



Graph Data Modeling

Tips and Tricks



About Me!

01

Max De Marzi
Neo4j Field Engineer

02

maxdemarzi.com
About 160 blog posts

03

@maxdemarzi

04

github.com/maxdemarzi
About 200 public repositories

Agenda

- Property Graph Data Model
- The **most important** Slide about Neo4j **you will ever see**
- Basics of Modeling
- Modeling **Acting**
- Modeling **Flights**
- Modeling **Twitter**
- Modeling **Forms**
- Modeling **Chains**

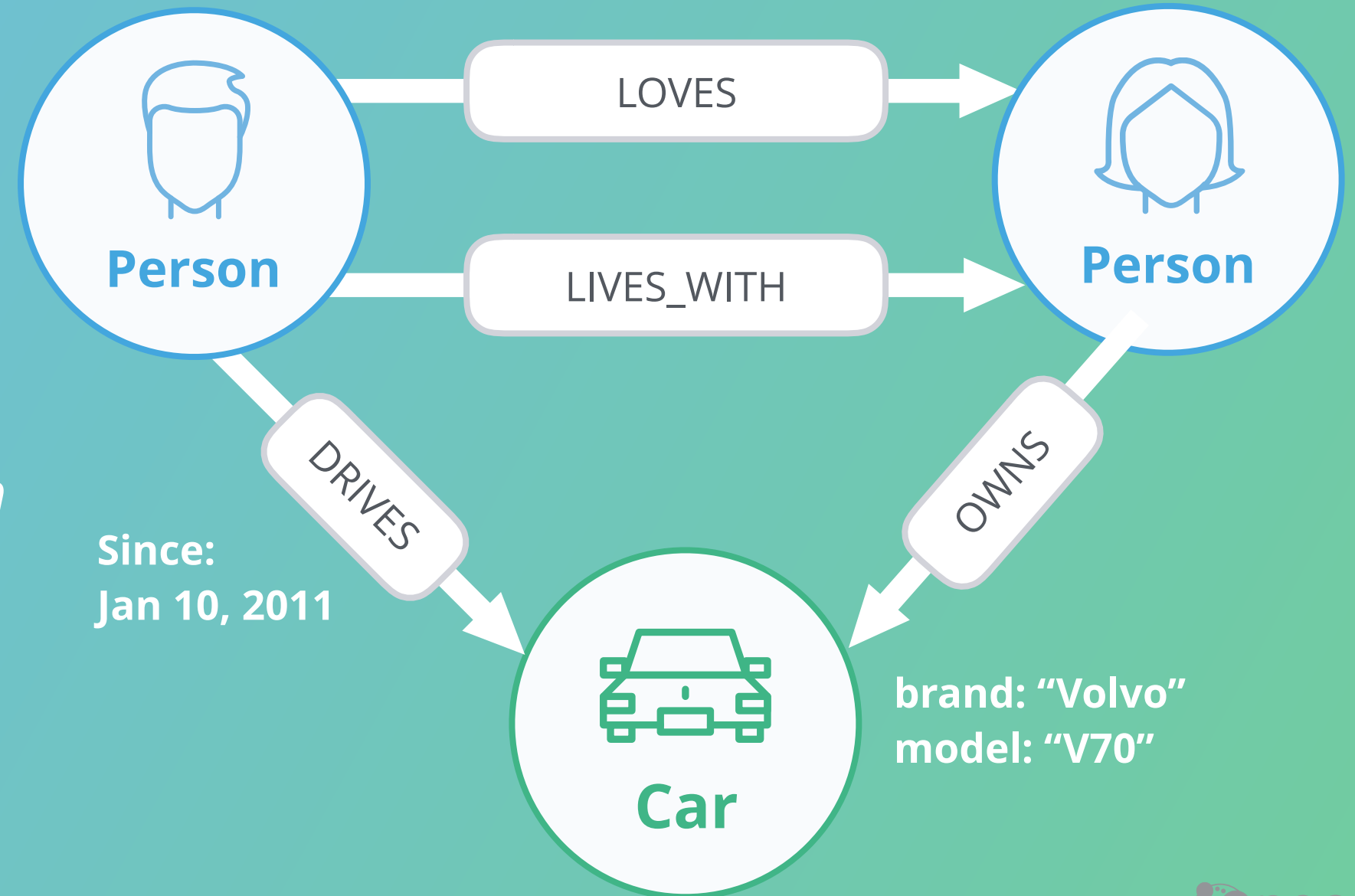


Property Graph Model

Property Graph Model Components

name:"Dan"
born: May 29, 1970
twitter:"@dan"

name:"Ann"
born: Dec 5, 1975



Nodes

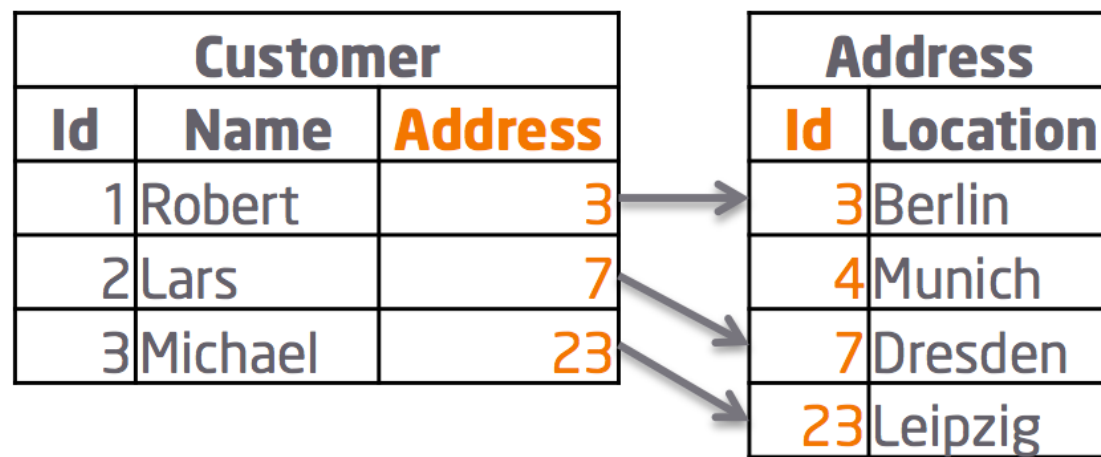
- Can have *Labels*
- Can have *Properties*

Relationships

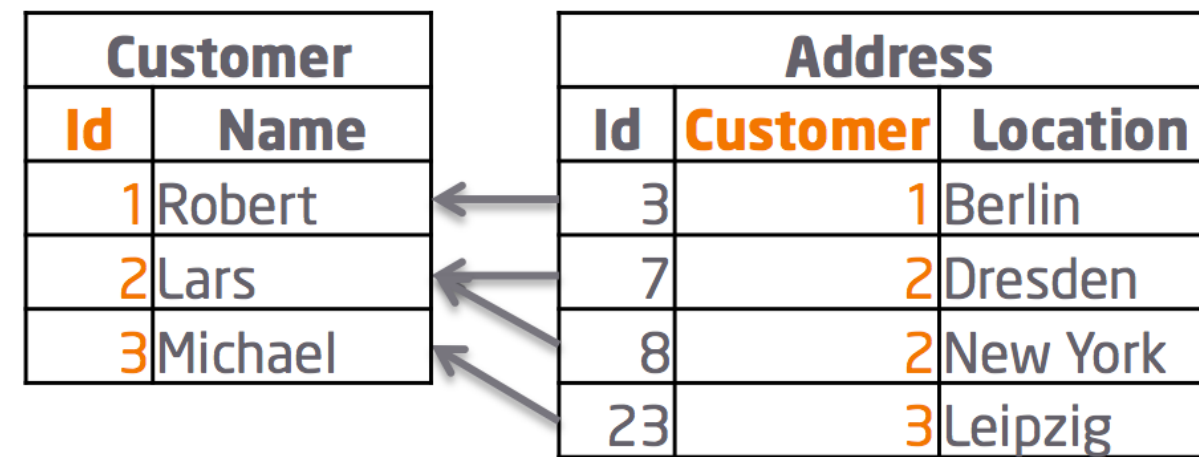
- Relate nodes by *type* and *direction*
- Can have *Properties*

What you (probably) already know:

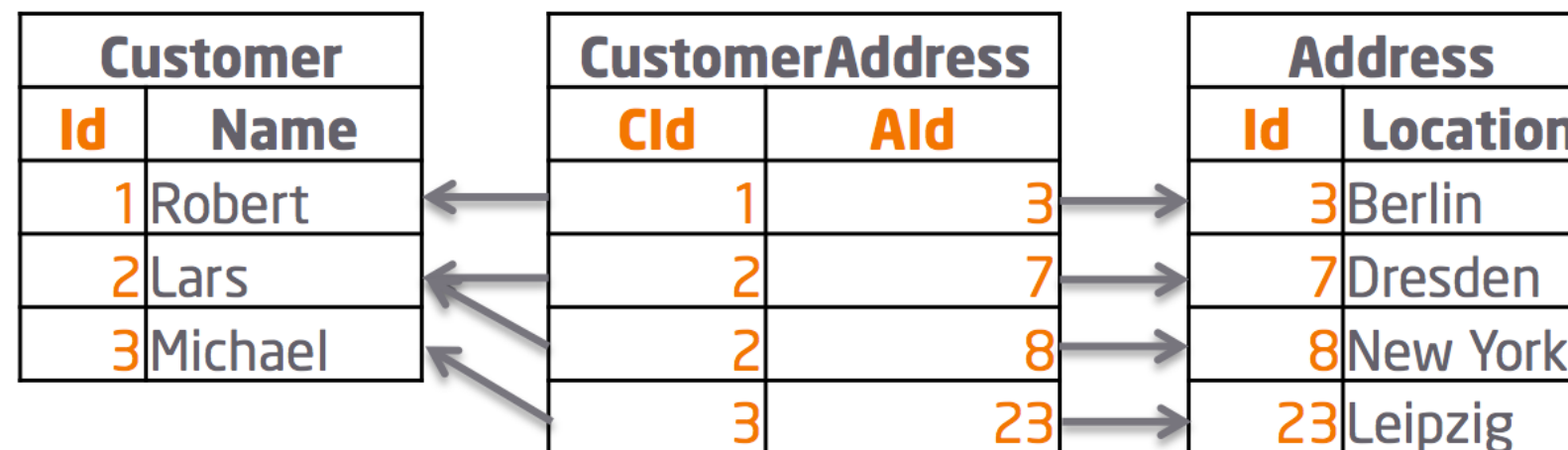
SQL Join Hell ⁽¹⁾



1:1 Relationship

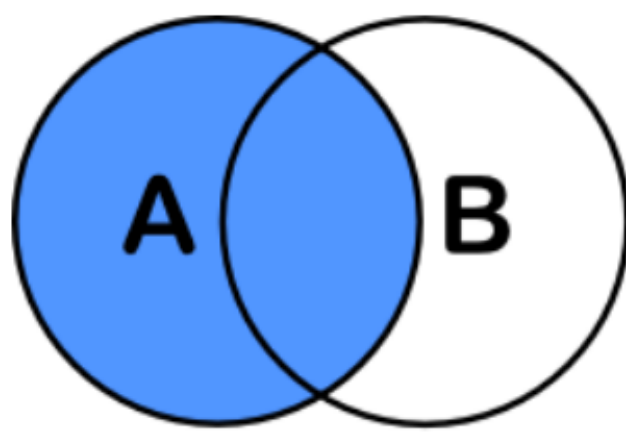


1:n Relationship

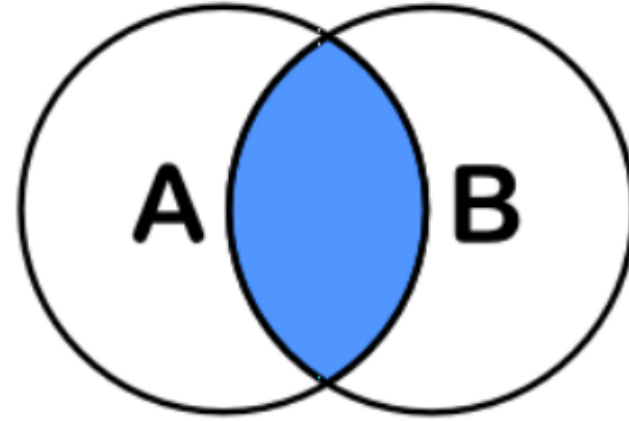


m:n Relationship

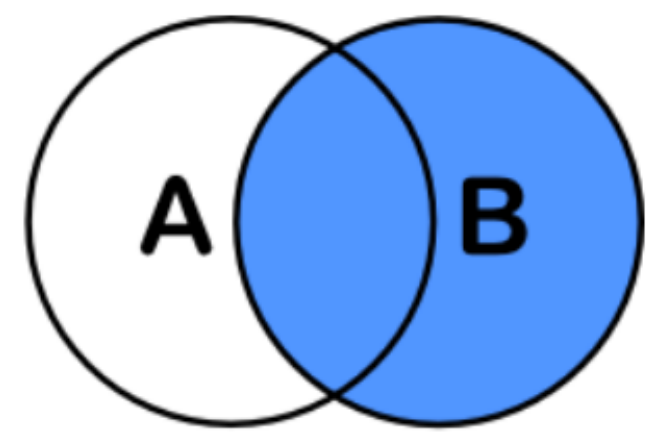
CHEATSHEET
**SQL
JOINS**



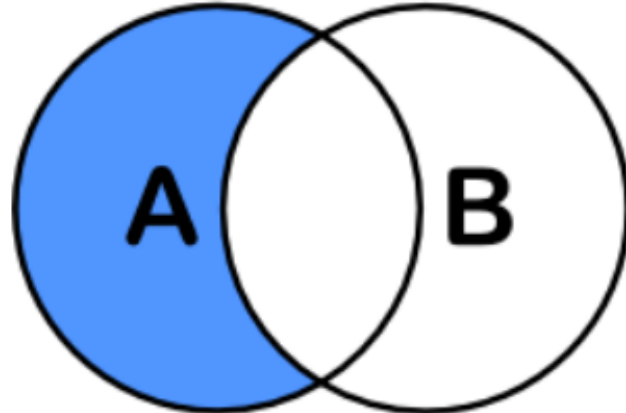
```
SELECT <auswahl>  
FROM tabelleA A  
LEFT JOIN tabelleB B  
ON A.key = B.key
```



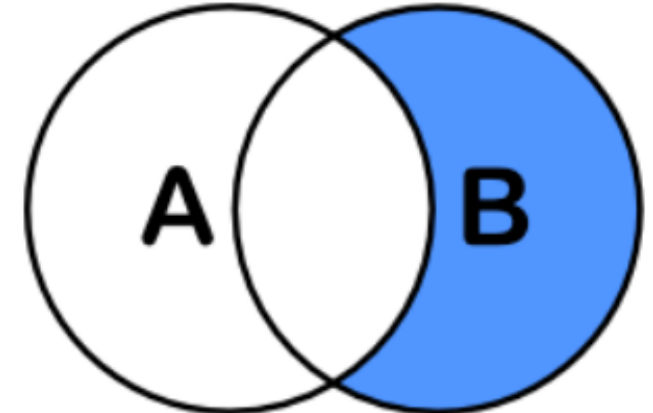
```
SELECT <auswahl>  
FROM tabelleA A  
INNER JOIN tabelleB B  
ON A.key = B.key
```



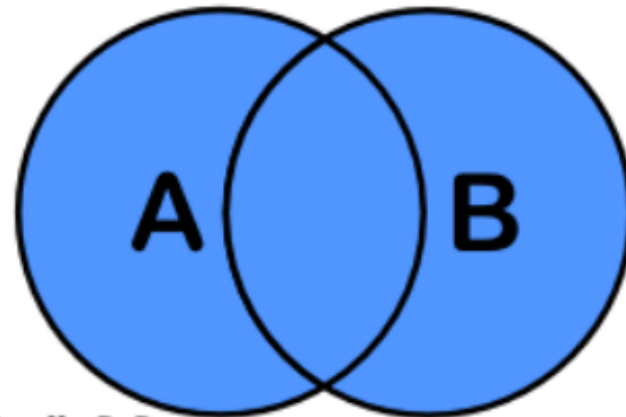
```
SELECT <auswahl>  
FROM tabelleA A  
RIGHT JOIN tabelleB B  
ON A.key = B.key
```



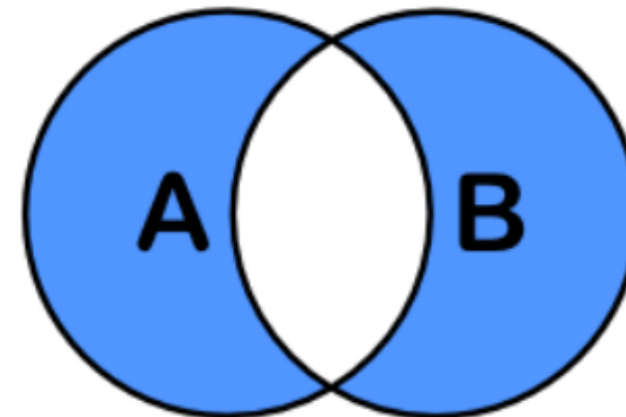
```
SELECT <auswahl>  
FROM tabelleA A  
LEFT JOIN tabelleB B  
ON A.key = B.key  
WHERE B.key IS NULL
```



```
SELECT <auswahl>  
FROM tabelleA A  
RIGHT JOIN tabelleB B  
ON A.key = B.key  
WHERE A.key IS NULL
```



```
SELECT <auswahl>  
FROM tabelleA A  
FULL OUTER JOIN tabelleB B  
ON A.key = B.key
```



```
SELECT <auswahl>  
FROM tabelleA A  
FULL OUTER JOIN tabelleB B  
ON A.key = B.key  
WHERE A.key IS NULL  
OR B.key IS NULL
```


The background of the slide is a reproduction of the painting 'The Scream' by Edvard Munch. It depicts a figure in the foreground with a pale, yellowish face and an open mouth, as if screaming. The background shows a dark, turbulent sea and a sky with swirling, vibrant colors of orange, red, and yellow, suggesting a sunset or sunrise. The overall mood is one of intense emotional distress or mental anguish.

The Problem

1

Joins are executed **every time** you query the relationship

2

Executing a Join means to **search** for a key

3

B-Tree Index: $O(\log(n))$

Your data grows by 10x, your time goes up by one step on each Join

4

More Data = More Searches
Slower Performance

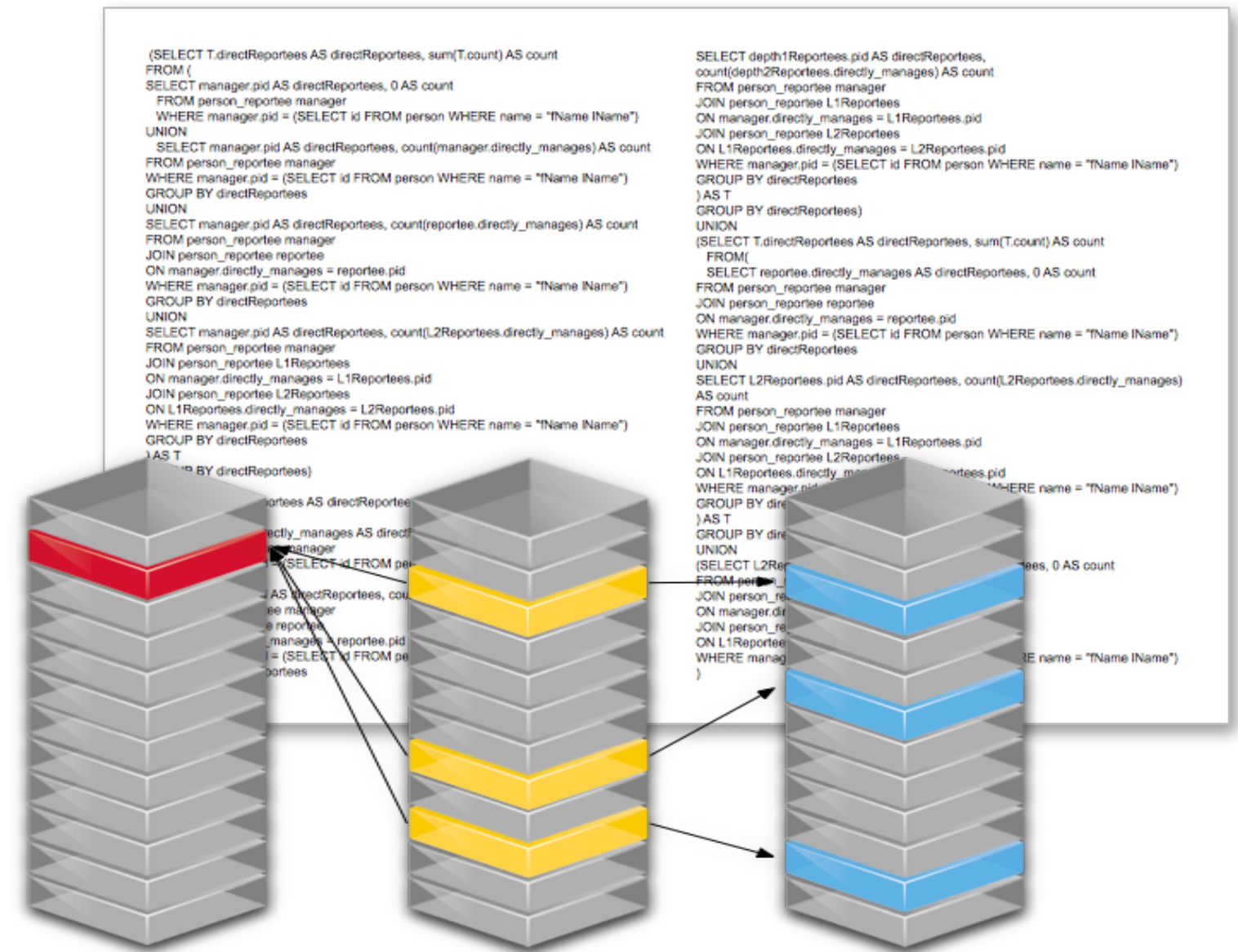
Relational Databases can't handle Relationships

1 Wrong Model
They cannot model or store relationships without complexity

2 Degraded Performance
Speed plummets as data grows and as the number of joins grows

3 Wrong Language
SQL was built with Set Theory in mind, not Graph Theory

4 Not Flexible
New types of data and relationships require schema redesign



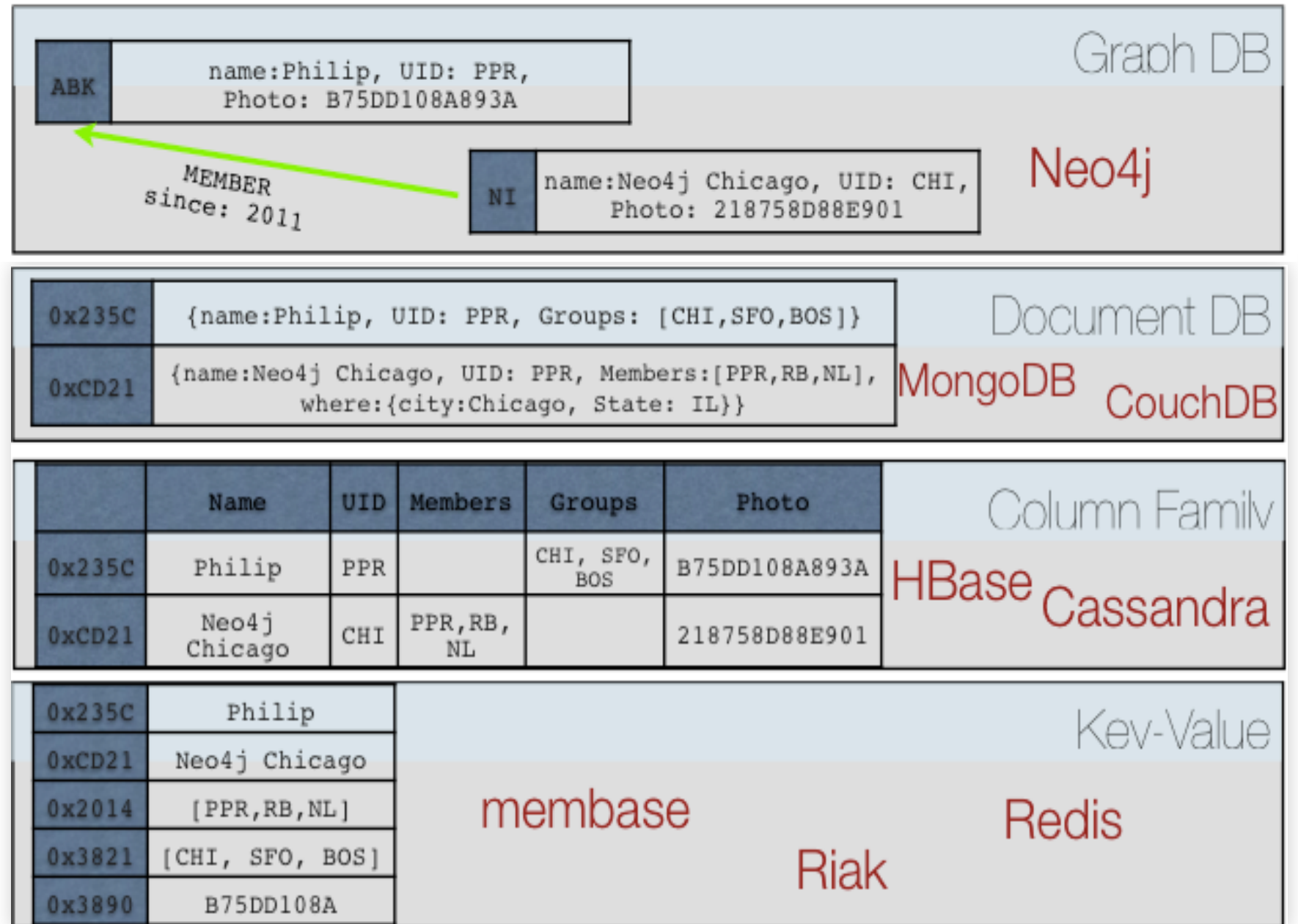
NoSQL Databases can't handle Relationships

1 Wrong Model
They cannot model or store relationships without complexity

2 Degraded Performance
Speed plummets as you try to join data together in the application

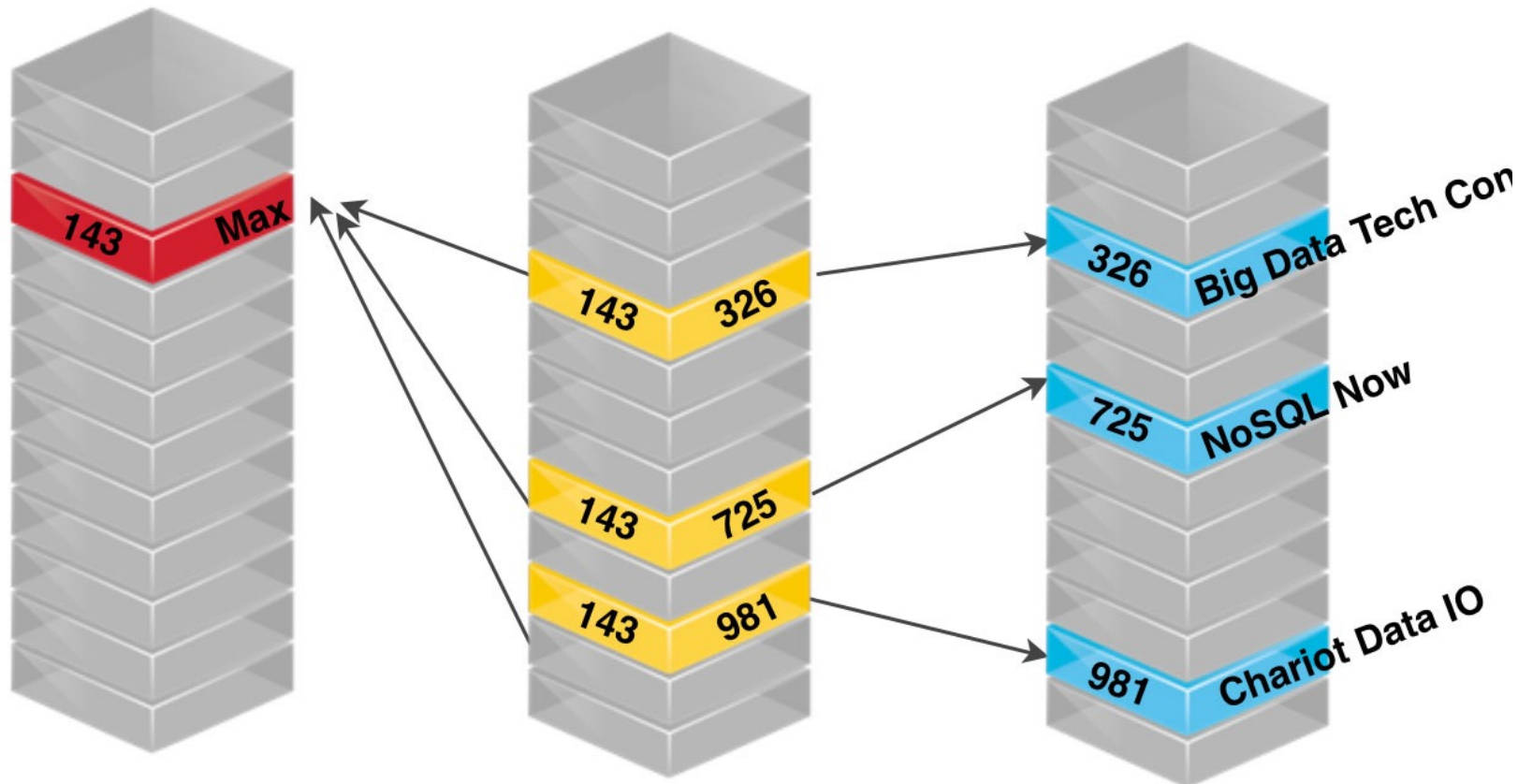
3 Wrong Languages
Lots of wacky "almost sql" languages terrible at "joins"

4 Not ACID
Eventually Consistent means Eventually Corrupt



Same Data, Different Layout

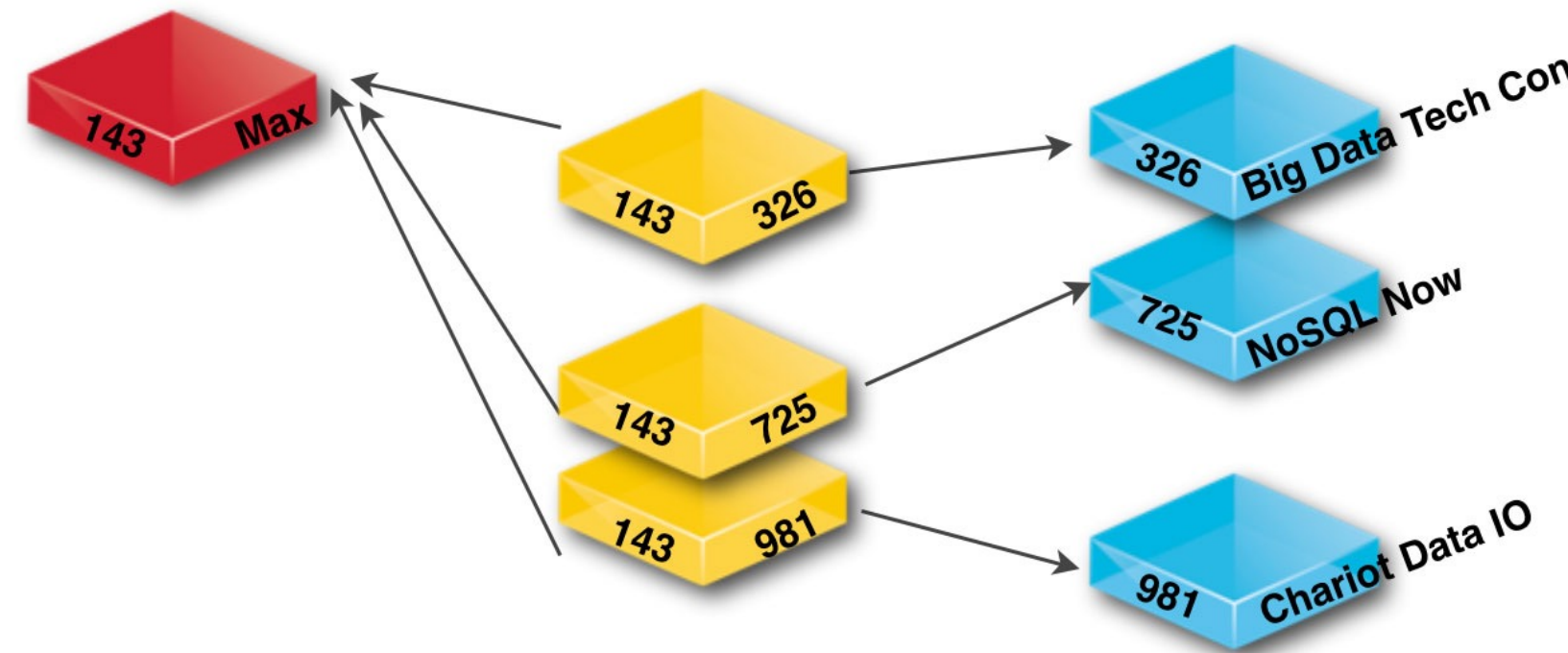
No more Tables, no more Foreign Keys, no more Joins



People

Attend

Conferences

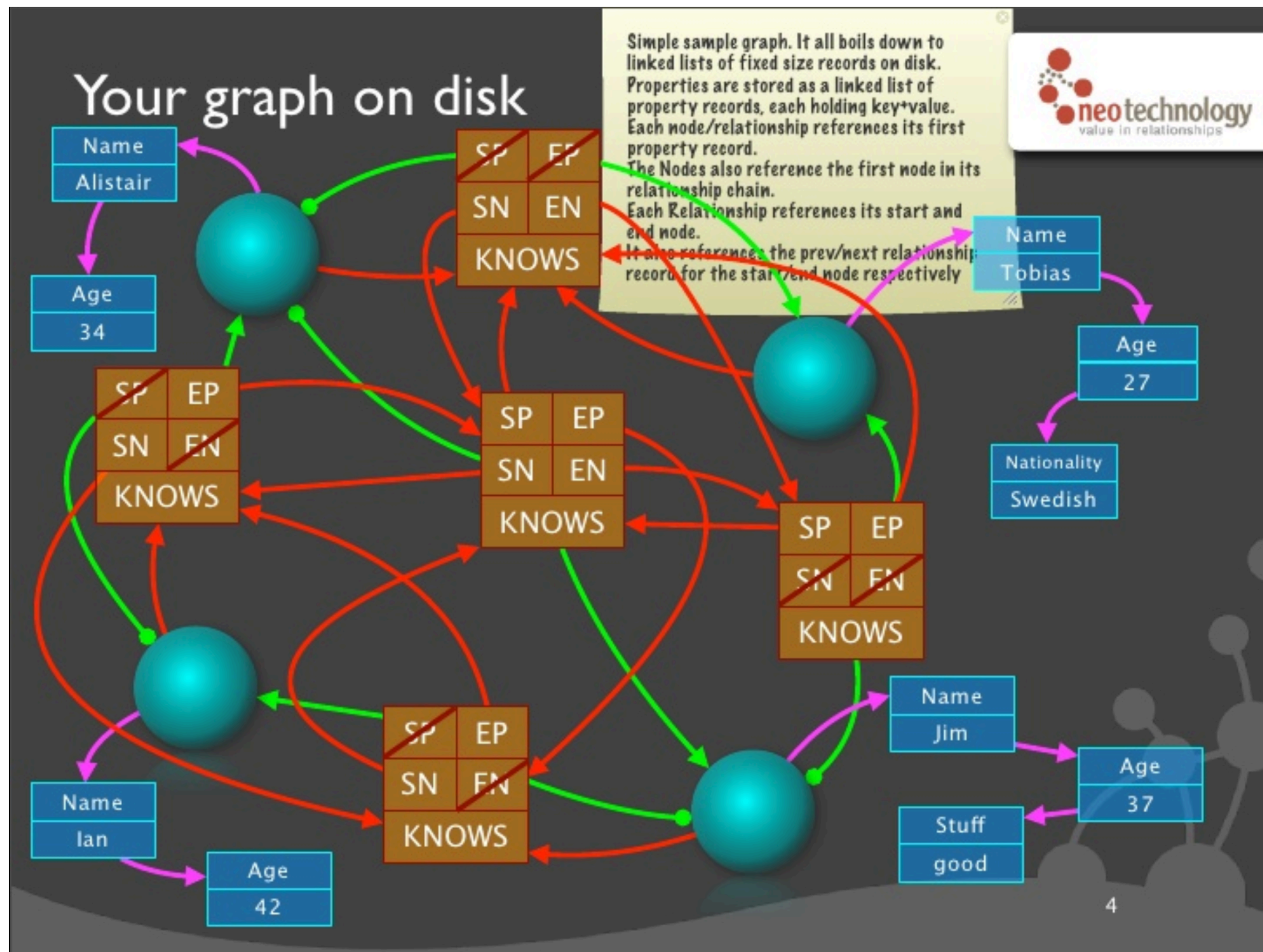


People

Attend

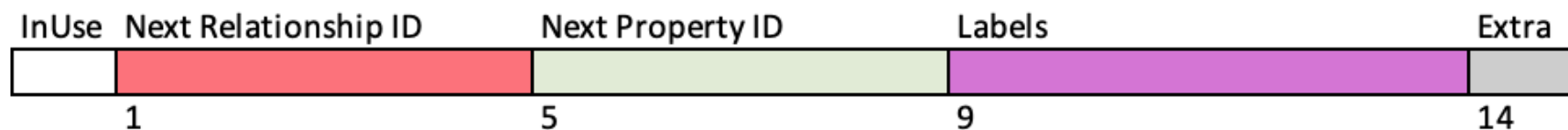
Conferences

Double Linked List Relationship Layout

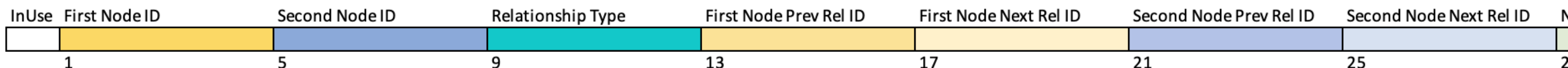


Neo4j Storage Record Layout

Node (15 Bytes)



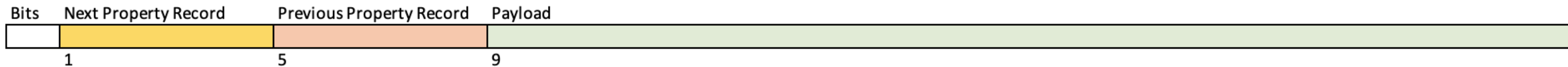
Relationship (34 Bytes)



RelationshipGroup (25 Bytes)




Property Record (41 Bytes)



Dynamic Record

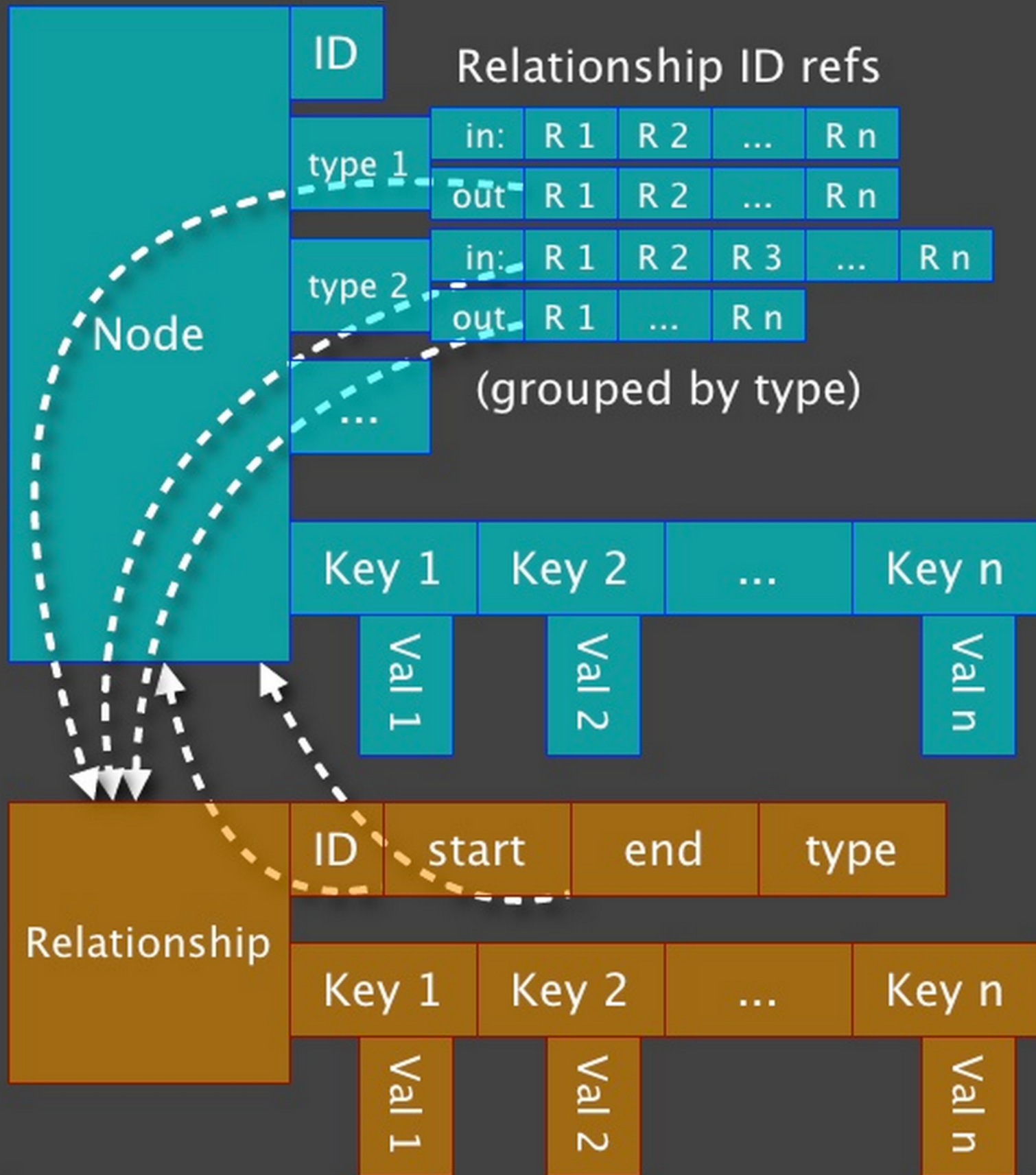




**The Most Important Slide
about Neo4j you will ever see**



What we put in cache



1

Pointers instead of Lookups

2

Fixed Sized Records

3

“Joins” on Creation

4

Spin Spin Spin through this data structure



The **S**ecretTM

Swedes

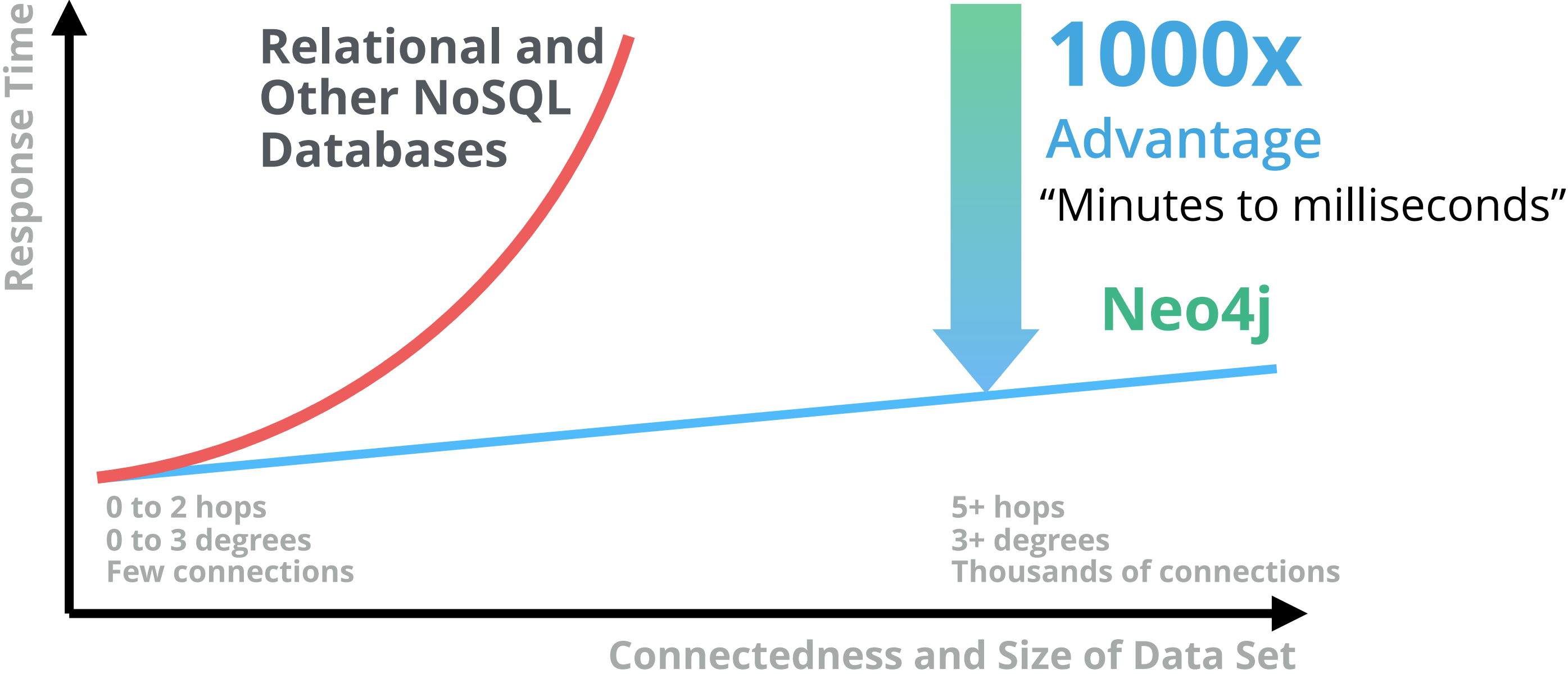


Partitions

Each Node's relationships are **partitioned** by type and direction.



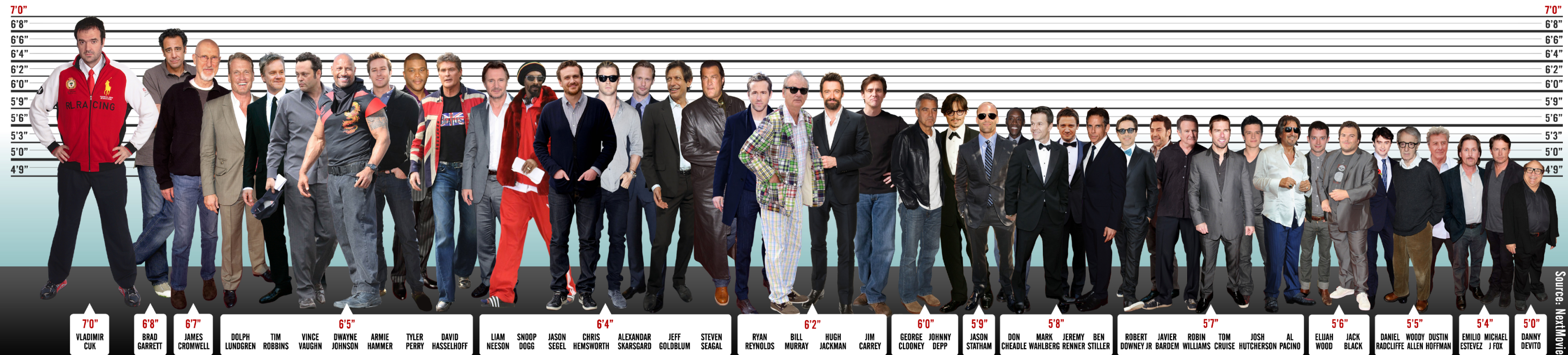
Real-Time Query Performance



But not for every query

I don't know the average height of all hollywood actors, but I do know the Six Degrees of Kevin Bacon

TALLEST ACTOR IN HOLLYWOOD



Source: NextMovie

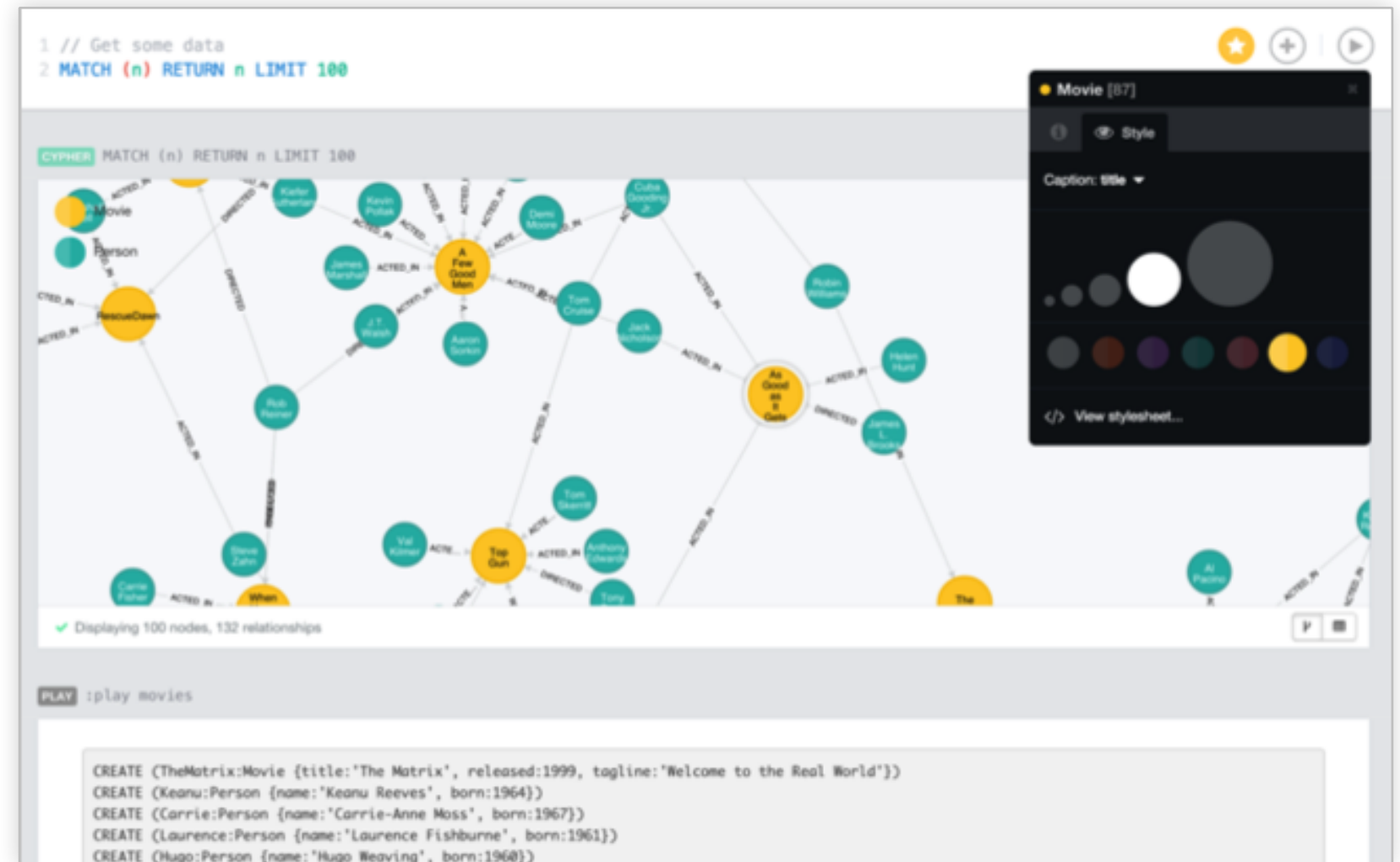
Reimagine your Data as a Graph

1 Right Model
Graphs simplify how you think

2 Better Performance
Query relationships in real time

3 Right Language
Cypher was purpose built for
Graphs

4 Flexible and Consistent
Evolve your schema seamlessly while
keeping transactions



**Agile, High Performance
and Scalable without Sacrifice**

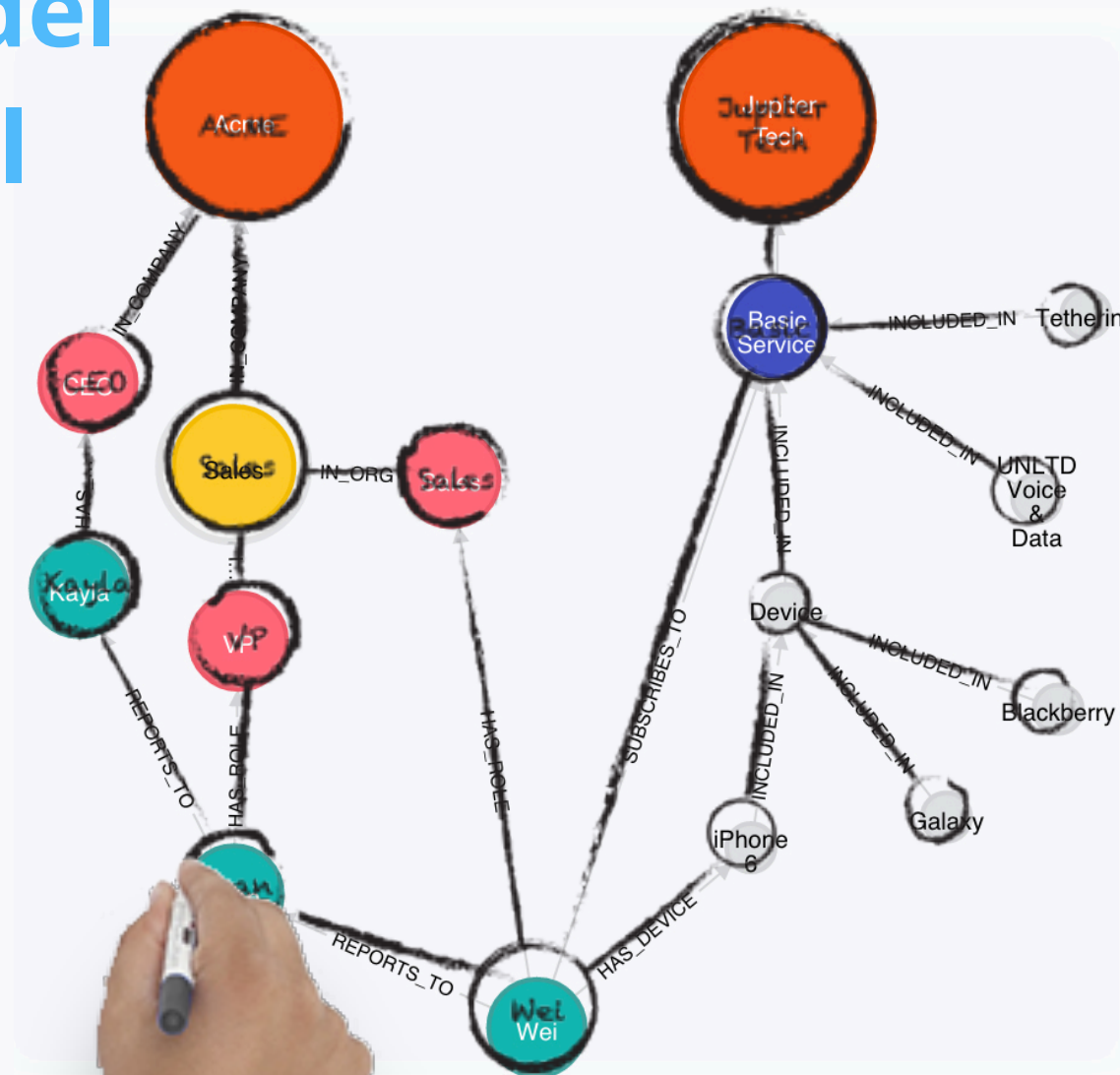
A network graph with various sized nodes and connecting lines, set against a blue-to-yellow gradient background. The nodes are represented by circles of different diameters, and they are interconnected by thin lines, forming a complex web structure. The largest node is at the bottom center, with many lines radiating outwards to other nodes of varying sizes scattered across the frame.

Basics of Modeling



Neo4j Property Graph

The Whiteboard Model
is the Physical Model



A unified view for
ultimate agility

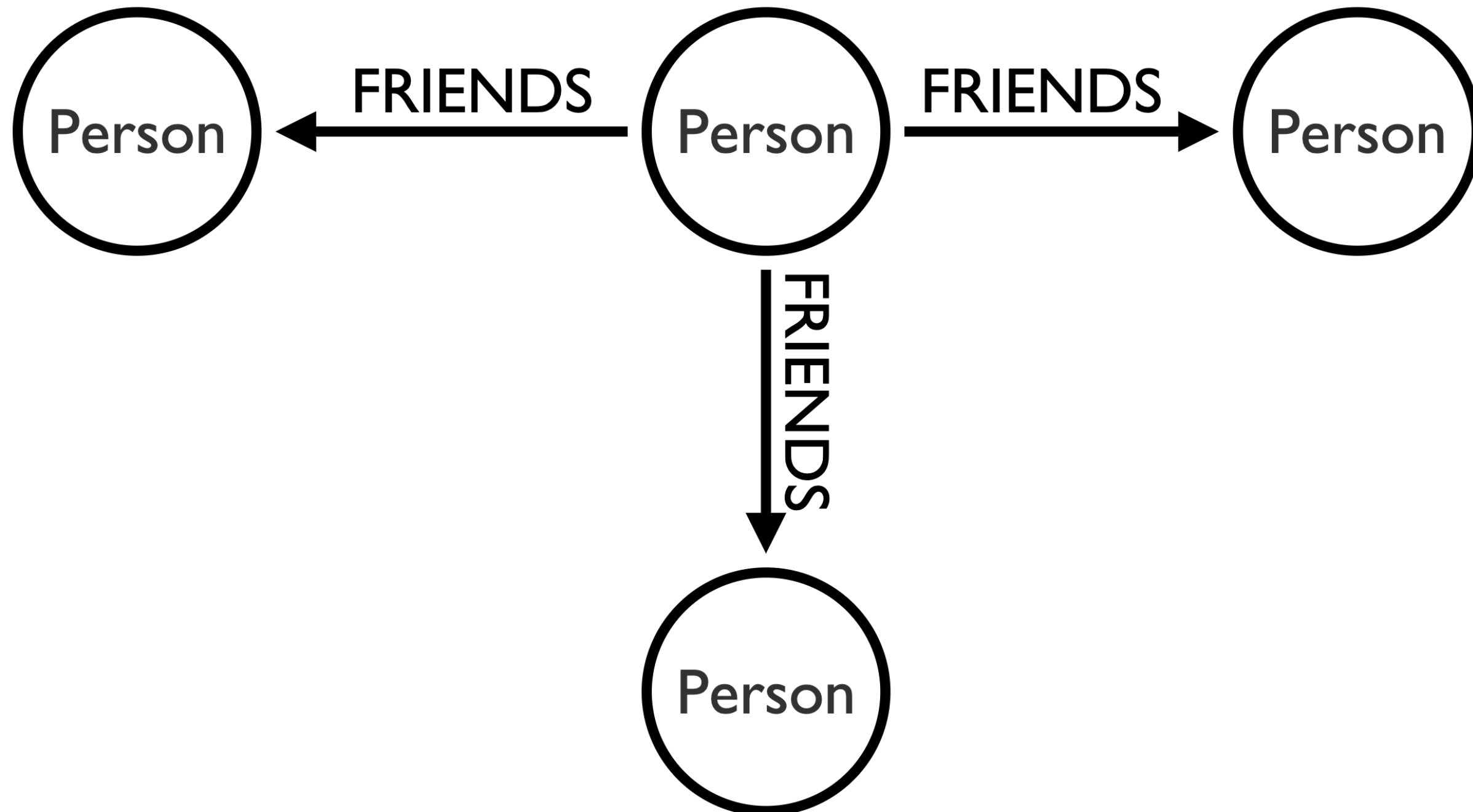
- *Easily understood*
- *Easily evolved*
- *Easy collaboration between business and IT*

Arrows

<http://www.apcjones.com/arrows/#>

It's simple, it's fast, it doesn't run out of ink.

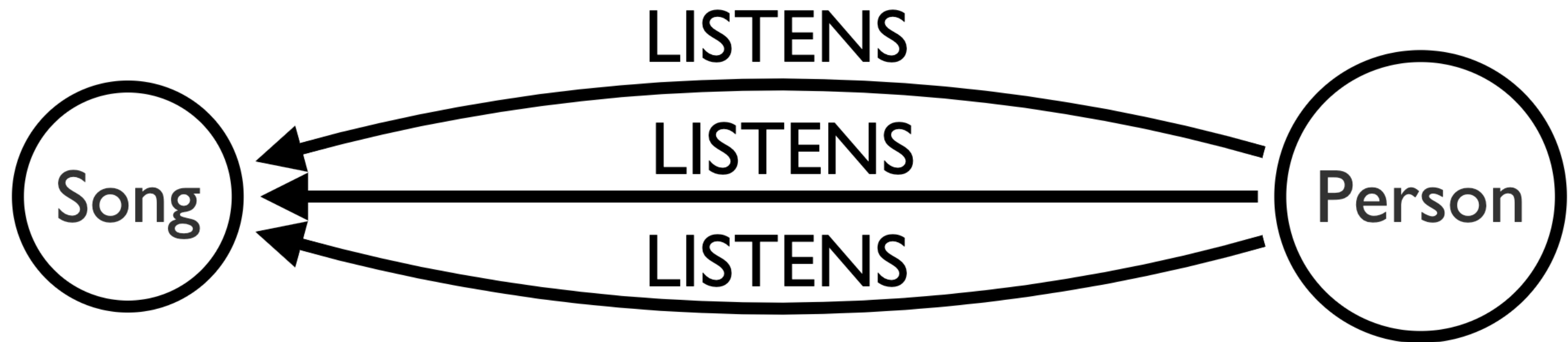
Nodes can have many relationships



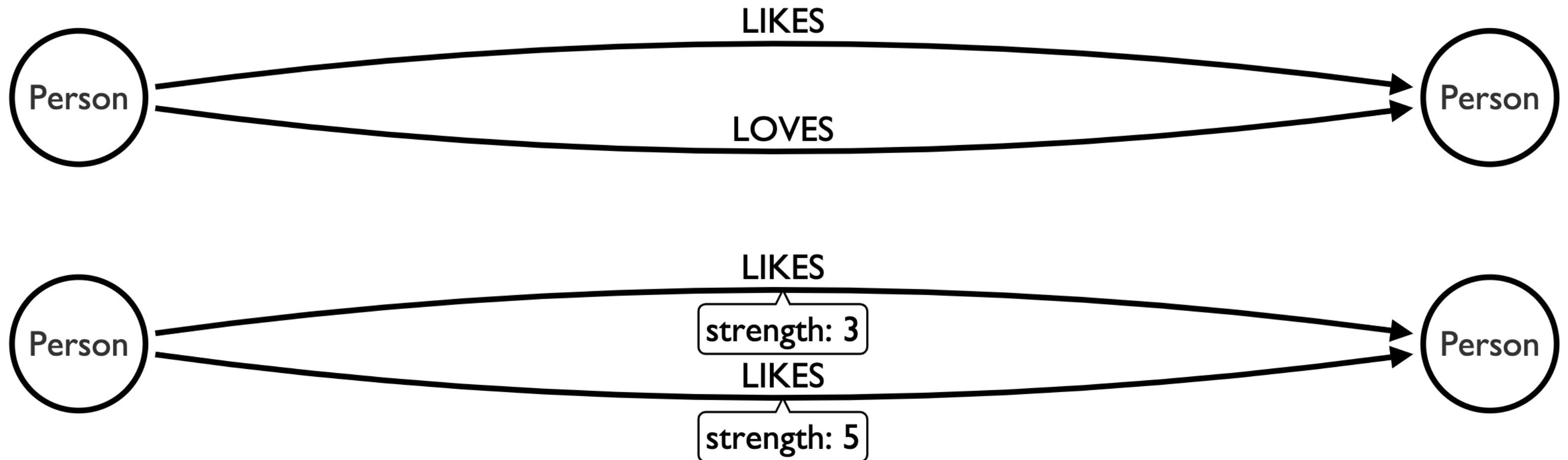
Two Nodes can be related by more than one relationship type



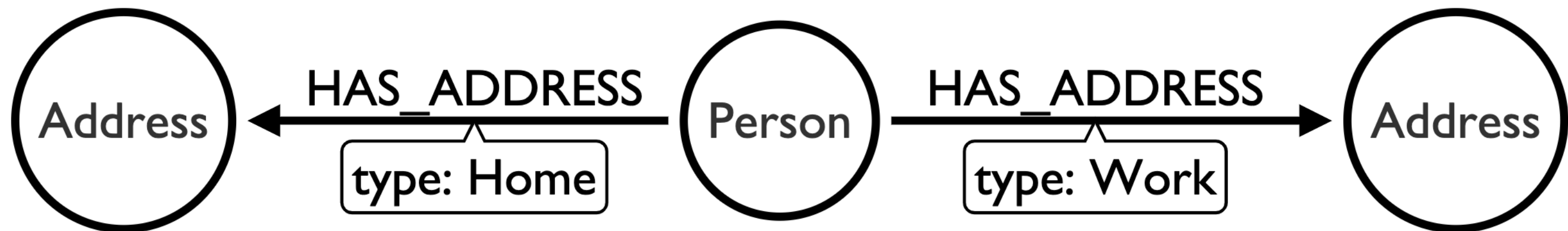
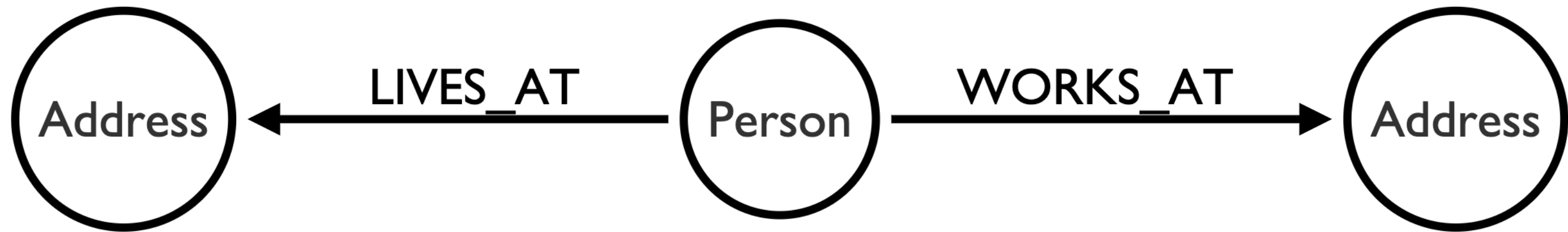
Two Nodes can be related by the same relationship type more than once



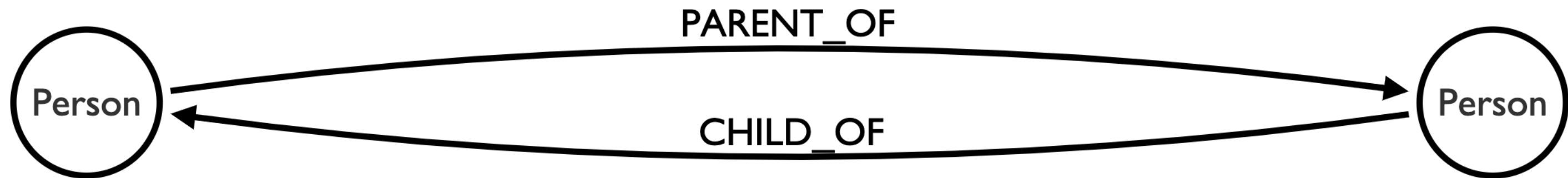
Relationships can have properties



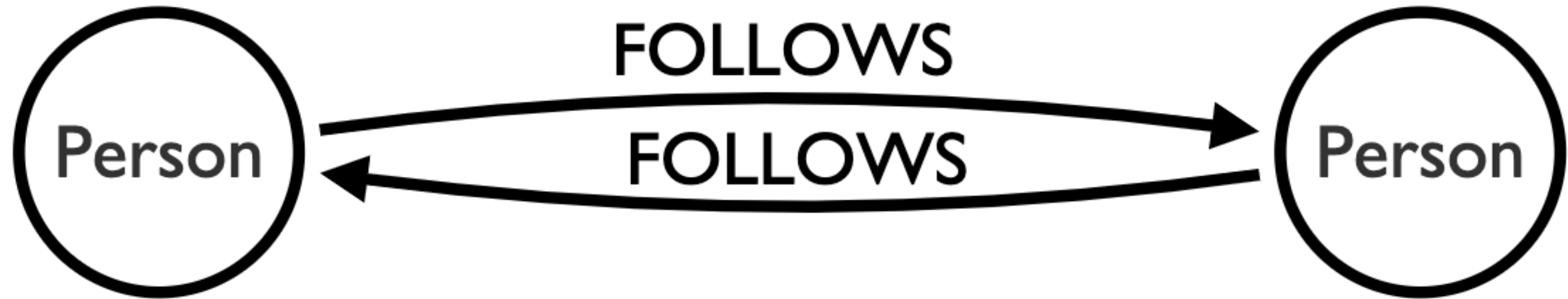
Don't overcomplicate your model



Pick one or the other, not both



...but this is fine:





Modeling Acting

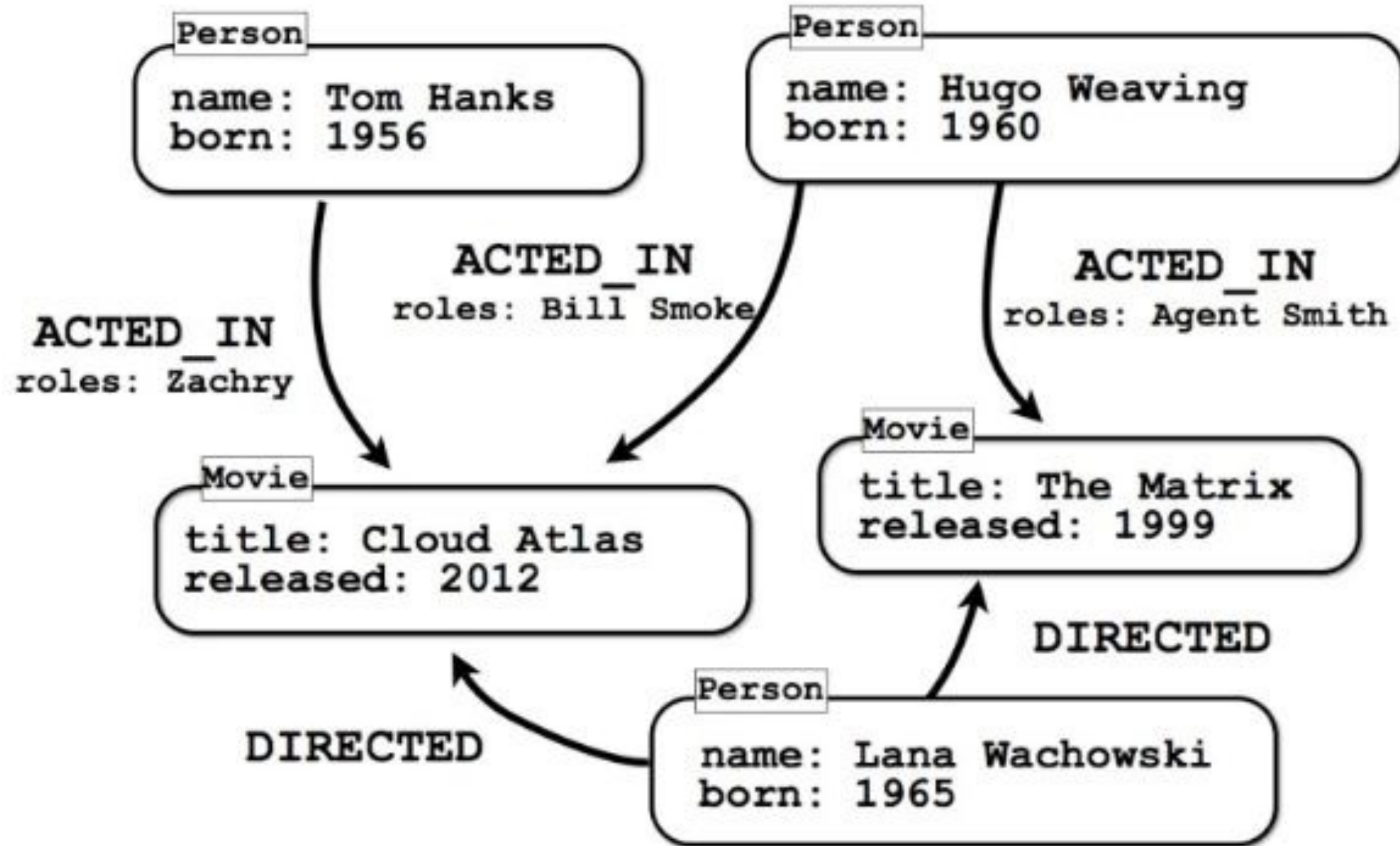
Graphs are Whiteboard Friendly

Just draw stuff and "walla" there is your data model



Some Models are Easy

Movie Property Graph

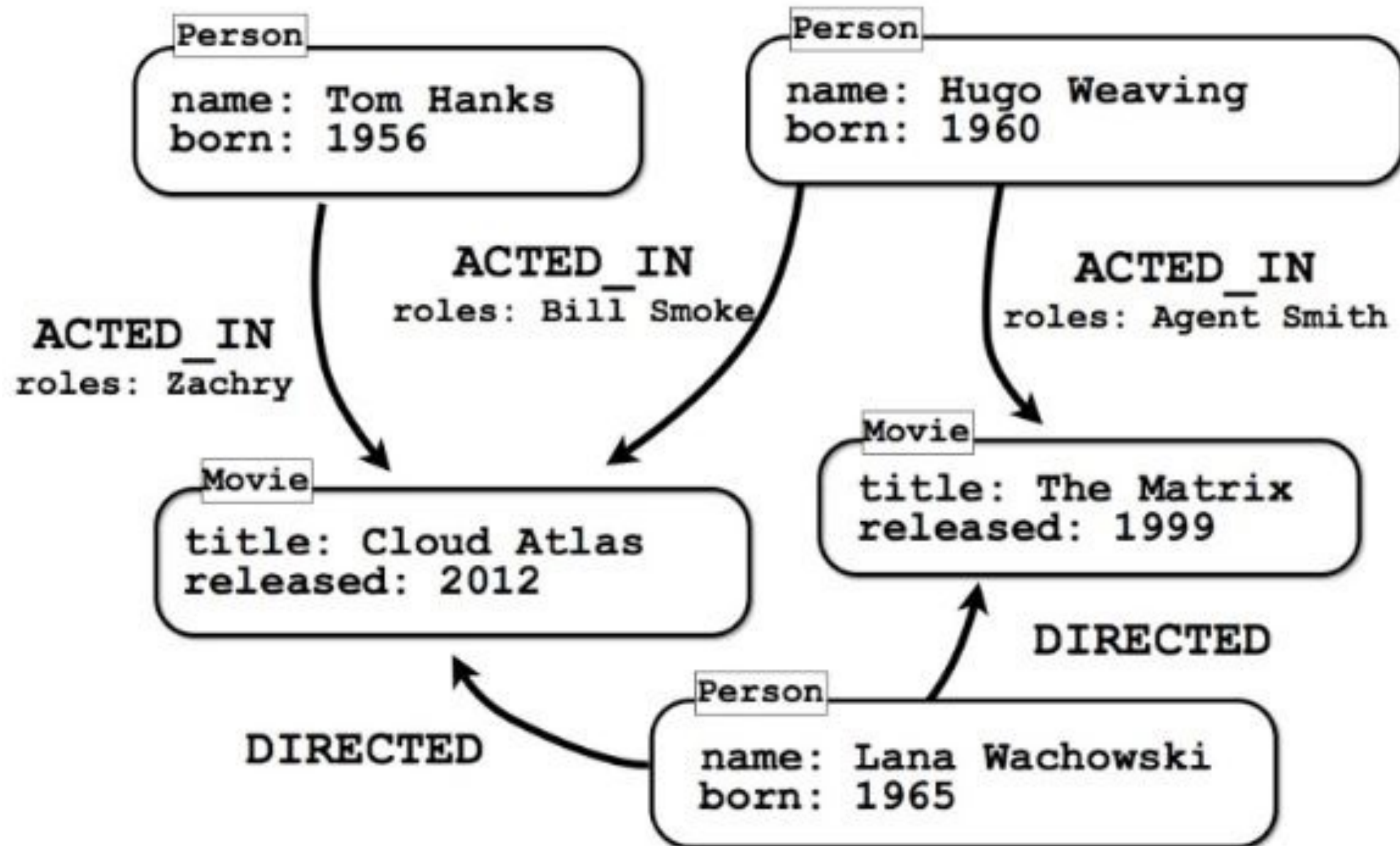




GLOBAL
JAMES BOND DAY

Some Models are Easy but not for all Questions

Should Roles be their own Node?

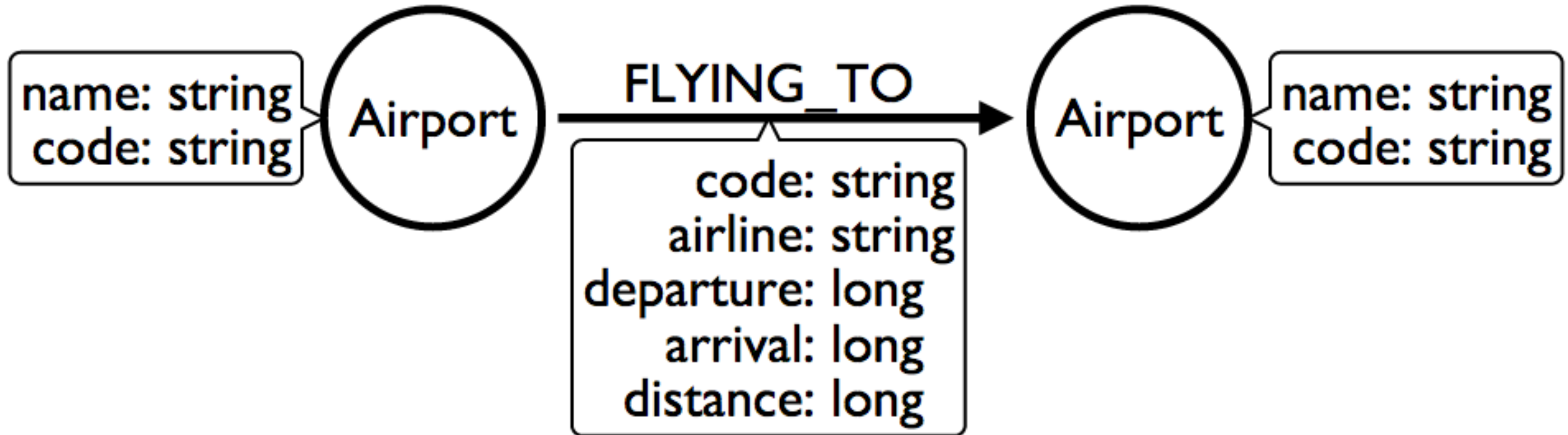




Modeling Flights

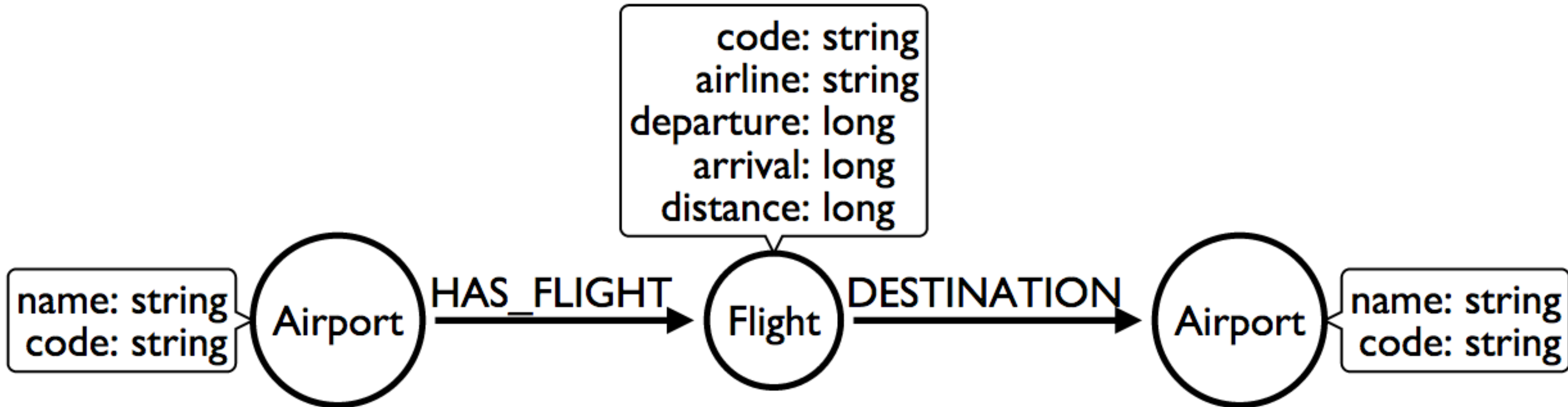
How do you model Flight Data?

Airports Nodes with Flying To Relationships



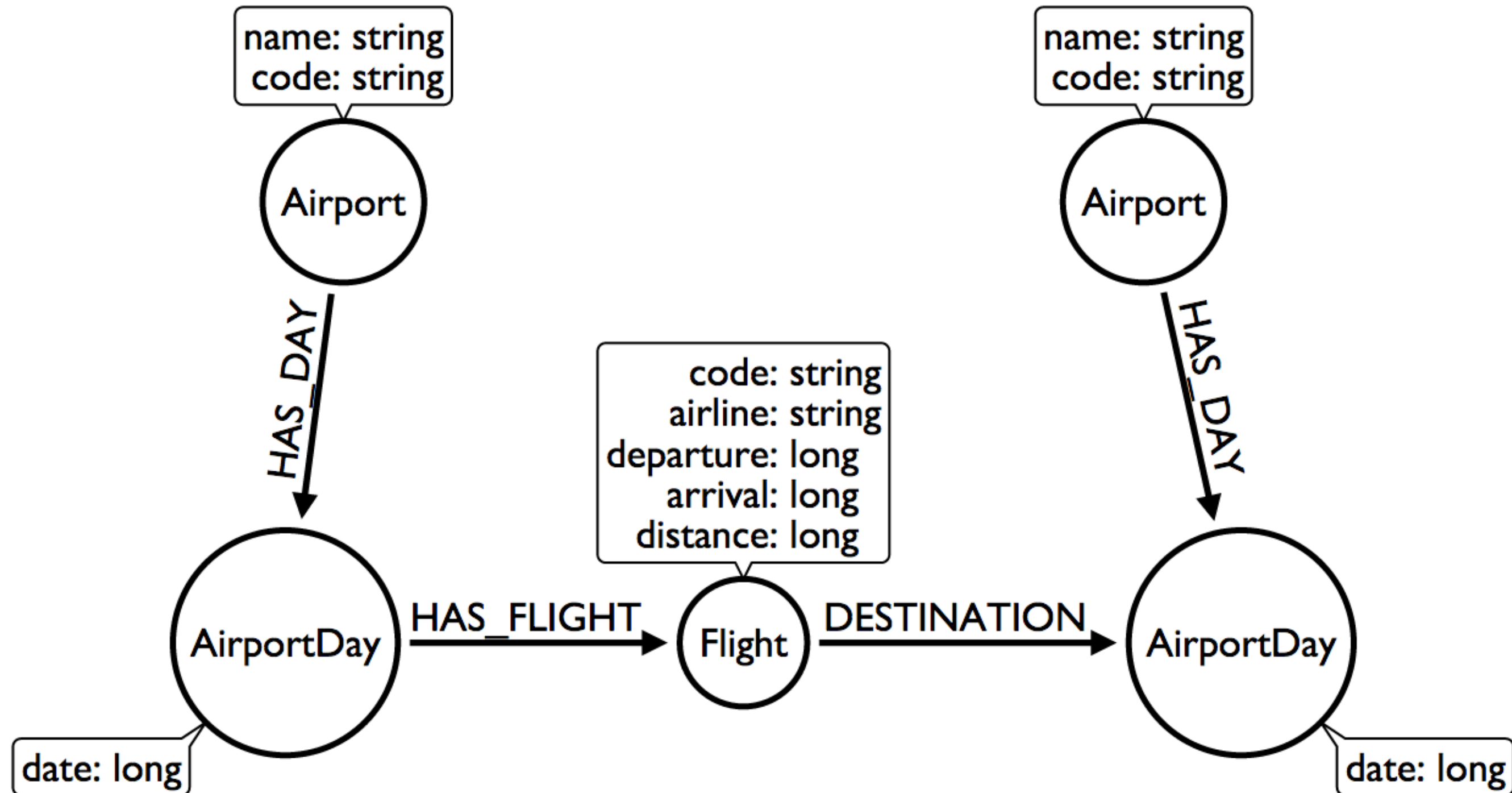
How do you model Flight Data?

Maybe Flight should be its own Node?



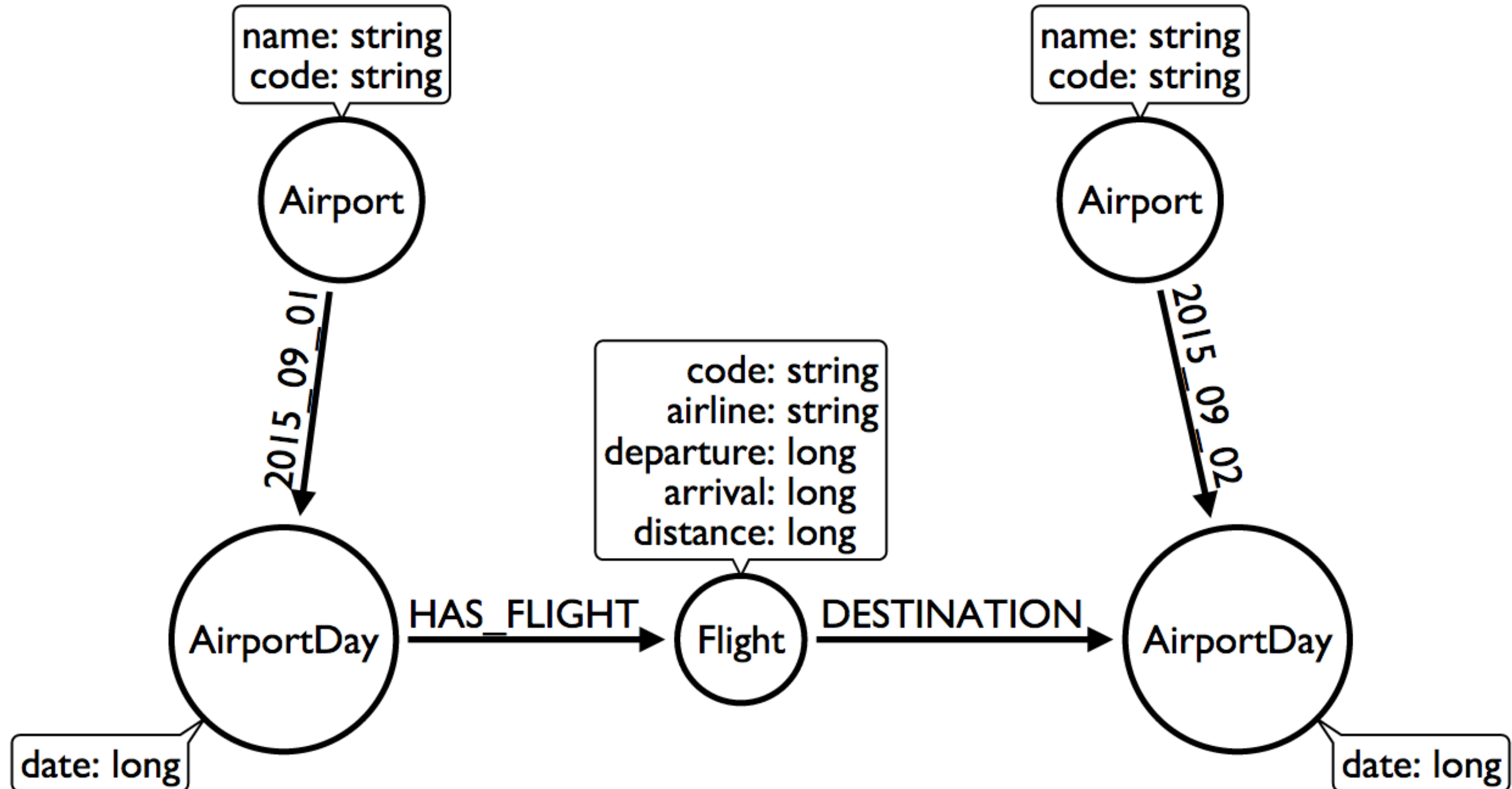
How do you model Flight Data?

Don't we care about Flights only on particular Days?



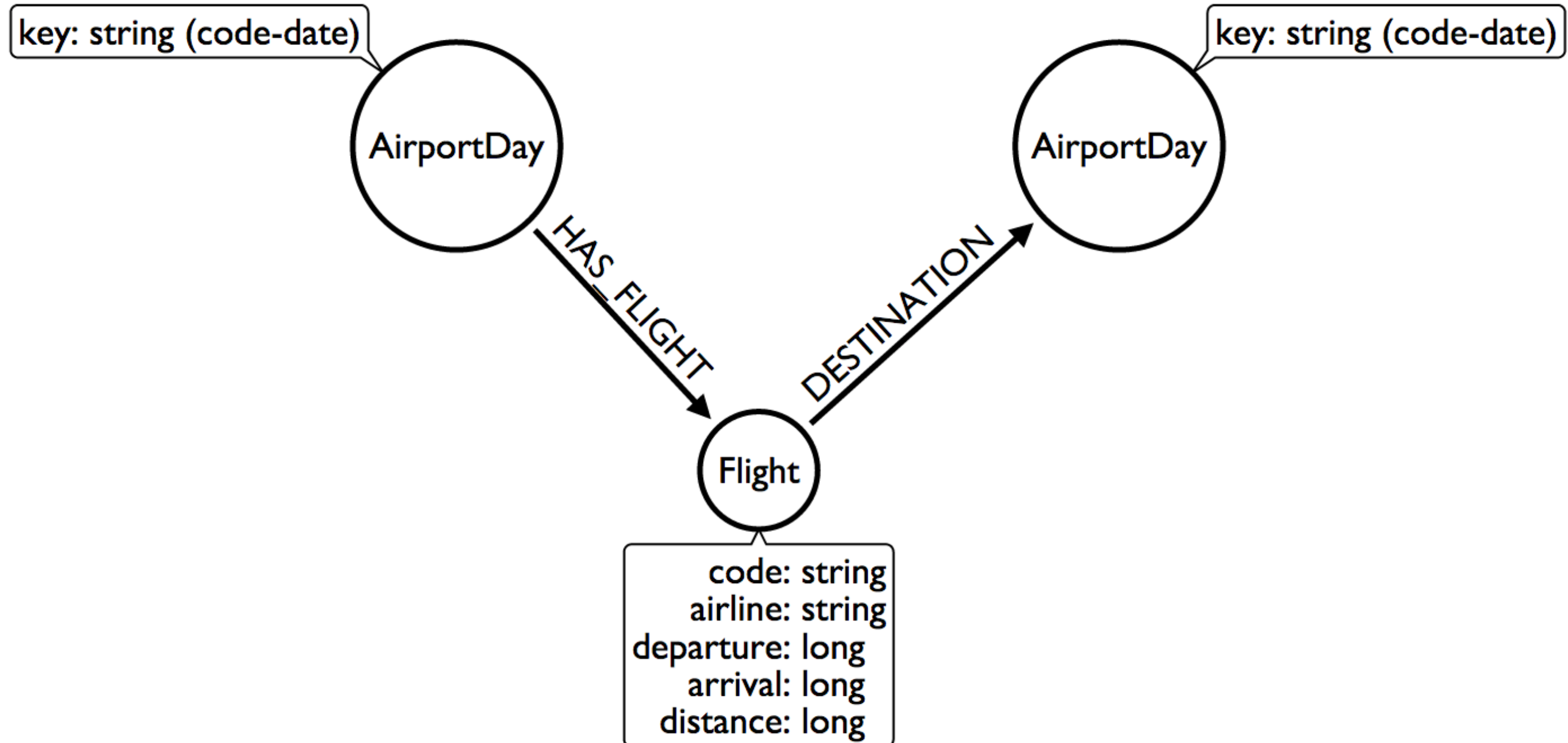
How do you model Flight Data?

What is this trick with the date in the relationship type?



How do you model Flight Data?

We don't need Airports if we model this way!

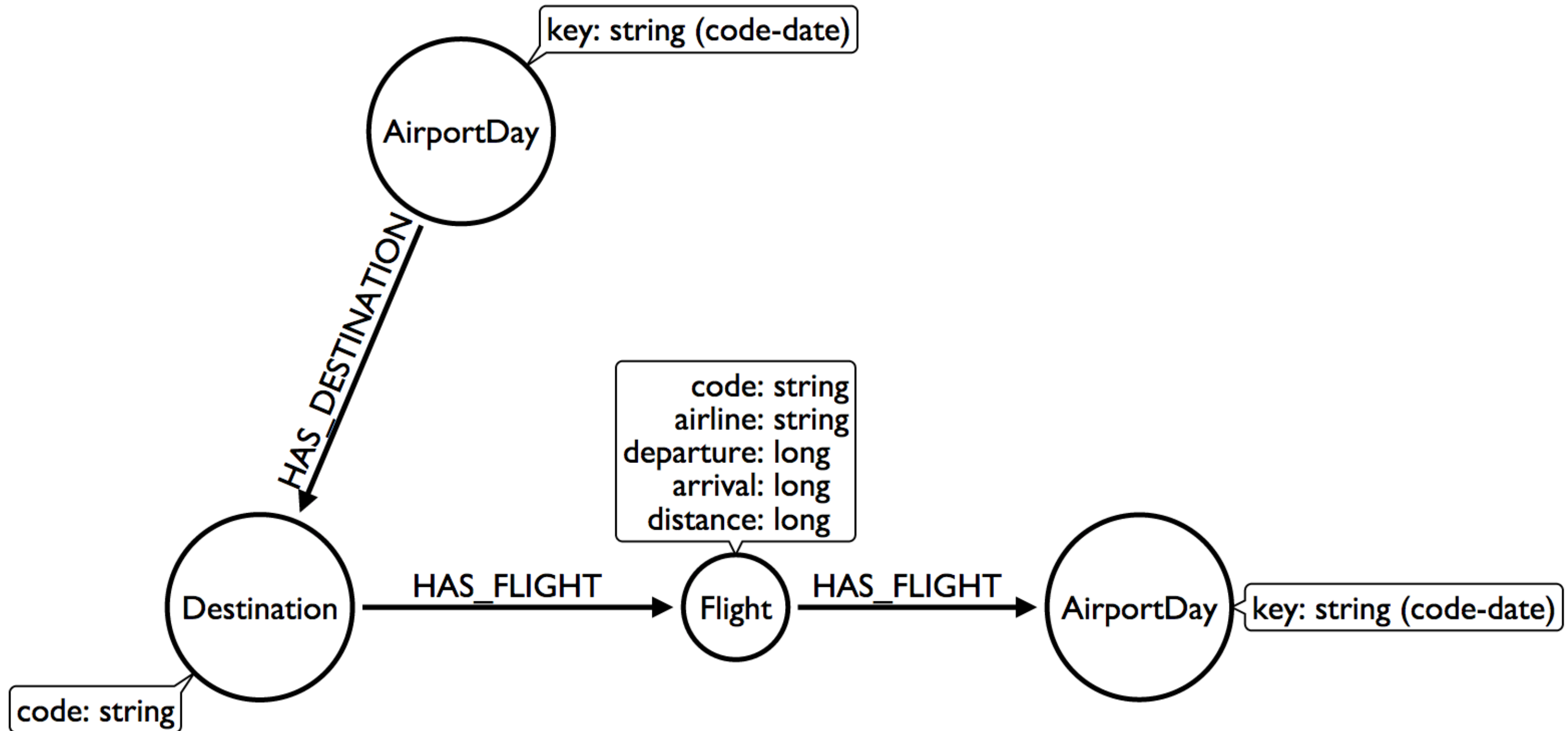


Lets get Creative



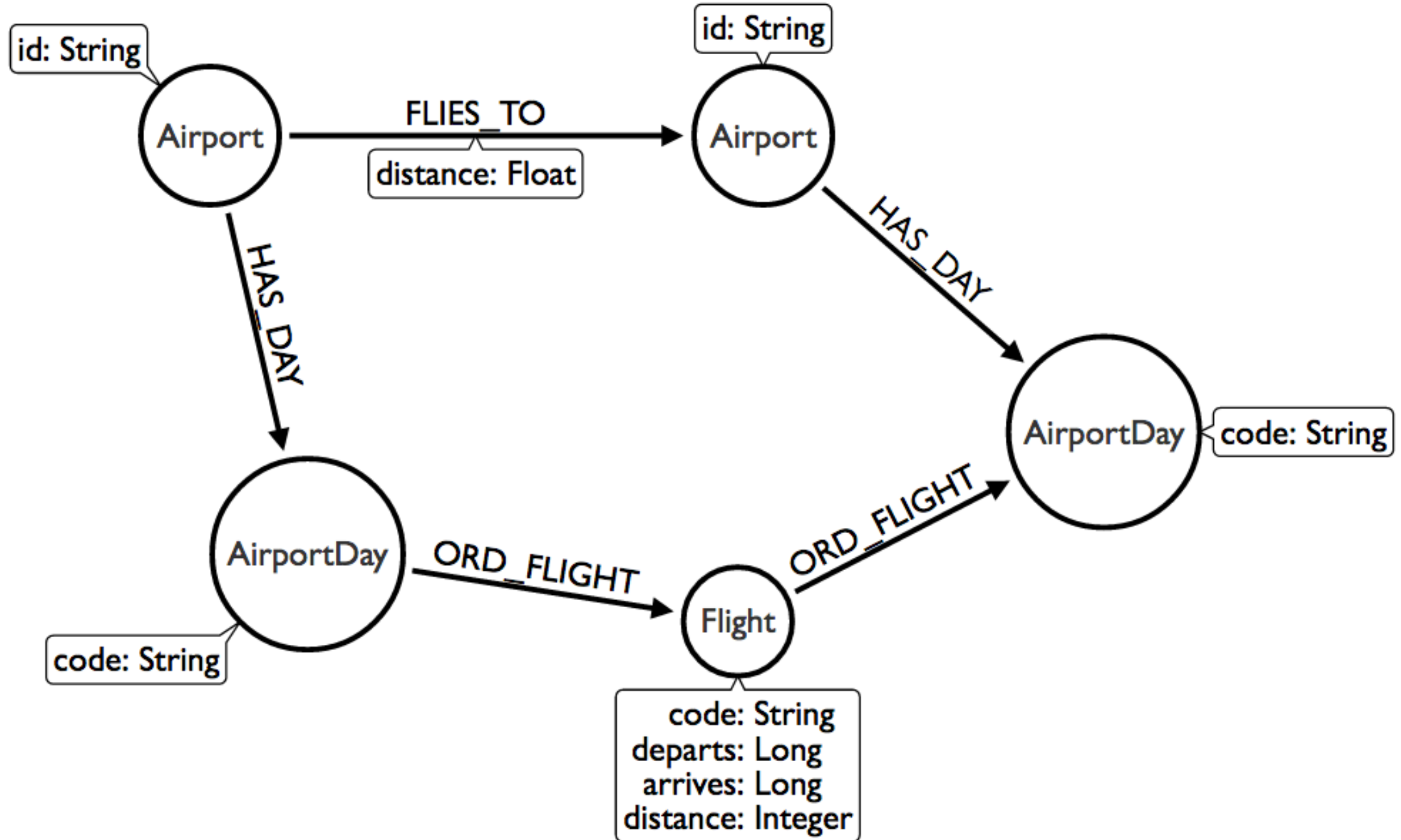
How do you model Flight Data?

Group Destinations together!

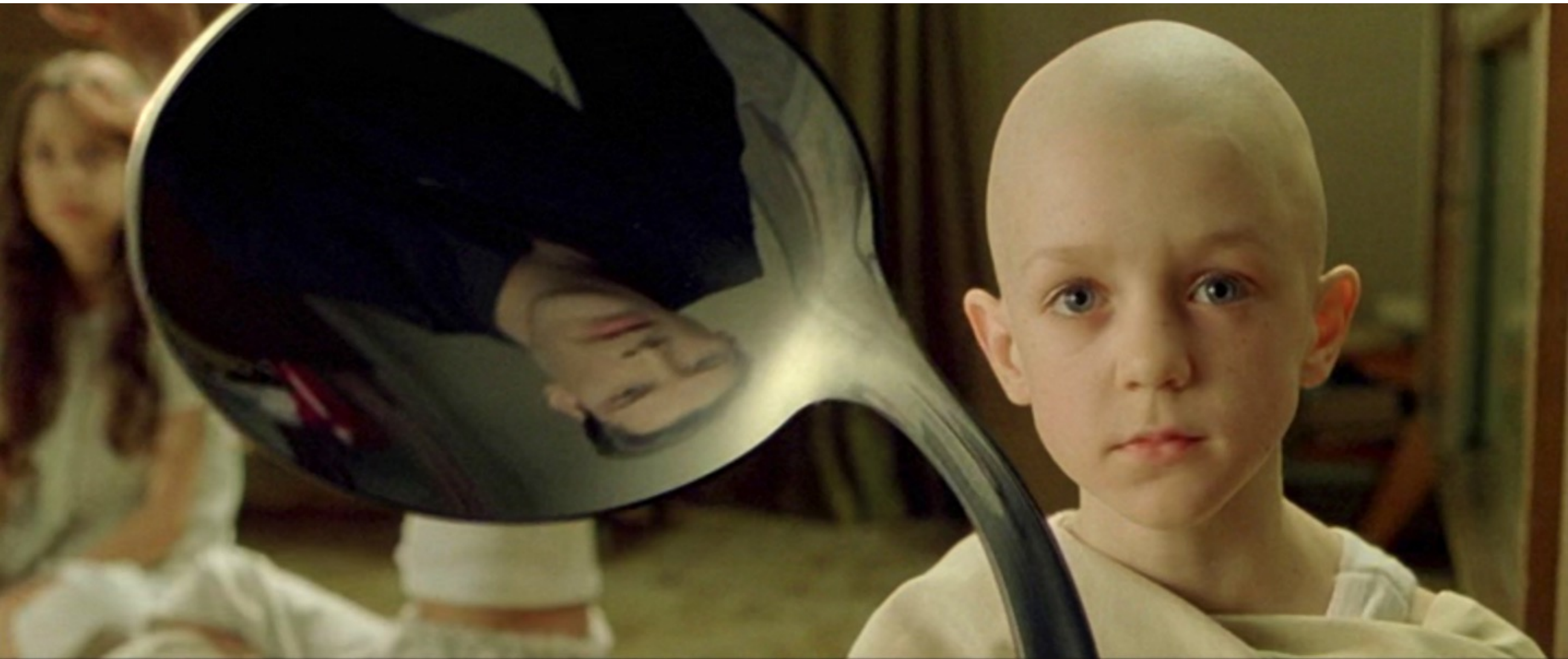


How do you model Flight Data?

OMG WAT!



Do *not* try and bend the data. That's *impossible*.

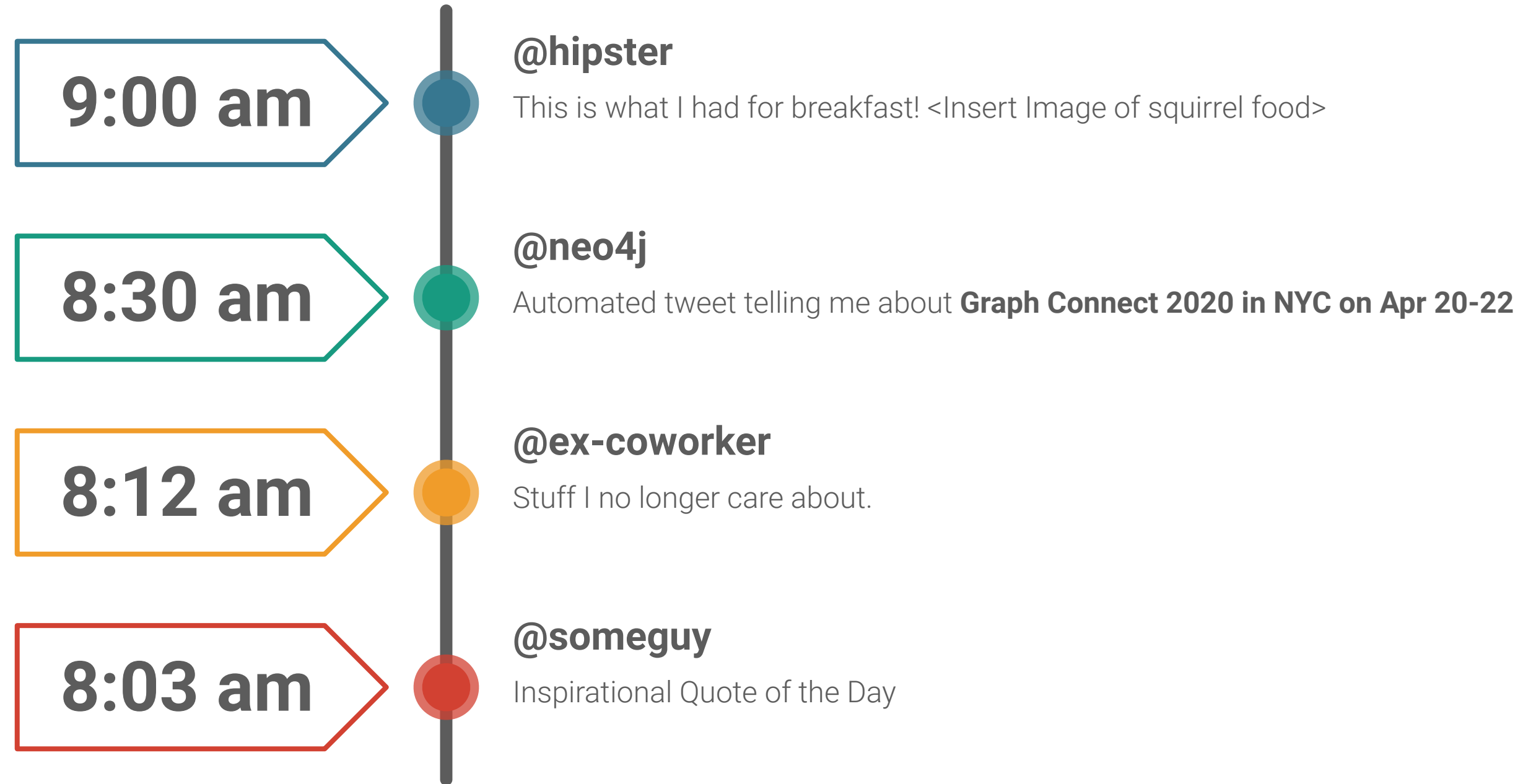




Modeling Twitter

Building a News Feed

Cloning Twitter



Cloning Twitter

How do others do it?

“After we create a post and we obtain the post ID, we need to LPUSH the ID in the timeline of every user that is following the author of the post” from [Tutorial: Design and implementation of a simple Twitter clone using PHP and the Redis key-value store](#)

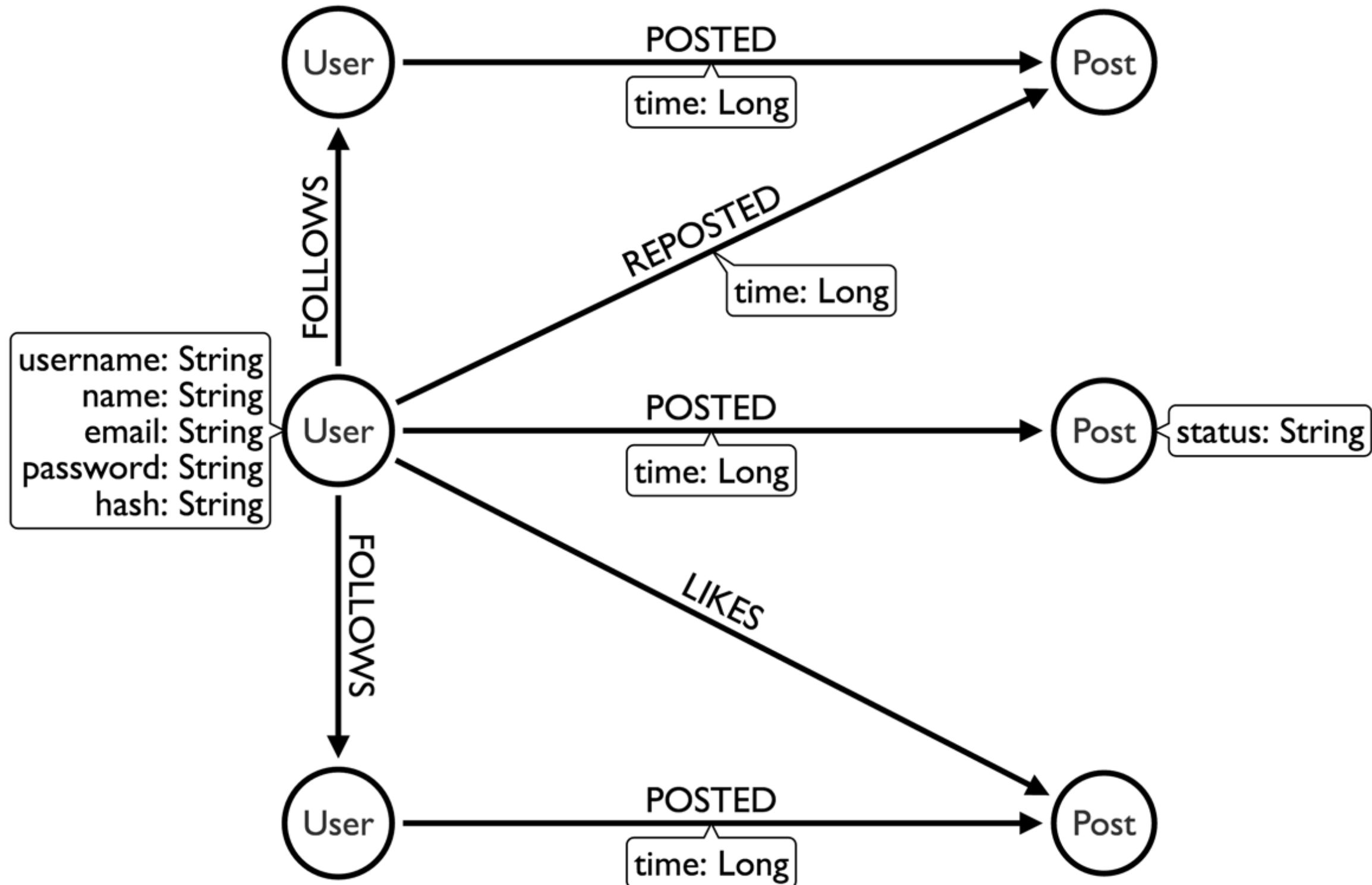
```
1 foreach($followers as $fid) {
2     $r->lpush("posts:$fid",$postid);
3 }
```

Another example using the document store [Firebase](#) instead uses the same approach: “When a new spark is posted, we’ll [put] it in the global list, and then put its ID in the feed of every user that is following the author.” from [An open source Twitter clone built with Firebase](#)

```
1 // Add spark ID to the feed of everyone following this user.
2 currentUser.child("followers").once("value", function(list) {
3     list.forEach(function(follower) {
4         var childRef = firebase.child("users").child(follower.name());
5         childRef.child("feed").child(sparkRefId).set(true);
6     });
7 });
```

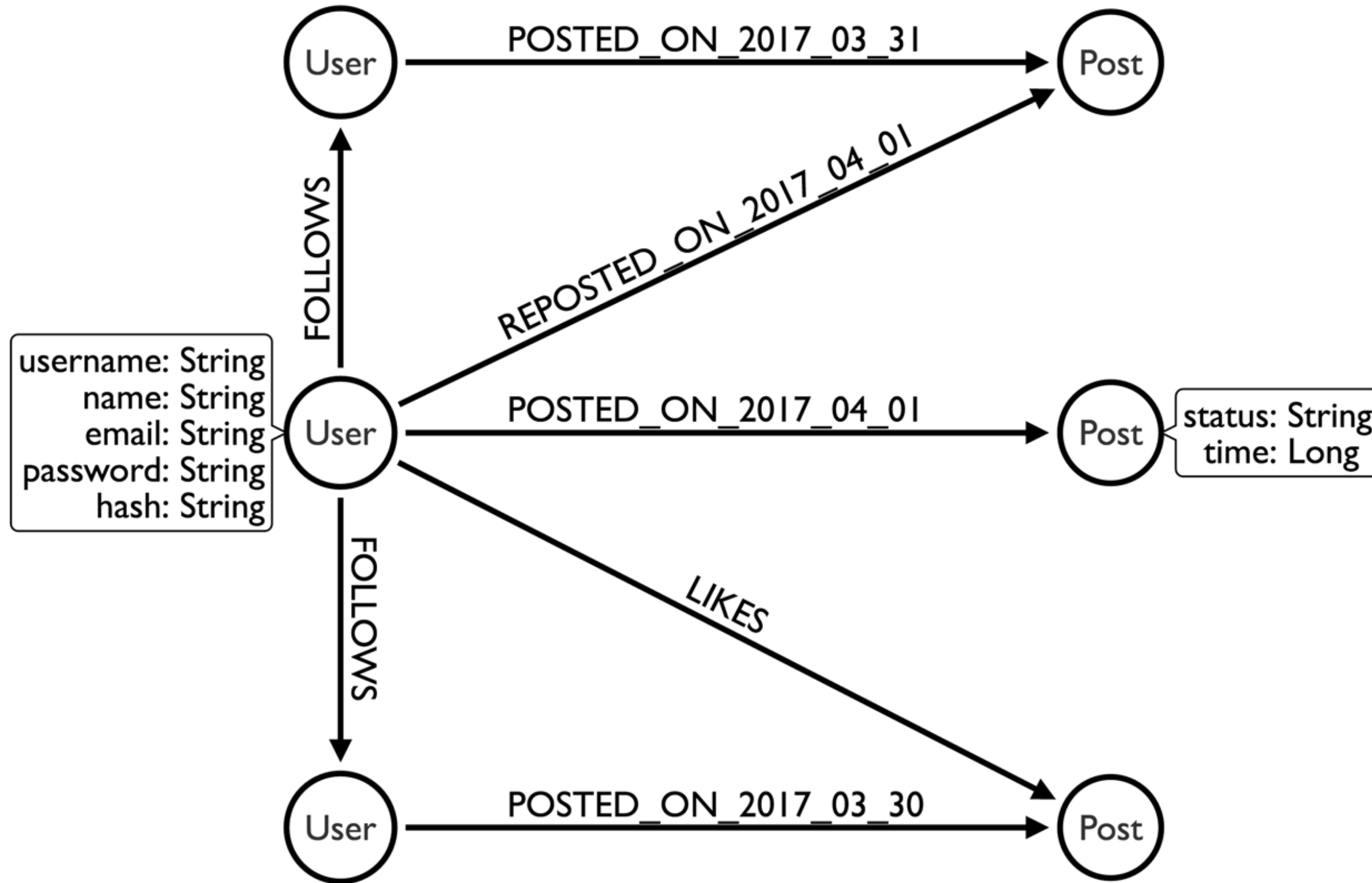

Modeling a Twitter Feed

The Wrong Way

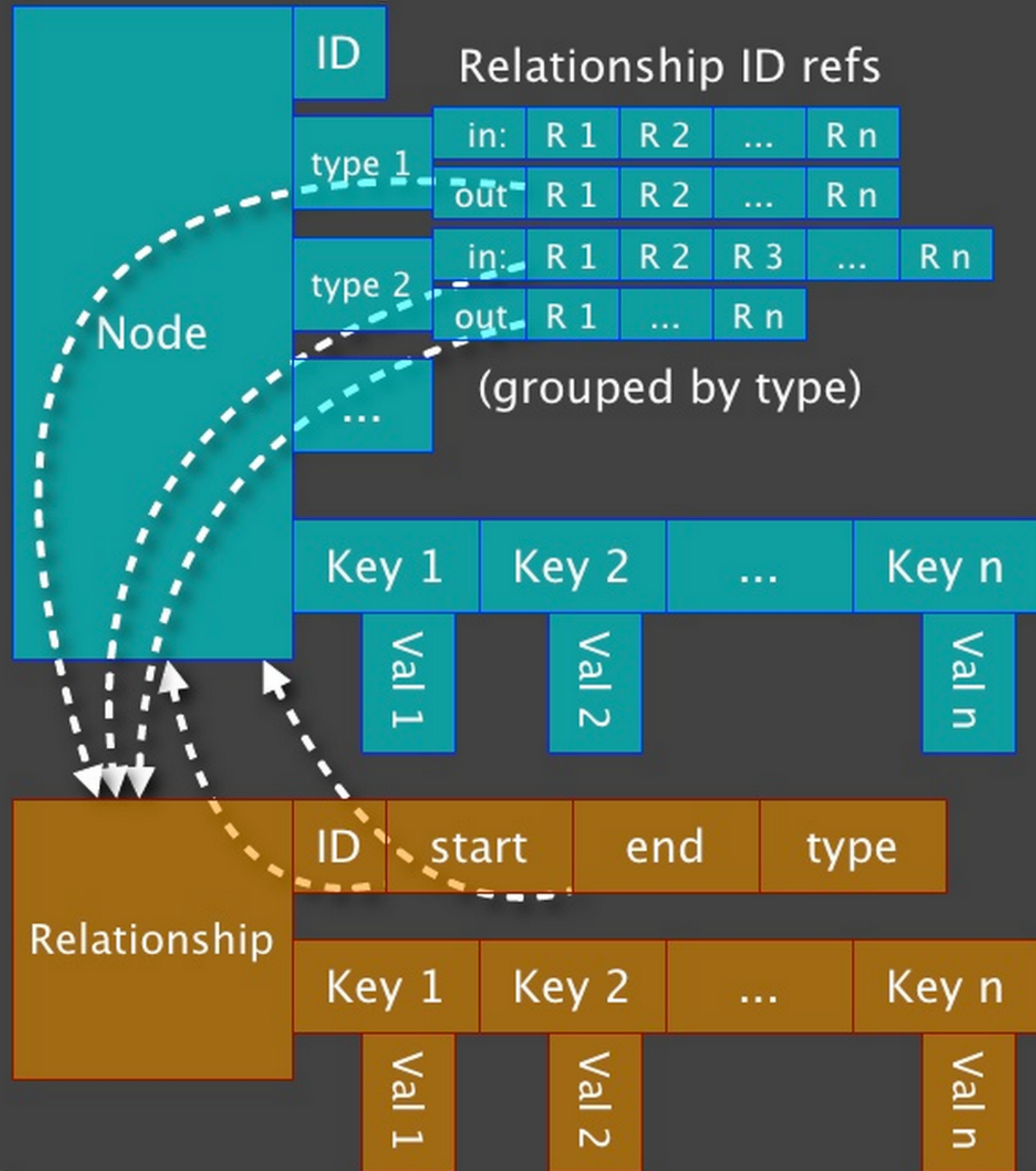


Modeling a Twitter Feed

A Better Way



What we put in cache



Neo4j Secret Sauce Yet Again

- 1 Pointers instead of Lookups
- 2 Fixed Sized Records
- 3 "Joins" on Creation
- 4 Spin Spin Spin through this data structure

Make the Queries Scale

The screenshot shows a website header with the text "Showcase & Discover Creative Work". Below the header is a navigation bar with links: "Best of Behance", "Curated Galleries", "Creative Tools", and "Schools & Organizations". The main content area is titled "Activity Feed:" and lists three entries:

Year	Project/Tool	Resources	Storage
2011	Mongo: FANTASTIC BEASTS	125 Servers	20 TB
2015	Cassandra: Seekers Base	48 Servers	50 TB
2018	Neo4j: Max Kostenko	3 Servers	33 GB

...and the database scales with them.
...and that's why **we don't make any money.**

SCALING OUT IS IN **FASHION**

But when your **model** and your **query**
match you don't have to.





Modeling Forms

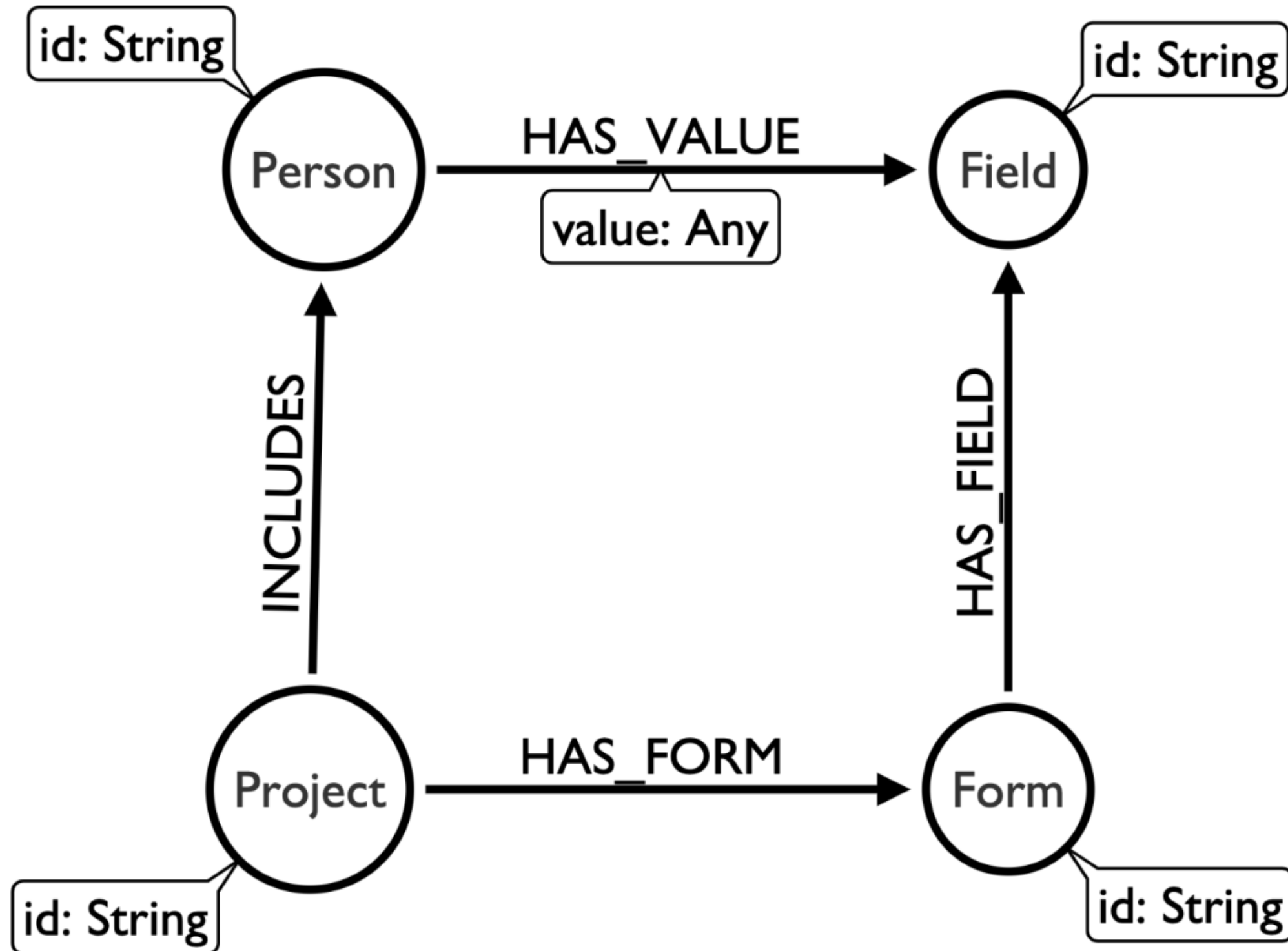
Fluid Schema



How many **arms** does the T1000 have in this picture?

Any Form

For all forms, all fields, all values you'll ever need.

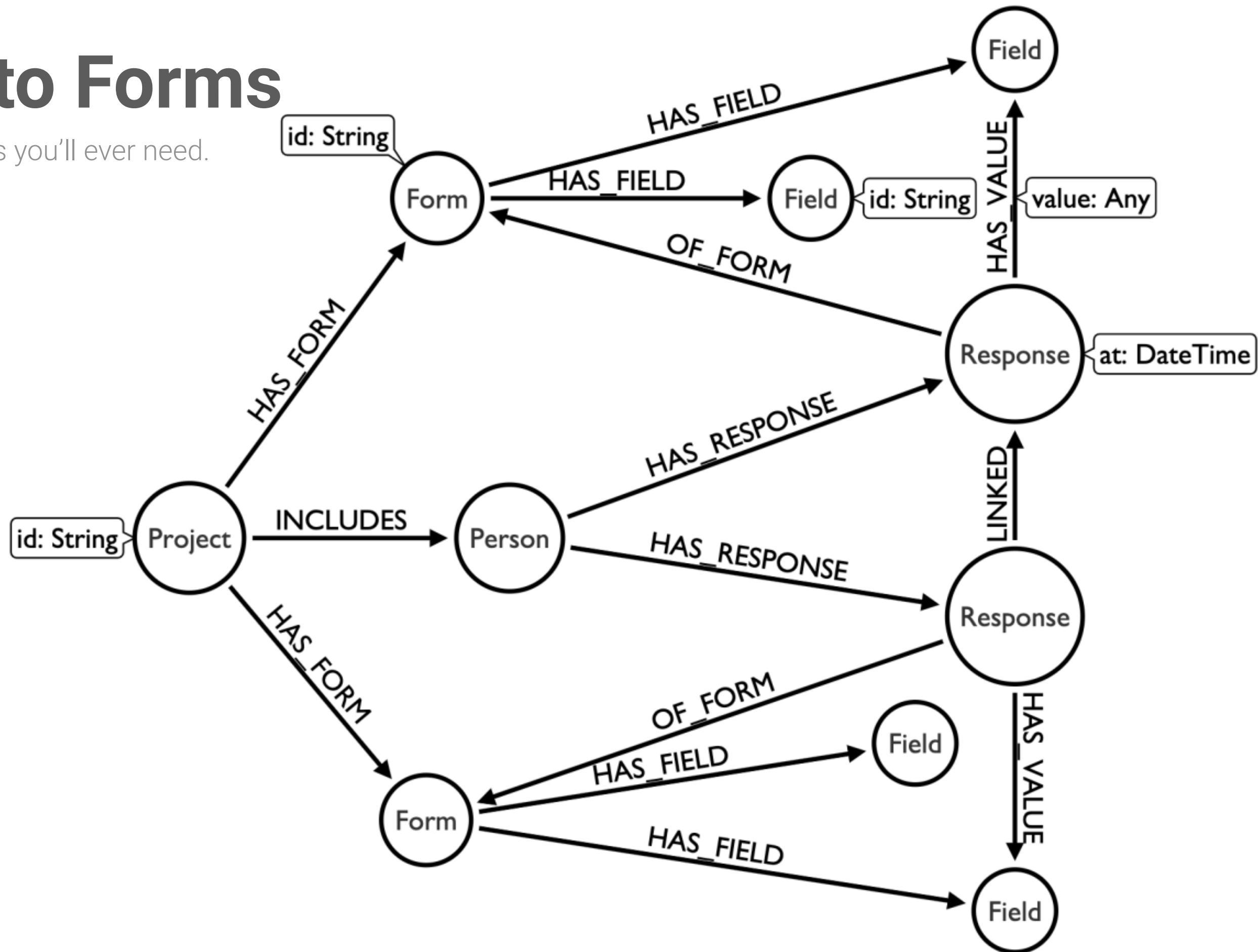


Wait a minute

- *What if they fill out lots of forms?*
- *What if they respond to the same form twice?*
- *What if responses to forms are linked together?*
- *Are we missing a concept?*

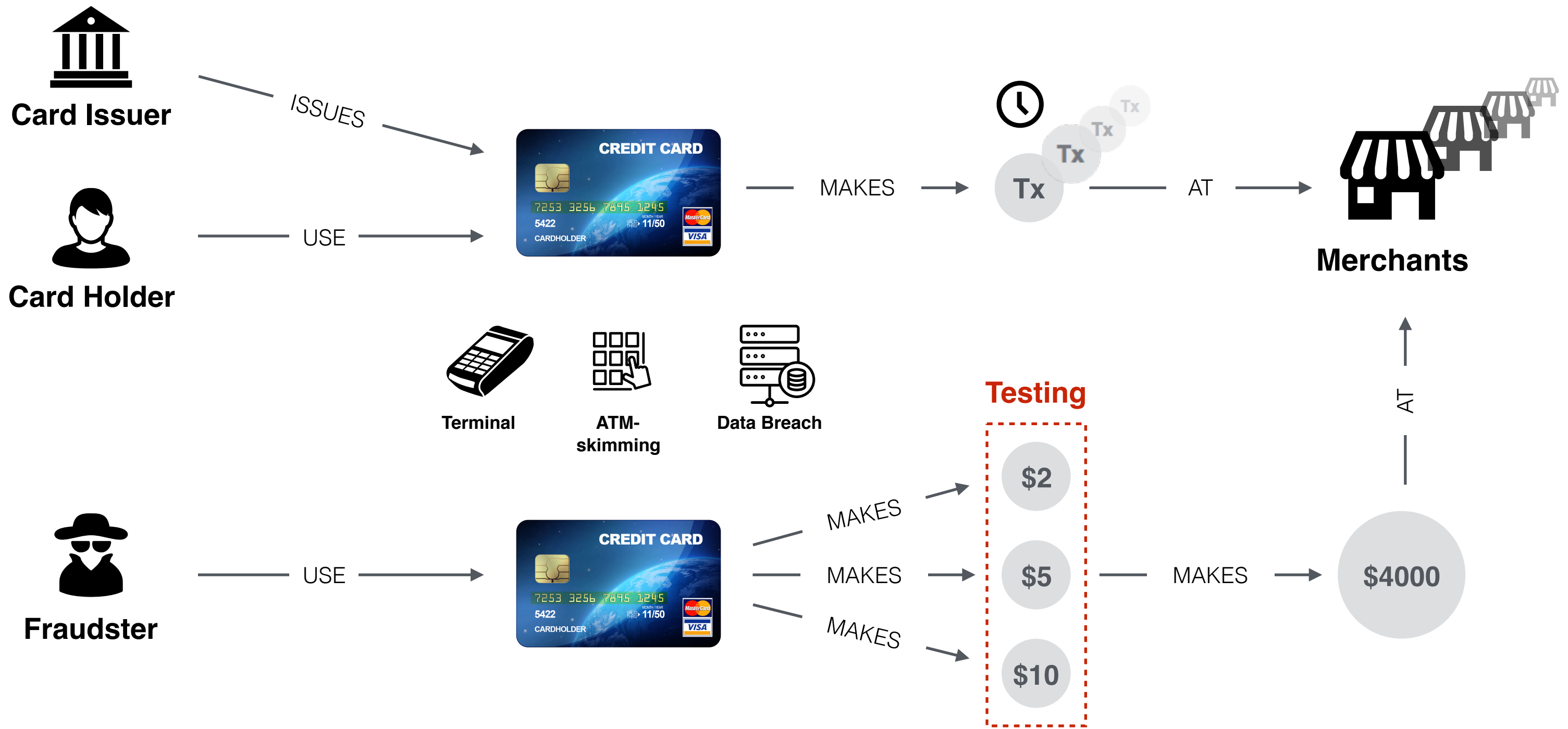
Responses to Forms

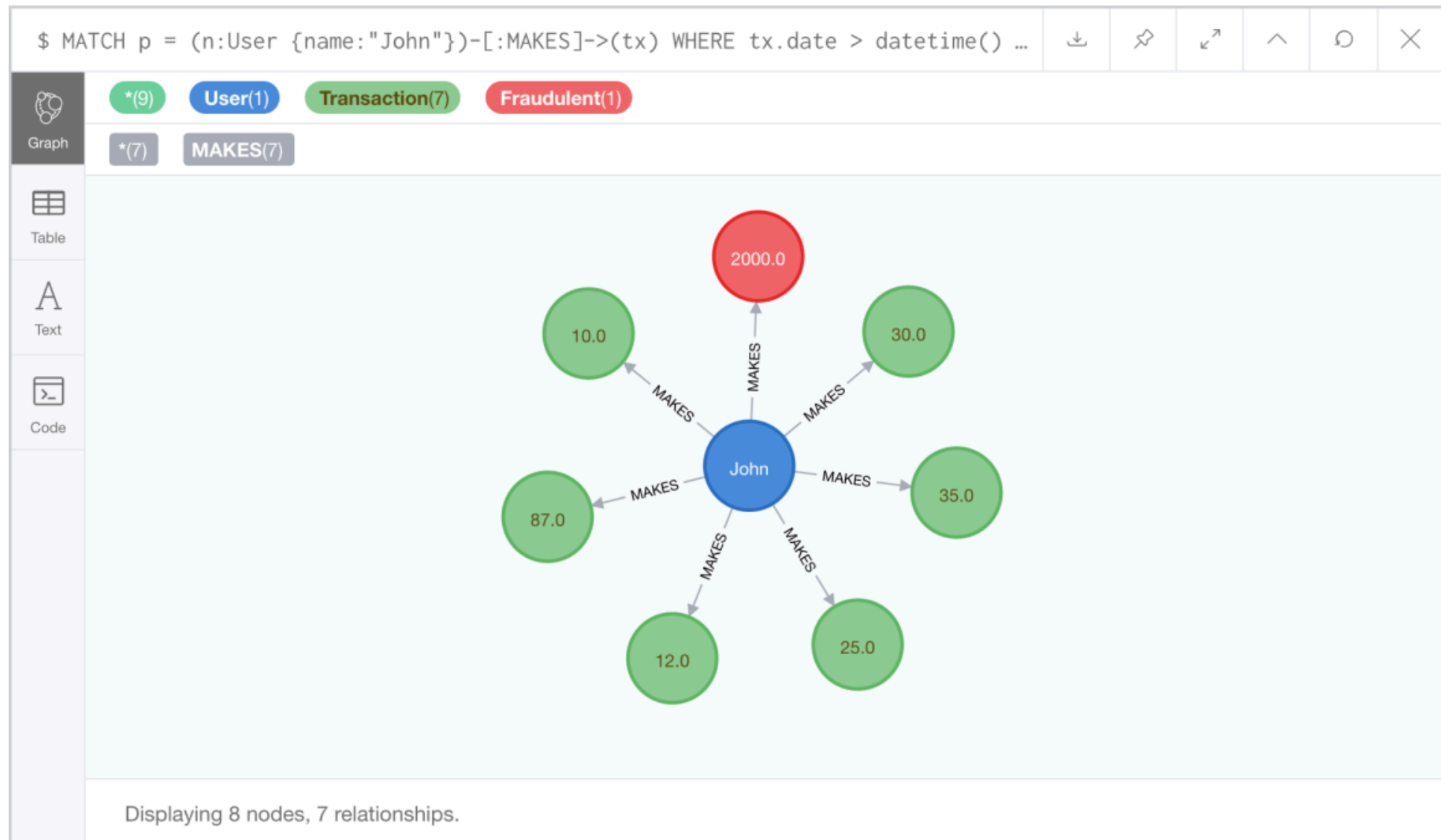
For all forms, all fields, all values you'll ever need.





Modeling Chains

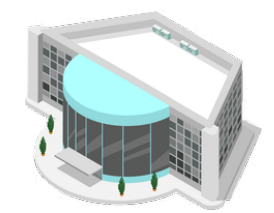




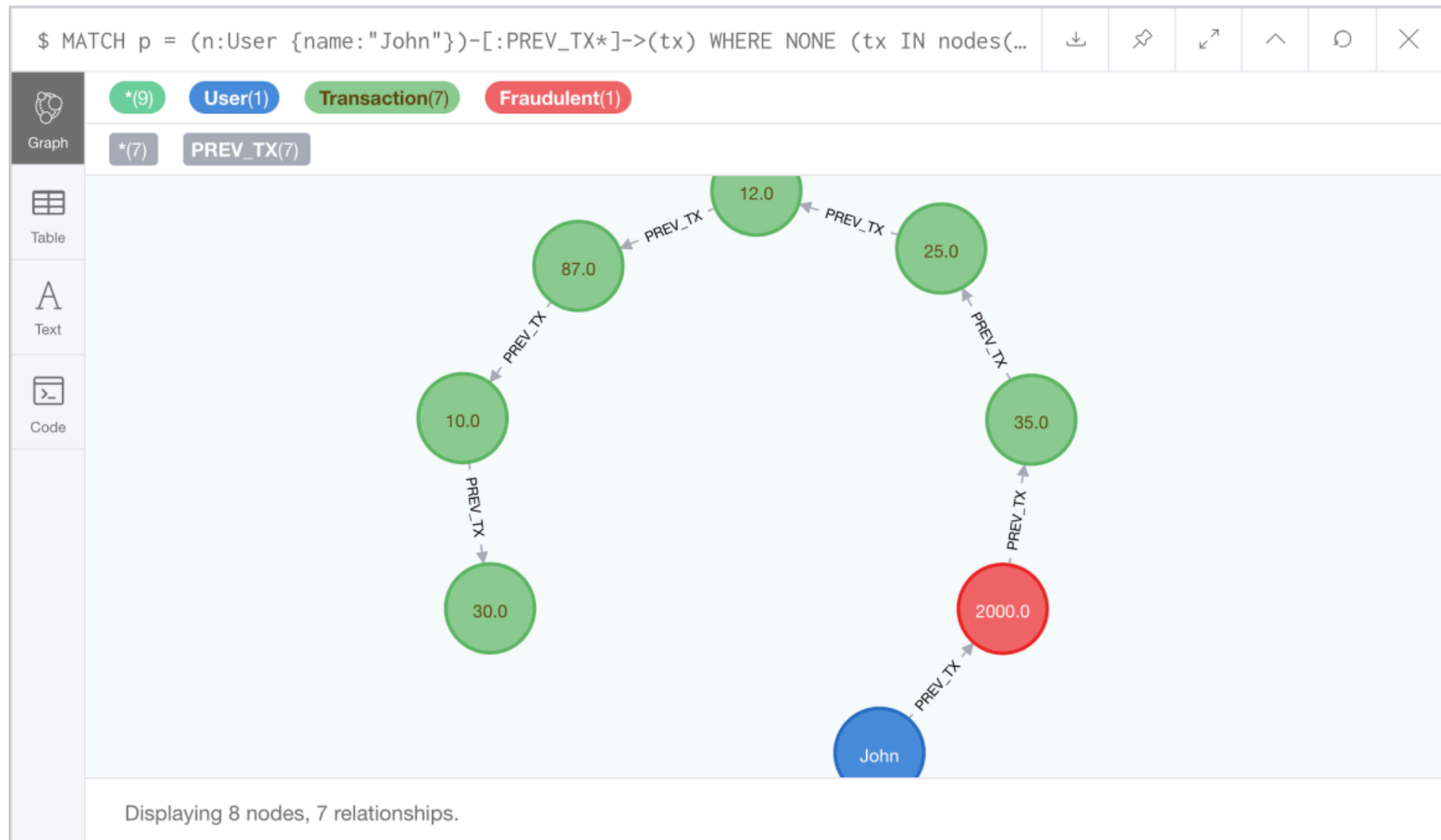
John 



\$2000



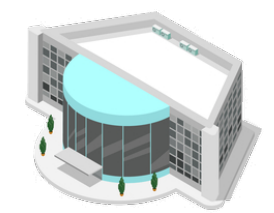
Computer Store



John 



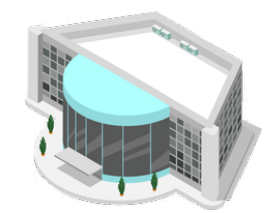
\$2000



Computer Store

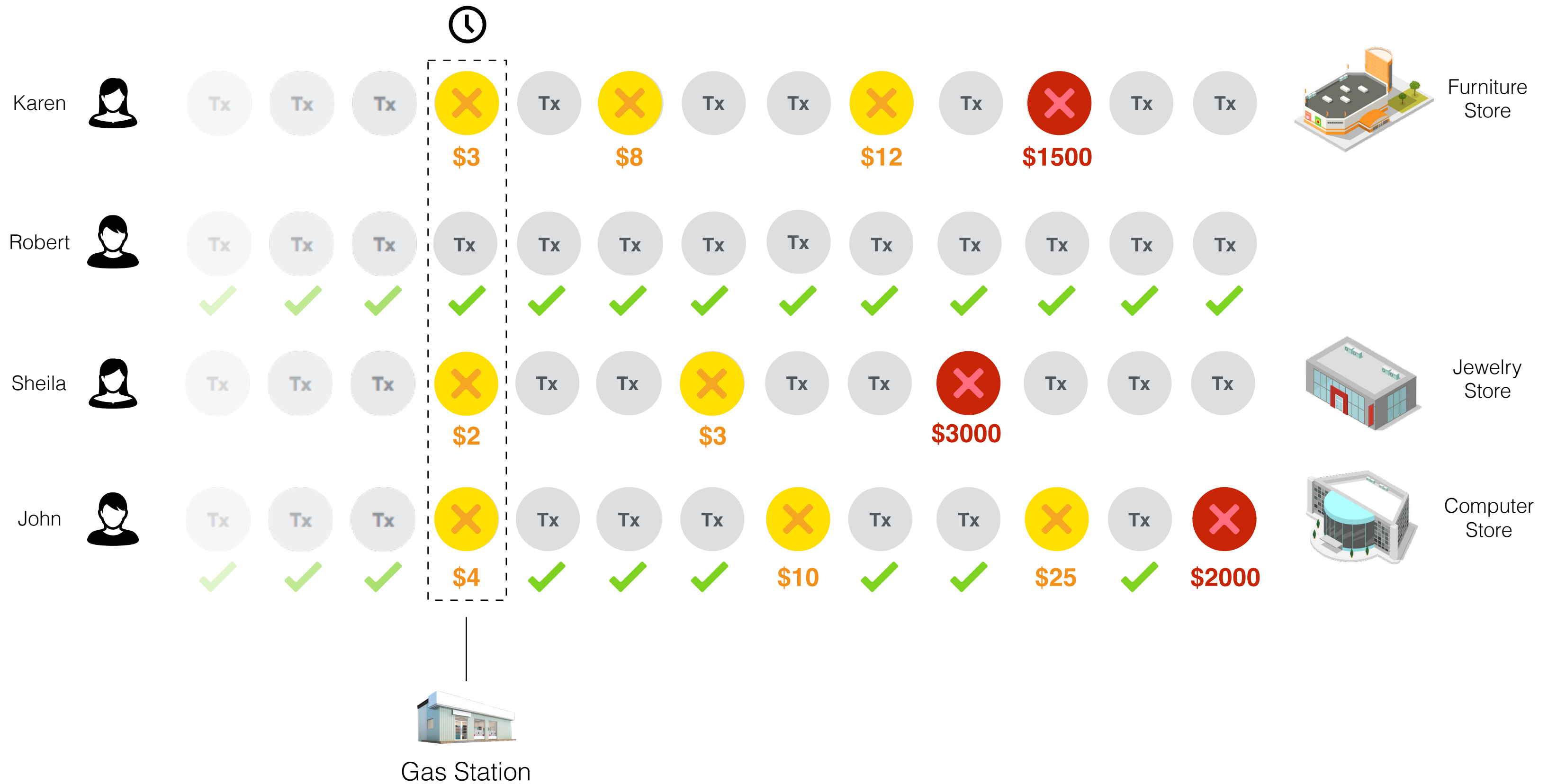


John 



Computer Store







Hunger Games

Questions

1. What is the **Tool** Max uses for Modeling?
2. Up to **how many** relationship types can I use?
 - A. 8192
 - B. 32768
 - C. 65536
3. What is the **secret** to Neo4j?



Thank you!

