DATAVERSITY WEBINAR:

GRAPH DATA MODELING IN FOUR DIMENSIONS

JANUARY 7TH, 2020

INSTRUCTED BY THOMAS FRISENDAL

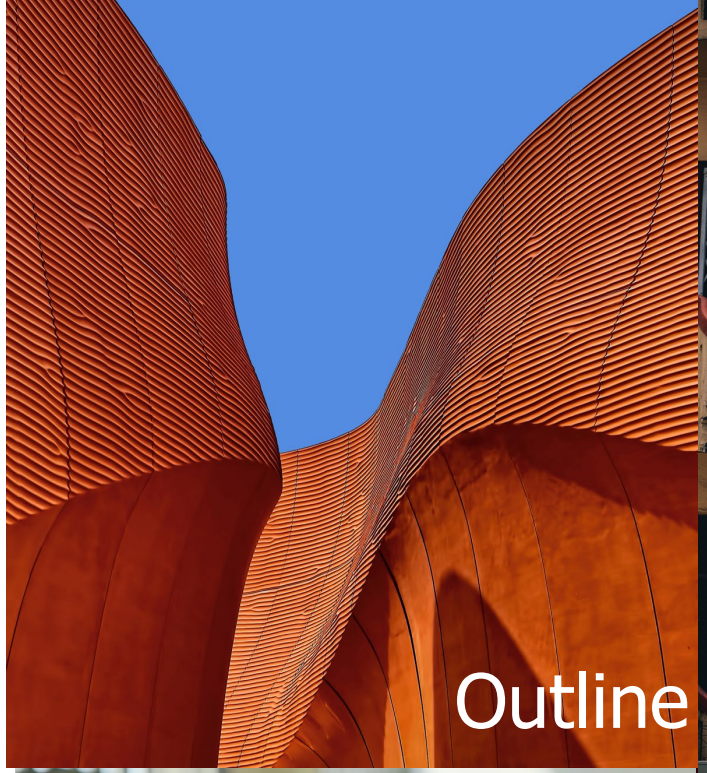


Photo by Ricardo Gomez Angel on Unsplash

Photo by Greg Jeanneau on Unsplash

Photo by Dennis Cortés on Unsplash

Photo by david latorre romero on Unsplash

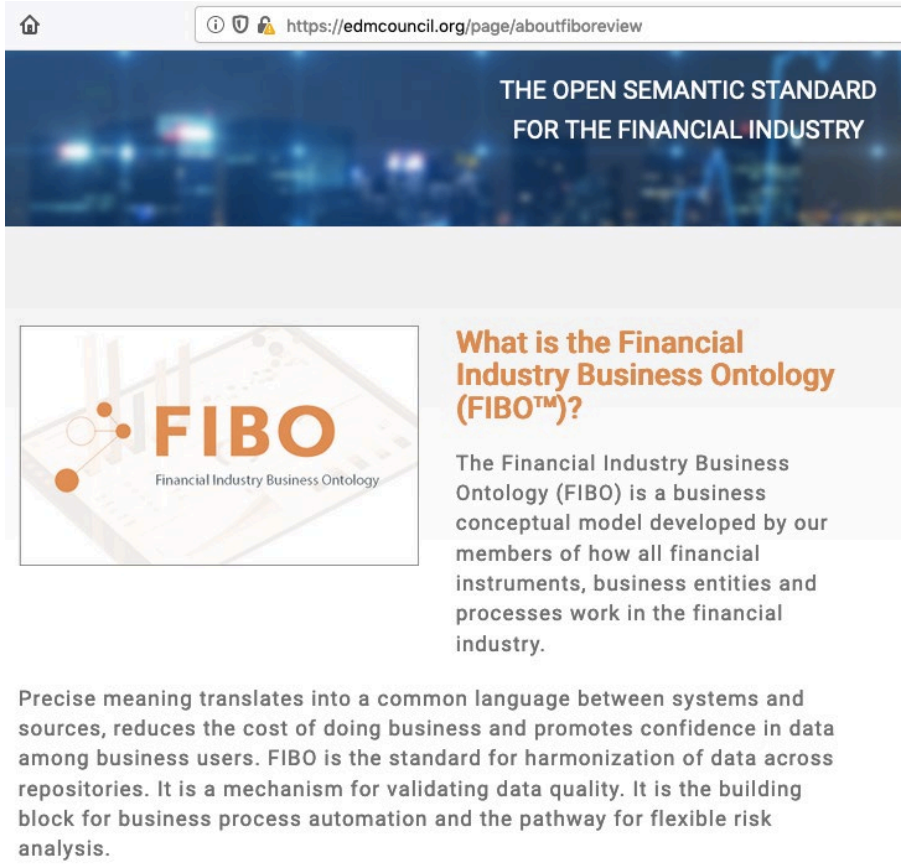
GRAPH DATA MODELING OUTLINE

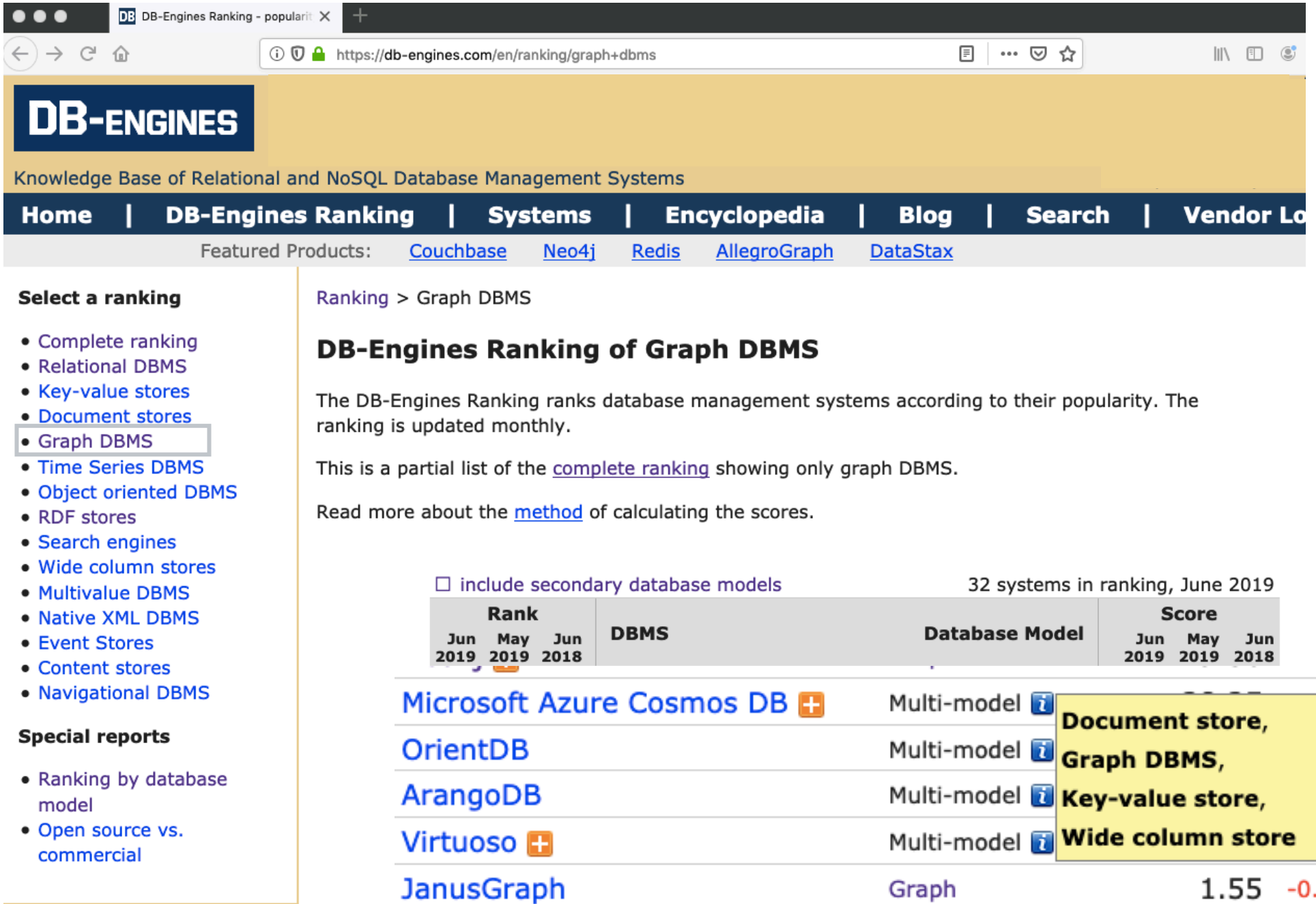
- What?
- Why?

TWO MAJOR CATEGORIES: PROPERTY GRAPH VS. RDF

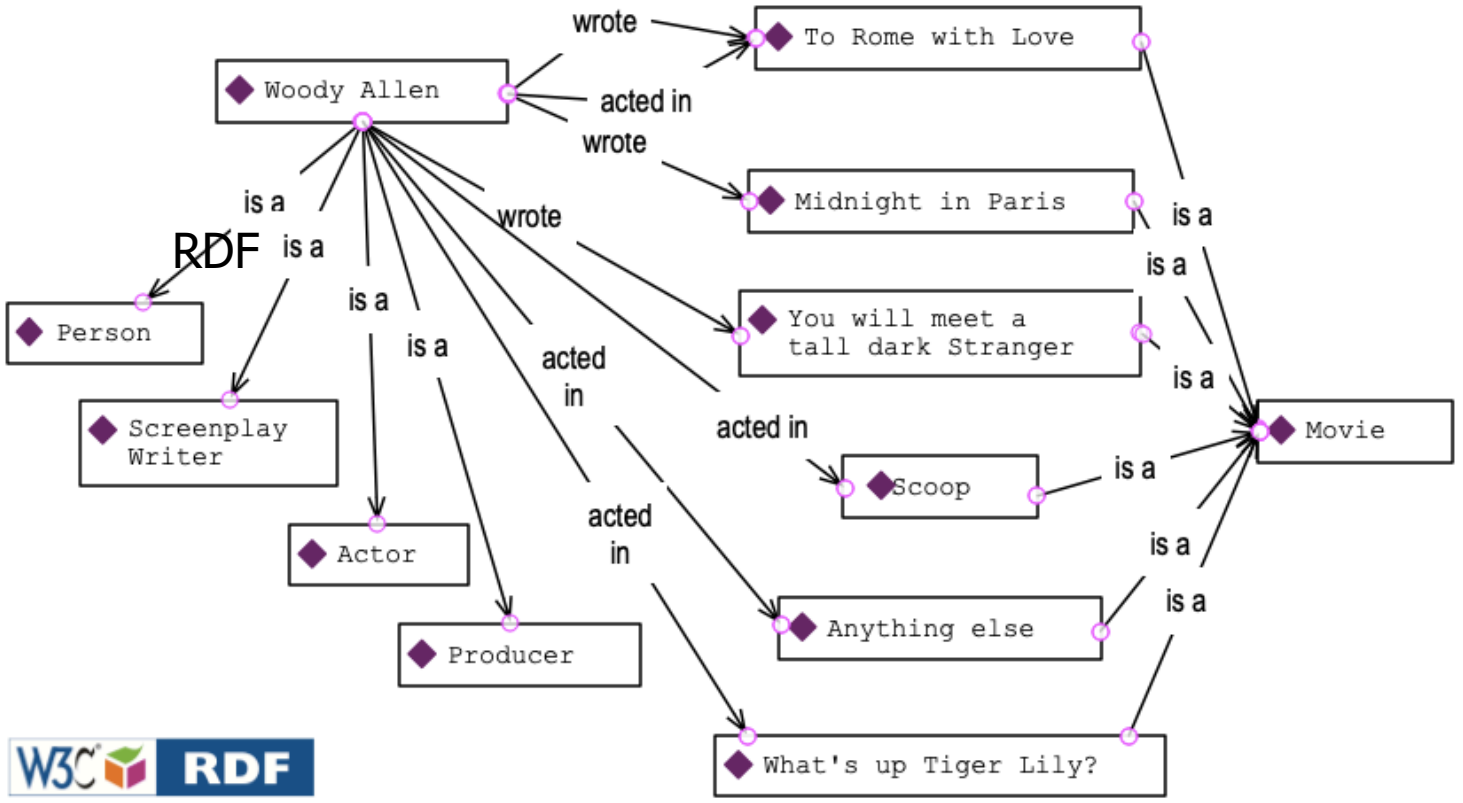
Property Graph – Heavily Connected Data

RDF: Linked Semantics

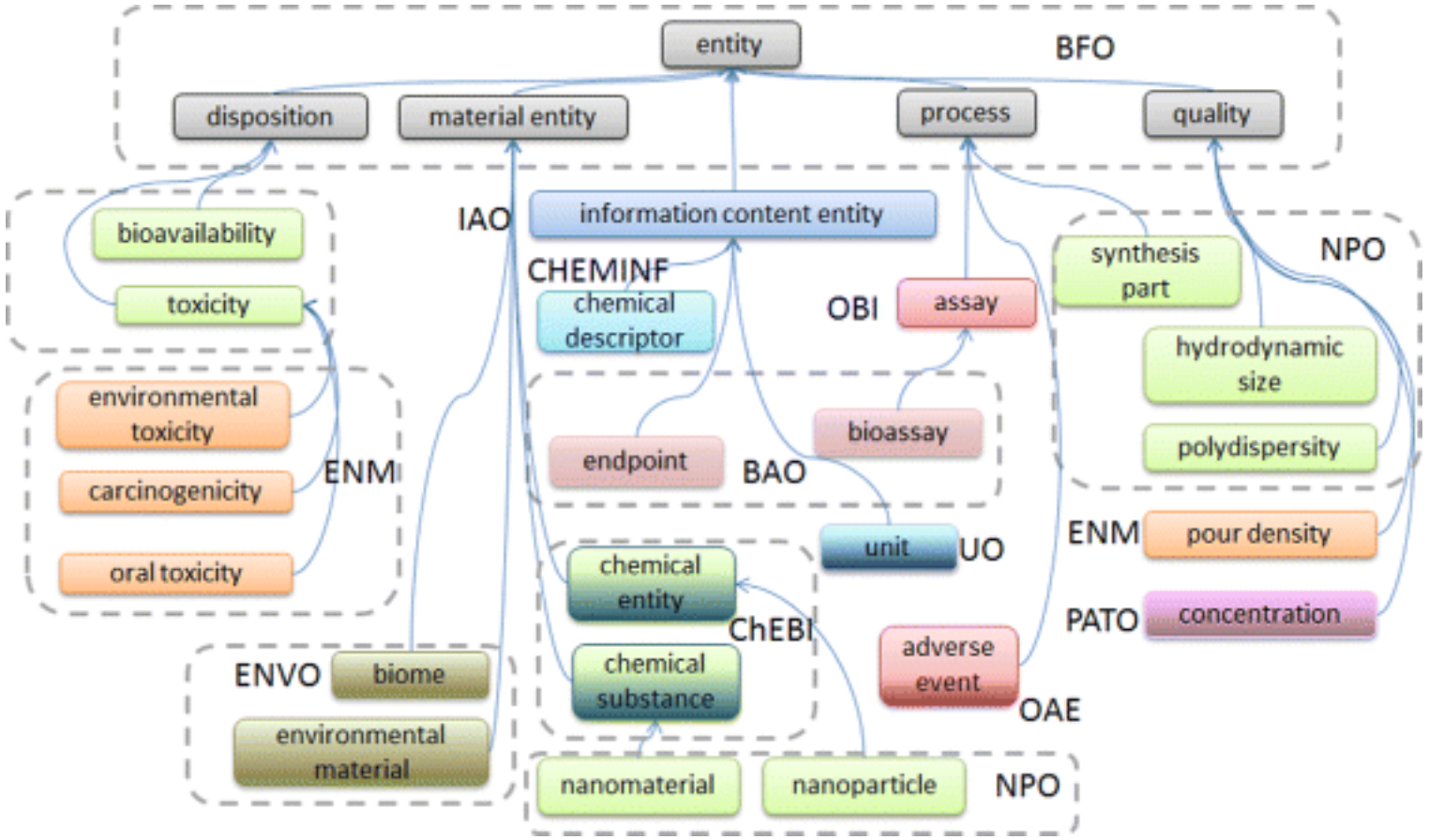



DB-Engines Ranking - popularit
https://db-engines.com/en/ranking/graph+dbms
DB-ENGINES
Knowledge Base of Relational and NoSQL Database Management Systems
Home | DB-Engines Ranking | Systems | Encyclopedia | Blog | Search | Vendor Lo
Featured Products: Couchbase Neo4j Redis AllegroGraph DataStax
Select a ranking
Complete ranking
Relational DBMS
Key-value stores
Document stores
Graph DBMS
Time Series DBMS
Object oriented DBMS
RDF stores
Search engines
Wide column stores
Multivalue DBMS
Native XML DBMS
Event Stores
Content stores
Navigational DBMS
Special reports
Ranking by database model
Open source vs. commercial
Ranking > Graph DBMS
DB-Engines Ranking of Graph DBMS
The DB-Engines Ranking ranks database management systems according to their popularity. The ranking is updated monthly.
This is a partial list of the complete ranking showing only graph DBMS.
Read more about the method of calculating the scores.
include secondary database models
32 systems in ranking, June 2019
Rank
Jun 2019
May 2019
Jun 2018
DBMS
Database Model
Score
Jun 2019
May 2019
Jun 2018
Microsoft Azure Cosmos DB
Multi-model
OrientDB
Multi-model
ArangoDB
Multi-model
Virtuoso
Multi-model
JanusGraph
Graph
1.55
-0.07
+1.19
Document store,
Graph DBMS,
Key-value store,
Wide column store

SEMANTIC WEB – RDF: RESOURCE DESCRIPTION FRAMEWORK – OWL: WEB ONTOLOGY LANGUAGE - W3C STANDARDS



SEMANTIC WEB – RDF: RESOURCE DESCRIPTION FRAMEWORK – OWL: WEB ONTOLOGY LANGUAGE - W3C STANDARDS



* By Janna Hastings, Nina Jeliazkova, Gareth Owen, Georgia Tsiliki, Cristian R Munteanu, Christoph Steinbeck, and Egon Willighagen - eNanoMapper: harnessing ontologies to enable data integration for nanomaterial risk assessment doi:10.1186/s13326-015-0005-5, CC BY 4.0, https://commons.wikimedia.org/w/index.php?curid=79191491

PROPERTY GRAPHS

Apache (2015) Tinkerpop / Gremlin (2009):

Amazon Neptune,
Azure CosmosDB,
Datastax Graph,
JanusGraph,
Cypher for Gremlin,
OrientDB,
Stardog

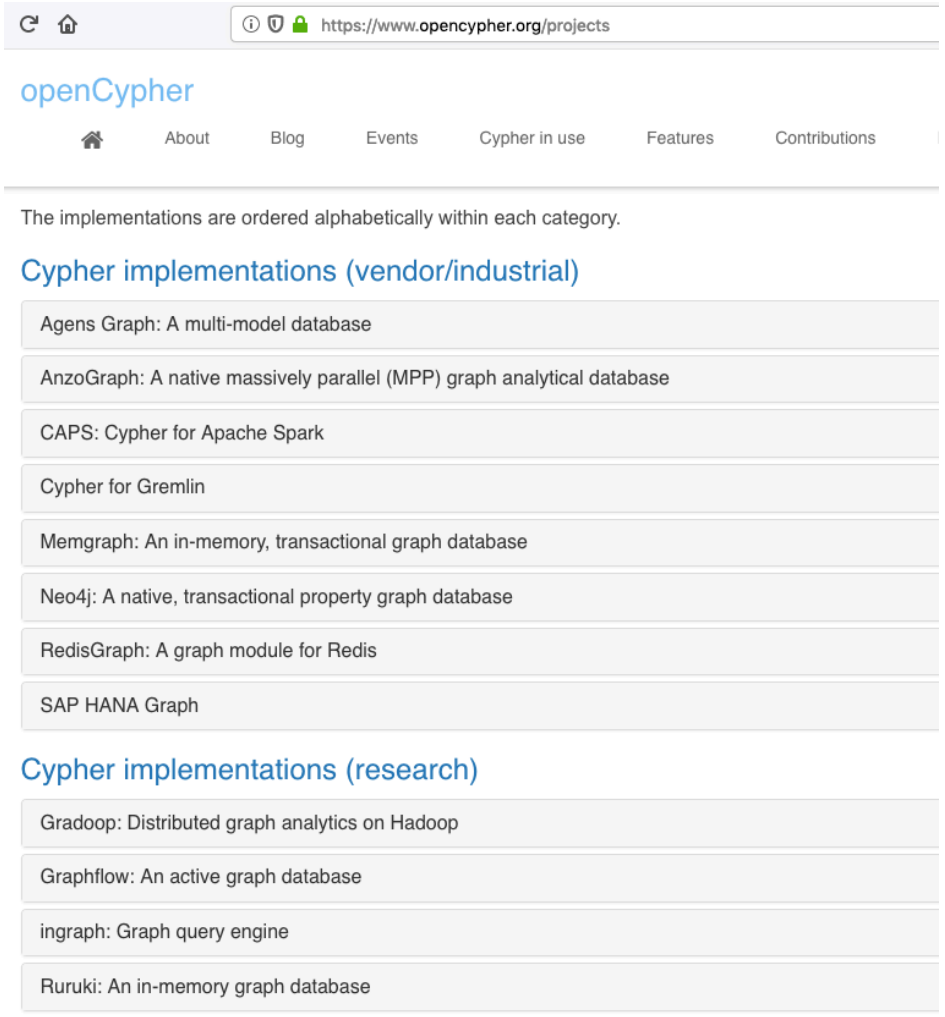
Neo4J Cypher (2011):

openCypher (2017) – see list to the right:

SQL Style:

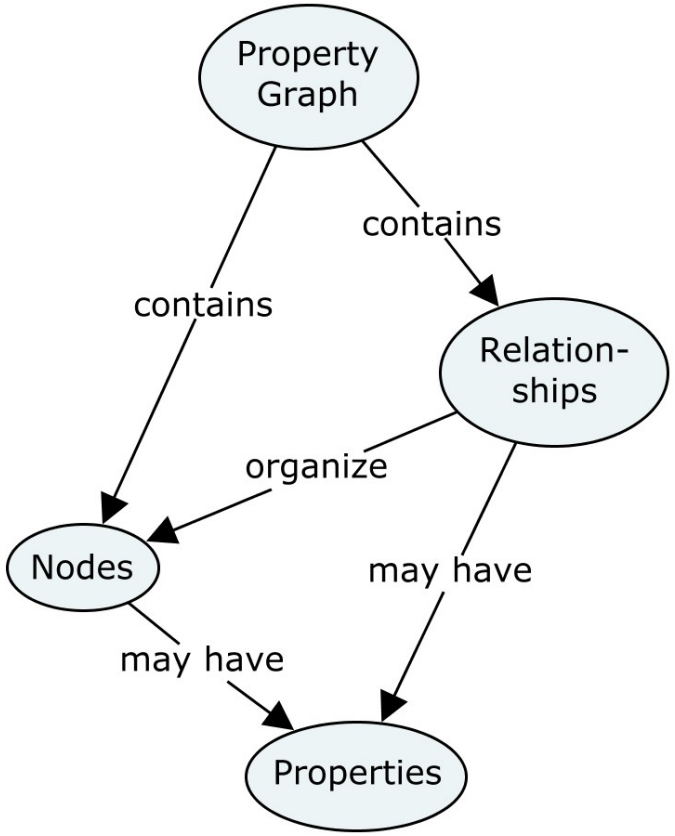
ISO SQL Property Graph Extensions (2017-2020?),
Oracle PGQL (2016),
SQL Server 2017 Graph,
TigerGraph GSQL (2018)

*(*Own not authoritative research mid 2019, errors and unintended omissions may exist. If so, I apologize)*

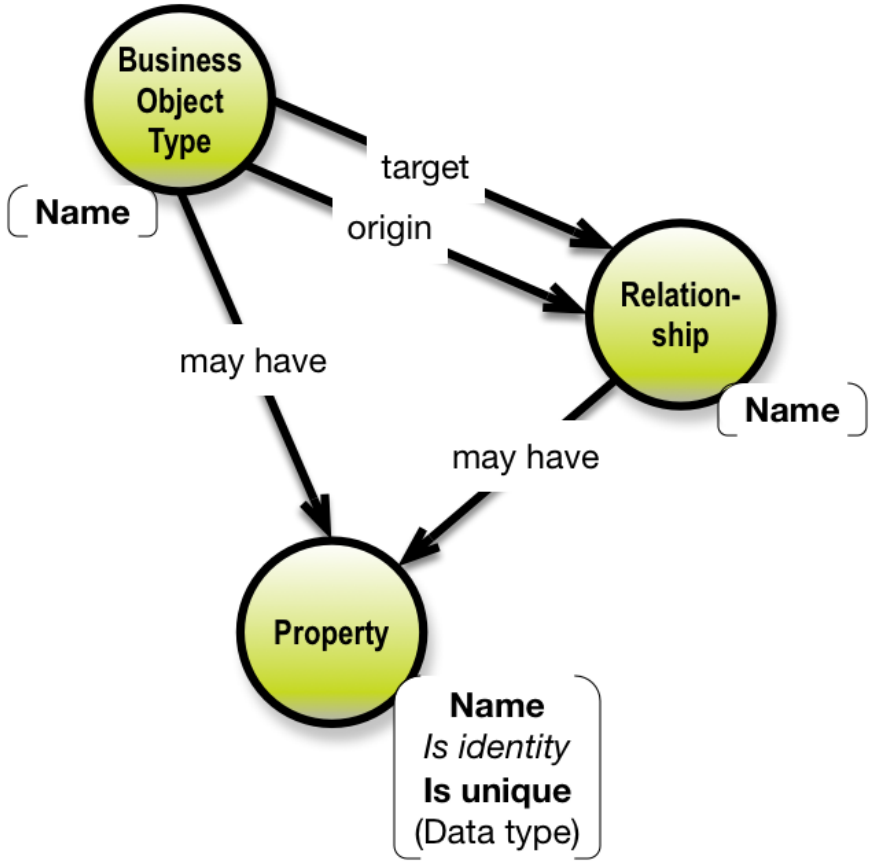


THE BASICS OF PROPERTY GRAPHS

Property Graph Concepts

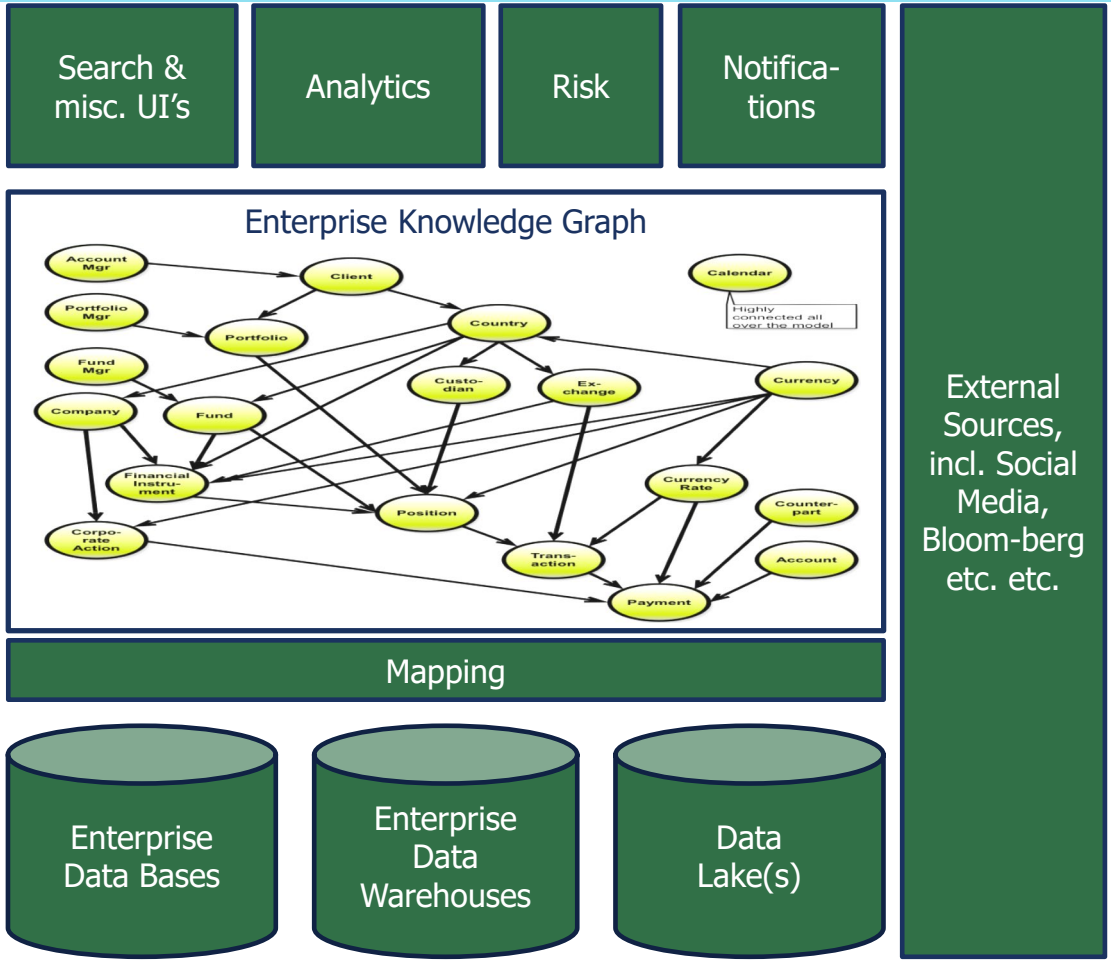


Property Graph Meta Model



WHAT ABOUT KNOWLEDGE GRAPHS?

- Example to the right is for the financial sector
- Navigation across silos
- Integration – even streaming
- Automation, e.g. of alerts
- *Pick your choice:*
  - *Strong semantic focus: RDF*
  - *Strong network focus: Property Graphs*
  - *Hybrid solutions*



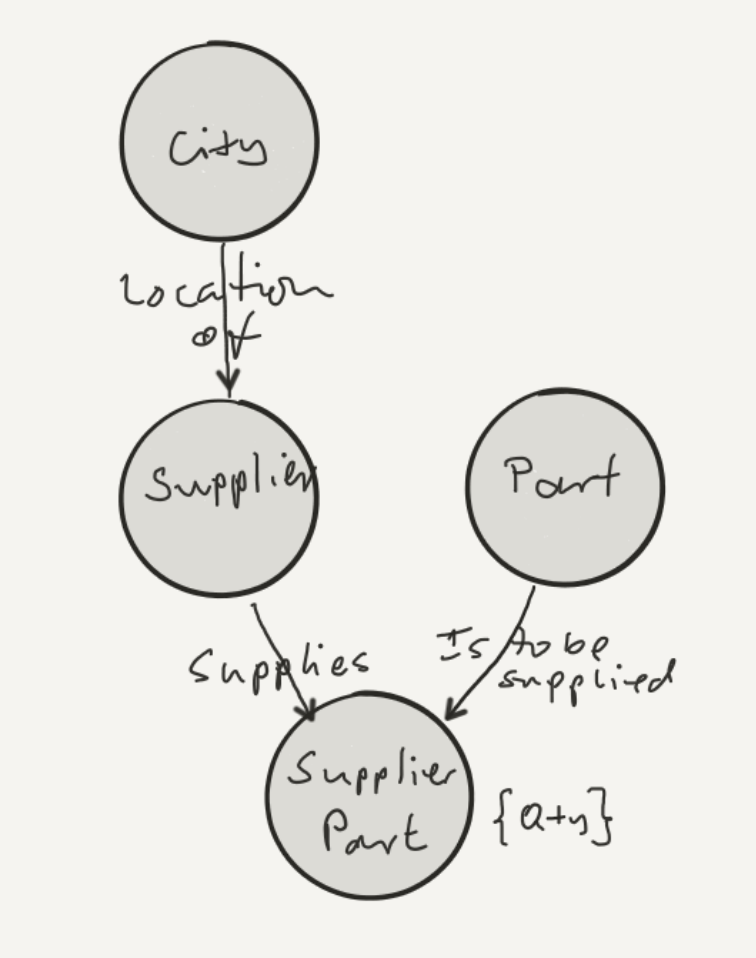
SQL?
W.I.P.: PROPERTY GRAPH EXTENSIONS

- In 2017, INCITS in the US invited contributors to "Standardizing Graph Database Functionality"
- Ongoing work in the ISO/IEC SC32/WG3 (the information below is not authoritative)
- Graph Query Language
- Embed graph queries in the FROM clause of SELECT statements
- Visualization of graph paths as part of the syntax
- Describe property graphs (schema syntax)
- Map SQL tables to property graph
- Snapshot from 2018 of work in progress
  - https://www.youtube.com/watch?v=hlhbUeoCPio

“EVERYTHING LOOKS LIKE A GRAPH”

GRAPHS ARE INTUITIVE, COMMUNICATE WELL, AND ARE FUN TO DRAW – TOGETHER!

COGNITION AND PERCEPTION:
GRAPHS JUST COME, NATURALLY, TO US ON THE WHITEBOARD



SPATIAL THINKING IS THE FOUNDATION OF ABSTRACT THOUGHT

Barbara Tversky, Prof. Emerita of Psychology, Stanford:

*Book: "Mind in Motion: How Action Shapes Thought",

Hachette UK, May 2019

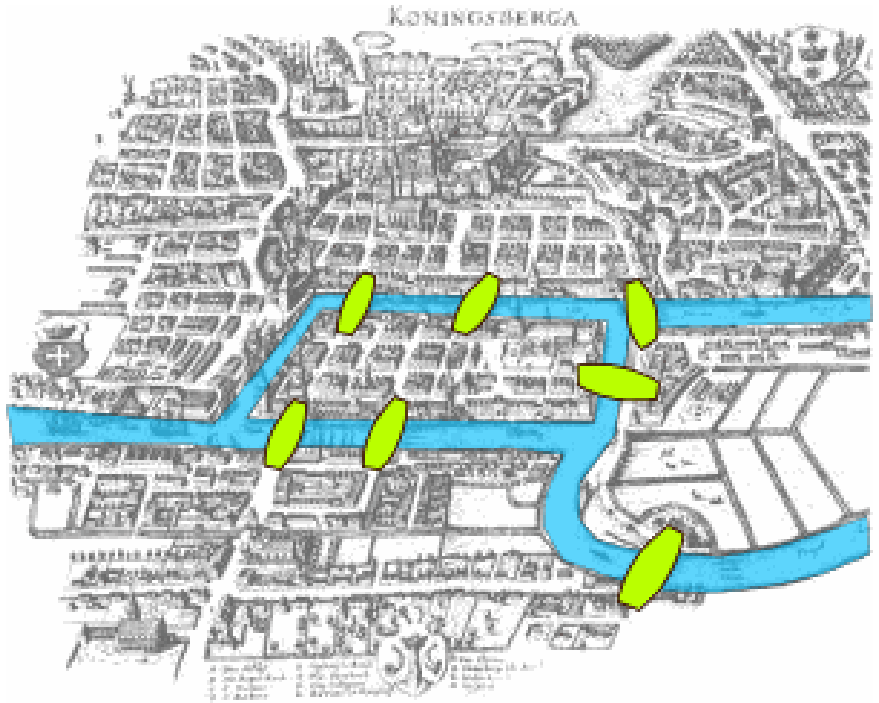
"Maps in Minds", "Space: Maps", "The World Is Diagrammed" …

The Nine Laws of Cognition

1. There are no benefits without costs
2. Action molds perception
3. Feeling comes first
4. The mind can override perception
5. Cognition mirrors perception
6. Spatial thinking is the foundation of abstract thought
7. The mind fills in missing information
8. When thought overflows the mind, the mind puts it into the world
9. We organize the stuff in the world the way we organize the stuff in the mind

Leonhard Euler, 1736:

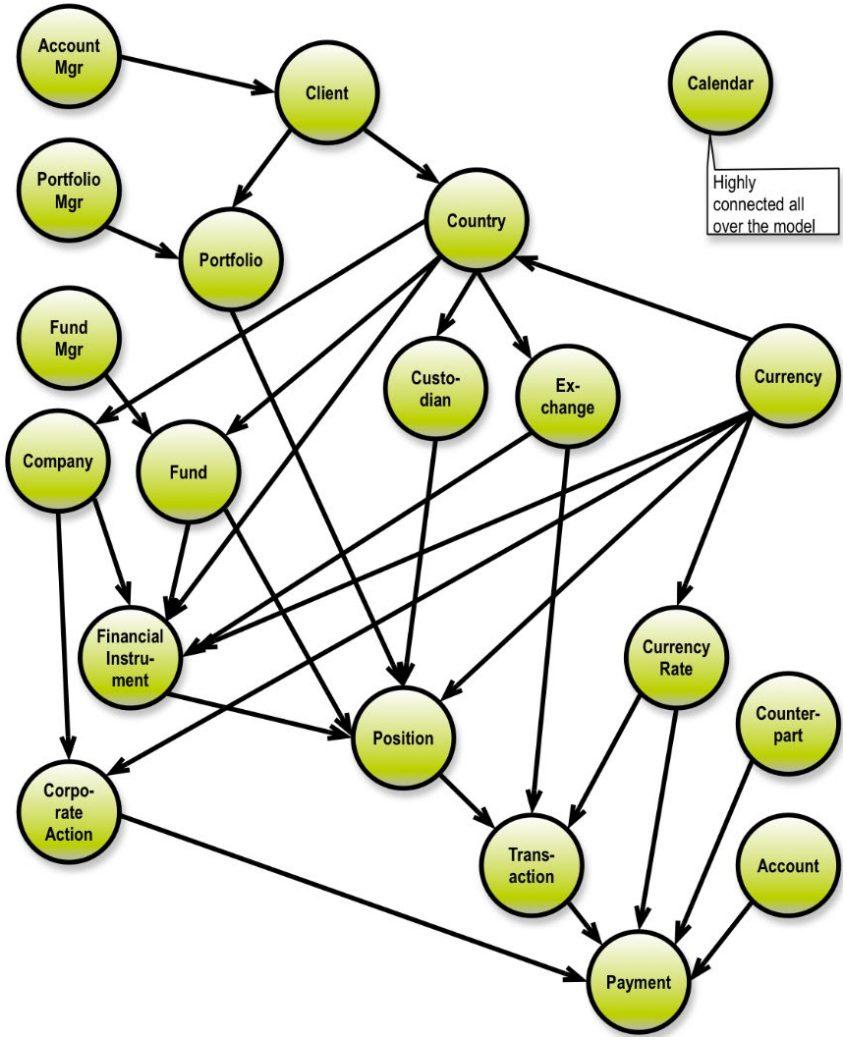
The Seven Bridges of Königsberg



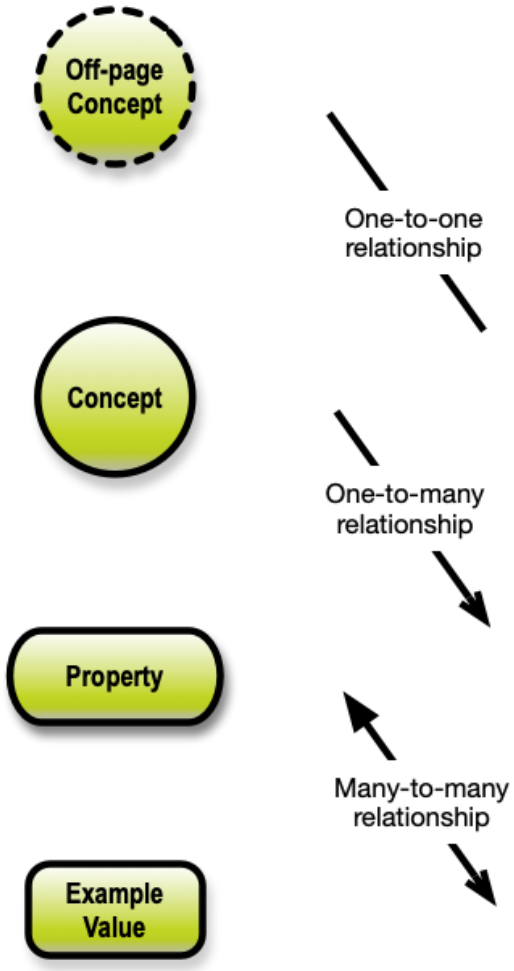
The birth of graph theory!

*https://commons.wikimedia.org/wiki/File%3AKonigsberg_bridges.png, by Bogdan Giuşcă (Public domain (PD), based on the image

MAKE COMPLEX FINANCIALS SIMPLE

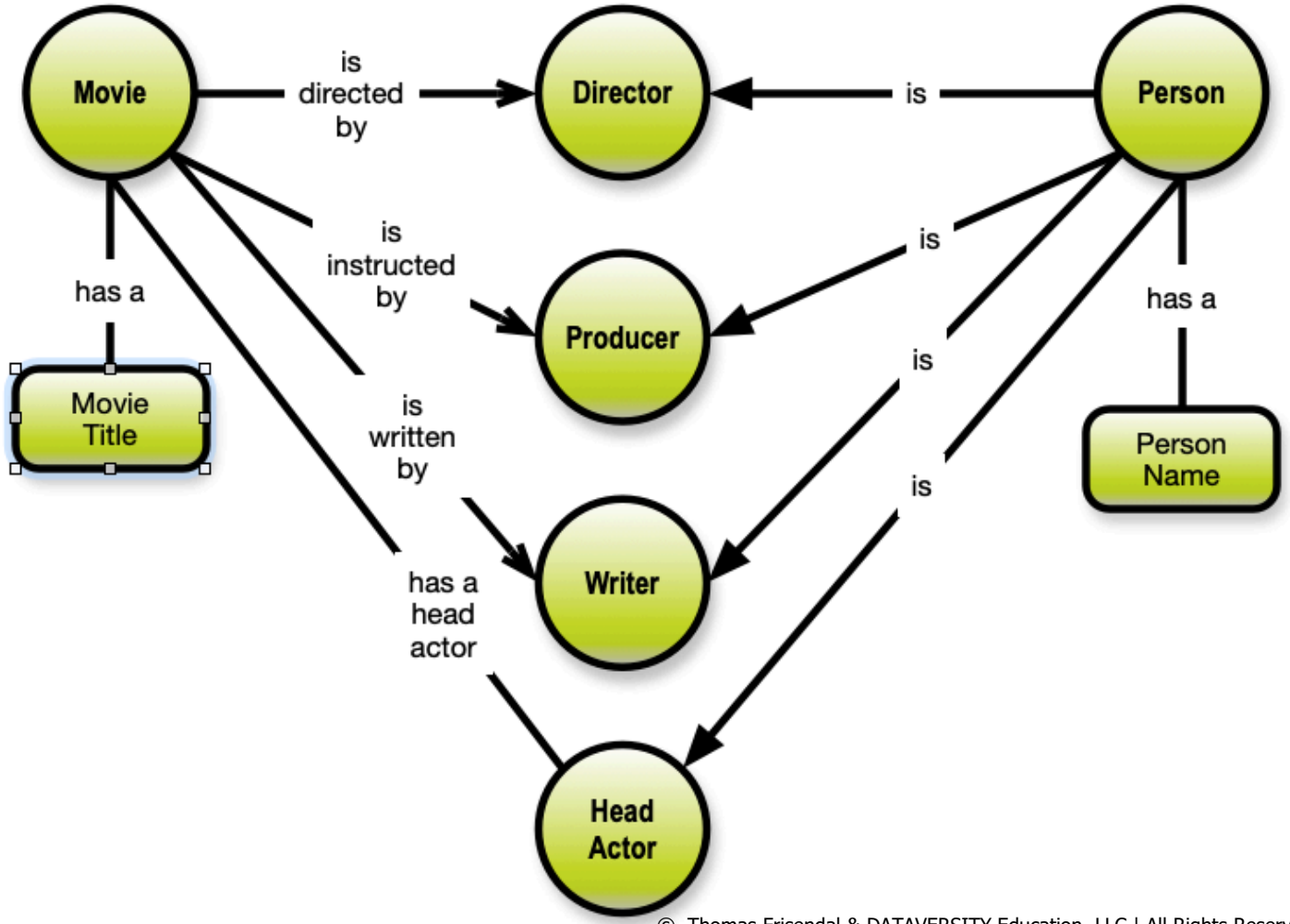


THE VISUAL SYNTAX (MY WAY)



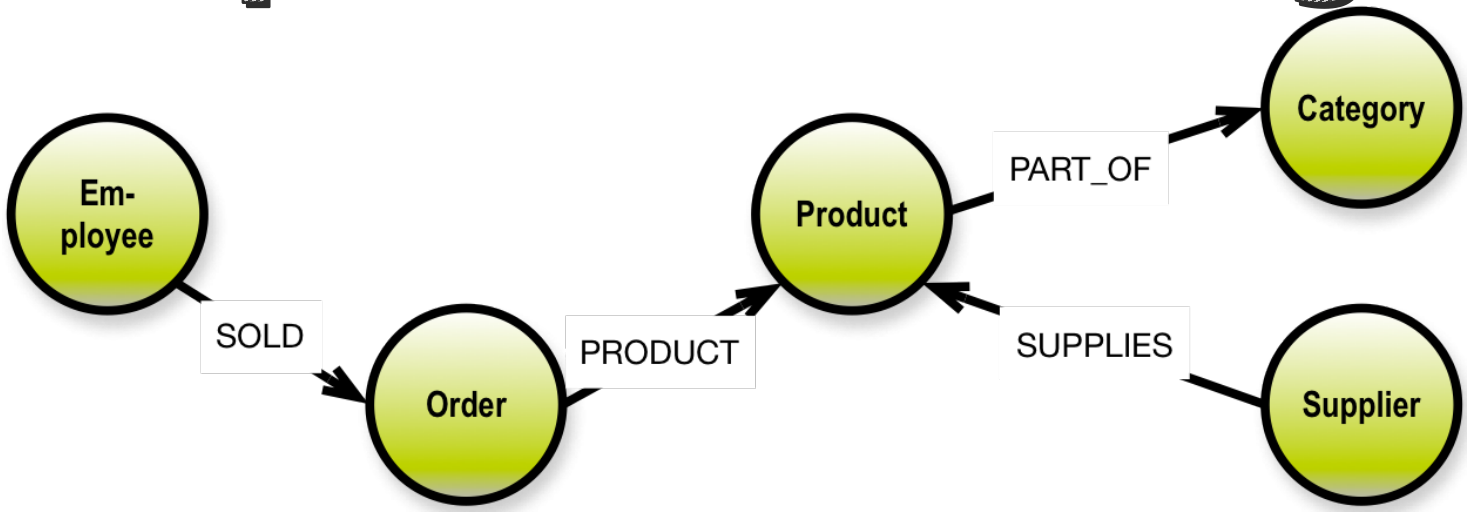
IMDB-INSPIRED DATA MODEL FOR MOVIES

- Read the Concept Model as little sentences!
- In CmapTools the texts on the arrow are called "Linking Phrases"
- In Relational Modeling they are called "Dependencies," but they did not have names (i.e., no semantics!)
- In Graph Data Models they are called "Relationships" (or "Edges" in the math version)

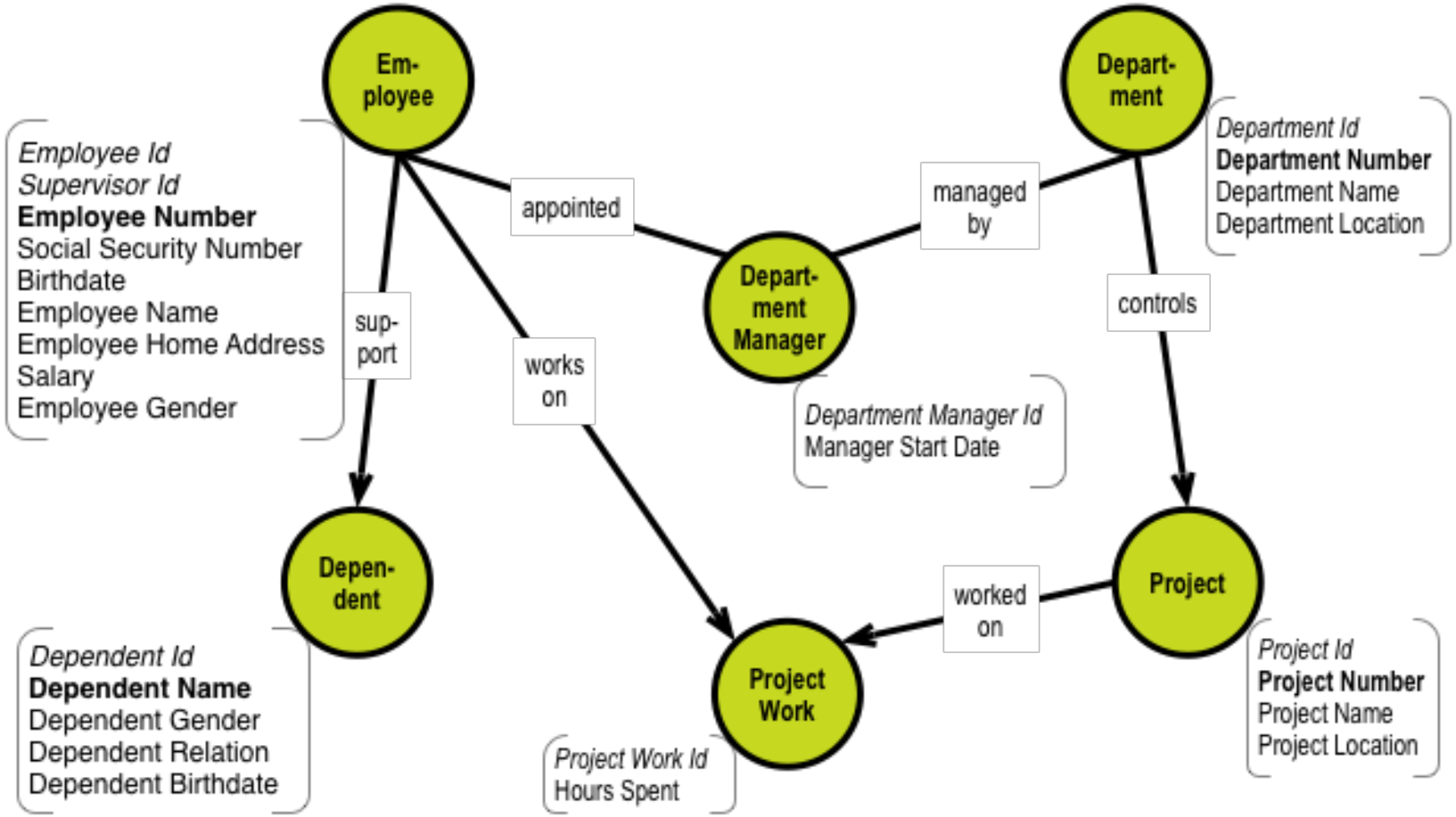


PRIMARY TAKEAWAY:

Named Relationships is what matters most in Graph Data Modeling!!!



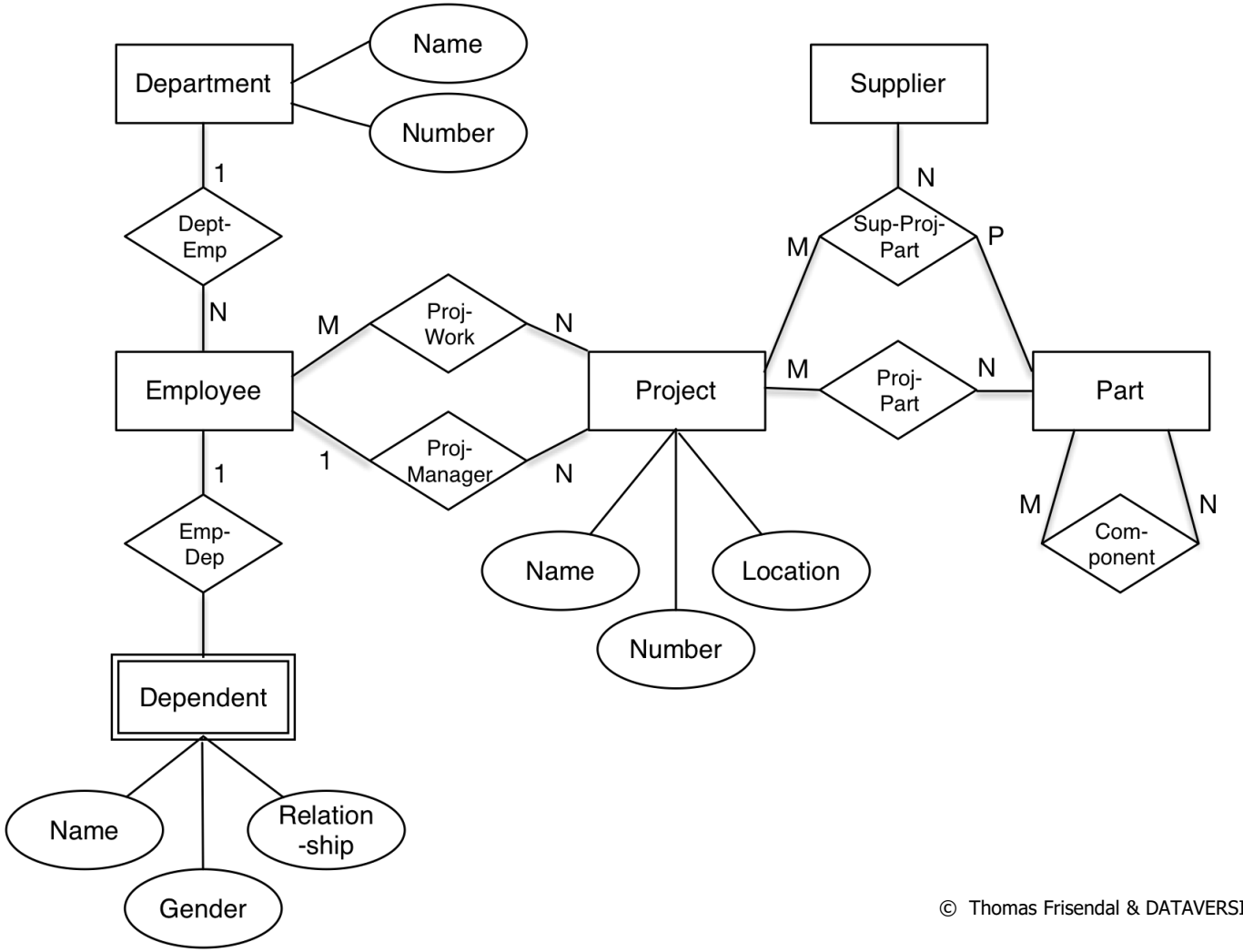
STRUCTURE AND MEANING EXPRESSED AS A PROPERTY GRAPH



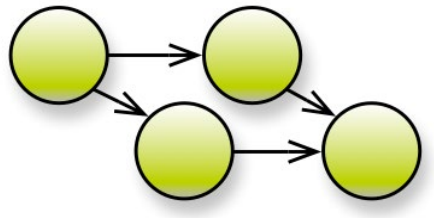
GRAPH DATA MODELING DIFFERENCES

- Graph Data Modeling compared to
- Classic Data Modeling

PETER CHEN – ENTITIES, ATTRIBUTES, RELATIONSHIPS 1976

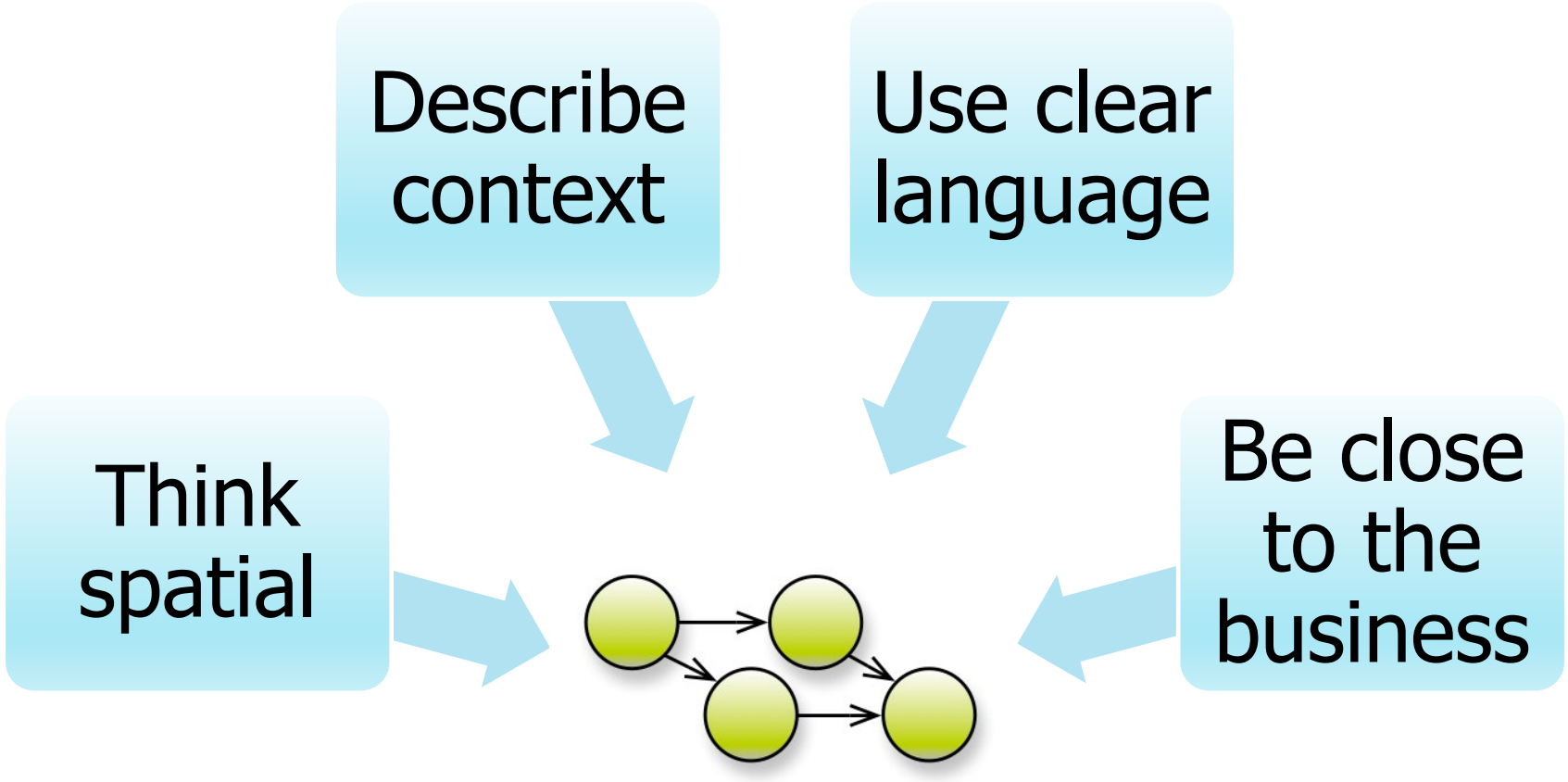


ISSUES IN MODELING DATA AS PROPERTY GRAPHS COMPARED TO CLASSIC DATA MODELING

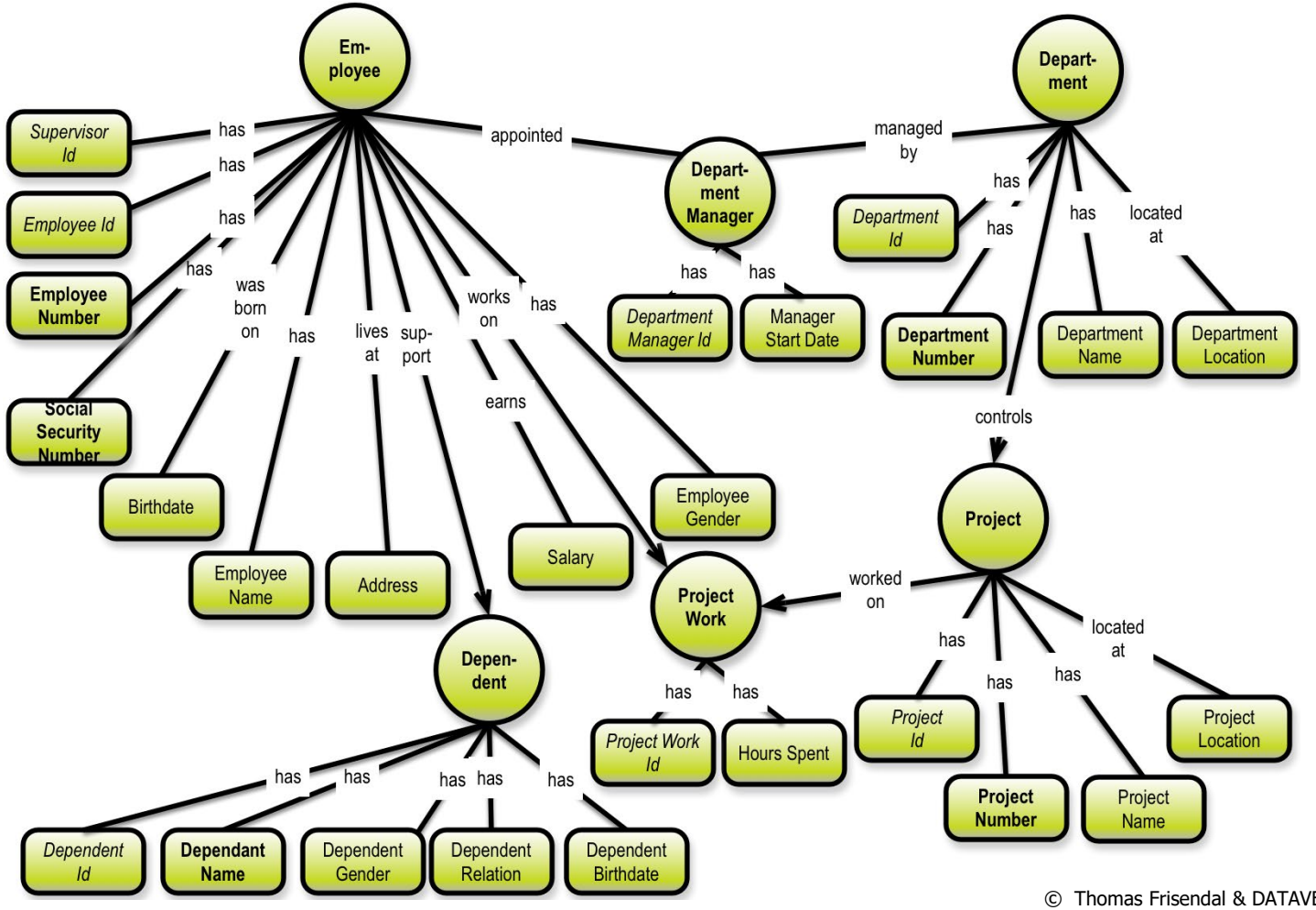


- Visualization
- Highly Connected Data
- Normalization
- Many to Many
- Think in Paths
- Labels (and Types)
- The Role of the Schema (if any)
  - Schema First
  - Schema Last
- New Opportunities for the Data Modeler

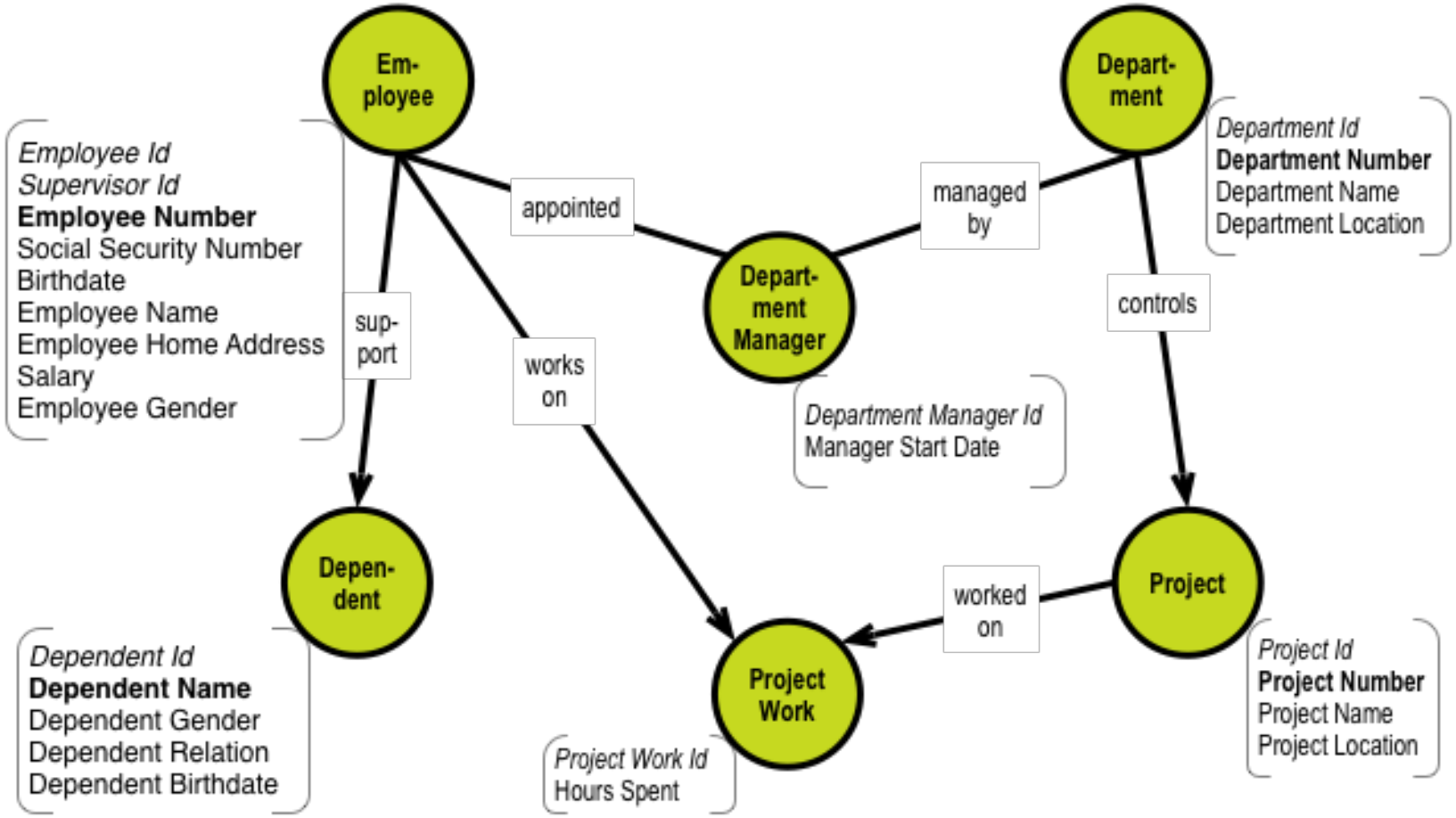
THE BASICS OF GRAPHING DATA



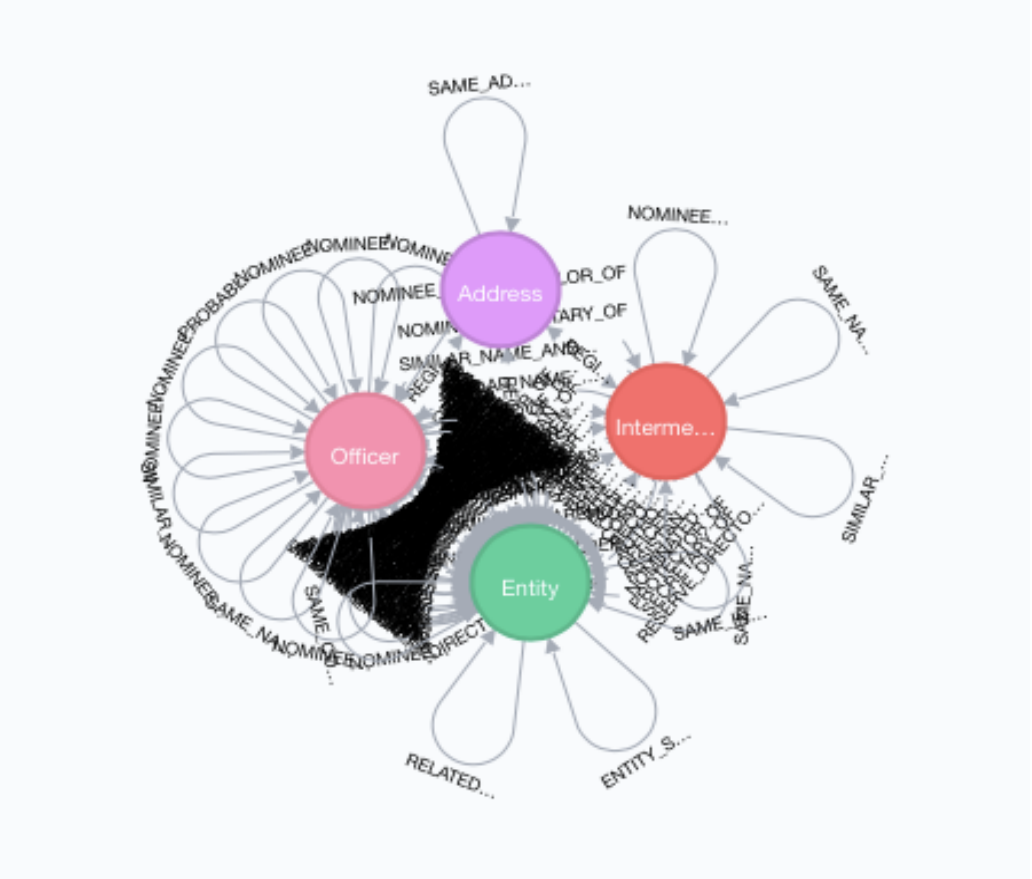
A CONCEPT MODEL (MAP)
"A DATA-DRIVEN USER STORY"



PROPERTY GRAPH REPRESENTATION COMMUNICATES STRUCTURE AND MEANING!



HIGHLY CONNECTED DATA: THE PANAMA PAPERS



THE RELATIONAL MODEL
MAKING SENSE BY EXPLORING RELATIONSHIPS?

Famous text book example from several of Chris Date's books and presentations about the relational model.

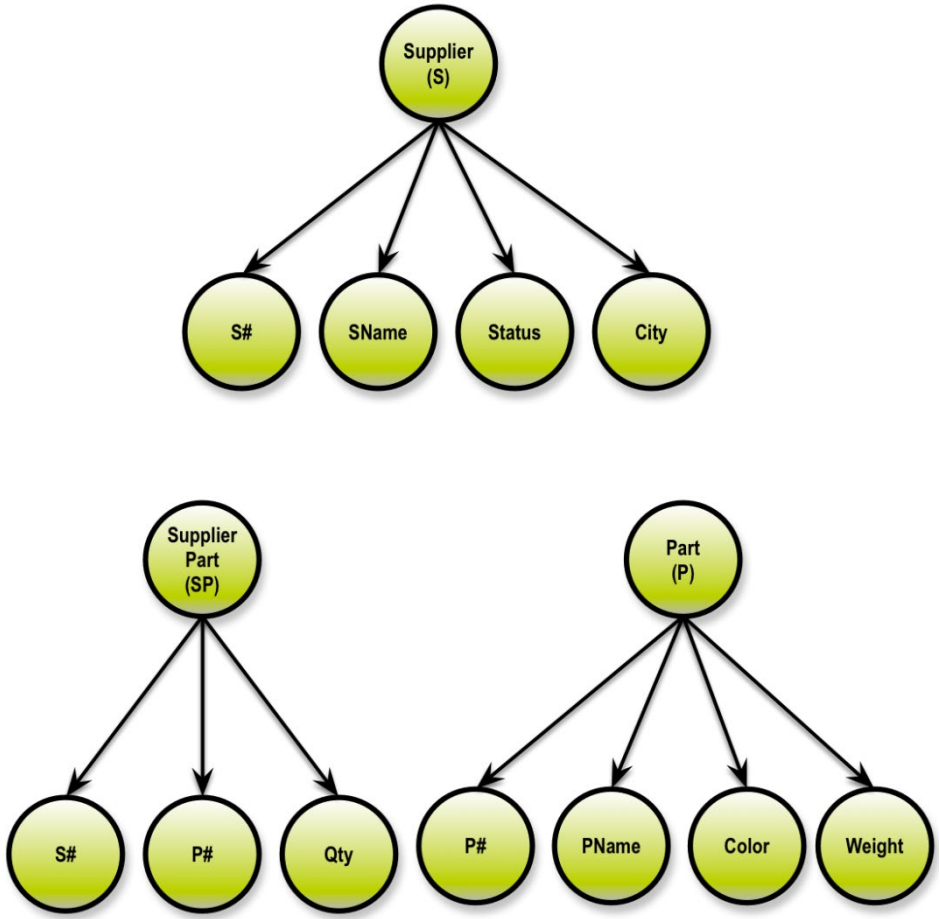
I believe the first appearance of this example was in "An Introduction to Database Systems, Volume 1", C.J. Date, Addison-Wesley (I have the Fourth Edition from 1986).

SNO	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	30	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

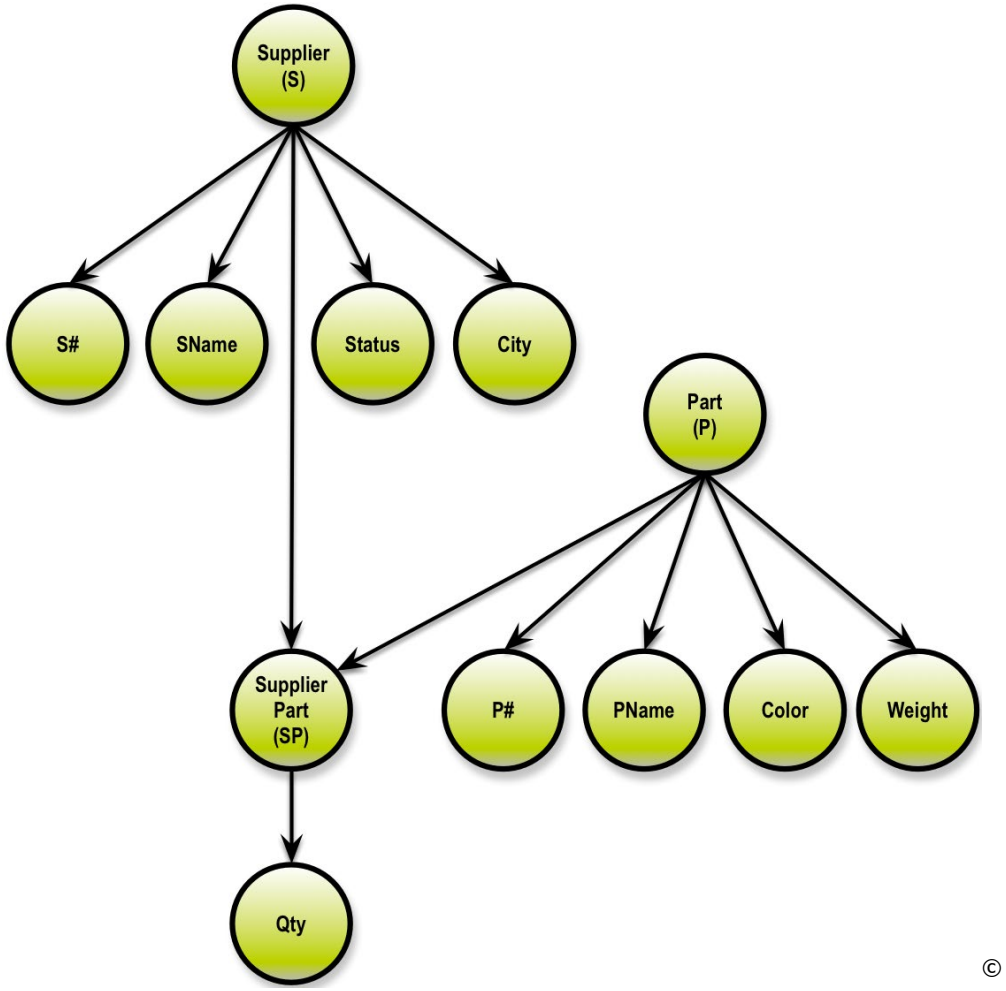
PNO	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12.0	London
P2	Bolt	Green	17.0	Paris
P3	Screw	Blue	17.0	Paris
P4	Screw	Red	14.0	London
P5	Can	Blue	12.0	Paris
P6	Cog	Red	19.0	London

SNO	PNO	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

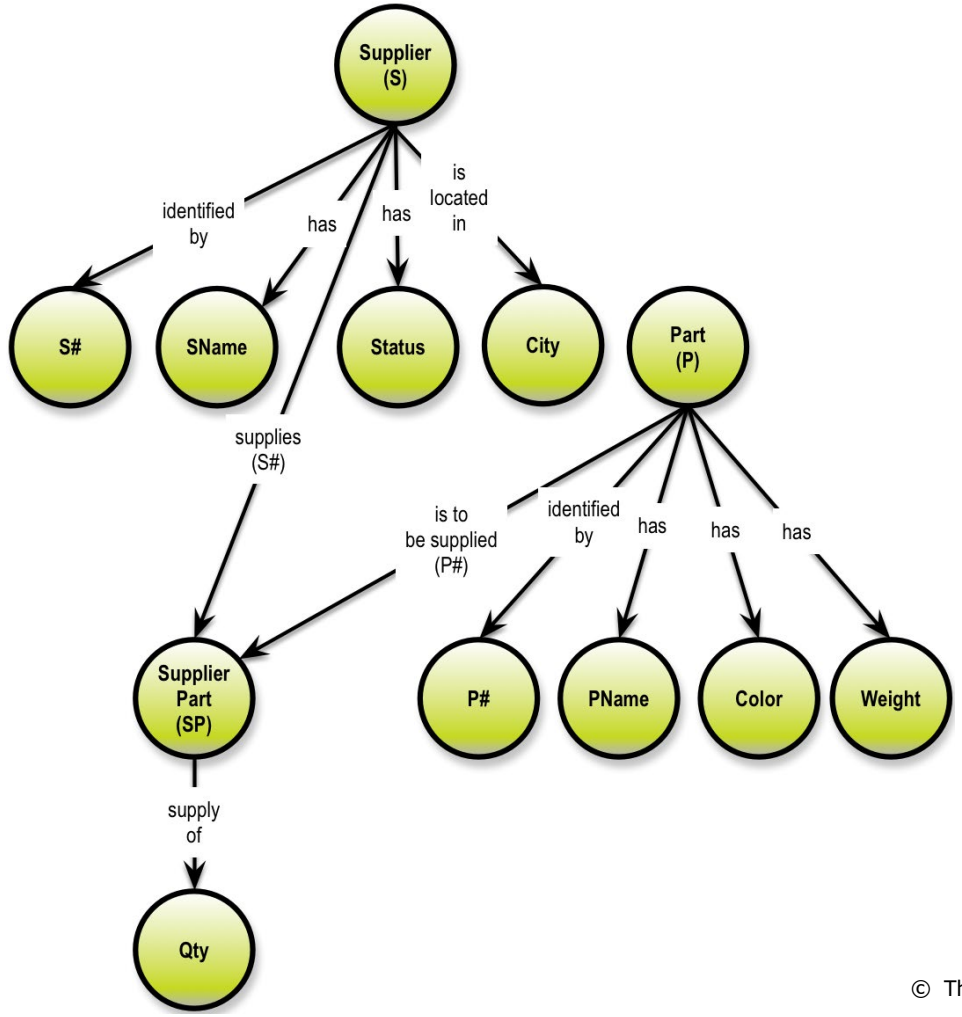
DRAW A NAIVE CONCEPT MAP



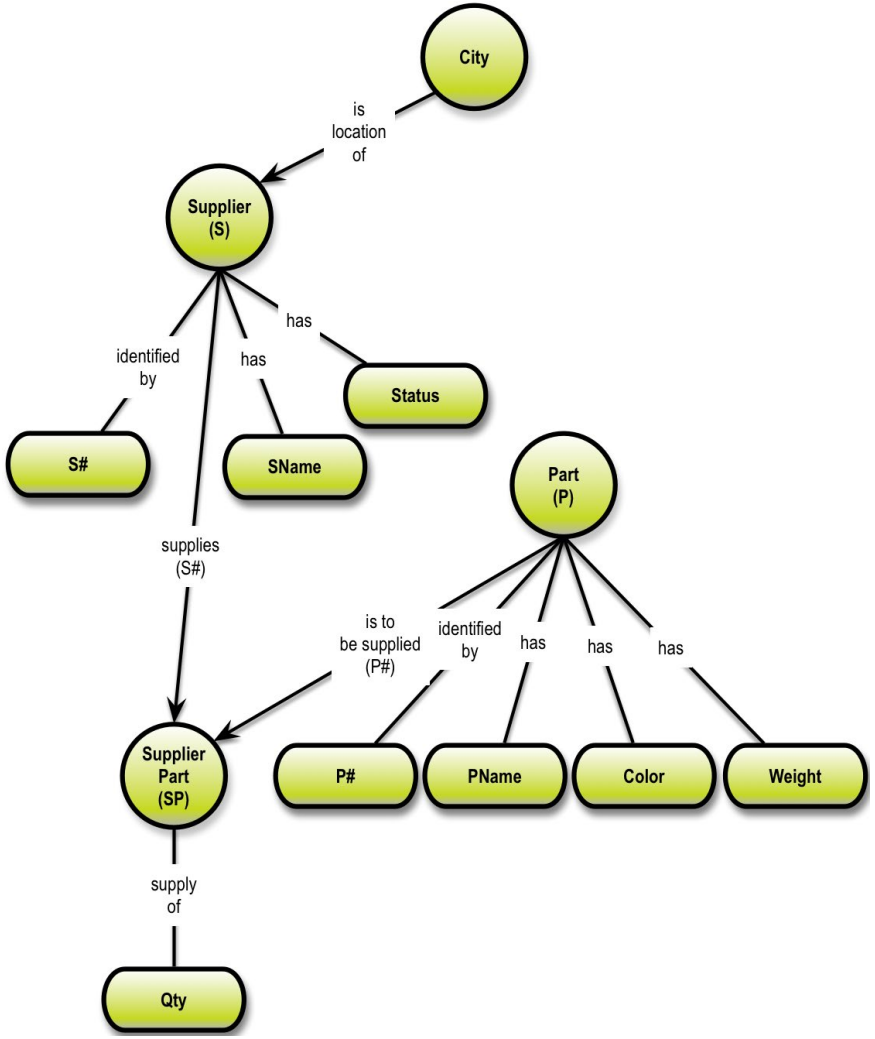
VISUALIZE IMPLICIT RELATIONSHIPS



NAME THE RELATIONSHIPS



VISUALIZE THE PROPERTIES AND FINALIZE THE CONCEPT DESIGN



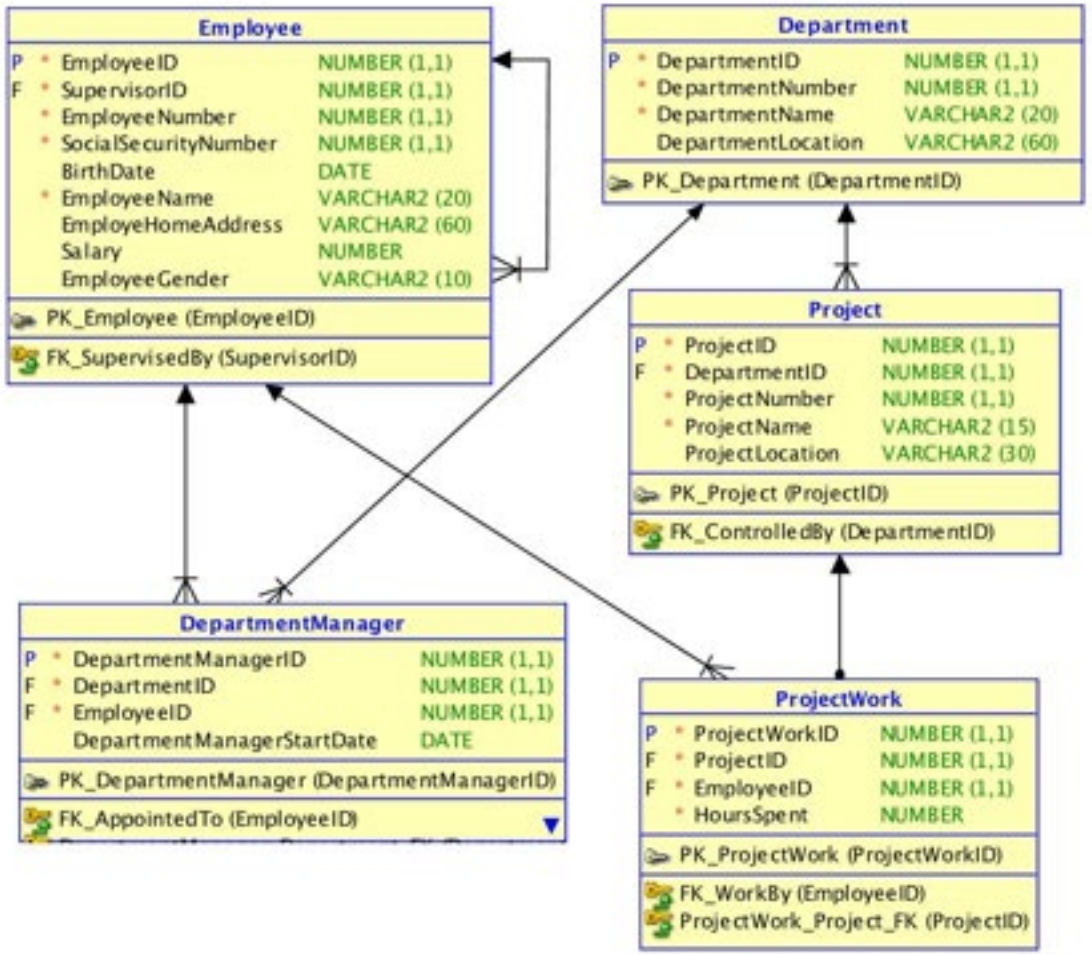
WHICH COMMUNICATES BEST?
THIS?

SNO	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	30	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

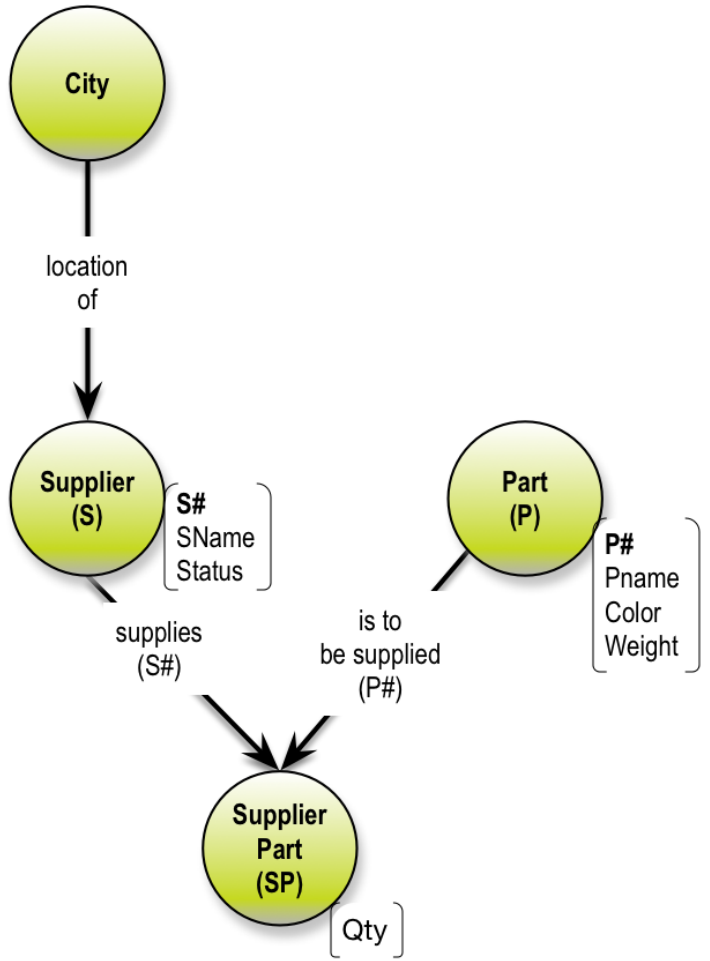
PNO	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12.0	London
P2	Bolt	Green	17.0	Paris
P3	Screw	Blue	17.0	Paris
P4	Screw	Red	14.0	London
P5	Can	Blue	12.0	Paris
P6	Cog	Red	19.0	London

SNO	PNO	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

OR THIS?



OR THIS?



THE CONTEMPORARY STYLE OF NORMALIZATION

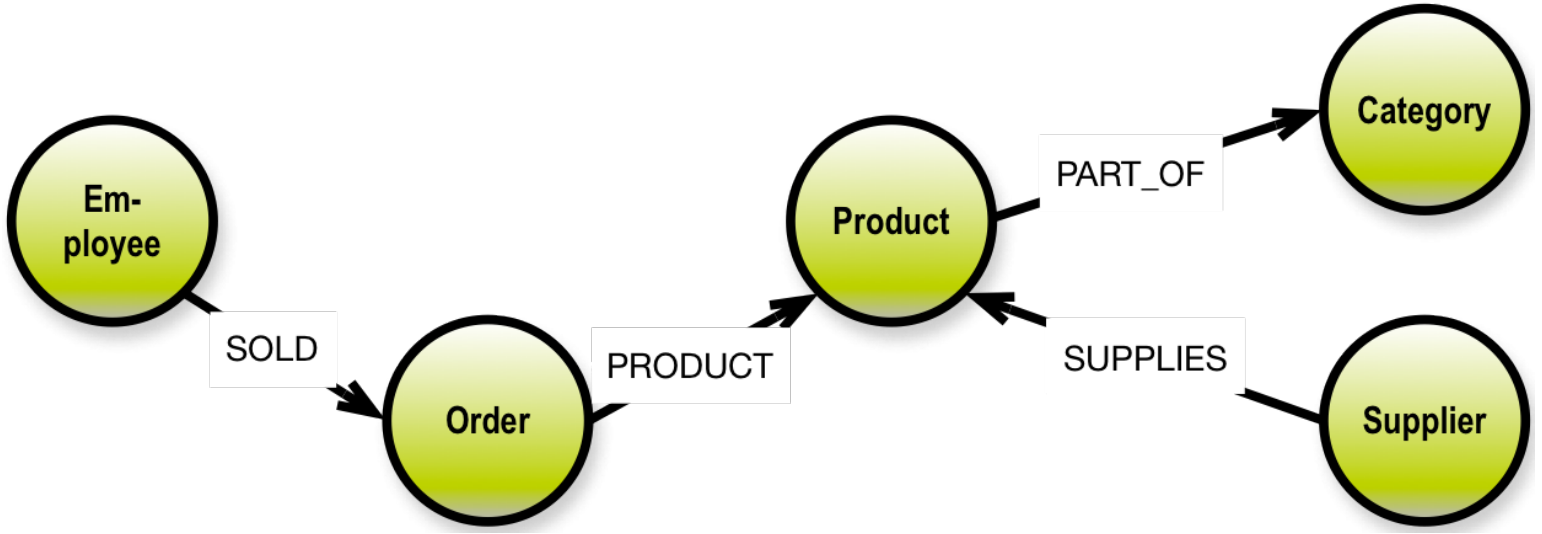
- Visualizing the dependencies as a directed graph gives a deep understanding of the functional dependency structure
- Named relationships between all concepts explain the semantics really well
- Identity and uniqueness contribute hugely to setting the context in precise ways

GRAPH DATA MODELING ARTISANSHIP

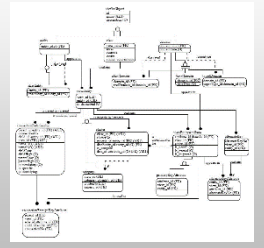
- Data is data, but you need structure and meaning to understand the data!
- Data models can be non-existent, bad, so-so or good!

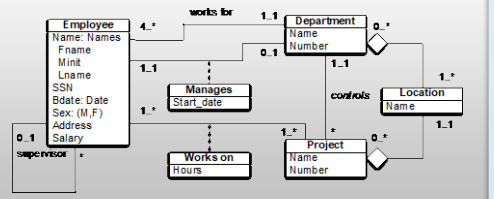
PRIMARY TAKEAWAY:

Named Relationships is what matters most in Graph Data Modeling!!!



Data Models to be Recycled





Business Concept Model

Exploration of Meaningful Connections and Knowledge

Solution Data Model

Ideation: Connected, Validated and Governable Structures and Knowledge Graphs

Physical Data Model

Implementation in a Physical GDBMS

Super Model

Load & Transform Data Model

Subset

Extend

Trans-form

Opti-mize

- **Scope**
- **New stuff**
- **Abstractions**
- **Lineage**
- **Naming relationships**
- **Uniqueness**
- **Identities**
- **Cardinalities**
- **DQ improvements**

Fast Track

Load & Transform Data

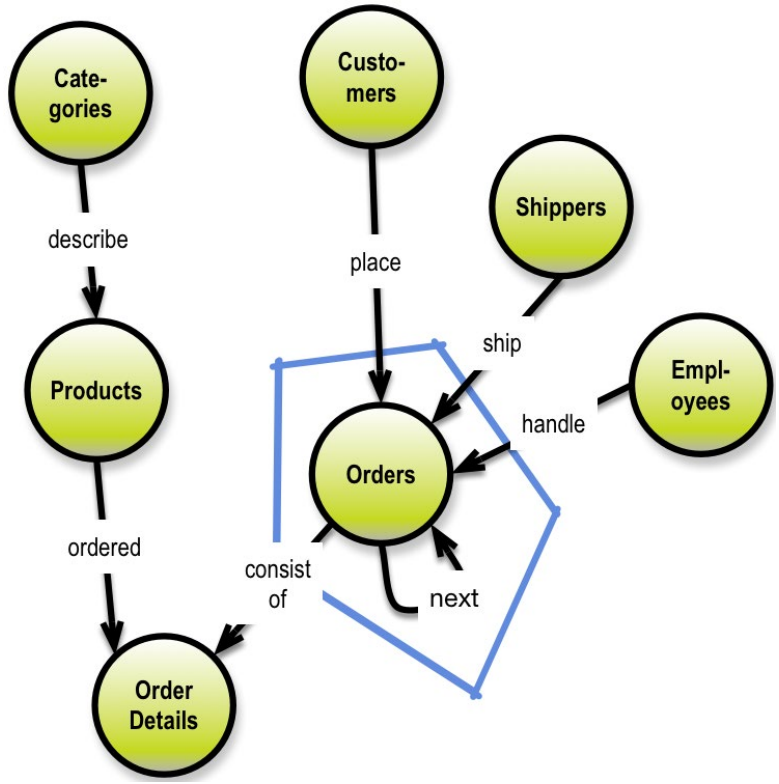
Generate a Physical Data Model

Opti-mize

"THE ATOMS AND MOLECULES" OF DATA MODELS

- Concepts, which materialize as either
  - Object types (Customer, Order etc.), or
  - Properties (CustomerNumber, ProductName etc., some of which are mandatory)
- Dependencies, which structure the model as either:
  - Functional dependencies (internal within the object type and determining the primary key), or
  - Intra-object relationships (like "business address of" and so forth), which have
    - Cardinalities
    - An associated key, which can be either a single identity or a combined set of uniqueness criteria
    - a type (name, not always present)
    - a direction (from/to, not always present)
- Uniqueness criteria (object type level),
  - either an identifying property (key) or
  - (frequently) a list of concatenated properties
- Identity (object type level),
  - either a single identifying property (key) like CustomerNumber, or
  - a system generated surrogate key
- Data types

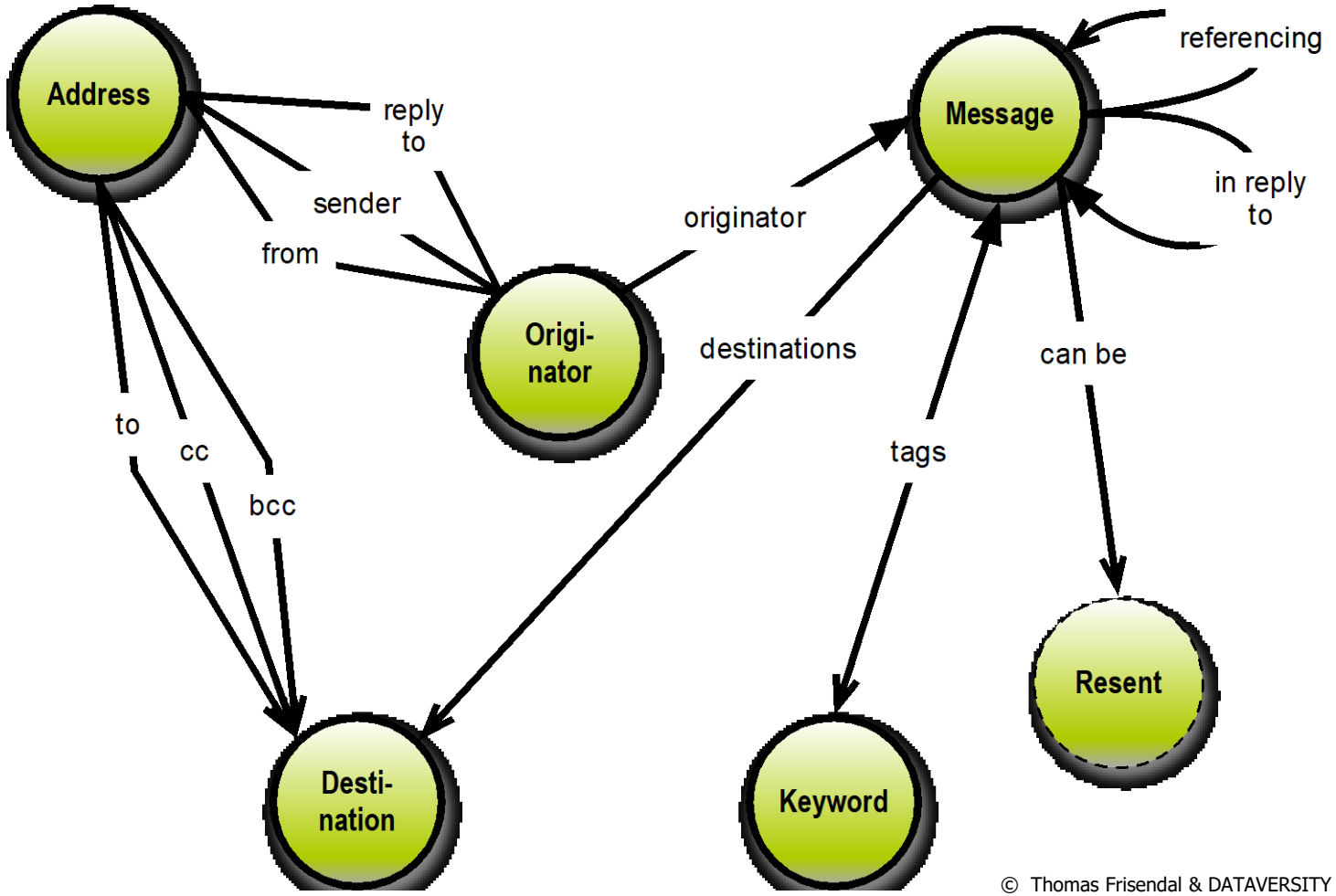
PROPERTY GRAPH DATABASES - THE UBIQUITOUS POINTER!



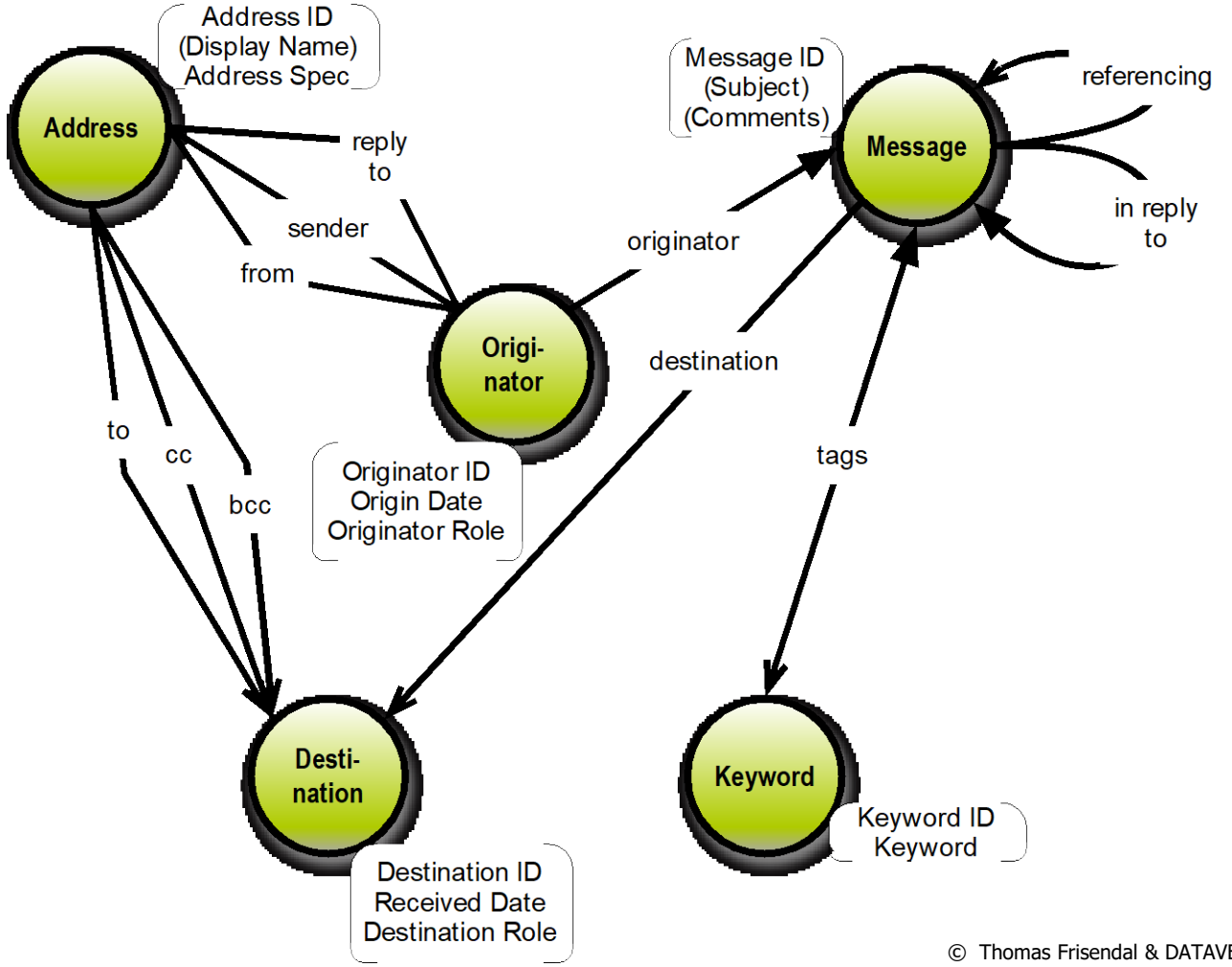
Next, prior, first, last, time-series ….

*https://commons.wikimedia.org/wiki/File%3ACute_young_afro_american_boy_child.jpg by Hillebrand Steve, U.S. Fish and Wildlife Service [Public domain], via Wikimedia Commons.

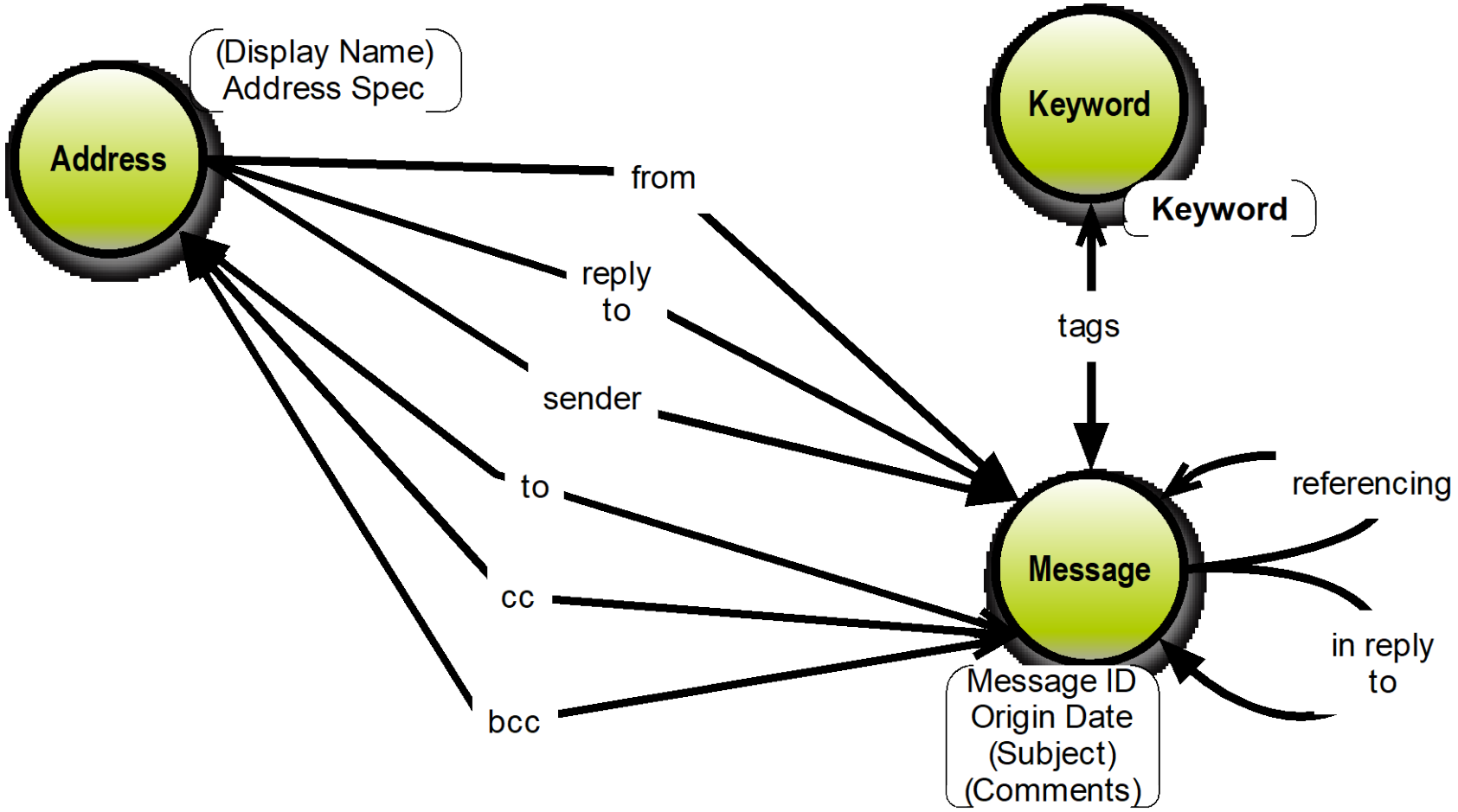
EMAIL CONCEPTS



NAIVE
EMAIL PROPERTY GRAPH



A "GRAPHISH"
EMAIL PROPERTY GRAPH



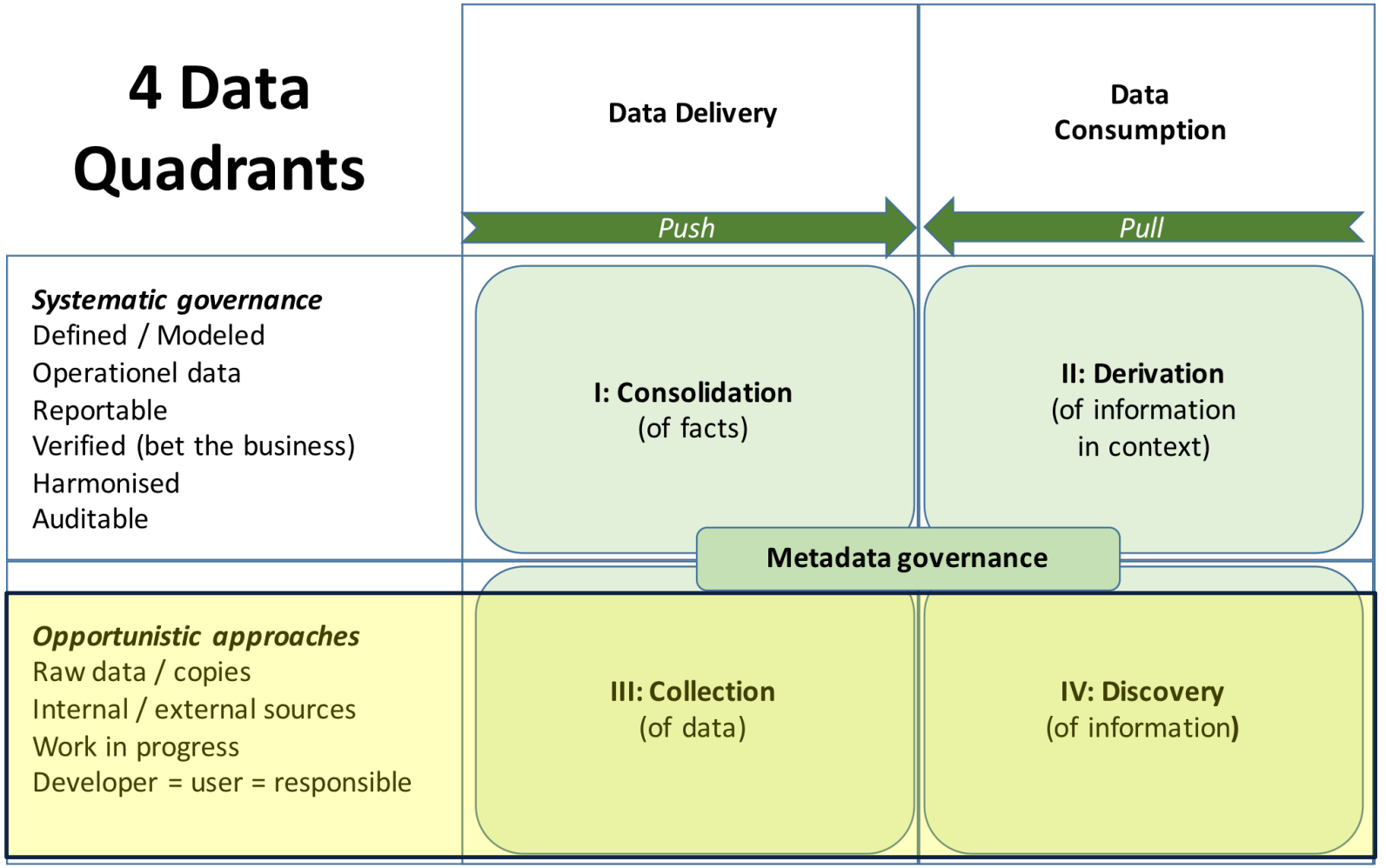
WHAT ABOUT M:M?

- **Relational**:
  - Relationship table (bridge)
  - No(?) properties?
  - ”Factless facts” in star schemas
  - Non-information bearing M:M relationships are relatively seldom:
    - At closer inspection they most often carry information
    - A quantity, an amount, a percentage, something related to the relatedness
- **Graph**
  - Many to many works fine without additional modelling
  - But: No object-owned properties on relationships, except for simple things
  - Objects / events are nodes
  - Objects / events may be role-playing (a reply is also an email)
  - Objects / events may be state-changing (Sold, Invoiced, Paid for)

GRAPH DATA MODELING AGILITY

- Schema first
- Schema less
- Schema eventually

TWO-BY-TWO DIMENSIONS: MANAGE THE CONCERNS, WHICH ARE COMPETING

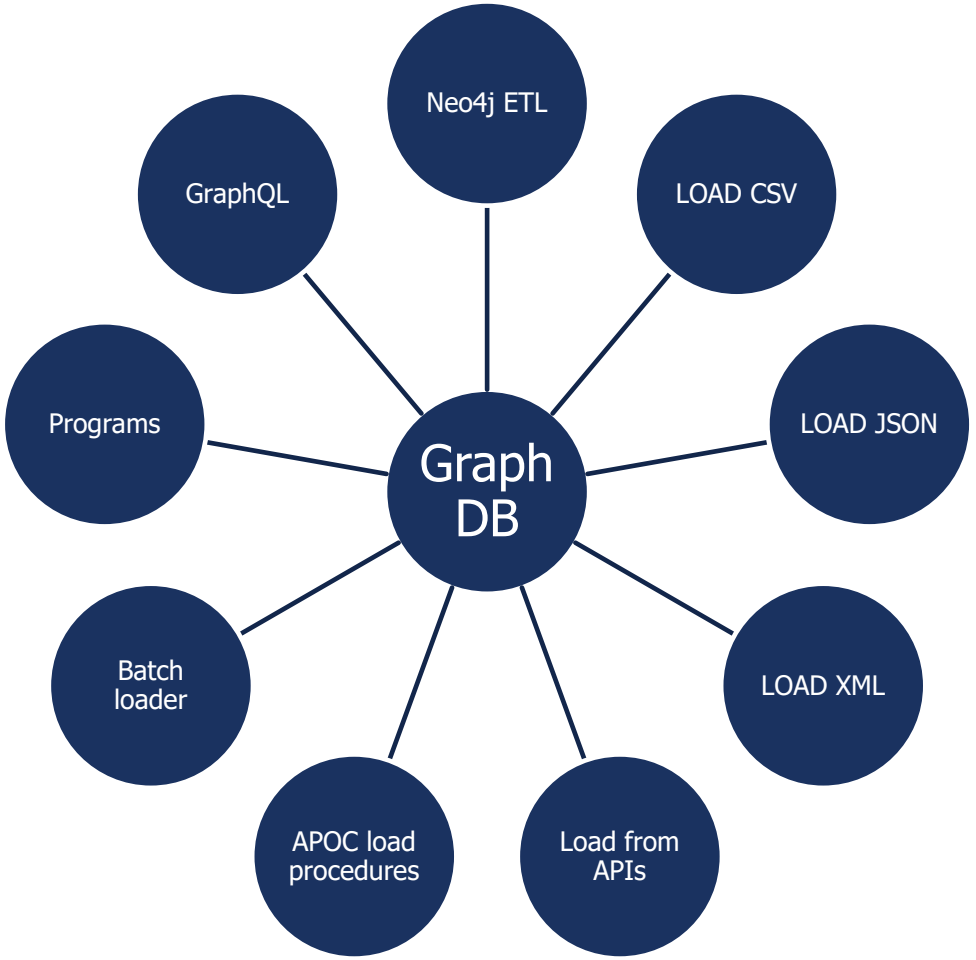


Inspirered by Ronald Damhof / Martijn Evers, Full Scale Data Architects. See fx: https://bit.ly/2St9ROn

THE BULL IS COMING AT YOU!

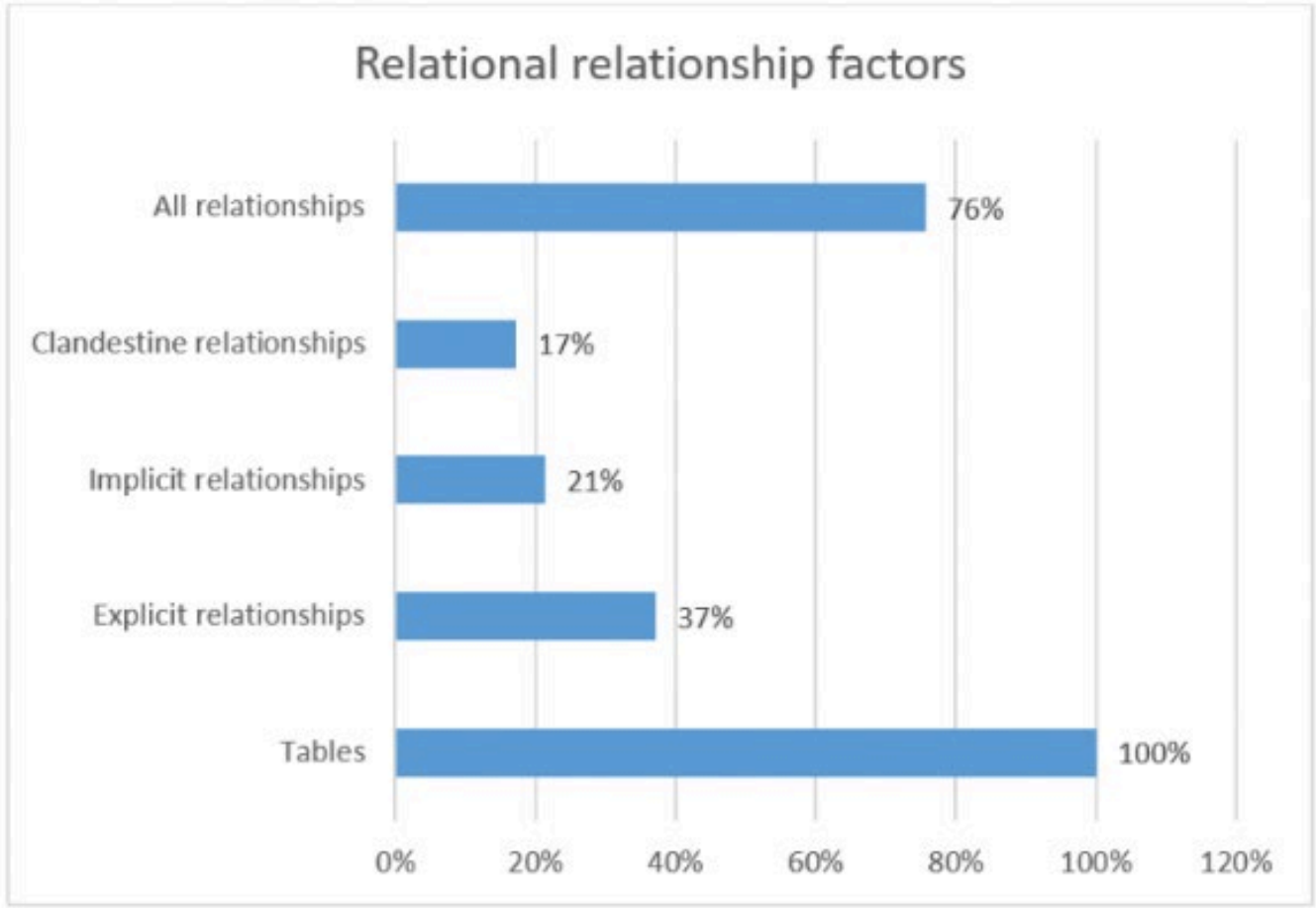
More on Facebook: https://www.facebook.com/FullScaleDataArchitects/,
and on Meetup: https://www.meetup.com/Full-Scale-Data-Architects/

WAYS TO LOAD DATA INTO A GRAPH DATABASE (NEO4J)



TWO MAJOR CHALLENGES

- What is the quality of the table and column names in the source?
- What is the quality of the relationship names?
  - Not all relationships declared (only 37 %)
  - Foreign key names?
    - FROM-TABLE_TO-TABLE?
    - PRIMARYKEY_FOREIGNKEY?
  - Physical primary keys are surrogate keys in many databases



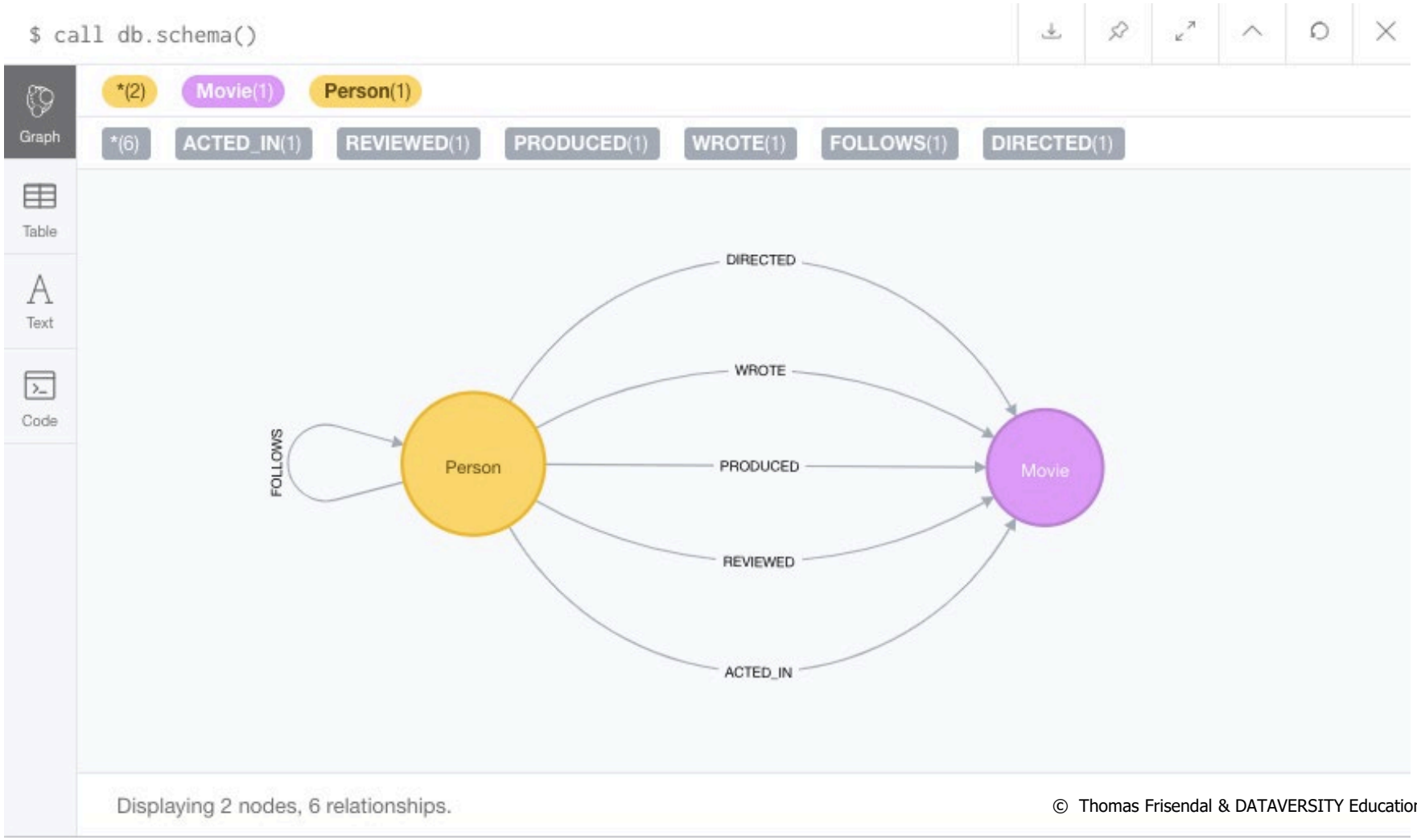
THE HUMAN FACTORS

- If data models are to be derived from the data, we need to master the Data Quality aspects rather completely
- A given data set may be interpreted in more than one way, and there are bound to be situations, where human judgement is necessary
- We are still better than machines when it comes to understanding a given context. Why is that? Because humans set the rules of the context.
- In our very human ways we leave room for oversights, simplifications, peculiar terminology, and laissez-faire "known errors." The rightful owners of the business terminology are the business people, and they are more human than they are androids.
- The path from data values to a context description (another way to describe what a data model is) is long and works upwards in multiple layers.
- The bottom layers is traditional stuff from Data Quality but the upper layers really do not have a good name, yet. "Smart Data Discovery" is more a marketing term and there are no clearly sovereign technologies.

PROFILING YOUR GRAPH DATA

WHAT DID I LOAD?

INFERRING THE SCHEMA FROM THE DATA (NEO4J)



HOW MANY NODE TYPES AND RELATIONSHIPS?

```
MATCH (n)
RETURN DISTINCT labels(n),
count(*) AS SampleSize,
avg(size(keys(n))) as Avg_PropertyCount,
min(size(keys(n))) as Min_PropertyCount,
max(size(keys(n))) as Max_PropertyCount,
avg(size( (n)-[]-() ) ) as Avg_RelationshipCount,
min(size( (n)-[]-() ) ) as Min_RelationshipCount,
max(size( (n)-[]-() ) ) as Max_RelationshipCount
```

labels(n)	OccCount	Avg_PropertyCount	Min_PropertyCount	Max_PropertyCount	Avg_RelationshipCount	Min_RelationshipCount	Max_RelationshipCount
["Movie"]	38	2.973684210526316	2	3	6.578947368421054	2	14
["Person"]	133	1.9624060150375942	1	2	1.924812030075188	1	13

EVALUATE THE UNIQUENESS OF A PROPERTY WITHIN A LABEL

```
MATCH (m:Movie)
RETURN count(DISTINCT m.title) AS DistinctTitle,
    count(m.title) AS TotalTitle,
    100*count(DISTINCT m.title)/count(m.title) AS Uniqueness
```

```
$ MATCH (m:Movie)  RETURN count(DISTINCT m.title) AS DistinctTitle,  count(…
```

Table

DistinctTitle	TotalTitle	Uniqueness
38	38	100

WHICH PROPERTIES ON RELATIONSHIPS?

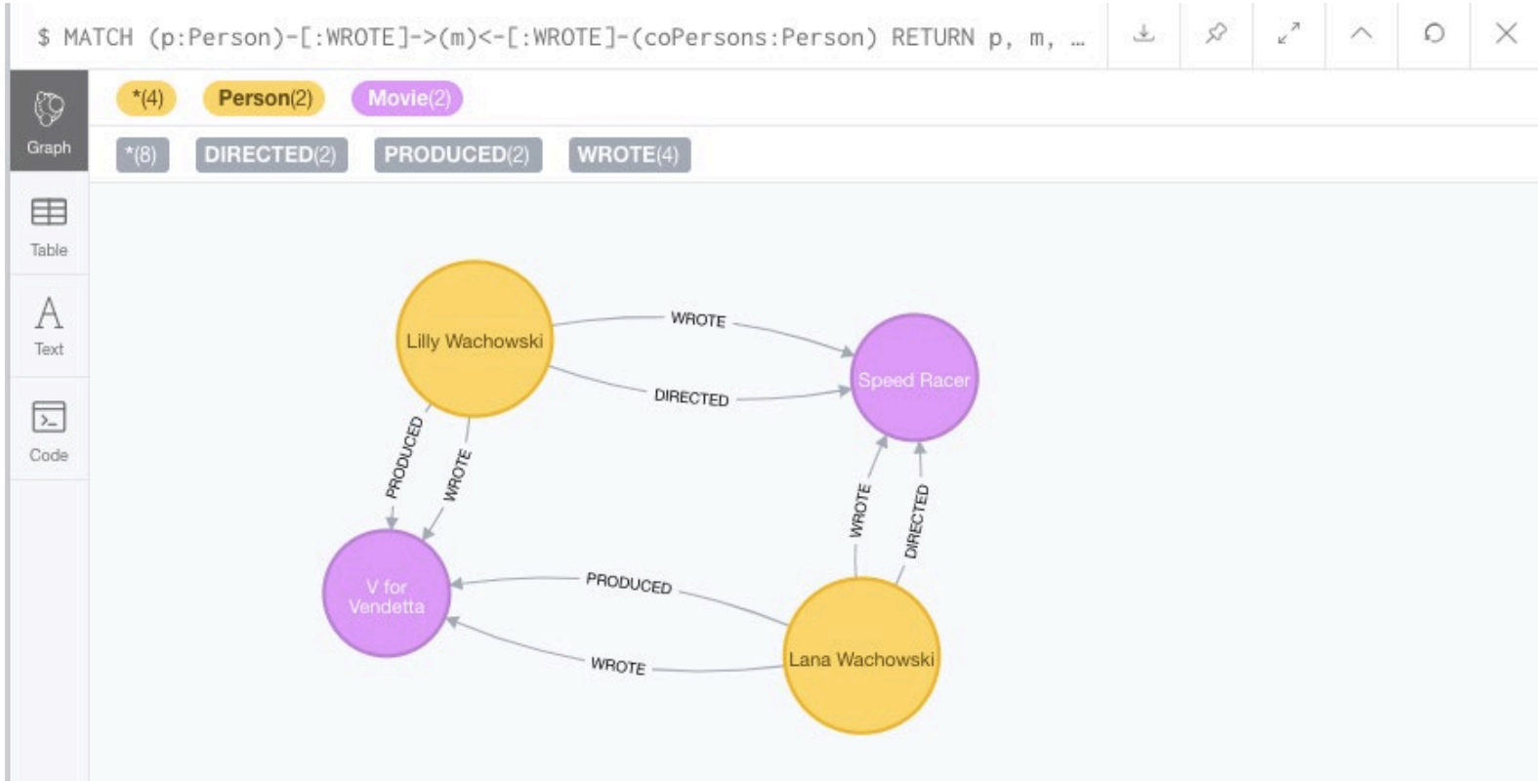
```
$ call db.schema.relTypeProperties()
```

Table | Text | Code

relType	propertyName	propertyTypes	mandatory
":`ACTED_IN`"	"roles"	["StringArray"]	true
":`DIRECTED`"	null	null	false
":`PRODUCED`"	null	null	false
":`WROTE`"	null	null	false
":`FOLLOWS`"	null	null	false
":`REVIEWED`"	"summary"	["String"]	true
":`REVIEWED`"	"rating"	["Long"]	true

FINDING M:M CARDINALITIES ACROSS RELATIONSHIPS

```
MATCH (p:Person)-[:WROTE]->(m)<-
[:WROTE]-(coPersons:Person)

RETURN p, m, coPersons
```



IDENTITY, UNIQUENESS, AND KEYS

- Surrogate keys
- Customer -> Order -> Orderline -> Product
- The ”zero records”:
- Consider GUID’s

CustomerID	FirstName	LastName	FullName
0	Not specified	Not specified	Not specified
291	Gustavo	Achong	Gustavo Achong
293	Catherine	Abel	Catherine Abel
295	Kim	Abercrombie	Kim Abercrombie
297	Humberto	Acevedo	Humberto Acevedo
299	Pilar	Ackerman	Pilar Ackerman
301	Frances	Adams	Frances Adams
303	Margaret	Smith	Margaret Smith
305	Carla	Adams	Carla Adams
307	Jay	Adams	Jay Adams
309	Ronald	Adina	Ronald Adina
311	Samuel	Agcaoili	Samuel Agcaoili
313	James	Aguilar	James Aguilar
315	Robert	Ahlering	Robert Ahlering

GRAPH REFACTORING – PART OF THE APOC LIBRARY

- Just to give you a sense of what refactoring looks like:

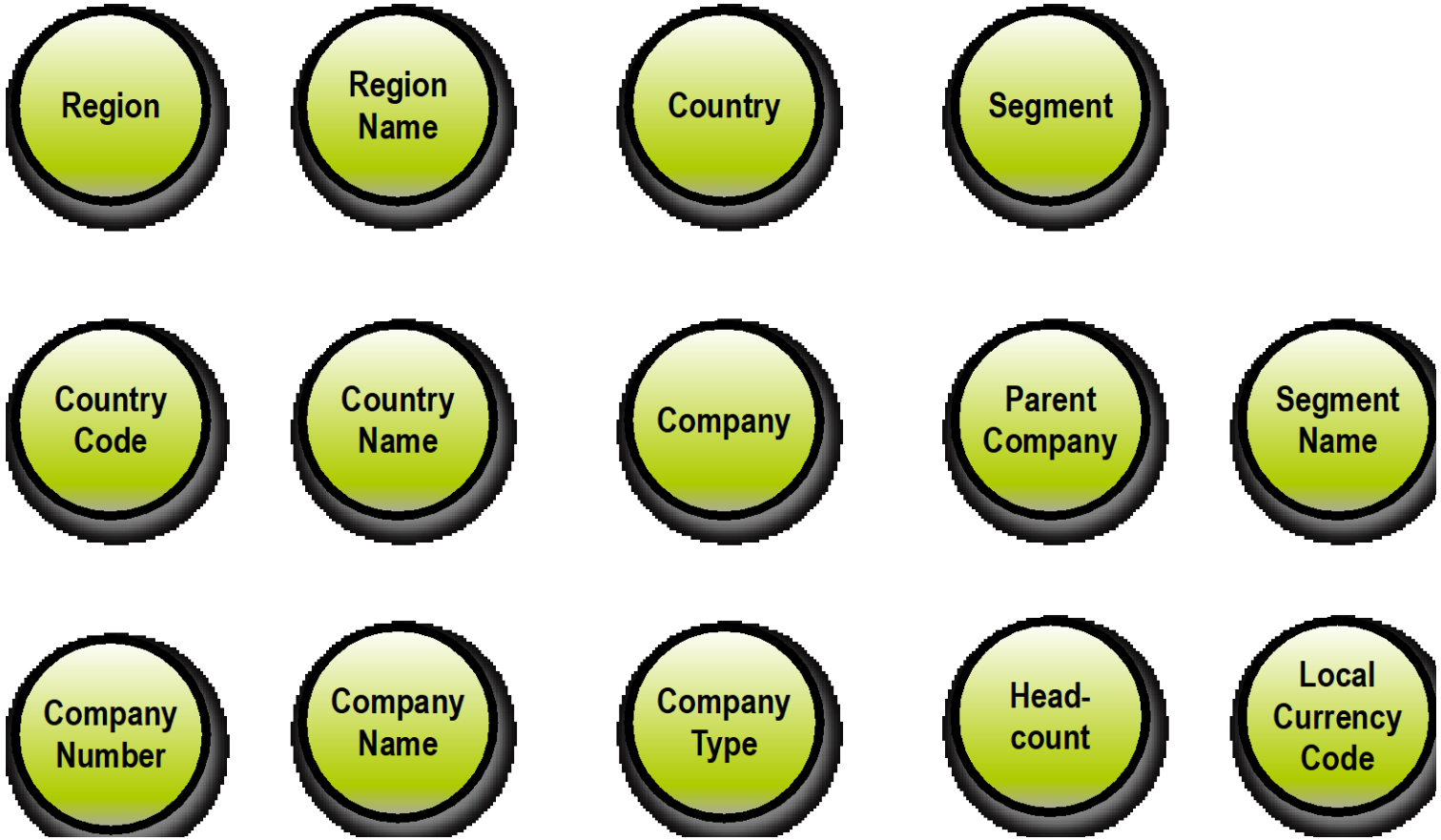
```
MATCH (a1:Person{name:'John'}), (a2:Person {name:'Tom'})

WITH head(collect([a1,a2])) as nodes

CALL apoc.refactor.mergeNodes(nodes,{properties:"combine", mergeRels:true}) yield node

MATCH (n)-[r:WORKS_FOR]->(c) return *
```

- Learn more: https://www.youtube.com/watch?v=wUMKM-uLNfg
- You may, of course, also DELETE and (re-)load parts of the graph to make structural changes

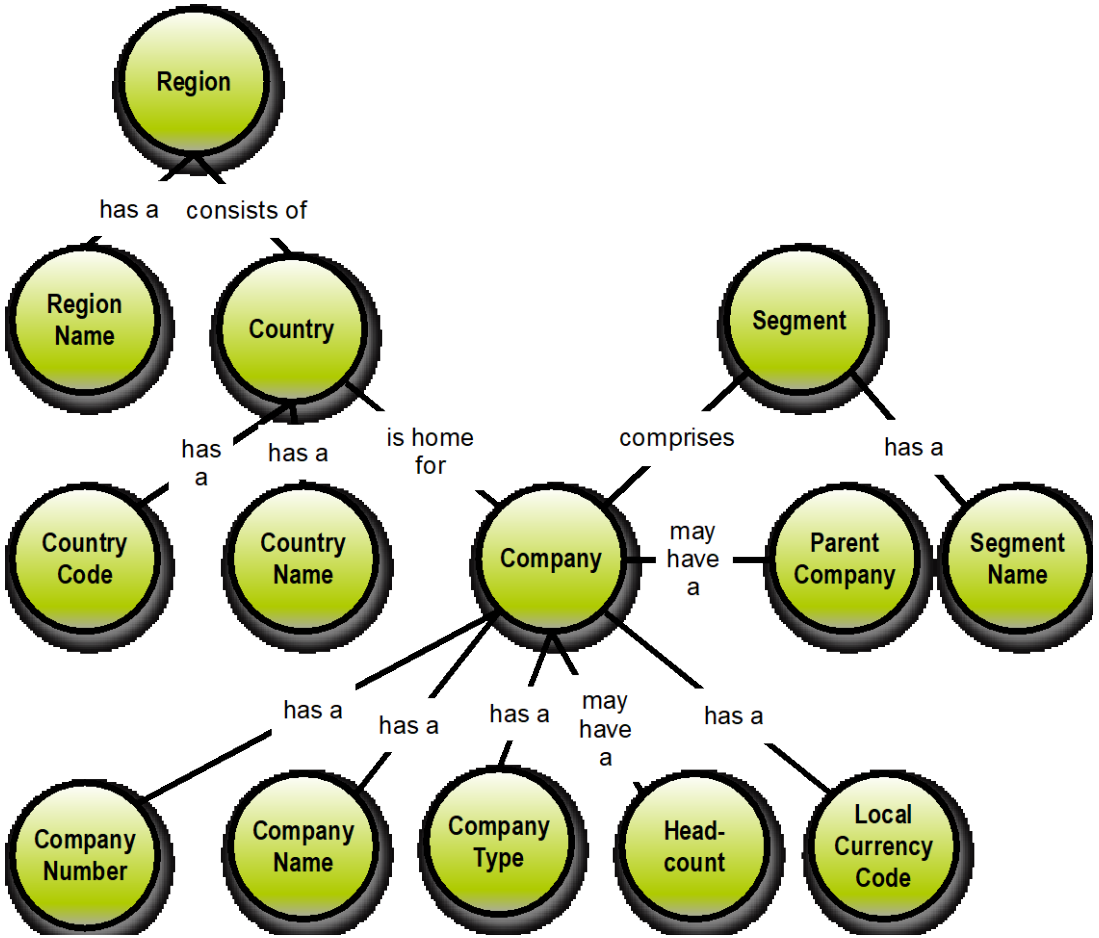
SUMMARY

THE GRAPH DATA MODELING APPROACH

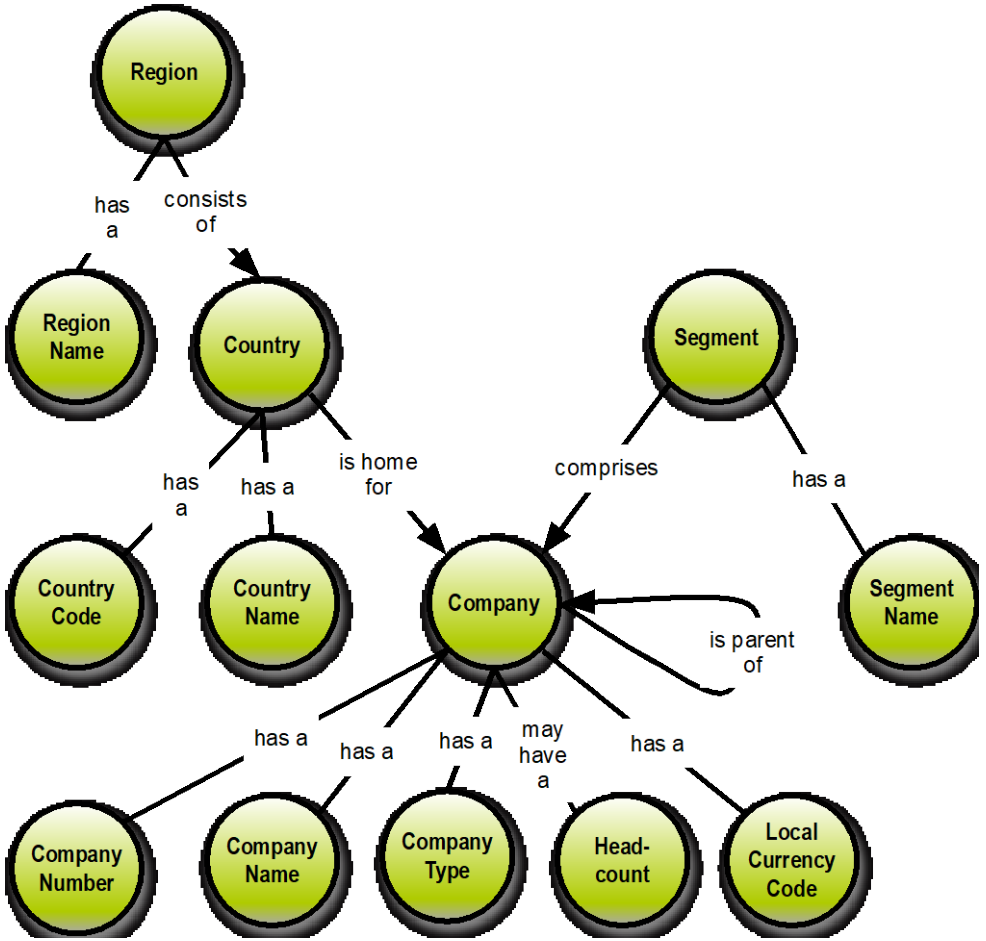
THE LIST OF CONCEPTS



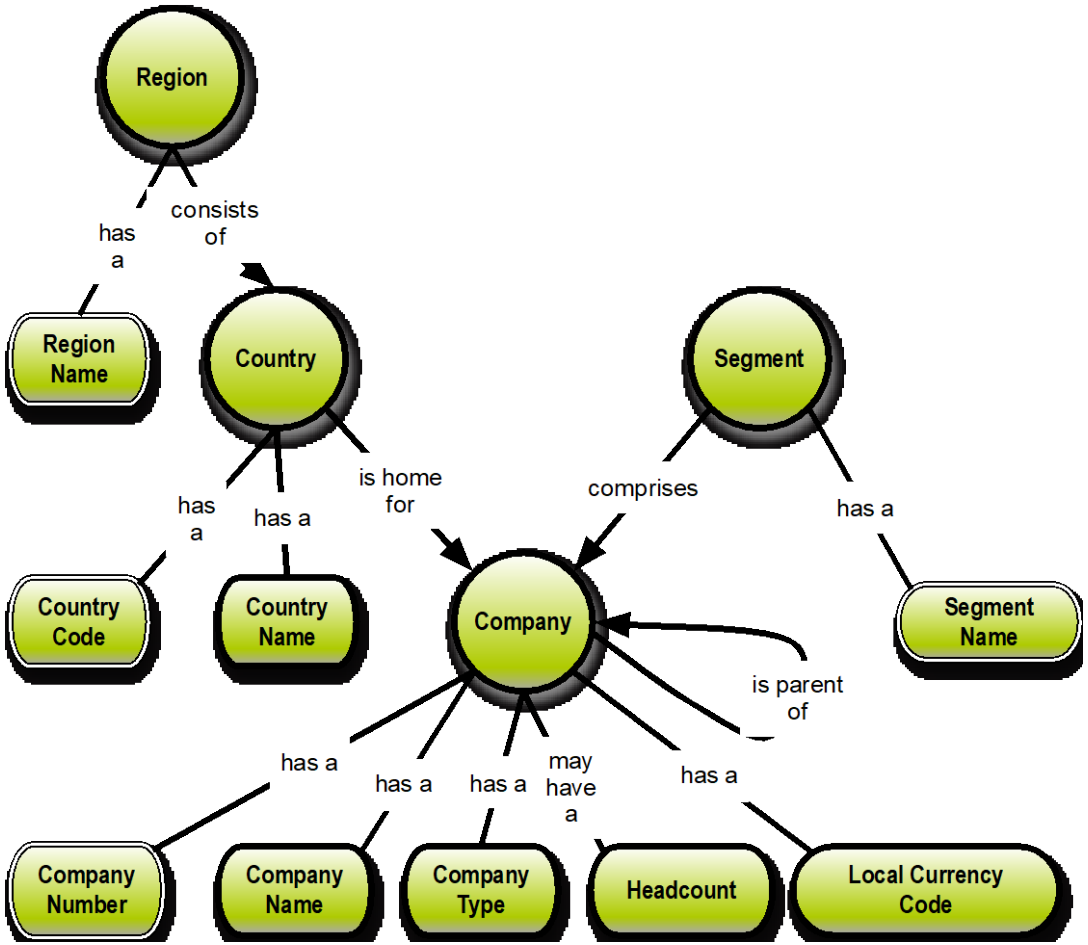
THE CONNECTED DATA MODEL



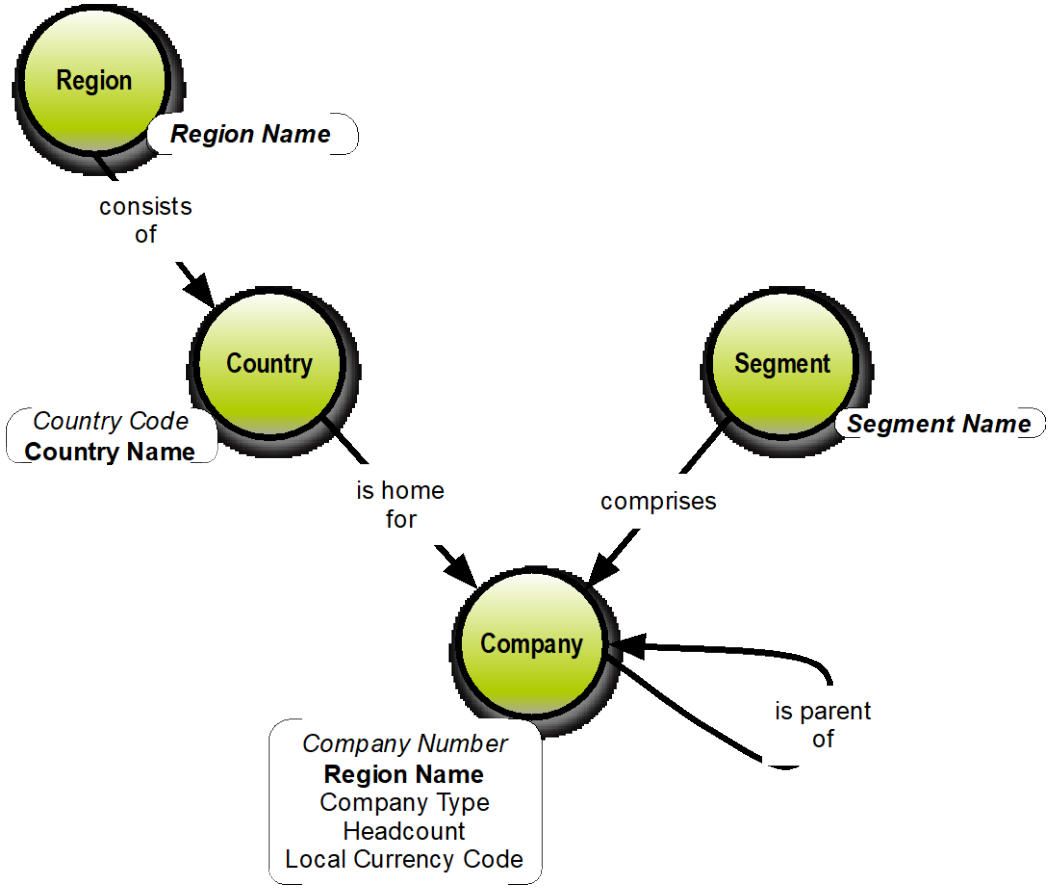
THE STRUCTURED DATA MODEL



THE GOVERNABLE DATA MODEL



THE LABELED PROPERTY GRAPH VERSION OF THE GOVERNABLE DATA MODEL

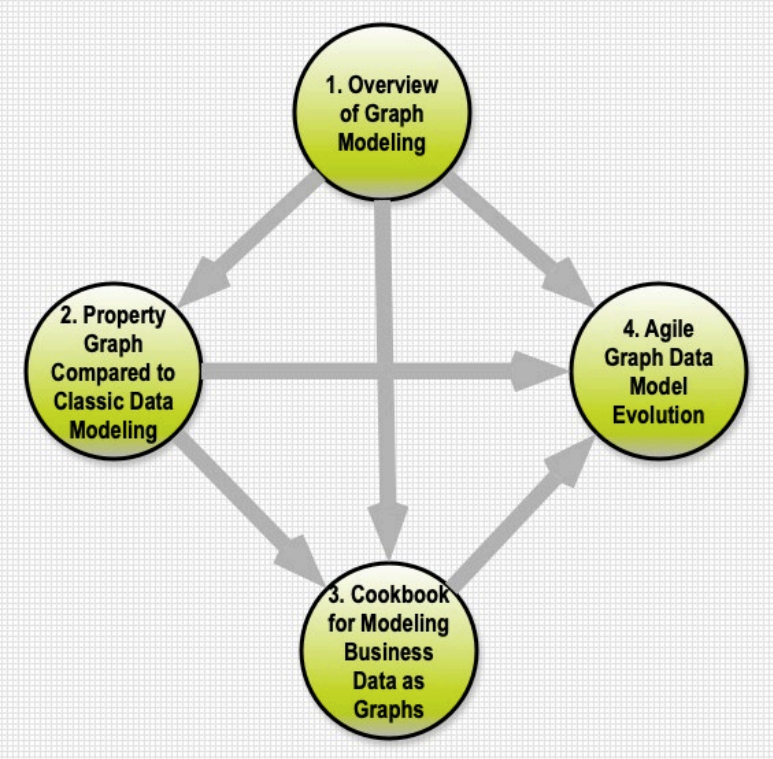


MODELING DATA AS GRAPHS – COURSES

Modeling Data as Graphs Learning Plan

at training.dataversity.net

- Course 1 – Overview of Graph Modeling
- Course 2 – Property Graph Compared to Classic Data Modeling
- Course 3 – Cookbook for Modeling Business Data as Graphs
- Course 4 – Agile Graph Data Model Evolution



THANK YOU!

- Thomas Frisendal
- Copenhagen (close to the airport)
- thomasf@tf-informatik.dk
- Member of the ISO IEC/JTC1/SC32/WG3 Database Languages Standards Comittee
- @VizDataModeler
- www.graphdatamodeling.com
- https://www.linkedin.com/in/thomas-frisendal-19a56a