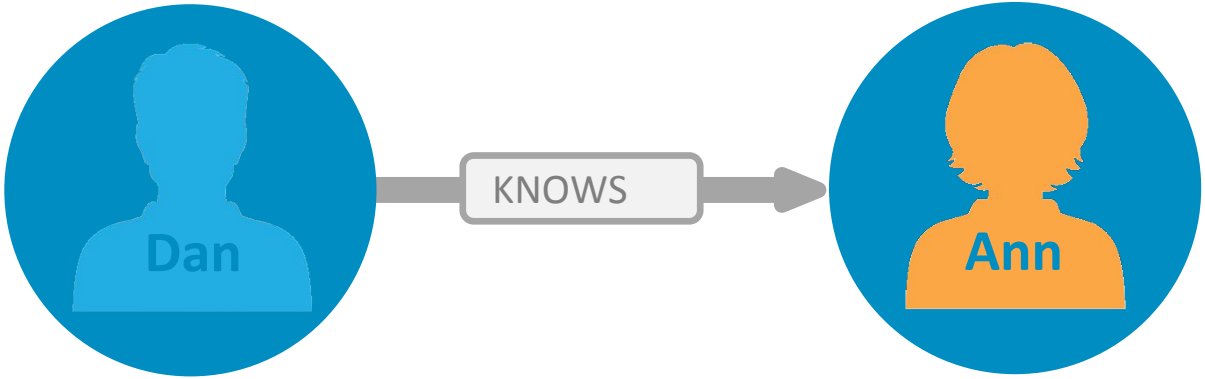
Identity and
Access Management



Whiteboard to Graph



Neo4j: All about Patterns

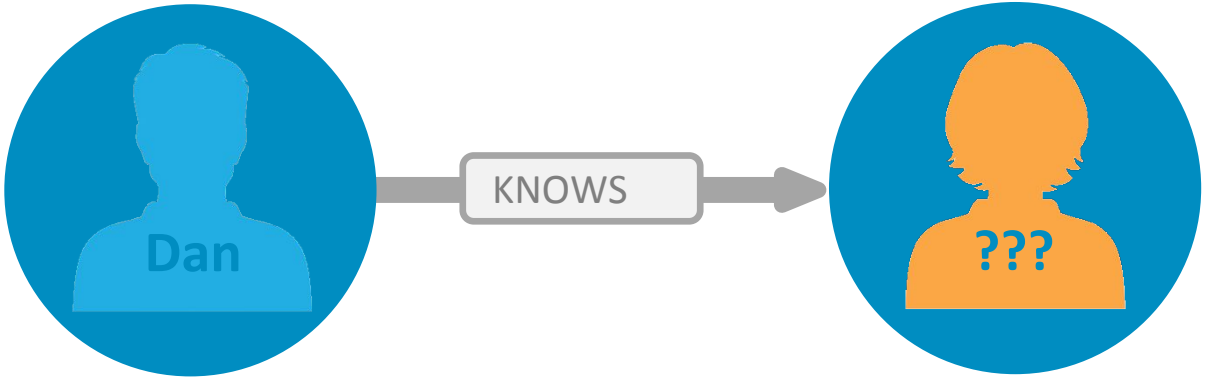


NODE NODE

(:Person { name:"Dan"}) -[:KNOWS]-> (:Person {name:"Ann"})

LABEL PROPERTY LABEL PROPERTY

Cypher: Find Patterns



NODE NODE

```
MATCH (:Person { name:"Dan"} ) -[:KNOWS]-> (who:Person) RETURN who
```

LABEL PROPERTY ALIAS LABEL ALIAS

Cypher: Clauses

CREATE *pattern*

MERGE *pattern*

SET

DELETE

REMOVE

Cypher: Clauses

neo4j

MATCH *pattern*

WHERE *predicate*

ORDER BY *expression*

SKIP … LIMIT …

RETURN *expression* AS *alias* …

Cypher: Clauses

WITH *expression* AS *alias*, ...

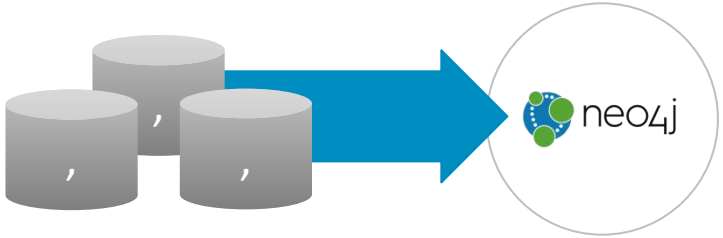
UNWIND *list* AS *item*

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

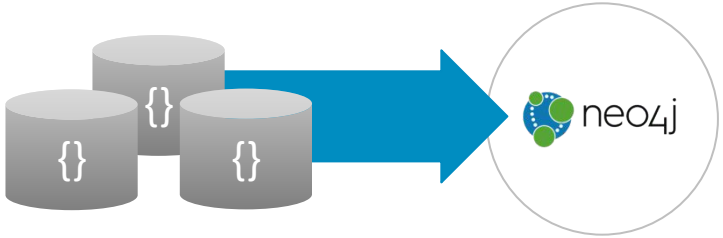



```
LOAD CSV WITH HEADERS FROM "url" AS row
MERGE (:Person {name:row.name,
                age:toInt(row.age)});
```

Getting Data into Neo4j

Load JSON with Cypher

- Load JSON via procedure
- Deconstruct the document
- Into a non-duplicated graph model

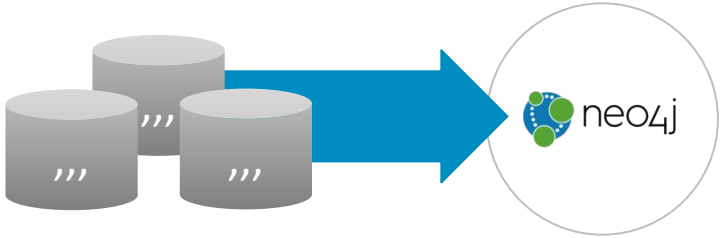



```
CALL apoc.load.json("url") yield value as doc
UNWIND doc.items as item
MERGE (:Contract {title:item.title,
                  amount:toFloat(item.amount)});
```

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
 --nodes:Company companies.csv
 --relationship:STAKEHOLDER stakeholders.csv
```


International Consortium
ICIJ
of Investigative Journalists

The Steps Involved in the Document Analysis

1. **Acquire** documents

2. **Classify** documents
   - Scan / OCR
   - Extract document metadata

3. Whiteboard **domain** and **questions**, determine
   - **entities** and their **relationships**
   - potential entity and relationship **properties**
   - **sources** for those entities and their properties

The Steps Involved in the Document Analysis

4. Develop analyzers, rules, parsers and named entity recognition
5. Parse and store metadata, document and entity relationships
   - Parse by author, named entities, dates, sources and classifications
6. Infer entity relationships
7. Compute similarities, transitive cover and triangles
8. Analyze data using graph queries and visualizations

We need a Data Model

Either based on our use cases & questions

On the entities present in our meta-data and data.

Meta Data Entities

- Document, Email, Contract, DB-Record
- Meta: Author, Date, Source, Keywords
- Conversation: Sender, Receiver, Topic
- Money Flows

Actual Entities

- Person
- Representative (Officer)
- Address
- Client
- Company
- Account

Data Model – Relationships

neo4j

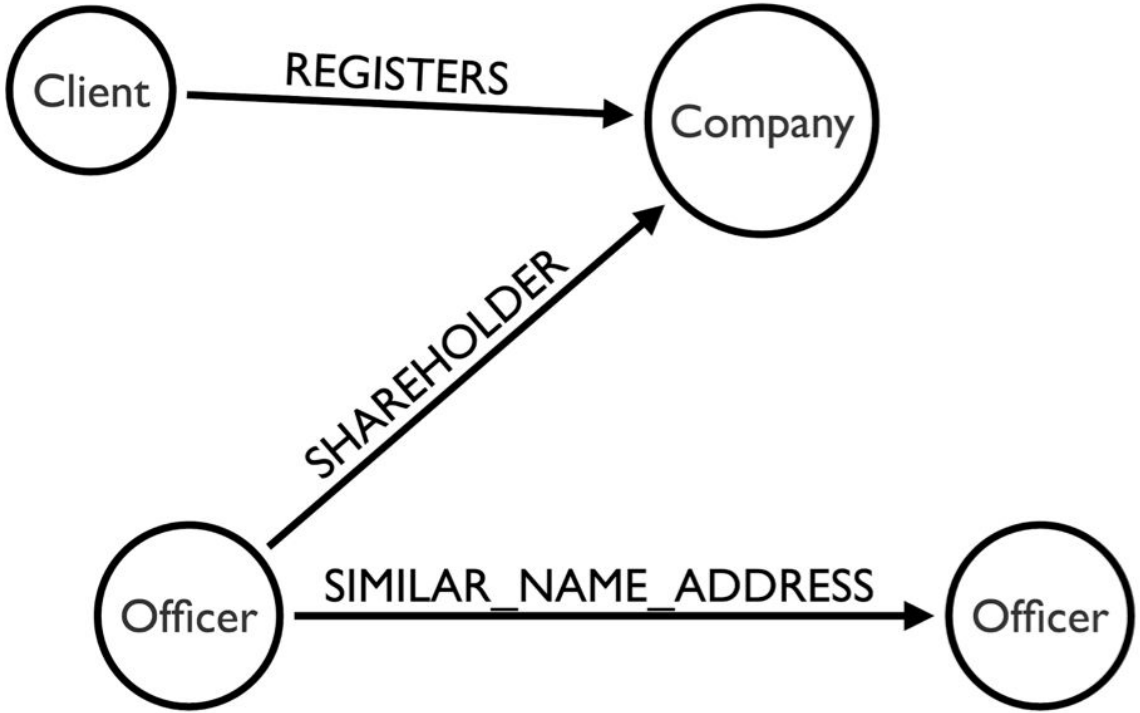
Meta-Data

- sent, received, cc‘ed
- mentioned, topic-of
- created, signed
- attached
- roles
- family relationships

Activities

- open account
- manage
- has shares
- registered address
- money flow

The ICIJ Data Model

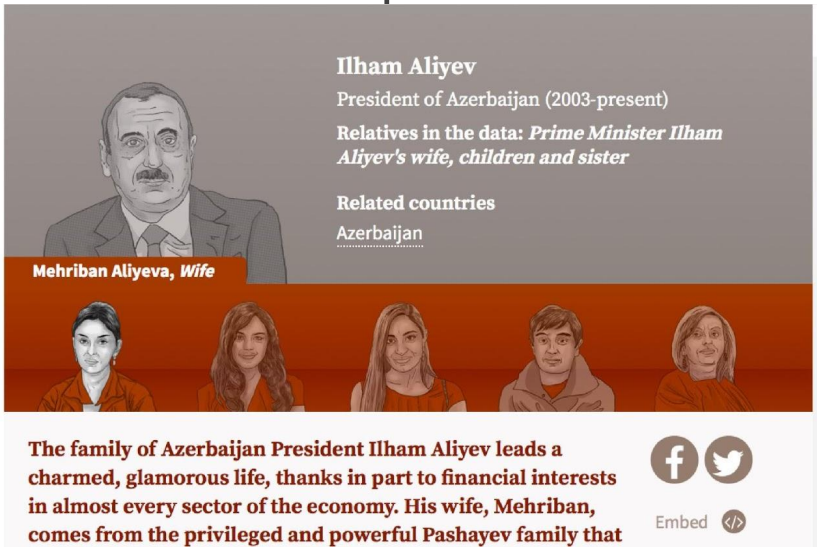


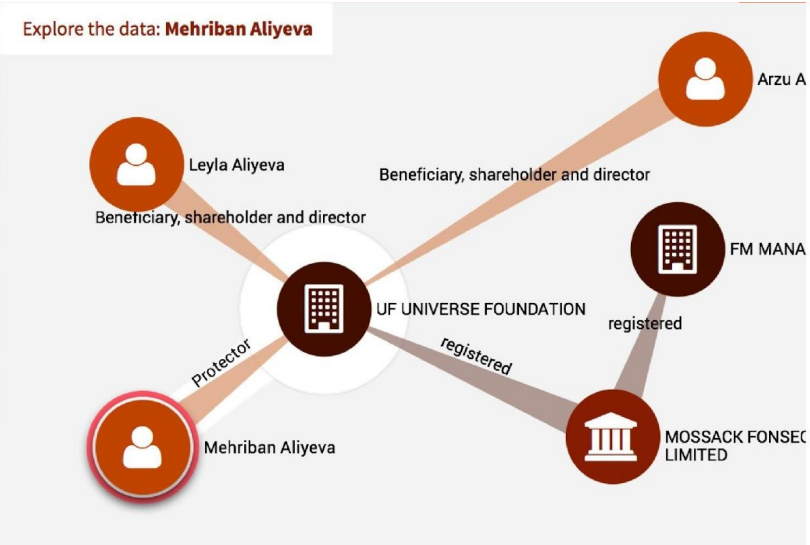
The ICIJ Data Model

- Simplistic Datamodel with 4 Entities and 5 Relationships
- We only know the published model
- Missing
  - Documents, Metadata
  - Family Relationships
  - Connections to Public Record Databases
- Contains Duplicates
- Relationship information stored on entities
- Could use richer labeling

Example Dataset - Azerbaijan's President Ilham Aliyev

- was already previously investigated
- whole family involved
- different shell companies & involvements





http://neo4j.com/graphgist/ec65c2fa-9d83-4894-bc1e-98c475c7b57a

Demo Time – Follow Along

:play http://guides.neo4j.com/graphgist/panama_papers.html

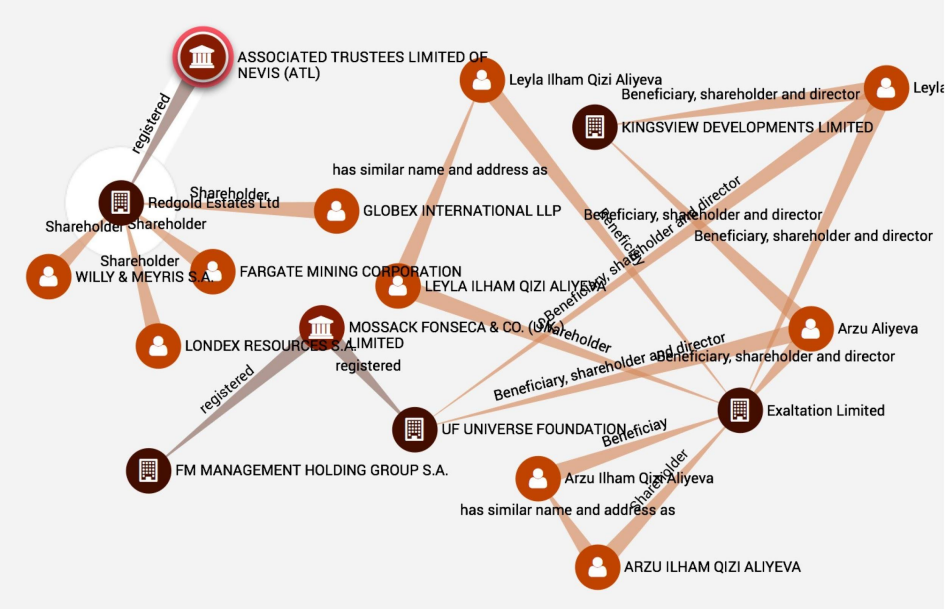
Based On: http://neo4j.com/blog/analyzing-panama-papers-neo4j/

Visual Graph Search

For Non-Developers

Linkurious.js

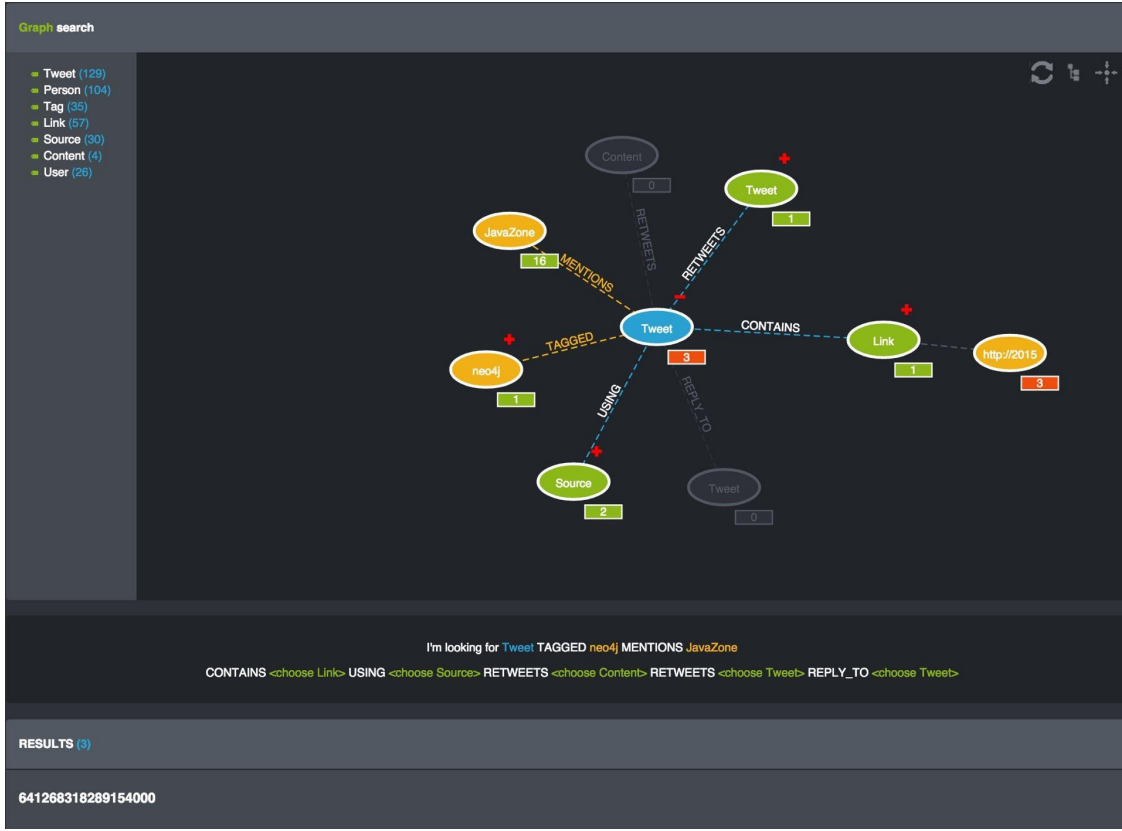
- JS library based on **sigma.js**
- Integrates with Neo4j using Cypher



https://github.com/Linkurious/linkurious.js/

Popoto.js

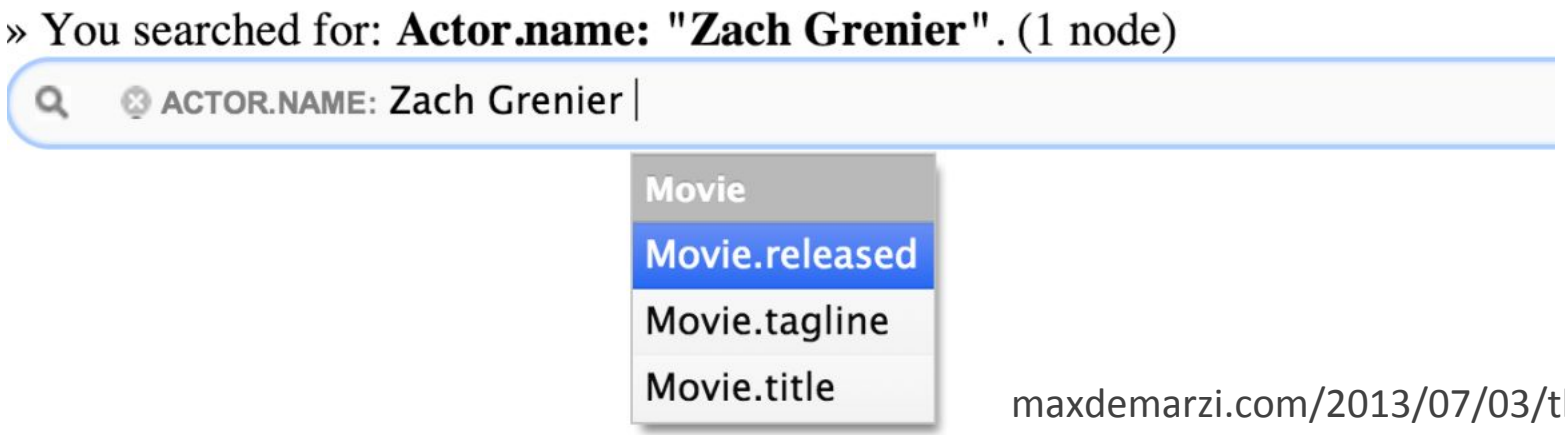
- JS library based on **d3.js**
- Uses Graph Metadata to offer visual search
- Categories to filter Instances
- Component based extensions
- Zero Config with Web Extension



http://www.popotojs.com/

Visual Search Bar

- Based on visualsearch.js
- Uses graph metadata for parametrization
- Limit suggestions by selected items



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

Neo Graph Search Home Profile Graph Search Likes Friends Visualization Max De Marzi

Graph Search

friends who like Neo4j Search Try: friends who like wine , people who like wine and cheese , people who like cycling and live in Florida

Cypher Query:

```
START me = node({me}), thing = node:things({thing})
MATCH me -[:friends]-> people, people -[:likes]-> thing
RETURN DISTINCT people , people.uid, people.name, people.image_url
LIMIT 100
Parameters: {"me"=>1, "thing"=>"name: Neo4j"}
```

Want your own Graph Search? Contact me and learn more about Neo4j and NeoTechnology


Peter Neubauer


Andres Taylor

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

Discrete Data
Minimally connected data

Connected Data
Focused on Data Relationships

Other NoSQL | **Relational DBMS** | **Graph DBMS**

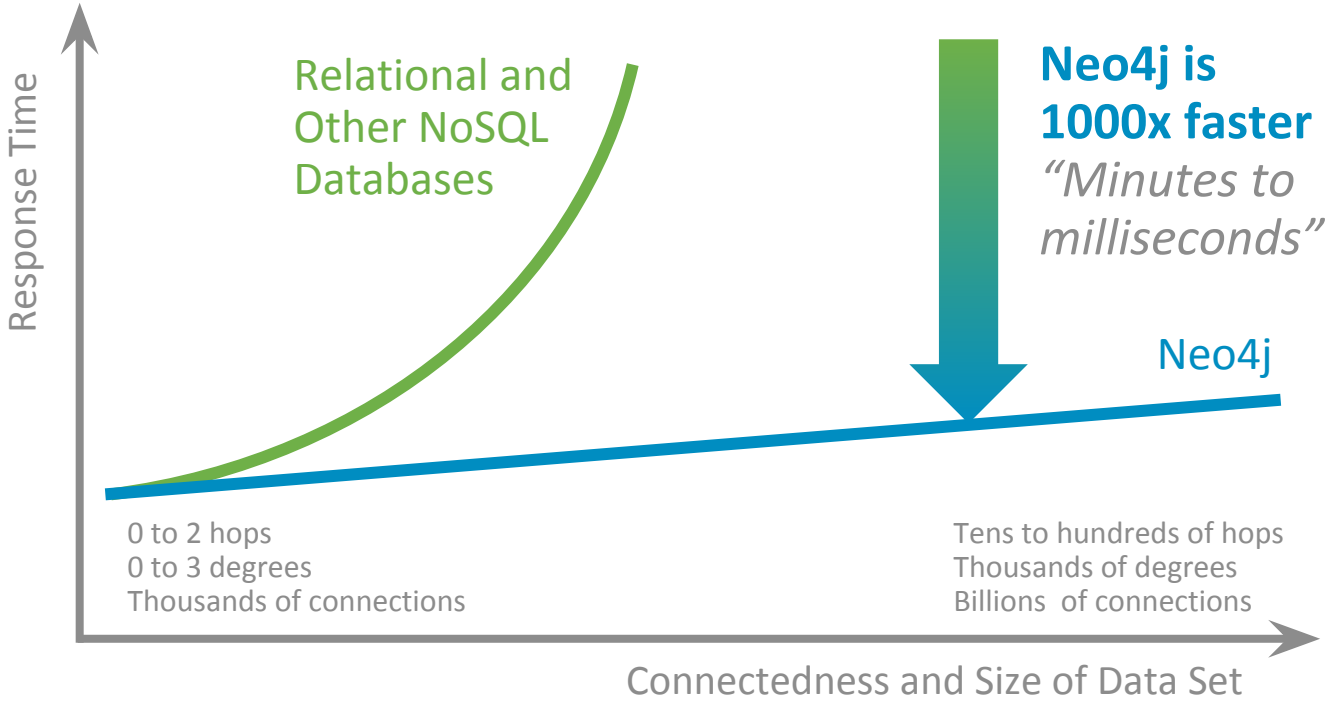
Graph Databases are designed for data relationships

Development Benefits
Easy model maintenance
Easy query

Deployment Benefits
Ultra high performance
Minimal resource usage

Real-Time Query Performance

Graph Versus Relational and Other NoSQL Databases



ICIJ editor Mar Cabra presenting at GraphConnect

Mar Cabra is the Editor of the Data and Research Unit at the International Consortium of Investigative Journalists (ICIJ), the organization responsible for breaking the Panama Papers story.

Mar has over 11 years of experience working in data journalism, including the BBC, CNN and the Miami Herald.

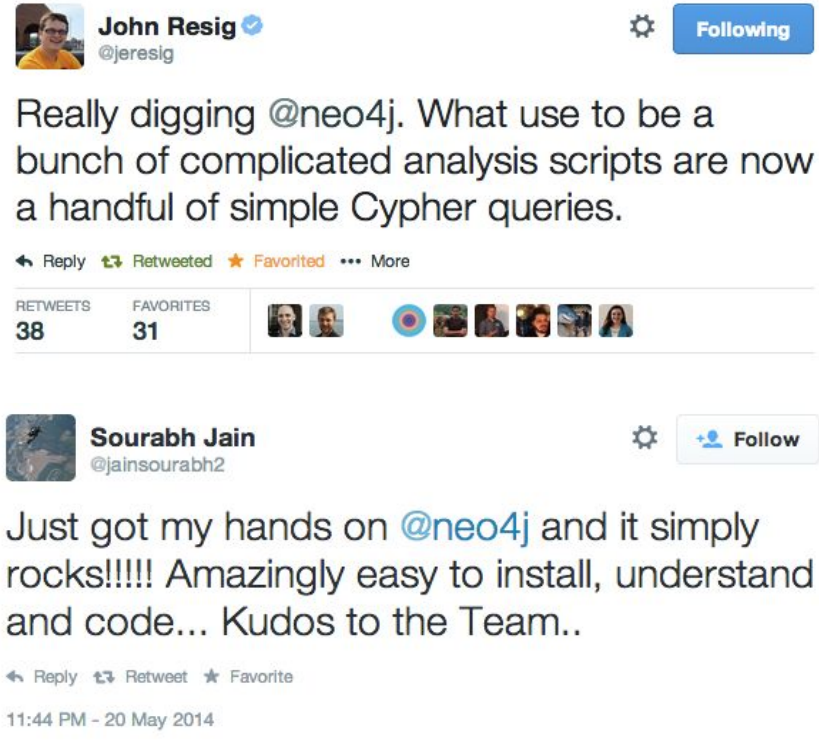
At GraphConnect, Mar will be presenting on "**How the ICIJ Used Neo4j to Unravel the Panama Papers.**"

neo4j.com/blog/top-10-graphconnect-europe-speakers/

More Insight

- Neo4j Blog
  - http://neo4j.com/blog/panama-papers/
  - http://neo4j.com/blog/analyzing-panama-papers-neo4j/
- ICIJ
  - https://panamapapers.icij.org/
  - https://panamapapers.icij.org/the_power_players/
  - https://panamapapers.icij.org/graphs/
- SZ
  - http://panamapapers.sueddeutsche.de/en/
- Guardian
  - http://www.theguardian.com/news/series/panama-papers

Users Love Neo4j – Will you too?



Get started with Neo4j today – Discover Value in Your Relationships

neo4j.com/developer

Thanks! Questions?

Slide Bucket

Why should I care?

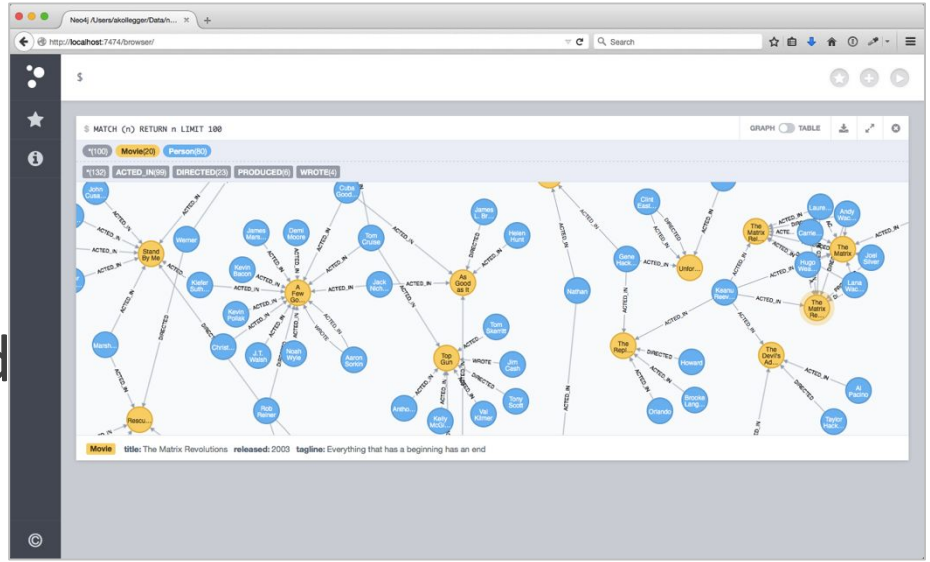
Because Relationships Matter

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

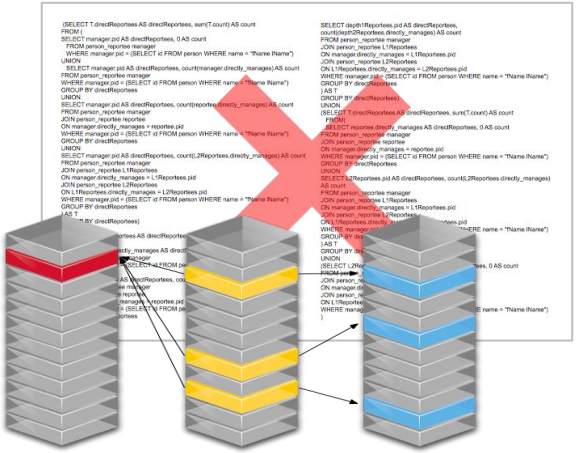
- Was built to efficiently
  - ***store***,
  - ***query*** and
  - ***manage*** highly connected
- Transactional, ACID
- Real-time OLTP
- Open source
- Highly scalable already on few machines



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

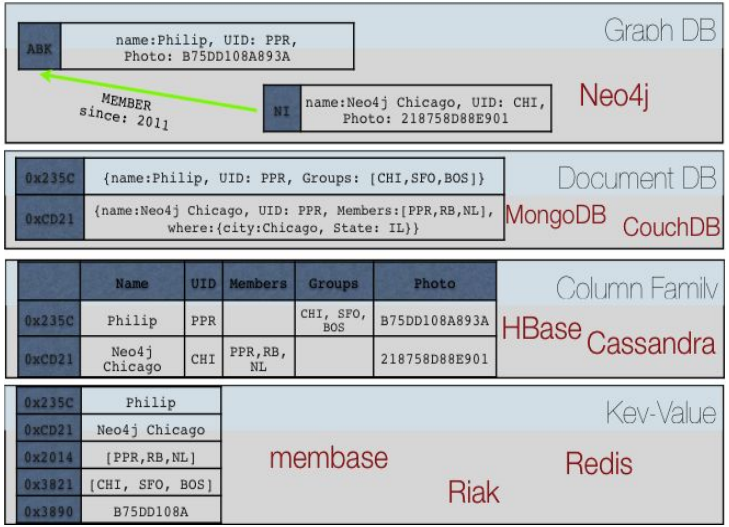
… making traditional databases **inappropriate** when data relationships are valuable in **real-time**



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**

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

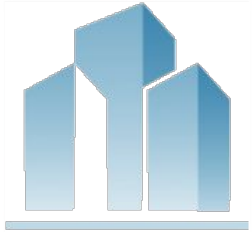
Common Use Cases

Internal Applications

Master Data Management

Network and
IT Operations

Fraud Detection



Customer-Facing Applications

Real-Time Recommendations

Graph-Based Search

Identity and
Access Management

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")
)
```