



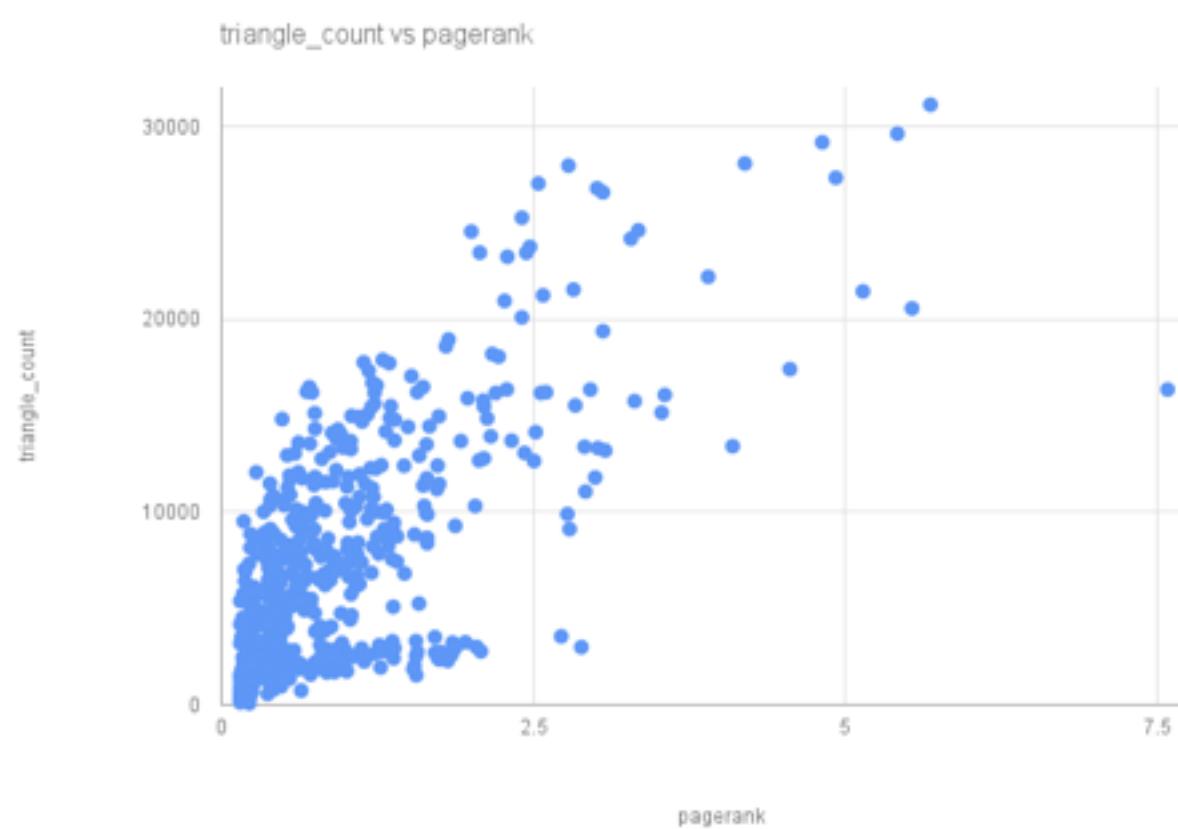
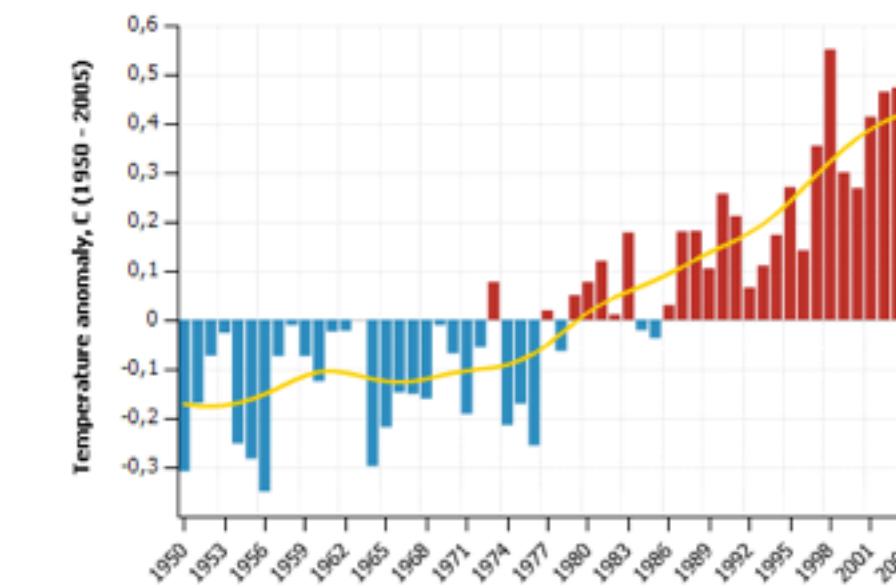
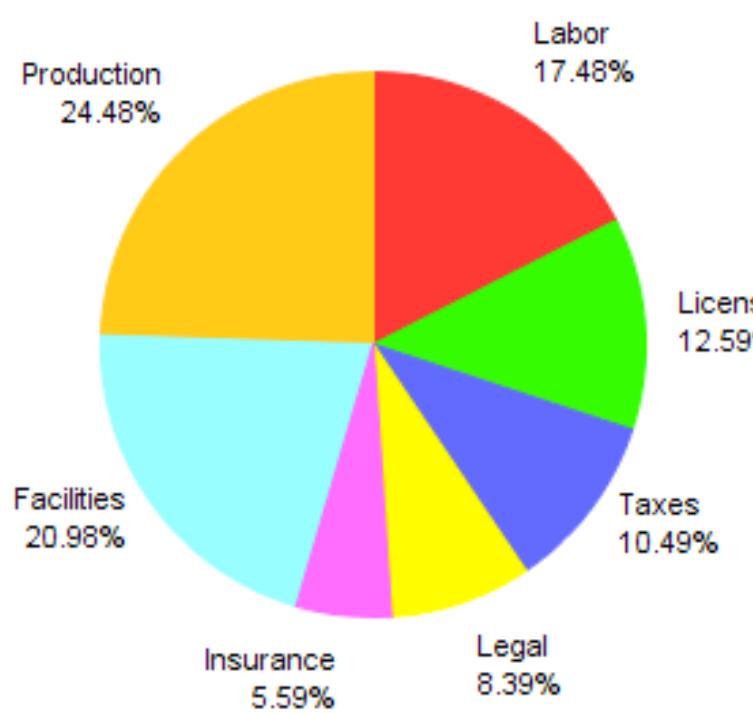
# neo4j

an Introduction to Graph Databases

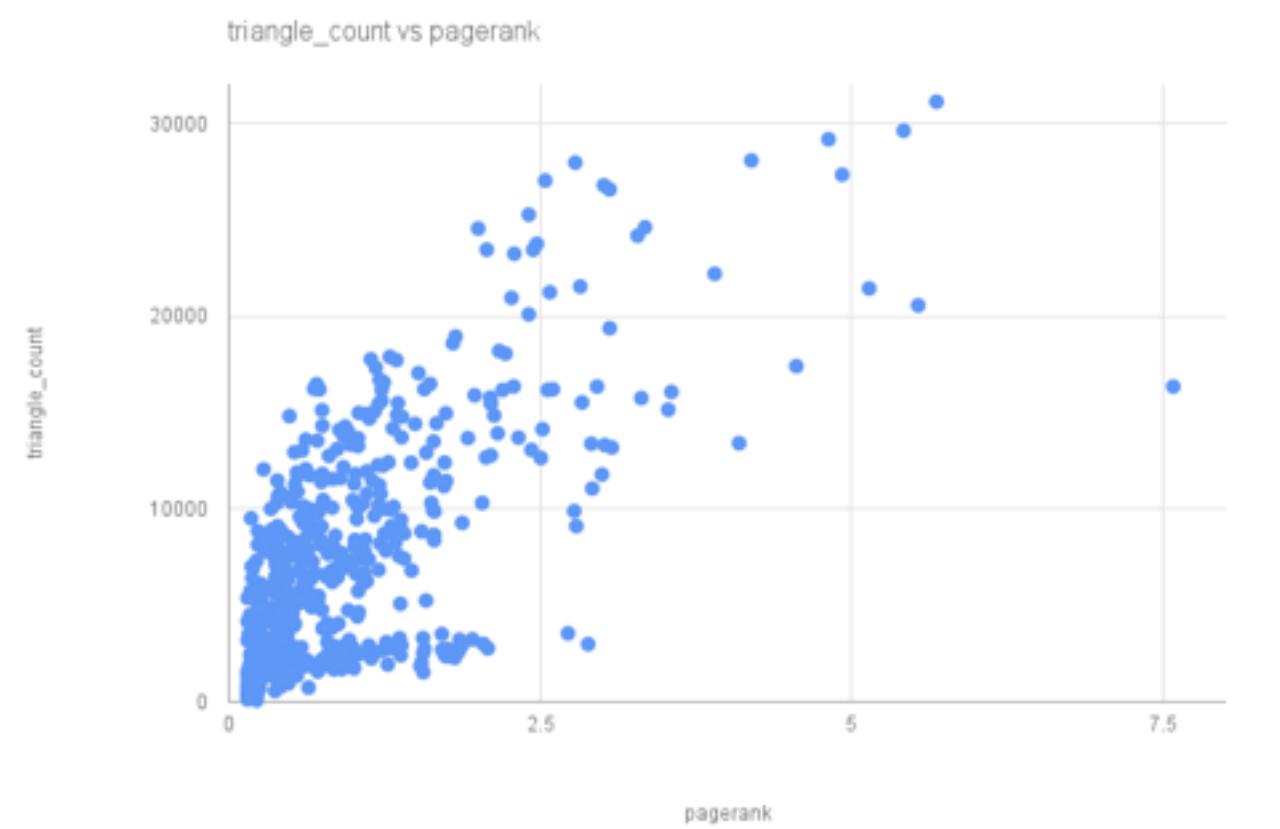
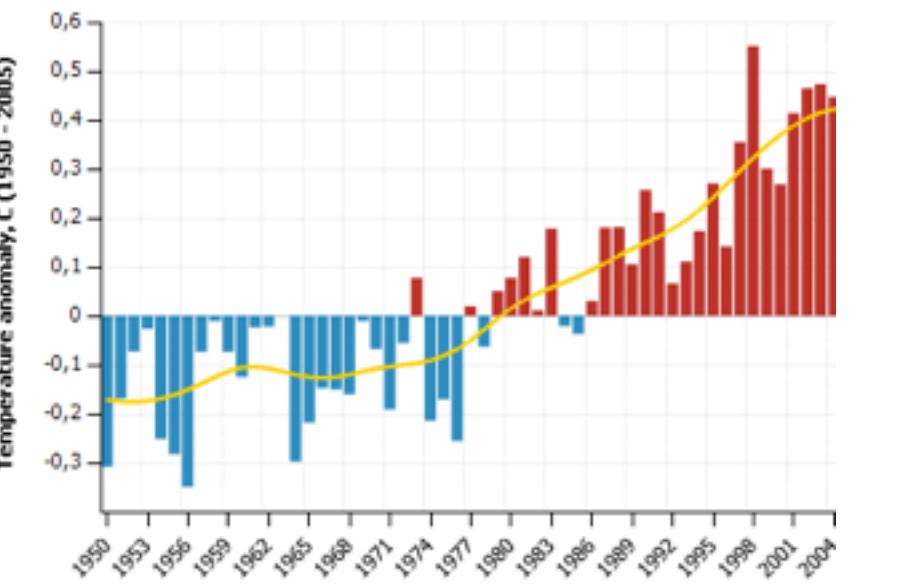
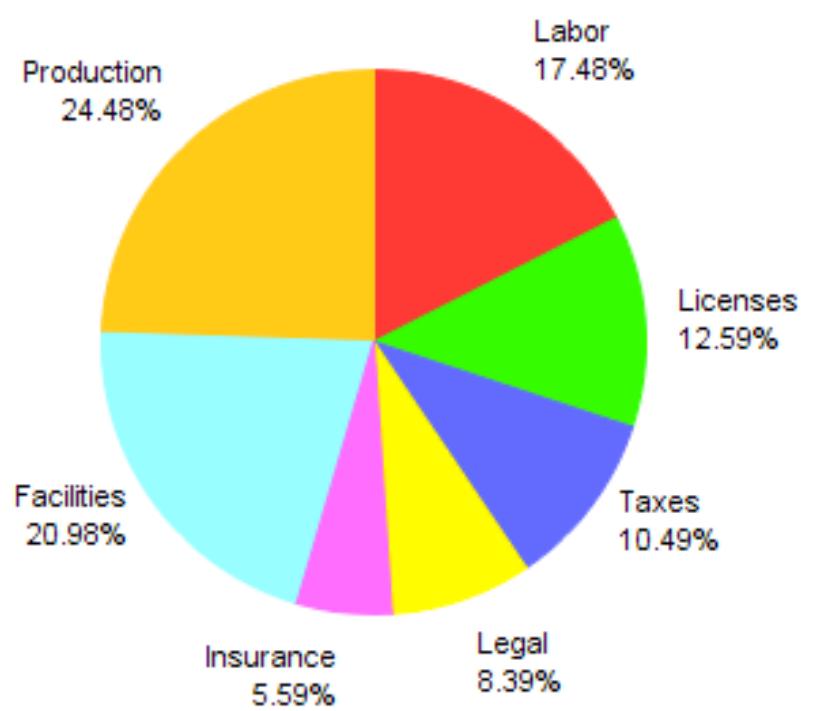
“Graph analysis is possibly the single most effective competitive differentiator for organizations pursuing data-driven operations and decisions after the design of data capture.”

**Gartner**<sup>®</sup>

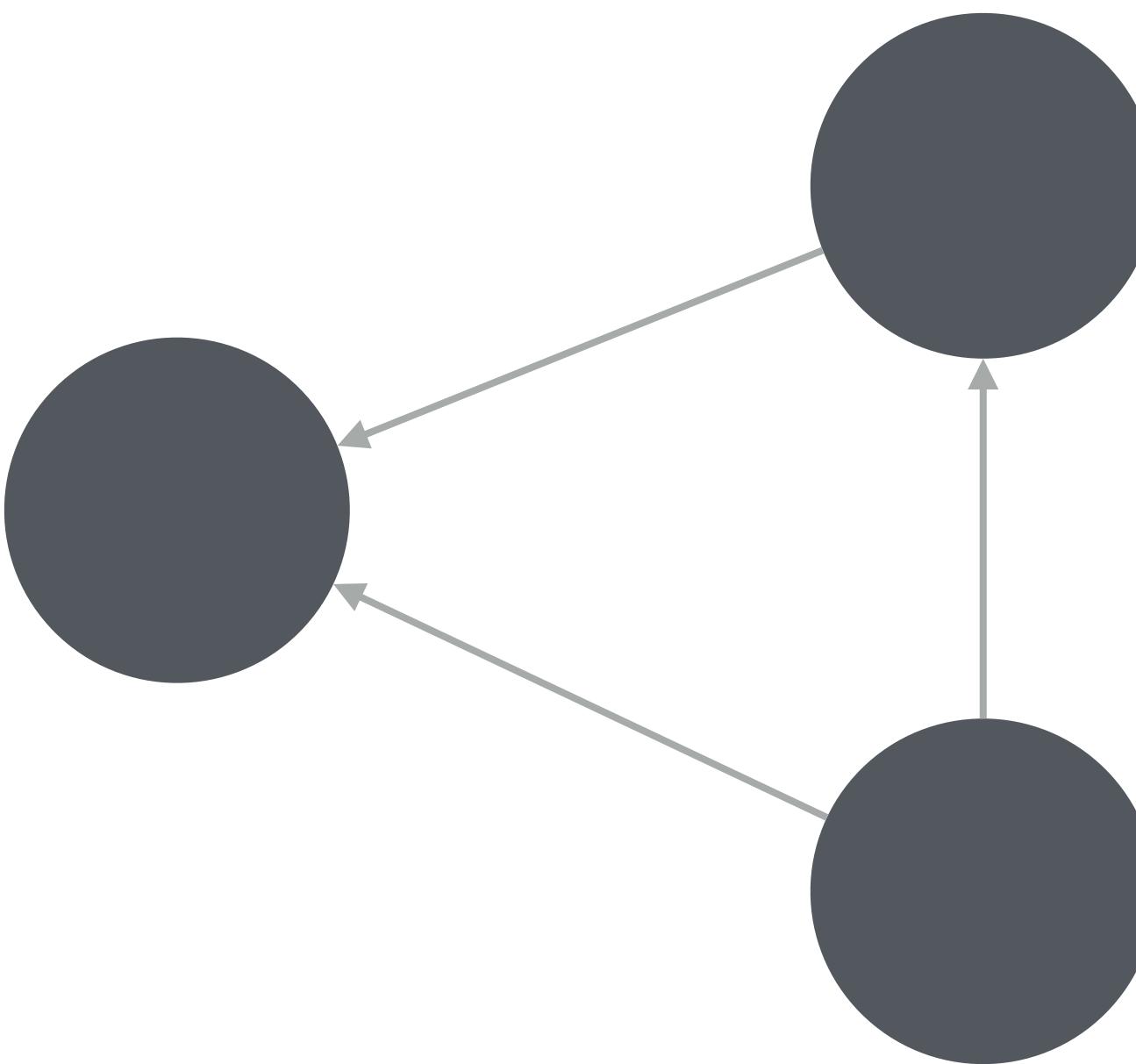
# Chart



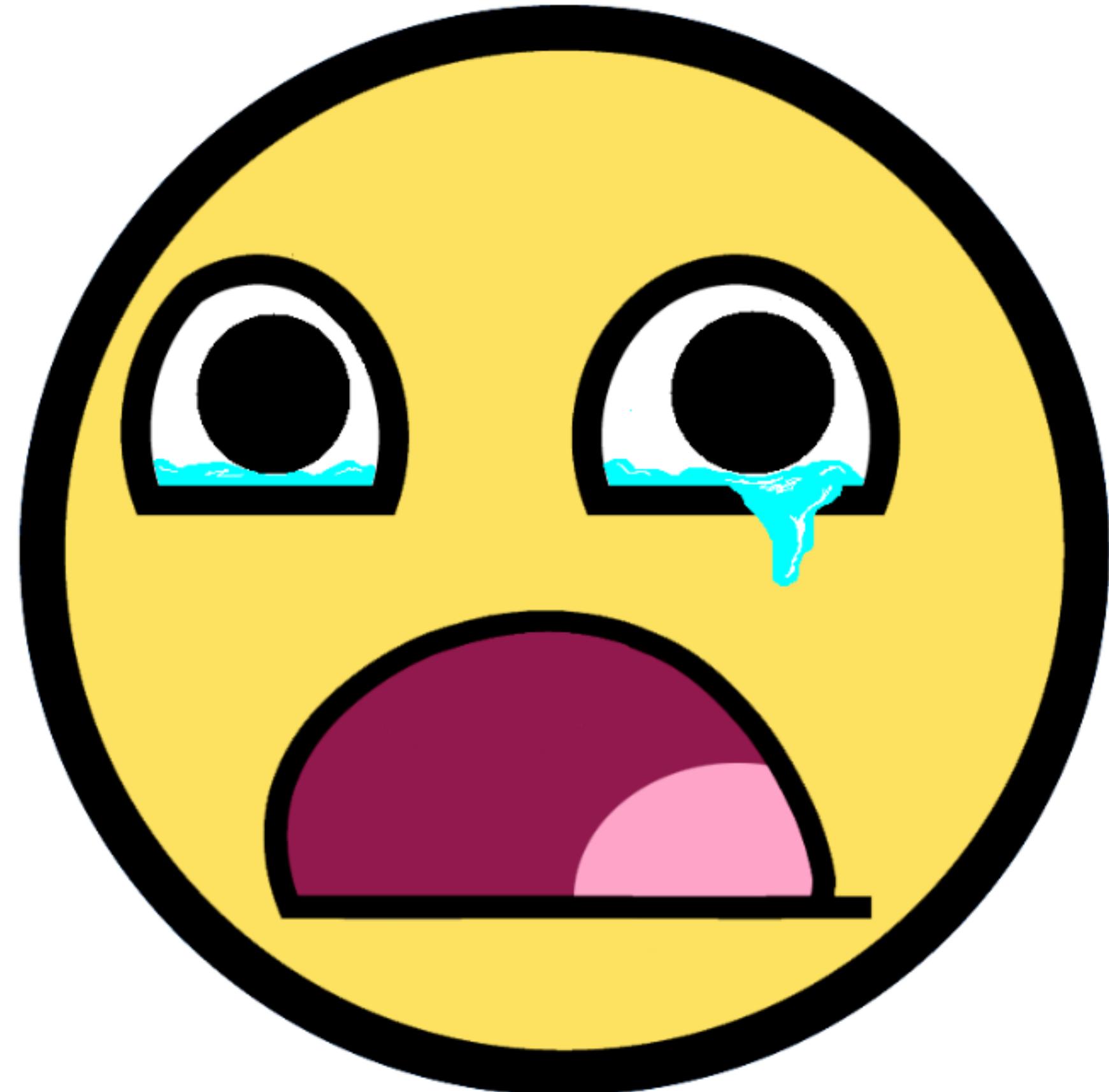
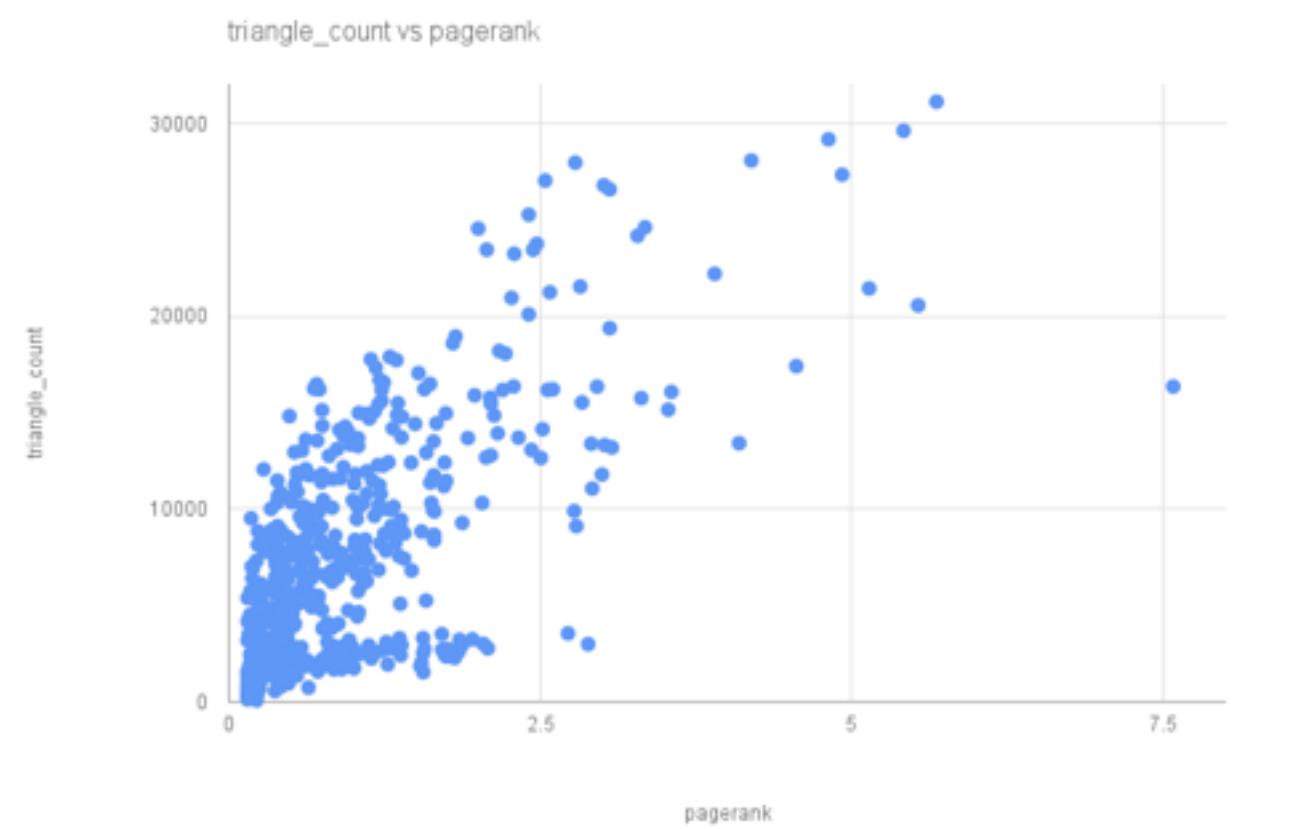
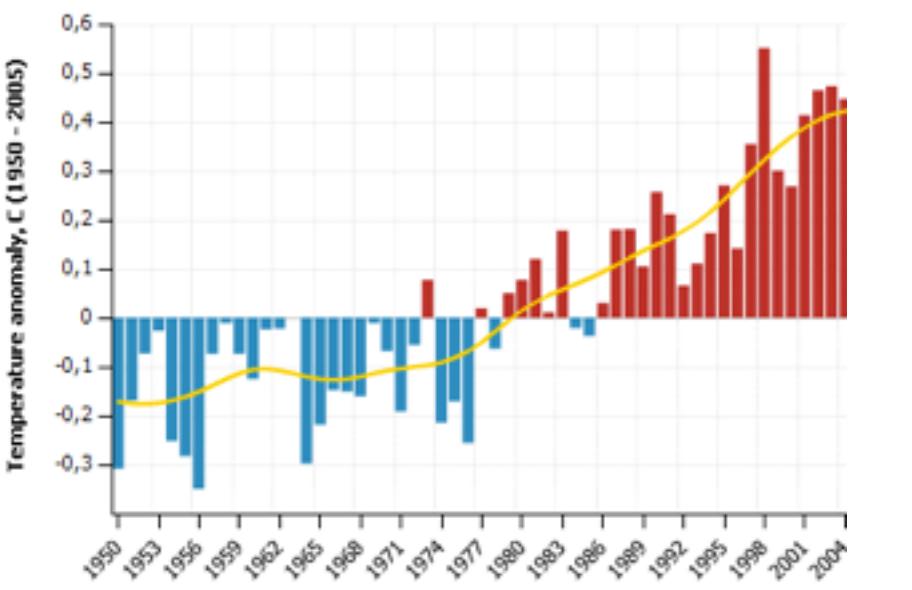
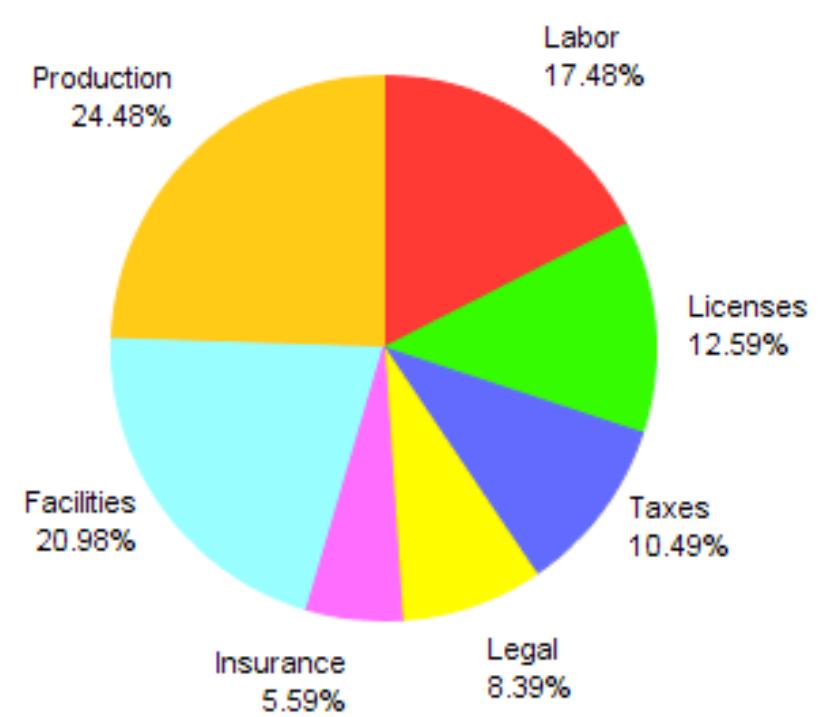
# Chart



# Graph!

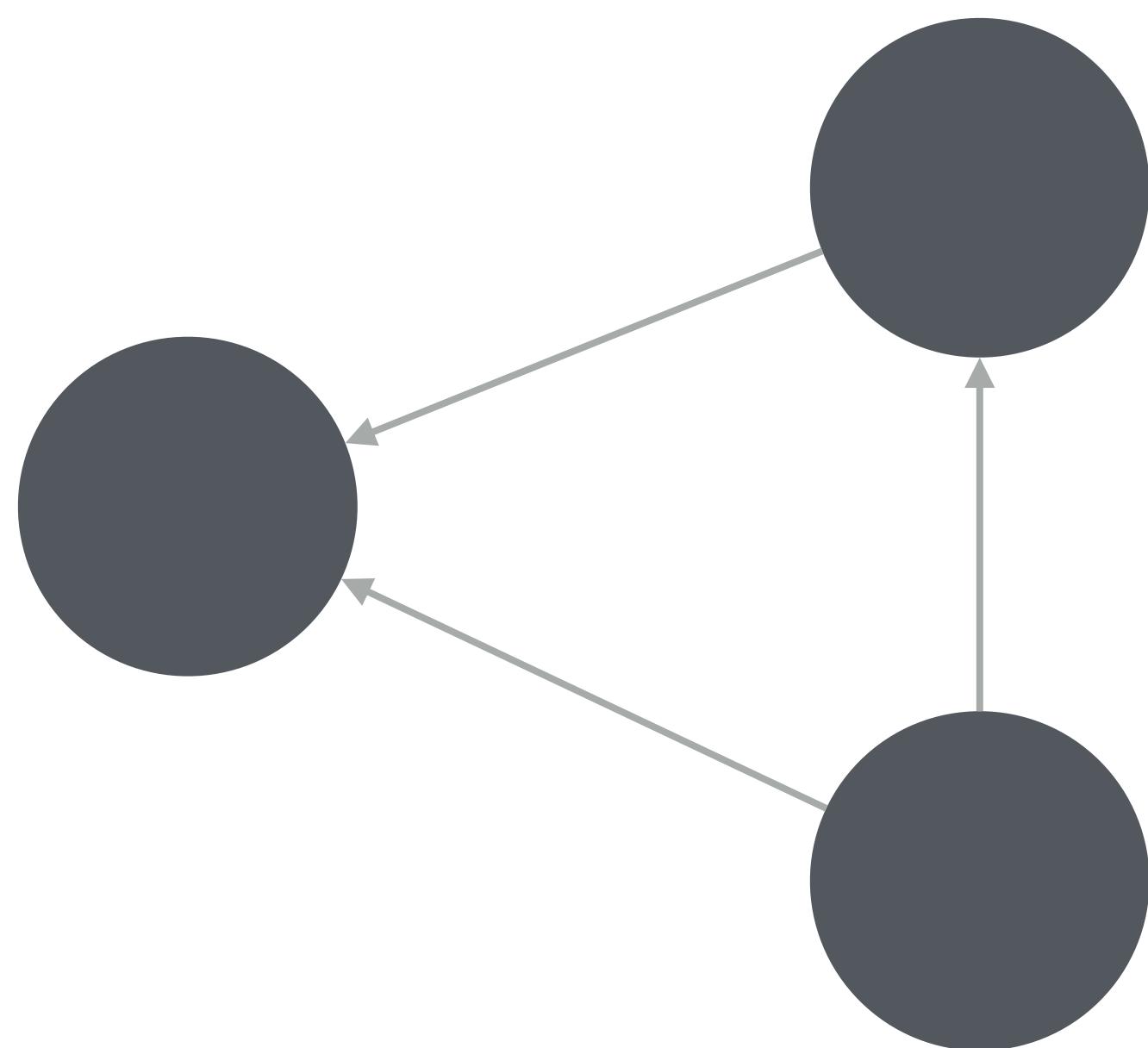


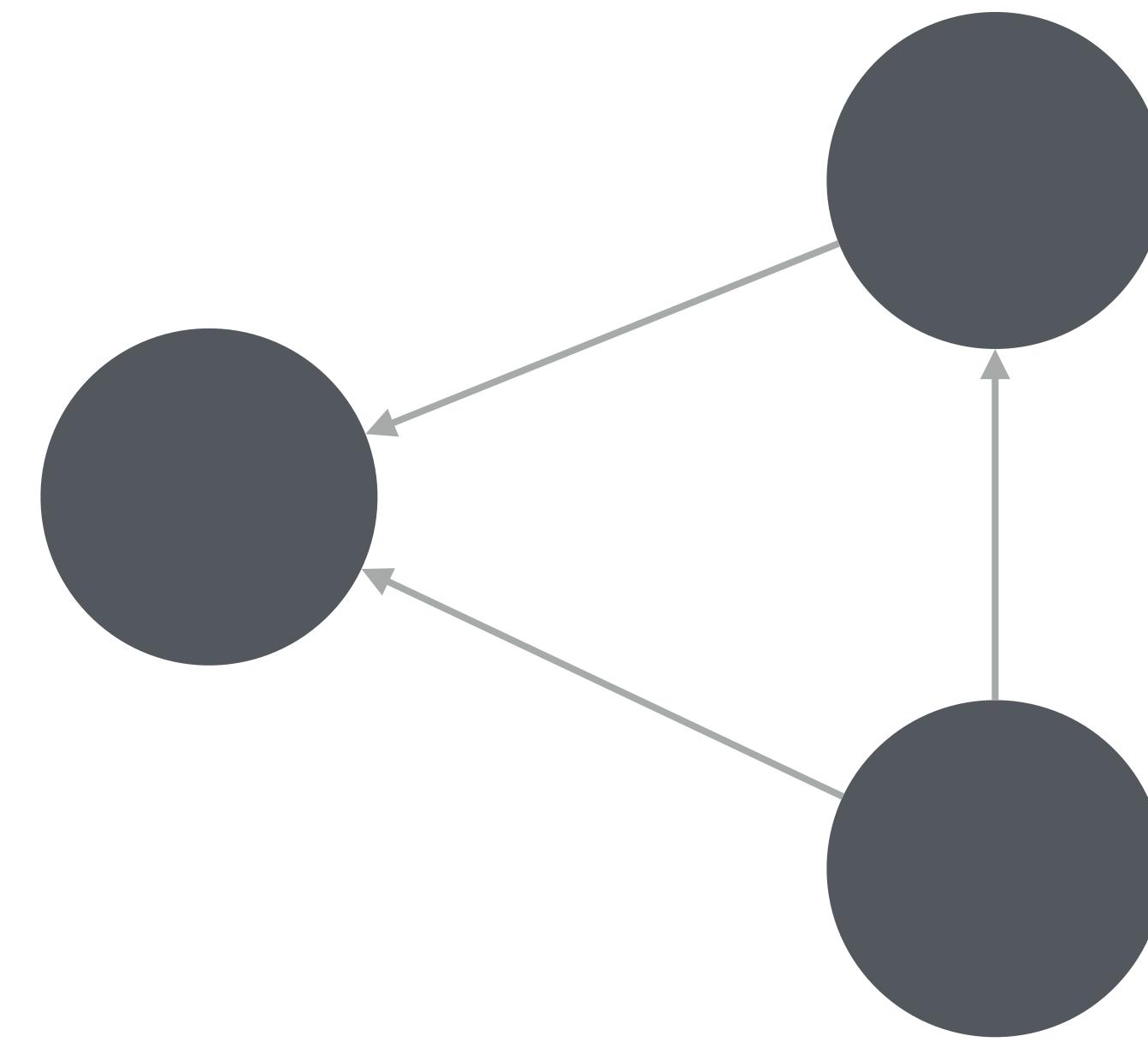
# Chart



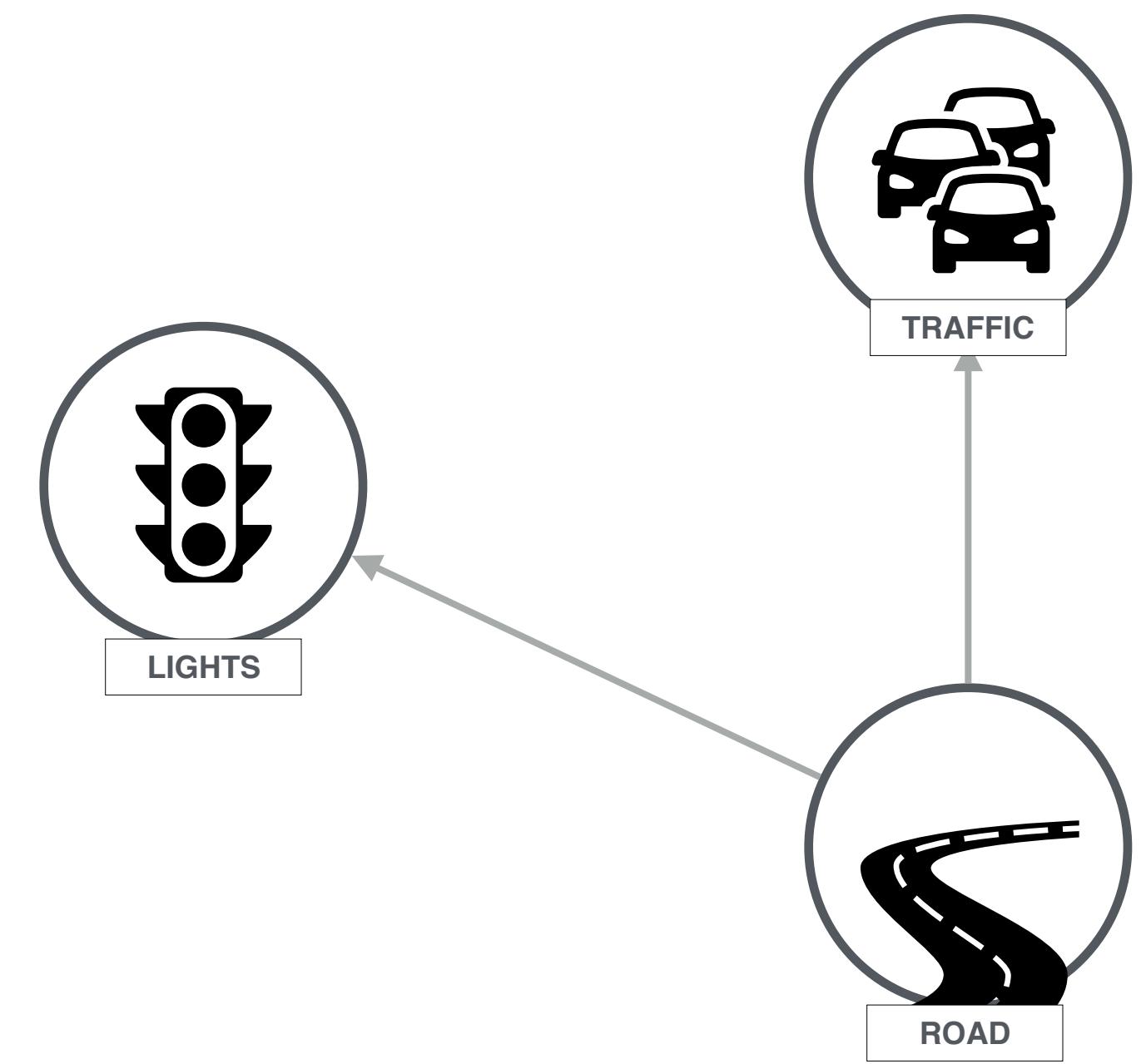


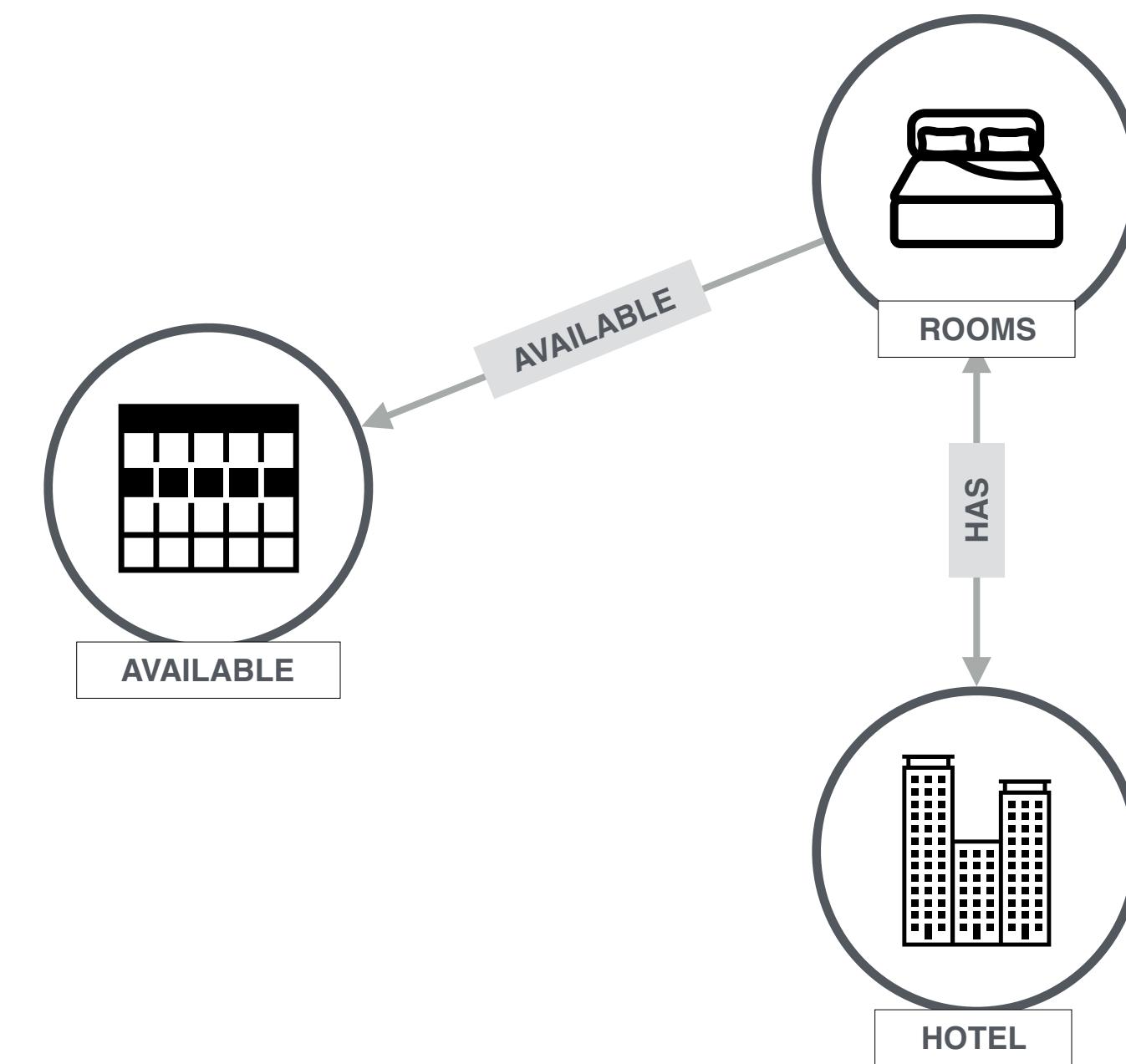
Graph!

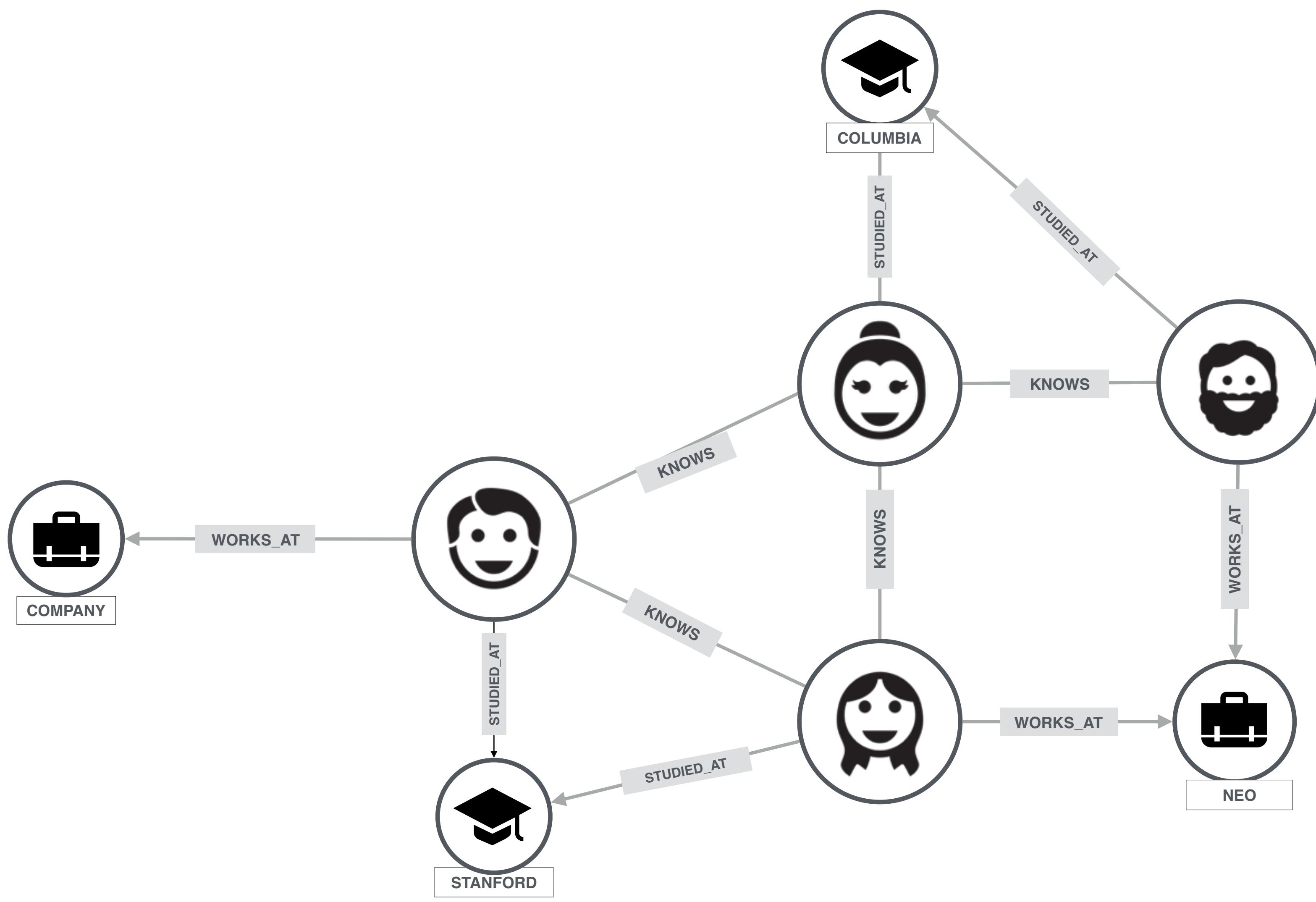


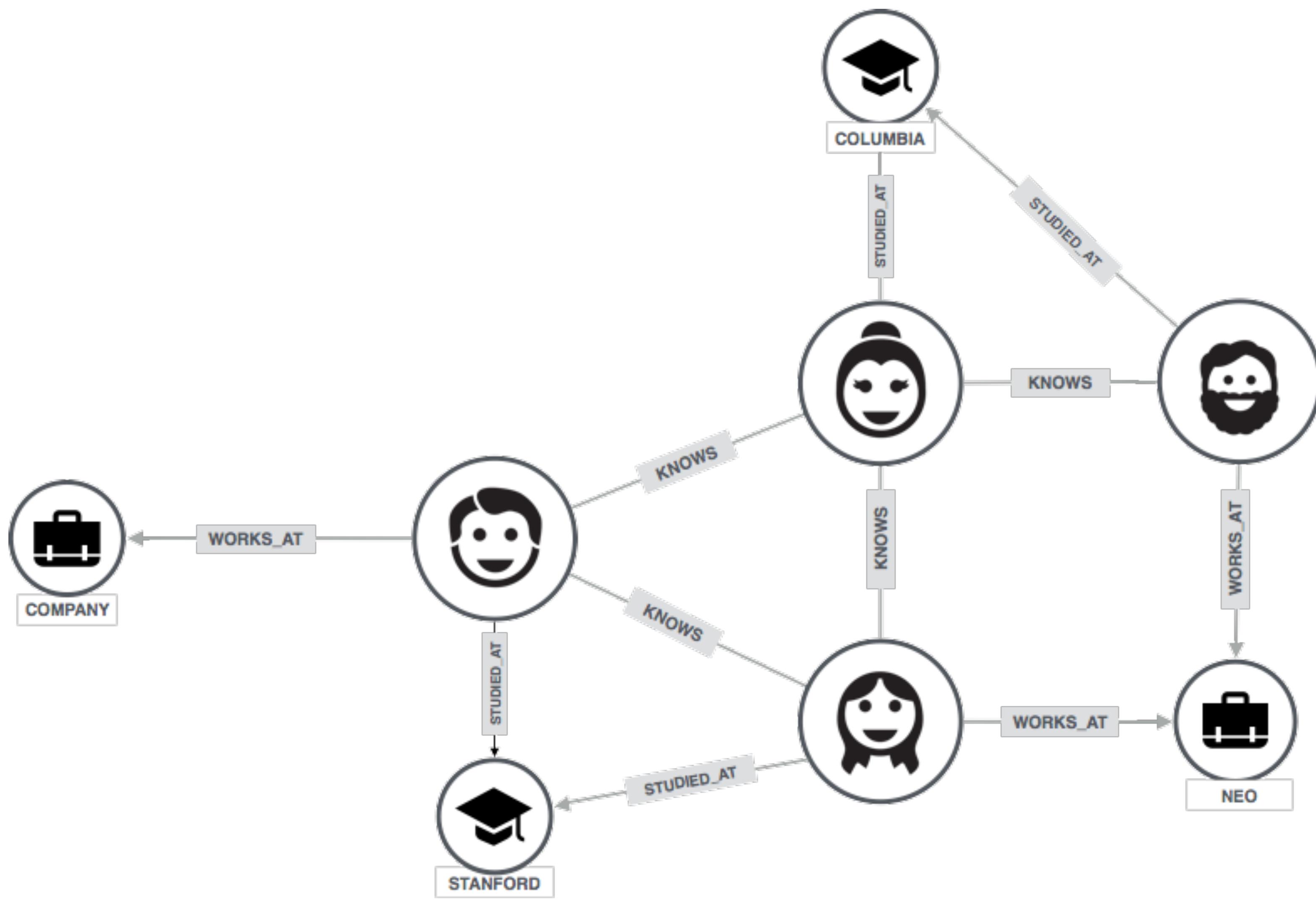


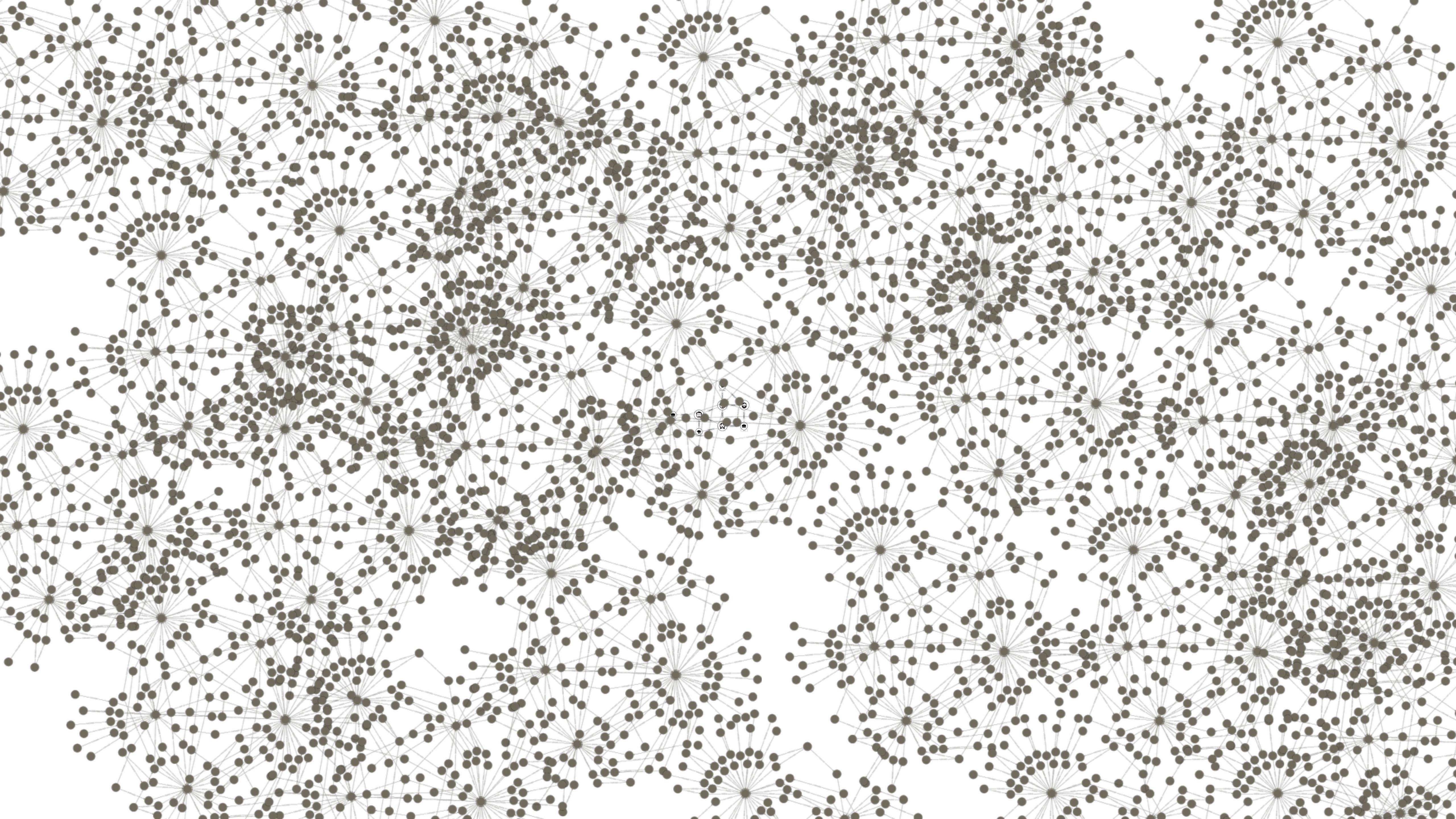
**A Graph Is Connected Data**







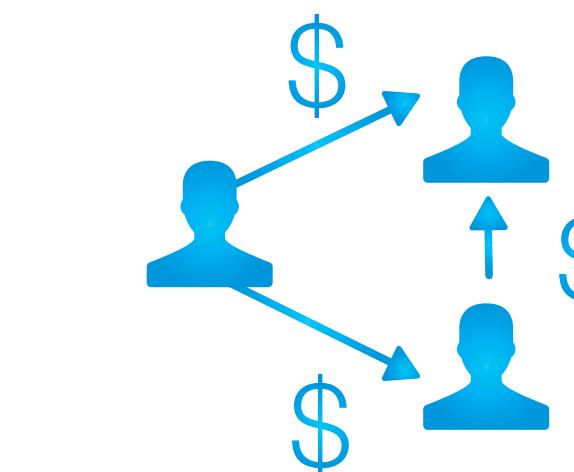
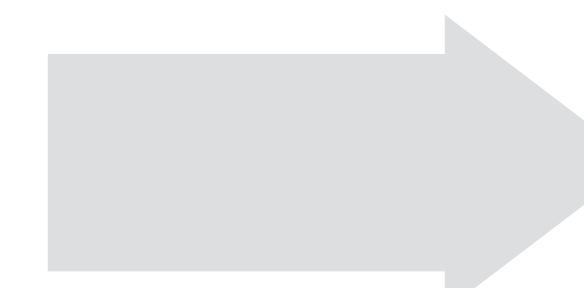
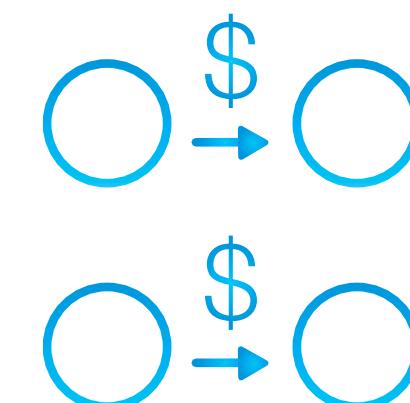
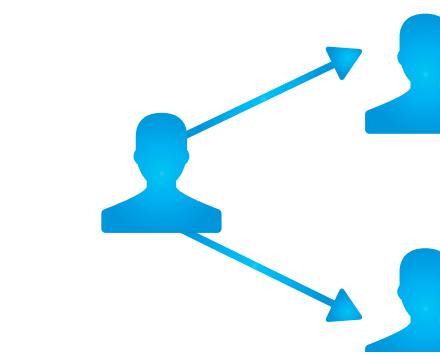
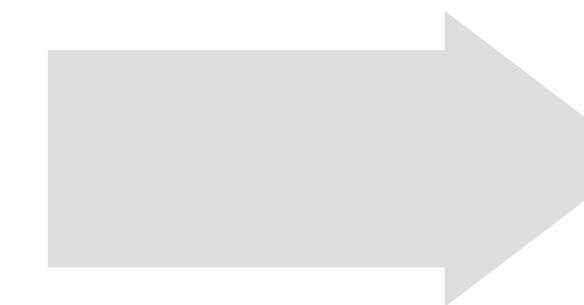
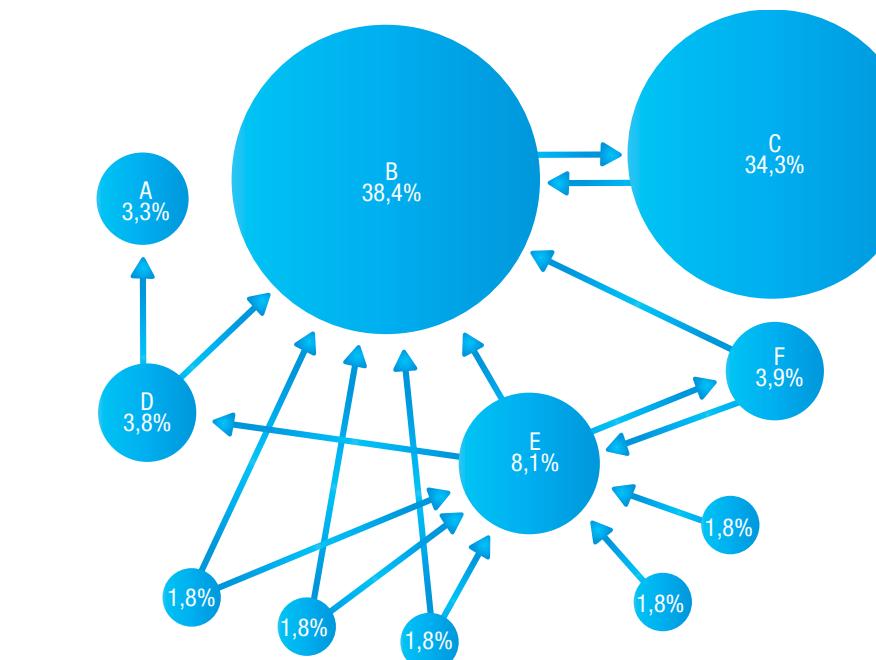
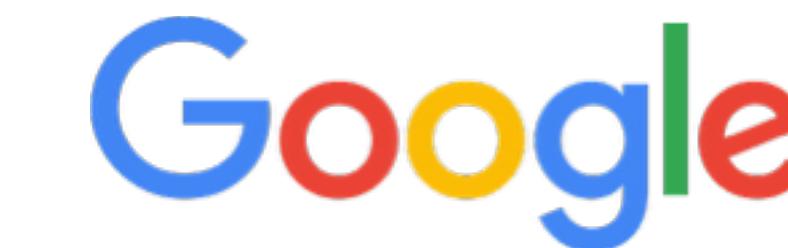
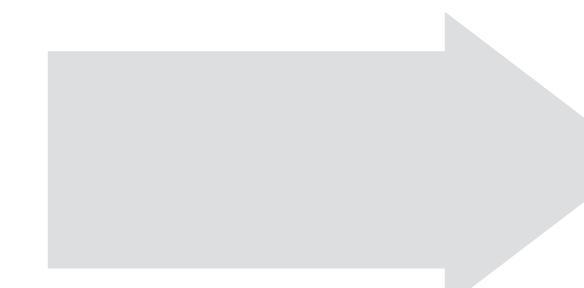




**But, who cares?**



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





Today we see graph-projects in virtually every industry



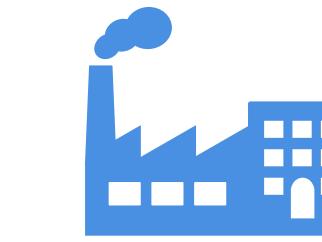
**Software**



**Finance**



**Retail**



**Media &  
Broadcasting**



**Social networks**



**Telco**



**Health Care**

# Neo4j Adoption by Selected Verticals



SOFTWARE	FINANCIAL SERVICES	RETAIL	MEDIA & BROADCASTING	SOCIAL NETWORKS	TELECOM	HEALTHCARE
<b>EQUILAR</b> <b>POLY VORE</b> <b>TRACEONE</b> <small>drive consumer trust</small> <b>53</b> <b>identropy</b> <b>ALERTLOGIC</b> <small>Security. Compliance. Cloud.</small> <b>NetApp</b> <b>Zenoss</b> <b>VIRTUAL INSTRUMENTS</b> <b>pitney bowes</b> <b>SERENA</b> <b>TOMTOM</b>	<b>UBS</b> <b>FirstData</b> <b>die Bayerische</b> <b>ADP</b> <b>Fortune 500 CONFIDENTIAL</b> <b>veda applied intelligence</b> <b>bpifrance</b> <b>NOMURA</b> <b>mallowstreet</b> <b>Ria</b> <b>noble group</b>	<b>Walmart</b> <b>eBay</b> <b>NORDSTROM</b> <b>B&amp;H PHOTO - VIDEO - PRO AUDIO</b> <b>ROPOSO</b> <small>What you ❤️ is fashion</small> <b>spree</b> <b>LifeWay</b> <small>Biblical Solutions for Life</small> <b>adidas</b>	<b>Beamly</b> <b>scrippsnetworks interactive</b> <b>livestation</b> <b>Aol.</b> <b>THOMSON REUTERS</b> <small>the answer company™</small> <b>Glowbl</b> <b>gamesys</b> <b>SCRIPPS</b> <b>Lufthansa</b>	<b>LinkedIn</b> <b>Medium</b> <b>SNAP INTERACTIVE</b> <b>ACADEMIA</b> <b>Care.com</b> <b>doximity</b> <b>Hinge</b> <b>InfoJobs</b> <b>classmates™</b>	<b>CISCO</b> <b>SFR</b> <b>telenor</b> <b>StarHub</b> <b>CenturyLink®</b> <b>rancard</b> <b>Three.com</b> <b>hp</b>	<b>RALLY™</b> <b>curaspan™</b> <b>CHRONOTRACK</b> <b>NaviNet</b> <small>part of NantHealth</small> <b>compugen</b> <b>doximity</b>



- HQ in Silicon Valley
  - Offices in London, Munich, Paris and Malmö
- 200+ enterprise subscription customers including over 60 of the Global 2000
- 1,000,000+ downloads, adding 50,000+ per month
- 25,000+ education registrants
- 40,000 Meetup members
- 100+ technology and service partners

**But why does the database  
need to be a graph?**

Name	Country	Hair	Uni
John	UK	Red	Princeton
Mary	USA	Blonde	Yale
Li	China	Black	Princeton
Kate	UK	Brown	Princeton
Michal	Israel	Brown	Brown

ID	Country
17	UK
12	USA
19	China
17	UK
112	Israel

ID	Country	Leader
17	UK	Cameron
12	USA	Obama
19	China	XiJinping
17	UK	Cameron
112	Israel	Bibi

Name	Country	Hair	Uni
John	17	Red	Princeton
Mary	12	Blonde	Yale
Li	19	Black	Princeton
Kate	17	Brown	Princeton
Michal	112	Brown	Brown

ID	Name	Pres	State
92	Princeton	Eisgrub	NJ
34	Yale	Salovey	CT
1	Brown.	Paxson	RI

Name	Country	Hair	Uni
John	17	Red	92
Mary	12	Blonde	34
Li	19	Black	92
Kate	17	Brown	92
Michal	112	Brown	1

ID	Country	Leader
17	UK	Cameron
12	USA	Obama
19	China	Xi Jinping
17	UK	Cameron
112	Israel	Bibi

ID	Name	Pres	State
92	Princeton	Eisgrub	NJ
34	Yale	Salovey	CT
1	Brown	Paxson	RI

Name	Country	Hair	Uni
John	17	Red	92
Mary	12	Blonde	34
Li	19	Black	92
Kate	17	Brown	92
Michal	112	Brown	1

```
SELECT
    p.name,
    c.country, c.leader, p.hair,
    u.name, u.pres, u.state
FROM
    people p
    LEFT JOIN country c ON c.ID=p.country
    LEFT JOIN uni u ON p.uni=u.id
WHERE
    u.state='CT'
```





**Have you seen  
Ted's UUID?**

Name	Country	Leader	Hair	Uni	Pres	State
John	UK	Cameron	Red	Princeton	Eisgrub	NJ
Mary	USA	Obama	Blonde	Yale	Salovey	CT
Li	China	Xi Jinping	Black	Princeton	Eisgrub	NJ
Kate	UK	Cameron	Brown	Princeton	Eisgrub	NJ
Michal	Israel	Bibi	Brown	Brown	Paxson	RI

# SQL Trouble

- Complex to model and store relationships
- Performance degrades with increases in data
- Queries get long and complex
- Maintenance is painful

# Graph Motivations

- Easy to model and store relationships
- Performance of relationship traversal remains constant with growth in data size
- Queries are shortened and more readable
- Adding additional properties and relationships can be done on the fly - no migrations

**What does this  
Property Graph look like?**

# Four Building Blocks

- Nodes
- Relationships
- Properties
- Labels

# Properties

- Key Value Pairs that are stored on either nodes or relationships
  - Used to represent entity attributes and/or metadata (e.g. timestamps, version)
  - Key-value pairs
    - Java primitives including Strings, Floats, Longs, etc. etc.
    - Arrays
    - null is not a valid value
  - Every node can have a different number of properties as well as different types of properties

# Nodes

- Used to represent entities and complex value types in your domain
  - Entities
    - Have unique conceptual identity
    - Change attribute values, but identity remains the same
  - Value types
    - No conceptual identity
    - Can substitute for each other if they have the same value
    - Simple: single value (e.g. color, category)
    - Complex: multiple attributes (e.g. address)

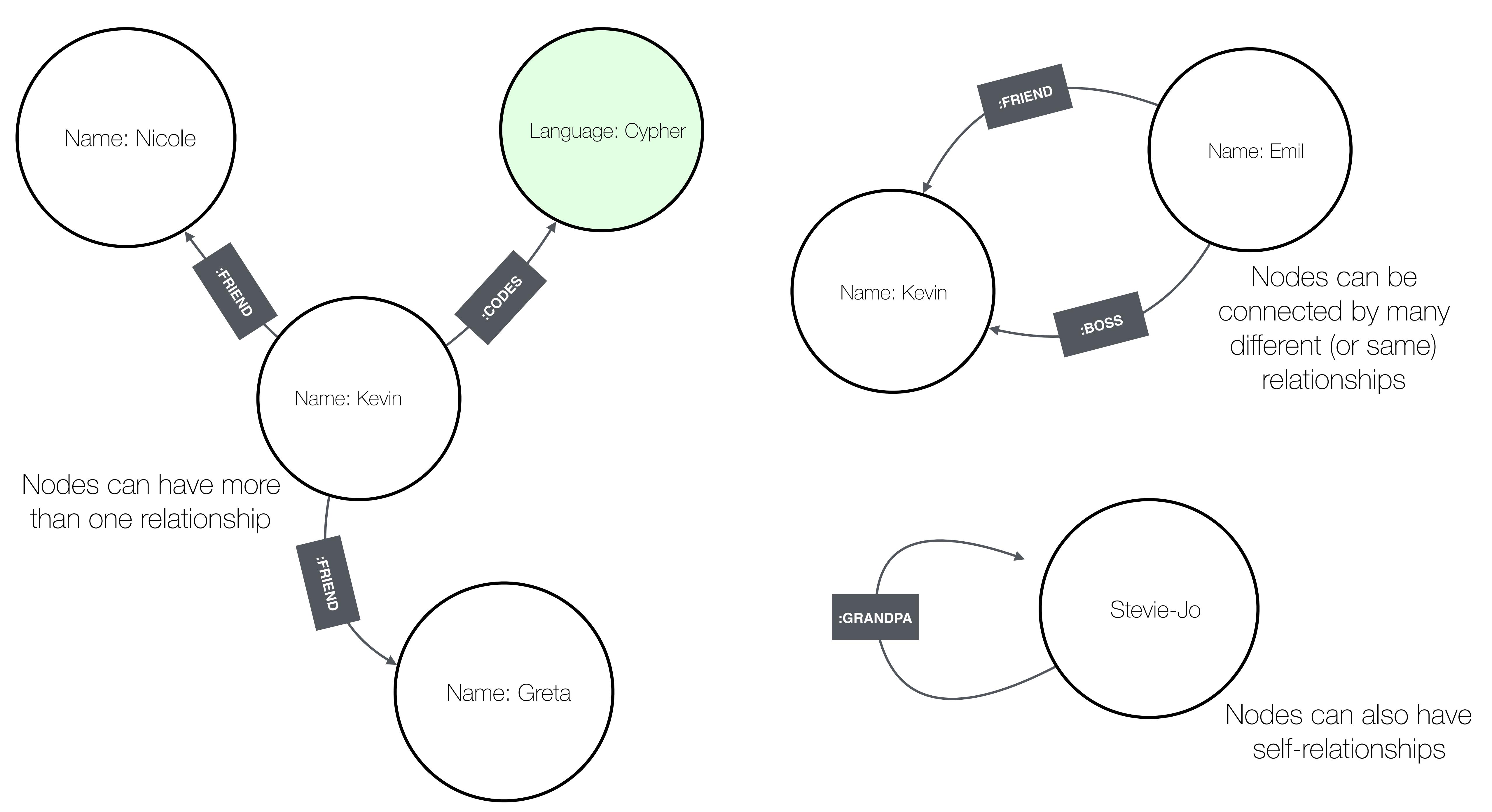
# Relationships

- Every relationship has a name and a direction
  - Add structure to the graph
  - Provide semantic context for nodes
- Can contain properties
  - Used to represent quality or weight of relationship, or metadata
- Every relationship must have a start node and end node
  - No dangling relationships

# Labels

- Every node can have zero or more labels
- Typically used to group similar nodes into sub-graphs
  - Roles: e.g., user or product
- Allows us to associate indexes and constraints with groups of nodes

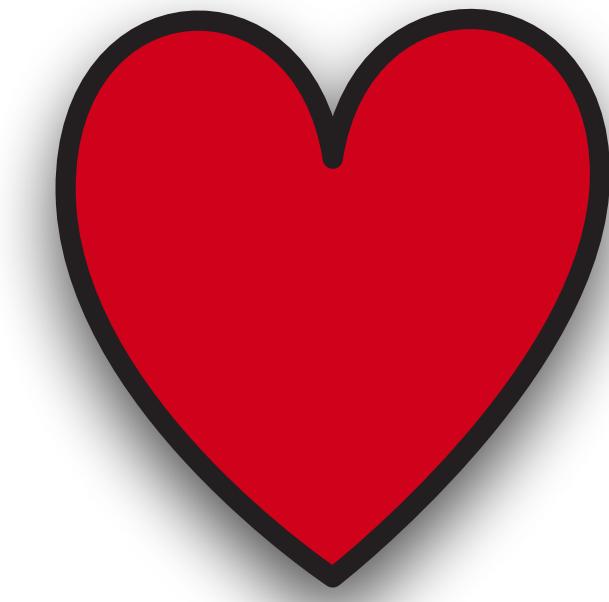
```
CREATE (p:Person)    // create labeled node  
SET p:Person          // set label on node  
REMOVE p:Person        // remove label  
MATCH (p:Person)       // match nodes with label  
RETURN labels(p)        // get label collection
```



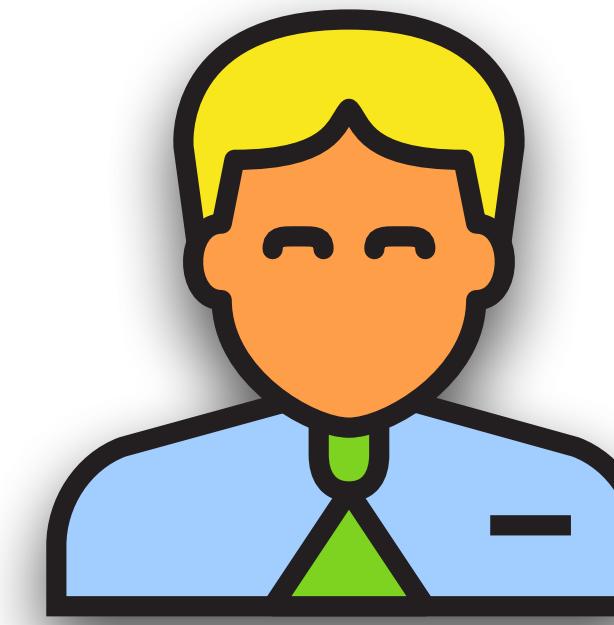
# Ann Loves Dan



Ann

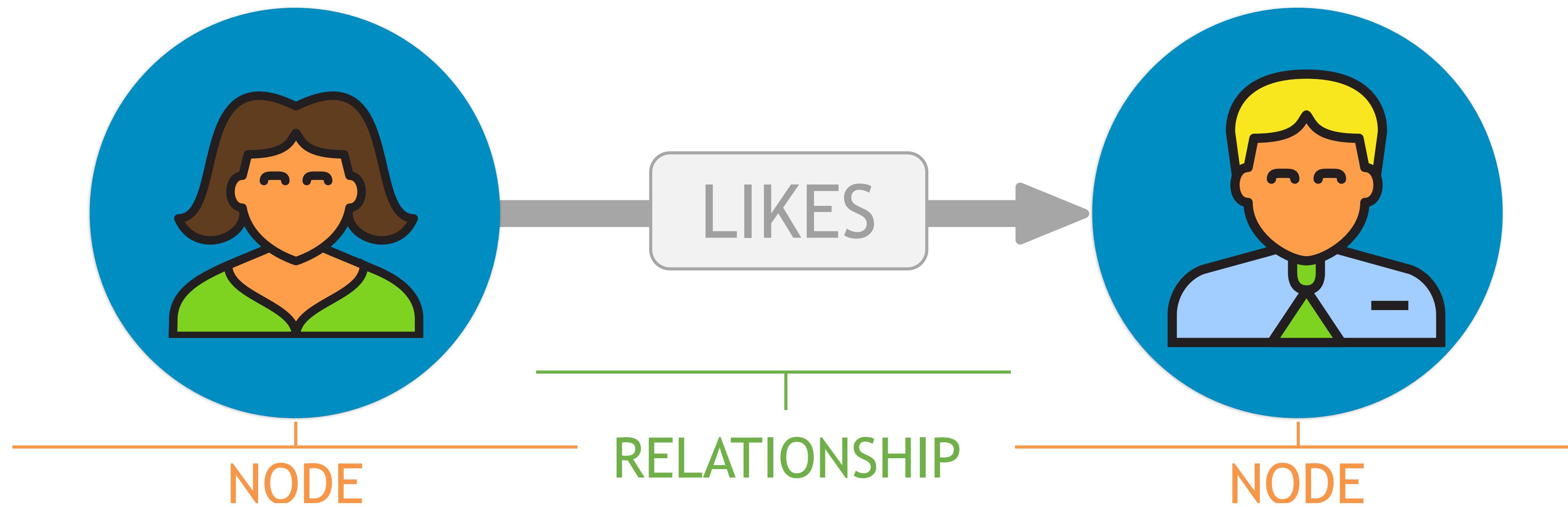


Loves

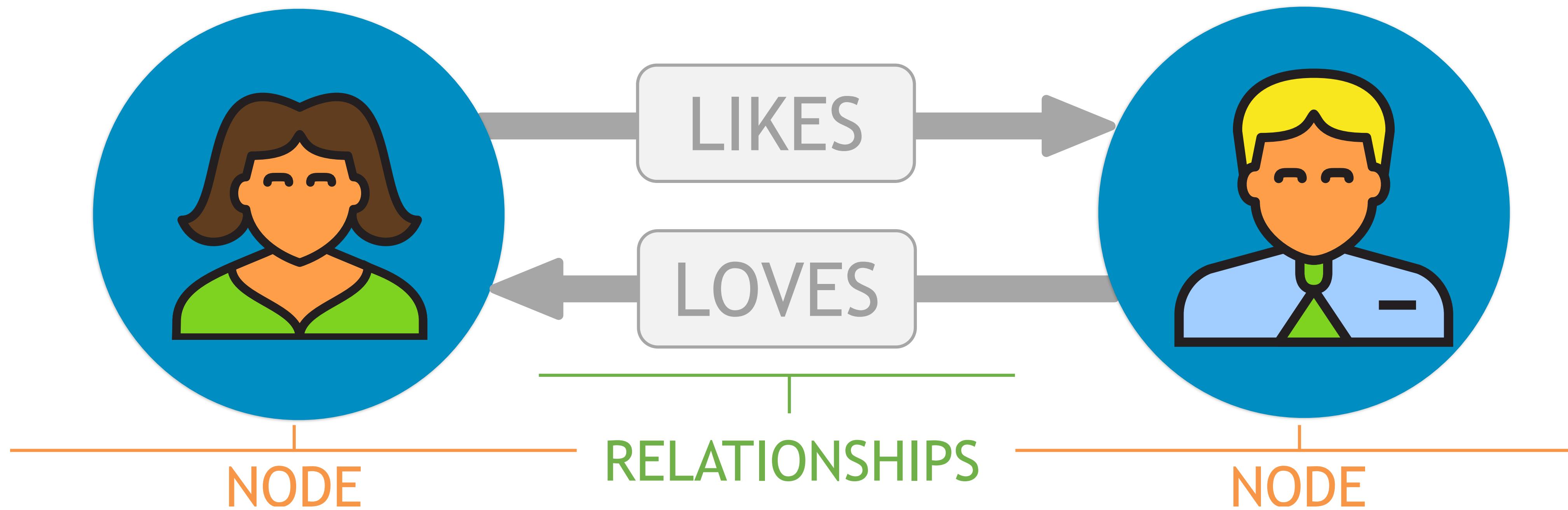


Dan

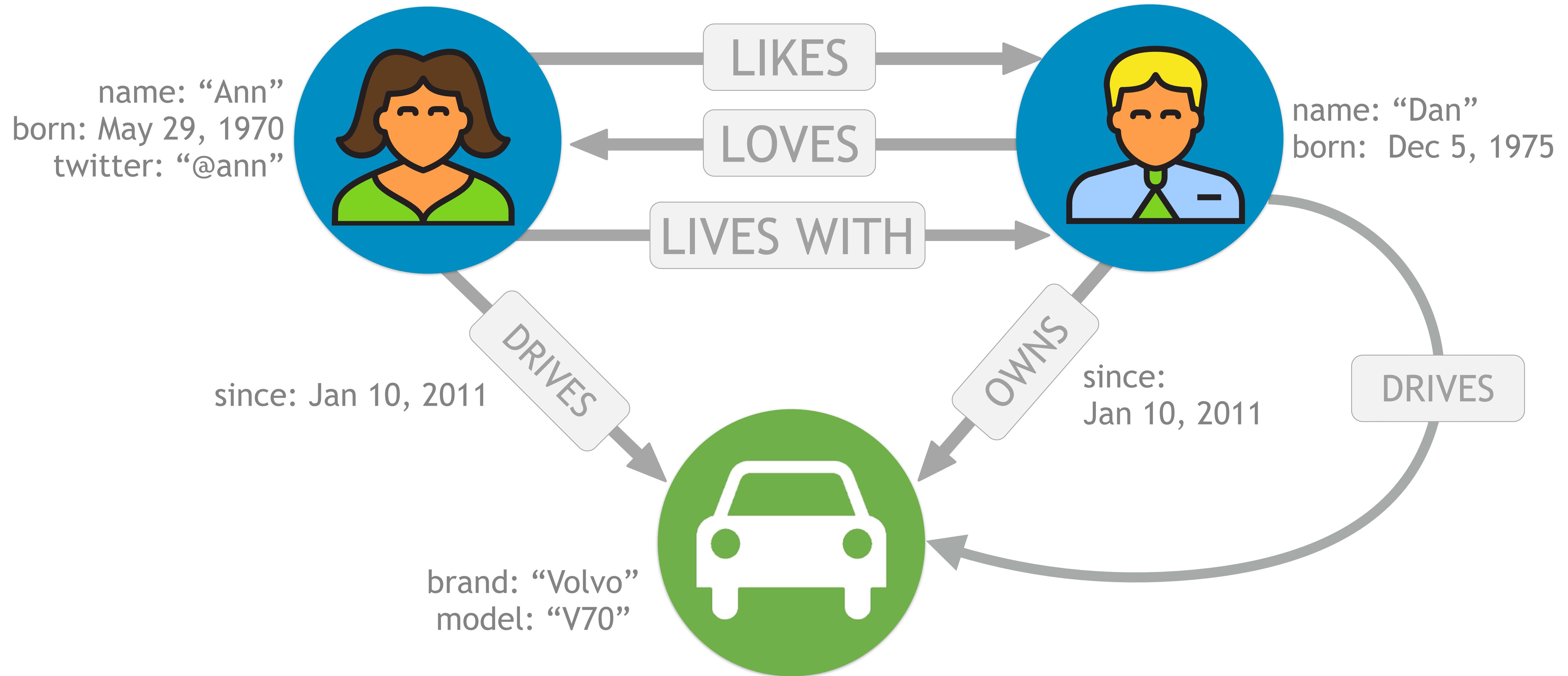
# Ann Loves Dan



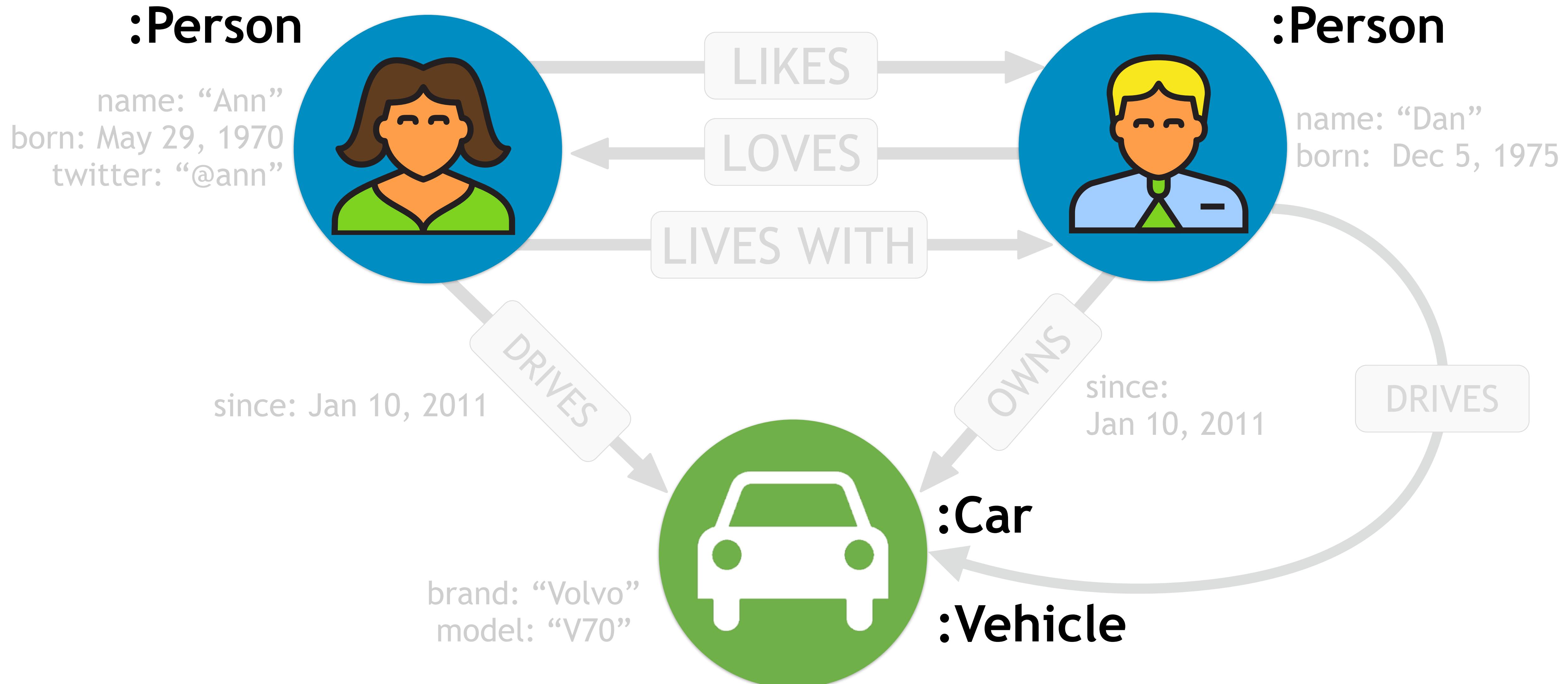
# Relationships are Directional



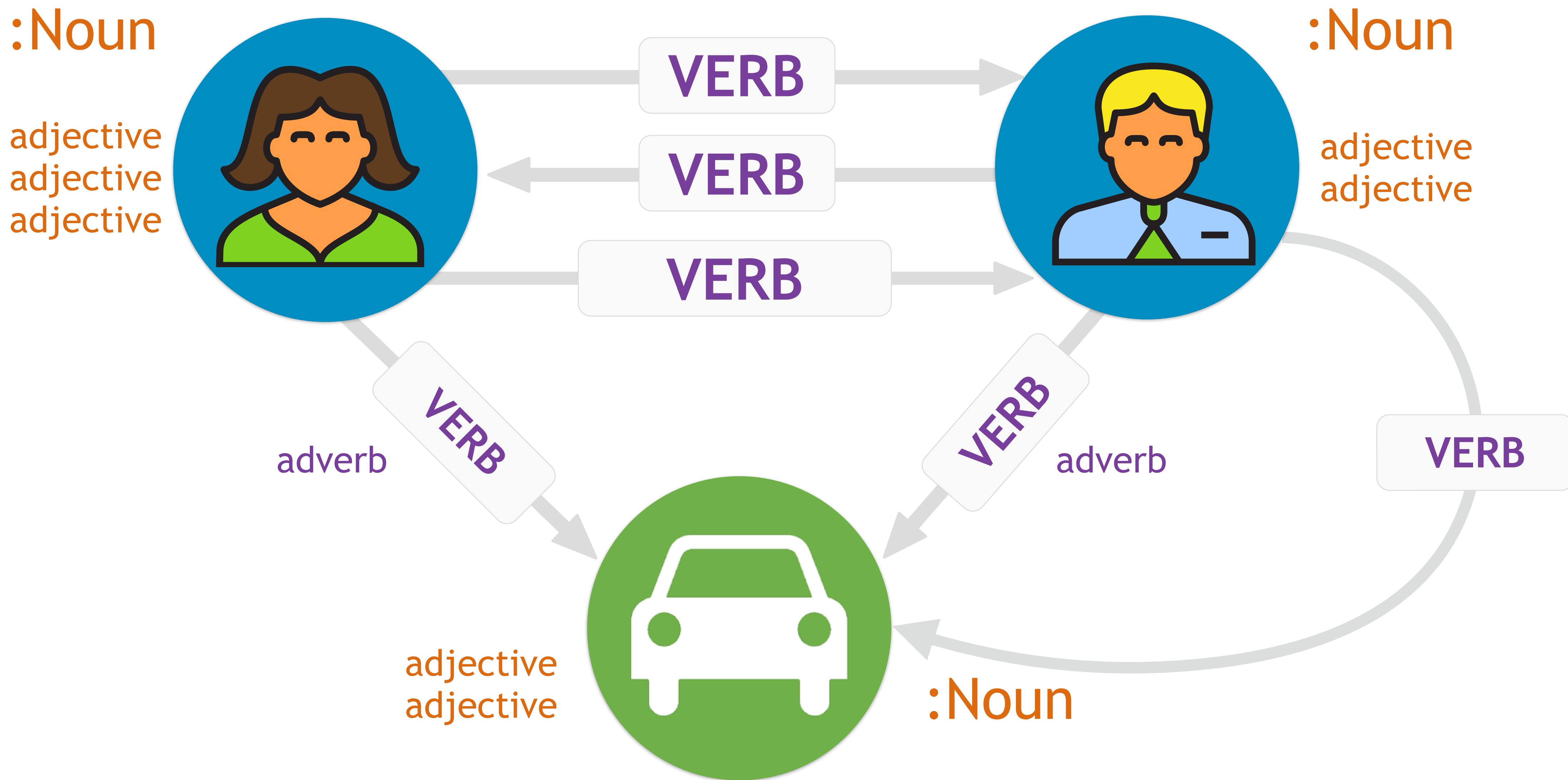
# Detailed Property Graph



# Labeled Property Graph



# Mapping to Languages



Cypher Tutorial —> Open Neo4jFundamentals.html