Introduction to Neo4j - a hands-on crash course

Michael Hunger
Developer Relations
@mesirii

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

This is an *interactive* session!
Tell us where you're *from!*

And please ask *Questions* as we go along!

In any of the chats

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 AuraDB Free
  - dev.neo4j.com/aura-login
- Sneak peek: stackoverflow data
- Continuing your graph journey

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

Because it takes a few minutes to launch

Let's get started on AuraDB Free

dev.neo4j.com/aura-login

Ask in the chat if something doesn't work!

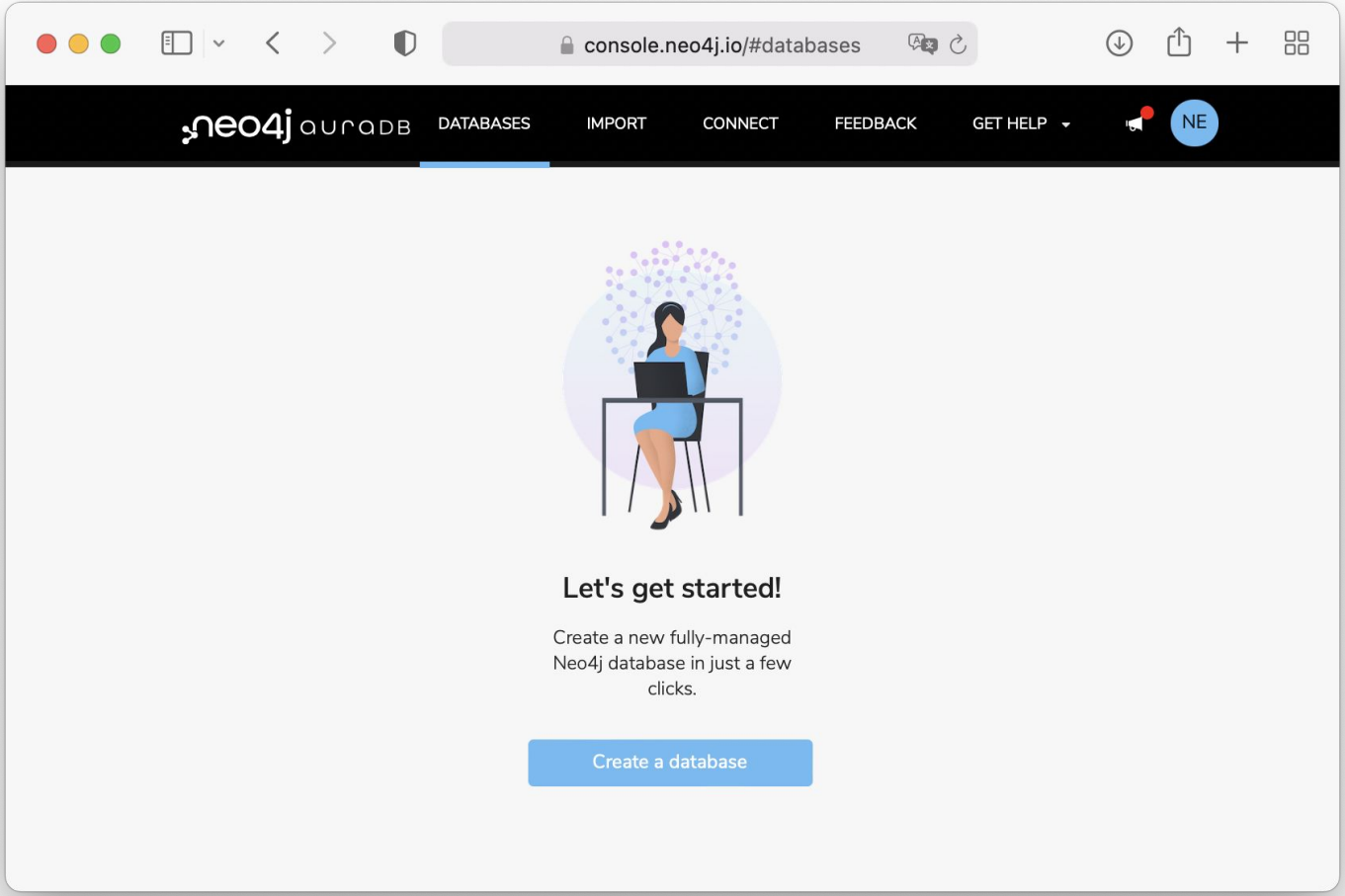
Time to have a go! With AuraDB Free

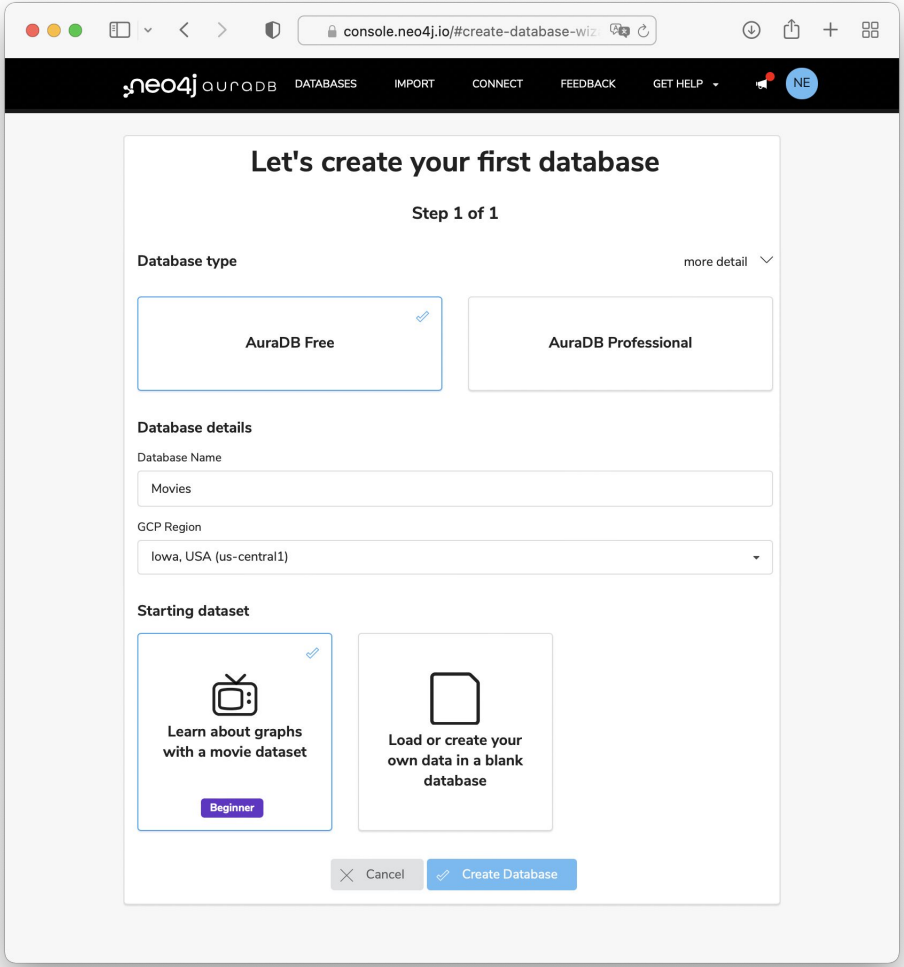
We are going to:

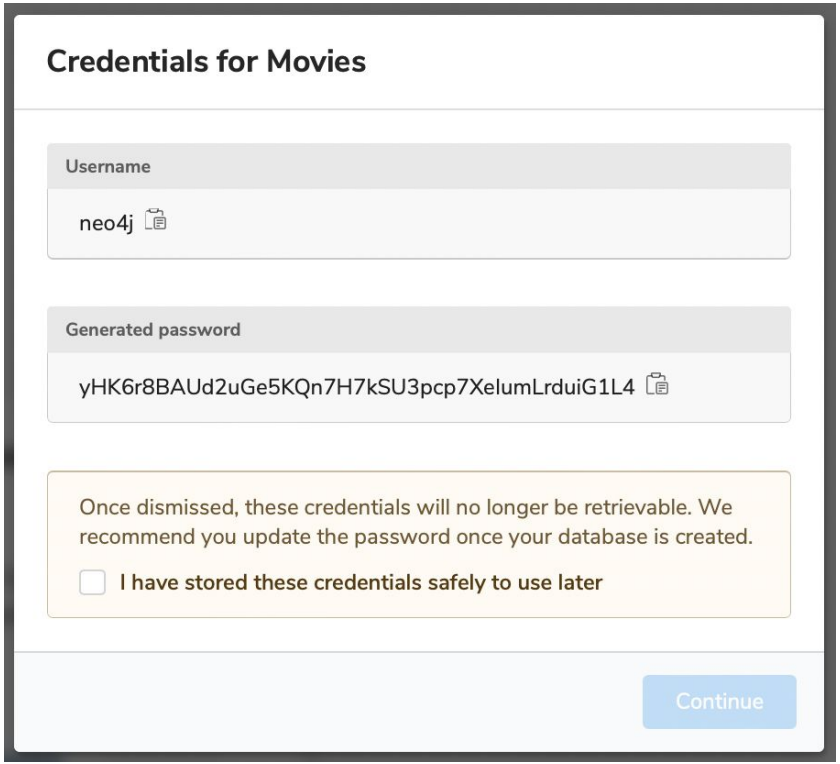
1. Go to dev.neo4j.com/aura-login
2. Sign in & click “Create a database”
3. Selected “AuraDB Free” database size
4. Give your database a name
5. Select a Region
6. **Select Movies Database**
7. Click “Create Database”
8. Make a **copy of the generated password** - keep it safe!
9. Wait for 3-5 minutes

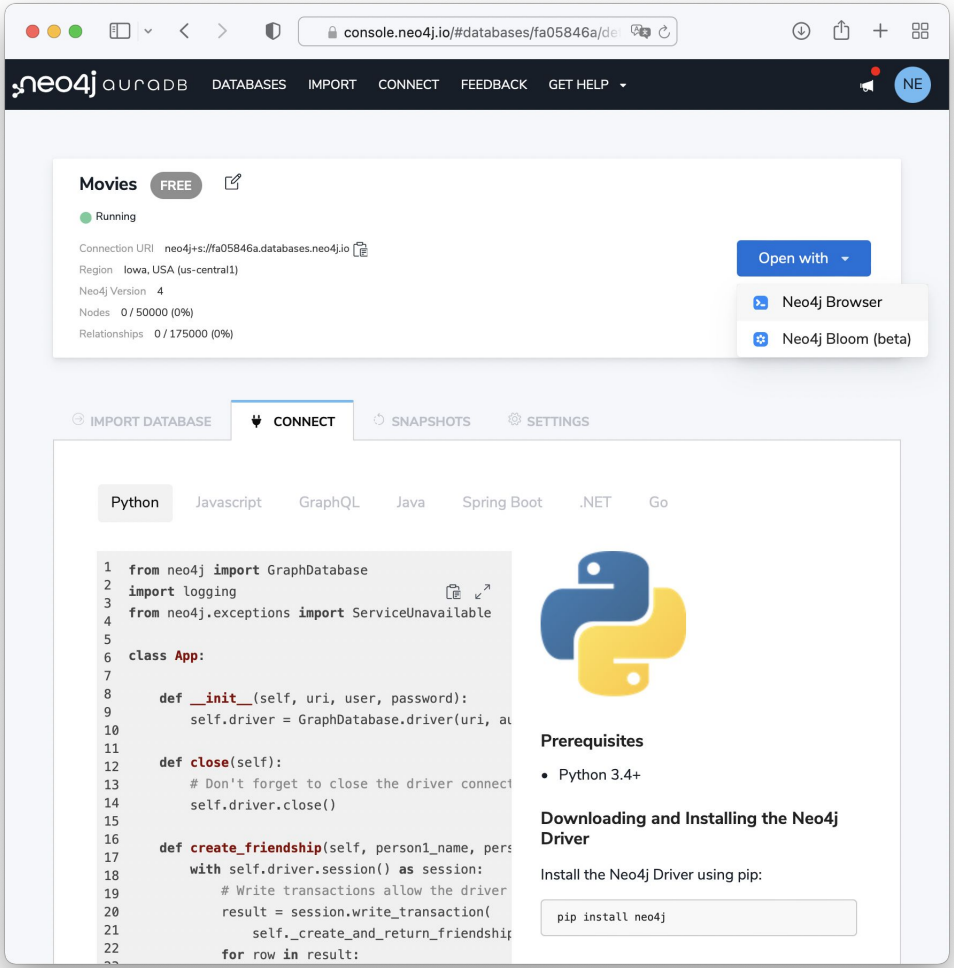
Can’t access Aura Free? No problem! Use Neo4j Sandbox:

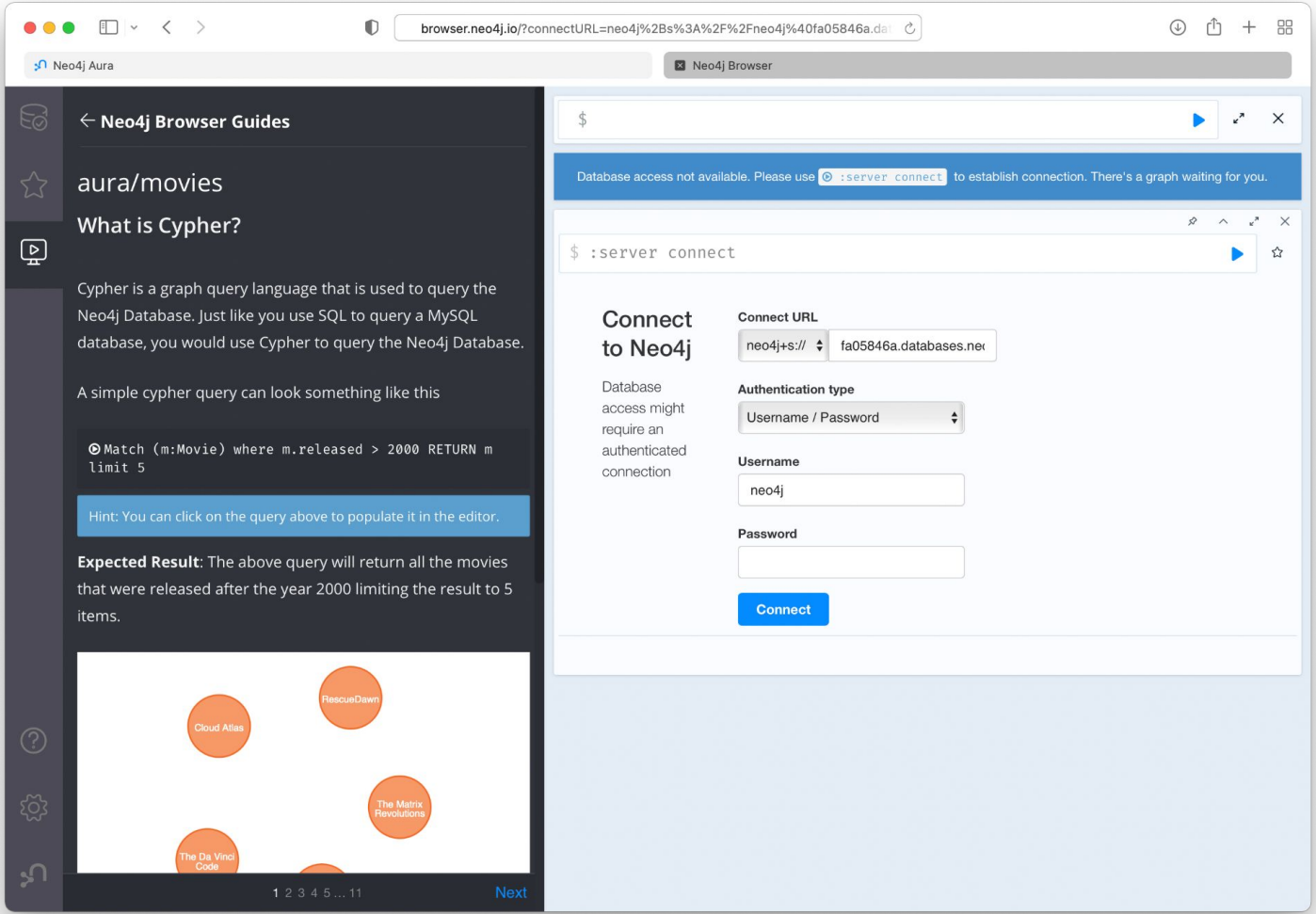
- Go to dev.neo4j.com/try
- Sign in & click “Blank sandbox”



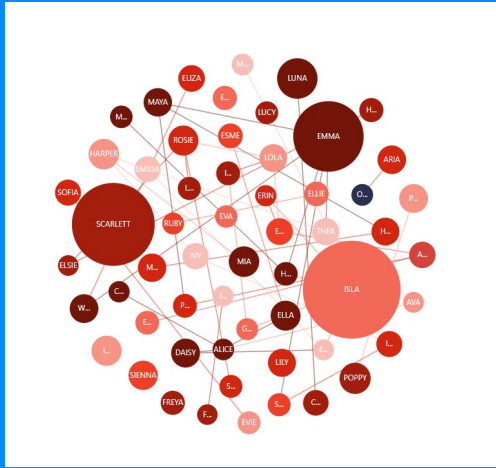




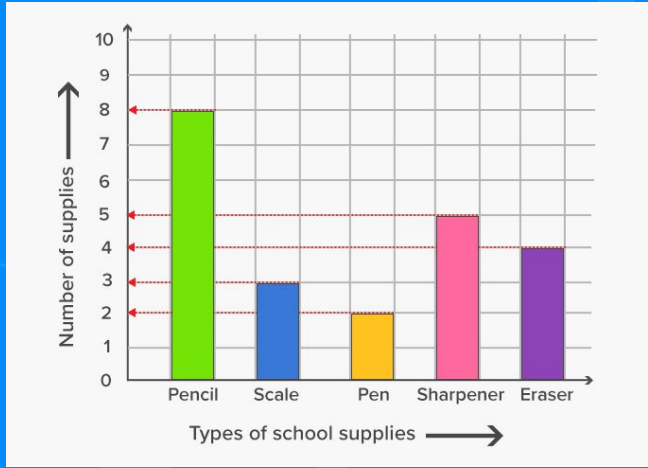




What is a graph?

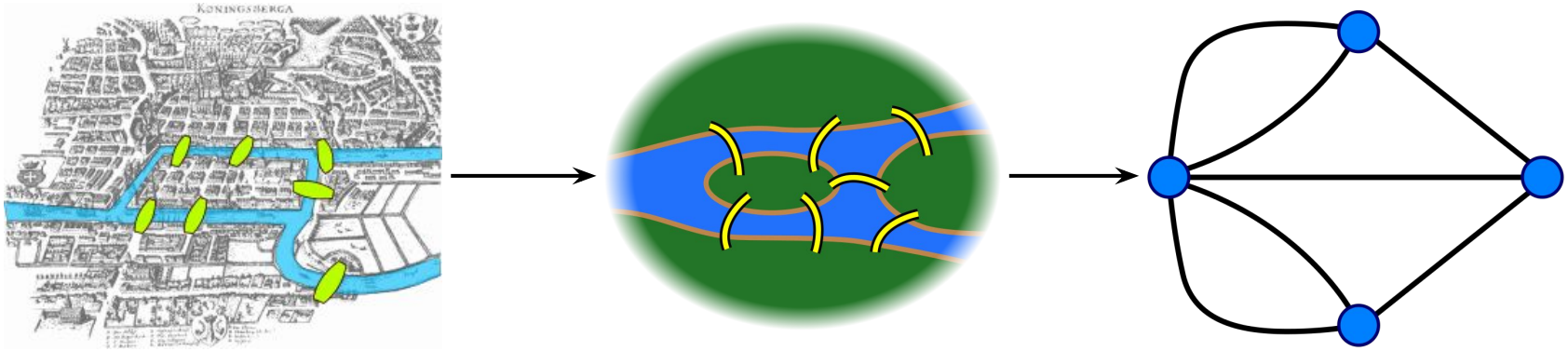


versus



A graph is...

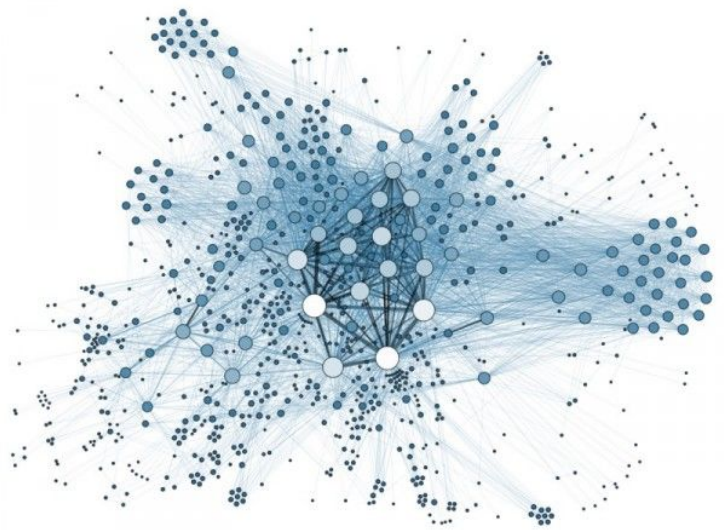
...a set of discrete entities, each of which has some set of relationships with the other entities



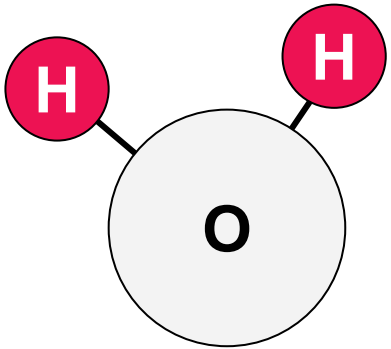
Seven Bridges of Konigsberg problem. Leonhard Euler, 1735

Anything can be a graph - do you have examples too?

the Internet

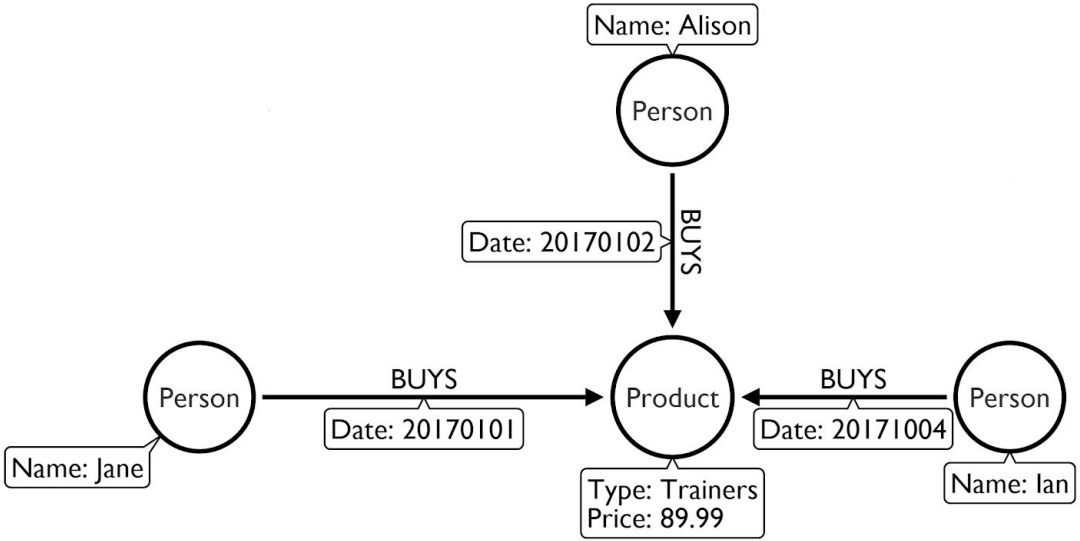


a water molecule



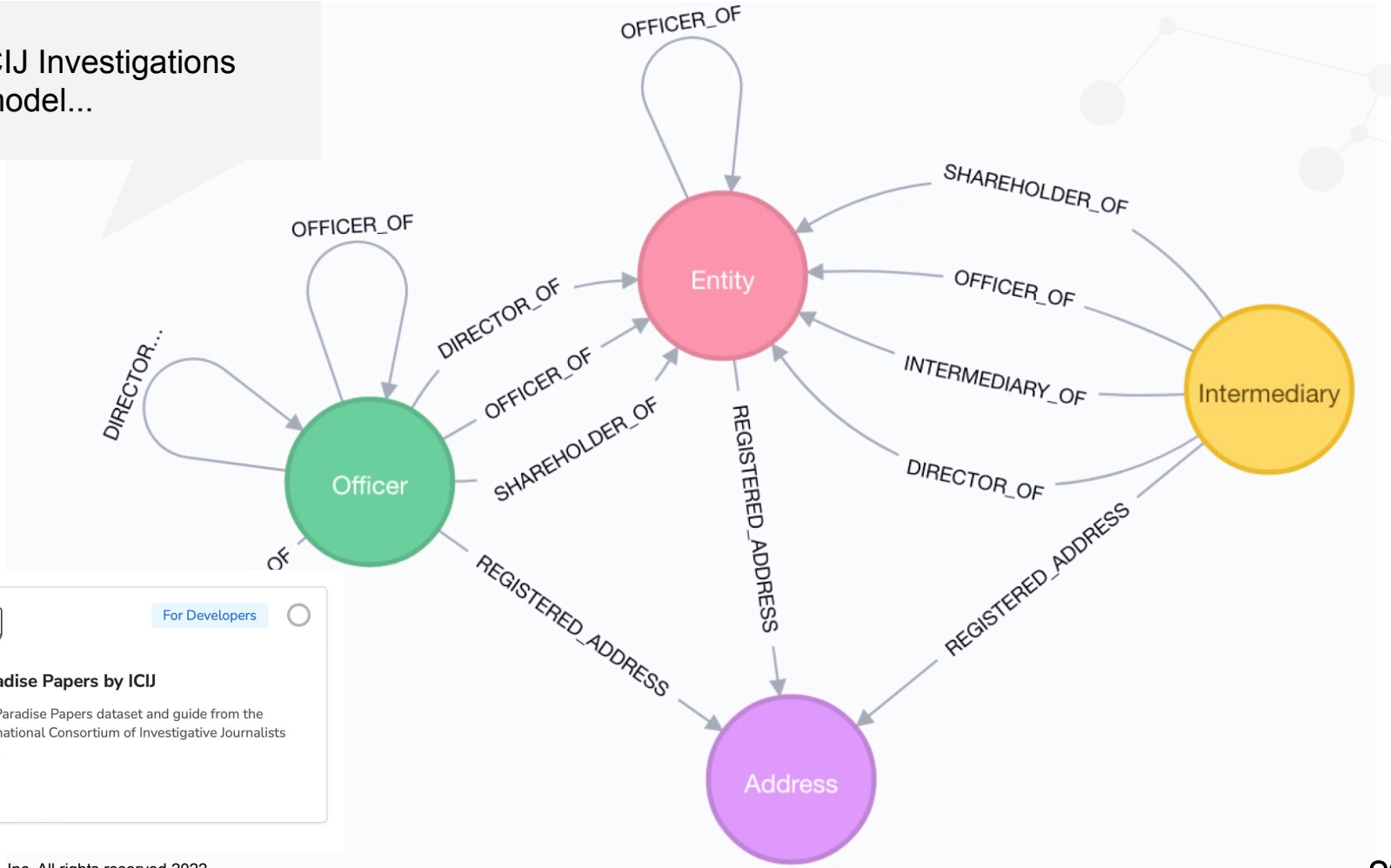
Why are graphs amazing?

Follow the flow - buying trainers



Panama, paradise, pandora papers: simple model, powerful outcome

The ICIJ Investigations data model...



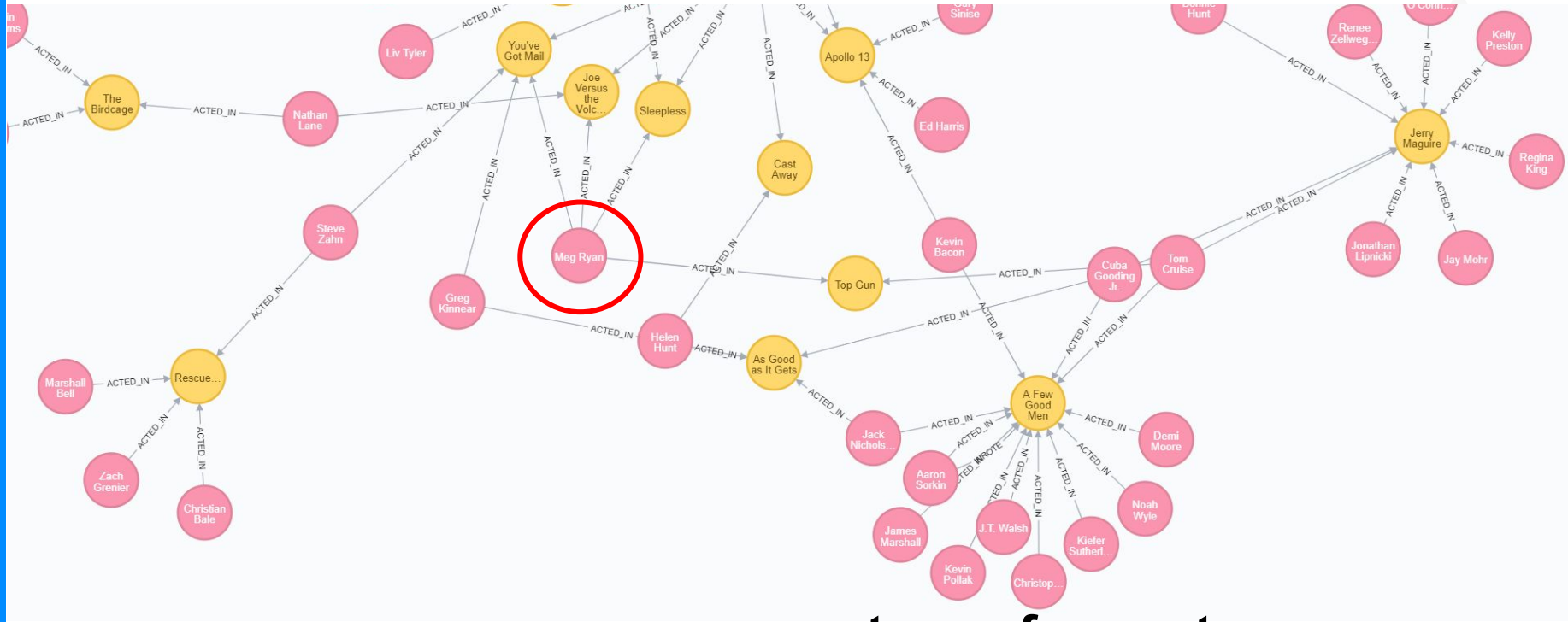
For Developers

Paradise Papers by ICIJ

The Paradise Papers dataset and guide from the International Consortium of Investigative Journalists (ICIJ).

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

Friends of friends

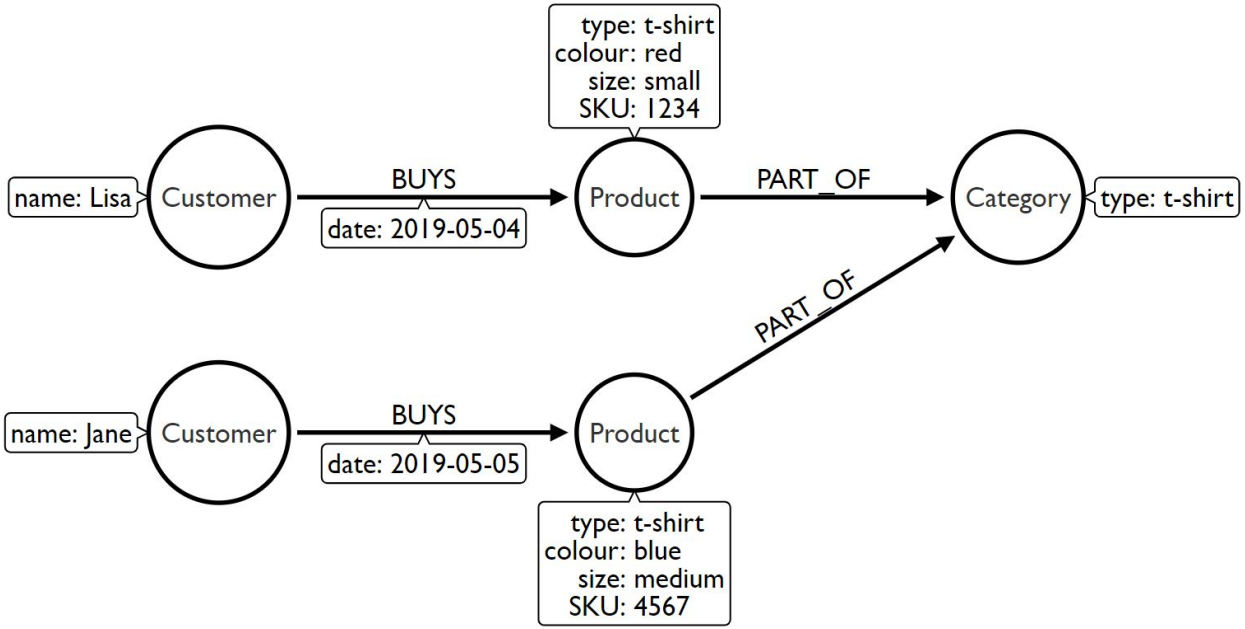


...or co-actors of co-actors

What are good graph scenarios?

Identifying good graph scenarios

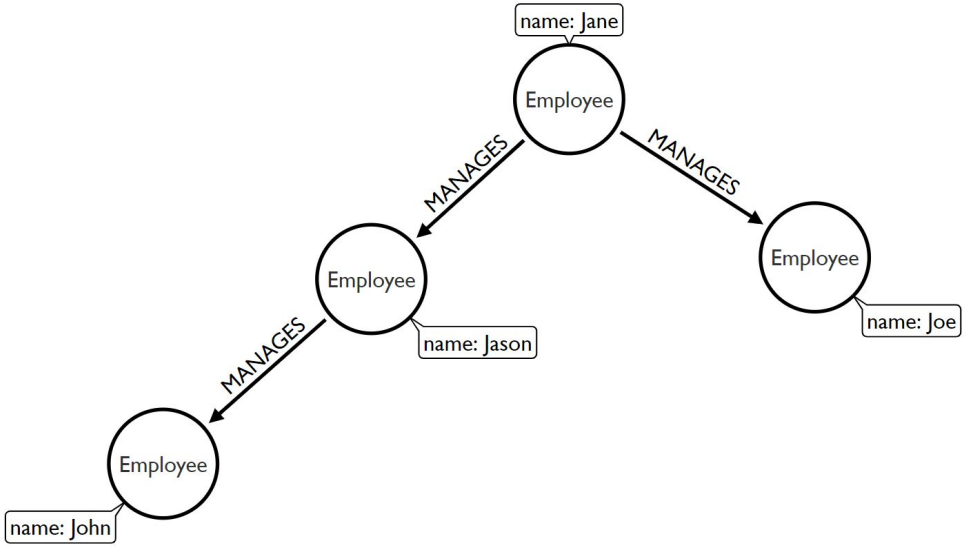
Scenario 1: Does our problem involve understanding relationships between entities?



- Recommendations
- Fraud detection
- Finding duplicates
- Data lineage
- Social Networks

Identifying good graph scenarios

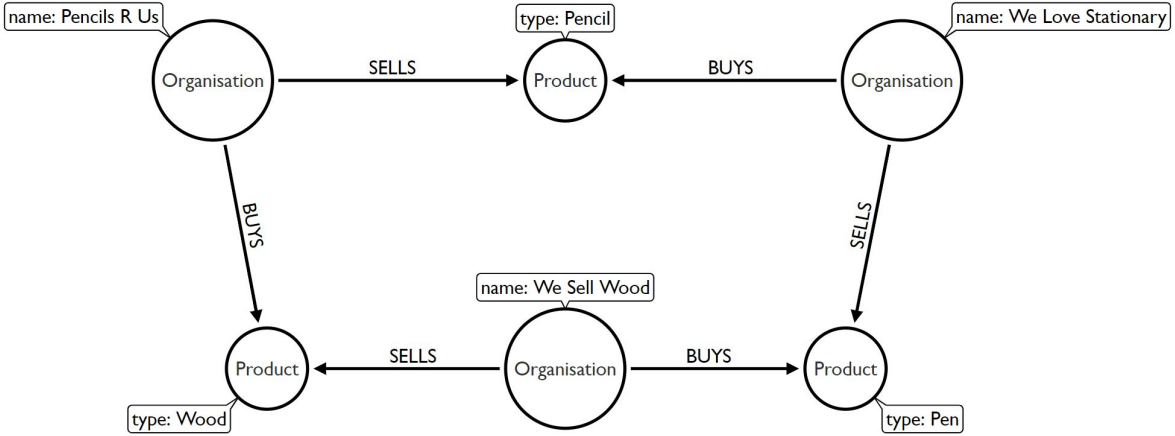
Scenario 2: Does the problem involve a lot of self-referencing to the same type of entity?



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

Identifying good graph scenarios

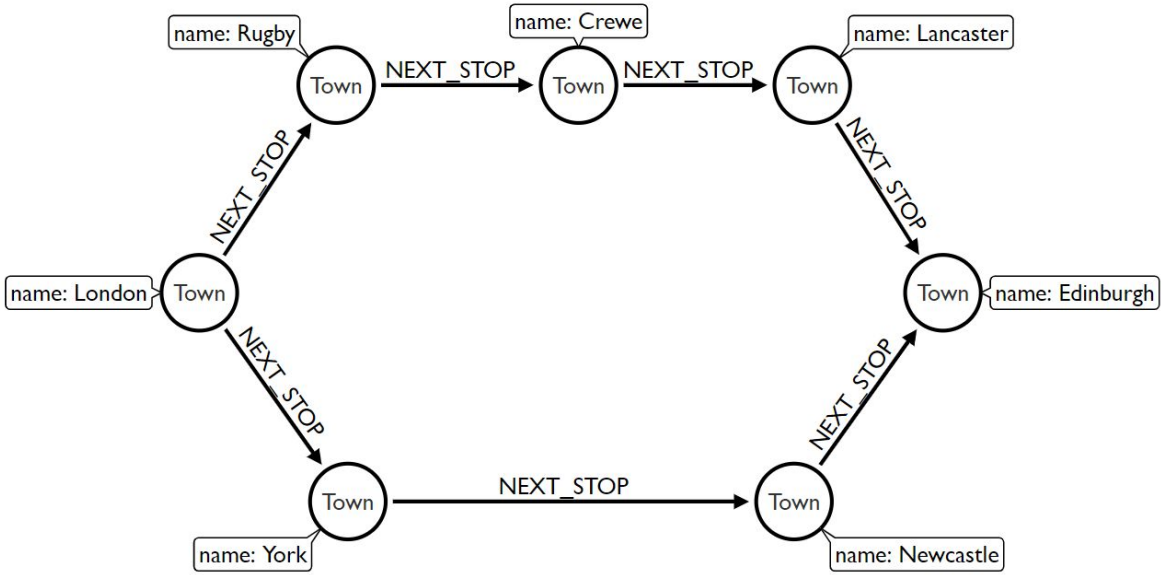
Scenario 3: Does the problem explore relationships of varying or unknown depth?



- Supply chain visibility
- Bill of Materials
- Network management
- Routing

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)

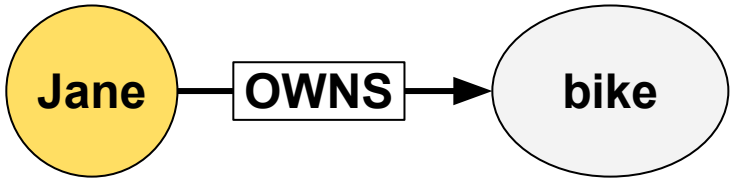
- The main data element from which graphs are constructed

Graph components

Node (Vertex)

- The main data element from which graphs are constructed

Relationship (Edge)

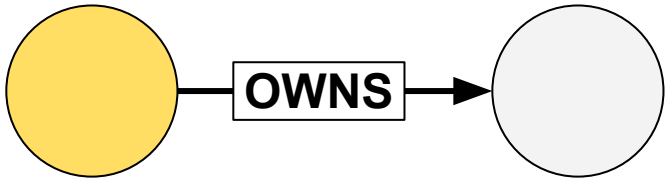


- A link between two nodes. Has:
  - Direction
  - Type
- *A node without relationships is permitted. A relationship without nodes is not*

Property graph database

Node (Vertex)

Relationship (Edge)



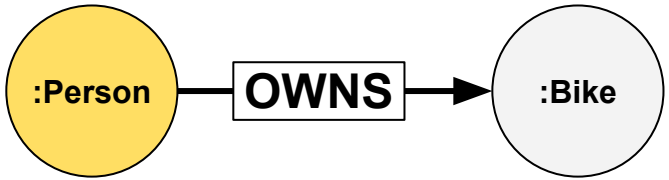
Property graph database

Node (Vertex)

Relationship (Edge)

Label

- Define node role (optional)



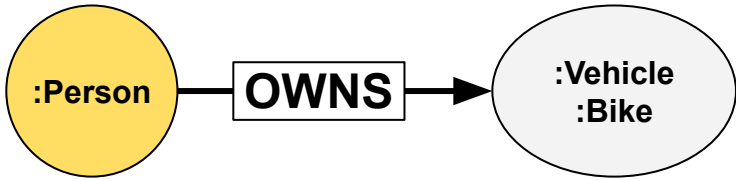
Property graph database

Node (Vertex)

Relationship (Edge)

Label

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



Property graph database

Node (Vertex)

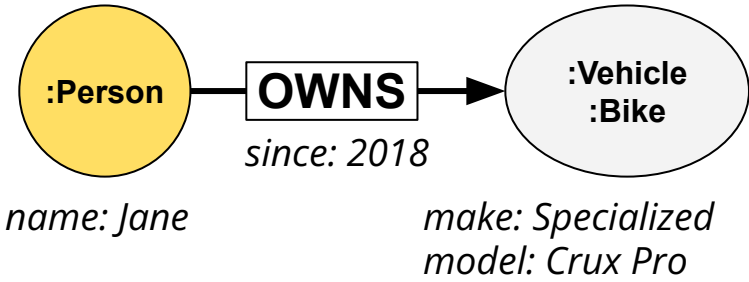
Relationship (Edge)

Label

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

Properties

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



And now query the graph together!

Cypher

A **pattern**-matching query language made for graphs

Cypher

A **pattern** matching query language made for graphs

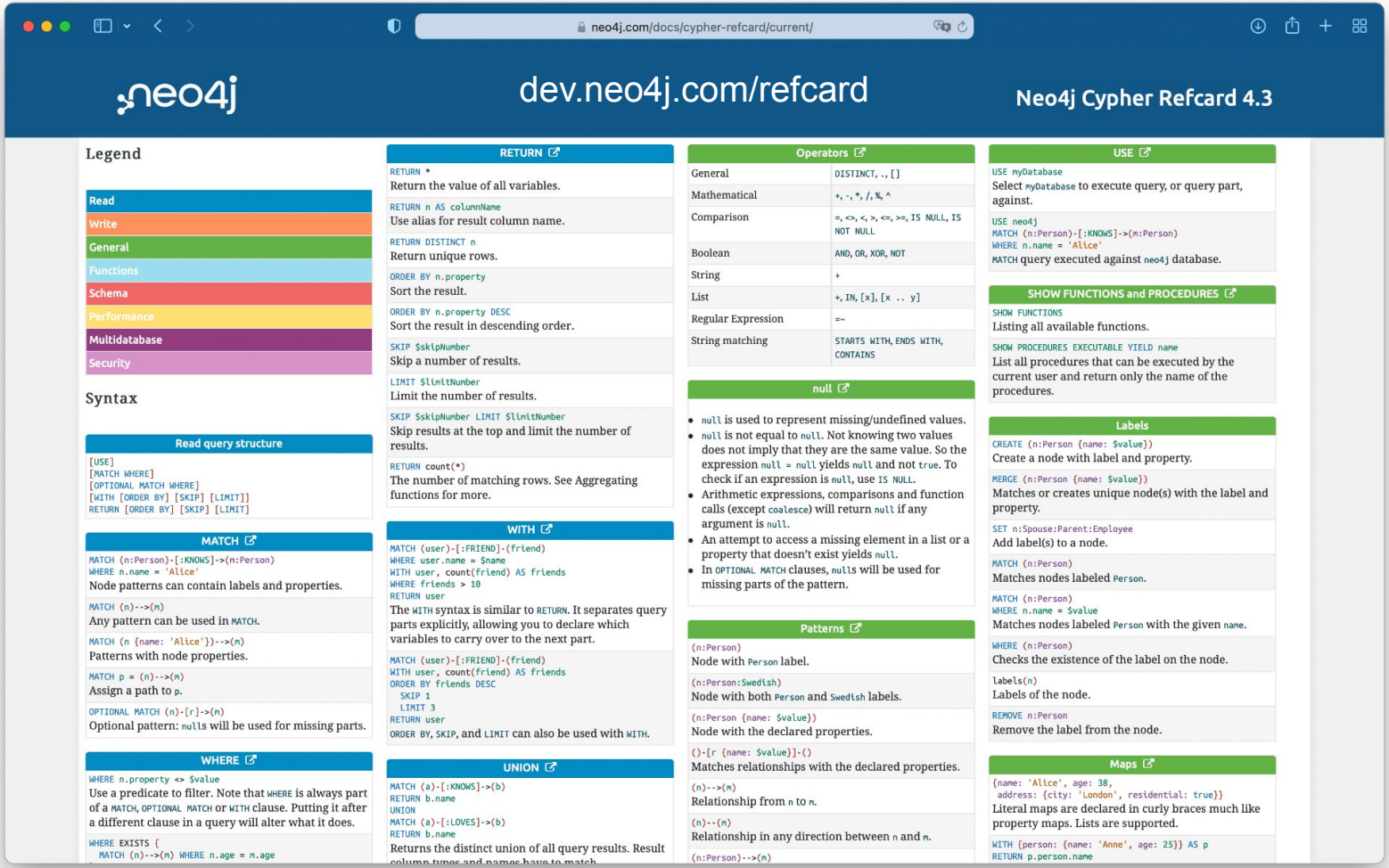
- Declarative
- Expressive
- Pattern-Matching

Cypher

A pattern matching query language made for graphs

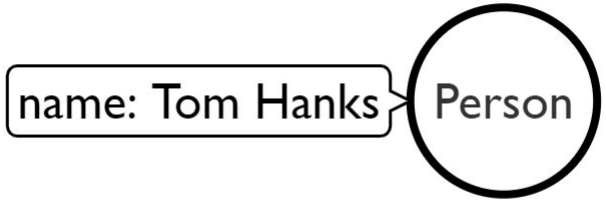
- Declarative
- Expressive
- Pattern Matching

With ASCII ART ¯_(ツ)_/¯



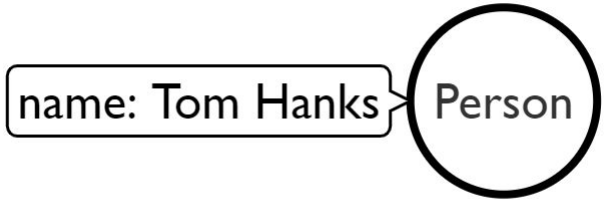
Use MATCH to retrieve nodes

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



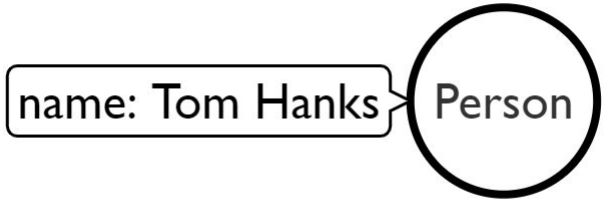
Use MATCH to retrieve nodes

```
//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

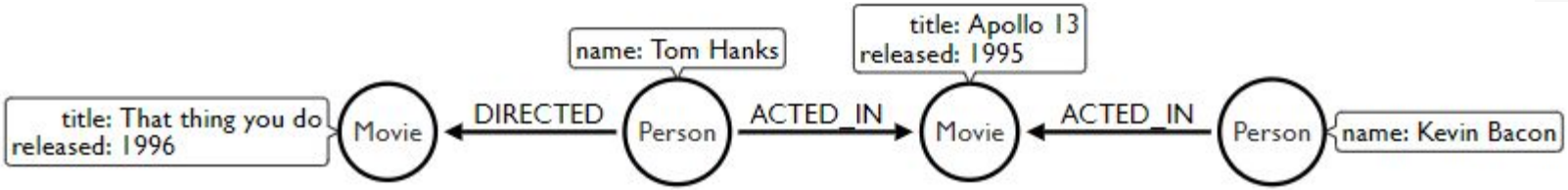
```
//Return nodes with label Person and name property is "Tom Hanks" - Inline
MATCH (p:Person {name: "Tom Hanks"}) //Only works with exact matches
RETURN p;
```

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

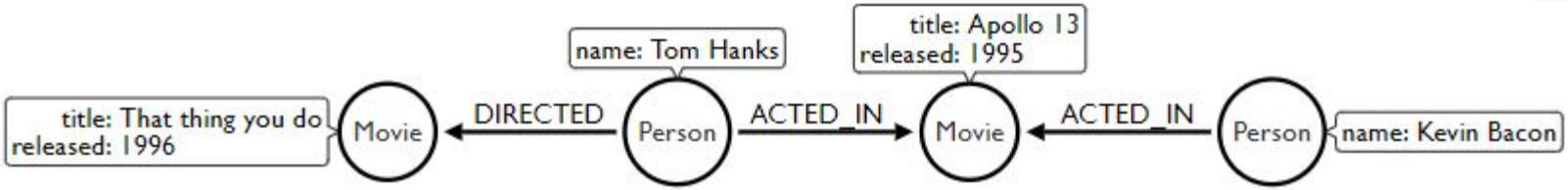
Use MATCH and properties to retrieve nodes

```
//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

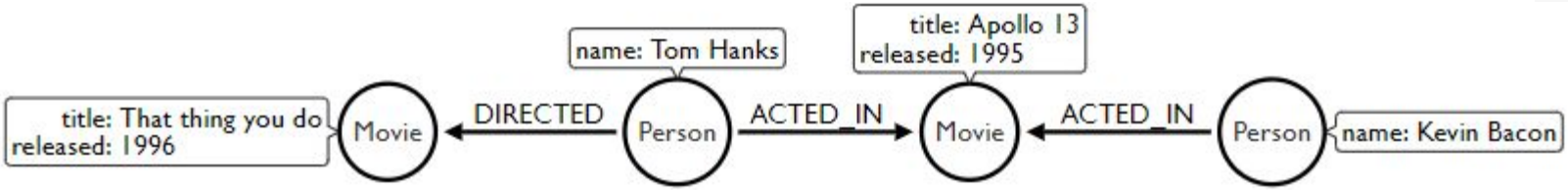


Extending the MATCH




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

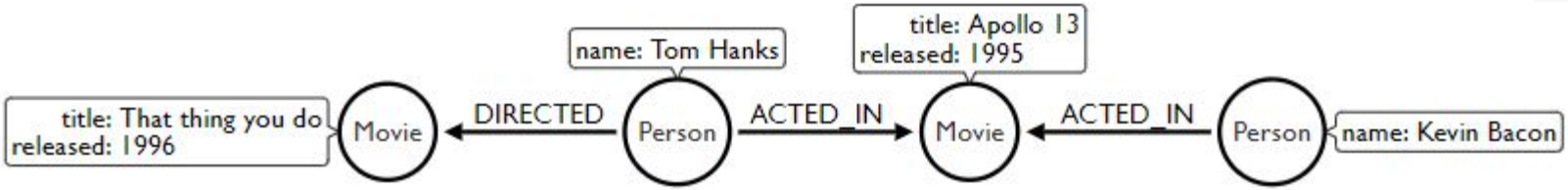
Extending the MATCH




```
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;
```

MERGE

```
//Uniquely create a person node called "Tom Hanks"
MERGE (p:Person {name:"Tom Hanks"});
```

MERGE

```
//Create a person node called "Tom Hanks"
MERGE (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"})
MERGE (p)-[:ACTED_IN]->(m);
```

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 variable, label/type and an inline property	`(p:Person {name: 'Bob'})`	`-[r:ACTED_IN {role: 'Rob'}]-`

Stackoverflow Demo

:play sandbox/stackoverflow

What Else is There?

What else is there?

Connectors

- Spark
- Kafka
- JDBC

Drivers

- Python
- JavaScript
- Java
- .Net
- Go

Data Science

- Graph Data Science Library
- Bloom Visualization

Libraries / Integrations

- neo4j/graphql
- Spring Data Neo4j
- neosemantics (RDF)
- APOC (utility)

So how do I continue my graph journey?

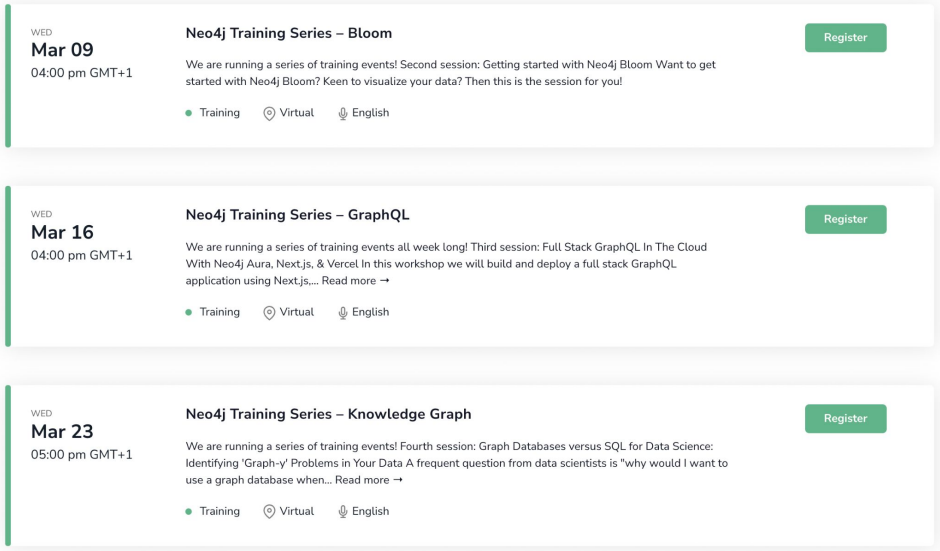
A training class each week - Tuesdays, 3pm UTC

09 Mar: Getting Started with Neo4j Bloom

16 Mar: Build APIs with Neo4j GraphQL Library

23 Mar: Create a Knowledge Graph: A Simple ML Approach

Read all about it!
dev.neo4j.com/training



Continue your journey

Free online training and certification:

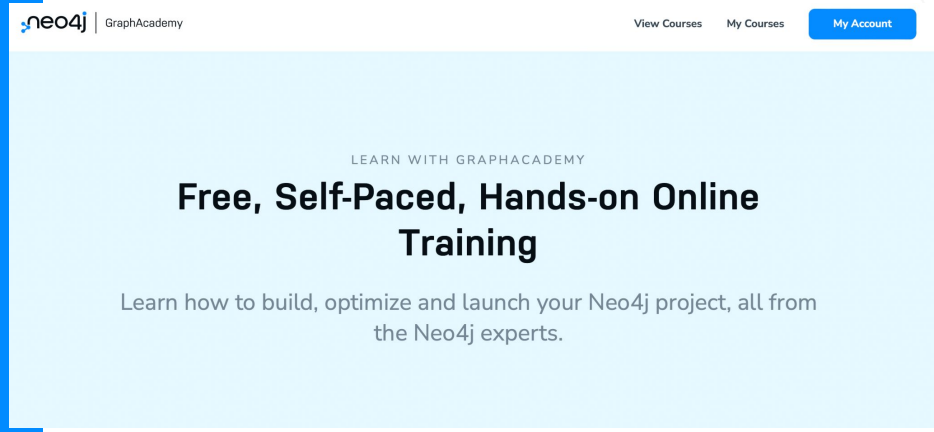
- *dev.neo4j.com/learn*
- *dev.neo4j.com/datasets*

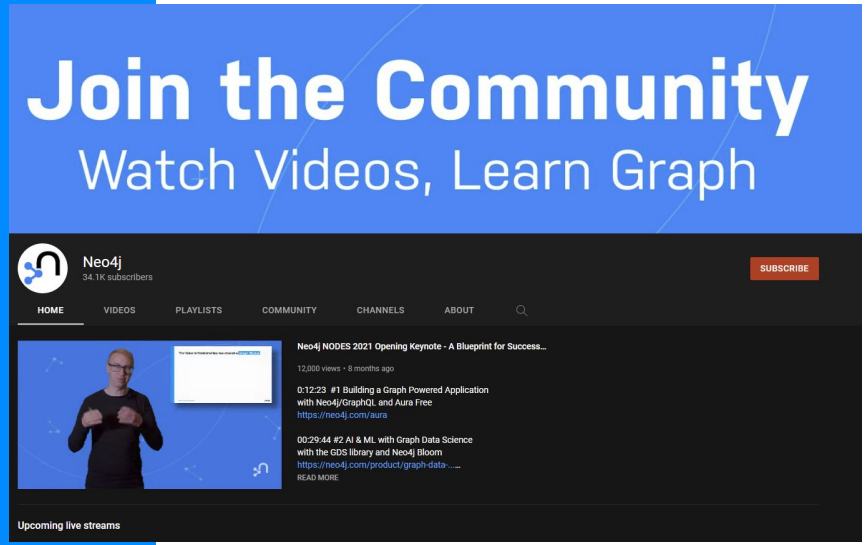
How to, best practices, hands on and community stories:

- *dev.neo4j.com/videos*

Come say hello :)

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





Michael Hunger

Developer Relations

michael@neo4j.com

@mesirii

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