Introduction to Neo4j - a hands-on crash course

Lju Lazarevic
Developer Relations
🐦 @ElLazal

dev.neo4j.com/forum
dev.neo4j.com/chat

In this session

We will cover:

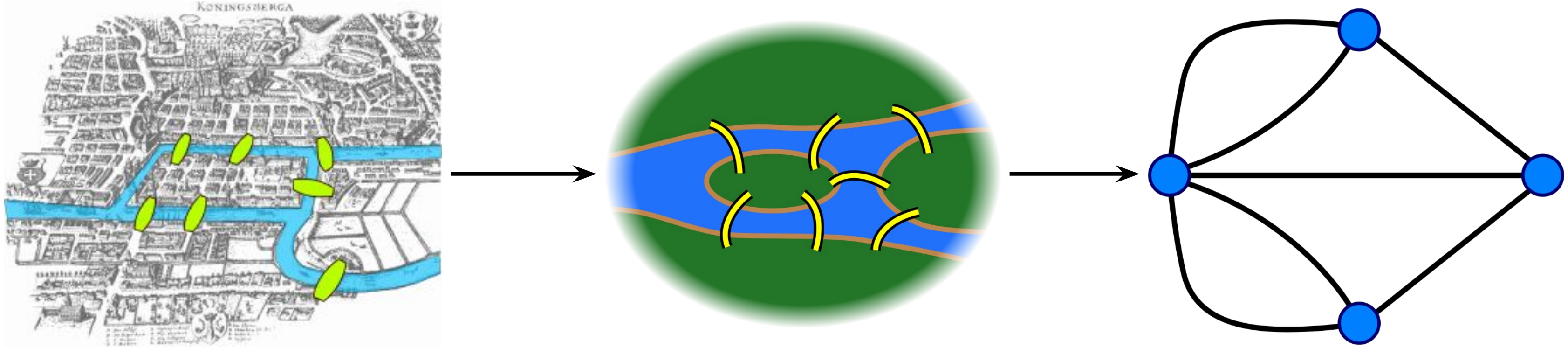
- What is a graph and why they are amazing
- Spotting good graph scenarios
- Property graph database anatomy and introduction to Cypher
- Hands-on: the movie graph on Neo4j Aura Free
  - dev.neo4j.com/aura-login
- Continuing your graph journey

Useful reference: https://dev.neo4j.com/rdbms-gdb

What is a graph?

A graph is...

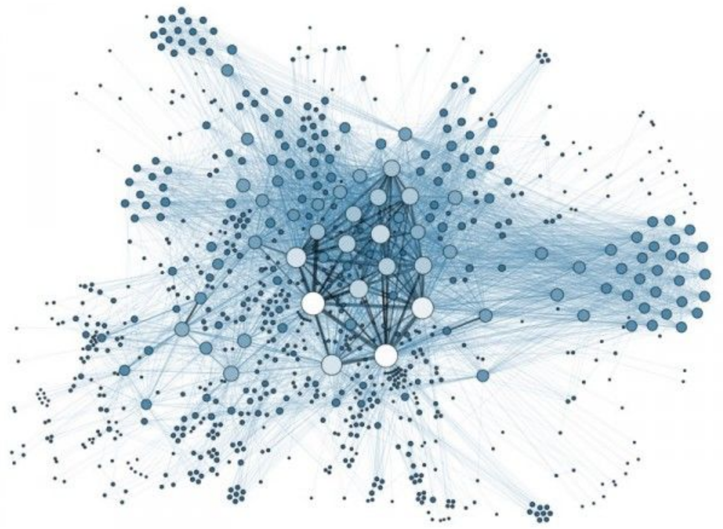
...a set of discrete objects, each of which has some set of relationships with the other objects



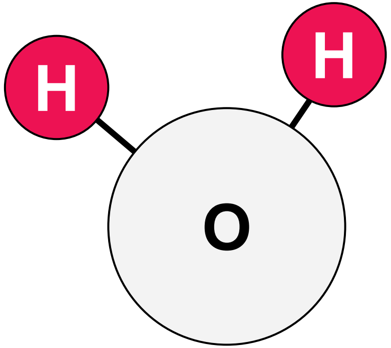
Seven Bridges of Konigsberg problem. Leonhard Euler, 1735

Anything can be a graph

the Internet

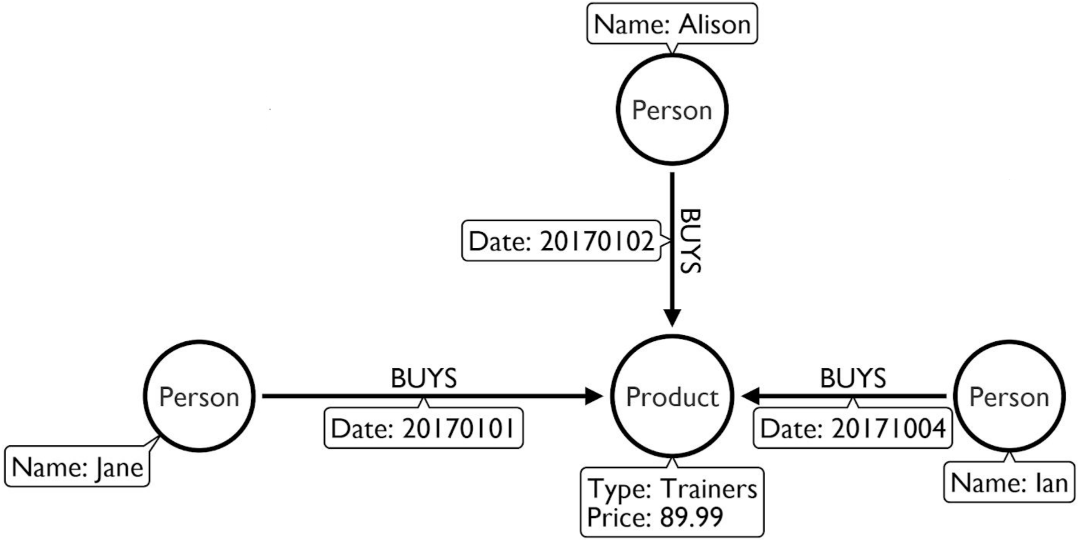


a water molecule



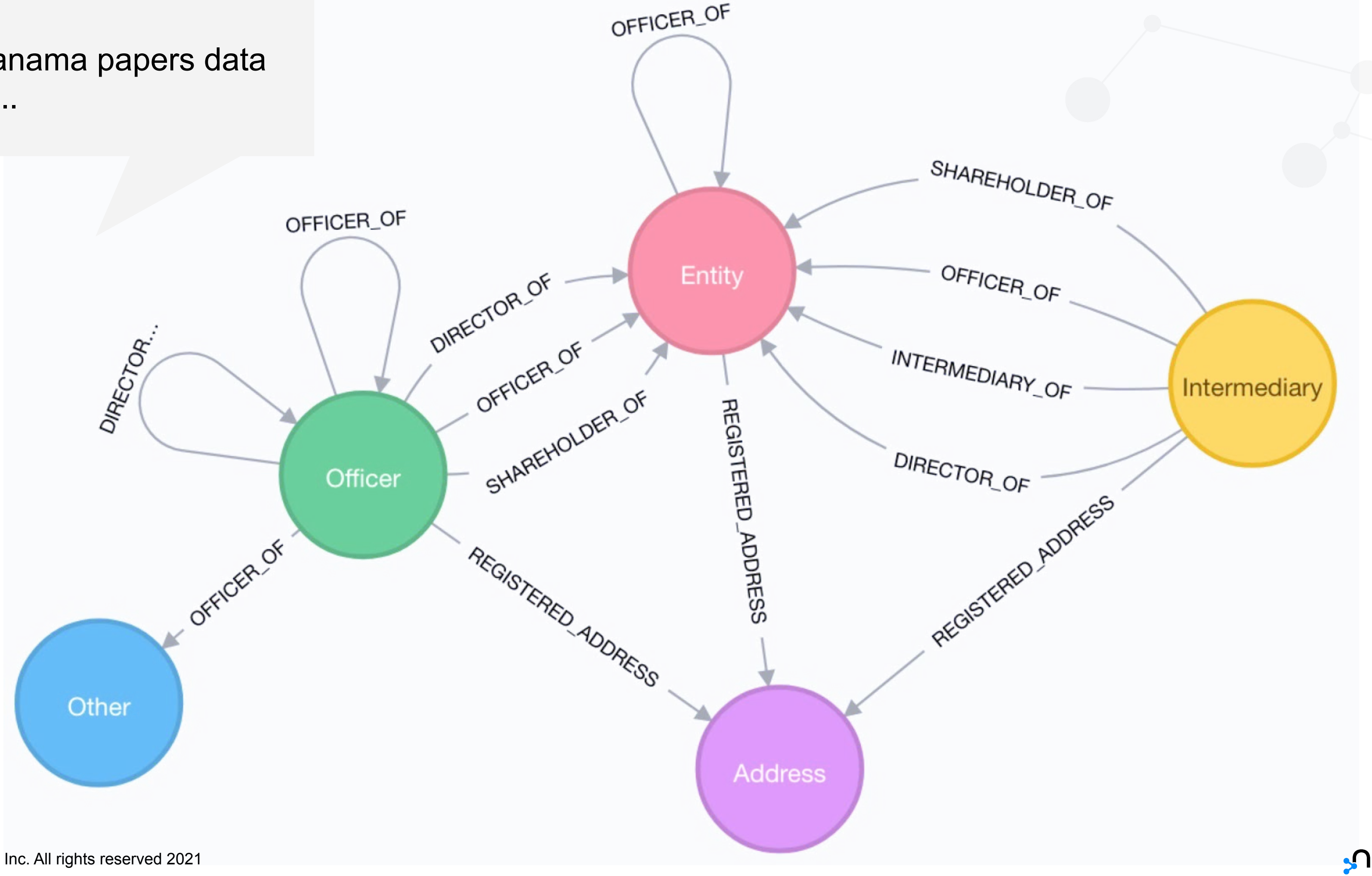
Why are graphs amazing?

Follow the flow - buying trainers



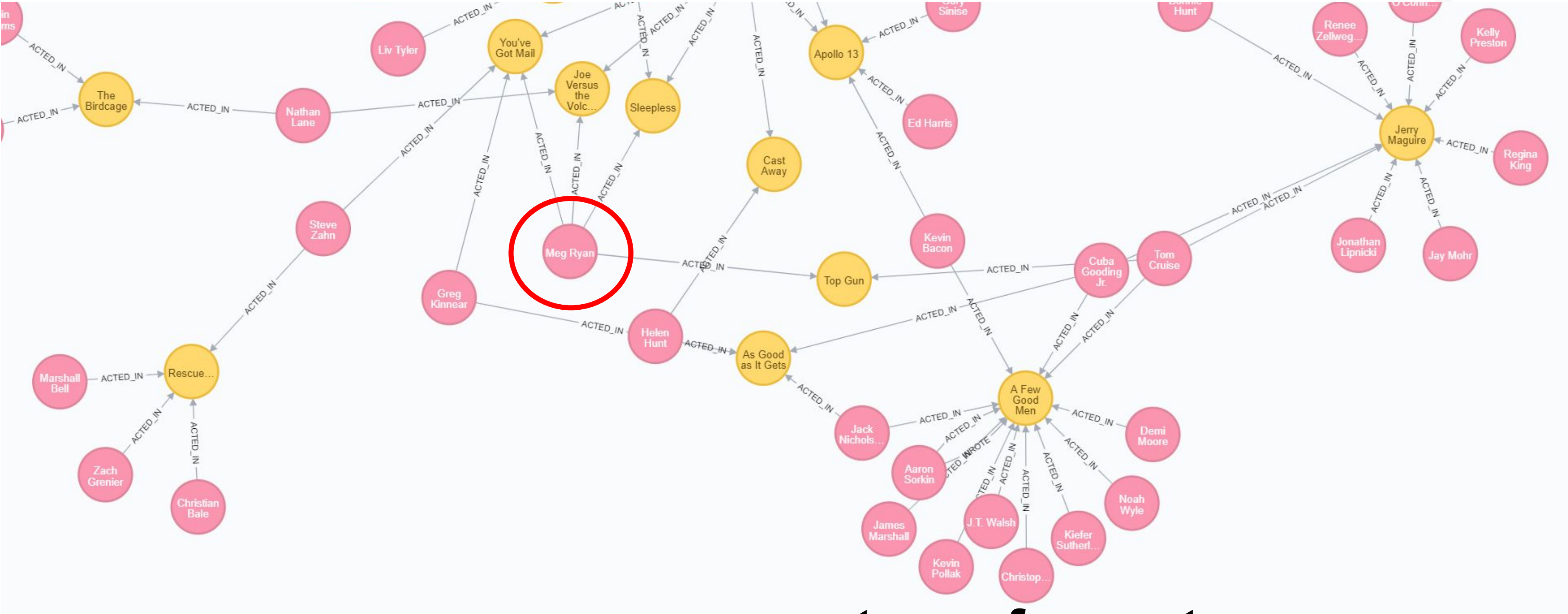
Panama papers:
simple model, powerful outcome

The Panama papers data model...



Roses are red,
facebook is blue,
No mutual friends,
So who are you?

Friends of friends

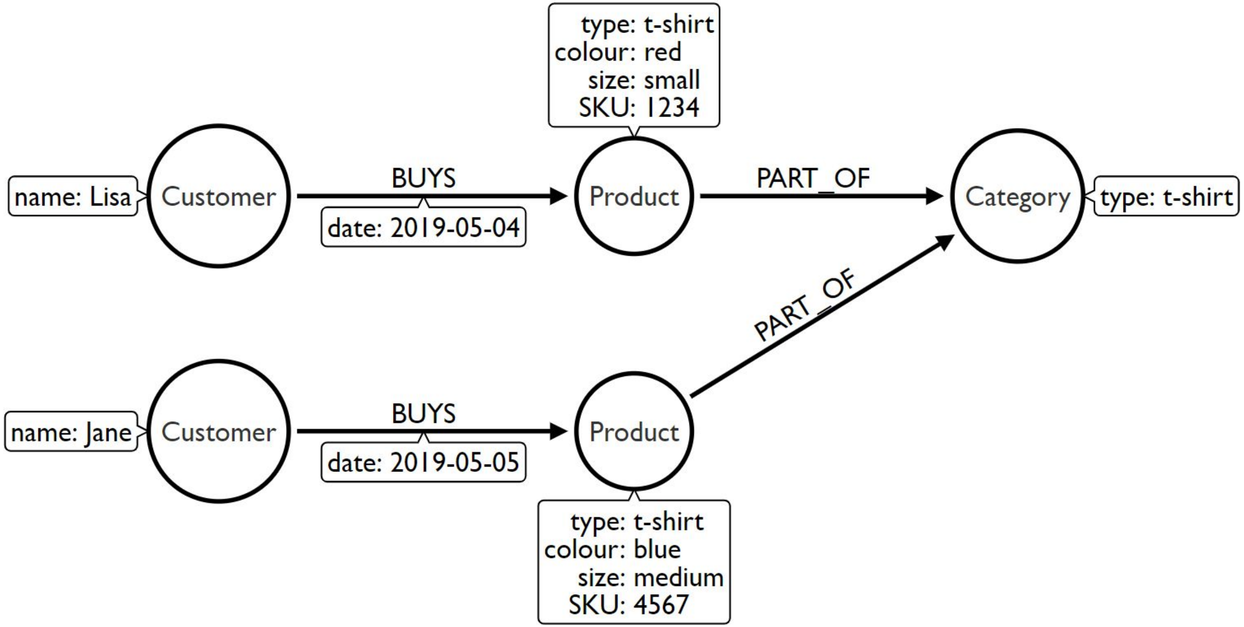


...or co-actors of co-actors

What are good graph scenarios?

Identifying good graph scenarios

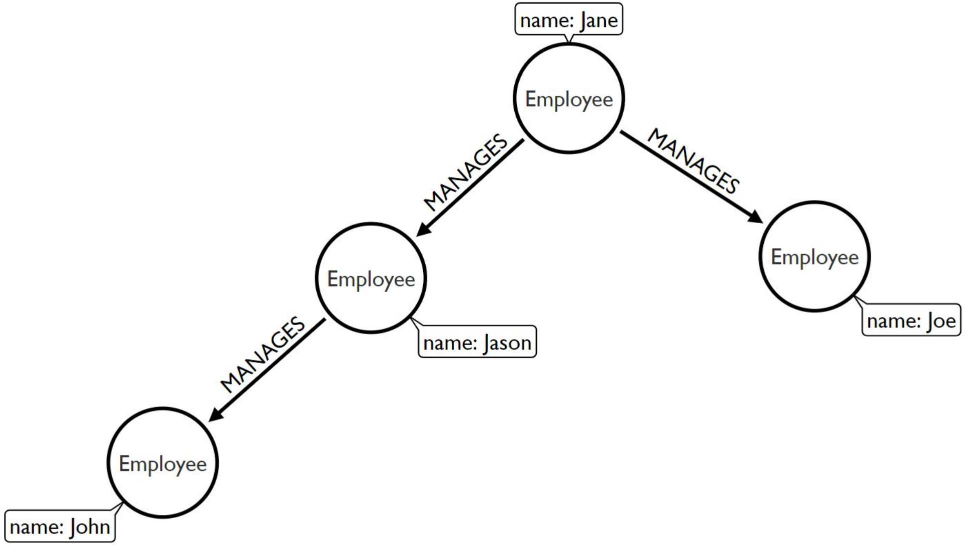
Scenario 1: Does our problem involve understanding relationships between entities?



- Recommendations
- Fraud detection
- Finding duplicates
- Data lineage

Identifying good graph scenarios

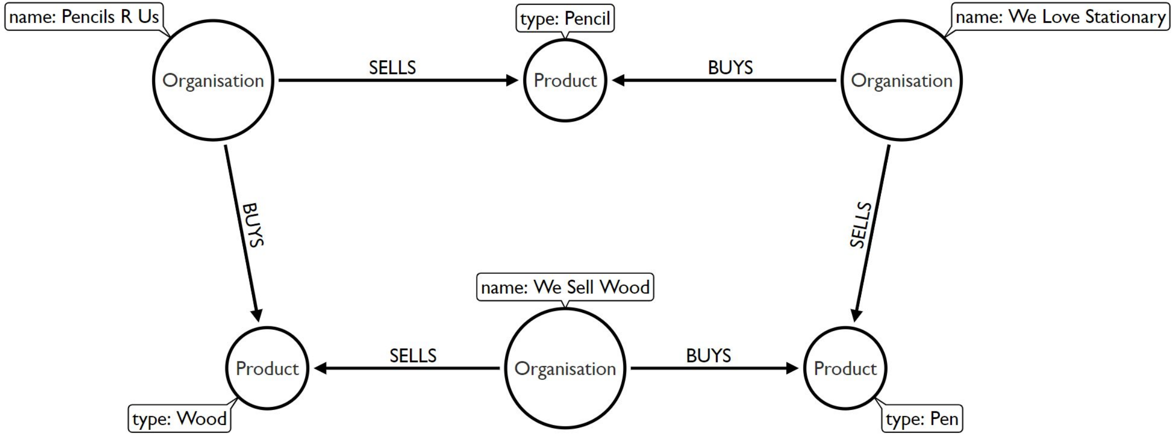
Scenario 2: Does the problem involve a lot of self-referencing to the same type of entity?



- Organisational hierarchies
- Access management
- Social influencers
- Friends of friends

Identifying good graph scenarios

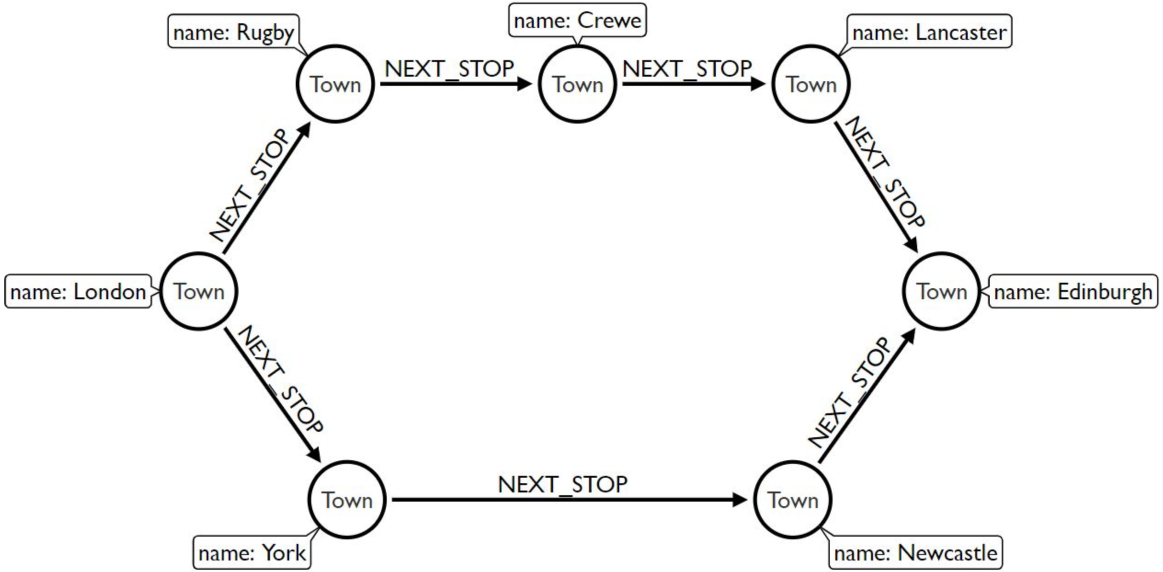
Scenario 3: Does the problem explore relationships of varying or unknown depth?



- Supply chain visibility
- Bill of Materials
- Network management

Identifying good graph scenarios

Scenario 4: Does our problem involve discovering lots of different routes or paths?



- Logistics and routing
- Infrastructure management
- Dependency tracing

So what does a (property) graph look like?

Graph components

Node (Vertex)

- The main data element from which graphs are constructed

Jane

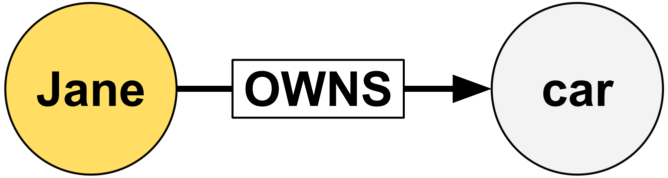
car

Graph components

Node (Vertex)

- The main data element from which graphs are constructed

Relationship (Edge)

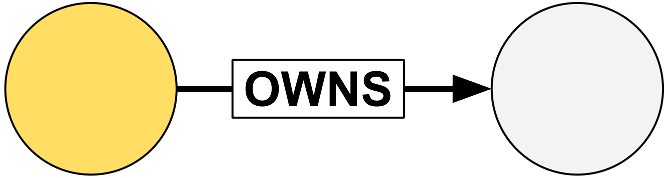


- A link between two nodes. Has:
    - Direction
    - Type
- *A node without relationships is permitted. A relationship without nodes is not*

Property graph database

Node (Vertex)

Relationship (Edge)



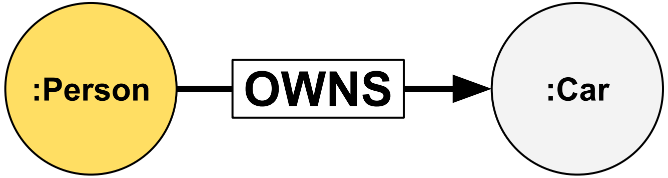
Property graph database

Node (Vertex)

Relationship (Edge)

Label

- Define node category (optional)



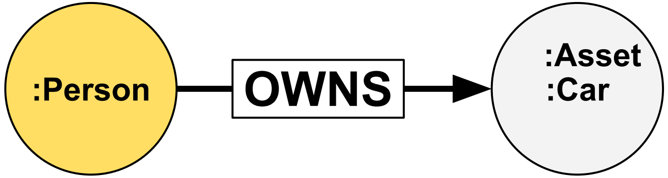
Property graph database

Node (Vertex)

Relationship (Edge)

Label

- Define node category (optional)
- Can have more than one



Property graph database

Node (Vertex)

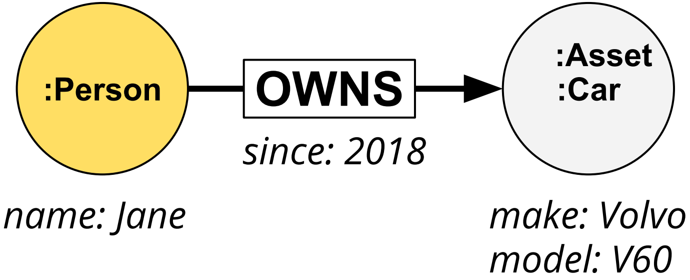
Relationship (Edge)

Label

- Define node category (optional)
- Can have more than one

Properties

- Enrich a node or relationship
- No need for nulls!



How do I query the graph?

Cypher

A pattern-matching query language made for graphs

Cypher

A pattern matching query language made for graphs

- Declarative
- Expressive
- Pattern-Matching

Cypher

A pattern matching query language made for graphs

- Declarative
- Expressive
- Pattern Matching

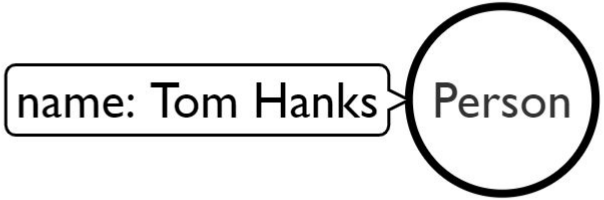
With ASCII ART ¯_(ツ)_/¯

Nodes and relationships at a glance

Description	Node	Relationship
Generic	()	-- --> -[]-
With a reference	(n)	-[r]-
With a node label or rel type	(:Person)	-[:ACTED_IN]-
With a label/type and an inline property	(:Person {name: ‘Bob’})	-[:ACTED_IN {role: ‘Dave’}]-
With a reference, label/type and an inline property	(p:Person {name: ‘Bob’})	-[r:ACTED_IN {role: ‘Rob’}]-

Use MATCH to retrieve nodes

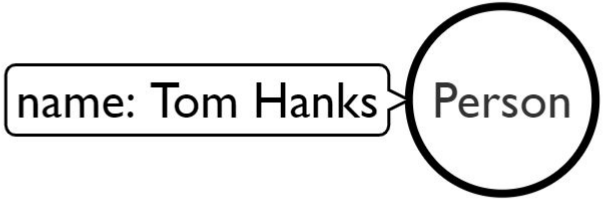
```
//Match all nodes
MATCH (n)
RETURN n;
```



Use MATCH to retrieve nodes

```
//Match all nodes
MATCH (n)
RETURN n;

//Match all nodes with a Person label
MATCH (n:Person)
RETURN n;
```

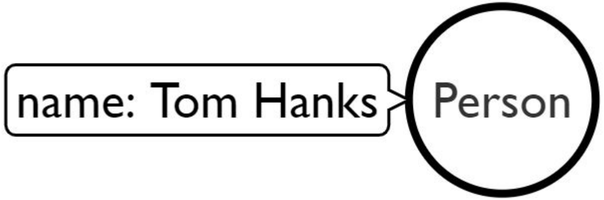


Use MATCH to retrieve nodes

```
//Match all nodes
MATCH (n)
RETURN n;

//Match all nodes with a Person label
MATCH (n:Person)
RETURN n;

//Match all nodes with a Person label and property name is "Tom Hanks"
MATCH (n:Person {name: "Tom Hanks"})
RETURN n;
```



Use MATCH and properties to retrieve nodes

```
//Return nodes with label Person and name property is "Tom Hanks" -
Inline
MATCH (p:Person {name: "Tom Hanks"}) //Only works with exact matches
RETURN p;
```

Use MATCH and properties to retrieve nodes

```
//Return nodes with label Person and name property is "Tom Hanks" -
Inline
MATCH (p:Person {name: "Tom Hanks"}) //Only works with exact matches
RETURN p;
```

```
//Return nodes with label Person and name property equals "Tom Hanks"
MATCH (p:Person)
WHERE p.name = "Tom Hanks"
RETURN p;
```

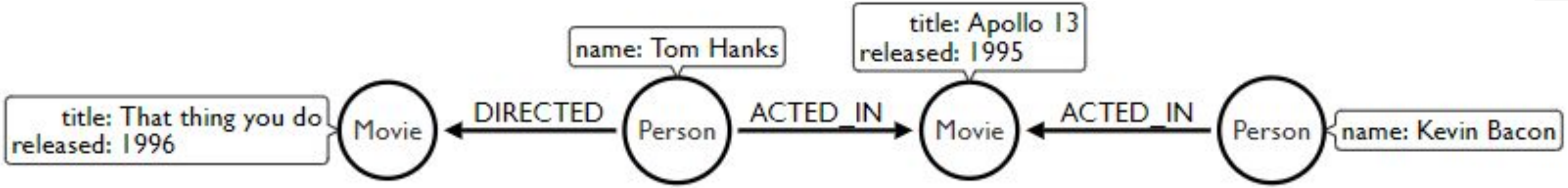
Use MATCH and properties to retrieve nodes

```
//Return nodes with label Person and name property is "Tom Hanks" - Inline
MATCH (p:Person {name: "Tom Hanks"}) //Only works with exact matches
RETURN p;
```

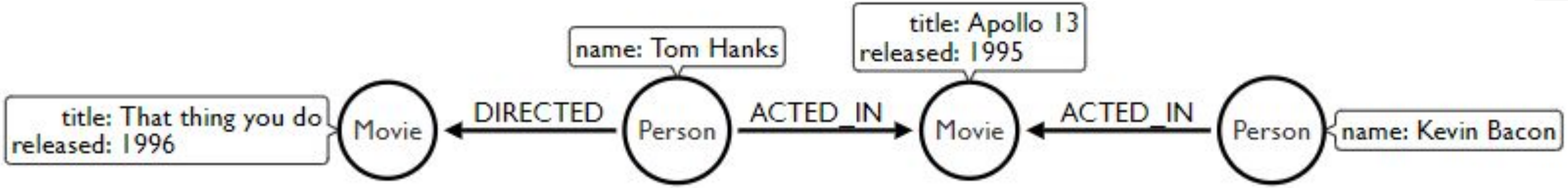
```
//Return nodes with label Person and name property equals "Tom Hanks"
MATCH (p:Person)
WHERE p.name = "Tom Hanks"
RETURN p;
```

```
//Return nodes with label Movie, released property is between 1991 and 1999
MATCH (m:Movie)
WHERE m.released > 1990 AND m.released < 2000
RETURN m;
```

Extending the MATCH

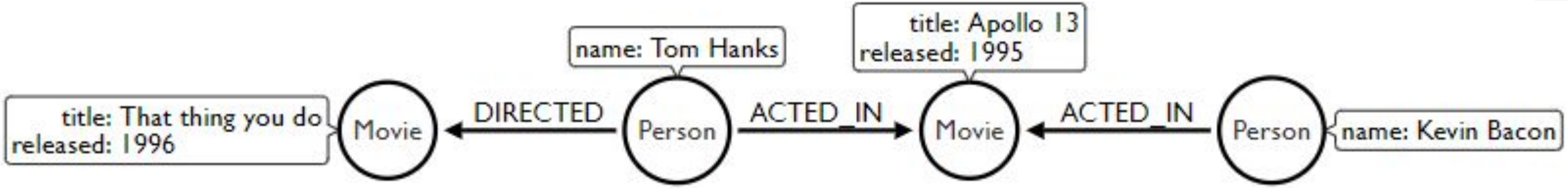


Extending the MATCH



```
MATCH (:Person {name:"Tom Hanks"})--(m:Movie)
RETURN m.title;
```

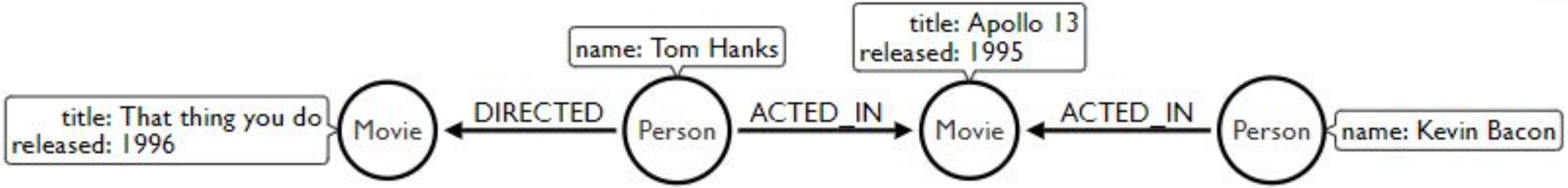
Extending the MATCH



```
MATCH (:Person {name:"Tom Hanks"})--(m:Movie)
RETURN m.title;
```

```
//Find all the movies Tom Hanks directed and order by latest movie
MATCH (:Person {name:"Tom Hanks"})-[:DIRECTED]-(m:Movie)
RETURN m.title, m.released ORDER BY m.released DESC;
```

Extending the MATCH



```
MATCH (:Person {name:"Tom Hanks"})--(m:Movie)
RETURN m.title;
```

```
//Find all the movies Tom Hanks directed and order by latest movie
MATCH (:Person {name:"Tom Hanks"})-[:DIRECTED]-(m:Movie)
RETURN m.title, m.released ORDER BY m.released DESC;
```

```
//Find all of the co-actors Tom Hanks have worked with
MATCH (:Person {name:"Tom Hanks"})--(:Movie)-[:ACTED_IN]-(coActor:Person)
RETURN coActor.name;
```

CREATE

```
//Create a person node called "Tom Hanks"
CREATE (p:Person {name:"Tom Hanks"});
```

CREATE

```
//Create a person node called "Tom Hanks"
CREATE (p:Person {name:"Tom Hanks"});

//Create an ACTED_IN relationship between "Tom Hanks" and "Apollo 13"
MATCH (p:Person {name:"Tom Hanks"}), (m:Movie {title:"Apollo 13"})
CREATE (p)-[:ACTED_IN]->(m);
```

CREATE

```
//Create a person node called "Tom Hanks"
CREATE (p:Person {name:"Tom Hanks"});

//Create an ACTED_IN relationship between "Tom Hanks" and "Apollo 13"
MATCH (p:Person {name:"Tom Hanks"}), (m:Movie {title:"Apollo 13"})
CREATE (p)-[:ACTED_IN]->(m);

//Create the pattern of "Tom Hanks" ACTED_IN "Apollo 13"
//This will create the entire pattern, nodes and all!
CREATE (:Person {name:"Tom Hanks")-[:ACTED_IN]->(:Movie {title:"Apollo
13});
```

Time to have a go!

We are going to:

- Go to dev.neo4j.com/aura-login
- Sign in & click “Create a database”
- Give your database a name
- Selected “Shared” database size
- Click “Create Database”
- Make a copy of the generated password - keep it safe!

Can’t access Aura Free? No problem! Use Neo4j Sandbox:

- Go to dev.neo4j.com/try
- Sign in & click “Blank sandbox”

So how do I continue my graph journey?

More training this week - all starting at 1pm UTC

Tuesday: Hands-on with Neo4j Aura Free Tier

Wednesday: Getting Started with Neo4j Bloom

Thursday: Build APIs with Neo4j GraphQL Library

Friday: Create a Knowledge Graph: A Simple ML Approach

Read all about it!
https://dev.neo4j.com/training-week

Continue your journey

Free online training and certification:

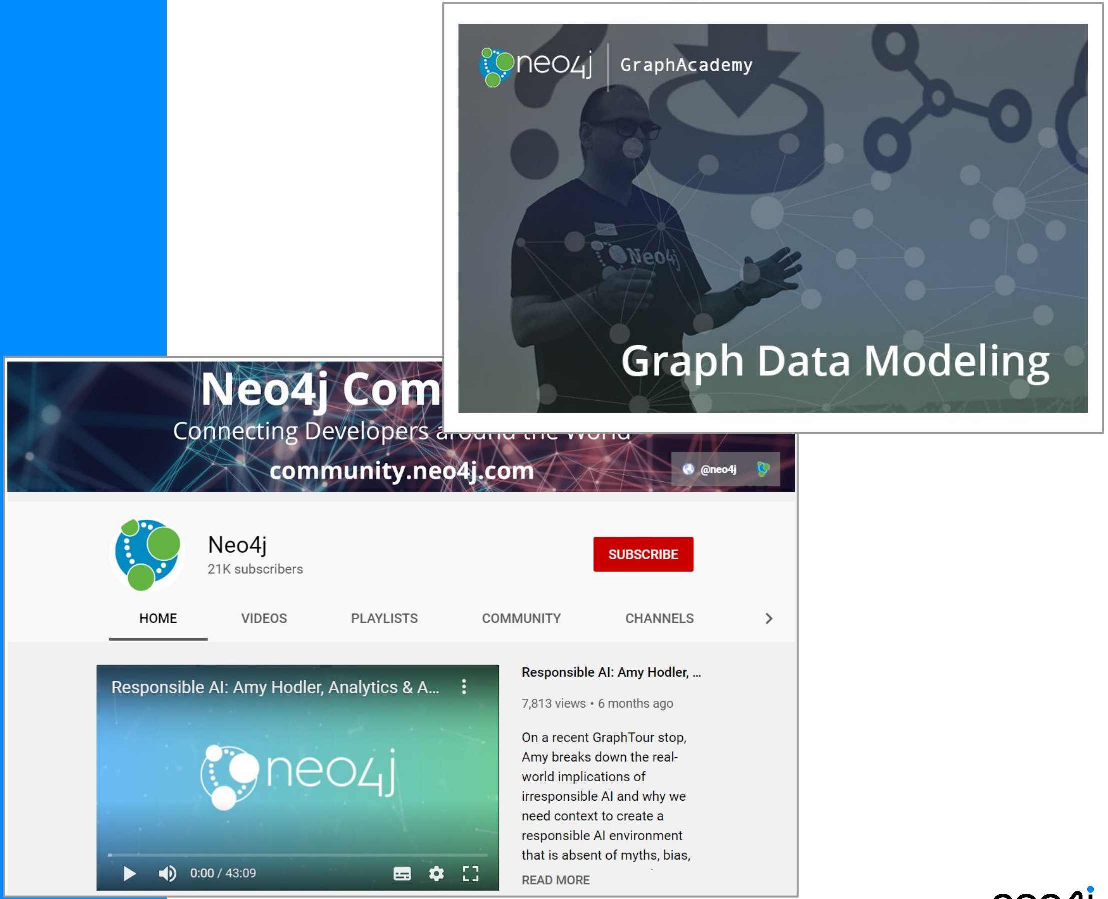
- *dev.neo4j.com/learn*

How to, best practices, hands on and community stories:

- *dev.neo4j.com/videos*

Come say hello :)

- *dev.neo4j.com/chat*
- *dev.neo4j.com/forum*



Neo4j
21K subscribers

Responsible AI: Amy Hodler, ...

7,813 views • 6 months ago

On a recent GraphTour stop, Amy breaks down the real-world implications of irresponsible AI and why we need context to create a responsible AI environment that is absent of myths, bias,

READ MORE

Ljubica Lazarevic

Developer Relations

lju@neo4j.com

@ElLazal

Join the conversation at **dev.neo4j.com/forum**