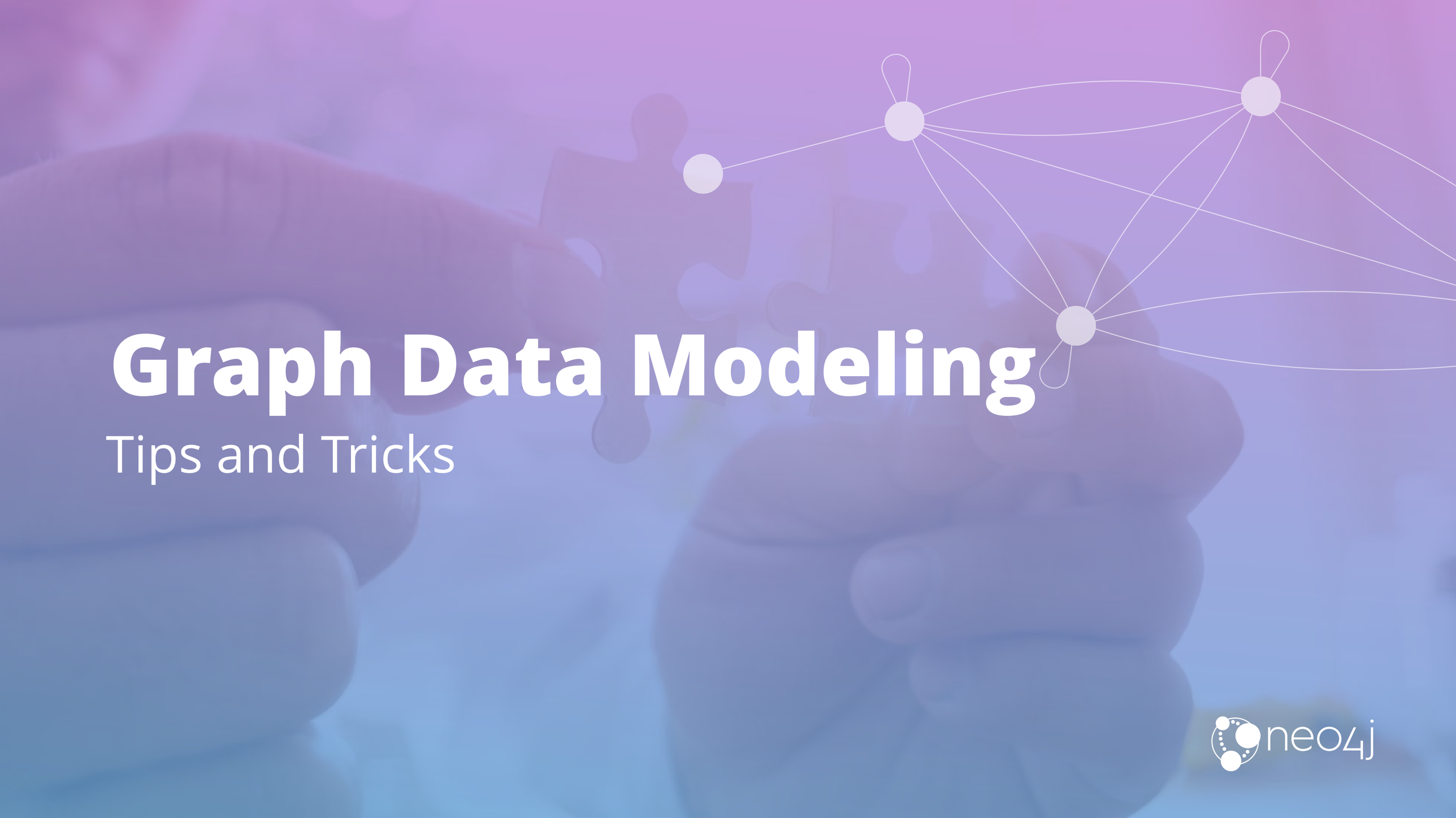


Graph Data Modeling

Tips and Tricks

neo4j

About Me !

01 **Max De Marzi**
Neo4j Field Engineer

02 **maxdemarzi.com**
About 160 blog posts

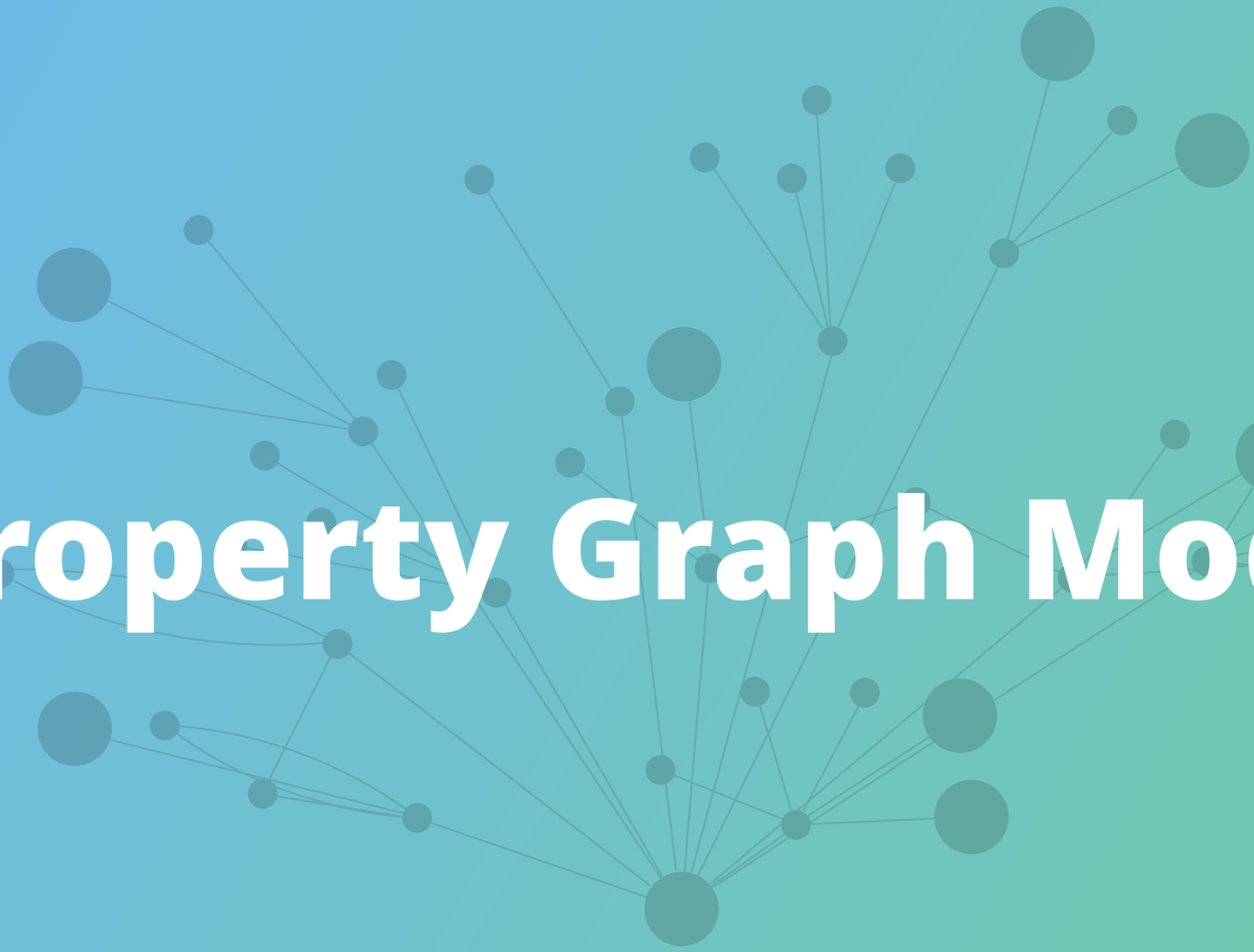
03 **@maxdemarzi**

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



neo4j

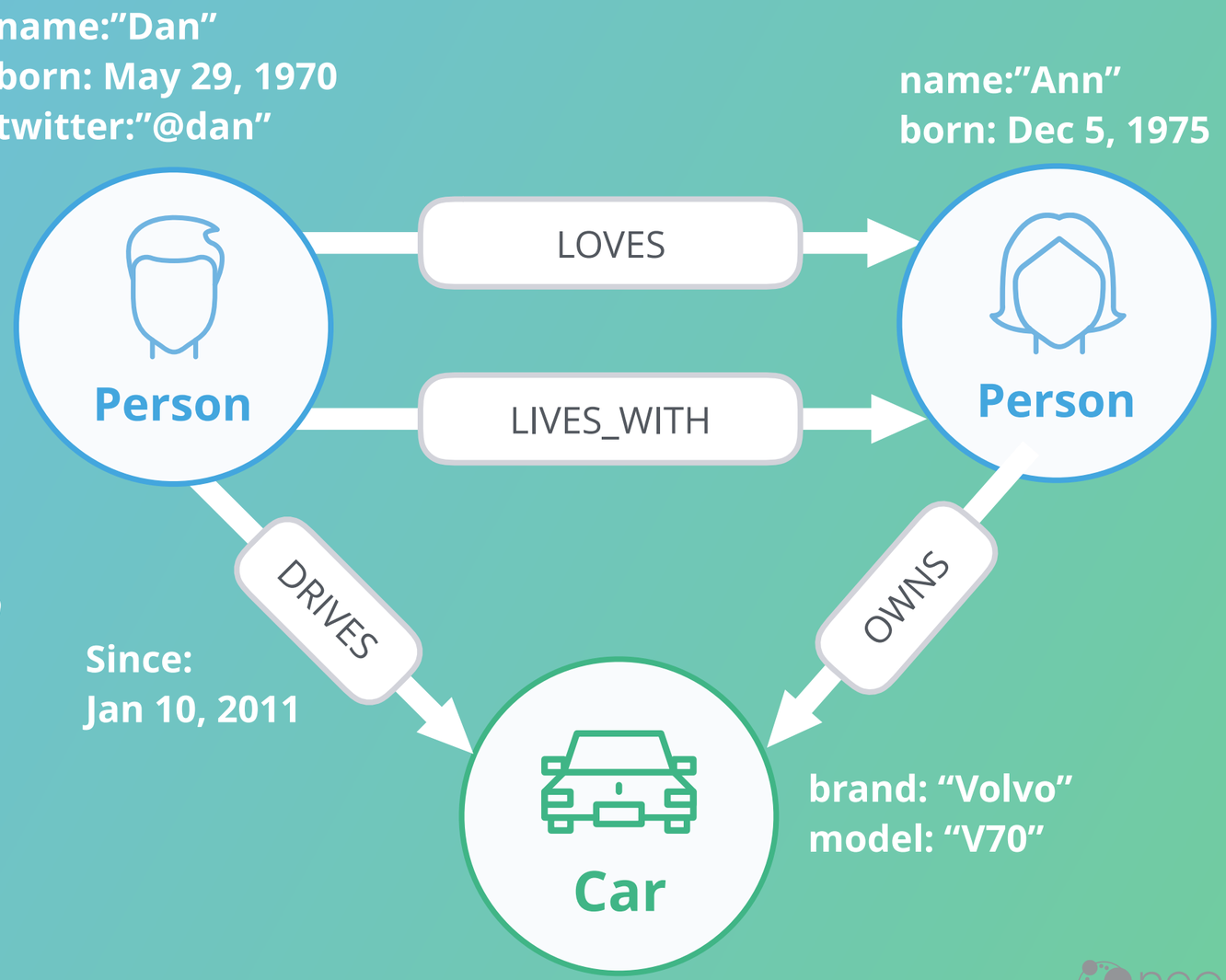
Property Graph Model Components

Nodes

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

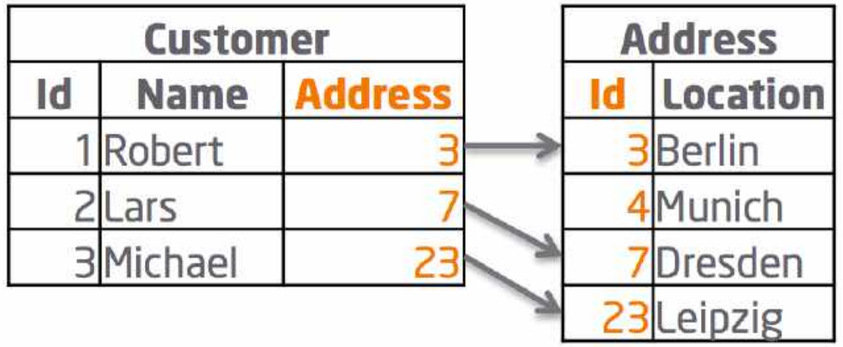
Relationships

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



What you (probably) already know:

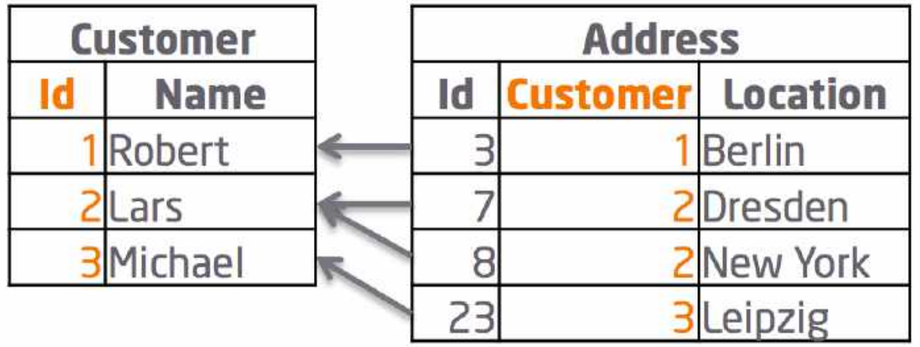
SQL Join Hell (1)



Customer		
Id	Name	Address
1	Robert	3
2	Lars	7
3	Michael	23

Address	
Id	Location
3	Berlin
4	Munich
7	Dresden
23	Leipzig

1:1 Relationship



Customer	
Id	Name
1	Robert
2	Lars
3	Michael

Address		
Id	Customer	Location
3	1	Berlin
7	2	Dresden
8	2	New York
23	3	Leipzig

1:n Relationship

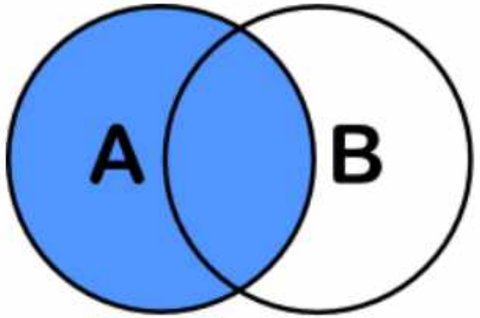
Customer	
Id	Name
1	Robert
2	Lars
3	Michael

CustomerAddress	
CId	AId
1	3
2	7
2	8
3	23

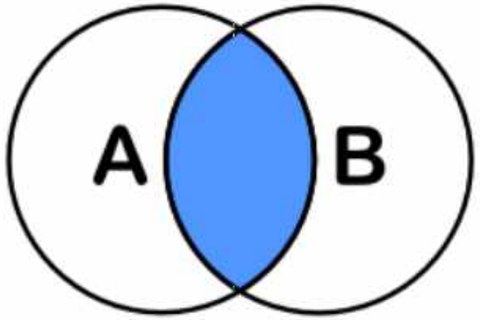
Address	
Id	Location
3	Berlin
7	Dresden
8	New York
23	Leipzig

m:n Relationship

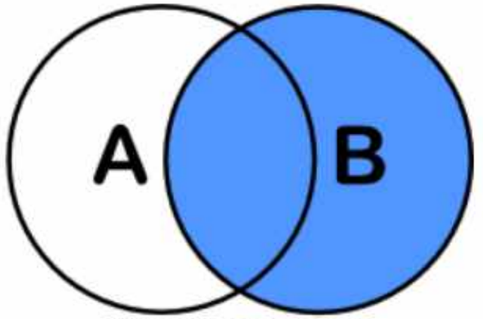
CHEATSHEET SQL JOINS



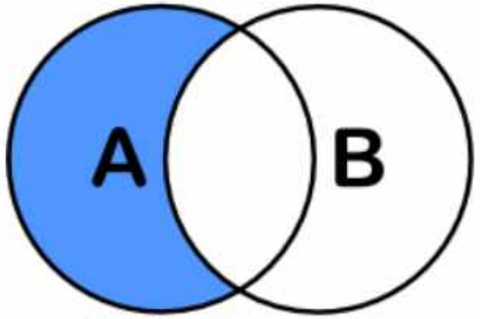

```
SELECT <auswahl>
FROM tabelleA A
LEFT JOIN tabelleB B
ON A.key = B.key
```



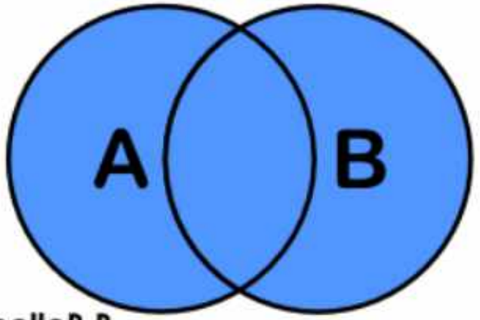

```
SELECT <auswahl>
FROM tabelleA A
INNER JOIN tabelleB B
ON A.key = B.key
```



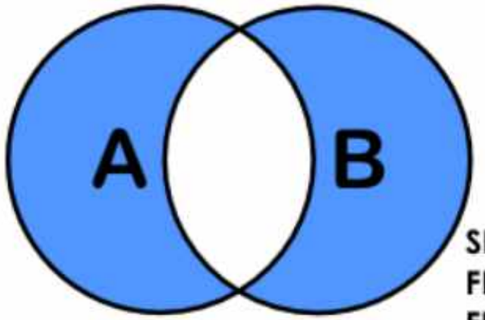

```
SELECT <auswahl>
FROM tabelleA A
RIGHT JOIN tabelleB B
ON A.key = B.key
```



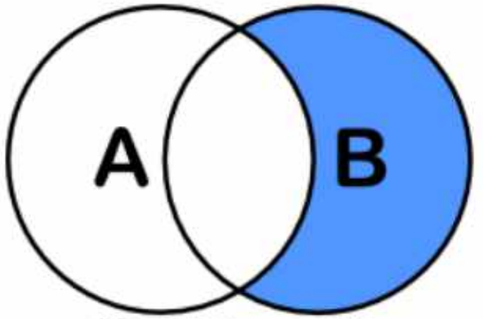

```
SELECT <auswahl>
FROM tabelleA A
LEFT JOIN tabelleB B
ON A.key = B.key
WHERE B.key IS NULL
```



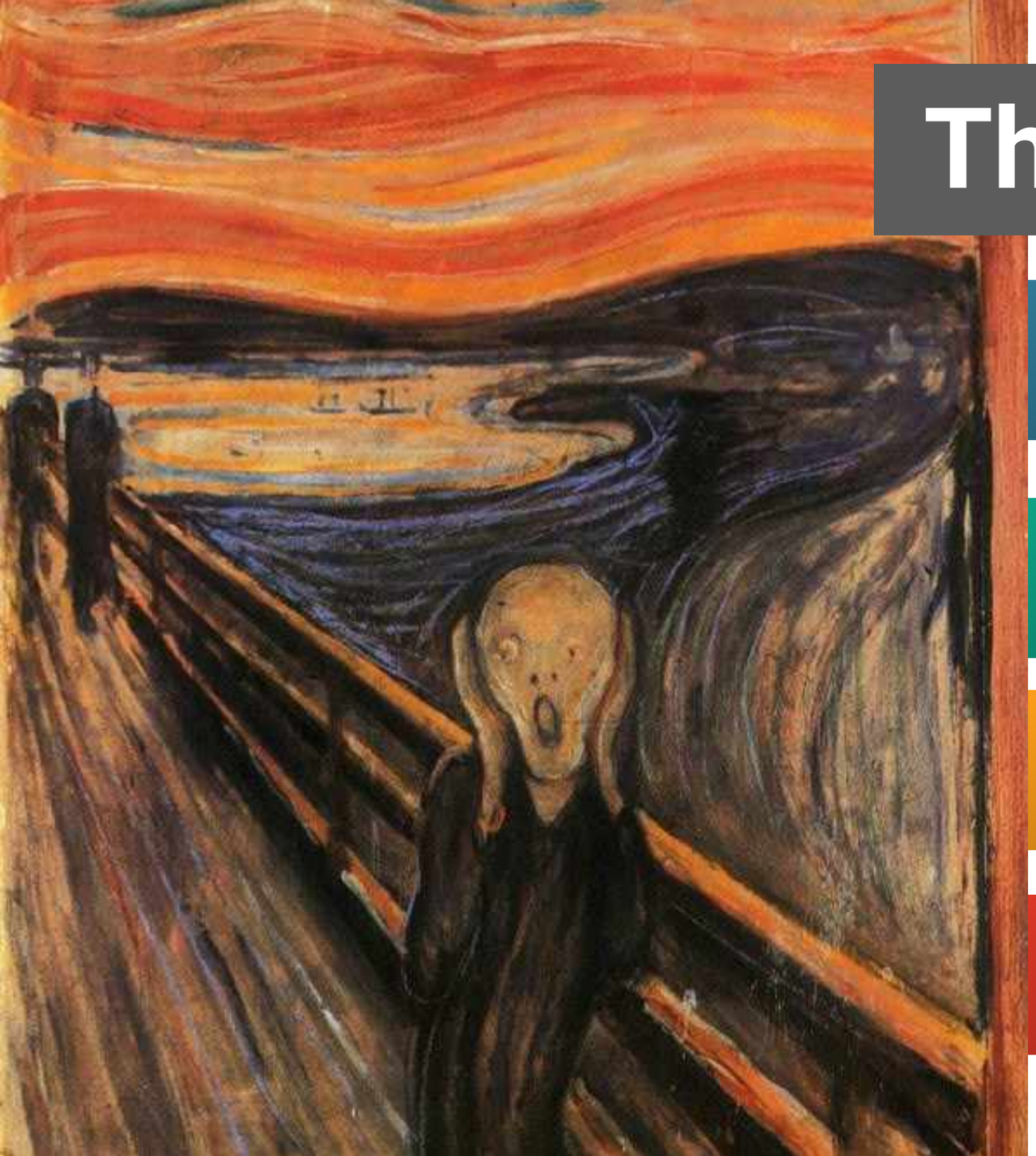

```
SELECT <auswahl>
FROM tabelleA A
FULL OUTER JOIN tabelleB B
ON A.key = B.key
```




```
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
```




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



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

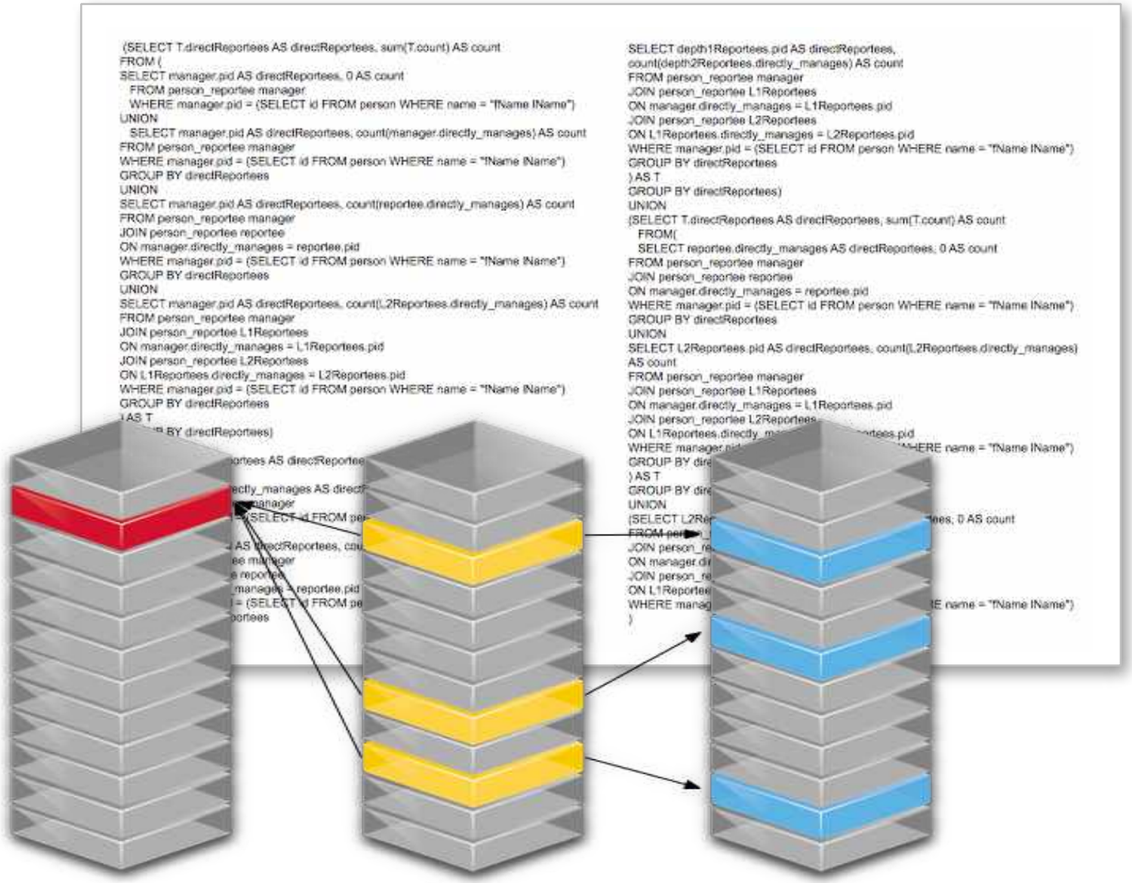
Wrong Language

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

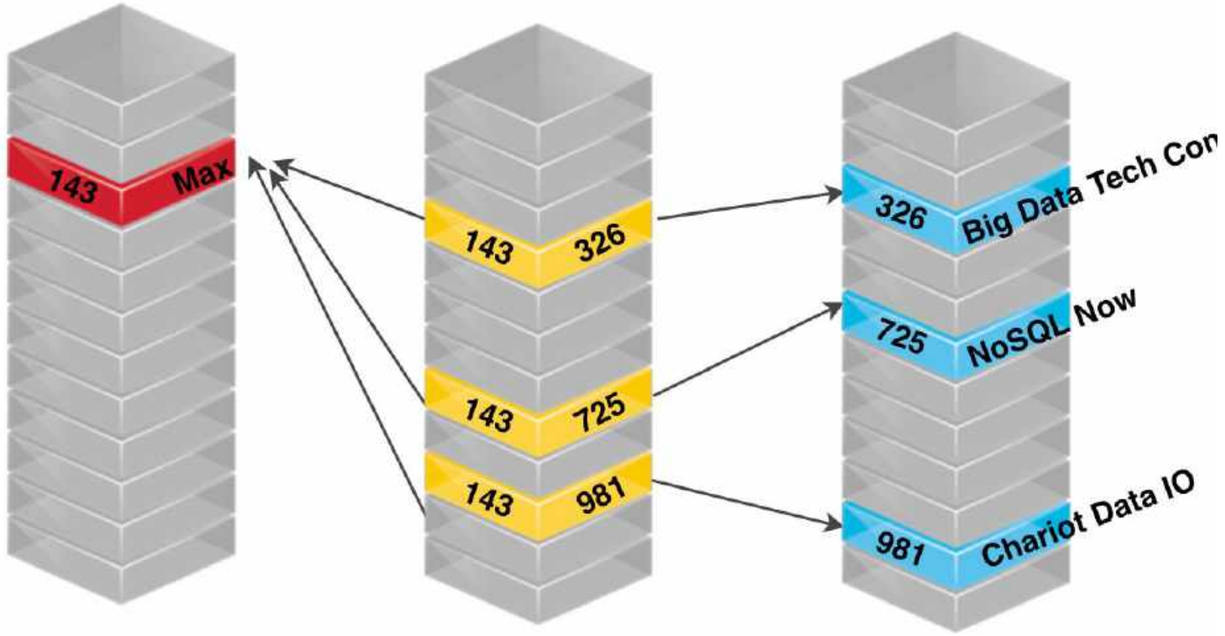
2 Degraded Performance
Speed plummets as you try to join data together in the application

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

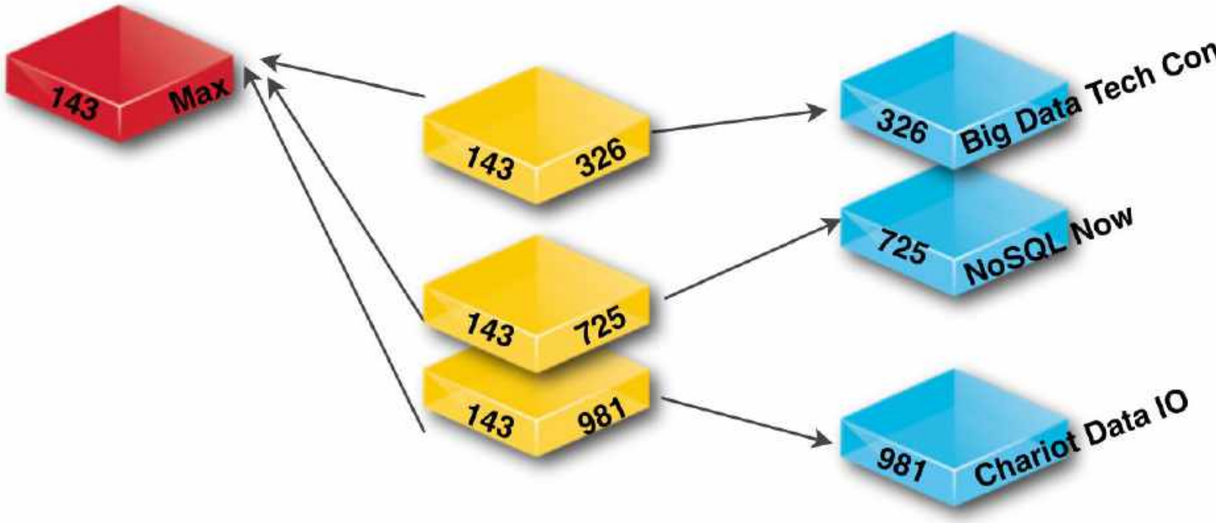
4 Not ACID
Eventually Consistent means Eventually Corrupt

Same Data, Different Layout

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

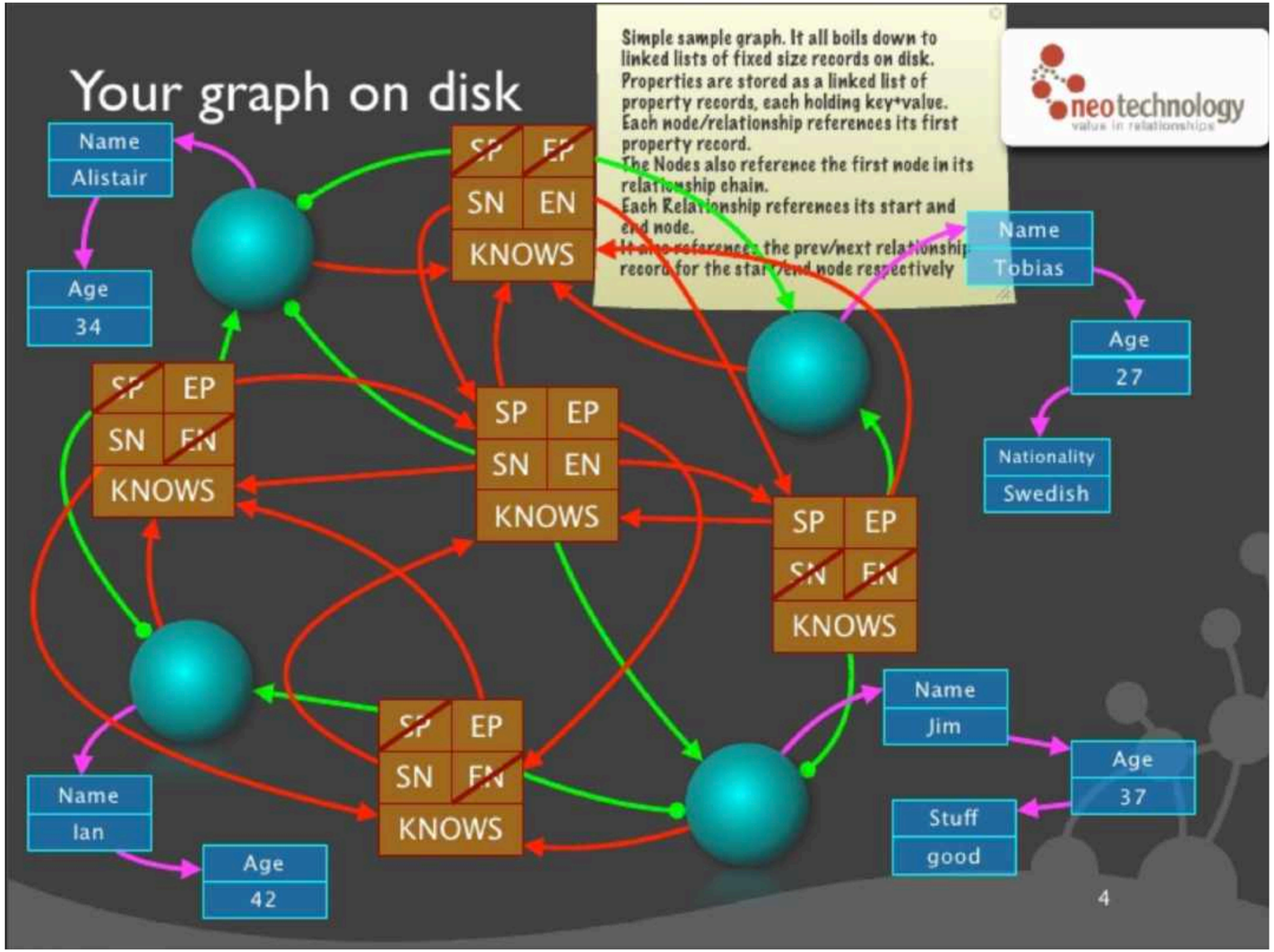


People Attend Conferences



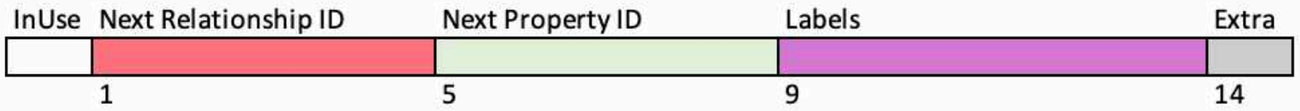
People Attend Conferences

Double Linked List Relationship Layout

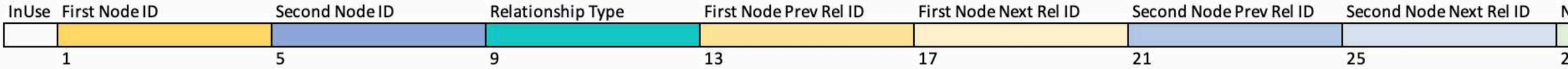


Neo4j Storage Record Layout

Node(15 Bytes)

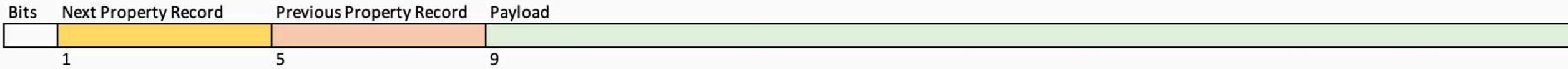


Relationship (34 Bytes)



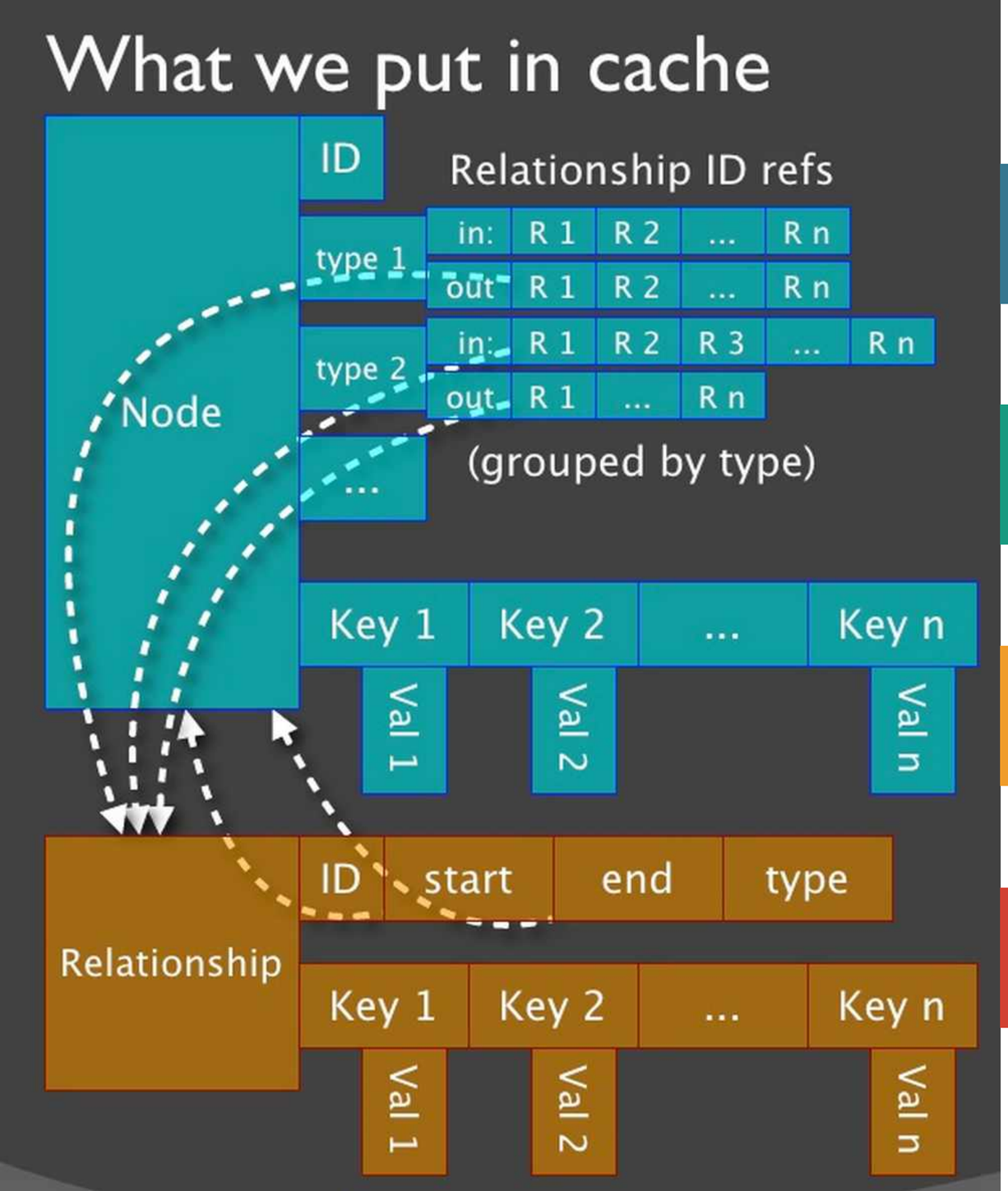
RelationshipGroup (25 Bytes)

Property Record (41 Bytes)



Dynamic Record

The Most Important Slide about Neo4j you will ever see



1. **Pointers instead of Lookups**
2. **Fixed Sized Records**
3. **“Joins” on Creation**
4. **Spin Spin Spin through this data structure**

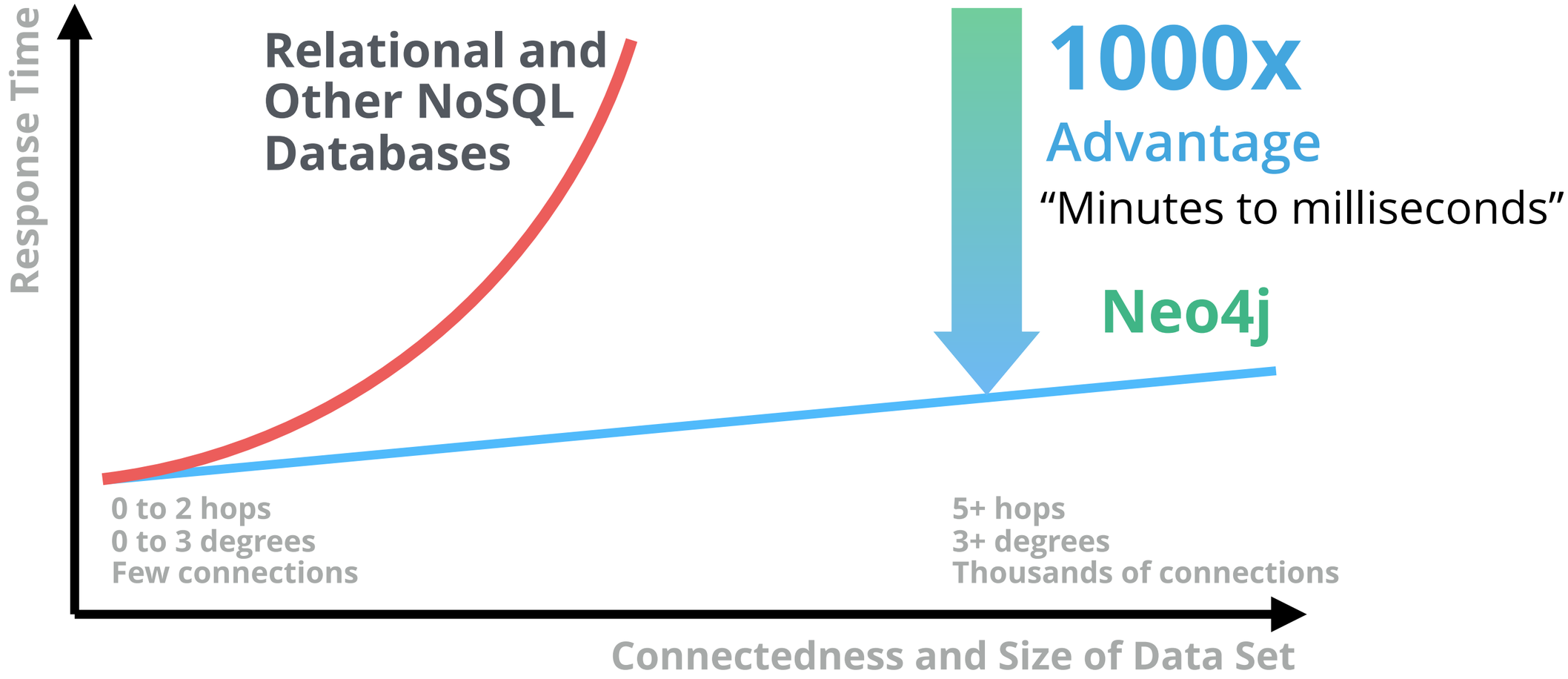

the Secret
TM

Swedes

Partitions

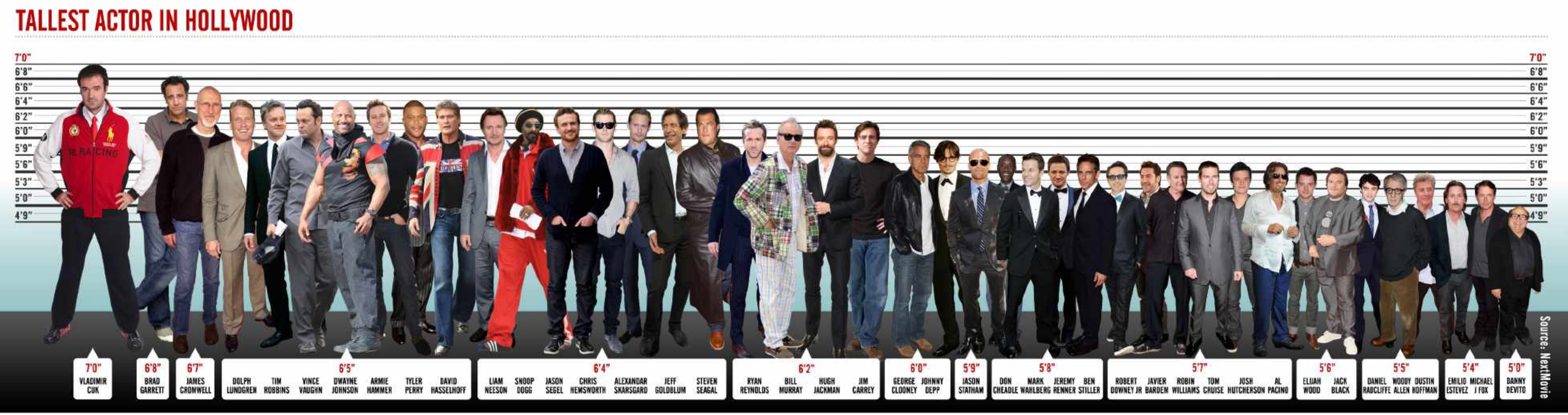
Each Node's relationships are **partitioned** by type and direction.

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



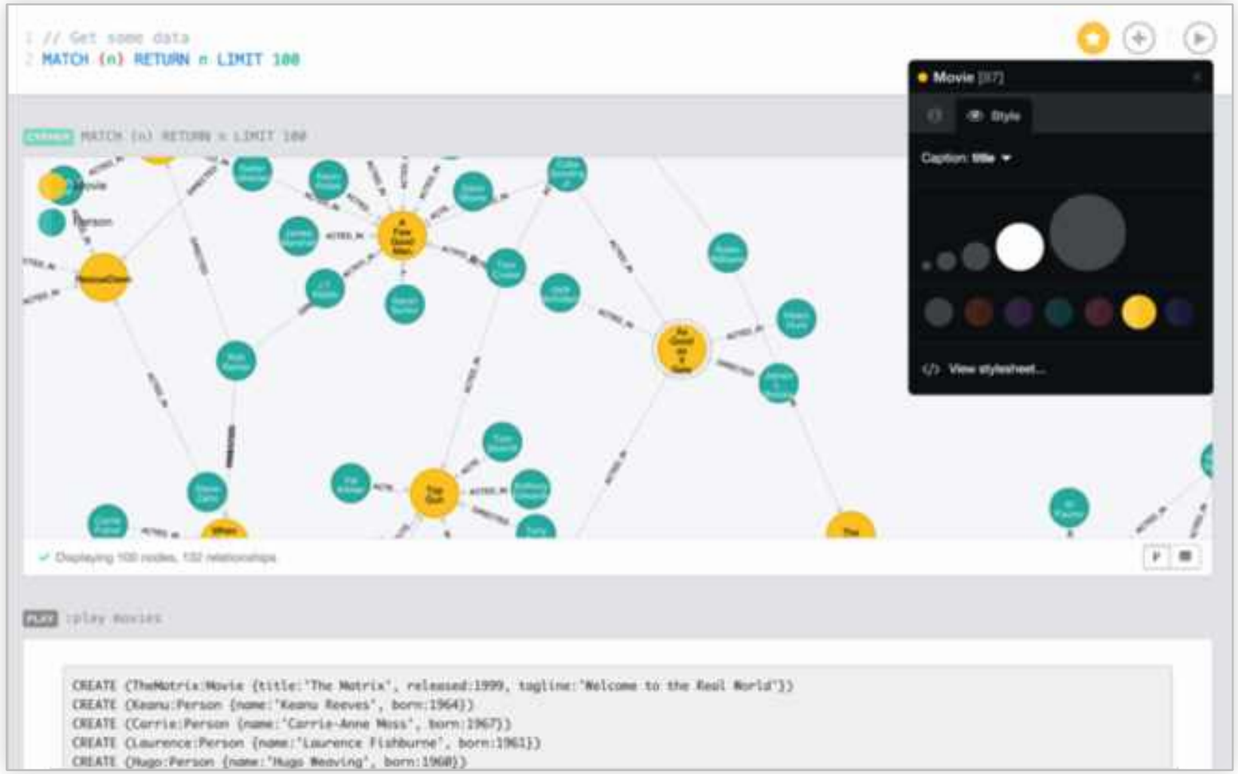
Reimagine your Data as a Graph

1 **Right Model**
Graphs simplify how you think

2 **Better Performance**
Query relationships in real time

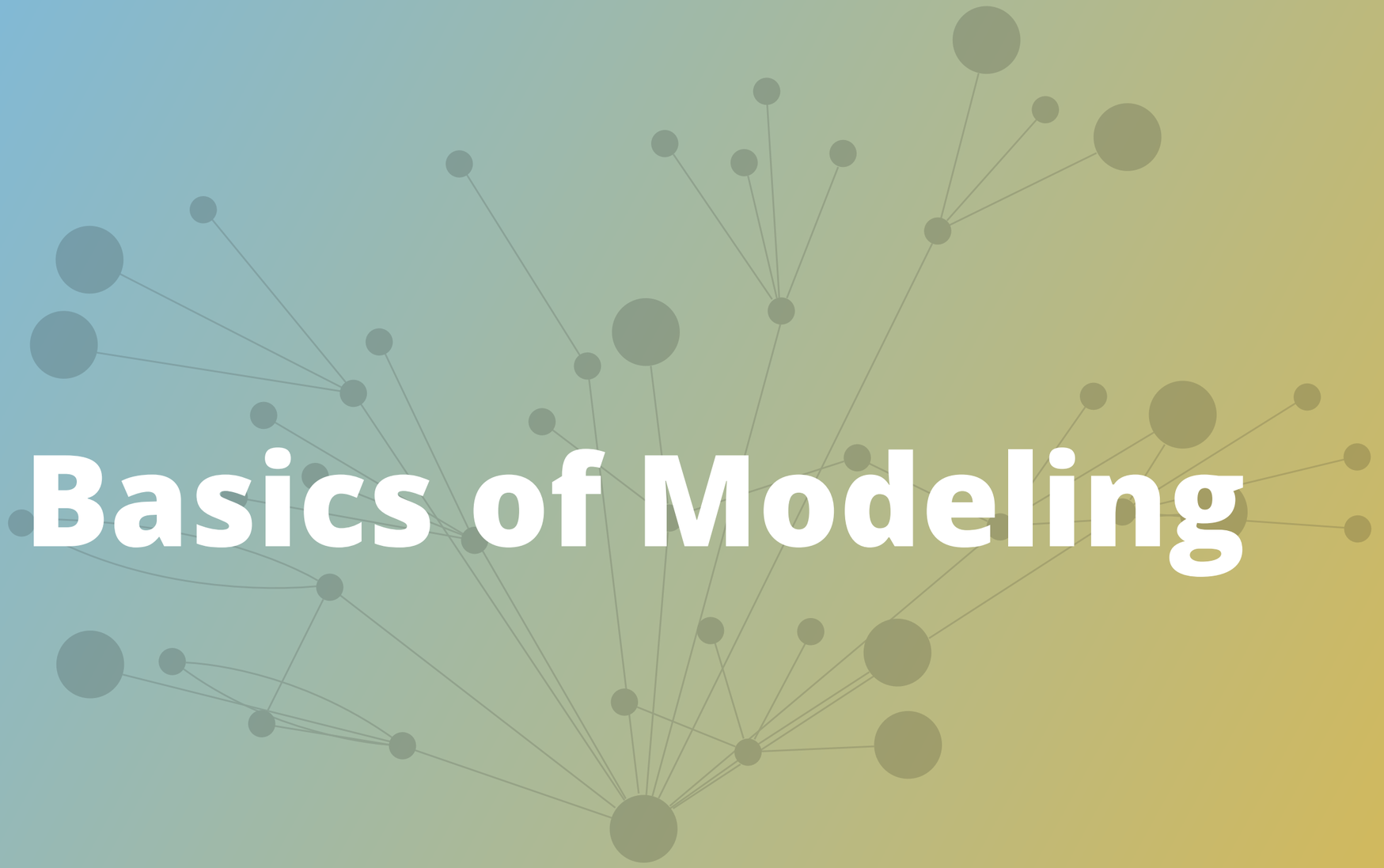
3 **Right Language**
Cypher was purpose built for Graphs

4 **Flexible and Consistent**
Evolve your schema seamlessly while keeping transactions



Agile, High Performance and Scalable without Sacrifice

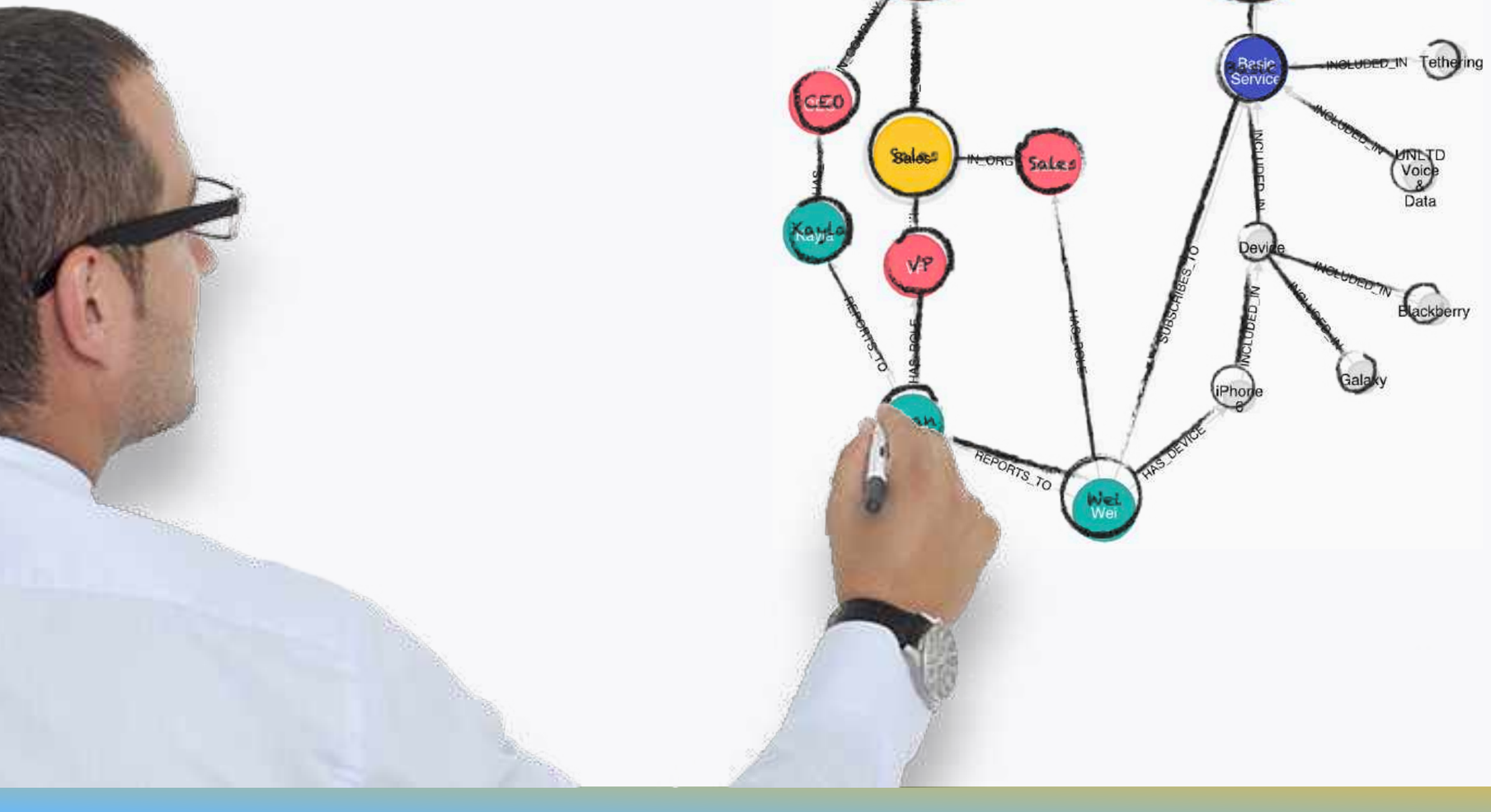
Basics of Modeling



neo4j

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*

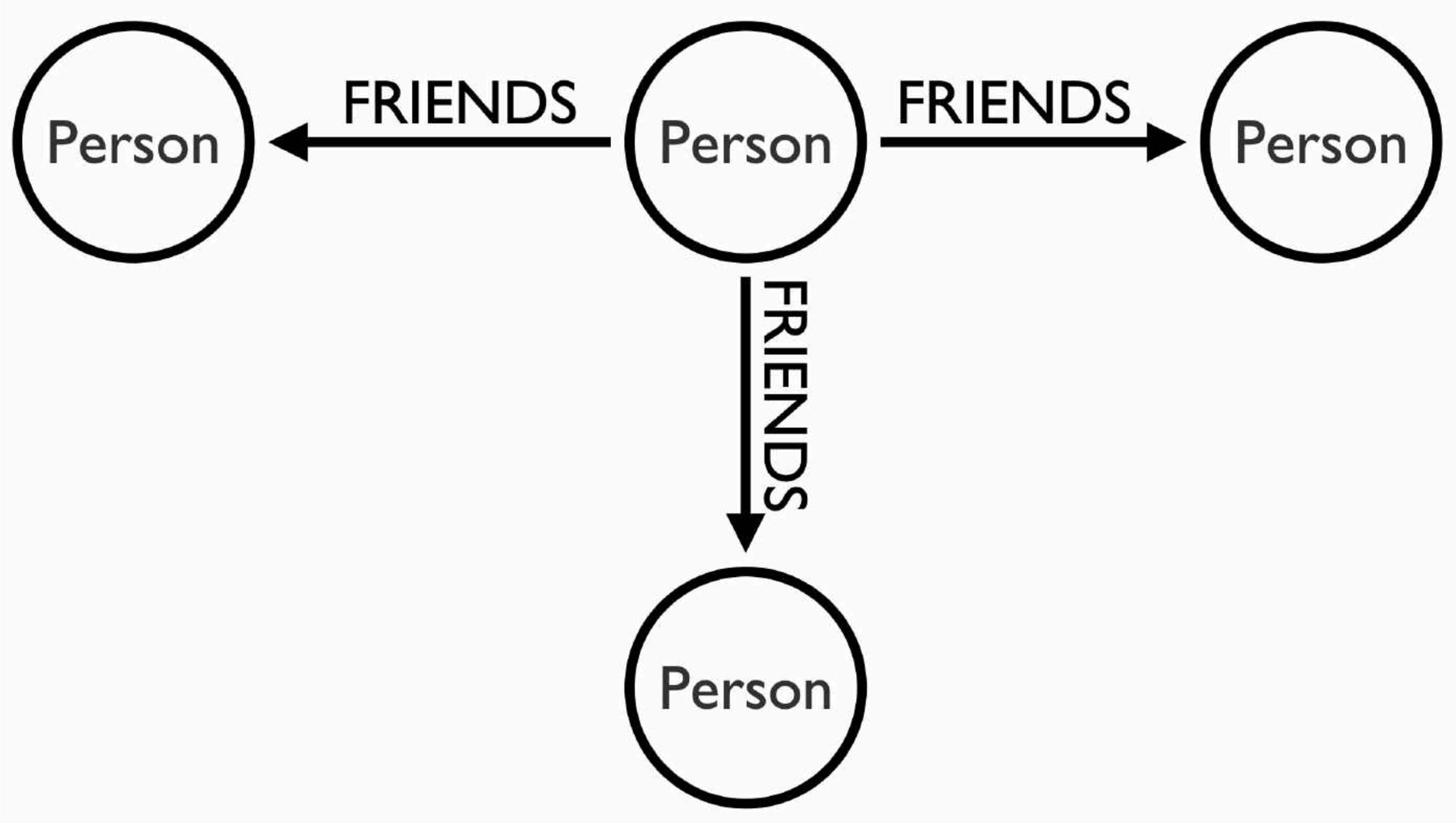
neo4j

Arrows

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

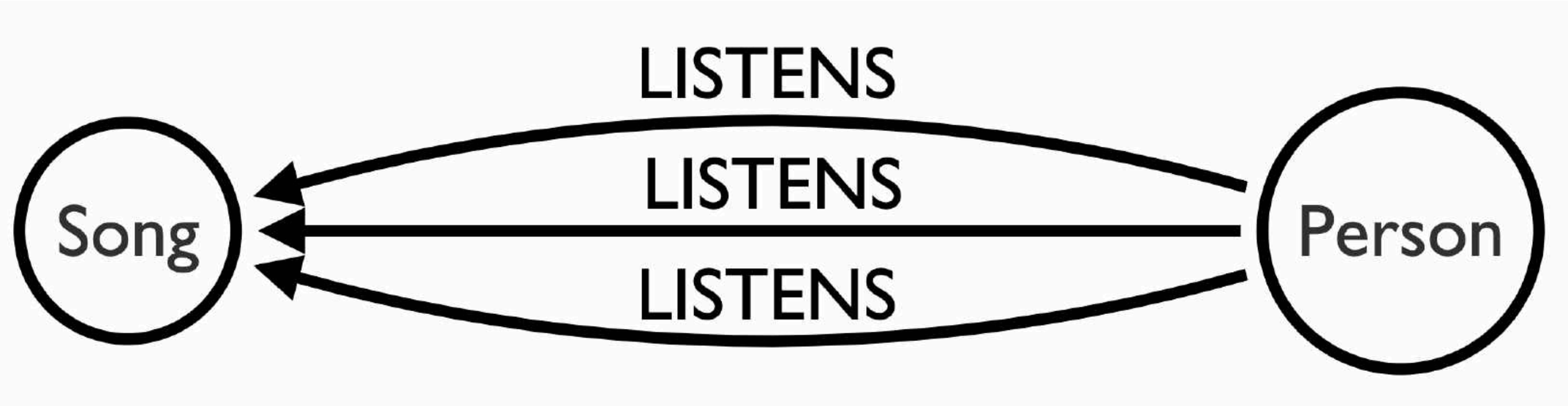
It’s simple, it’s fast, it doesn’t run out of ink.

Nodes can have many relationships

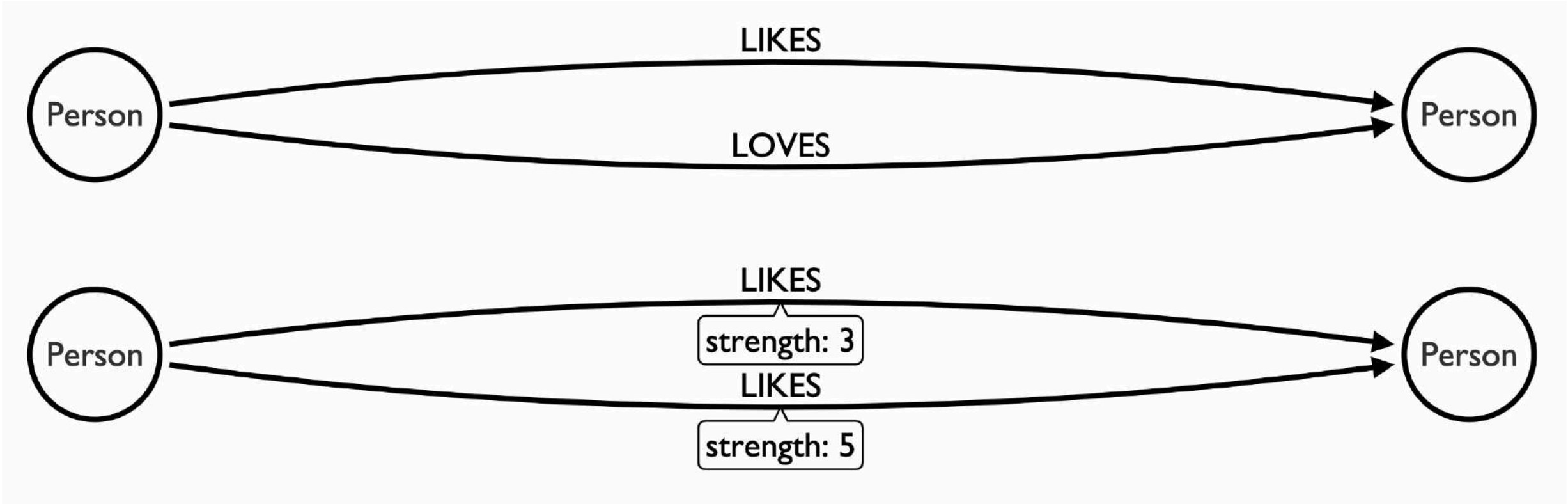


Two Nodes can be related by more than one relationship type

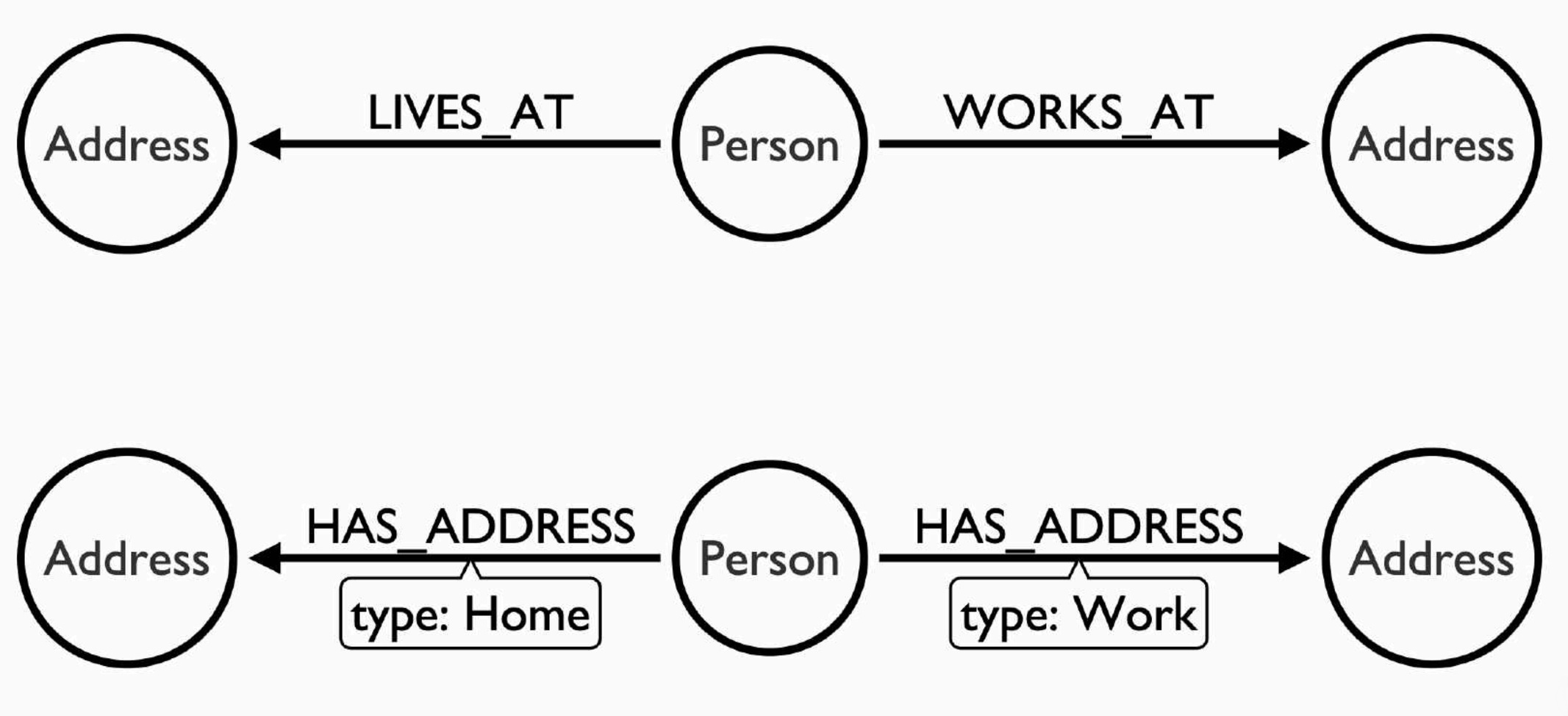
Two Nodes can be related by the same relationship type more than once



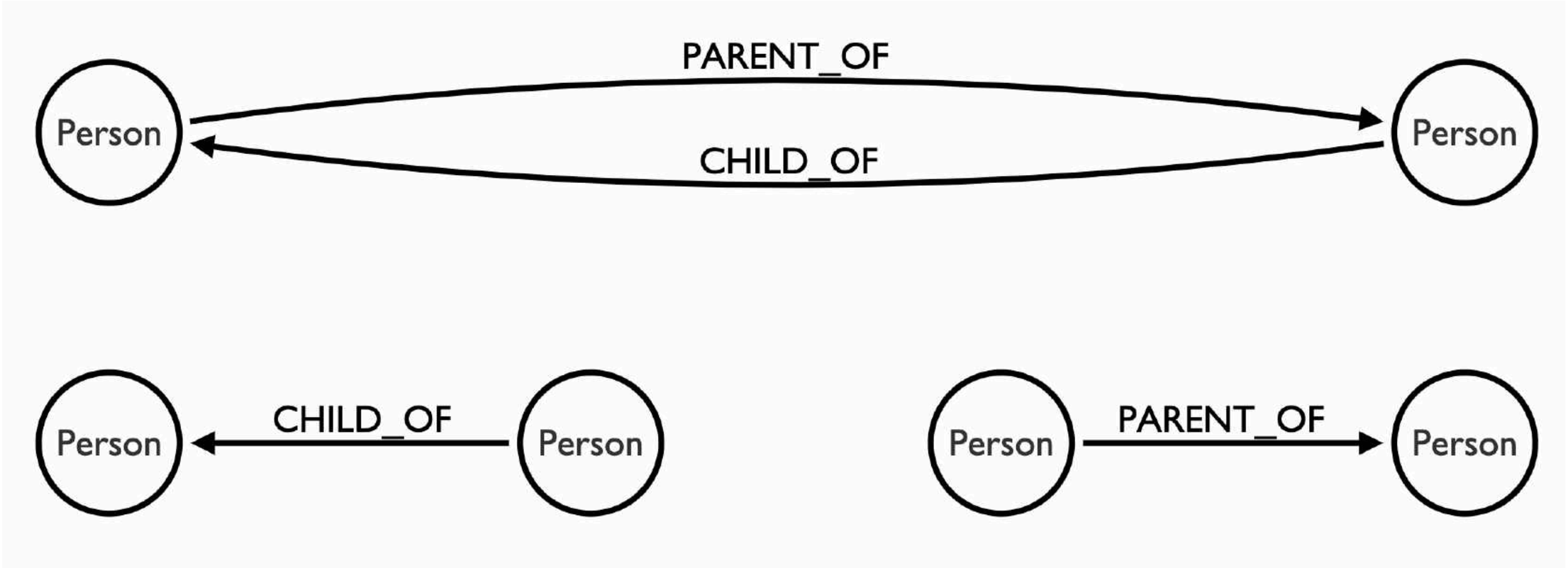
Relationships can have properties



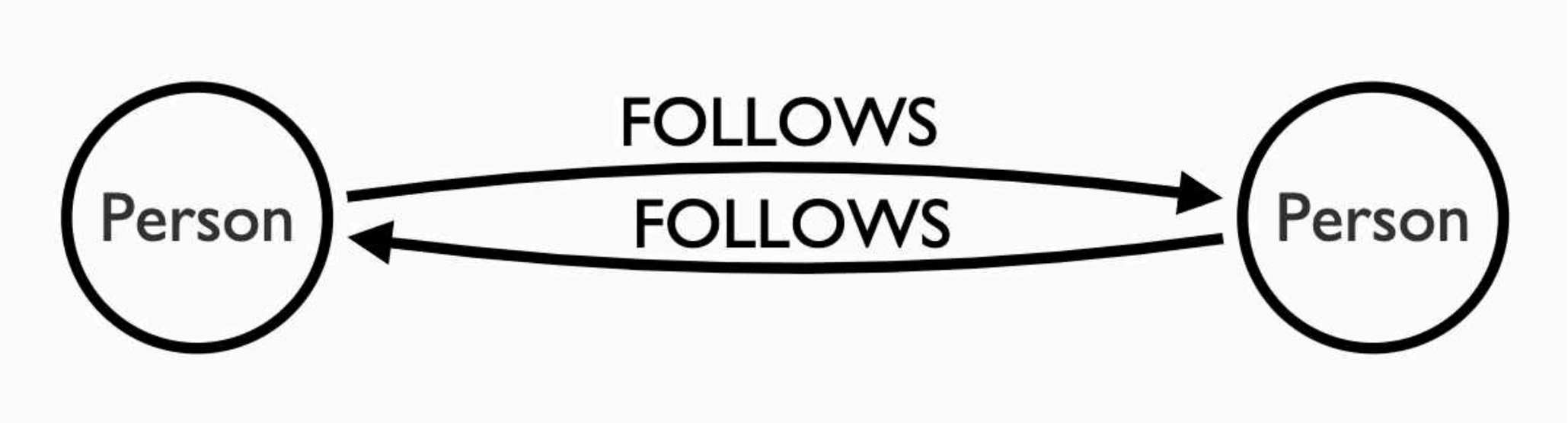
Don’t overcomplicate your model



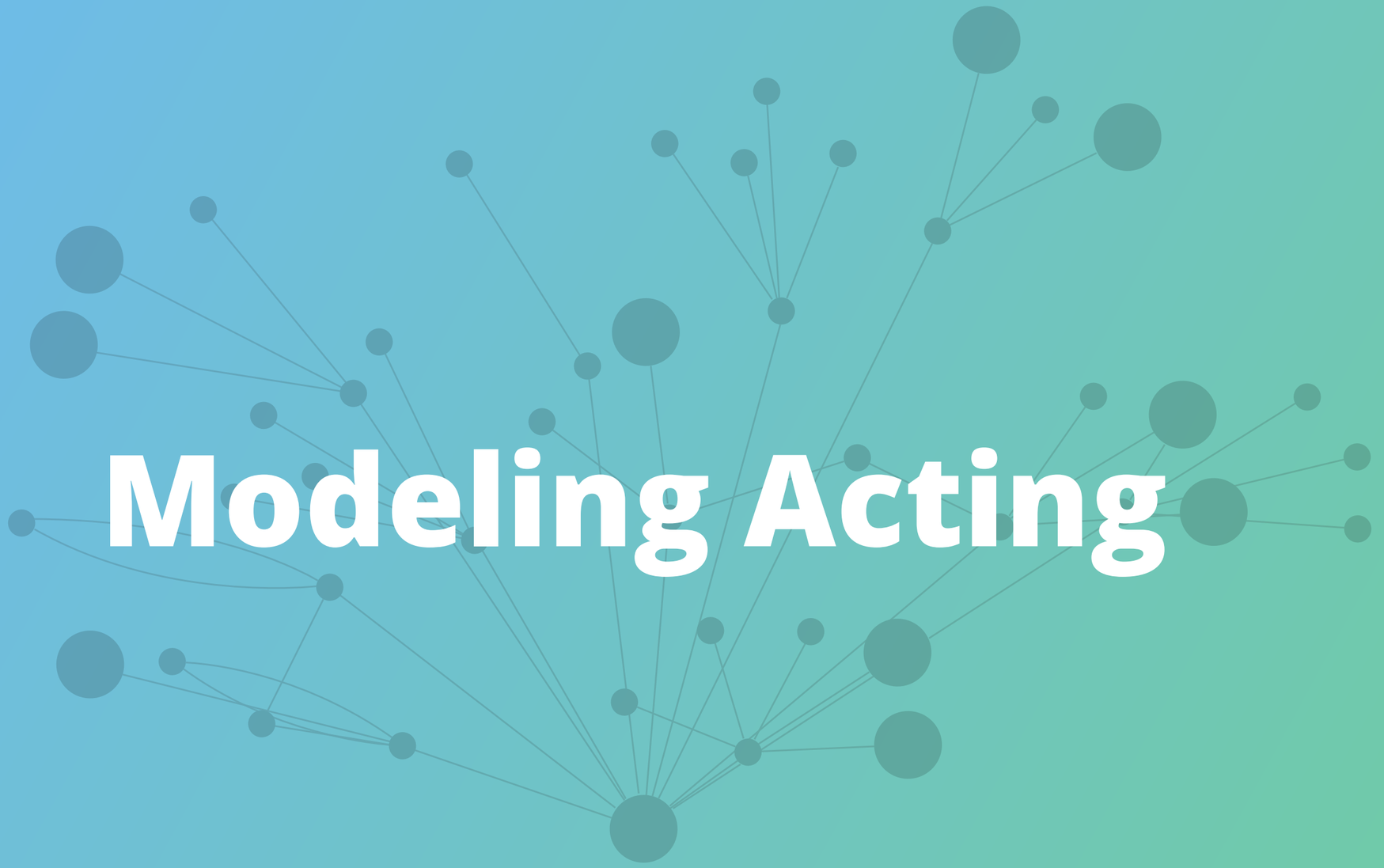
Pick one or the other, not both



...but this is fine:



Modeling Acting



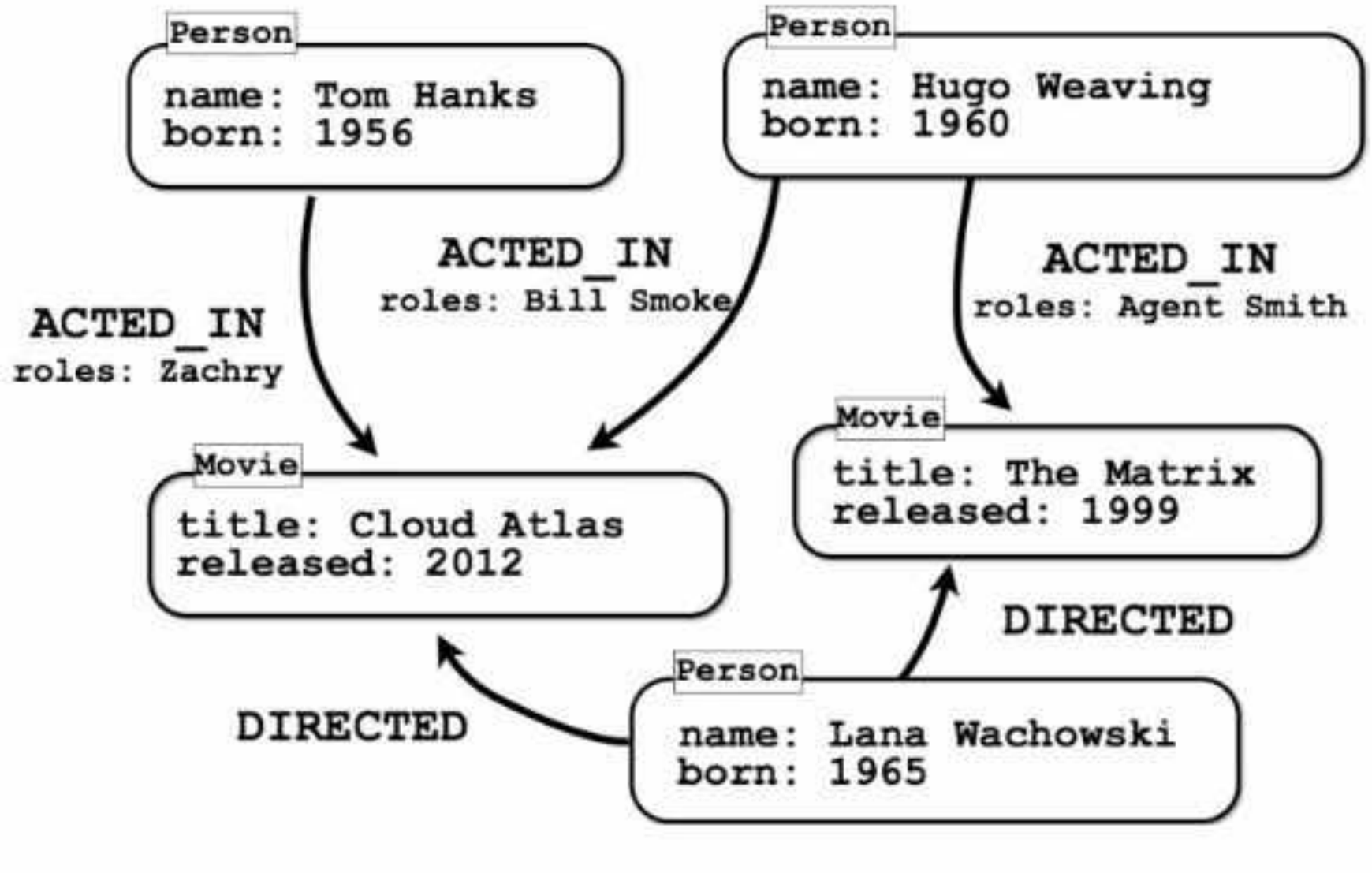
neo4j

Graphs are Whiteboard Friendly

Just draw stuff and “walla” there is your data model

Some Models are Easy

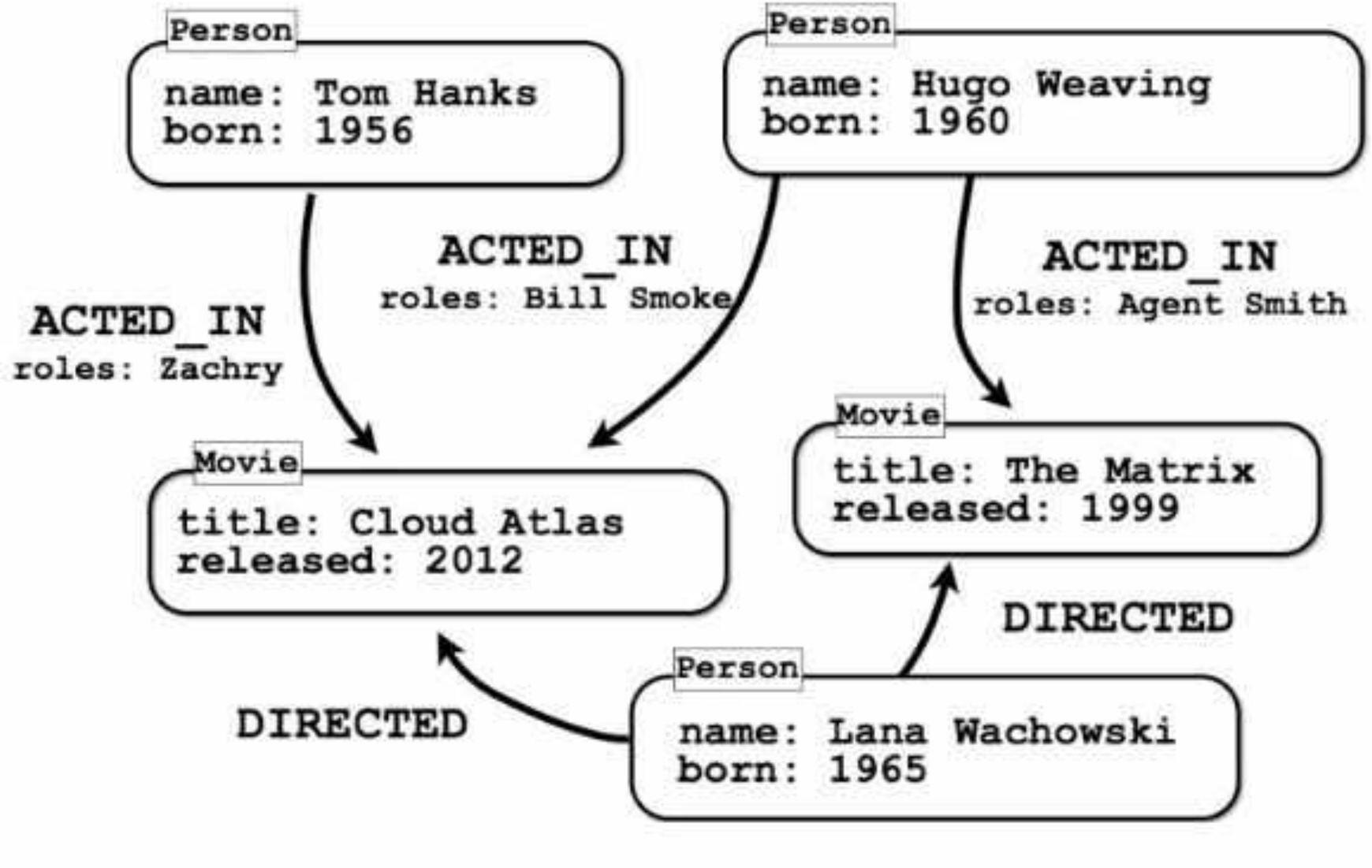
Movie Property Graph



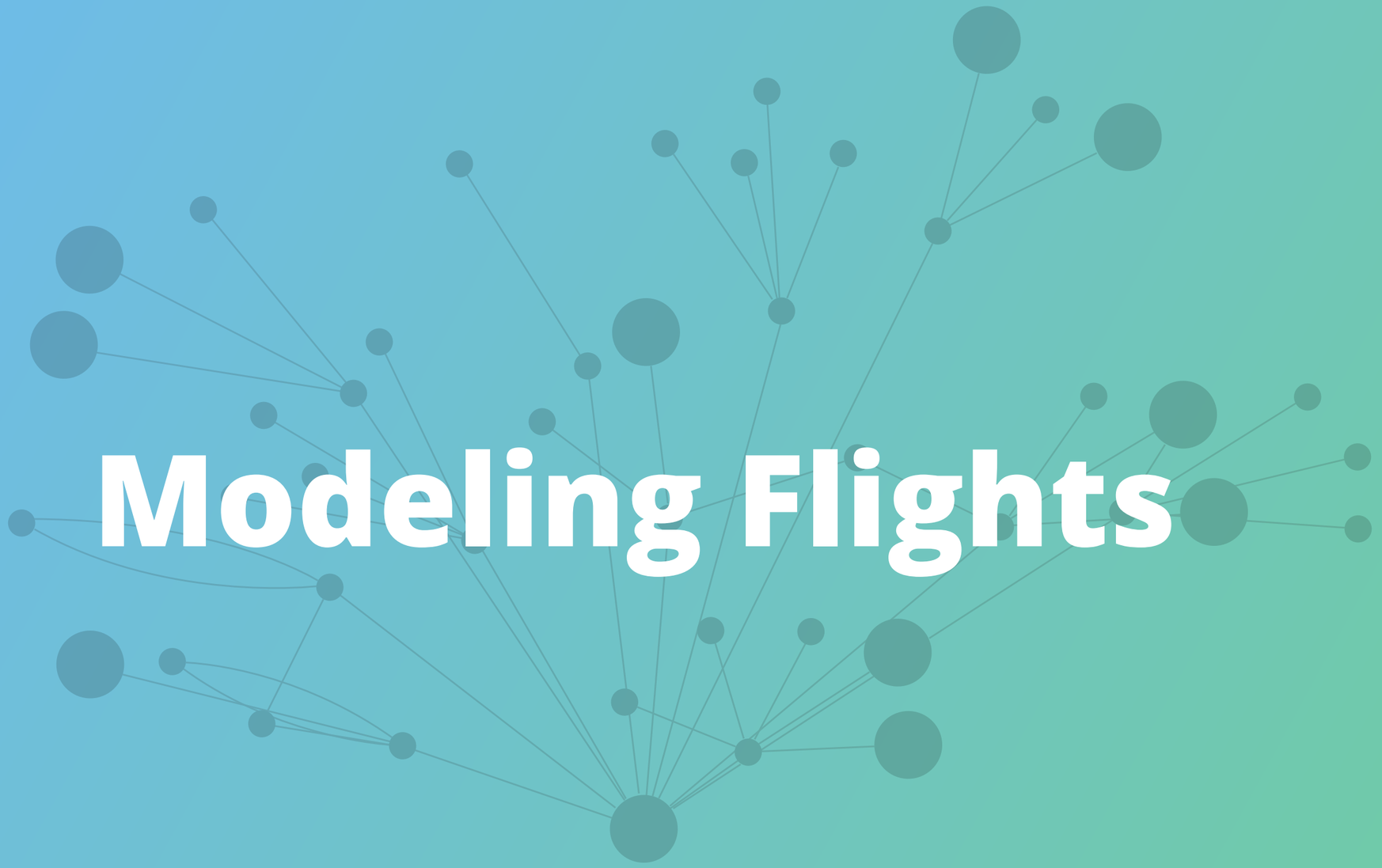

GLOBAL
JAMES BOND DAY

Some Models are Easy but not for all Questions

Should Roles be their own Node?



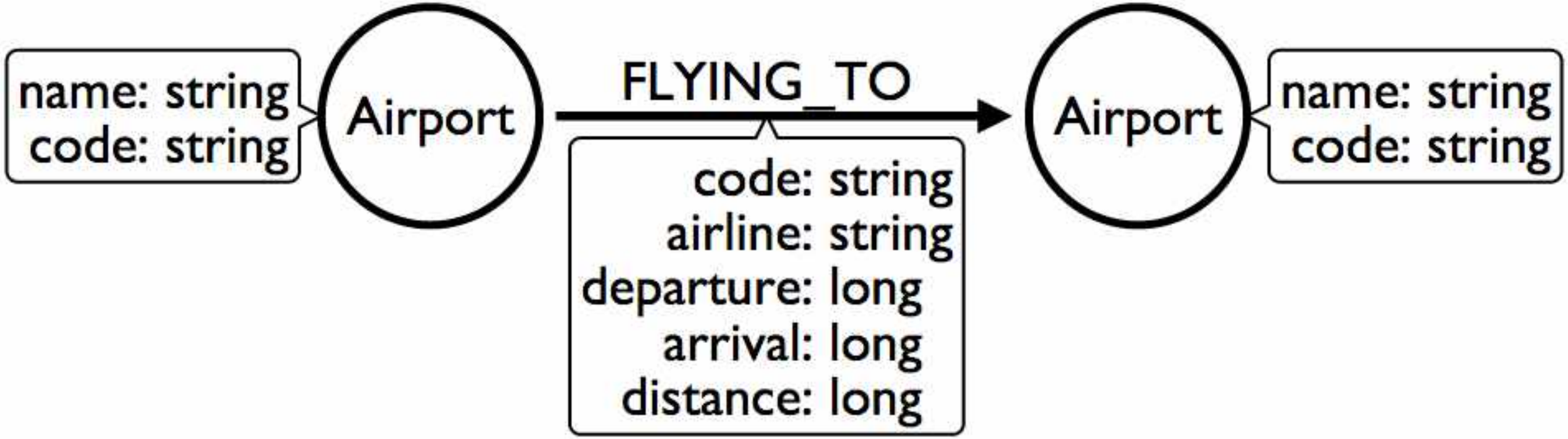
Modeling Flights



neo4j

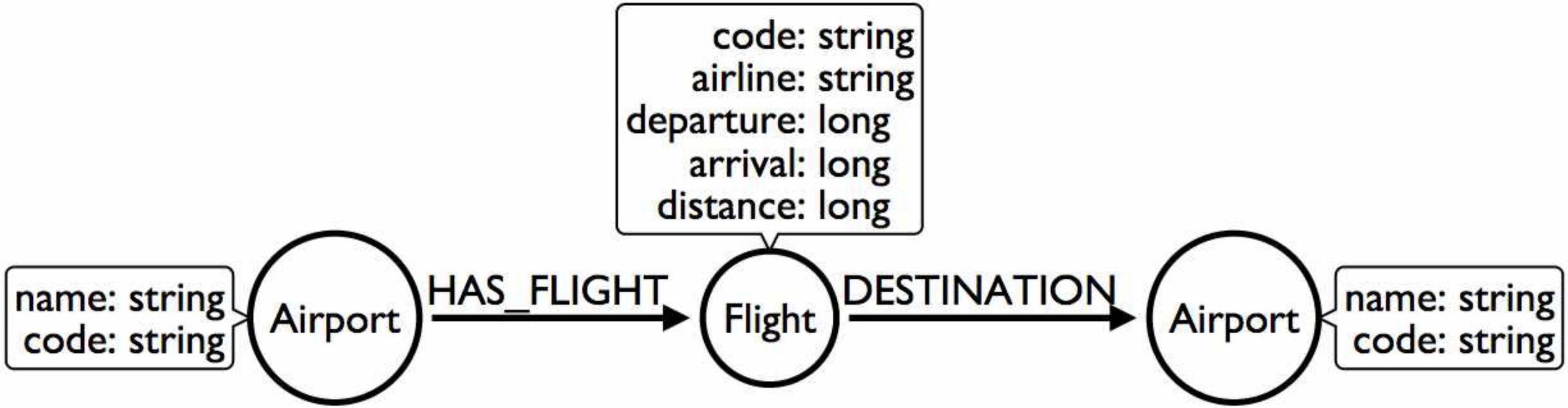
How do you model Flight Data?

Airports Nodes with Flying To Relationships



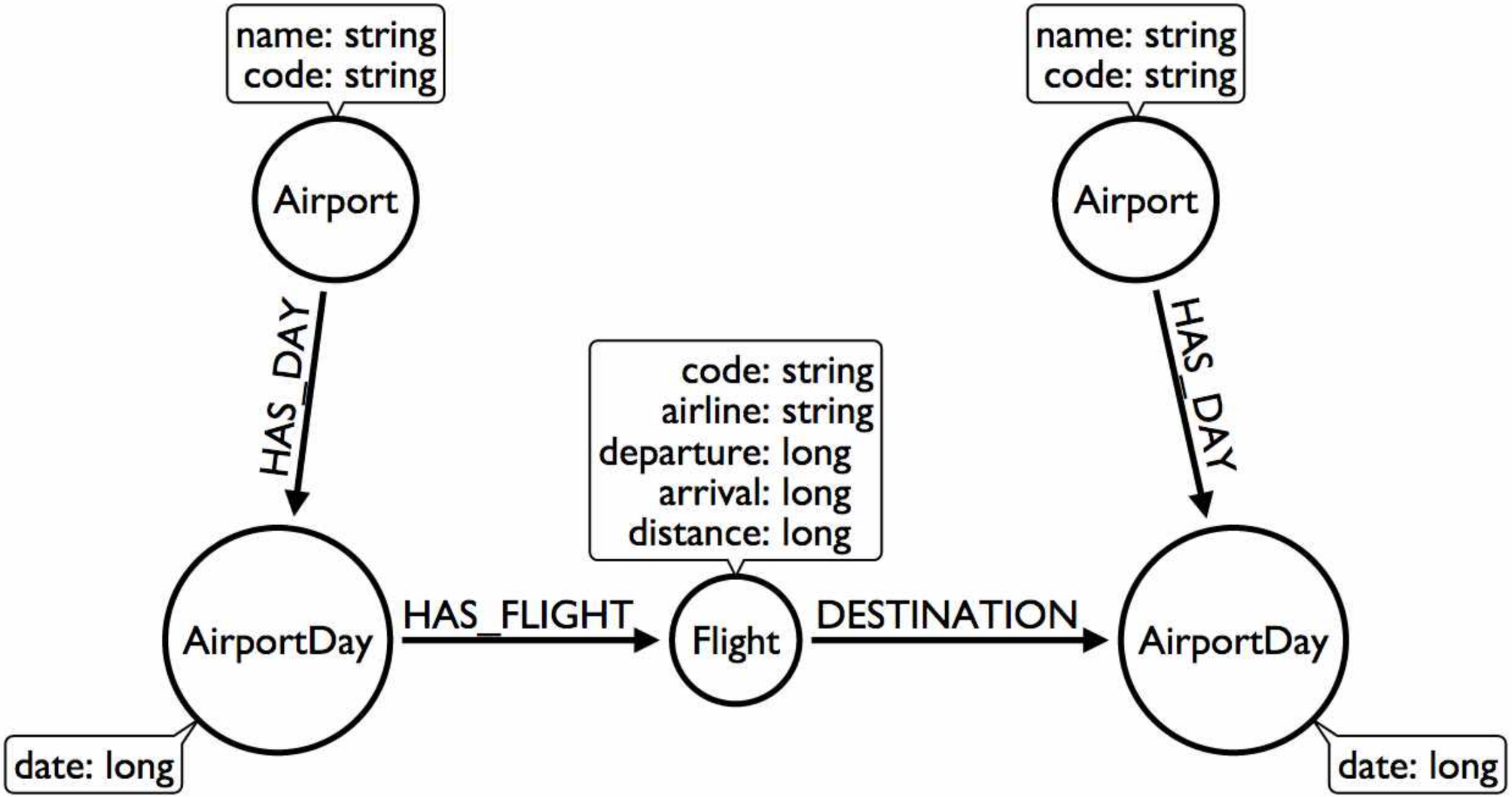
How do you model Flight Data?

Maybe Flight should be its own Node?



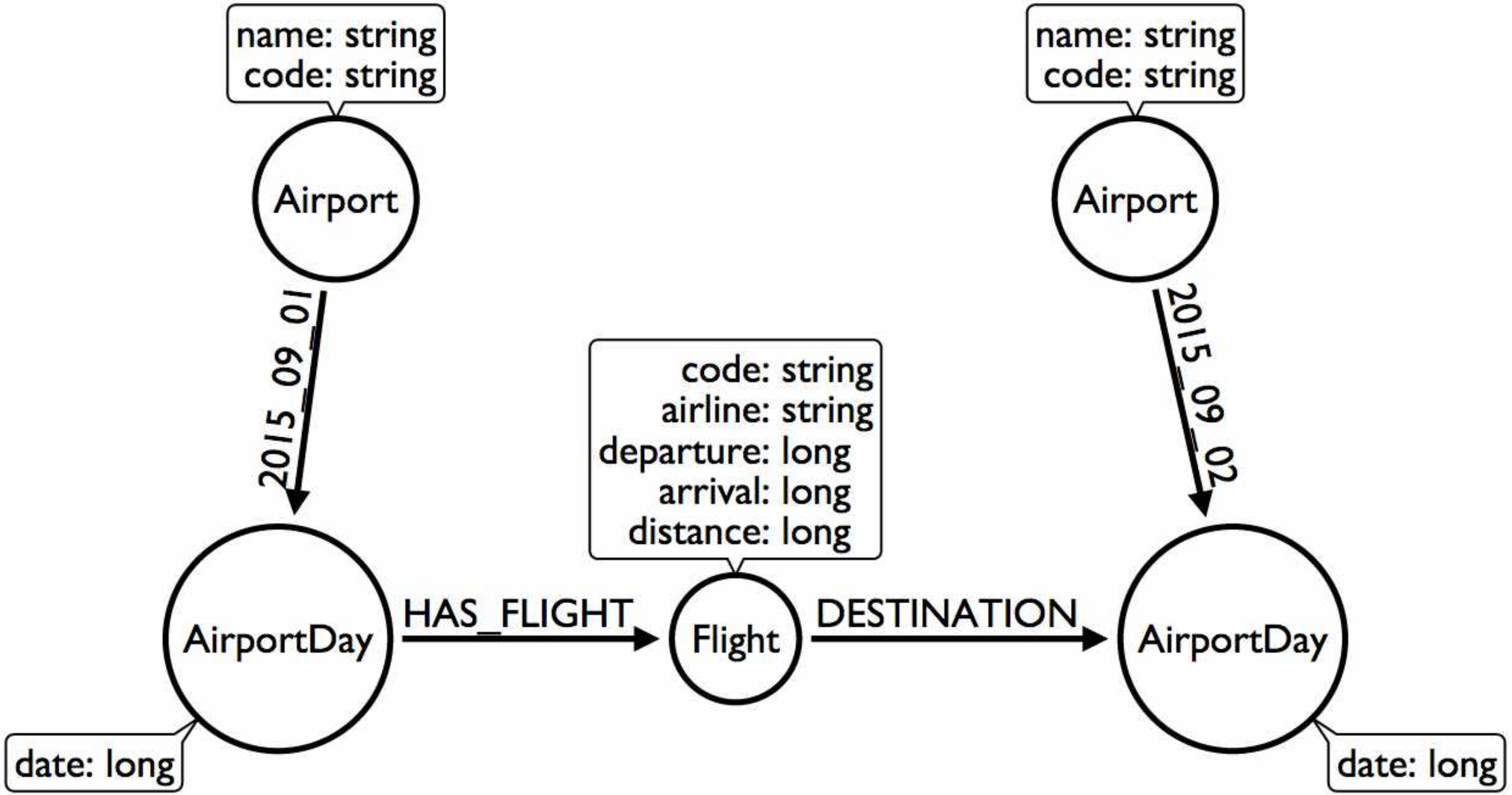
How do you model Flight Data?

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



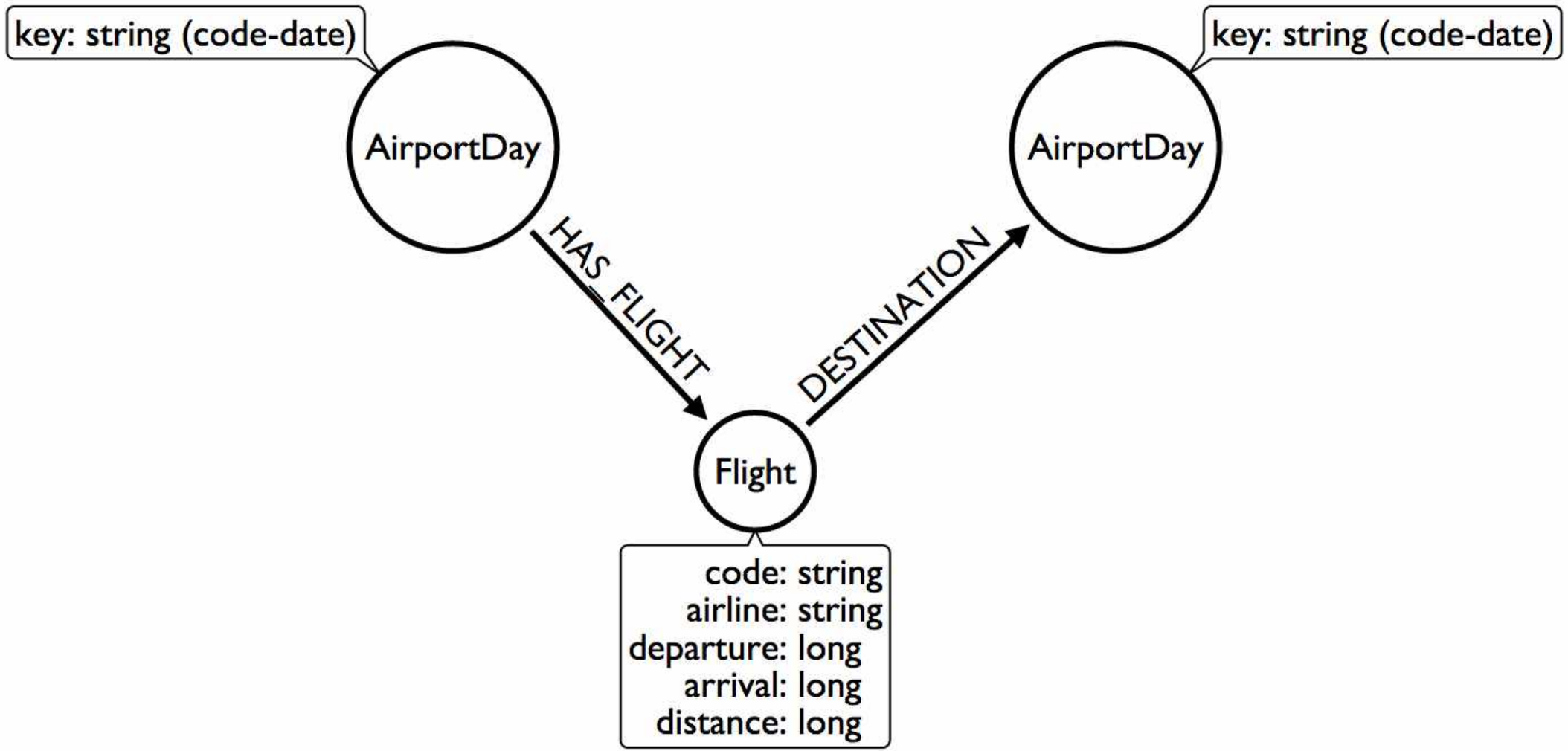
How do you model Flight Data?

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



How do you model Flight Data?

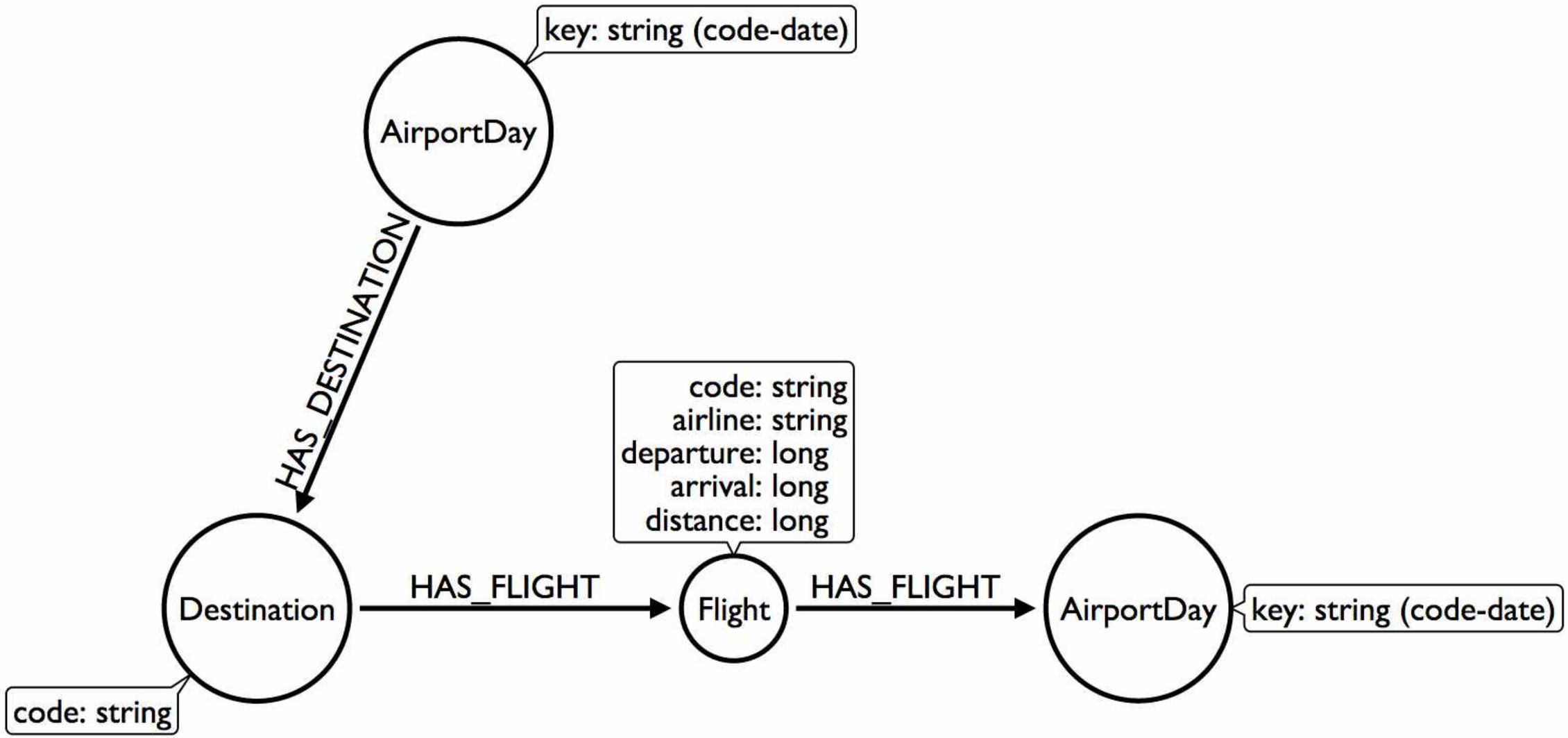
We don’t need Airports if we model this way!



Lets get Creative

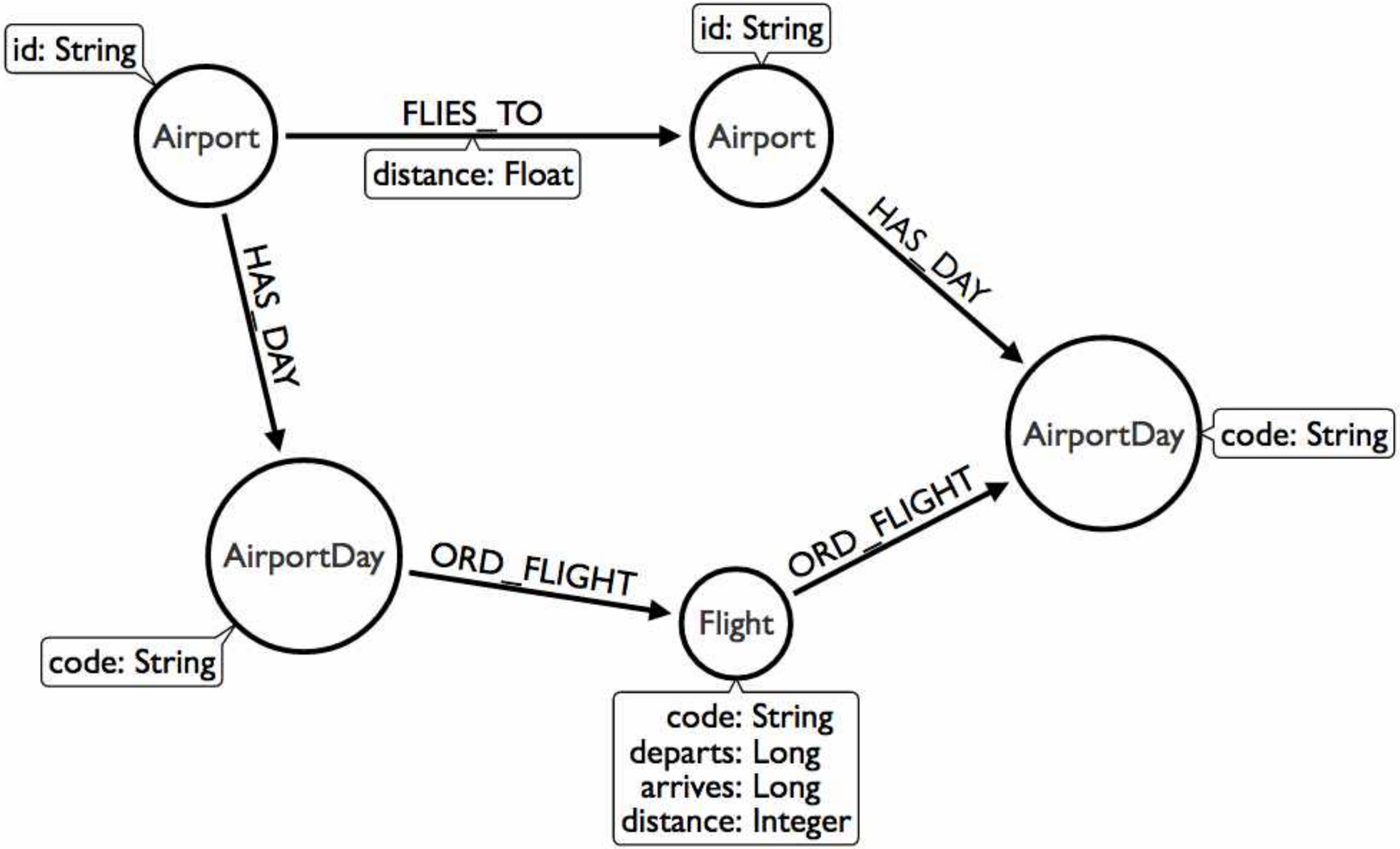
How do you model Flight Data?

Group Destinations together!

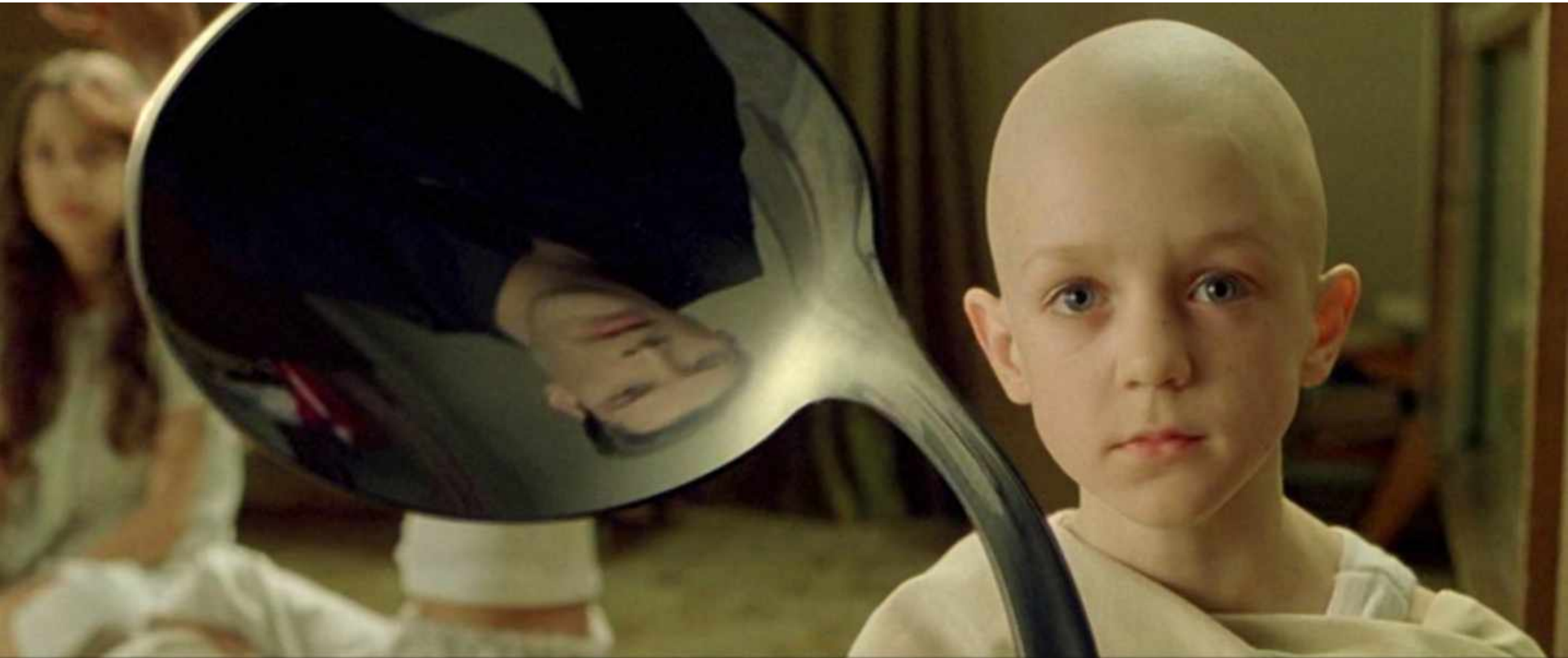


How do you model Flight Data?

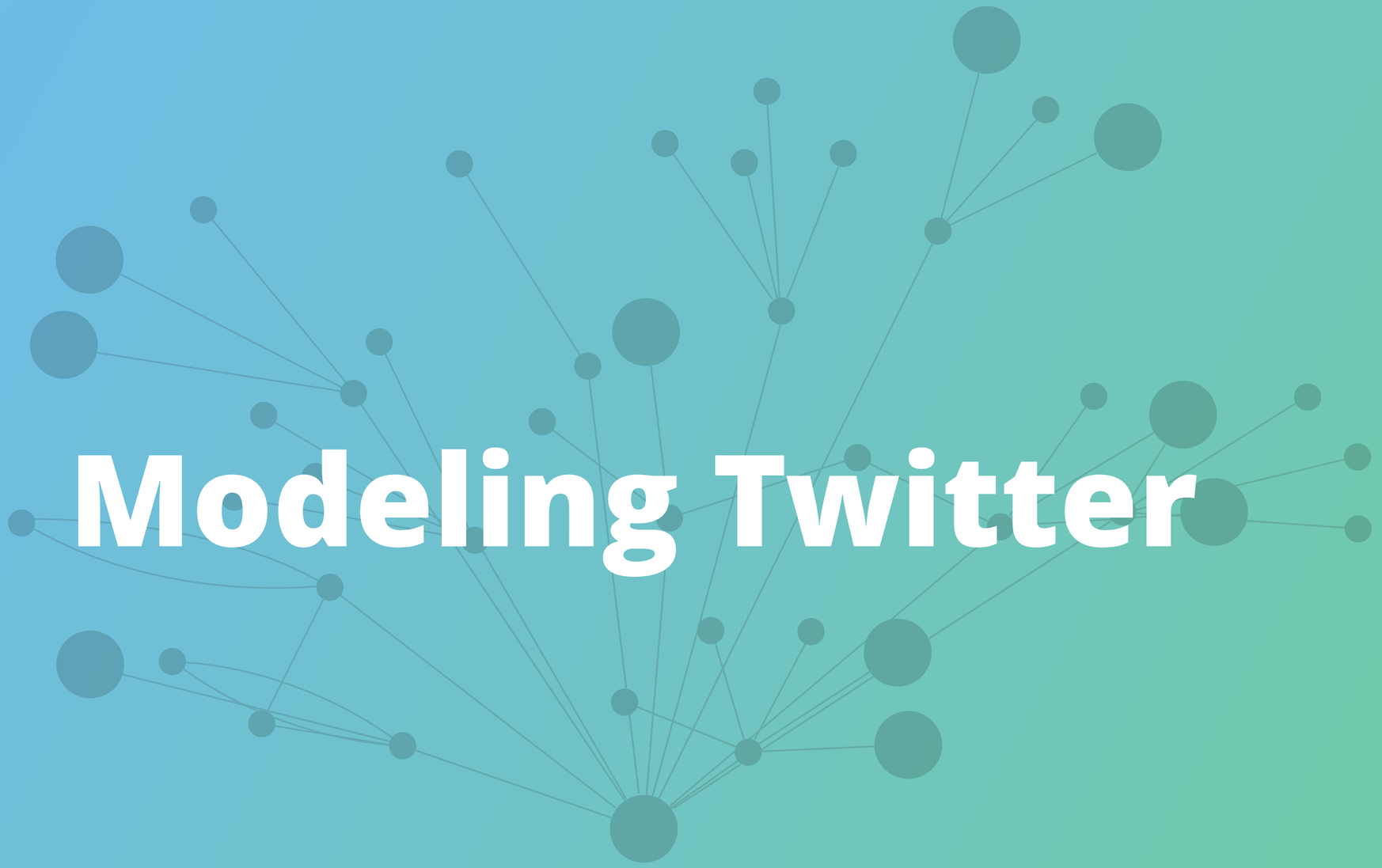
OMG WAT!



Do ~~*not*~~ try and bend the data. That's ~~*im*~~possible.



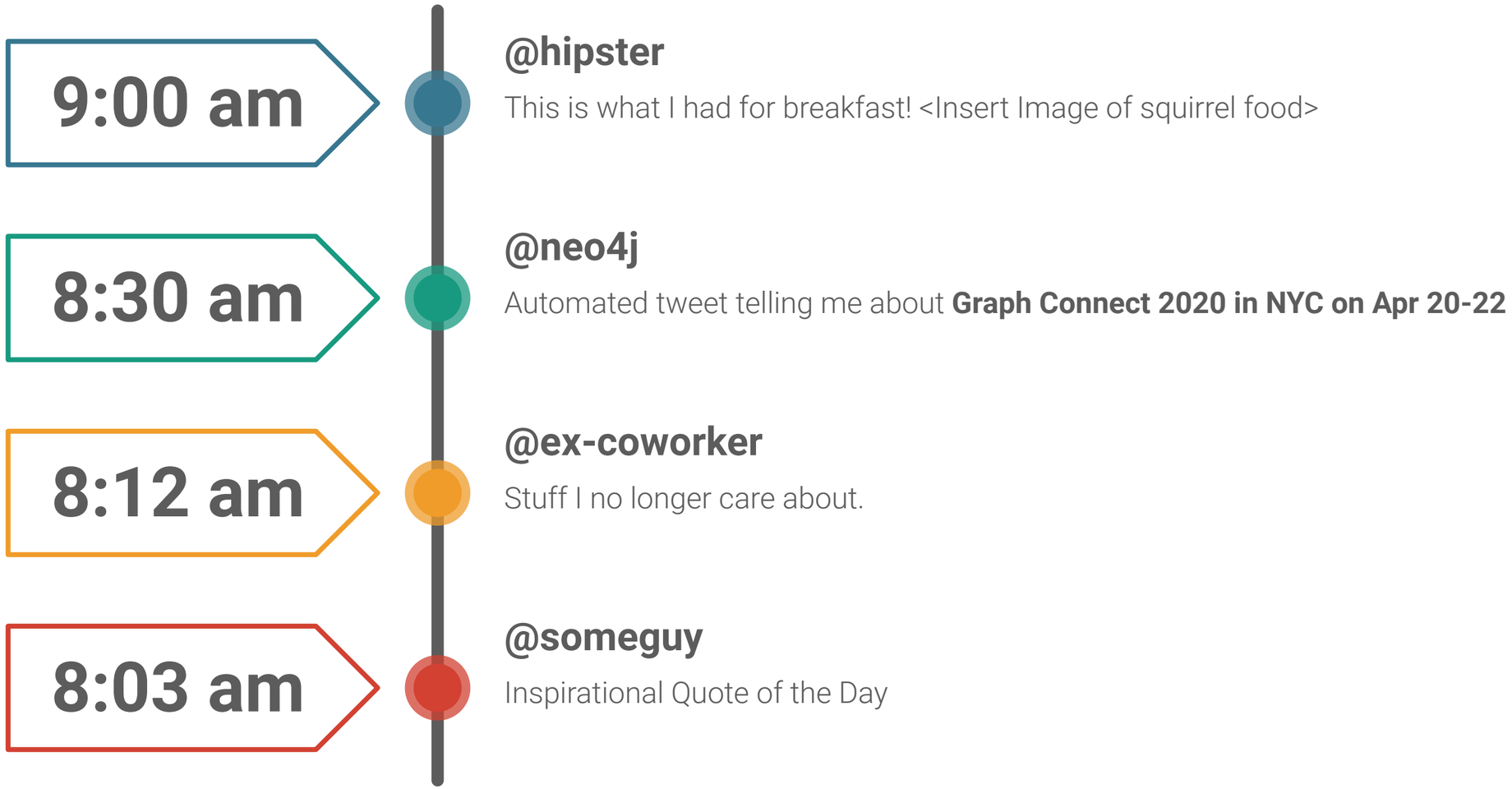
Modeling Twitter



neo4j

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

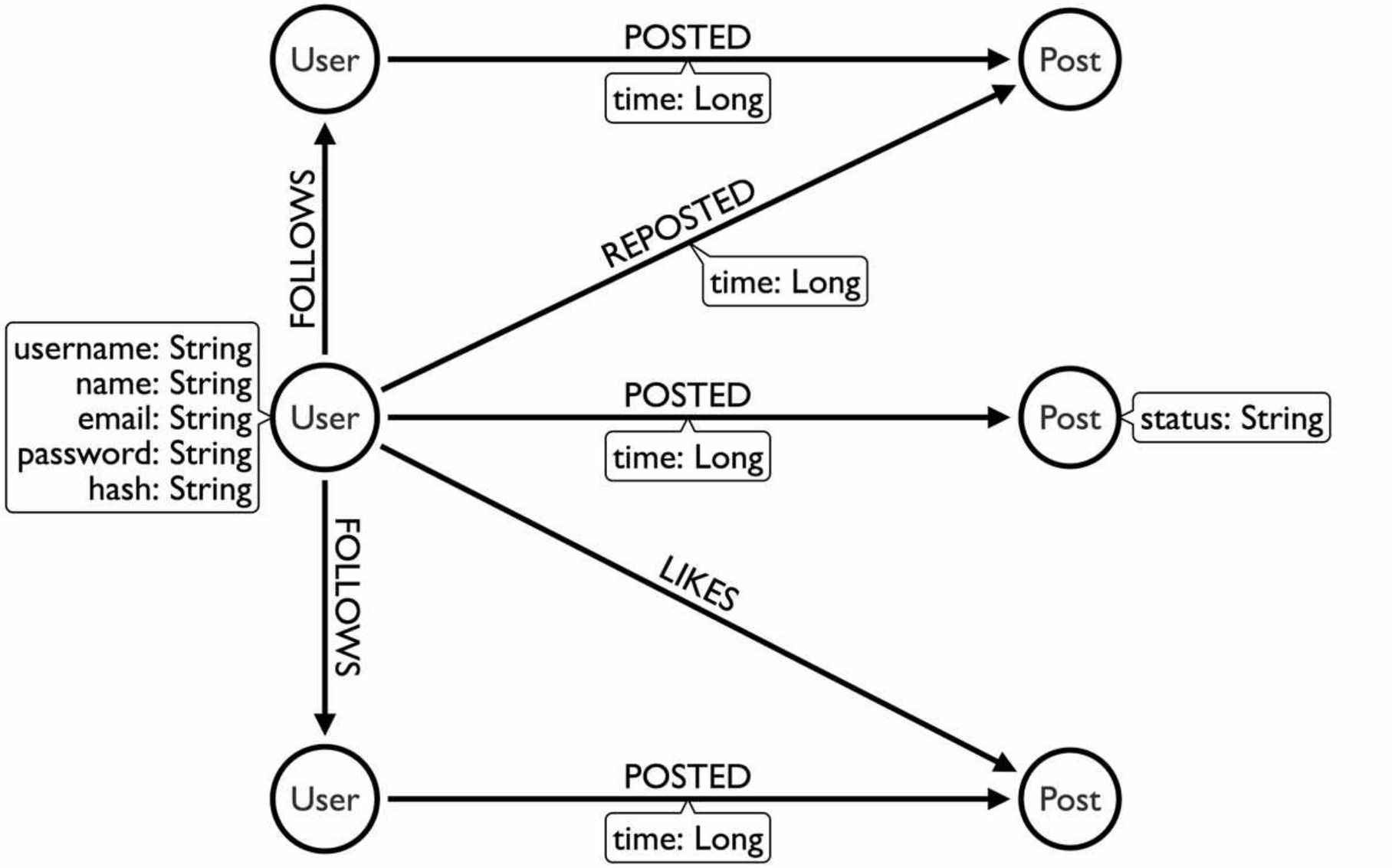
```php
foreach($followers as $fid) {
    $r->lpush("posts:$fid",$postid);
}
```

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

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

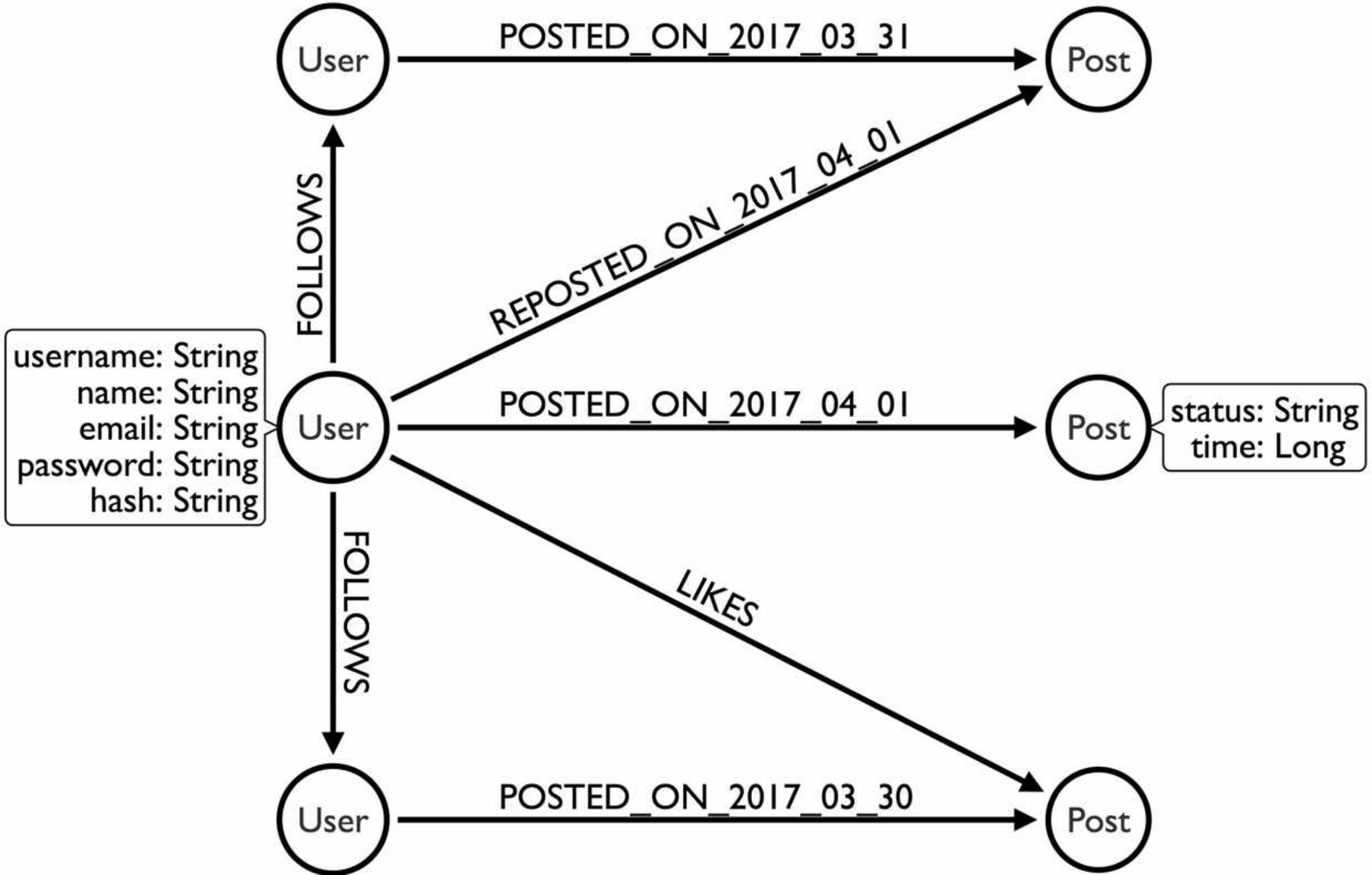
Modeling a Twitter Feed

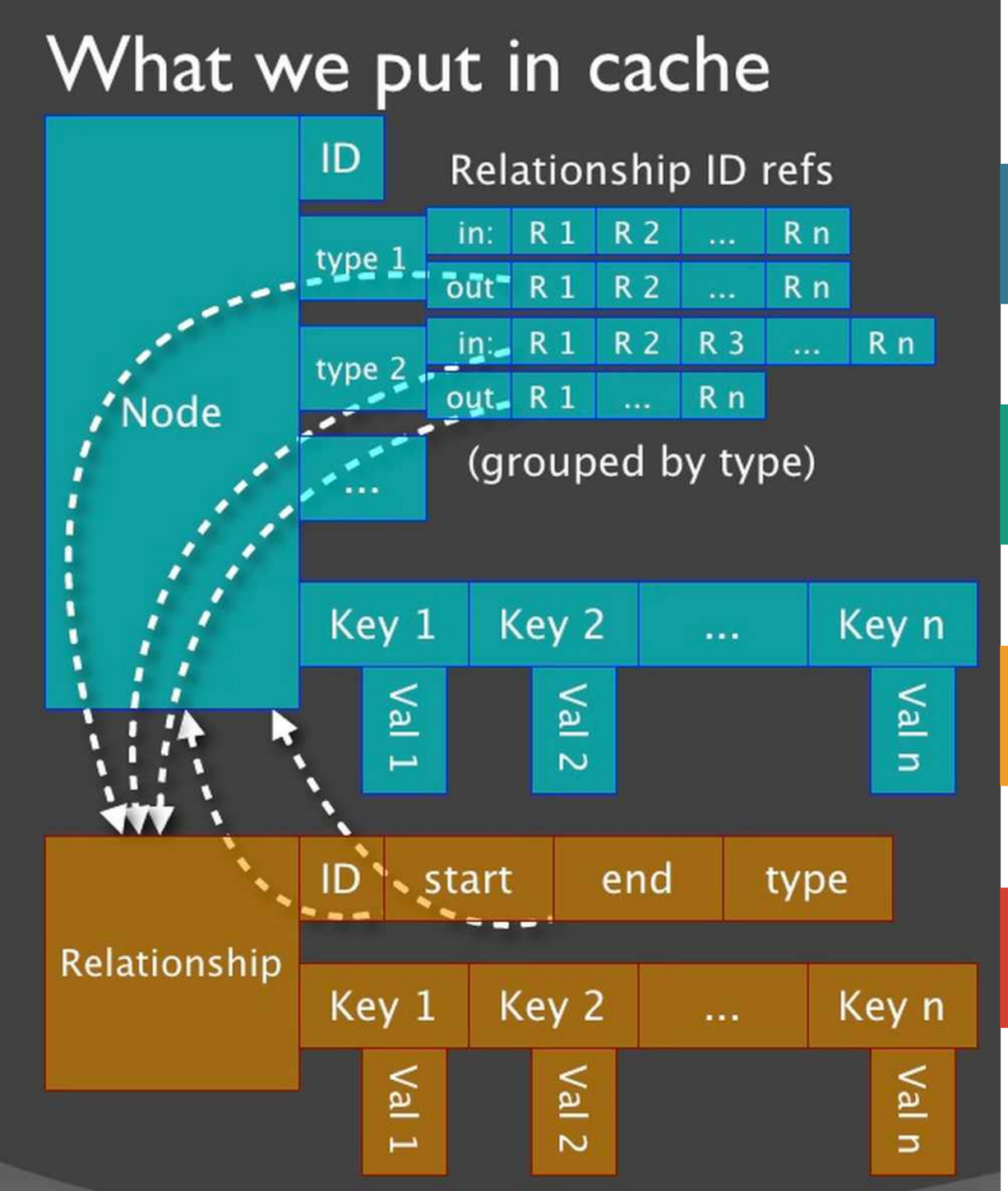
The Wrong Way



Modeling a Twitter Feed

A Better Way





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

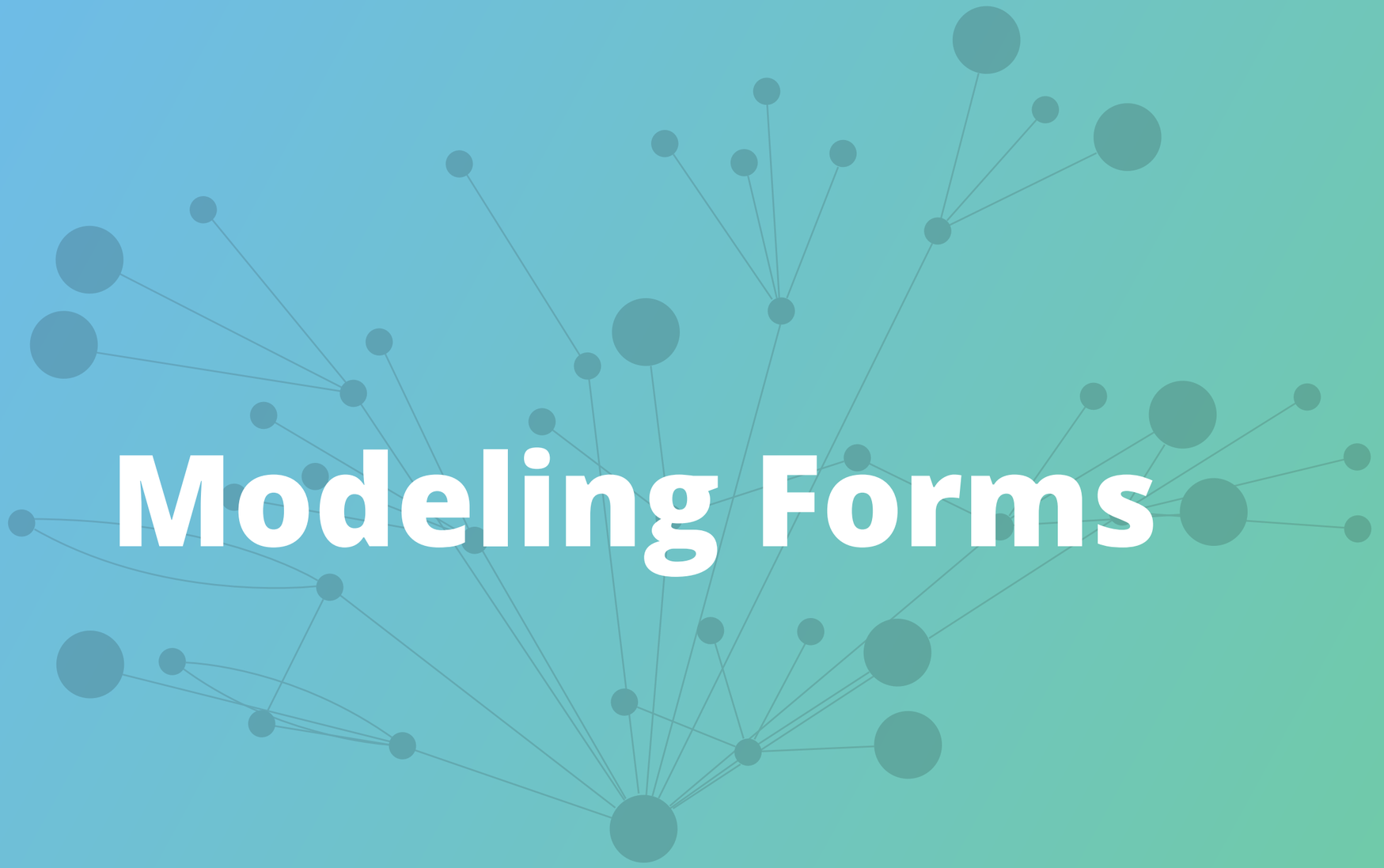
…and the database scales with them.

…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

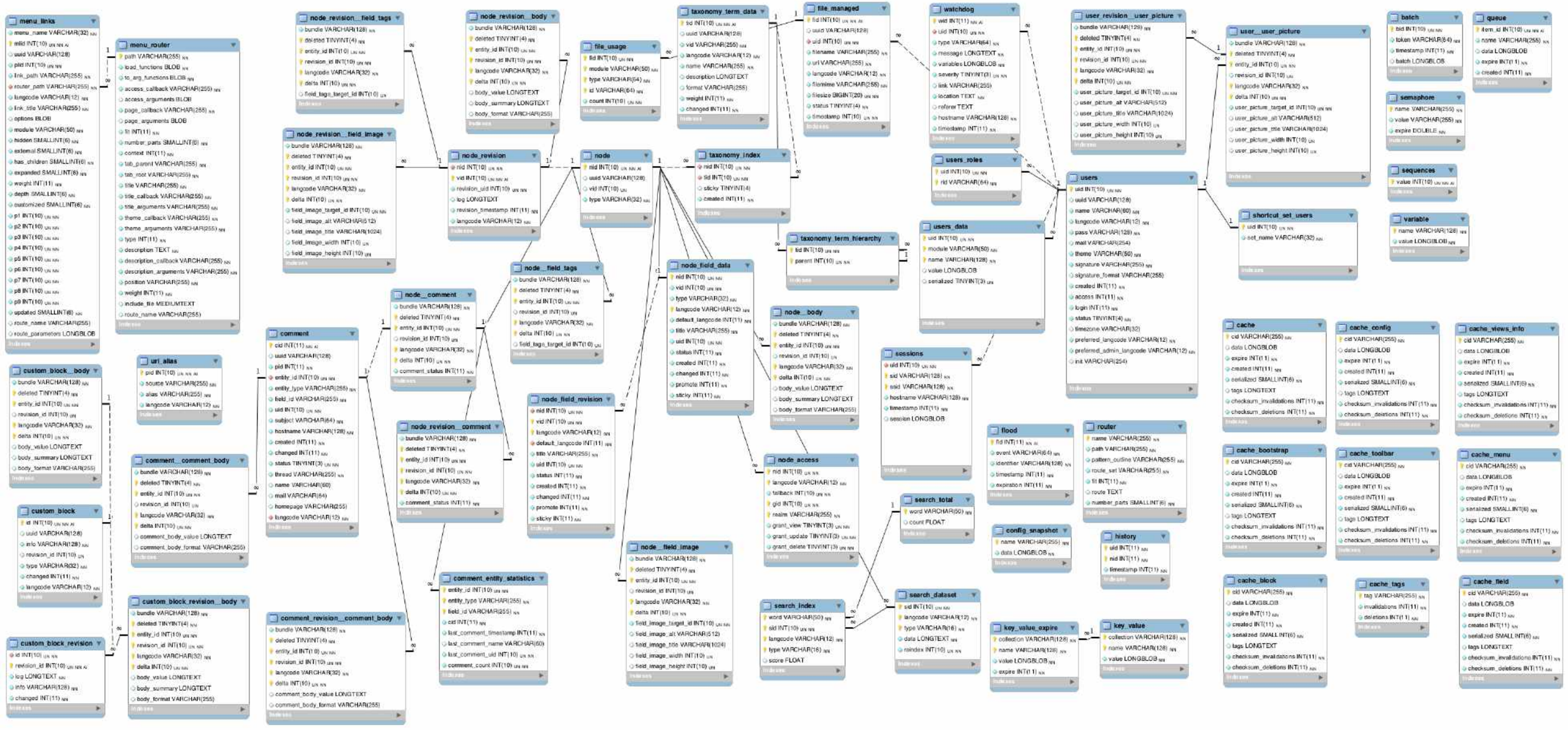


neo4j

Fluid Schema

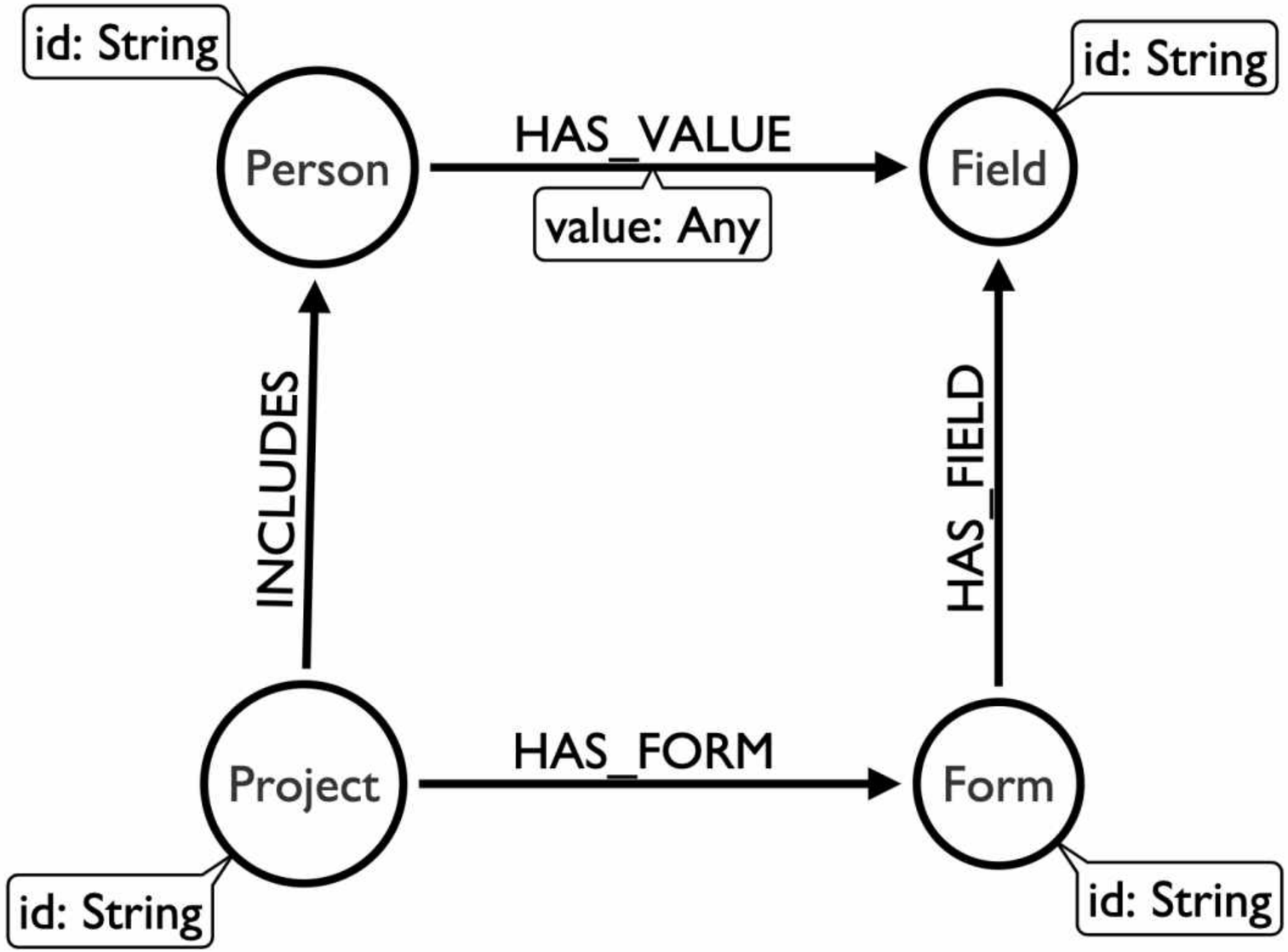
How many **arms** does the T1000 have in this picture?

No more “Wall Sized” Schema Diagrams



Any Form

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

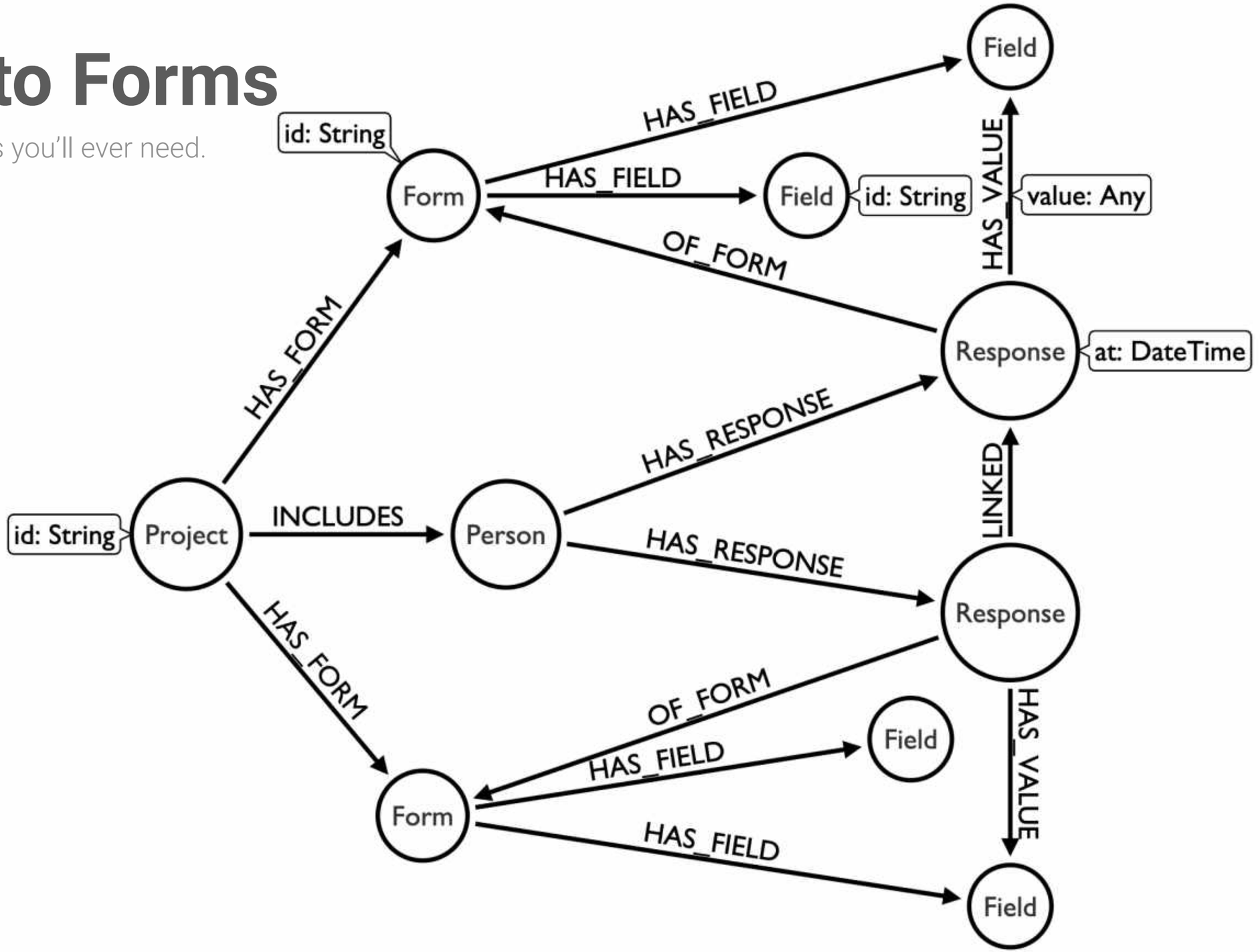


Wait a minute

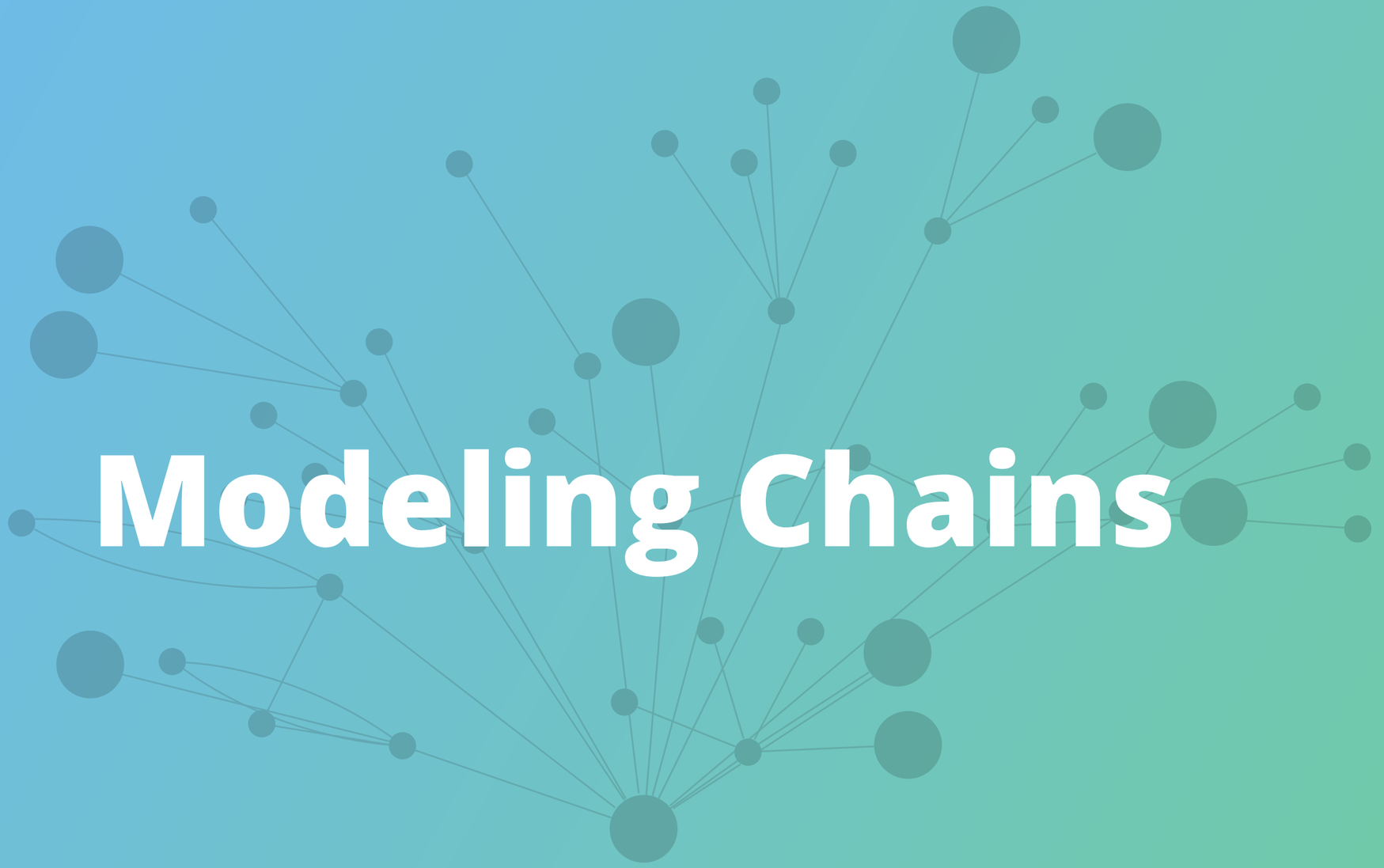
- *What if they fill out lots of forms?*
- *What if they respond to the same form twice?*
- *What if responses to forms are linked together?*
- *Are we missing a concept?*

Responses to Forms

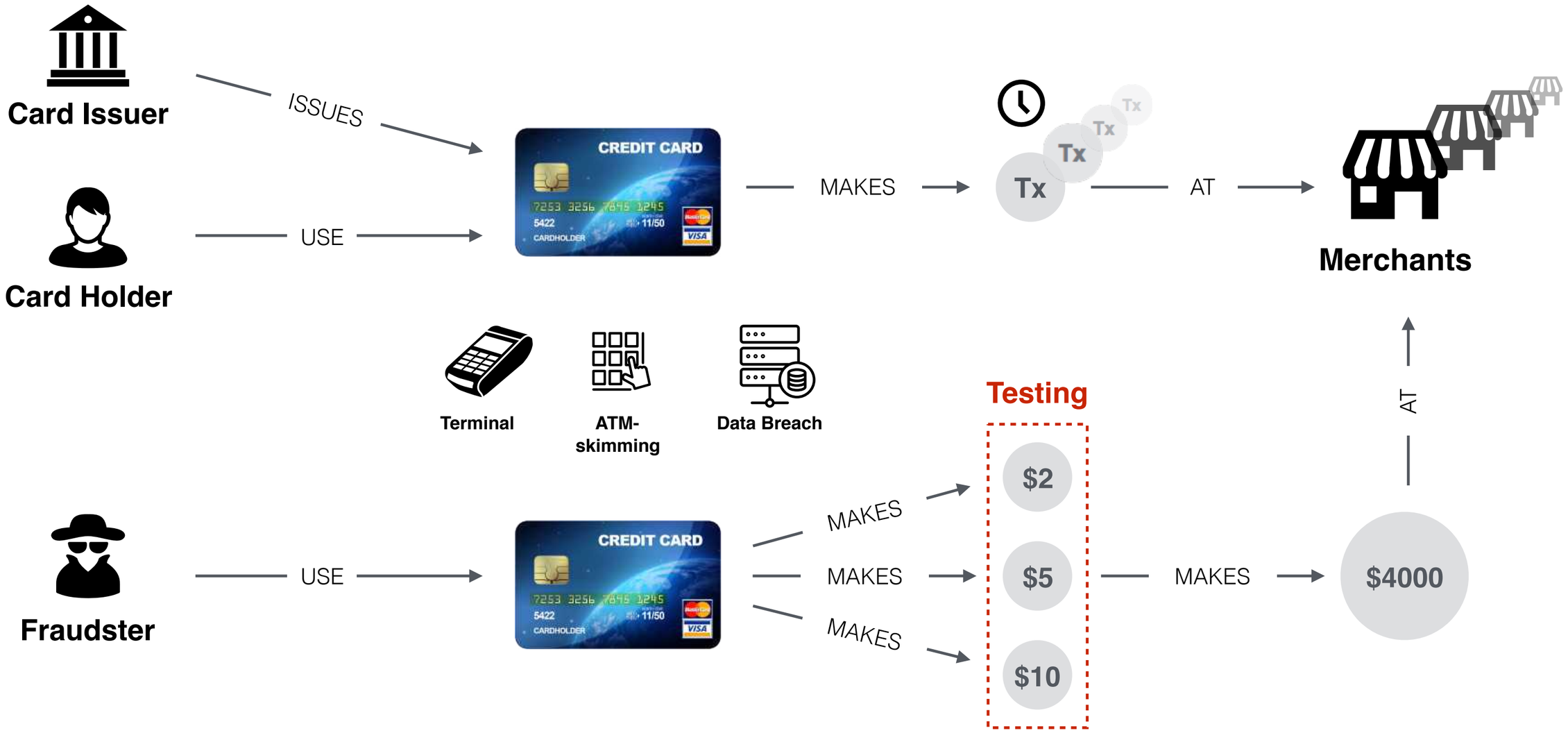
For all forms, all fields, all values you'll ever need.

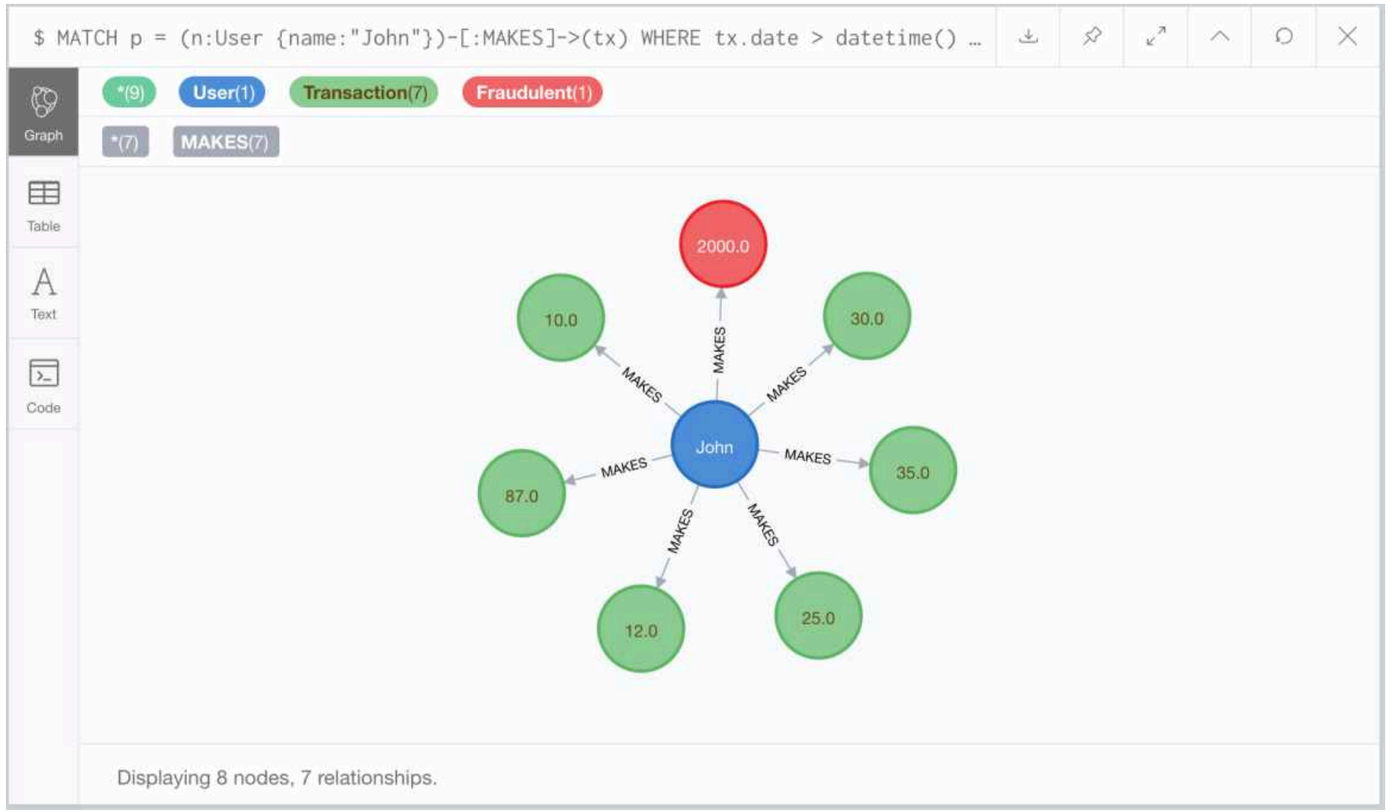


Modeling Chains



neo4j


Card Issuer
ISSUES
Card Holder
USE
CREDIT CARD
7253 3256 7895 1245
5422
11/50
CARDHOLDER
MAKES
Tx
Tx
Tx
Tx
AT
Merchants
Terminal
ATM-skimming
Data Breach
Testing
Fraudster
USE
CREDIT CARD
MAKES
MAKES
MAKES
$2
$5
$10
MAKES
$4000
AT

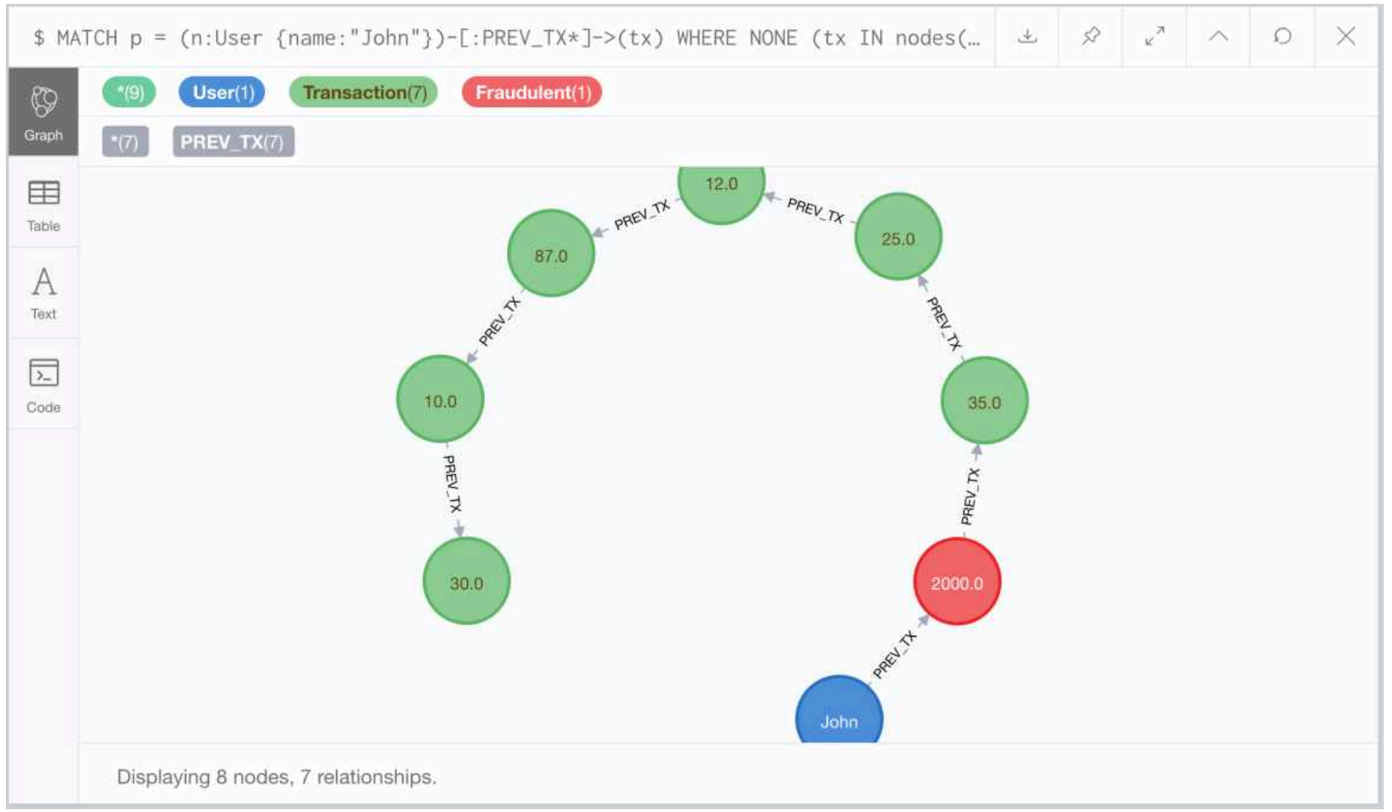


John

Tx Tx Tx Tx Tx Tx Tx Tx Tx Tx Tx Tx

$2000

Computer Store

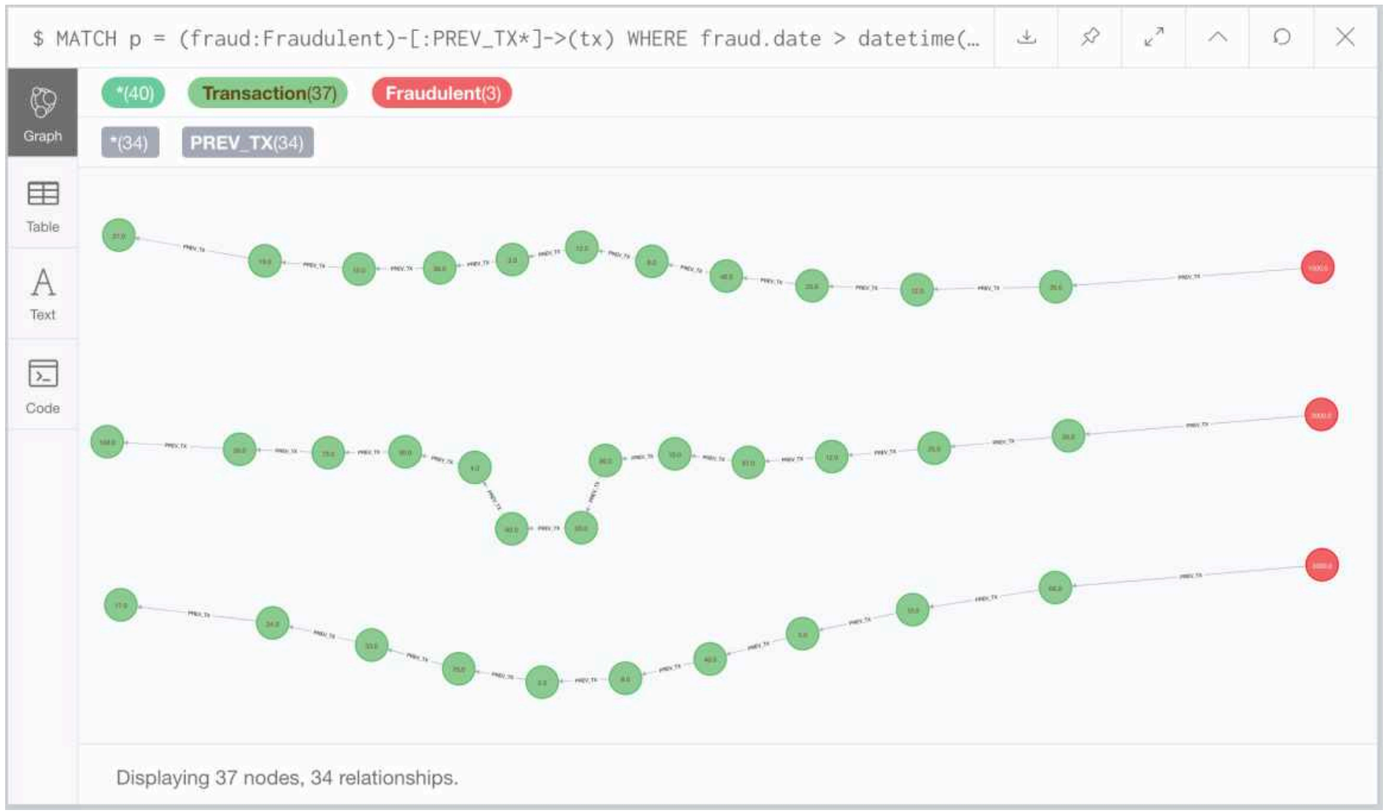


John

Tx Tx Tx Tx Tx Tx Tx Tx Tx Tx Tx Tx

$2000

Computer Store

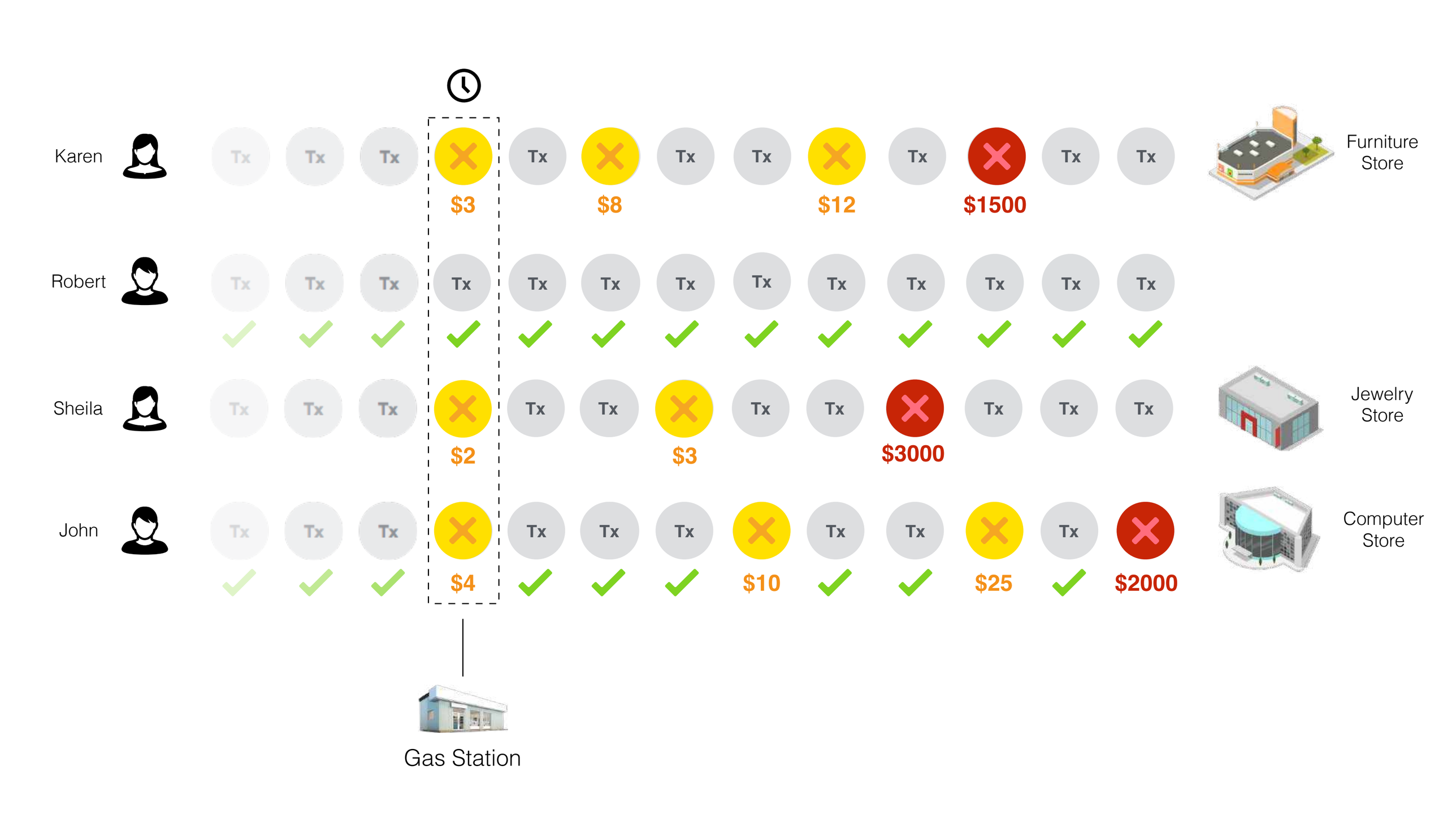


John

Tx Tx Tx Tx Tx Tx Tx Tx Tx Tx Tx Tx

$2000

Computer Store


Karen
Tx
Tx
Tx
$3
Tx
$8
Tx
Tx
$12
Tx
$1500
Tx
Tx
Furniture Store
Robert
Tx
Tx
Tx
Tx
Tx
Tx
Tx
Tx
Tx
Tx
Tx
Tx
Tx
Tx
Sheila
Tx
Tx
Tx
$2
Tx
Tx
$3
Tx
Tx
$3000
Tx
Tx
Tx
Jewelry Store
John
Tx
Tx
Tx
$4
Tx
Tx
Tx
$10
Tx
Tx
$25
Tx
$2000
Computer Store
Gas Station

Hunger Games

neo4j

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!

neo4j