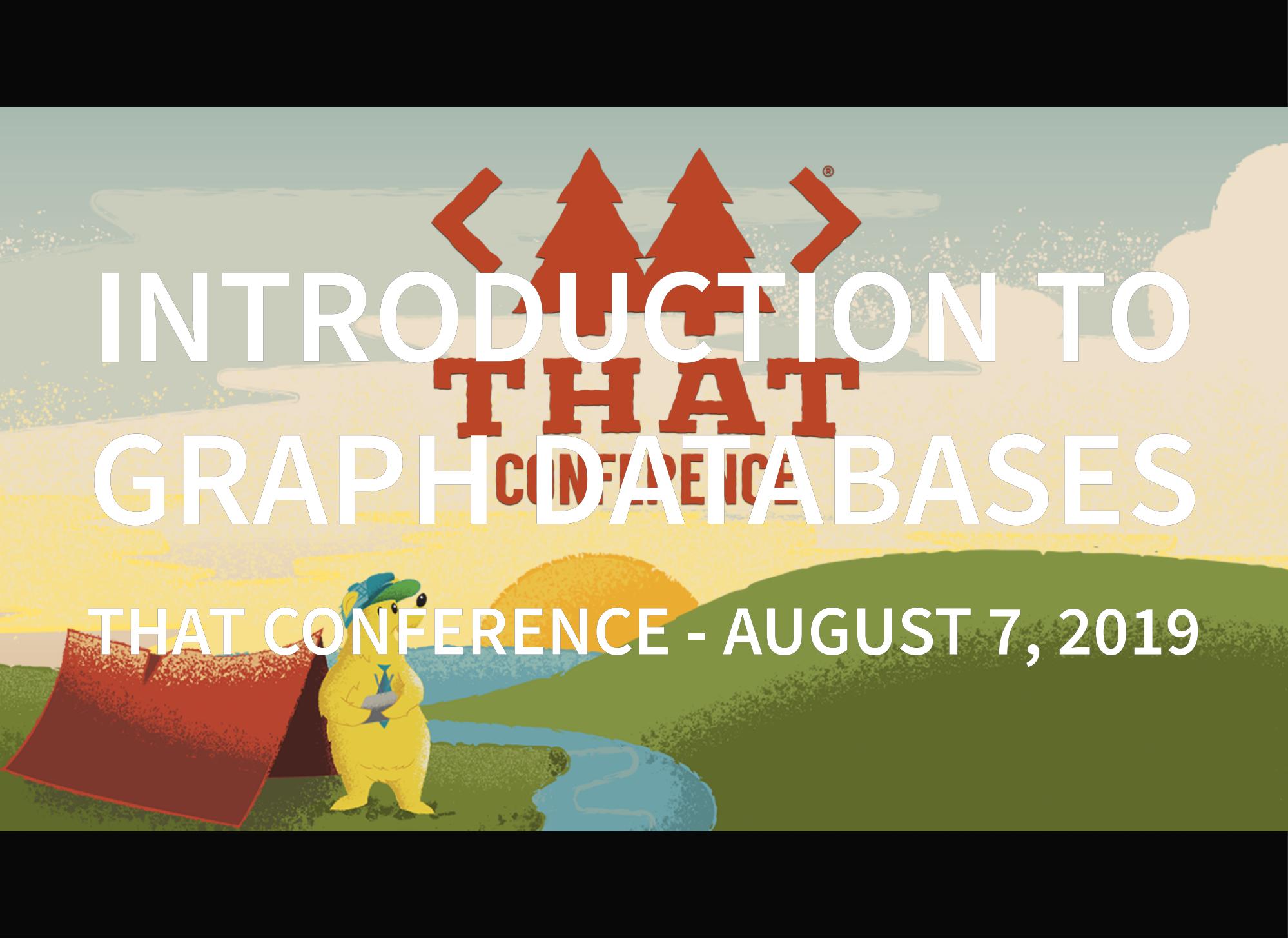
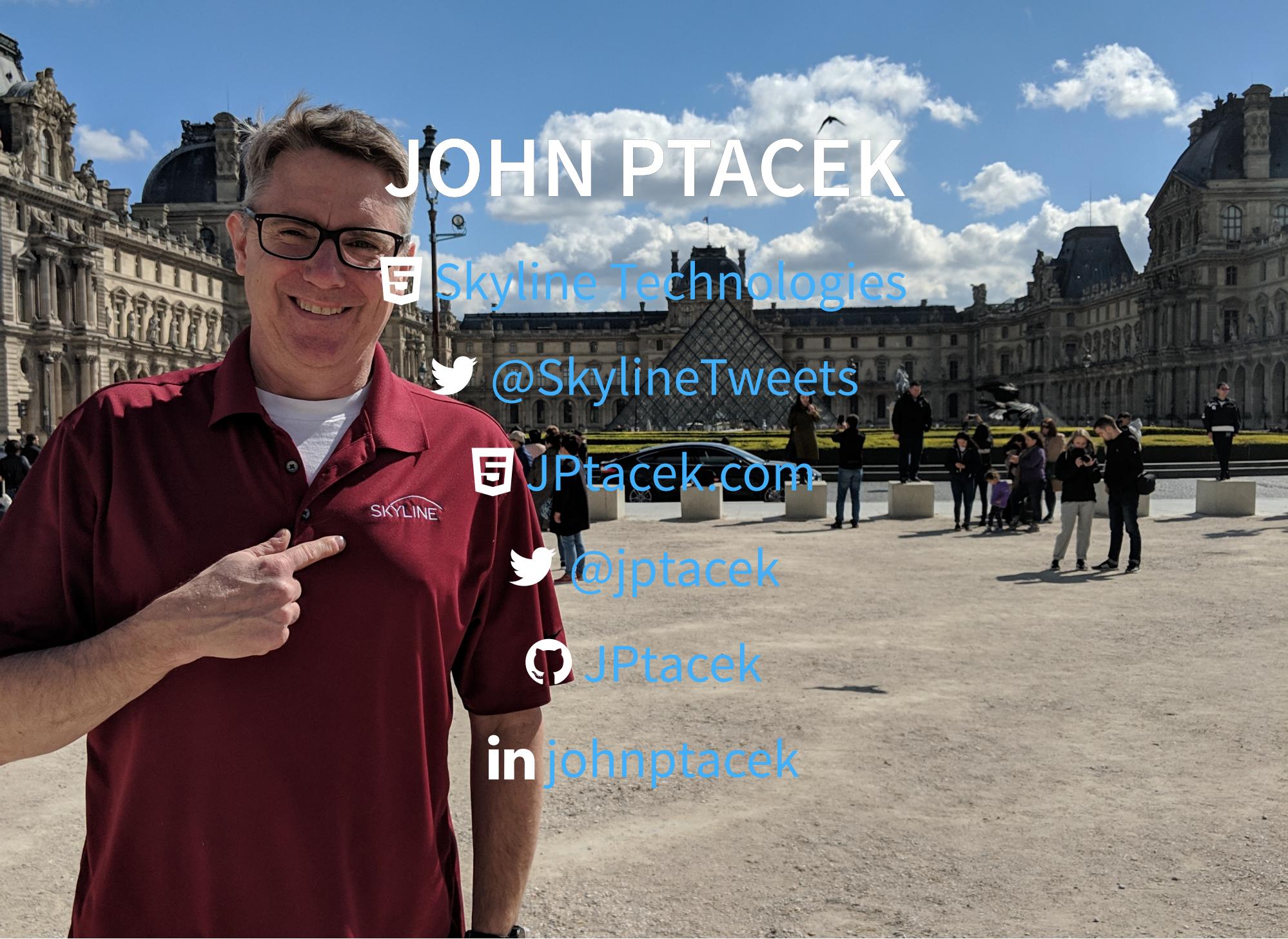


INTRODUCTION TO GRAPH DATABASES



The background features a stylized landscape with rolling green hills, a bright yellow field in the foreground, a red tent on the left, and a blue sky with white clouds. In the center, there are three red pine trees and a small yellow bird-like character wearing a blue cap and holding a blue ribbon.

THAT CONFERENCE - AUGUST 7, 2019



JOHN PTACEK

S Skyline Technologies

 @SkylineTweets

 JPtacek.com

 @jptacek

 JPtacek

in johnptacek



THANK YOU, THAT CONFERENCE SPONSORS!

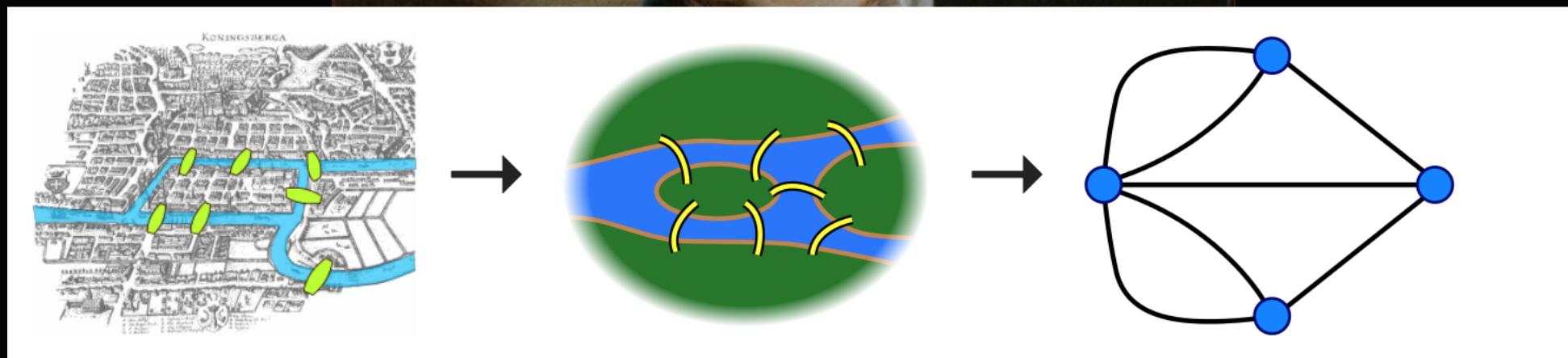


AGENDA

- What is a Graph?
- What is a Graph Database?
- SQL or Graph Database
- In the Real World
- Querying
- I Did this Thing

WHAT IS A GRAPH?

- Seven Bridges of Königsberg
- Walk through city and walk across each bridge once
- Euler creates graph theory



GRAPH THEORY

6

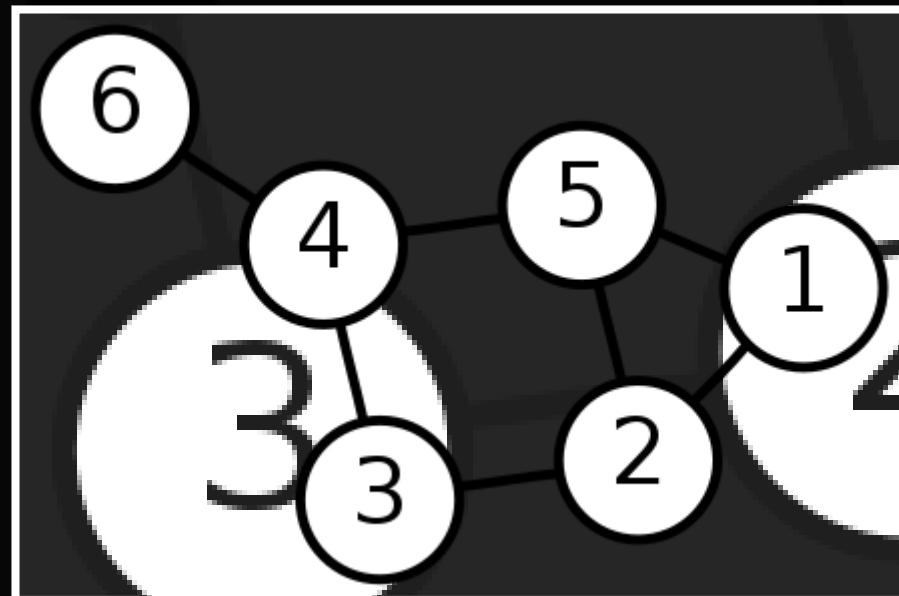
construct to model relationships between objects
vertex (node, point)

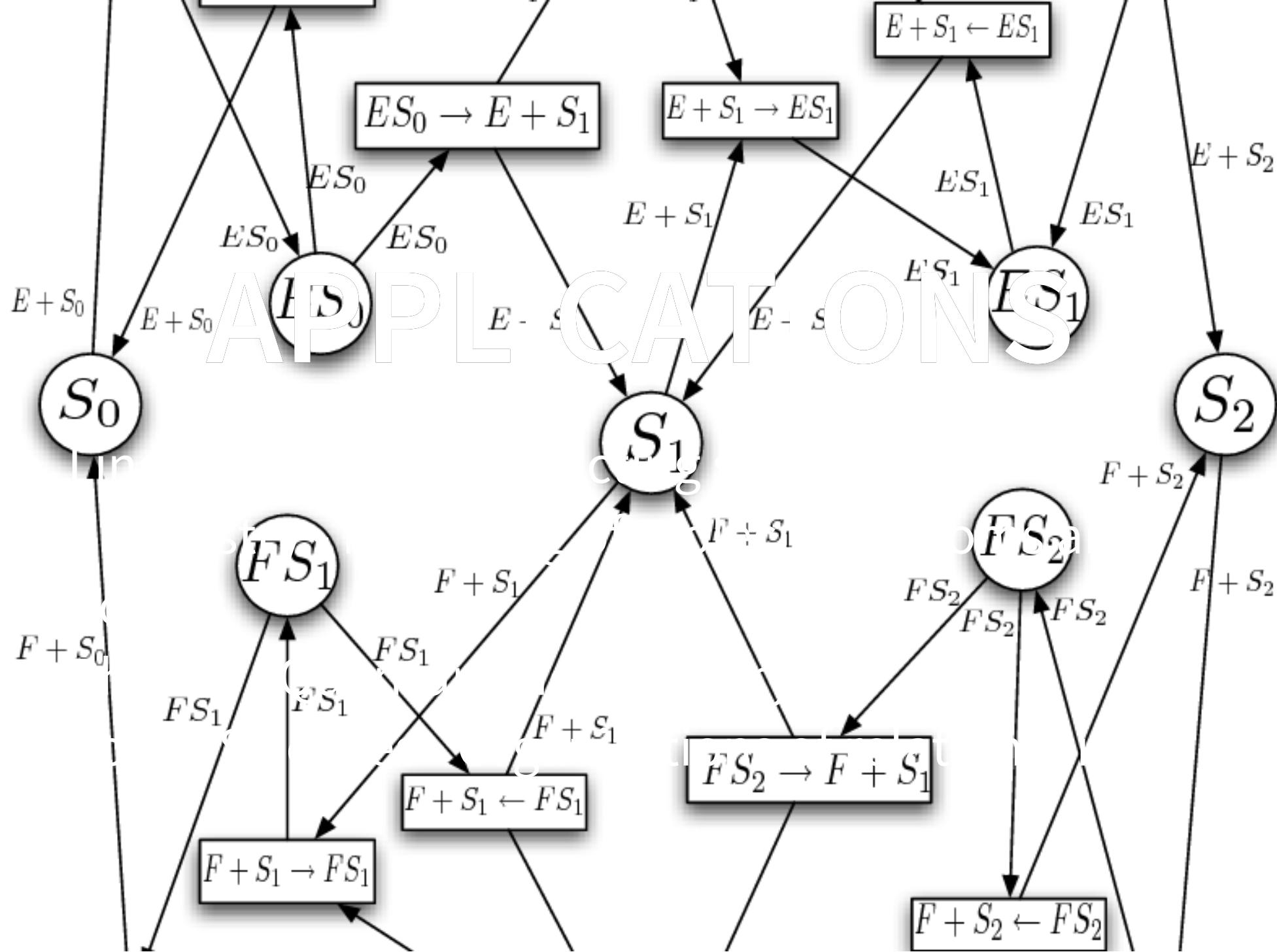
- Edges (links)
- Direct Graphs have direction
- And then it all happens

4

5

1

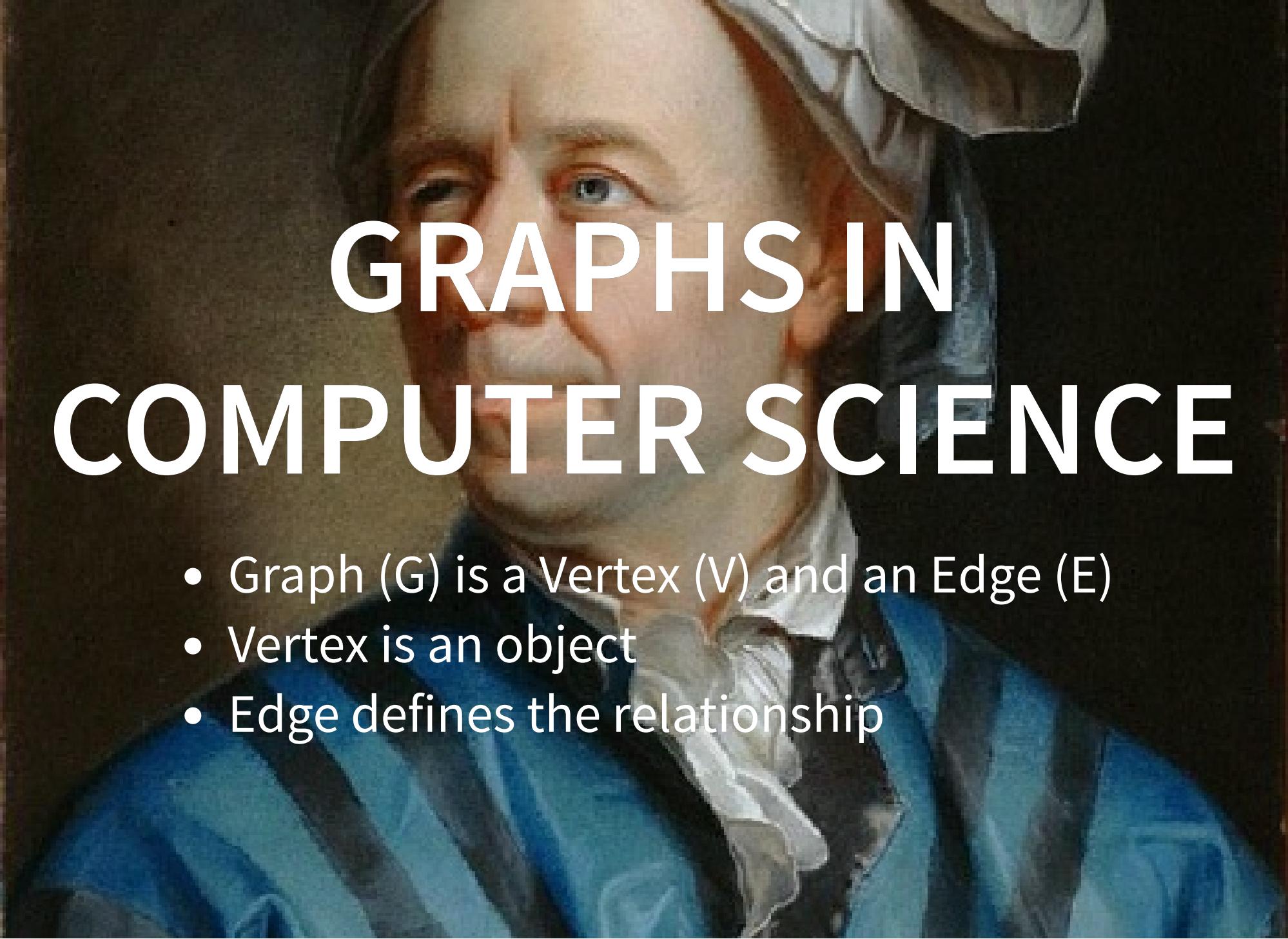




WHAT IS A GRAPH DATABASE?

The graph paradigm goes well beyond databases and application development; it's a reimagining of what's possible around the idea of connections.

Neo4j blog

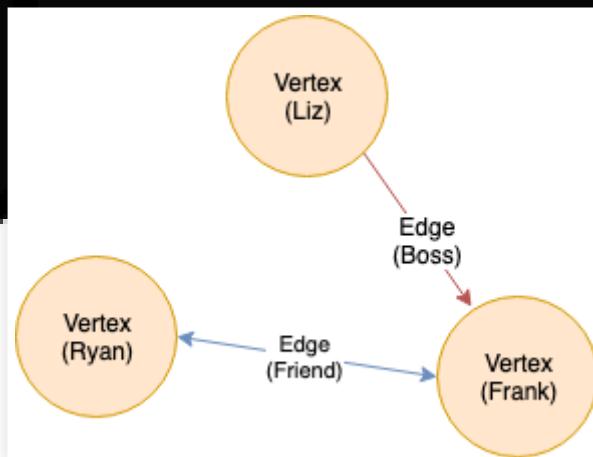
A close-up portrait of a man with white hair, wearing a blue shirt. He has a thoughtful expression, looking slightly upwards and to the side. His hands are clasped together in front of him, resting on what appears to be a keyboard or a set of keys.

GRAPHS IN COMPUTER SCIENCE

- Graph (G) is a Vertex (V) and an Edge (E)
- Vertex is an object
- Edge defines the relationship

6 SAMPLE GRAPH

- Ryan and Frank know each other
Connection is one way
- Liz and Frank know each other, Liz is the boss
Connection is one way



GRAPH THEORY -

Vertex

VERTICES

label: Person

ID: Hamilton

Properties: {

birthday: 1/1/1755

death: 7/12/1704

intro_text: First

secretary...}

- Vertices can have a label (Person, Event)
- A point in a graph
- Vertices can have arbitrary key-value properties
(Birth, Death, Intro_text)

Zero or more edges meet at a Vertex

6



1

2

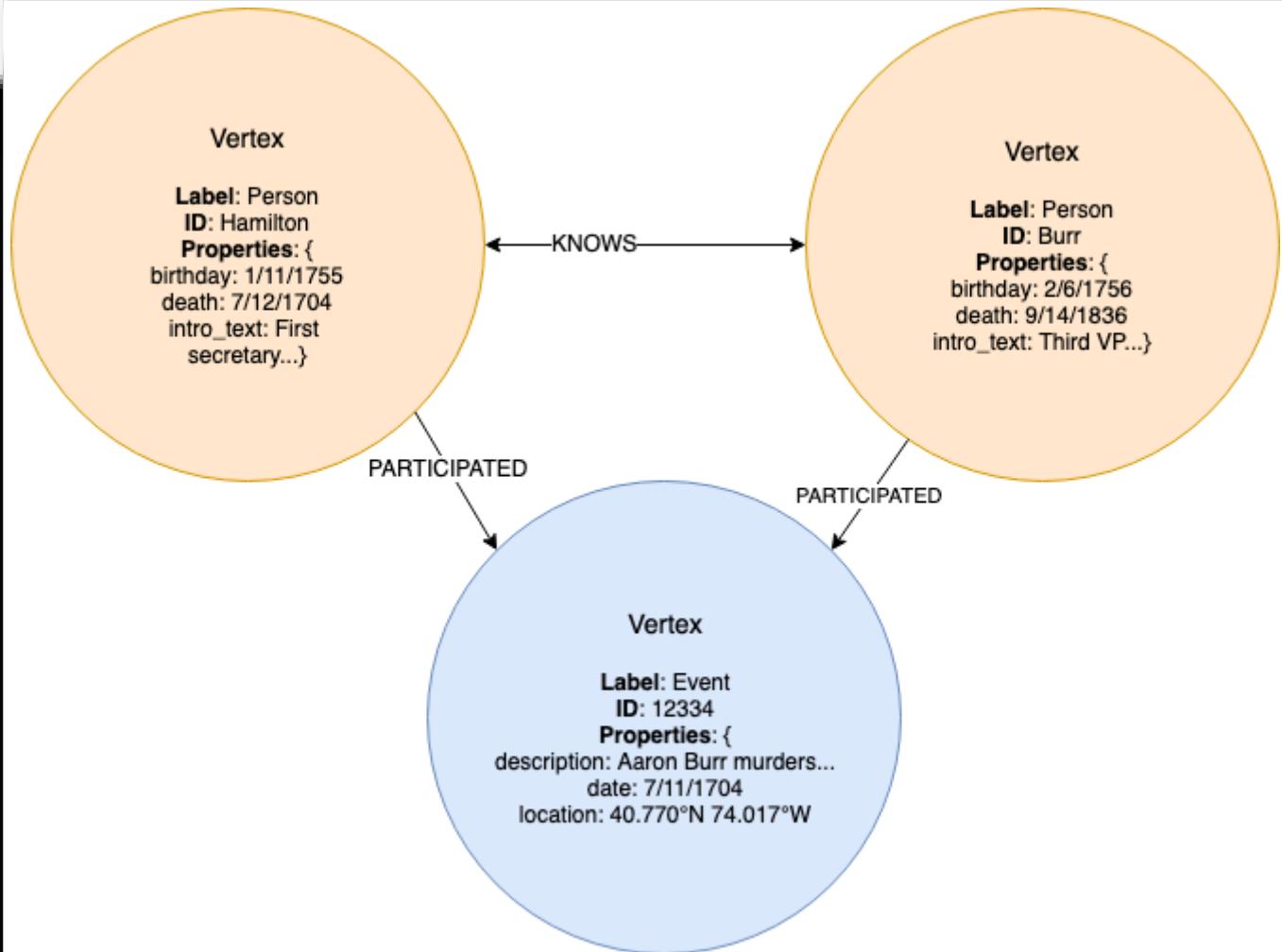
3

6 GRAPH THEORY -

4 EDGES

- Edges define the relationships between Vertices
- Edges can have a label
- Edges may have multiple set of key-value pairs for properties (which is mutable)
- A relationship between two vertices (e.g. friend and enemy)

6



1

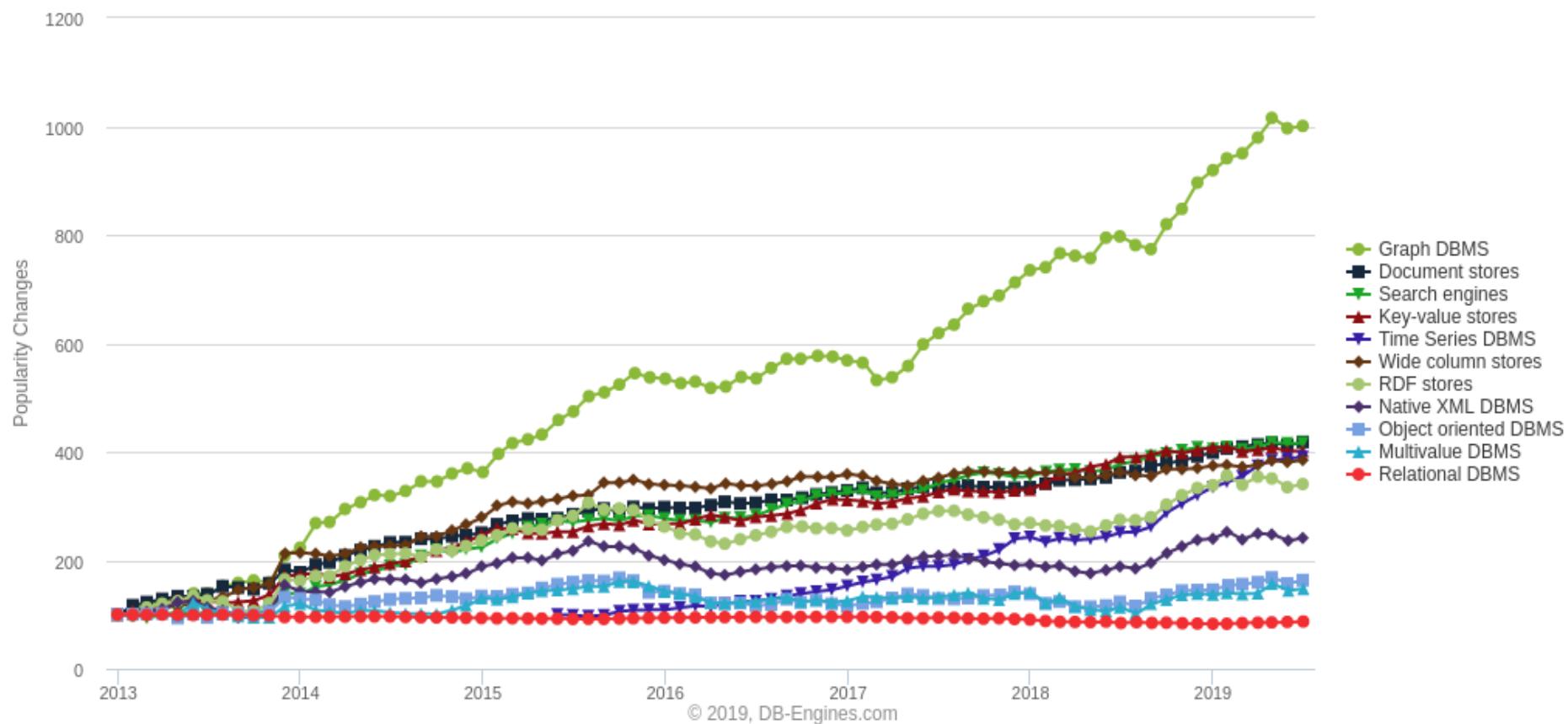
6 GRAPH MODELS

RELATIONSHIPS

- Solves the complexity of relationships
- Relationship can have attributes; works_with, works_for
- Richer than SQL databases for the type of datasets
- Imagine a friend and query in SQL with join table

EXPLOSIVE GROWTH

Complete trend, starting with January 2013



Source

SQL OR GRAPH DATABASE



USE CASES FOR A GRAPH DATABASE

- Recommendation Engines
- Artificial Intelligence
- Master Data
- Knowledge Graph
- Fraud Detection
- Centrality and Clustering

6 USE CASE FOR A GRAPH DATABASE

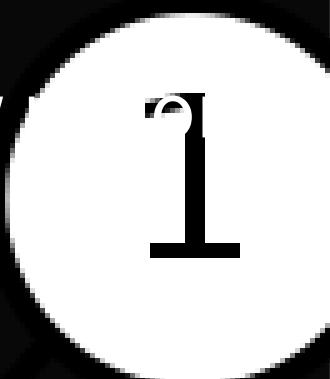
Connected/Recursive Data

- How do I know person X follows person Y
- How many people follow people who follow twitter

Recognize Patterns in Data

Clustering

- Finding significant figures in his contact
- Most influential contact
- parts fail at



6

USE CASE FOR A SQL DATABASE

- Selection/Search
- Aggregation

3

2

1

5

UNDERSTAND YOUR CHOICE

THE RIGHT TOOL

FOR THE RIGHT JOB



IN THE REAL WORLD

GRAPH DATABASES TYPES

1. Resource Definition Framework (RDF) Triplestore (W3C/semantic web)
2. Labeled Property Graph
 - Vertices and Edges support a label
 - Key-Value pairs

IT IS COMPLICATED

- Standards don't really exist
- Multiple vendors, multiple implementations

QUERY LANGUAGES

- SPARQL - W3C Standard query language for the semantic web. Some graph databases support
- GraphQL - Some vendors support GraphQL
- SQL-like things - A whole bunch of folks have SQL-esque implementations
- Cypher (and openCypher) - Query language developed by Neo4j and open sourced in 2015
- Gremlin - Part of Apache TinkerGraph. Implemented widest variety of vendors, including AWS Neptune, Azure Cosmos DB, JanusGraph, etc. Even Neo4J implements

GRAPH DATABASES

- SASS - AWS Neptune, Azure Cosmos DB
- Commercial - Neo4j, Datastax
- OSS - JanusGraph, TinkerPop

MAIN CHOICES

Neo4j - Very popular. It will win your Google eyes

- Cypher is easy to start with
- Offerings from cloud vendors

TinkerGraph/Gremlin - Open source from Apache

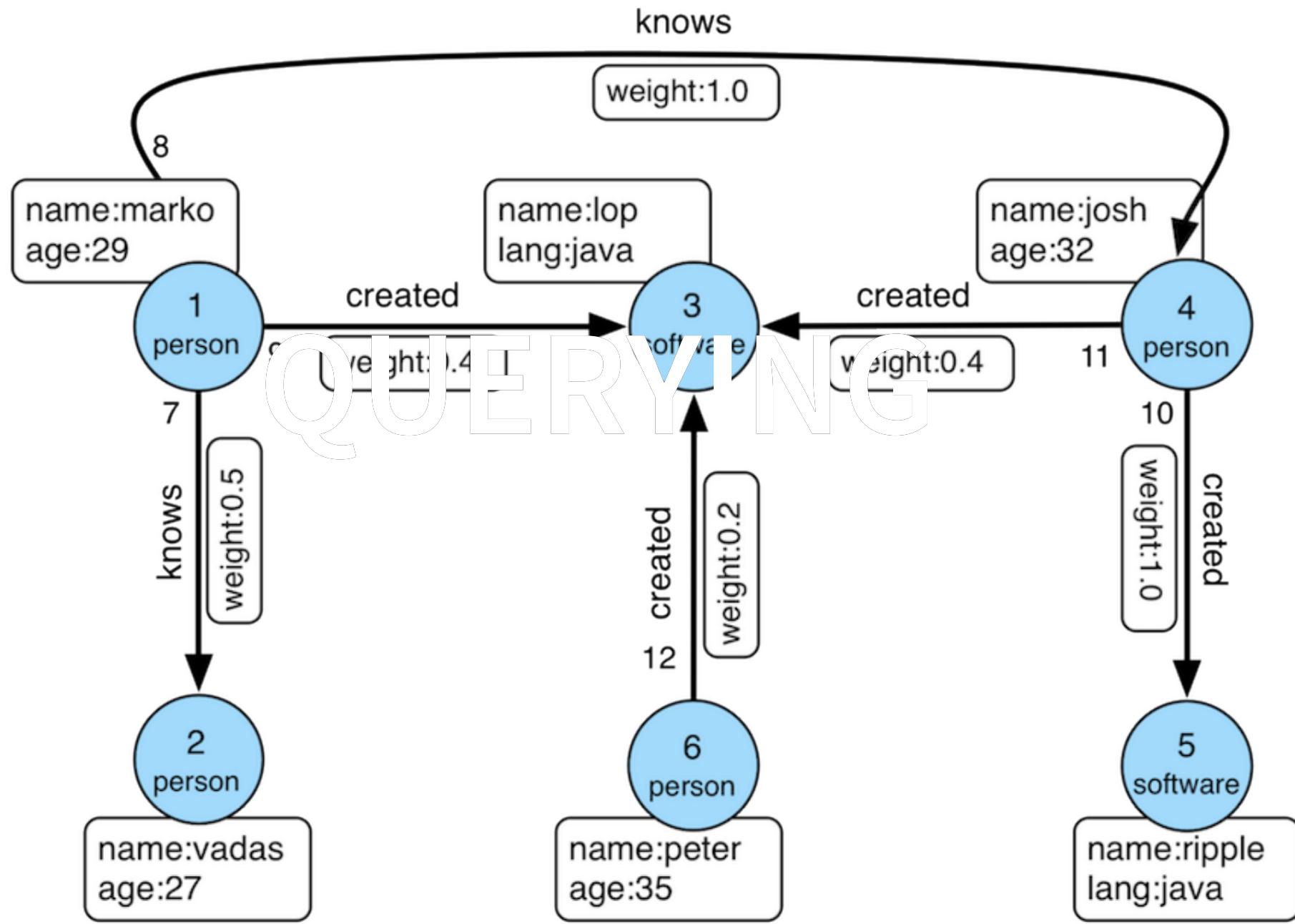
- Harder to get started
- MANY options

CHOICE

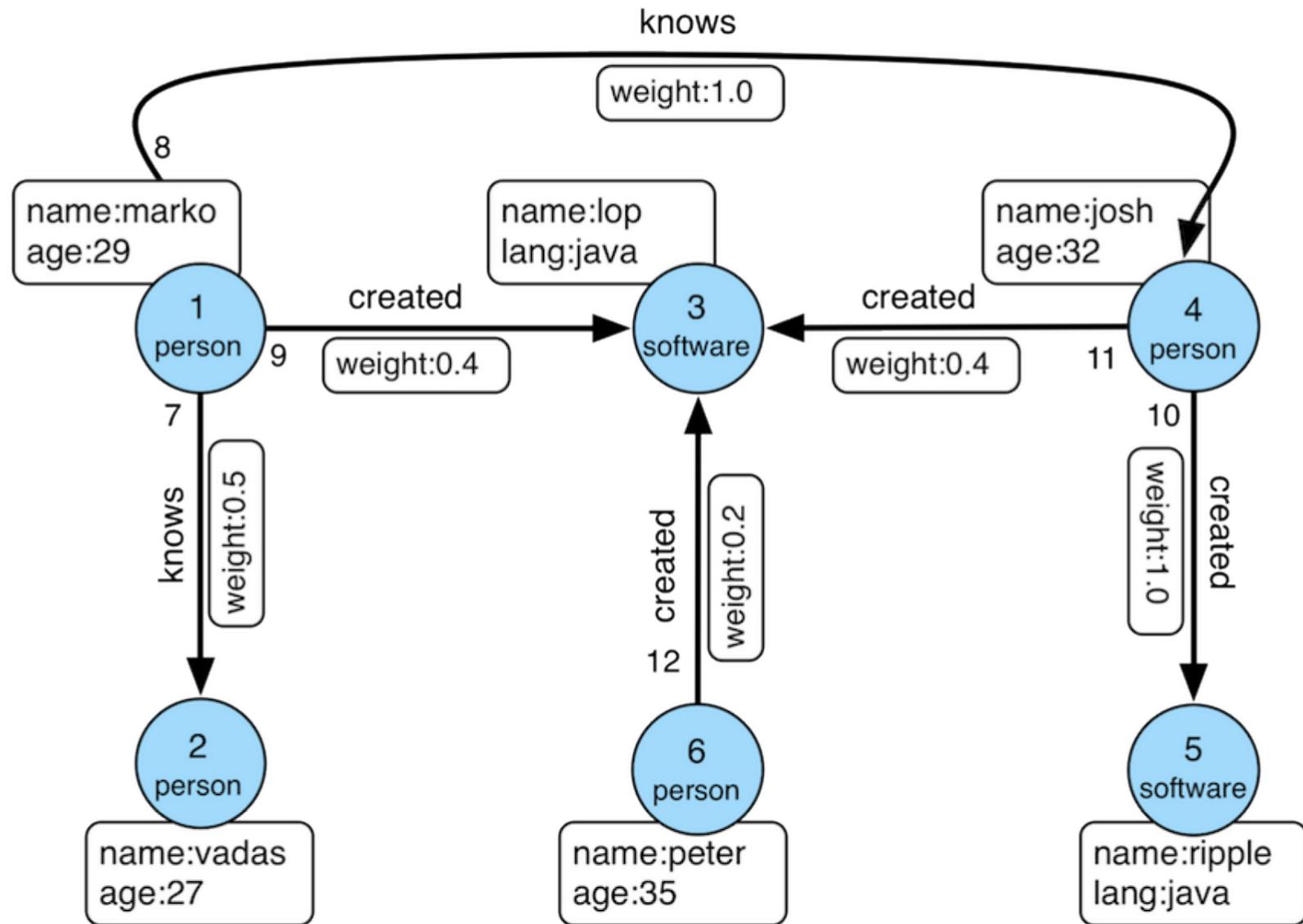
TinkerGraph/Gremlin over Neo4j/Cypher

- Gremlin helps you understand graphs better than Cypher
- Gremlin language is most portable amongst vendors
- Gremlin is the most widely available
- TinkerGraph implementations abound

Property Graph



Property Graph



QUERYING

TinkerPop has a starter Graph database available. We
create the graph by

```
graph = TinkerFactory.createModern()
```

QUERYING

Creating the graph

```
g = graph.traversal()
```

QUERYING

List all the things

```
g.V()
```

QUERYING

Get the first vertex

```
g.V(1)
```

QUERYING

Get a property off of the first vertex by key

```
g.V(1).values('name')
g.V(1).values('age')
```

QUERYING

Aggregate together

```
g.V(1).values('name', 'age')
```

QUERYING

List all of the edges out of a vertex with a label
property on the edge

```
g.V(1).outE('knows')
```

QUERYING

List all of the names. In this case, we need to get the connecting vertex. That vertex is an incoming node to the edge, so we use the inV keyword

```
g.V(1).outE('knows').inV().values('name', 'age')
```

QUERYING

We can also have comparison operators

```
g.V(1).outE('knows').inV().has('age',gt(30)).values('name','ag
```

QUERYING

We are going to start over. Quit command line (:q) and restart

```
:q
```

QUERYING

Create connection

```
graph = TinkerGraph.open()
```

QUERYING

Create graph traversal

```
g = graph.traversal()
```

QUERYING

Add a vertex

```
v1 = g.addV('person').property(id,1).property('name','marko').
```

QUERYING

Add person the second

```
v2 = g.addV('person').property(id,2).property('name','josh').p
```

QUERYING

Create an edge. This will connnect two item and provide a type of relationship

```
g.V(1).addE('knows').to(g.V(2))
```

QUERYING

Gremlin Query

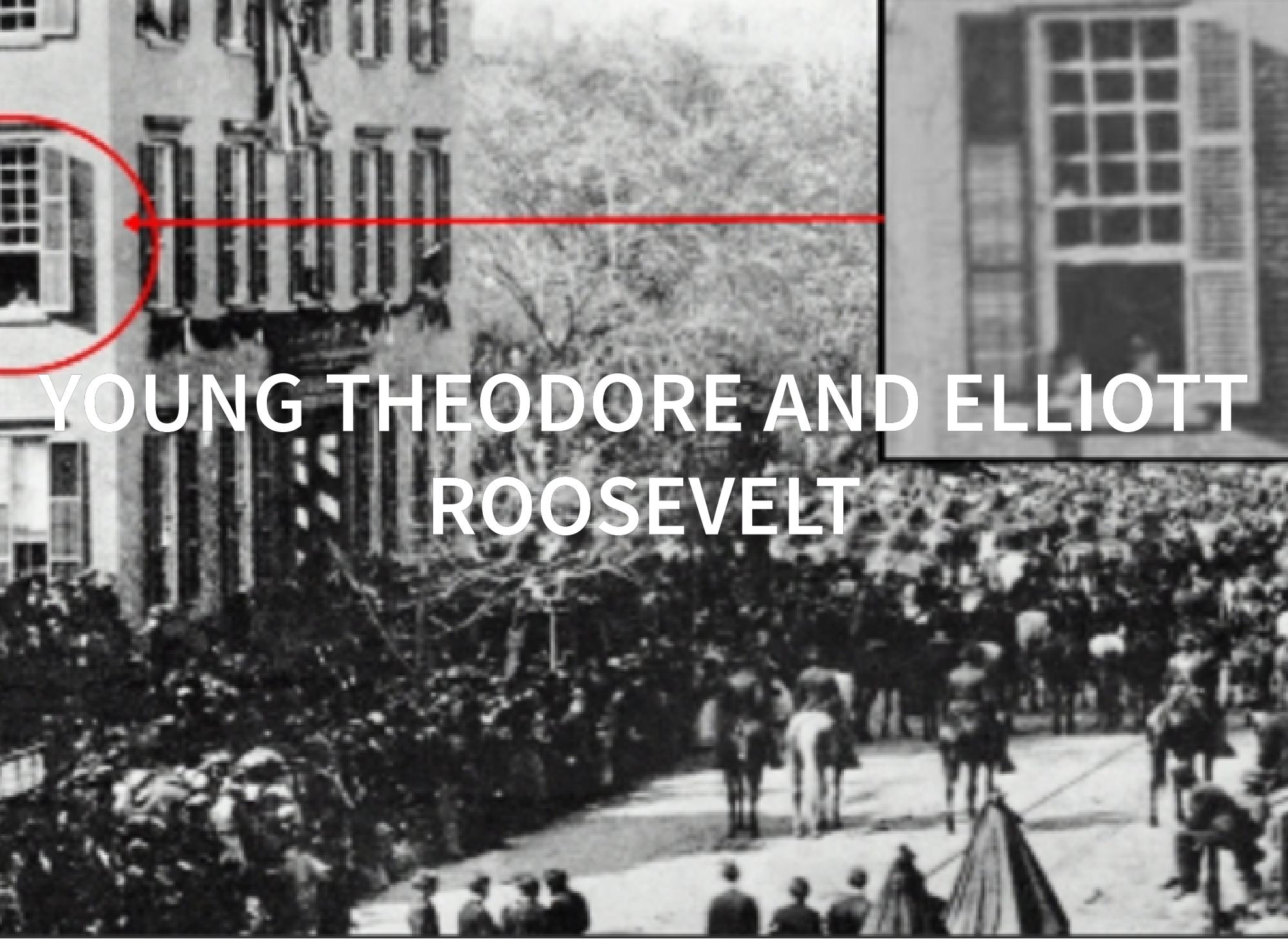
```
g.V().has('name', 'marko').out('knows').values('name')
```

Cypher Query

```
MATCH ( {name:'marko'} )-[ :knows ]->( x )
RETURN x.name
```

I DID THIS THING





YOUNG THEODORE AND ELLIOTT ROOSEVELT

- 
- Connections are important
 - Interactions matter
 - Relationships matter



COSIMO

- A graph database system to record
 - People
 - Events
 - Dates
 - Places

A portrait painting of a man with a serious expression, looking slightly to his left. He has light-colored hair and is wearing a red robe over a green tunic. The background is dark.

History.page

QUERYING

Counting vertices

```
g.V().count()  
g.V().hasLabel('people')  
g.V().hasLabel('event')
```

QUERYING

Existence check

```
g.V().hasLabel('people').has('id','Abraham Lincoln').count()  
g.V().hasLabel('people').has('id','Scott Hutchinson').count()
```

QUERYING

People associated with a person

```
g.V('Abraham Lincoln').repeat(both().simplePath()).emit().time
```

QUERYING

Events associated with a person

```
g.v('Abraham Lincoln').as('a').repeat(bothE('PARTICIPATED')).bo
```



SEE YOU NEXT YEAR!

AUGUST 3 - 6, 2020





QUESTIONS?

JOHN PTACEK



@jptacek



JPtacek