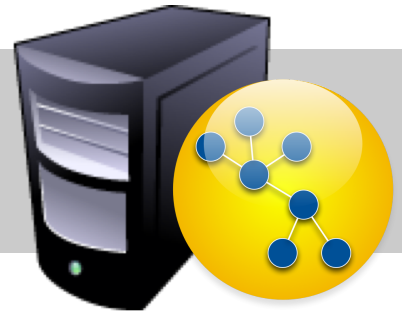


Let Me Graph That For You

@ianSrobinson
#neo4j



Outline

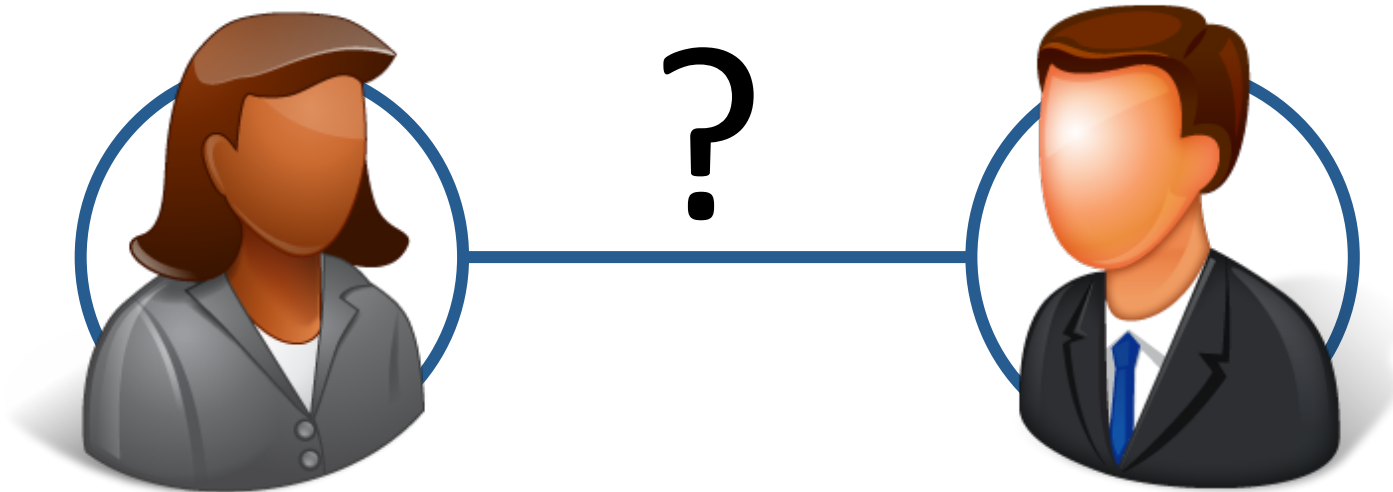


- Data complexity
- Graph data scenarios
- Using a graph database

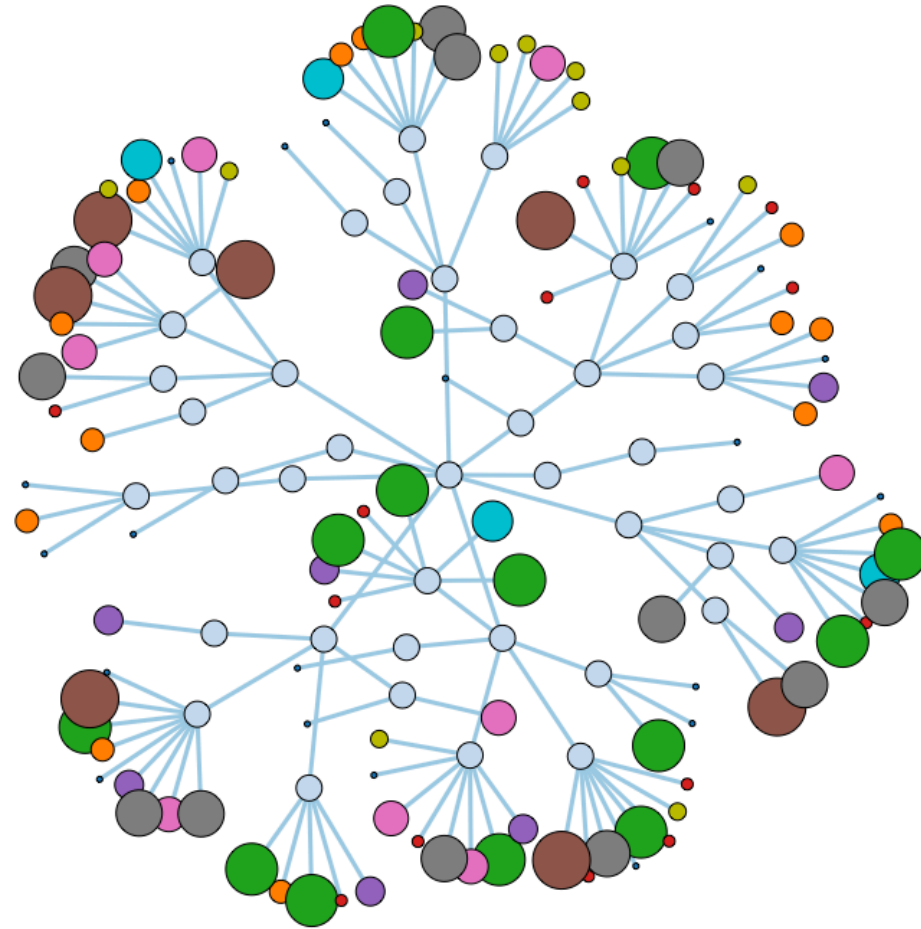
complexity = f(size, variable structure, connectedness)

Connectedness

- Existence
- Semantics
- Quality

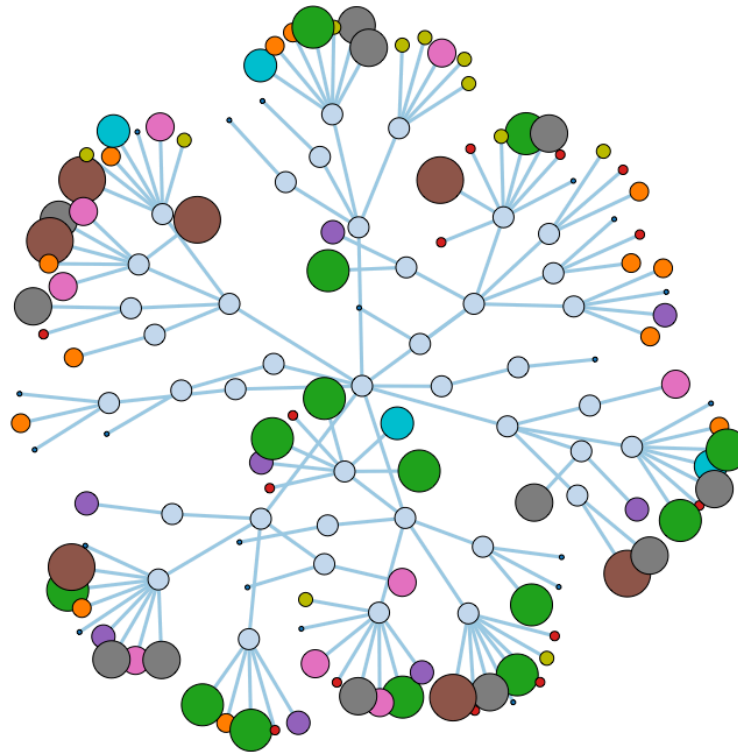


Graphs Are Everywhere

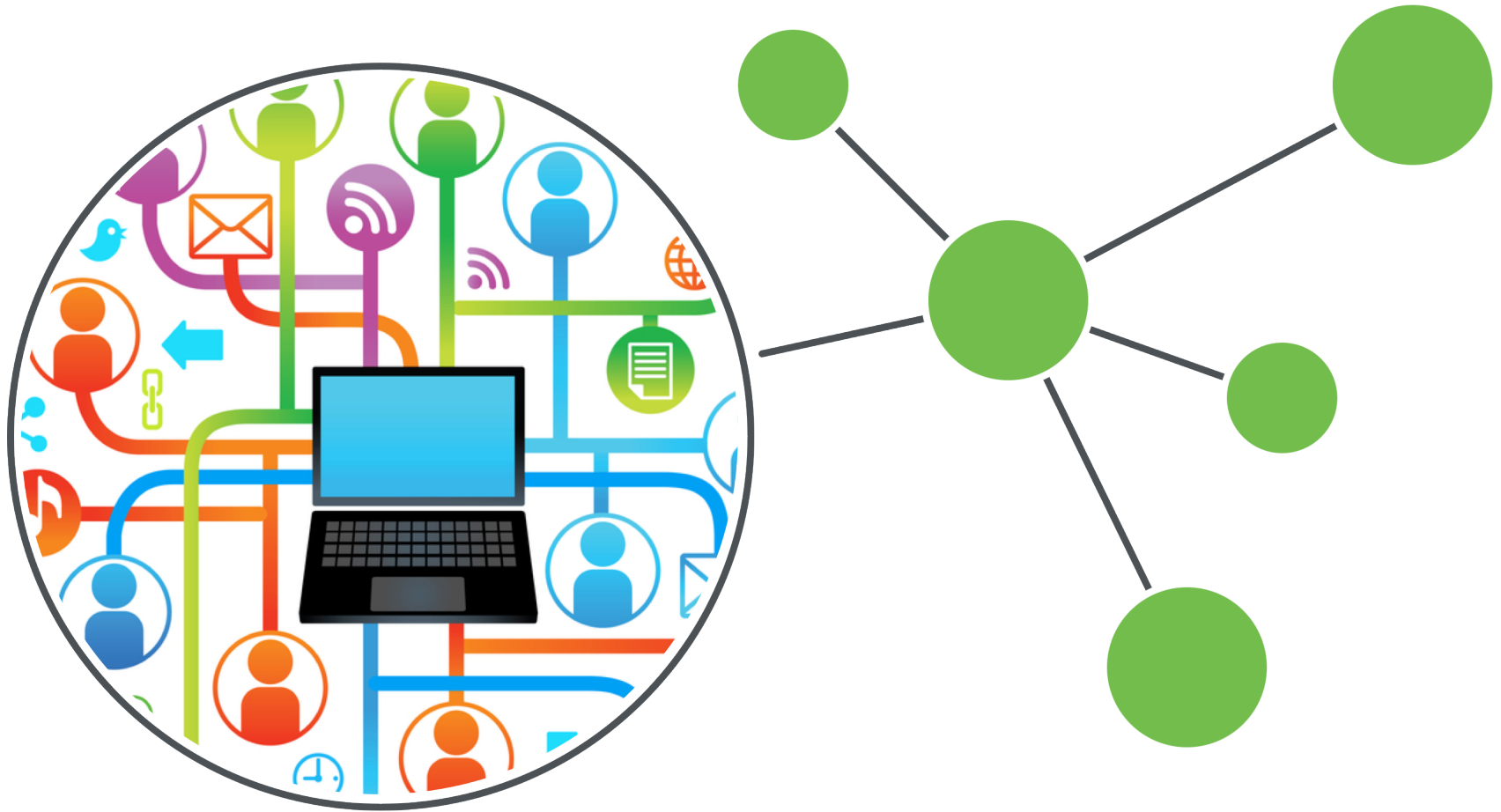


Graph Databases

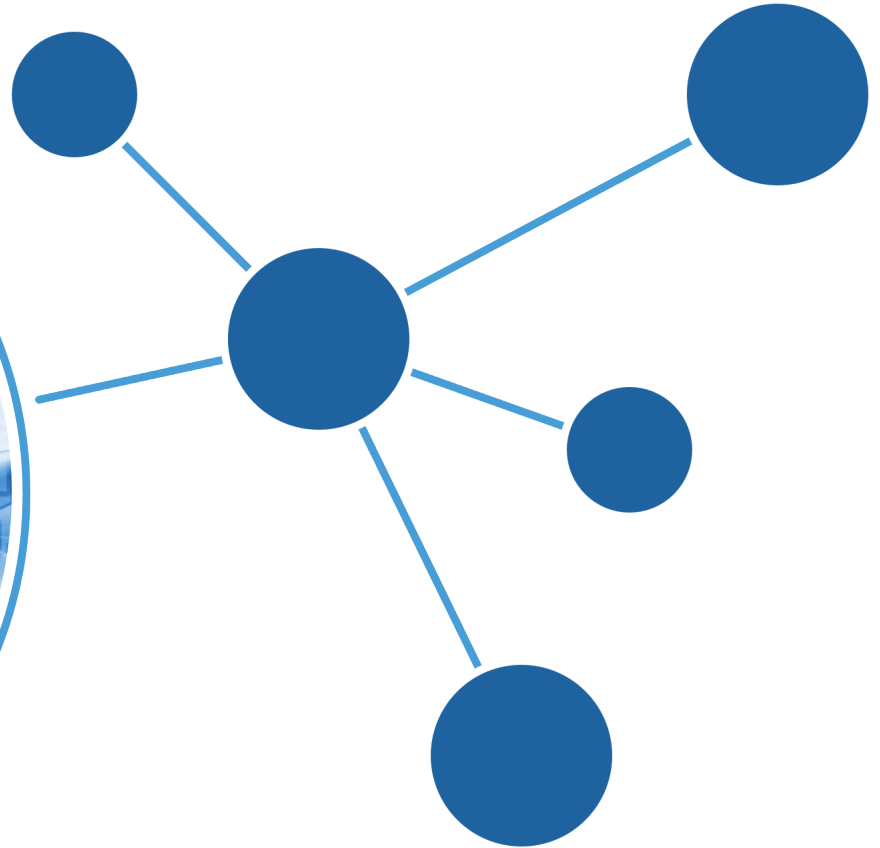
- Store
- Manage
- Query



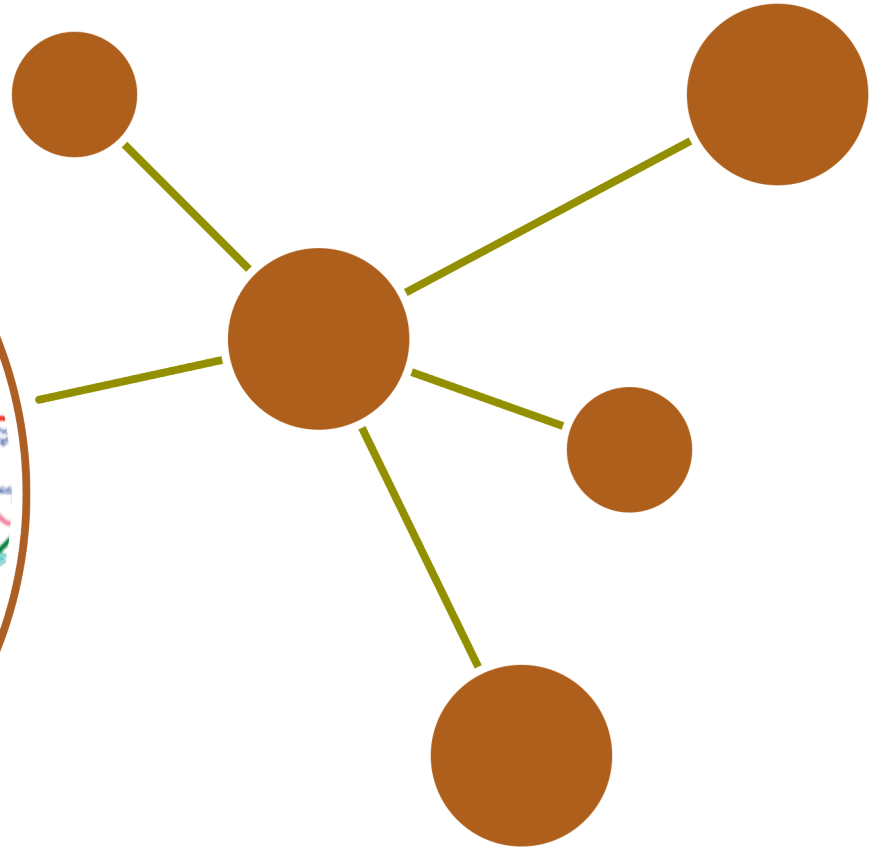
Social Network



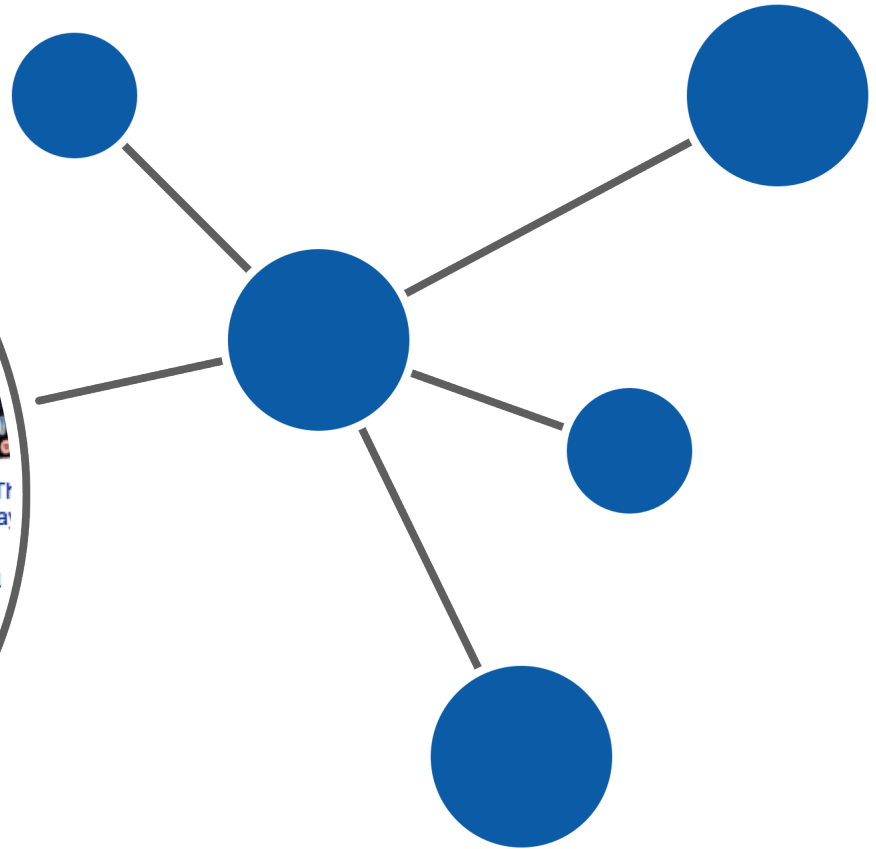
Network Impact Analysis



Route Finding



Recommendations



...ged the Kahler race, but a team of sc

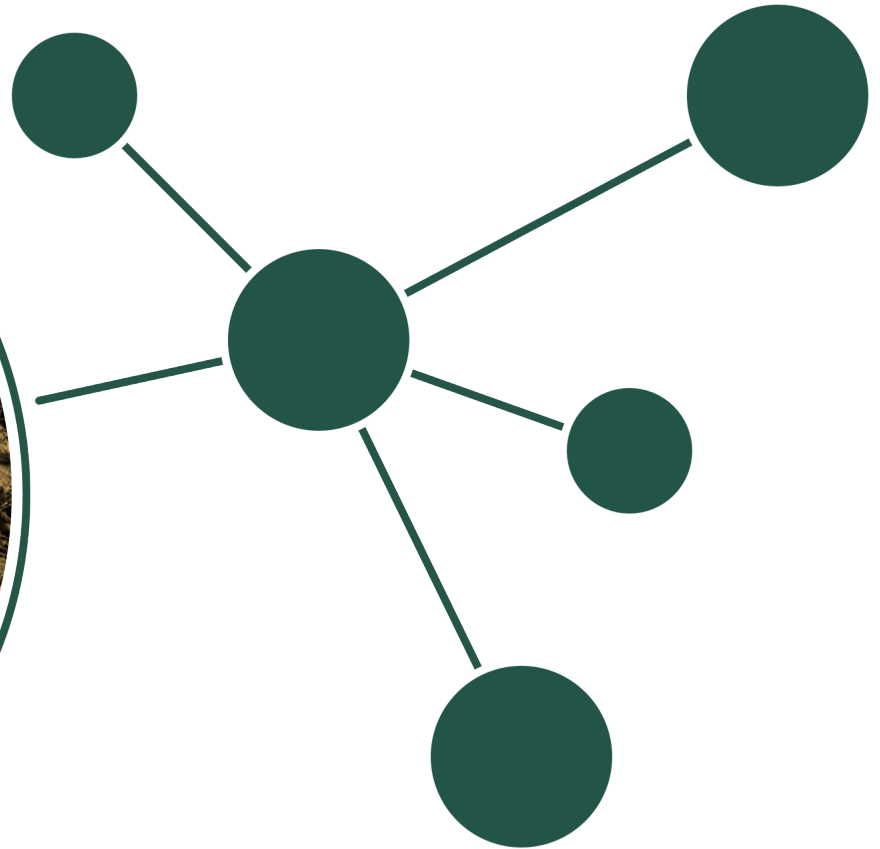
Customers Who Bought This Item Also Bought

		
Doctor Who - Series 7 Part 2 [Blu-ray]	Doctor Who Christmas Special 2011 - The ...	Doctor Who - The Complete Sixth Series [Blu-ray]
Matt Smith	Matt Smith	Matt Smith
★★★★★ (3)	★★★★☆ (81)	★★★★★ (1)
Blu-ray	Blu-ray	Blu-ray
£18.71	£12.99	£19.00

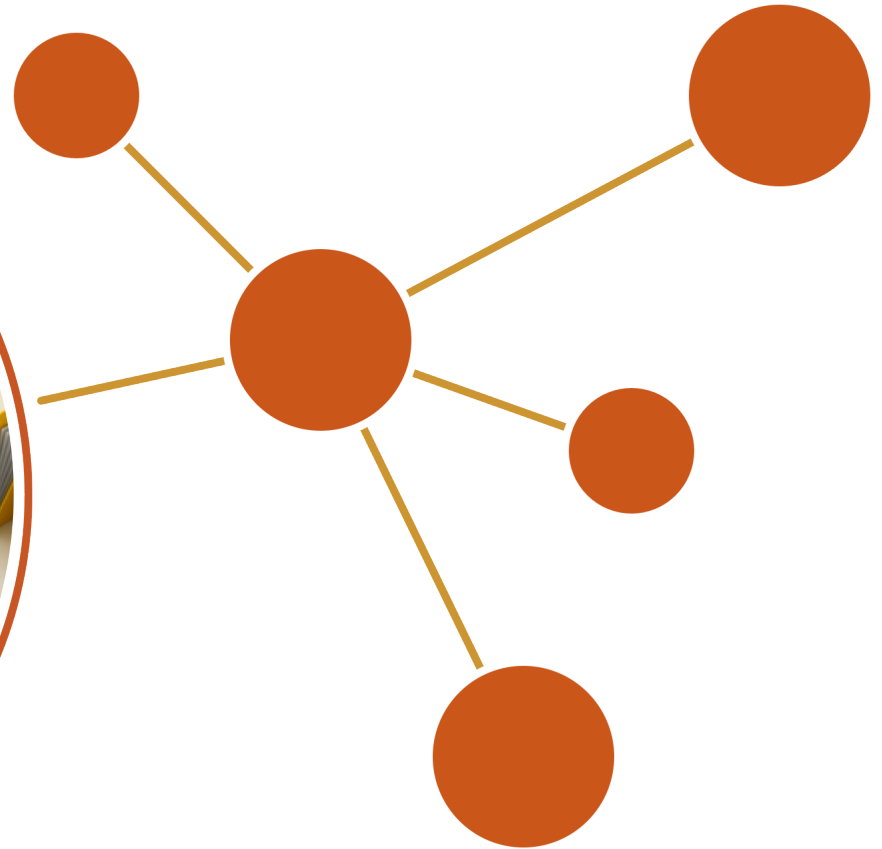
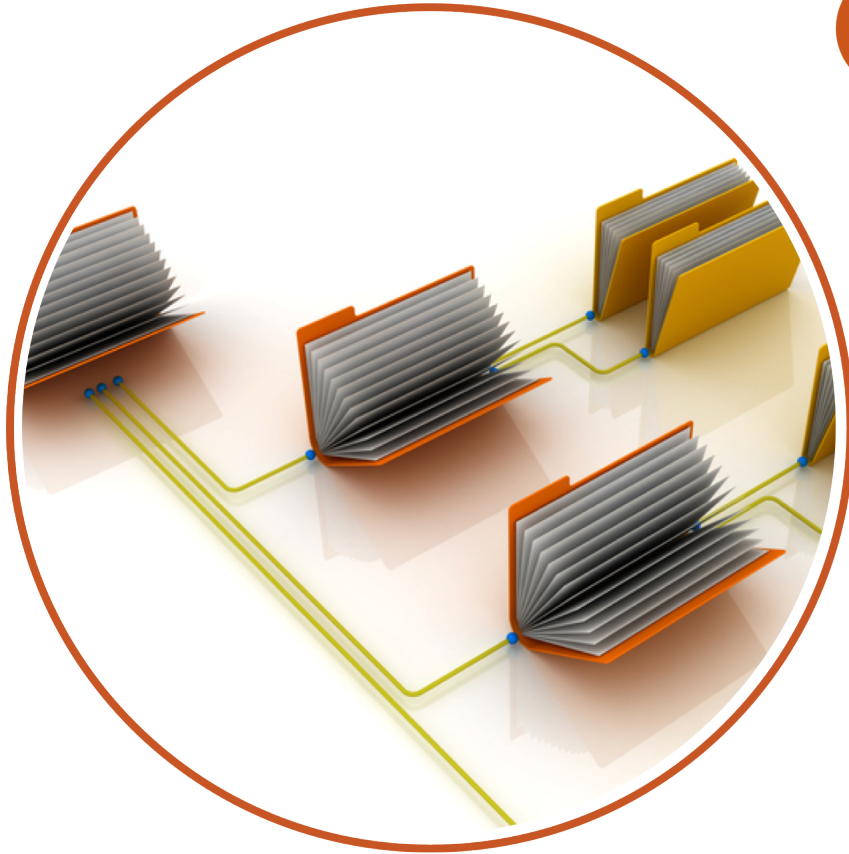
Reviews

	58		80 reviews
	11		4.5 out of 5 stars

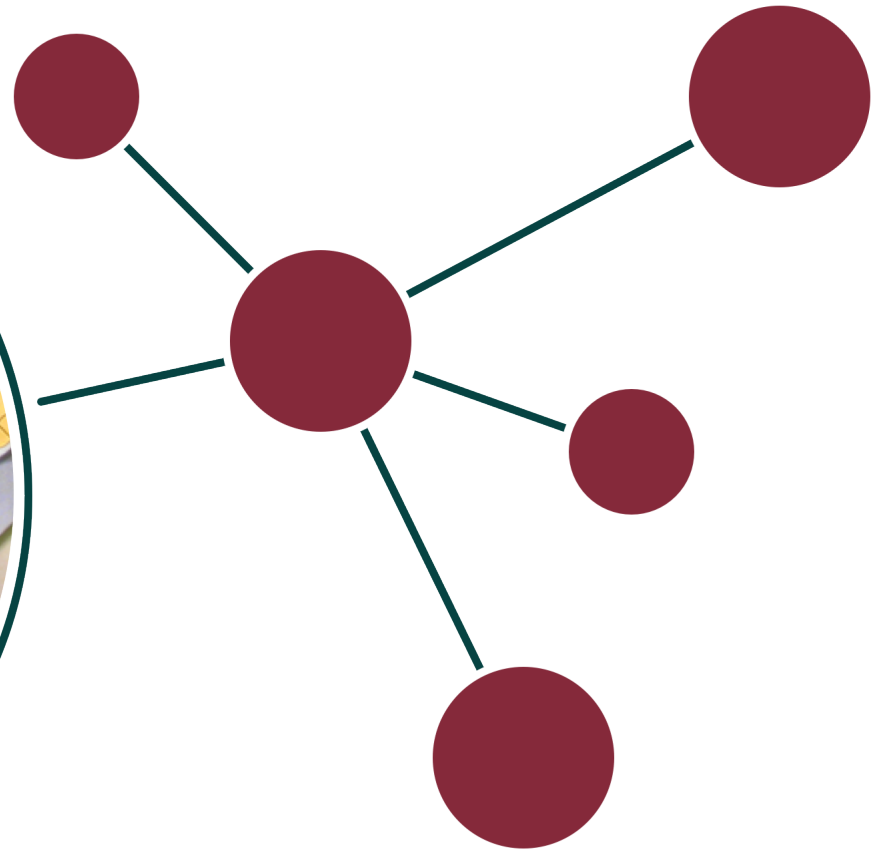
Logistics



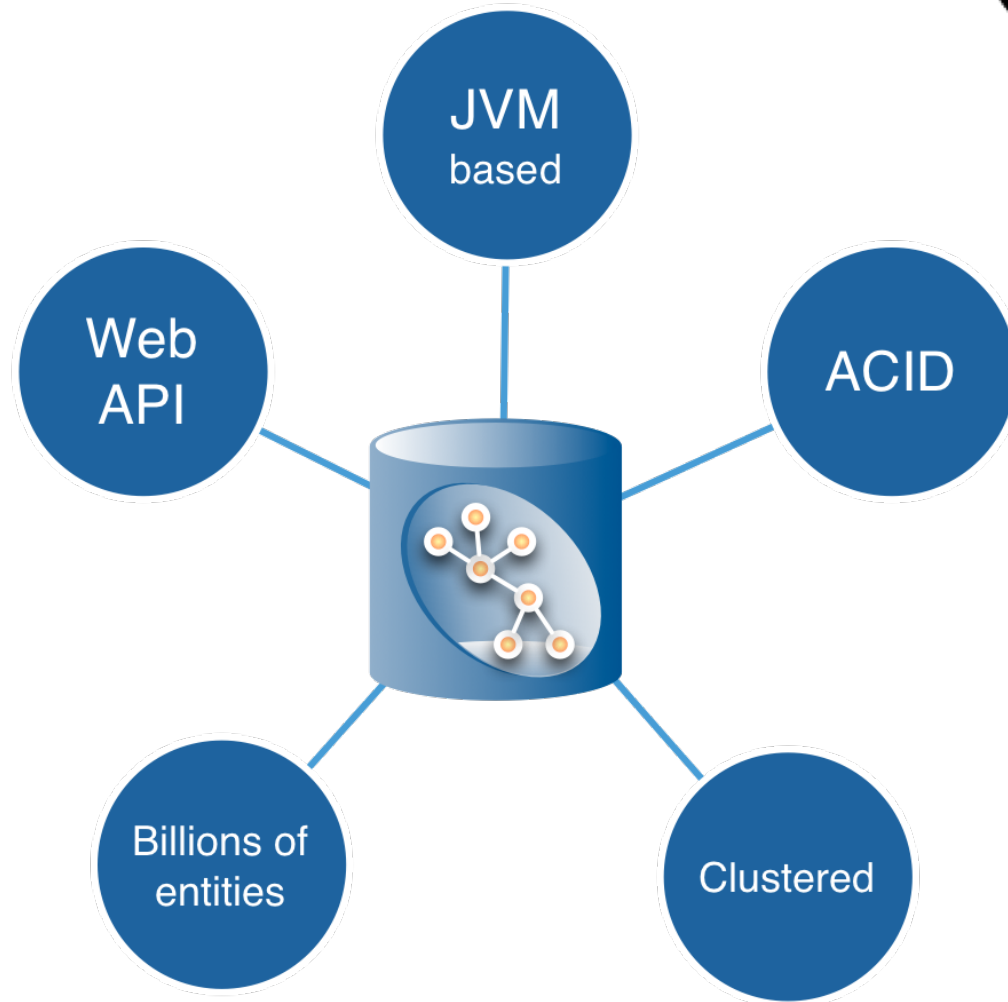
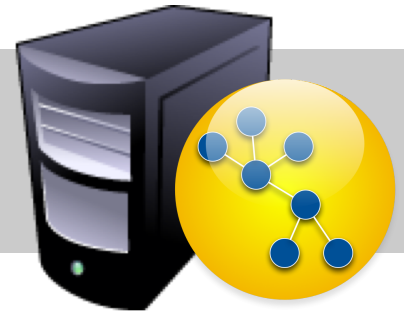
Access Control



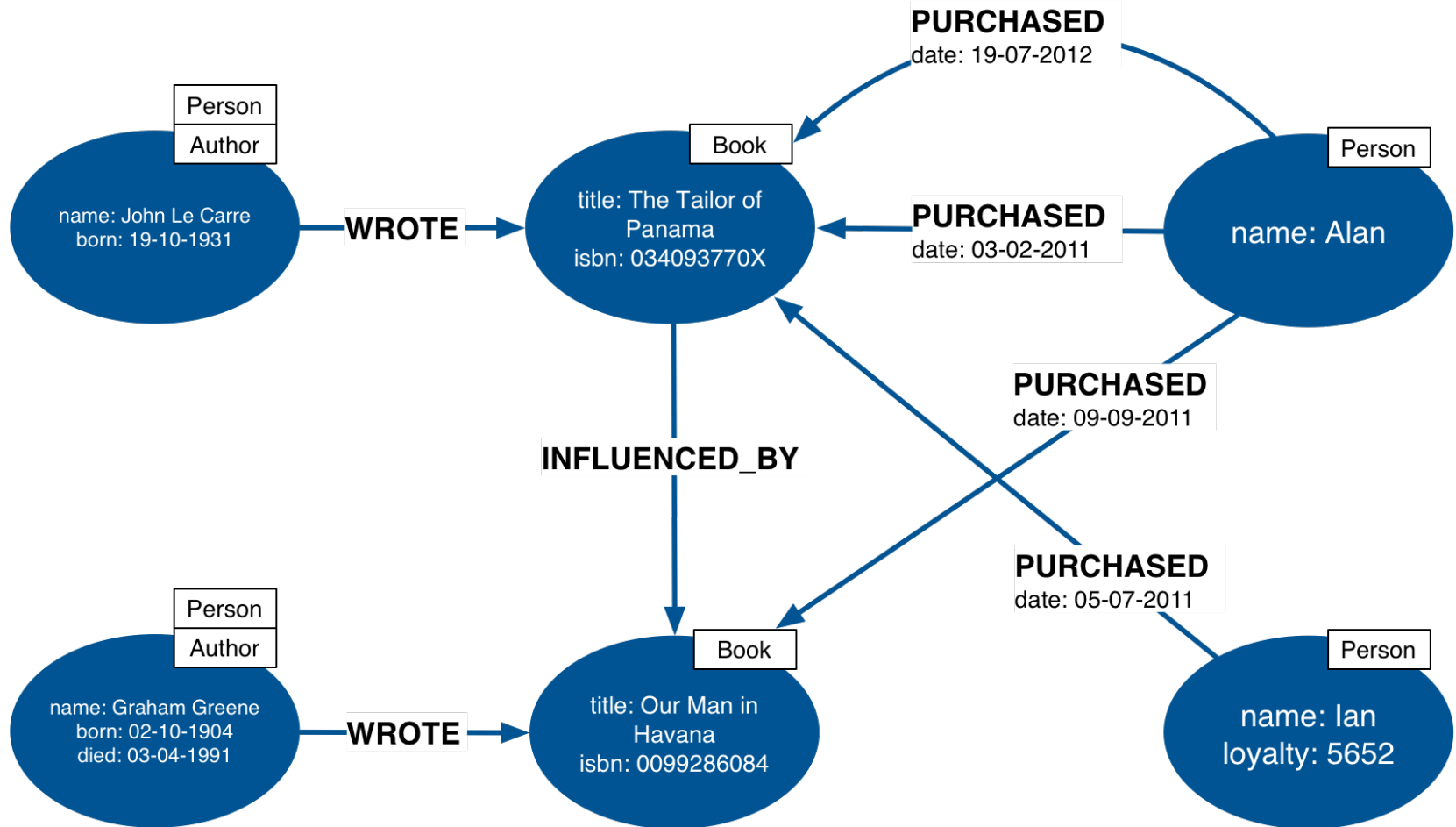
Fraud Analysis



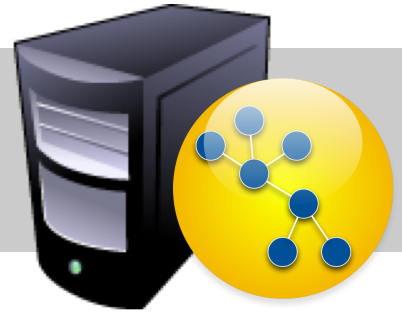
Neo4j



Labeled Property Graph

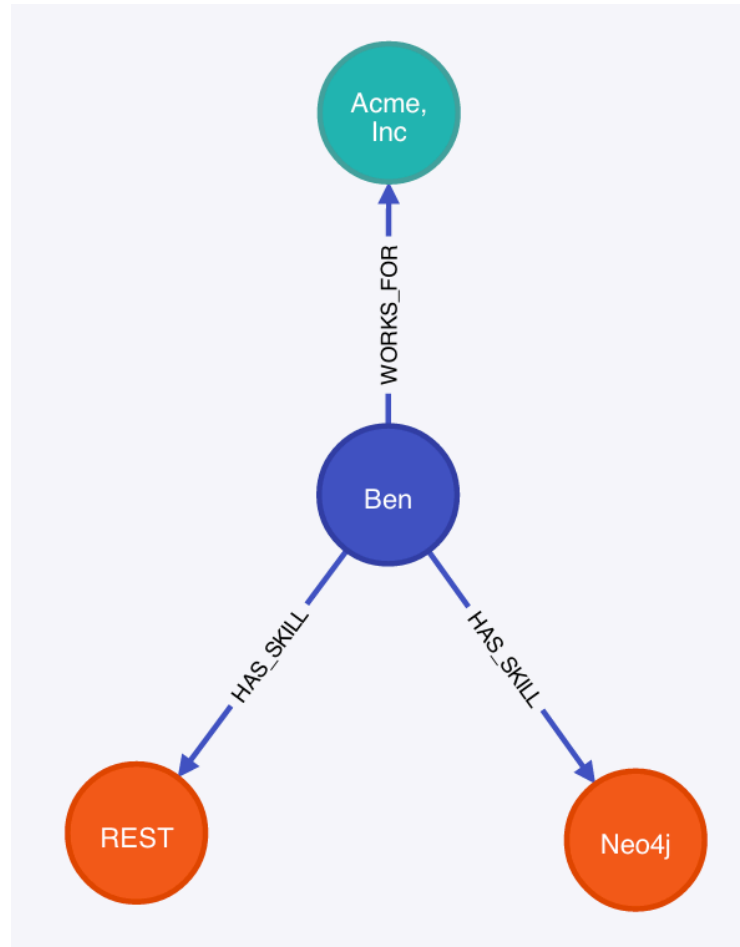


Querying Graph Data

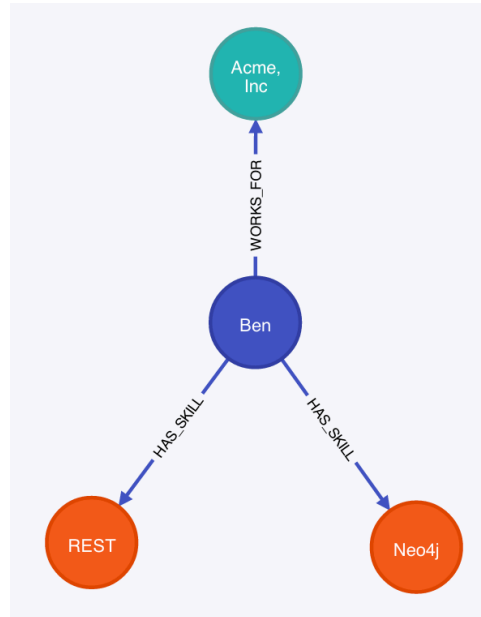


- Describing graphs
- Creating nodes, relationships and properties
- Querying graphs

How to Describe a Graph?



Cypher Pattern



`(ben)-[:WORKS_FOR]->(acme),`

`(ben)-[:HAS_SKILL]->(rest),`

`(ben)-[:HAS_SKILL]->(neo4j)`

Create Some Data

```
CREATE (ben:Person { name:'Ben' }),
       (acme:Company { name:'Acme' }),
       (rest:Skill { name:'REST' }),
       (neo4j:Skill { name:'Neo4j' }),
       (ben)-[:WORKS_FOR]->(acme),
       (ben)-[:HAS_SKILL]->(rest),
       (ben)-[:HAS_SKILL]->(graphs)
RETURN ben
```

Create Nodes

```
CREATE (ben:Person { name:'Ben' }),
       (acme:Company { name:'Acme' }),
       (rest:Skill { name:'REST' }),
       (neo4j:Skill { name:'Neo4j' }),
       (ben)-[:WORKS_FOR]->(acme),
       (ben)-[:HAS_SKILL]->(rest),
       (ben)-[:HAS_SKILL]->(graphs)
RETURN ben
```

Create Relationships

```
CREATE (ben:Person { name:'Ben' }),
      (acme:Company { name:'Acme' }),
      (rest:Skill { name:'REST' }),
      (neo4j:Skill { name:'Neo4j' }),
      (ben)-[:WORKS_FOR]->(acme),
      (ben)-[:HAS_SKILL]->(rest),
      (ben)-[:HAS_SKILL]->(graphs)
RETURN ben
```

Return Node

```
CREATE (ben:Person { name:'Ben' }),
       (acme:Company { name:'Acme' }),
       (rest:Skill { name:'REST' }),
       (neo4j:Skill { name:'Neo4j' }),
       (ben)-[:WORKS_FOR]->(acme),
       (ben)-[:HAS_SKILL]->(rest),
       (ben)-[:HAS_SKILL]->(graphs)

RETURN ben
```

Neo4j /Users/iansrobinson x
localhost:7474/browser/#/stream

CYPHER match n return n

- Person
- Skill
- Company

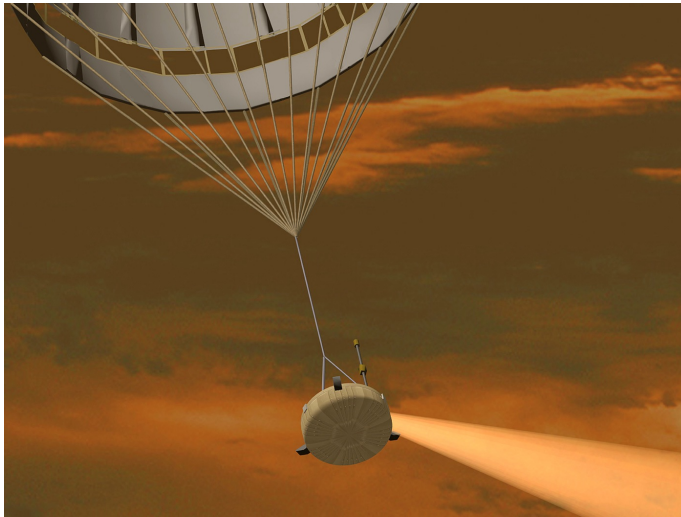
```
graph TD; Charlie((Charlie)) -- WORKS_FOR --> Acme((Acme, Inc)); Ben((Ben)) -- WORKS_FOR --> Acme; Ian((Ian)) -- WORKS_FOR --> Acme; Lucy((Lucy)) -- WORKS_FOR --> Acme; Arnold((Arnold)) -- WORKS_FOR --> Startup((Startup, Ltd)); Emily((Emily)) -- WORKS_FOR --> Startup; Gordon((Gordon)) -- WORKS_FOR --> Startup; Kate((Kate)) -- WORKS_FOR --> Startup; Charlie -- HAS_SKILL --> SQL((SQL)); Charlie -- HAS_SKILL --> Javascript((Javascript)); Ben -- HAS_SKILL --> REST((REST)); Ben -- HAS_SKILL --> Neo4j((Neo4j)); Ian -- HAS_SKILL --> REST; Ian -- HAS_SKILL --> Neo4j; Lucy -- HAS_SKILL --> Neo4j; Lucy -- HAS_SKILL --> Java((Java)); Arnold -- HAS_SKILL --> Clojure((Clojure)); Arnold -- HAS_SKILL --> Ruby((Ruby)); Arnold -- HAS_SKILL --> DotNet((DotNet)); Emily -- HAS_SKILL --> Ruby; Emily -- HAS_SKILL --> Python((Python)); Gordon -- HAS_SKILL --> Python; Gordon -- HAS_SKILL --> Architecture((Architecture)); Kate -- HAS_SKILL --> Python; Kate -- HAS_SKILL --> Architecture;
```

Returned 20 nodes in 157 ms

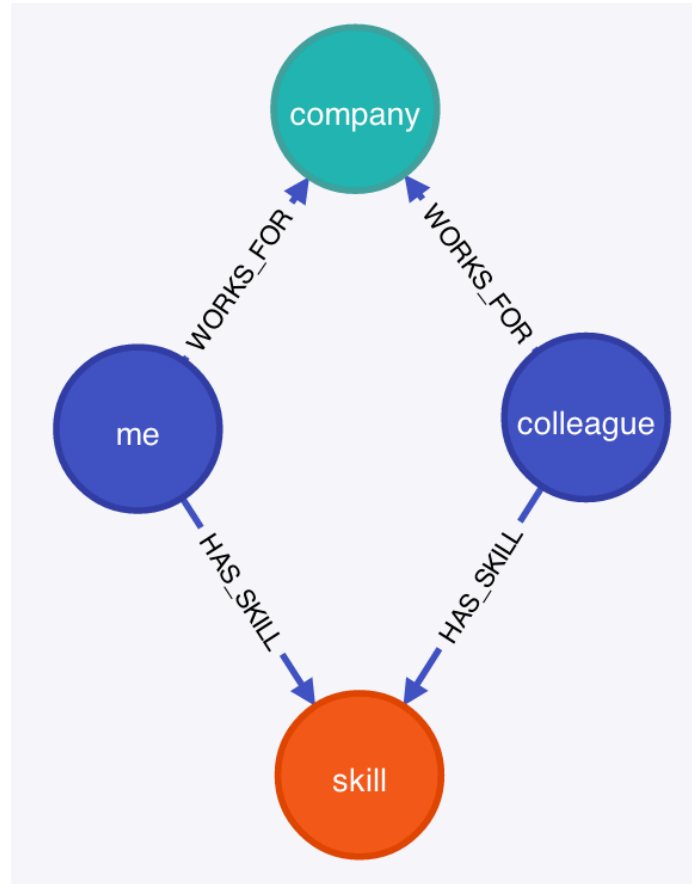
Querying a Graph

Graph Local

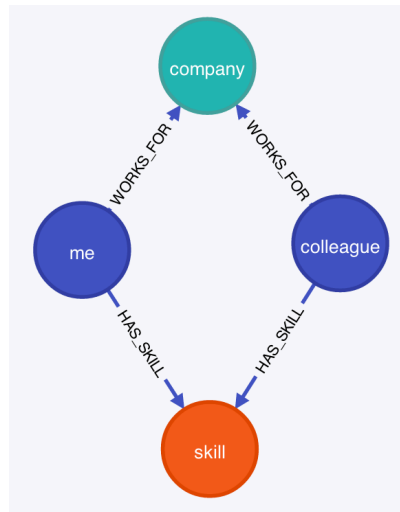
- Find one or more start nodes
- Explore surrounding graph
- Millions of hops per second



Which people, who work for the same company as me, share my skills?



Cypher Pattern

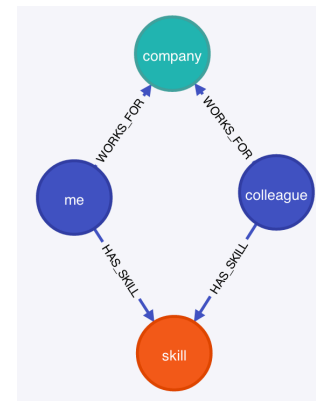


```
(company)<-[:WORKS_FOR]-(me)-[:HAS_SKILL]->(skill),  
(company)<-[:WORKS_FOR]-(colleague)-[:HAS_SKILL]->(skill)
```

Cypher Query

Which people, who work for the same company as me, have similar skills to me?

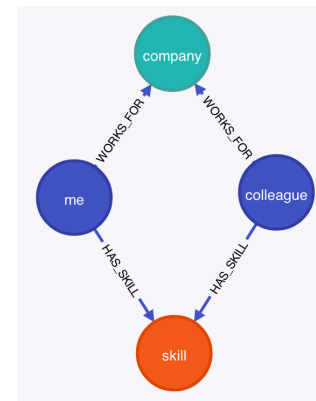
```
MATCH (company)<-[:WORKS_FOR]-(:Person{name:'ian'})
      -[:HAS_SKILL]->(skill),
      (company)<-[:WORKS_FOR]-(colleague)-[:HAS_SKILL]->(skill)
RETURN colleague.name AS name,
       count(skill) AS score,
       collect(skill.name) AS skills
ORDER BY score DESC
```



Graph Pattern

Which people, who work for the same company as me, have similar skills to me?

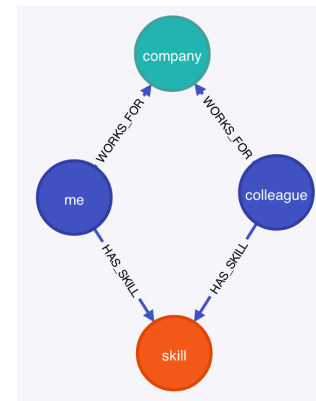
```
MATCH (company) <-[:WORKS_FOR]-(Person{name:'ian'})
      -[:HAS_SKILL]->(skill),
      (company) <-[:WORKS_FOR]-(colleague)-[:HAS_SKILL]->(skill)
RETURN colleague.name AS name,
       count(skill) AS score,
       collect(skill.name) AS skills
ORDER BY score DESC
```



Anchor Pattern in Graph

Which people, who work for the same company as me, have similar skills to me?

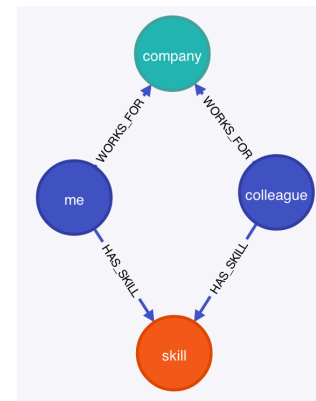
```
MATCH (company)<-[:WORKS_FOR]-(:Person{name:'ian'})
      -[:HAS_SKILL]->(skill),
      (company)<-[:WORKS_FOR]-(colleague)-[:HAS_SKILL]->(skill)
RETURN colleague.name AS name,
       count(skill) AS score,
       collect(skill.name) AS skills
ORDER BY score DESC
```

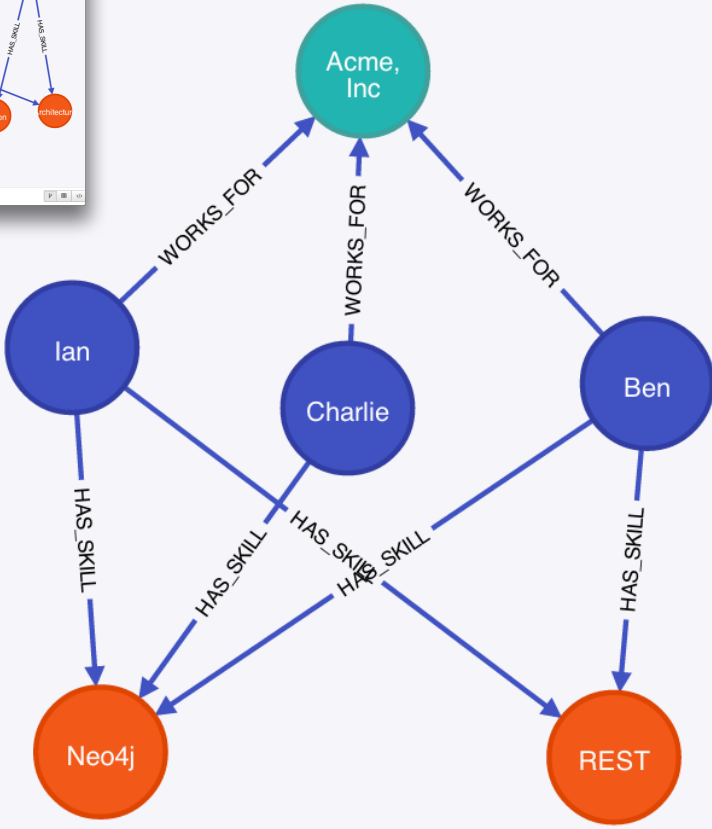
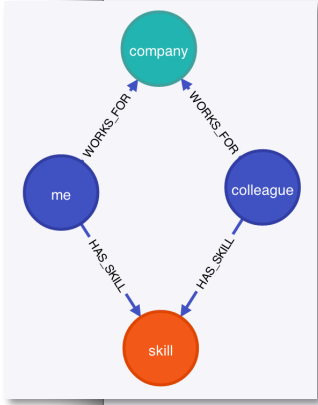
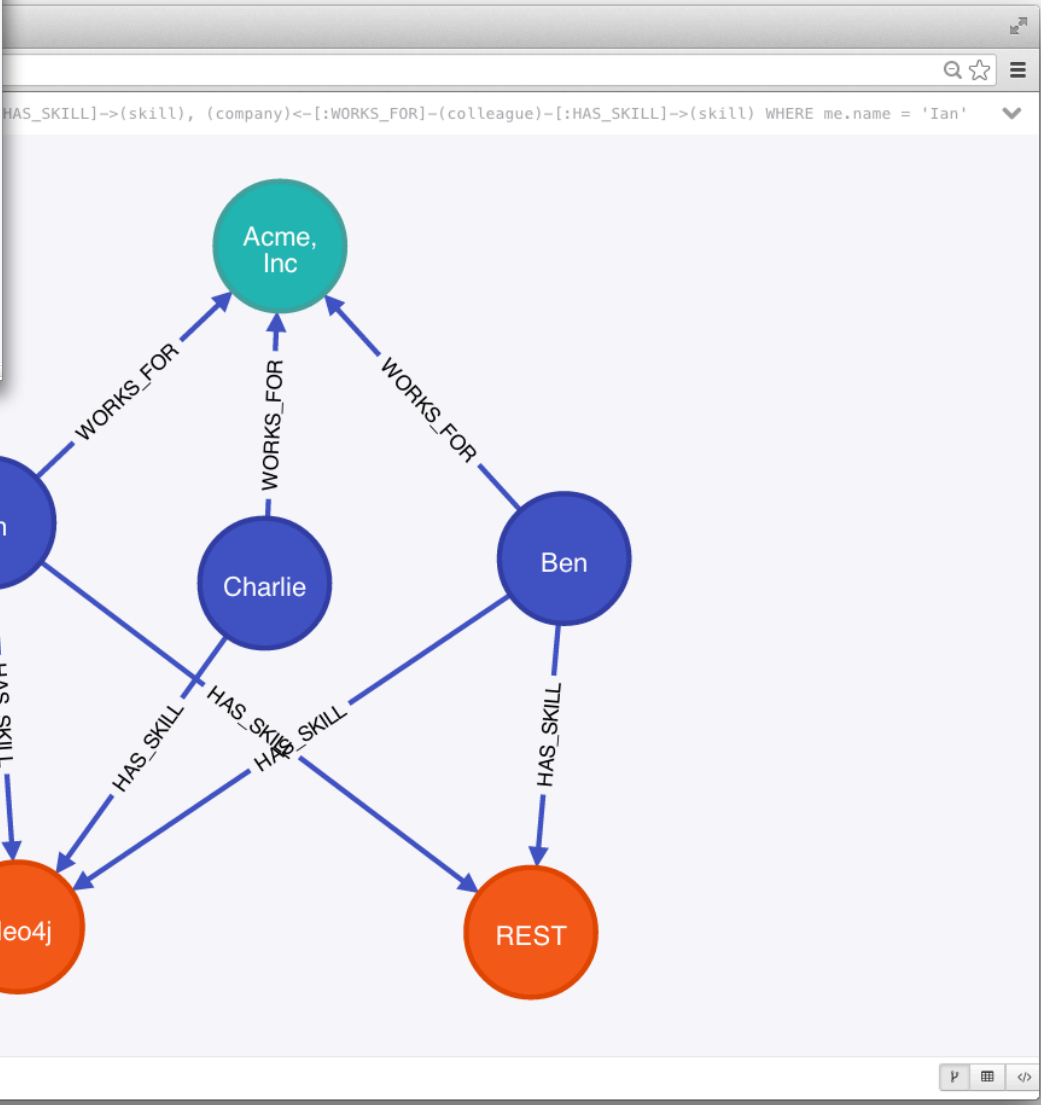
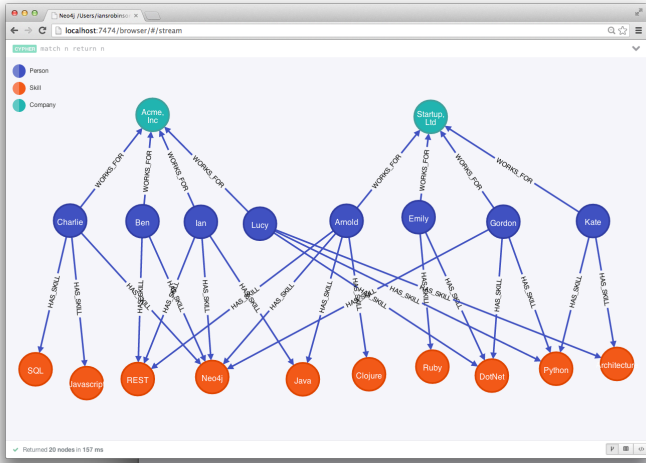


Create Results

Which people, who work for the same company as me, have similar skills to me?

```
MATCH (company)<-[:WORKS_FOR]-(:Person{name:'ian'})
      -[:HAS_SKILL]->(skill),
      (company)<-[:WORKS_FOR]-(:colleague)-[:HAS_SKILL]->(skill)
RETURN colleague.name AS name,
       count(skill) AS score,
       collect(skill.name) AS skills
ORDER BY score DESC
```





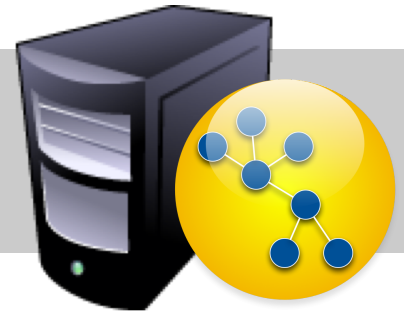
Returned 12 nodes in 128 ms

Results

```
+-----+
| name      | score | skills          |
+-----+
| "Ben"     | 2     | ["Neo4j", "REST"] |
| "Charlie" | 1     | ["Neo4j"]         |
+-----+
2 rows
```

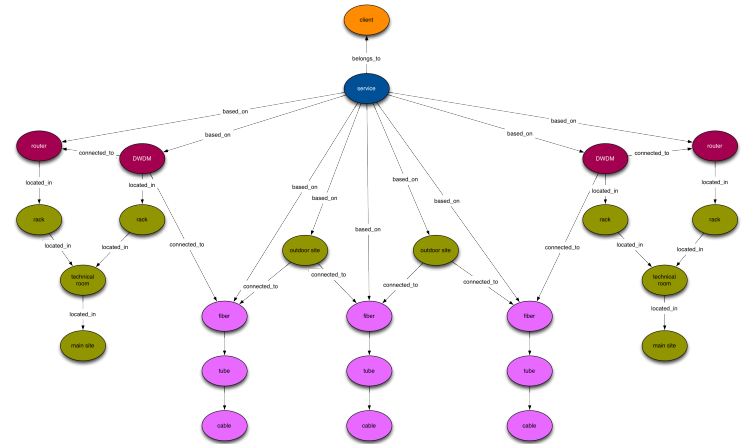
name	score	skills
"Ben"	2	["Neo4j", "REST"]
"Charlie"	1	["Neo4j"]

Case Studies



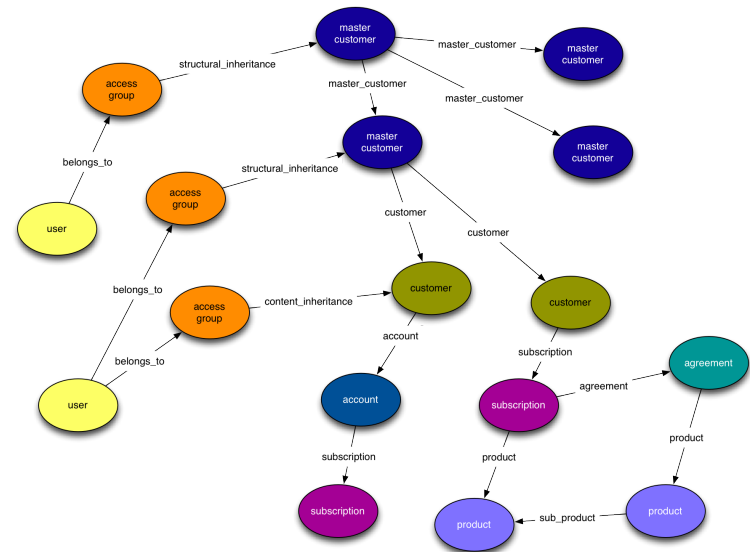
Network Impact Analysis

- Which parts of network does a customer depend on?
- Who will be affected if we replace a network element?



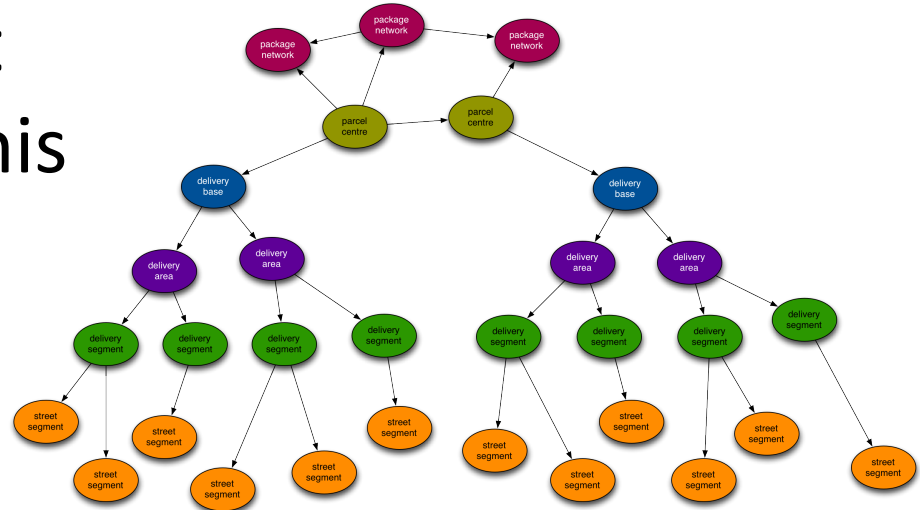
Asset Management & Access Control

- Which assets can an admin control?
- Who can change my subscription?



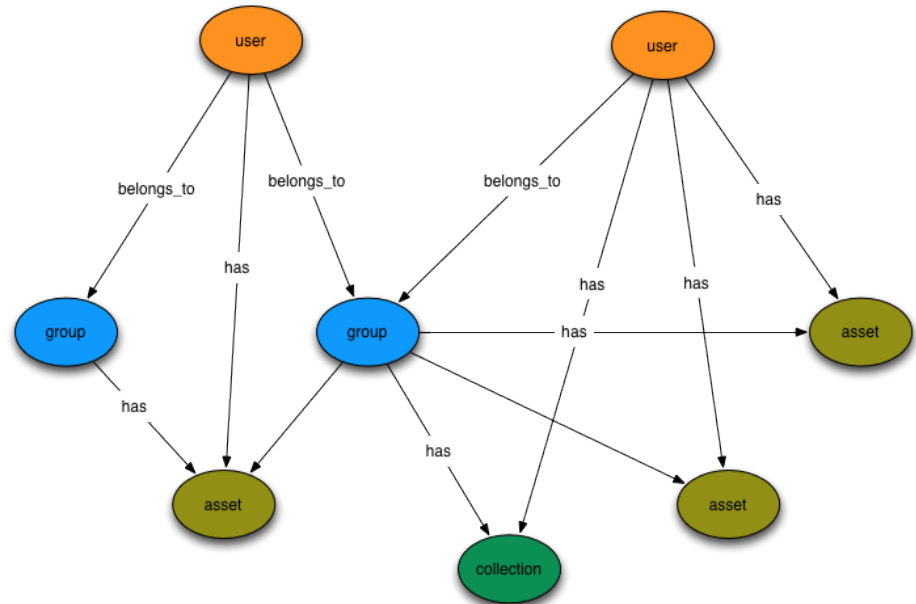
Logistics

- What's the quickest delivery route for this parcel?



Social Network & Recommendations

- Which assets can I access?
- Who shares my interests?



neo4j.com/online_course

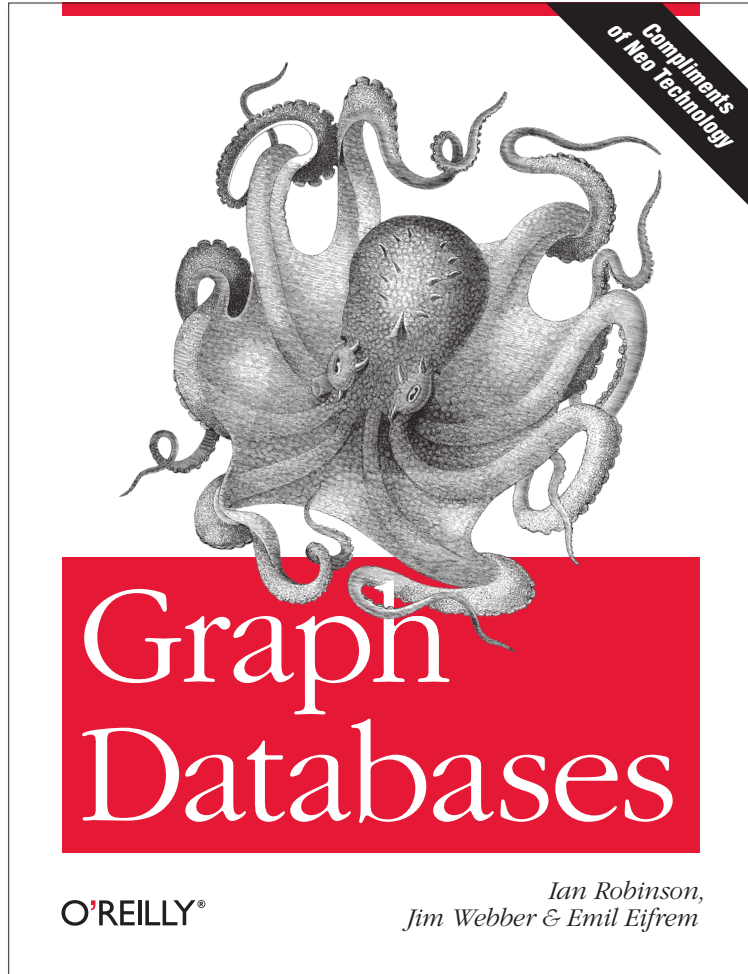
Online Training: Getting Started with Neo4j

Learn Neo4j at your own pace and time with our free online training course. Get introduced to graph databases, learn the core functionality of Neo4j, and practice Cypher with this engaging and interactive course.



[Get started today »](#)

graphdatabases.com



Thank you

@ianSrobinson
#neo4j