goto; conference aarhus

MODELING WITH GRAPHS

Alistair Jones

Neo Technology

INTERNATIONAL SOFTWARE DEVELOPMENT CONFERENCE

gotocon.com

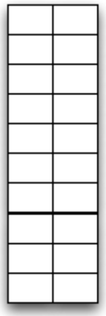
What is modeling?

Model Complexity

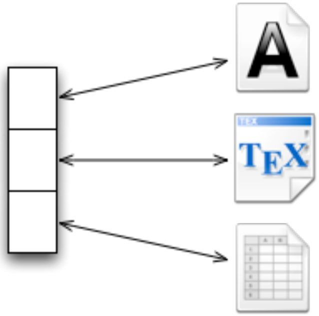
Simple

Complex

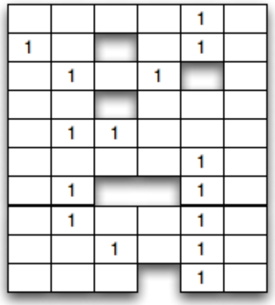
Key-Value



Document

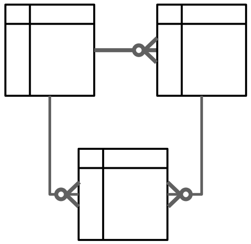


Column



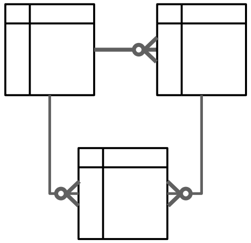
Graph

Relational



This talk compares

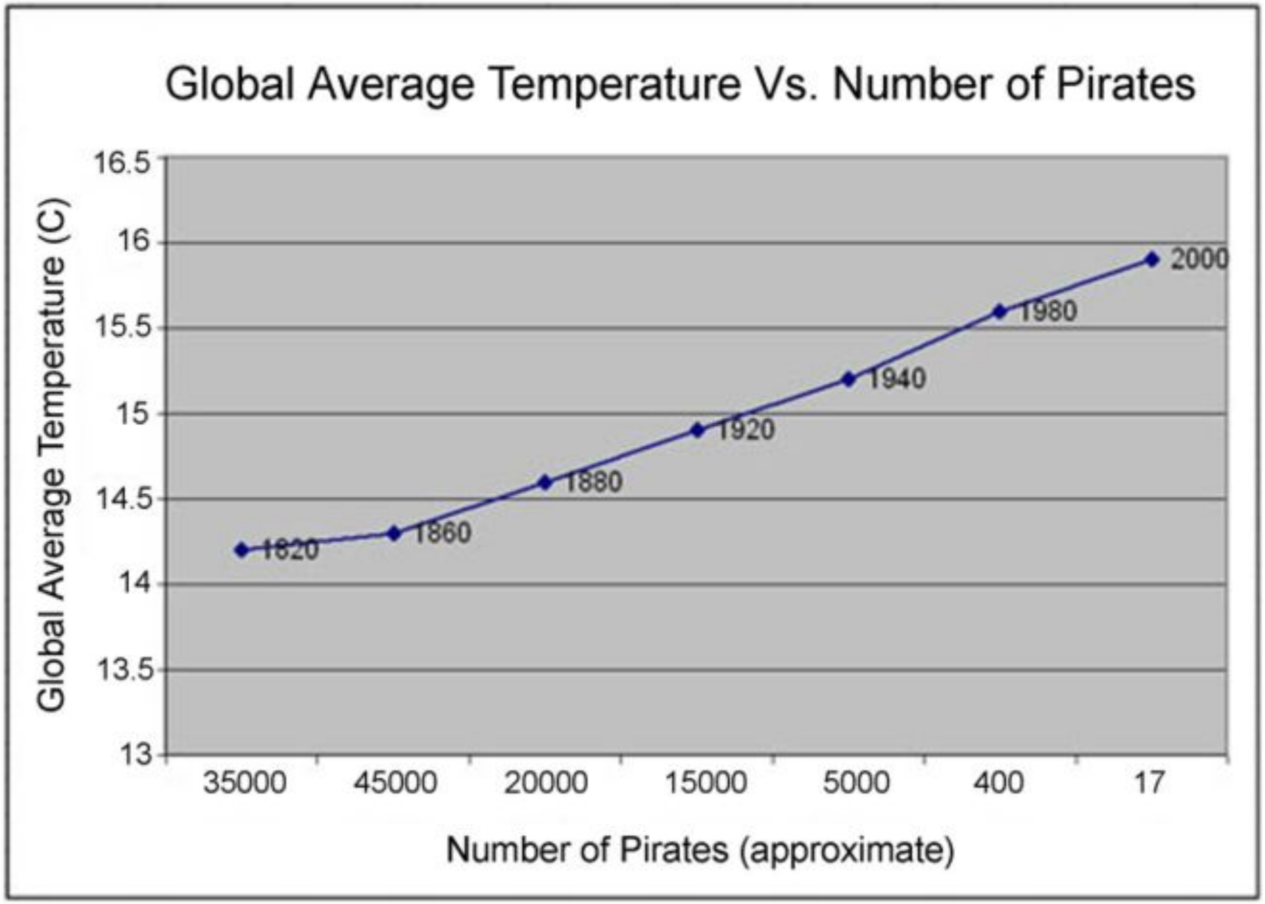
Relational



Graph DB

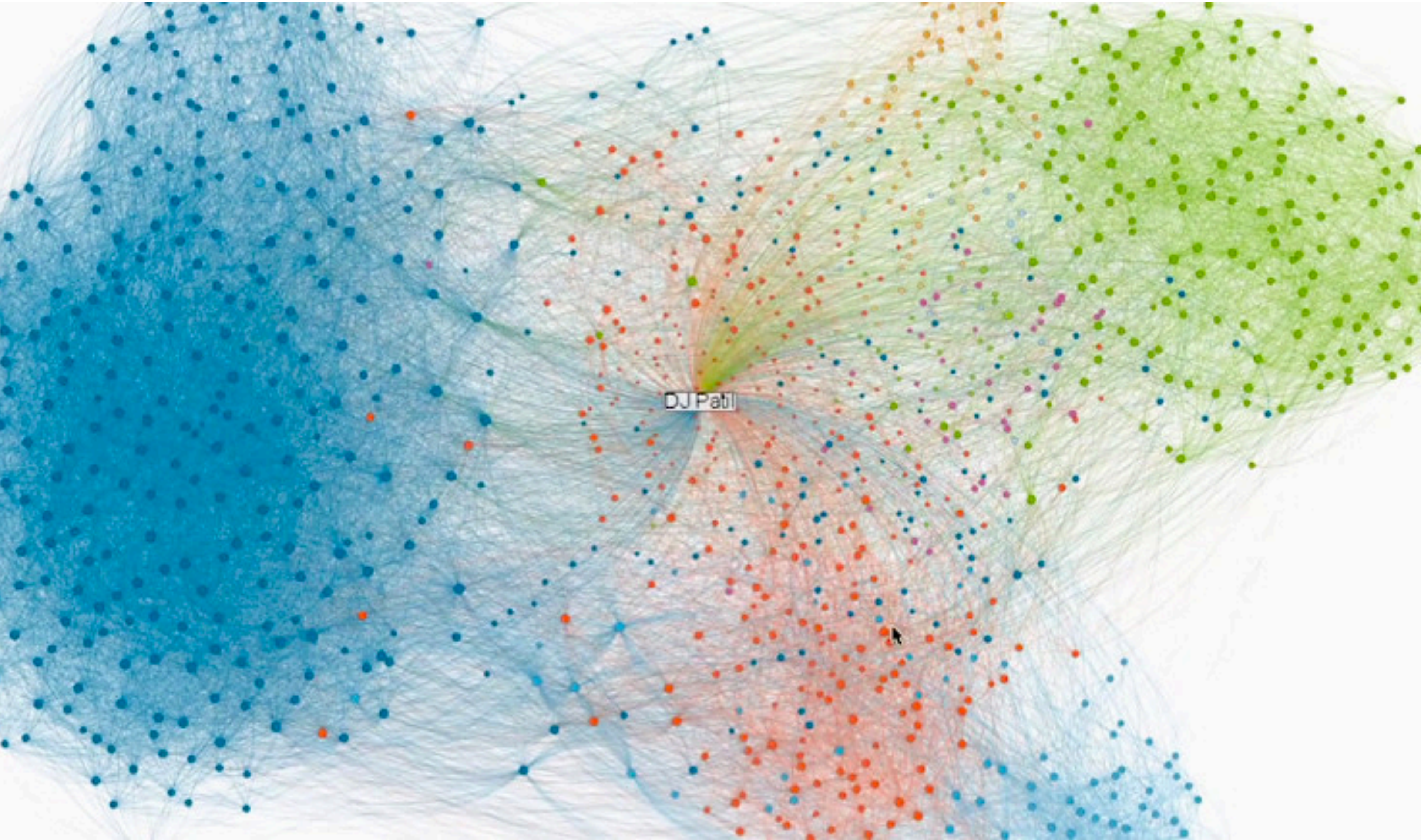
What’s a graph?

STOP GLOBAL WARMING: BECOME A PIRATE

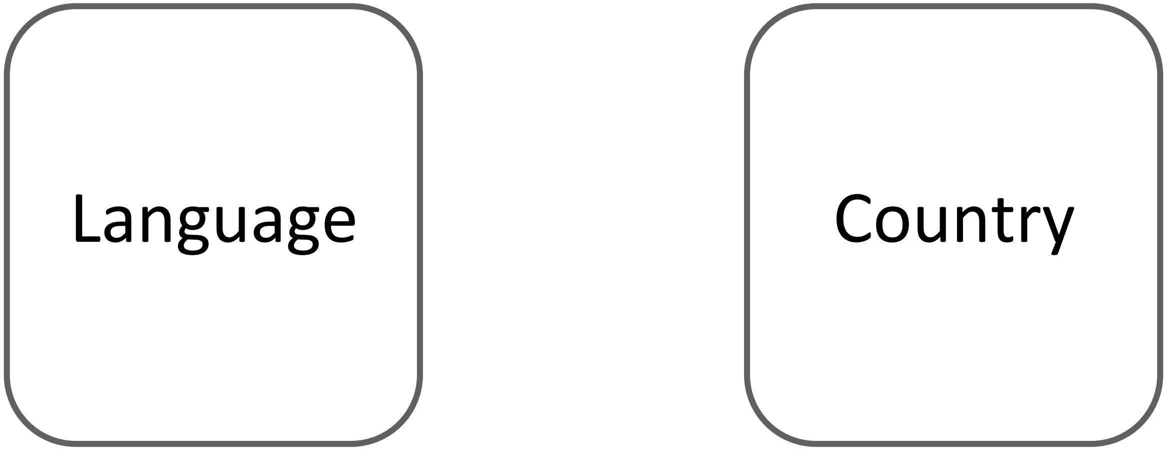


WWW.VENGANZA.ORG

Social (network | graph)



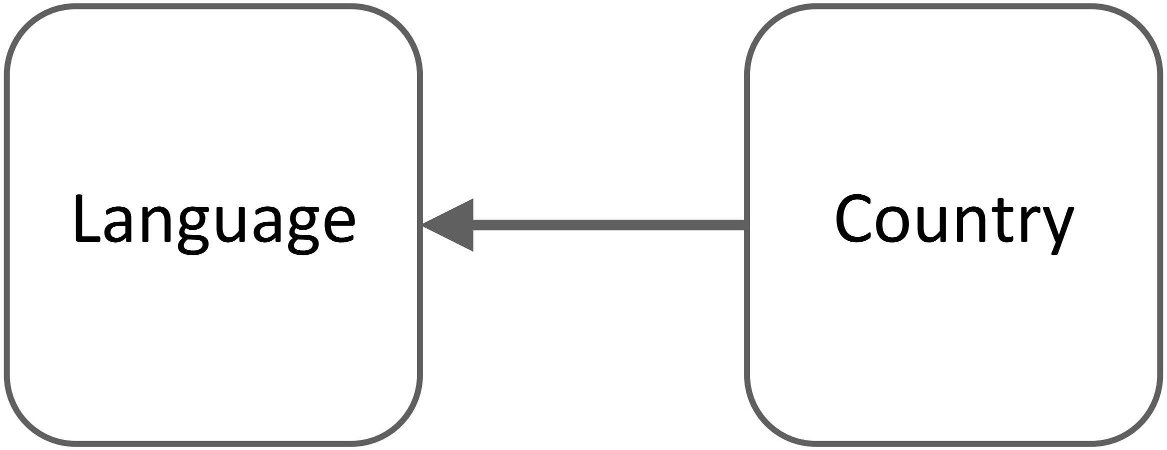
What language do they speak here?



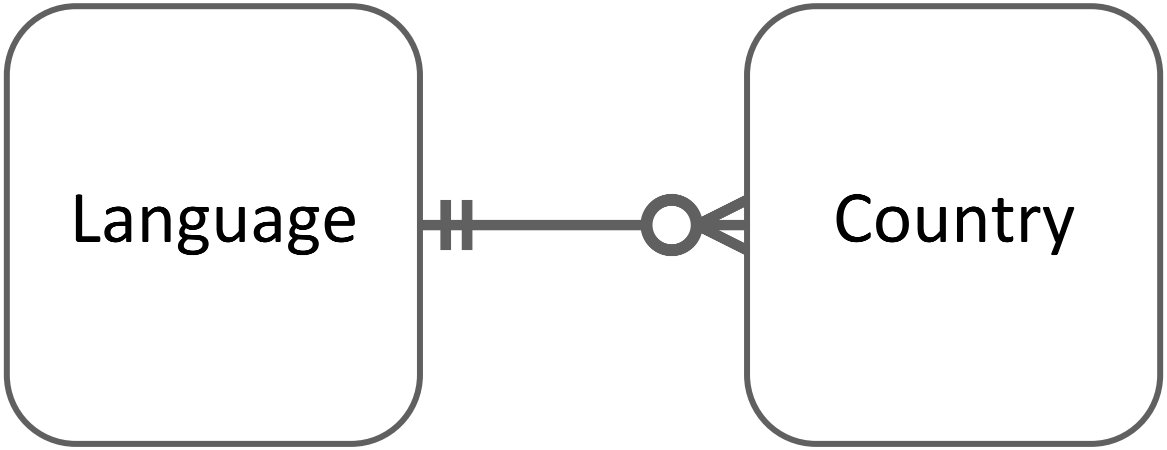
Language

Country

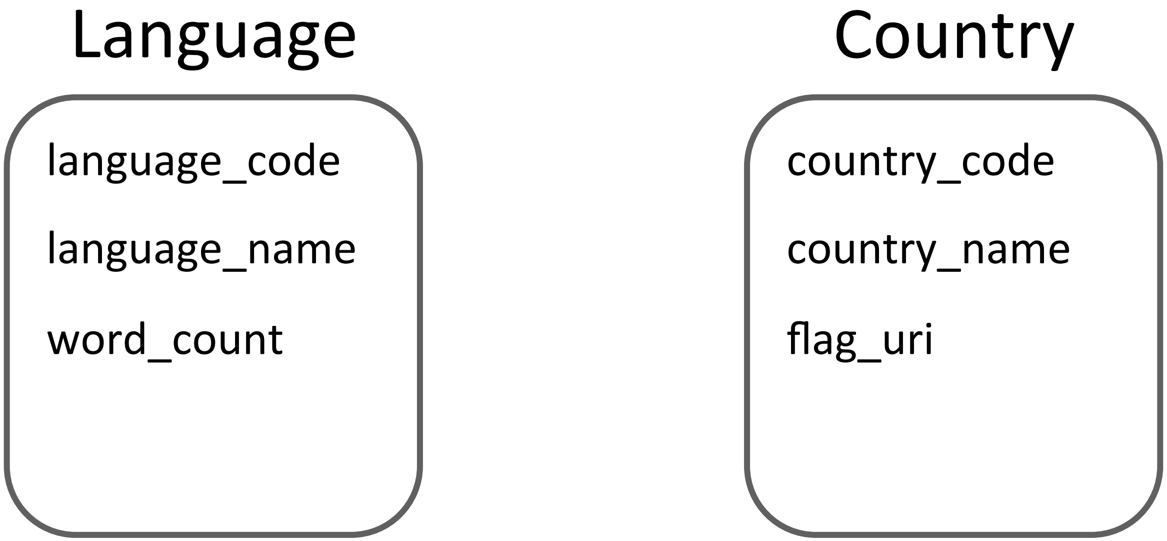
What language do they speak here?



What language do they speak here?



Tables



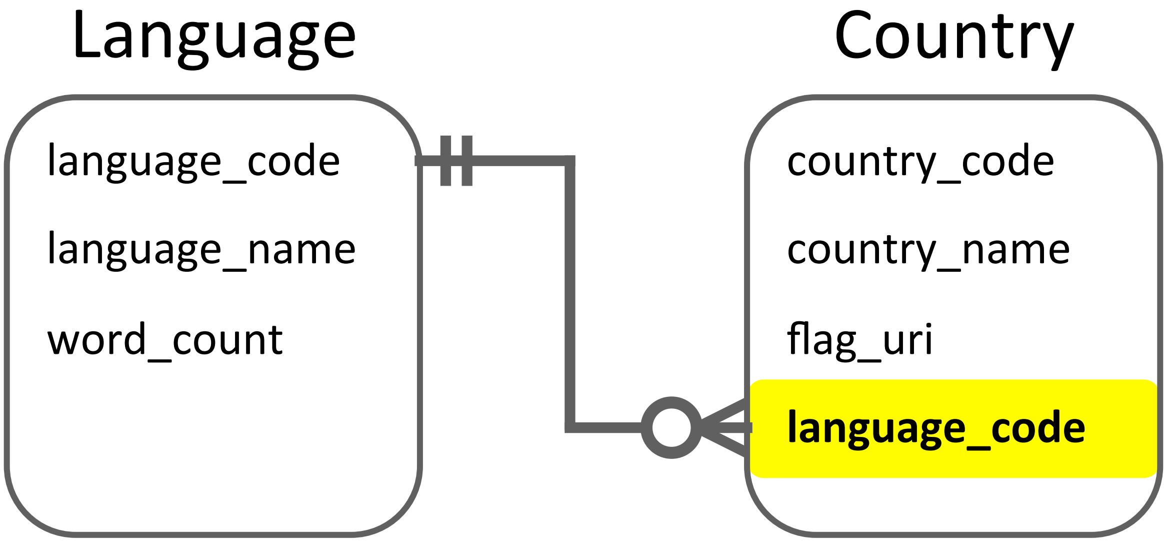
Language

- language_code
- language_name
- word_count

Country

- country_code
- country_name
- flag_uri

Need to model the relationship



Language

language_code

language_name

word_count

Country

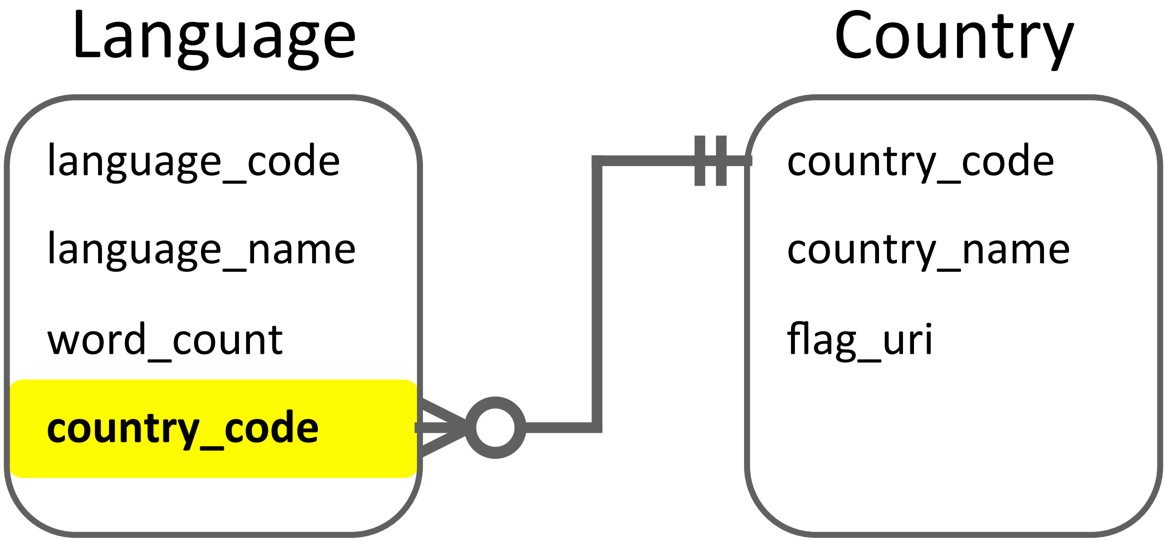
country_code

country_name

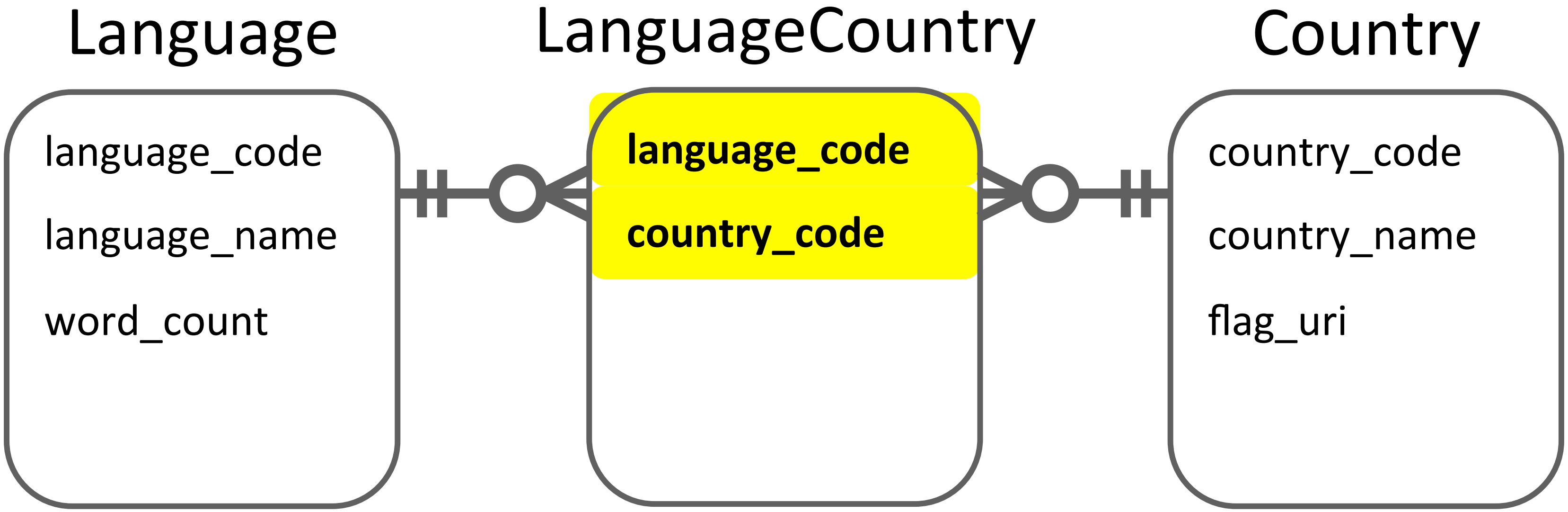
flag_uri

language_code

What if the cardinality changes?



Or we go many-to-many?



Language

language_code

language_name

word_count

LanguageCountry

language_code

country_code

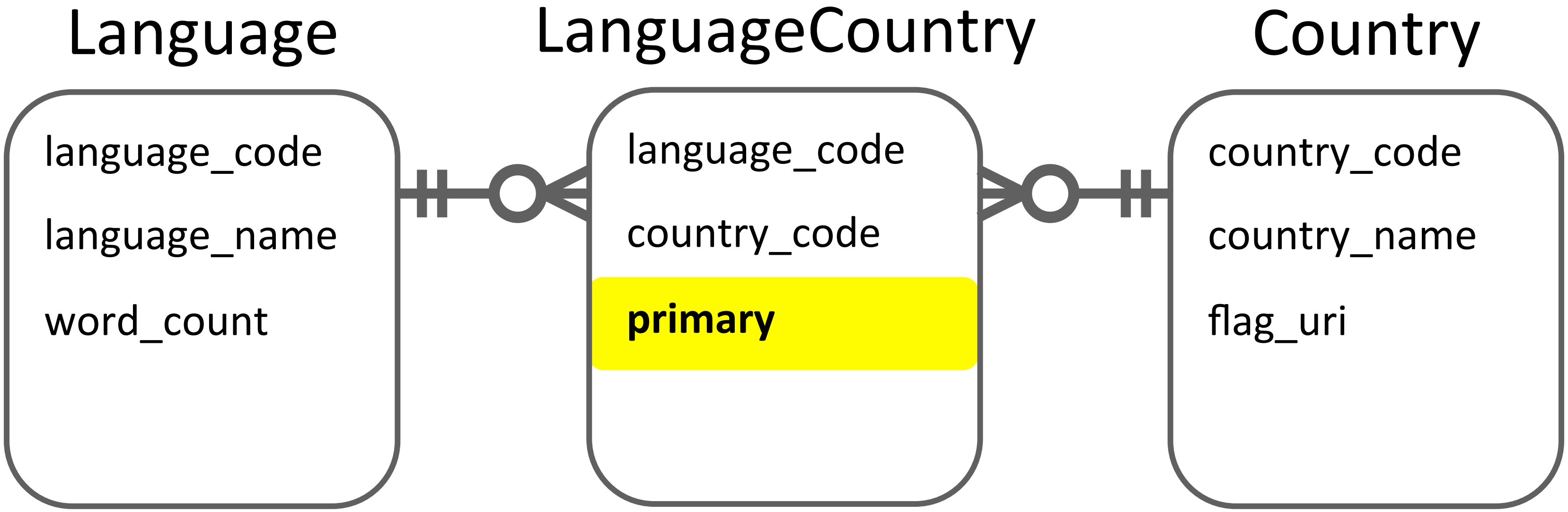
Country

country_code

country_name

flag_uri

Or we want to qualify the relationship?



Language

language_code

language_name

word_count

LanguageCountry

language_code

country_code

primary

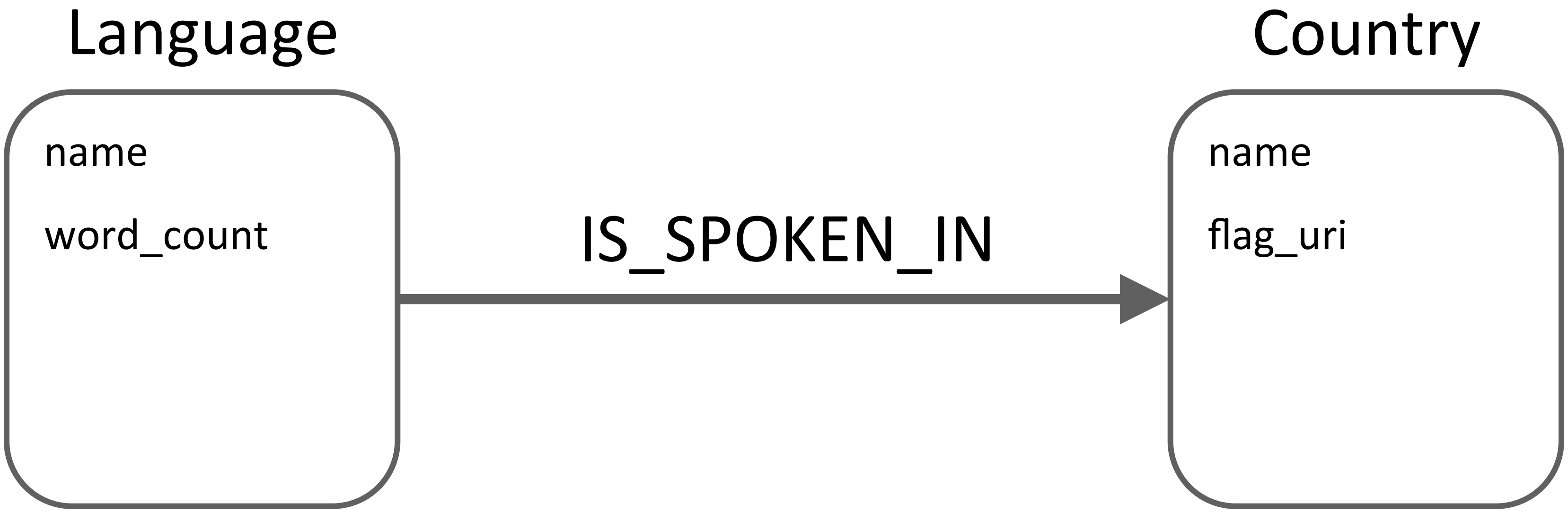
Country

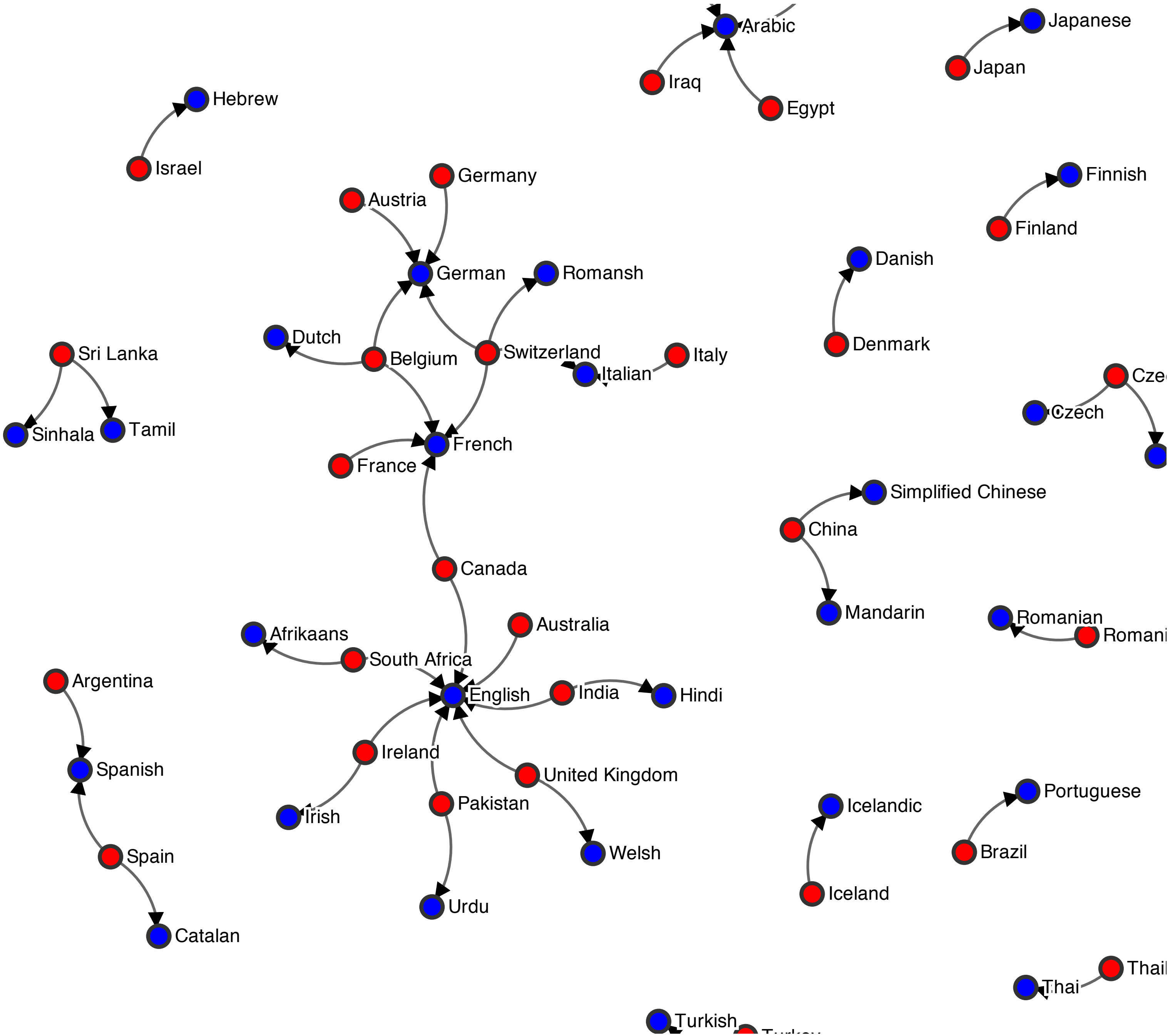
country_code

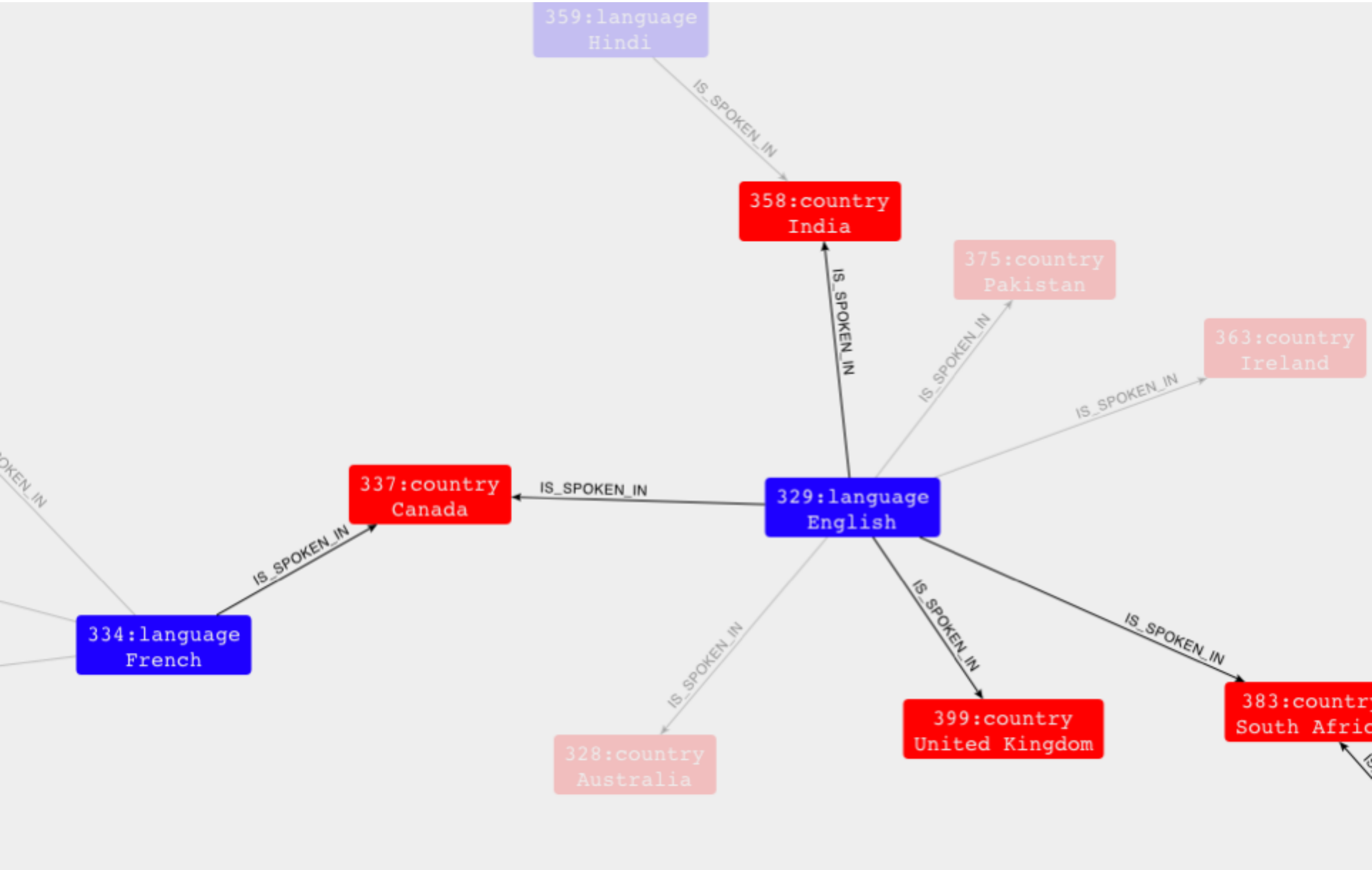
country_name

flag_uri

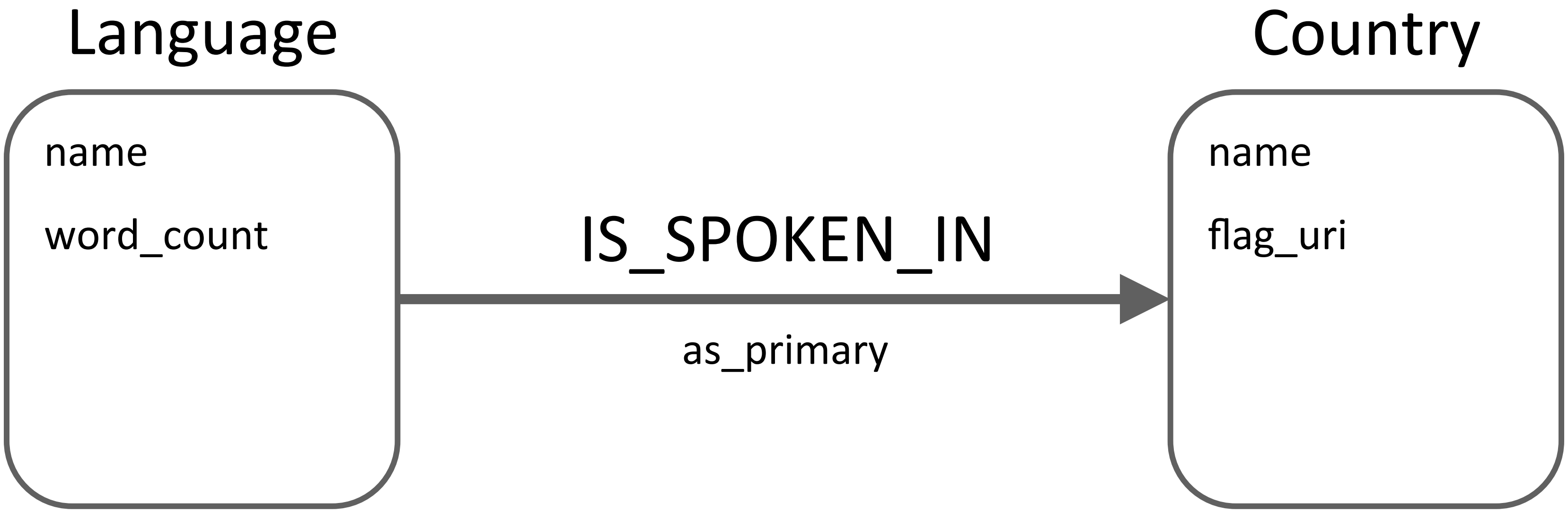
Explicit Relationship




Arabic
Japanese
Iraq
Japan
Egypt
Hebrew
Israel
Germany
Finnish
Austria
Finland
German
Romansh
Danish
Dutch
Switzerland
Italy
Belgium
Italian
Denmark
Sri Lanka
Cze
Czech
Sinhala
Tamil
French
France
Simplified Chinese
China
Canada
Mandarin
Australia
Romanian
Afrikaans
Romani
South Africa
Argentina
English
India
Hindi
Ireland
Spanish
United Kingdom
Pakistan
Icelandic
Portuguese
Irish
Welsh
Spain
Brazil
Urdu
Iceland
Catalan
Thail
Thai
Turkish


359:language
Hindi
IS_SPOKEN_IN
358:country
India
IS_SPOKEN_IN
375:country
Pakistan
IS_SPOKEN_IN
363:country
Ireland
IS_SPOKEN_IN
OKEN_IN
337:country
Canada
IS_SPOKEN_IN
329:language
English
IS_SPOKEN_IN
334:language
French
IS_SPOKEN_IN
IS_SPOKEN_IN
IS_SPOKEN_IN
383:country
South Afric
399:country
United Kingdom
328:country
Australia

Relationship Properties



Language

name

word_count

IS_SPOKEN_IN

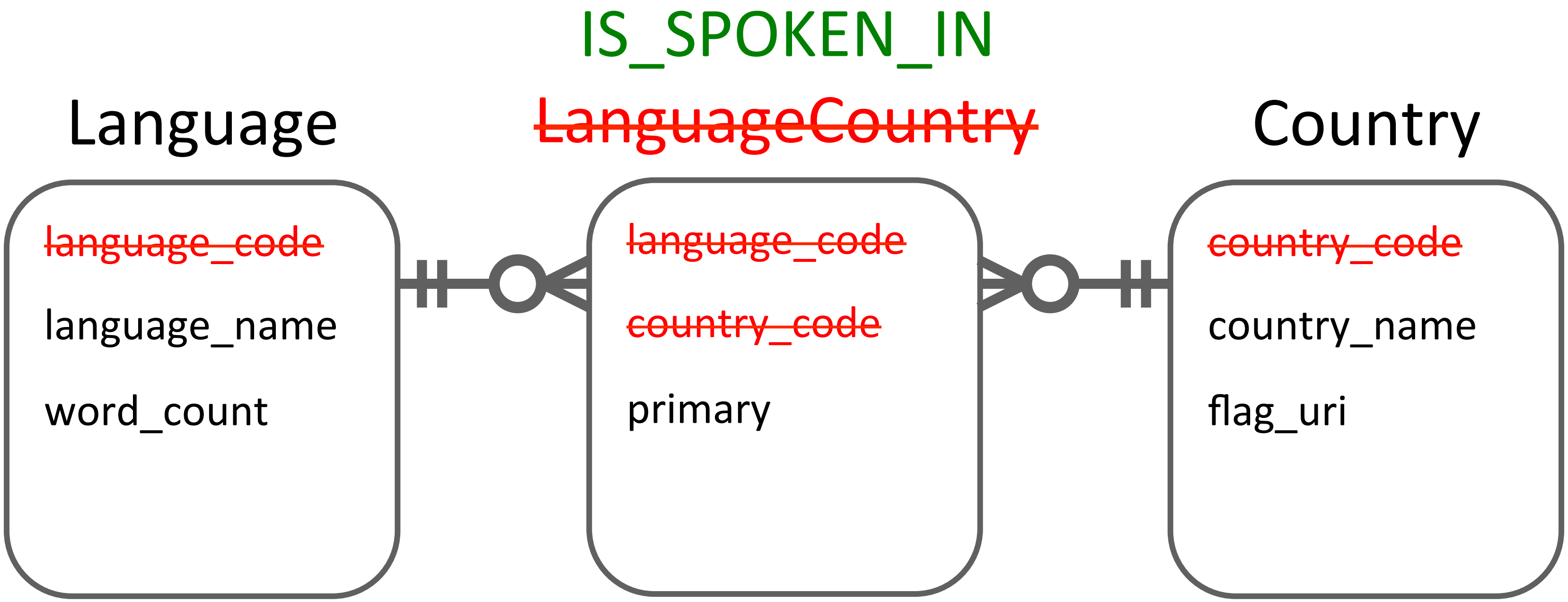
as_primary

Country

name

flag_uri

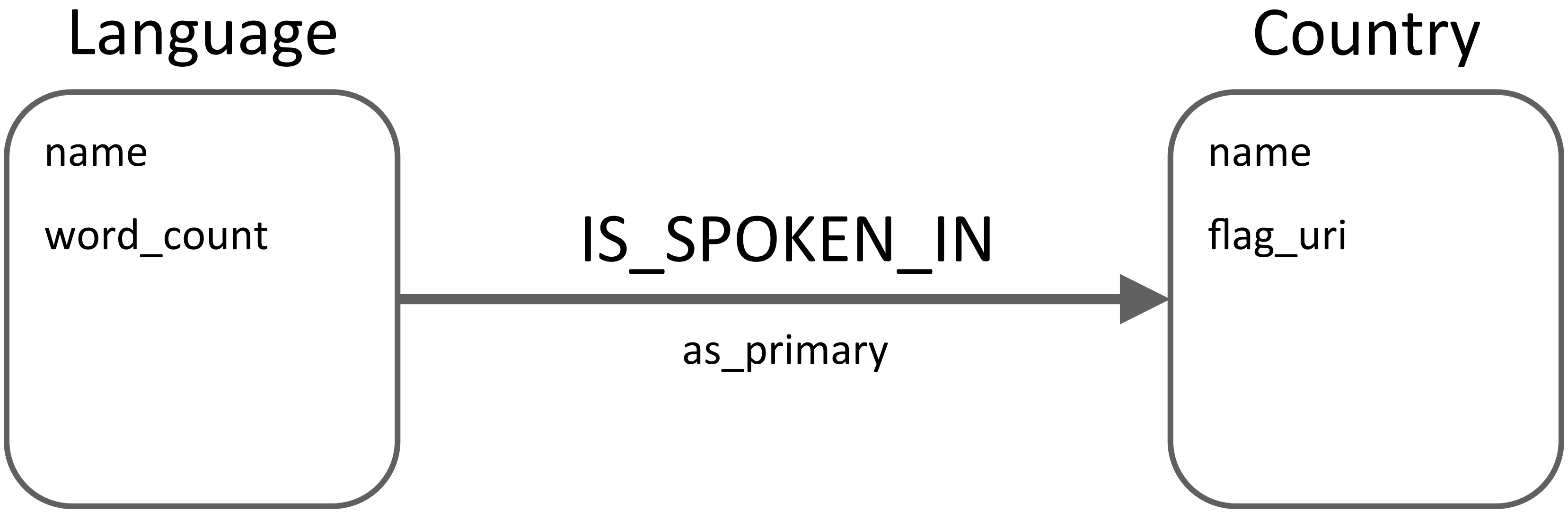
What’s different?



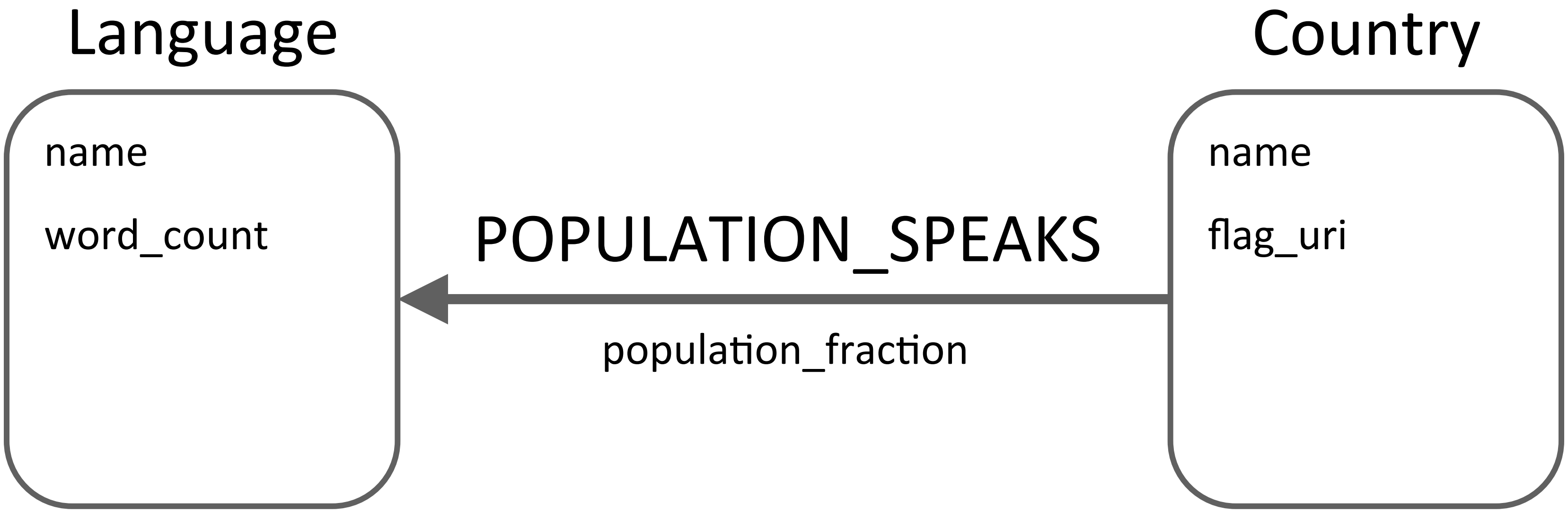
What’s different?

- Implementation of maintaining relationships is left up to the database
- Surrogate keys disappear or are unnecessary (with the exception of node id)
- Relationships get an explicit name

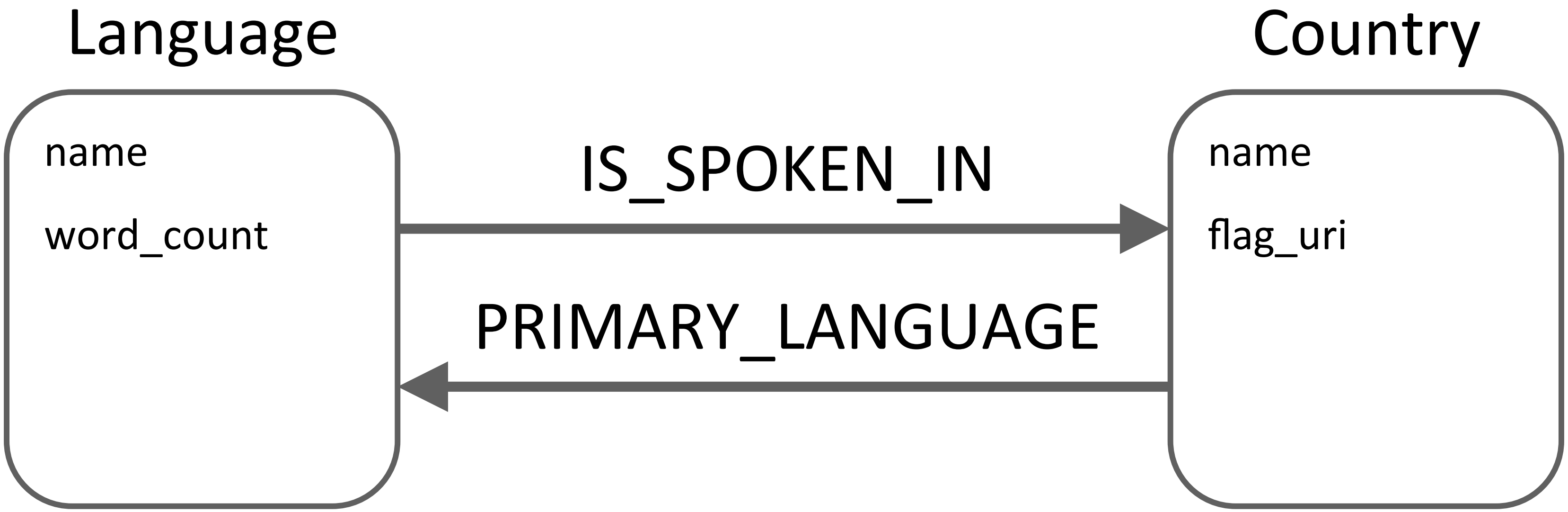
Relationship specialisation



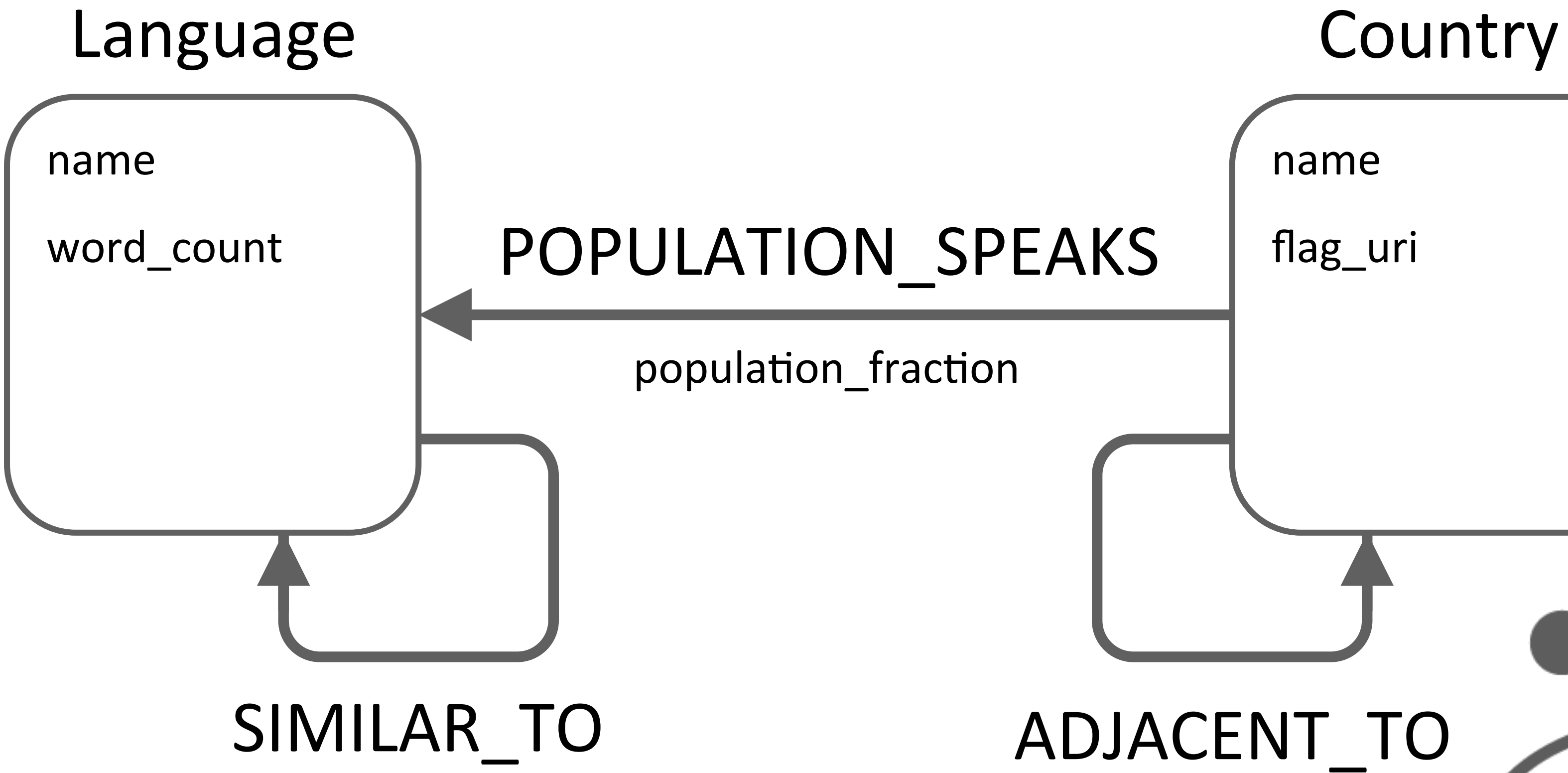
Weighted relationships



Bidirectional relationships



Keep on adding relationships



Graph Modeling Principles

1

2

3

Graph Modeling Principles

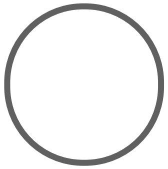
Embrace the Paradigm

2

3

Use the building blocks

- Nodes



- Relationships

RELATIONSHIP_NAME

- Properties

name: value

Anti-pattern: rich properties

name: “Canada”

languages_spoken: “[‘English’, ‘French’]”

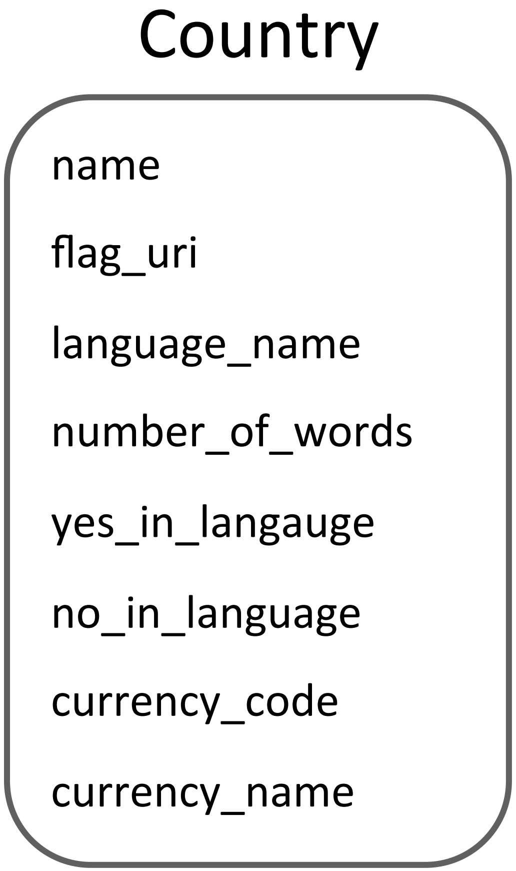
Graph Modeling Principles

Embrace the Paradigm

Nodes for Identity

3

Node represents multiple concepts



Country

name

flag_uri

language_name

number_of_words

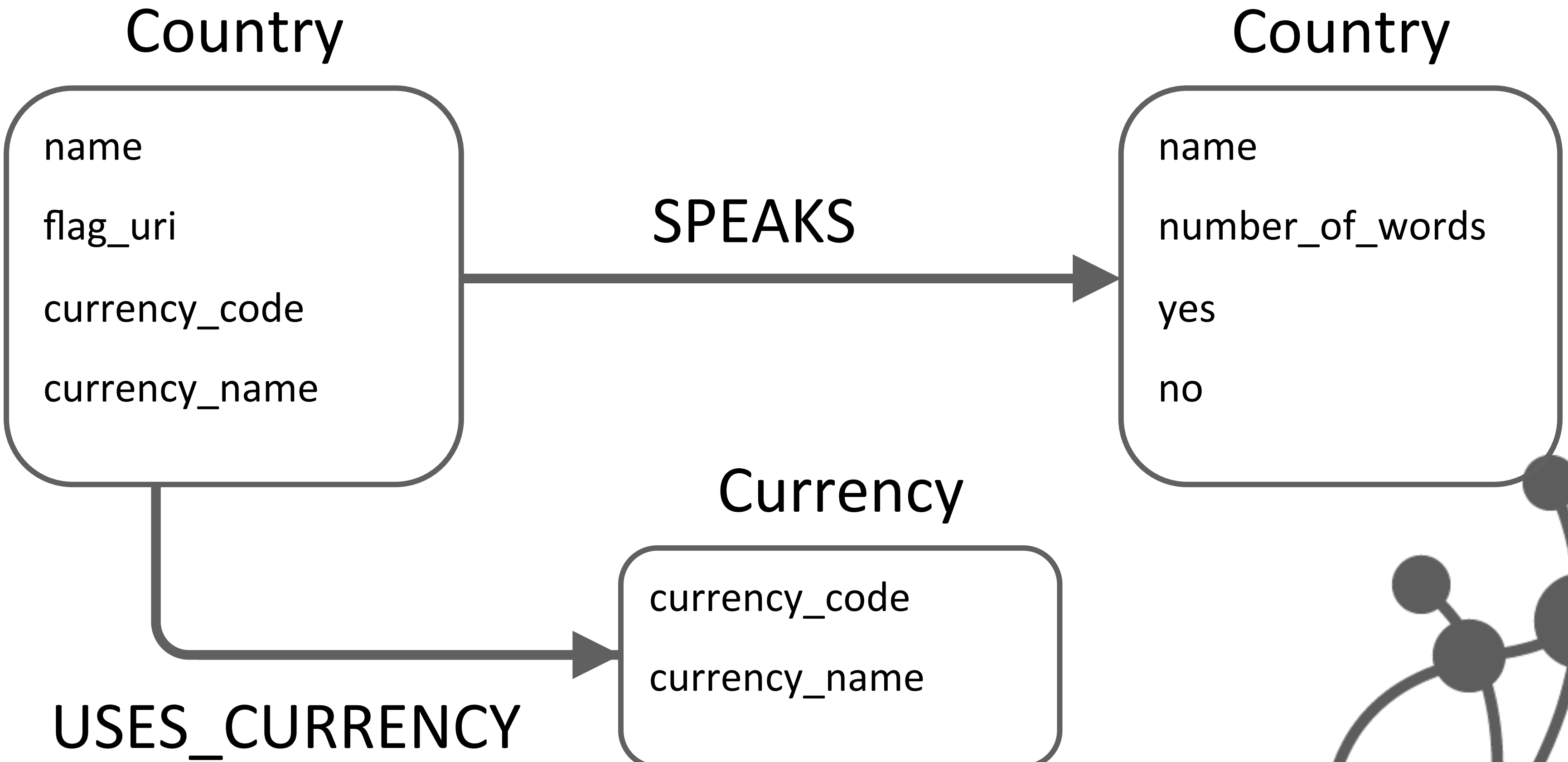
yes_in_langauge

no_in_language

currency_code

currency_name

Break out separate concepts



Country

name

flag_uri

currency_code

currency_name

SPEAKS

Country

name

number_of_words

yes

no

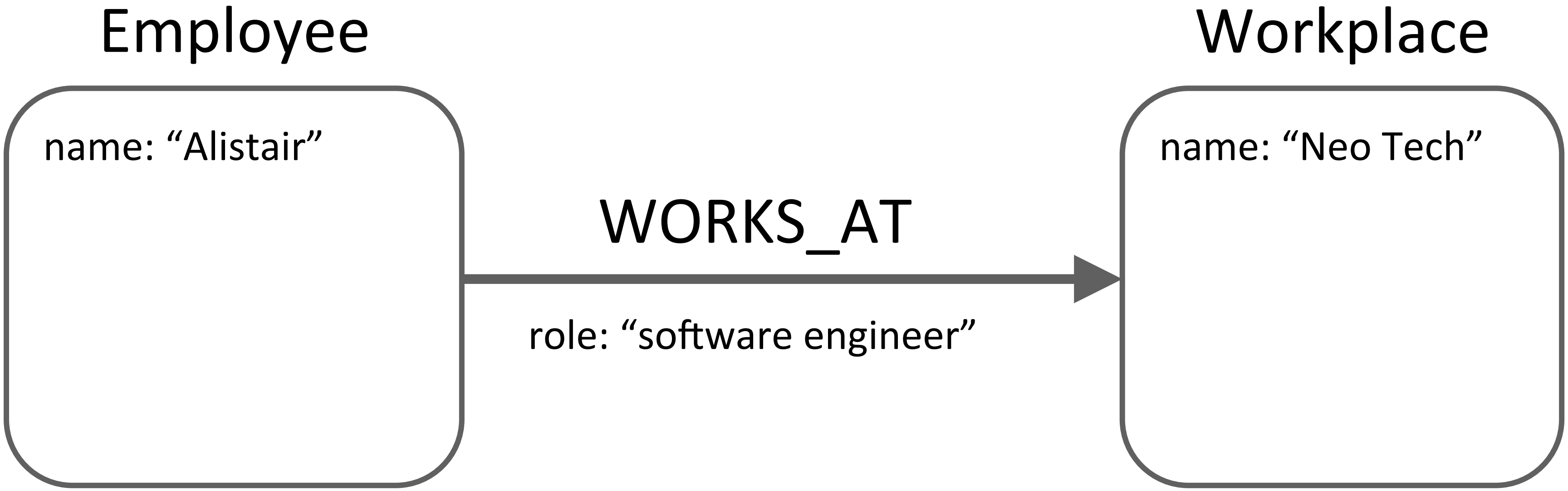
USES_CURRENCY

Currency

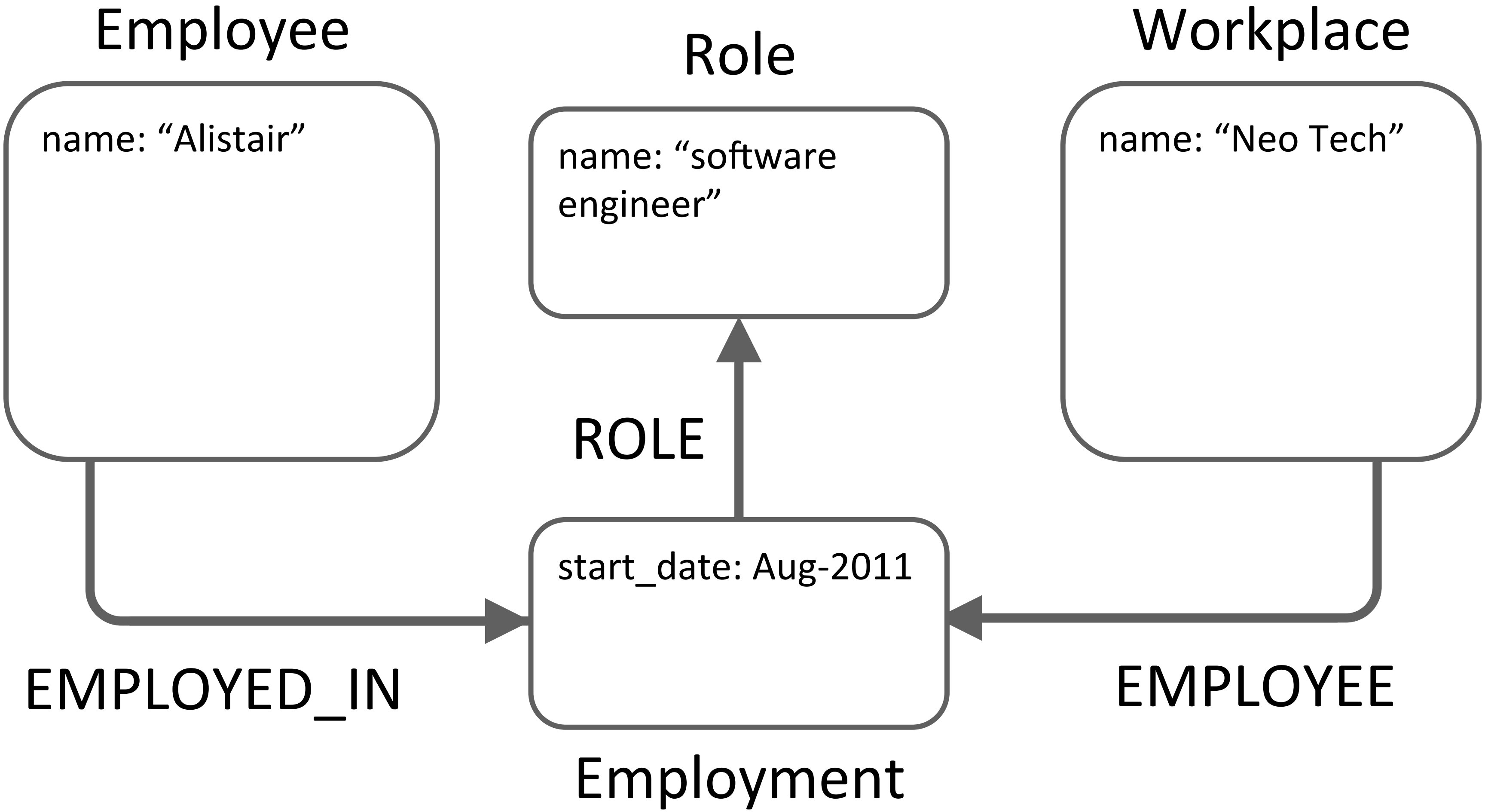
currency_code

currency_name

Property represents entity



(Reify | Nodify) connecting entities



Employee

name: “Alistair”

Role

name: “software engineer”

Workplace

name: “Neo Tech”

ROLE

start_date: Aug-2011

EMPLOYED_IN

EMPLOYEE

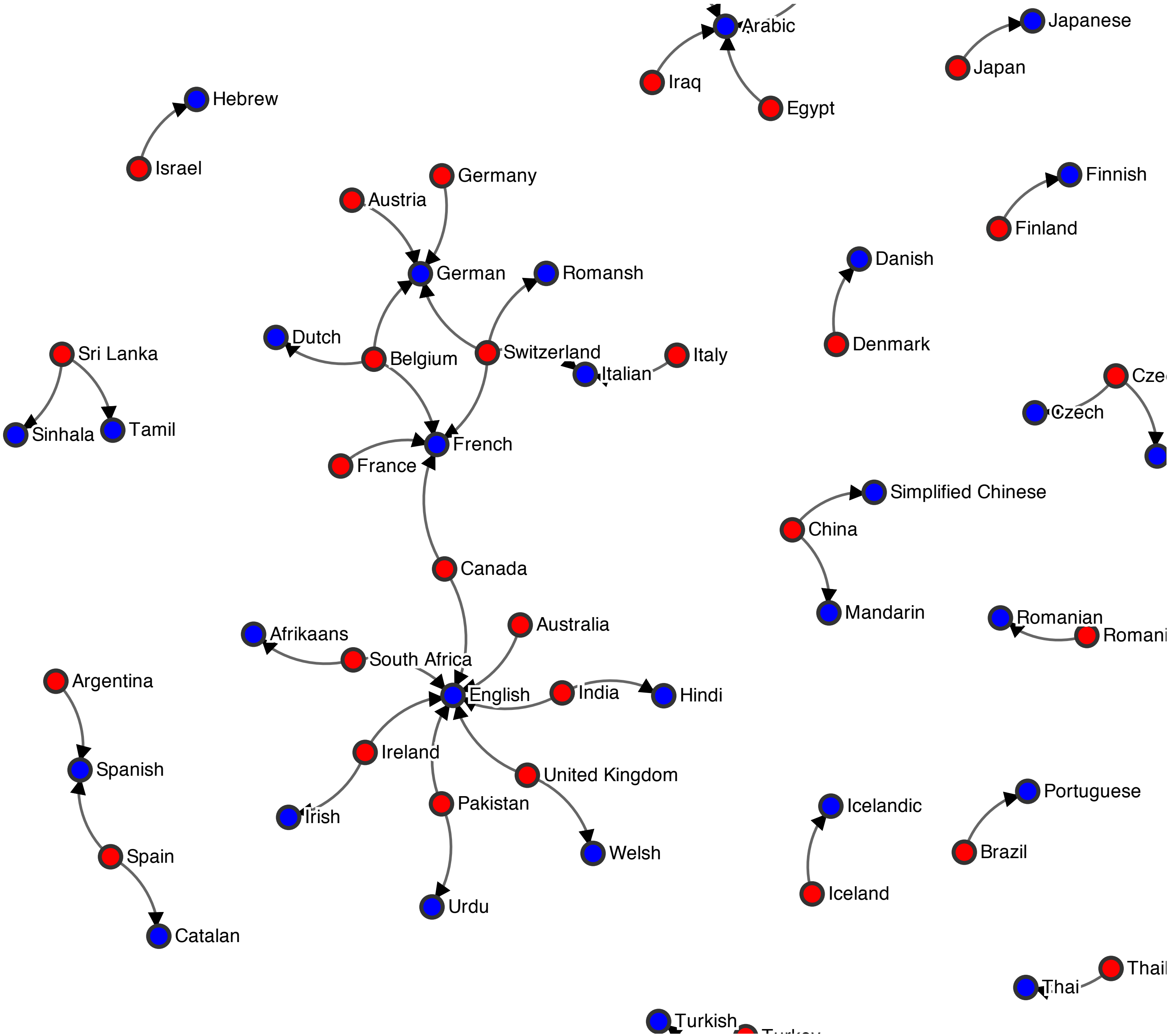
Employment

Graph Modeling Principles

Embrace the Paradigm

Nodes for Identity

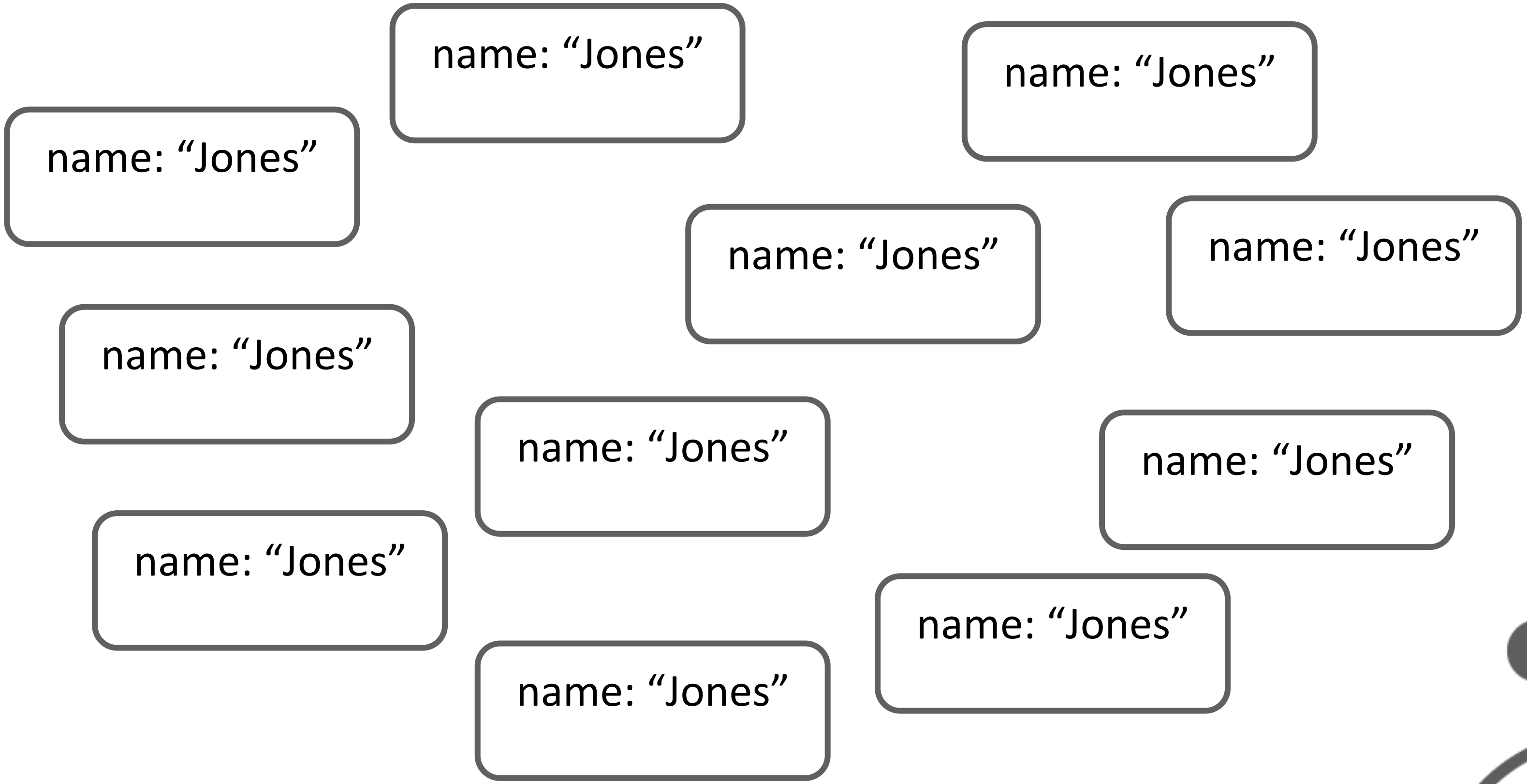
Relationships for Access


Arabic
Japanese
Iraq
Japan
Egypt
Hebrew
Israel
Germany
Finnish
Austria
Finland
German
Romansh
Danish
Dutch
Sri Lanka
Switzerland
Italy
Belgium
Denmark
Italian
Cze
Czech
Sinhala
Tamil
French
France
Simplified Chinese
China
Canada
Mandarin
Australia
Romanian
Romani
Afrikaans
South Africa
Argentina
English
India
Hindi
Ireland
Spanish
United Kingdom
Pakistan
Portuguese
Icelandic
Irish
Brazil
Spain
Welsh
Iceland
Urdu
Catalan
Thail
Thai
Turkish

Relationships for querying

- Relationships should the primary means to access nodes in the database
- Traversing relationships is cheap – that’s the whole design goal of a graph database
- Use indexes only to find starting nodes for a query

Anti-pattern: unconnected graph



name: “Jones”

name: “Jones”

name: “Jones”

name: “Jones”

name: “Jones”

name: “Jones”

name: “Jones”

name: “Jones”

name: “Jones”

name: “Jones”

name: “Jones”

Challenge: Property or Relationship?

- Can every property be replaced by a relationship?
- Should every entities with the same property values be connected?

Object Mapping

- Similar to how you would map objects to a relational database, using an ORM such as Hibernate
- Generally simpler and easier to reason about
- Examples
  - Java: Spring Data Graph
  - Ruby: Active Model

Aside: why map?

- Do you use mapping because you are scared of SQL?
- Following DDD, could you write your repositories directly against the graph API?

Graph Modeling Principles

Embrace the Paradigm

Nodes for Identity

Relationships for Access

Alistair Jones

alistair.jones@neotechnology.com

@apcj