http://goo.gl/io/0cuT0

Google I/O 11

Hash: #io2011 #AppEngine
Feedback http://goo.gl/LGT8c

Querying Freebase: Get More From MQL

Jamie Taylor
May 2011

Google I/O 11

Hash: #io2011 #AppEngine
Feedback http://goo.gl/LGT8c

Hash: #io2011 #AppEngine

Feedback http://goo.gl/LGT8c

The Freebase Landscape

Looking at Links

Property driven programming

Doing more with less

Hash: #io2011 #AppEngine

Feedback http://goo.gl/LGT8c


Freebase™

Freebase:
The landscape

PRIDE

AND

PREJUDICE:

A NOVEL.

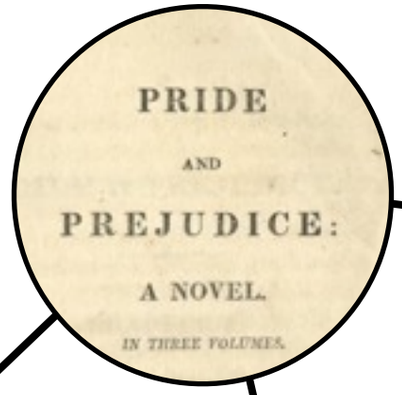
IN THREE VOLUMES.

influenced

born_in

contained_by

wrote



character_in

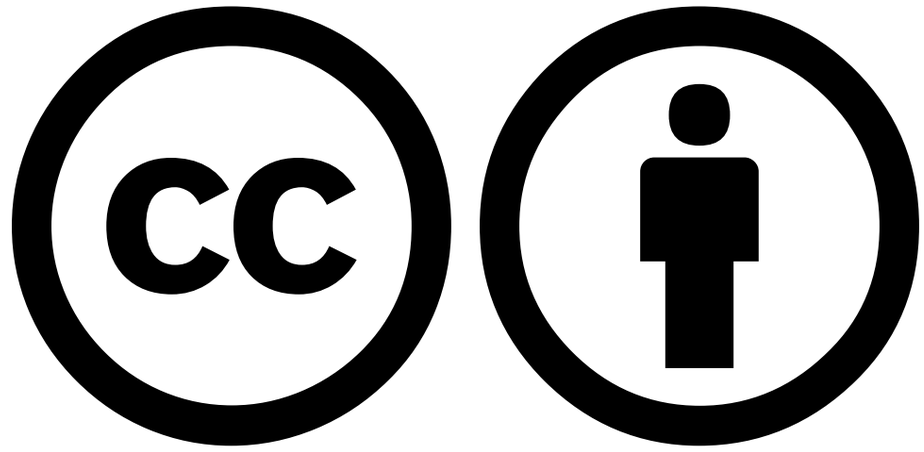
portrayed

adaptation

starred_in

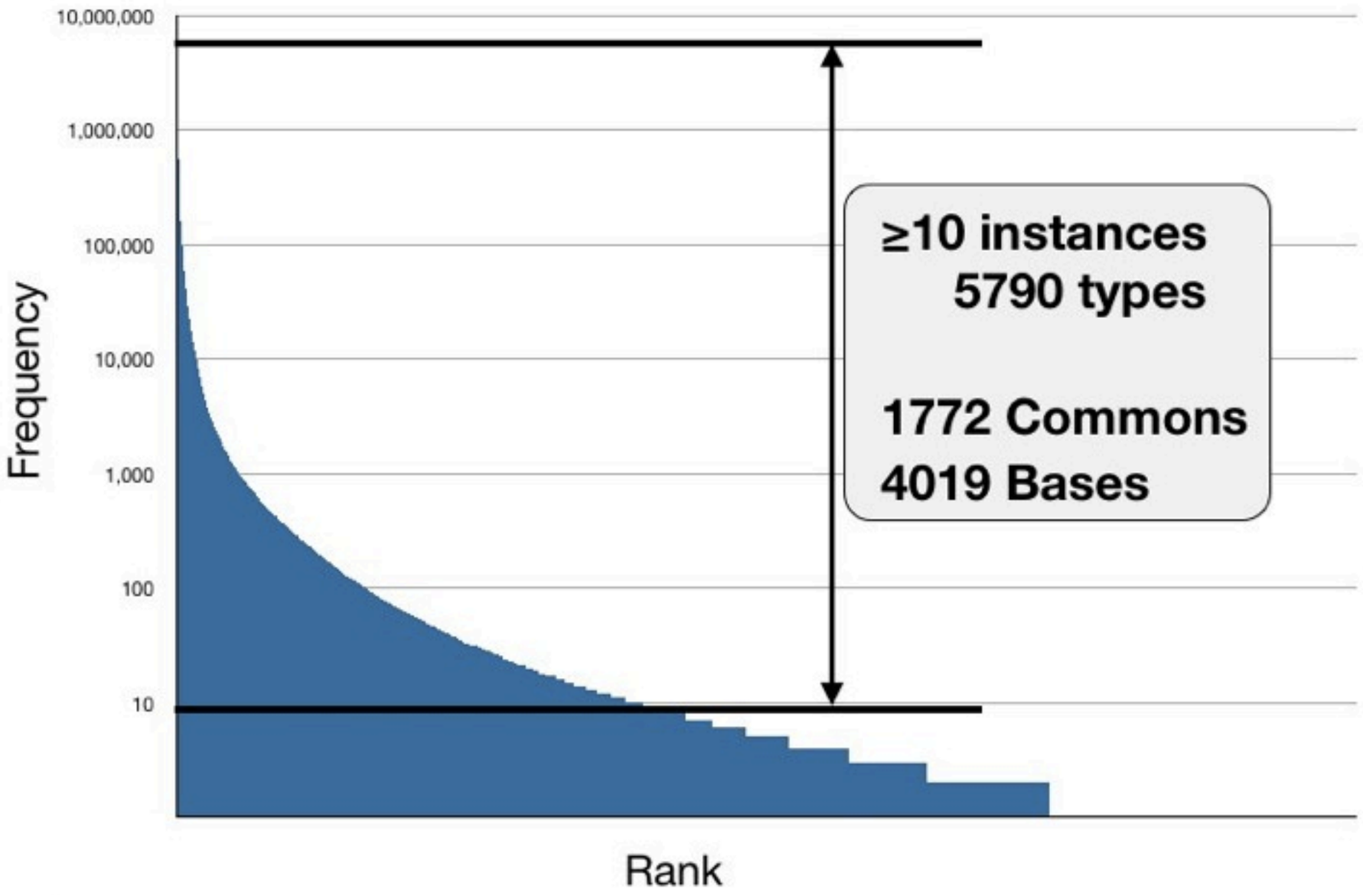
Knowledge you can use

- 22 Million Topics
- 400 Million Connections



Knowledge you can use

- 22 Million Topics
- 400 Million Connections



Topics: represent “things” in the world

Topics: represent “things” in the world

Topics: represent “things” in the world

Topics: represent “things” in the world

/en/jane_austen

/en/steventon

/en/fitzwilliam_darcy

Topics: represent “things” in the world

/en/jane_austen

/en/steventon

/en/fitzwilliam_darcy

- Objects have id’s
- Id’s are not names

Properties: relationships between objects

influenced

influenced_by

- Bidirectional
- Unique names in each direction

Properties: create meaning

links give meaning to objects

"Jane Austen"

name

place_of_birth

date_of_birth

"1775"

Google I/O 11

MQL:
Ask the Graph

Google I/O 11

MQL: Ask the Graph

place_of_birth

MQL: Ask the Graph

```
{ "id":"/en/jane_austen",
  "/people/person/place_of_birth":null }
```

place_of_birth

MQL: Ask the Graph

```
{ "id":"/en/jane_austen",
  "/people/person/place_of_birth":null }
```

```
{ "id":"/en/jane_austen",
   "/people/person/place_of_birth": "Steventon" }
```

place_of_birth

MQL: Ask the Graph

place_of_birth

MQL: Ask the Graph

```
{ "id":"/en/jane_austen",
  "/people/person/place_of_birth":{ } }
```

place_of_birth

MQL: Ask the Graph

```
{ "id":"/en/jane_austen",
  "/people/person/place_of_birth":{ } }
```

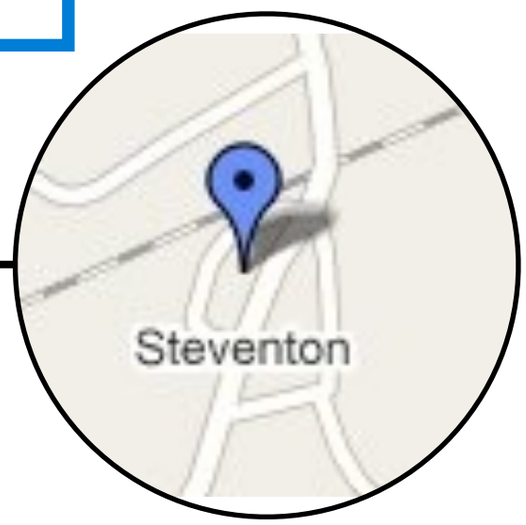
place_of_birth

MQL: Ask the Graph

```
{ "id":"/en/jane_austen",
  "/people/person/place_of_birth":{ } }
```

```
{ "id": "/en/jane_austen",
  "/people/person/place_of_birth": {
      "id":   "/en/steventon",
      "name": "Steventon",
      "type": [
        "/common/topic",
       "/location/location",
       "/location/citytown",
      ..... ]} }
```

place_of_birth



MQL: Ask the Graph

place_of_birth

MQL: Ask the Graph

```
{ "id":"/en/jane_austen",
  "/people/person/place_of_birth":{ "*":null} }
```

place_of_birth

MQL: Ask the Graph

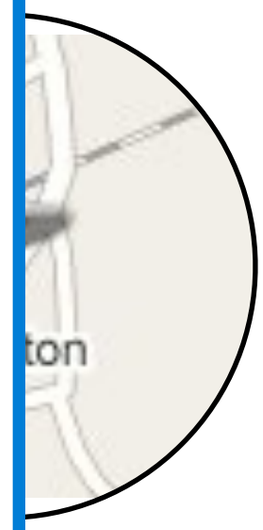
```
{ "id":"/en/jane_austen",
  "/people/person/place_of_birth":{ "*":null} }
```

place_of_birth

MQL: As

```
{ "id":"/e
  "/people
```

```
[{ "/people/person/place_of_birth": {
      "area":          null,
      "containedby": [ "United Kingdom",
                       "Hampshire" ],
      "contains":      [ ],
      "coterminous_with": [ ],
      "creator":       "/user/metaweb",
      "geolocation":   null,
      "geometry":      [ ],
      "guid": "#9202a8c04000641f80000000006ea9ed",
      "id": "/en/steventon",
      "key": [ "Steventon$002C_Hampshire",
               "867121",
               "3864309" ],
    "mid": [ "/m/06ybhf" ],
    "name": "Steventon",
    "near":  [ ],
    "type": [ "/common/topic",
              "/location/location",
              "/location/citytown" ],
     "usbg_name":     [ ]
   },
   "id": "/en/jane_austen"
}]
```



MQL: Ask the Graph

place_of_birth

MQL: Ask the Graph

```
{ "id":"/en/jane_austen",
  "/people/person/place_of_birth":{ "id":null} }
```

place_of_birth

MQL: Ask the Graph

```
{ "id":"/en/jane_austen",
  "/people/person/place_of_birth":{ "id":null} }
```

place_of_birth

MQL: Ask the Graph

```
{ "id":"/en/jane_austen",
  "/people/person/place_of_birth":{ "id":null} }
```

```
{ "id": "/en/jane_austen",
  "/people/person/place_of_birth": {
      "id":   "/en/steventon"
} }
```

place_of_birth

MQL: Ask the Graph

place_of_birth

MQL: Ask the Graph

```
[{ "id":null,
   "/people/person/place_of_birth":{ "id":"/en/steventon"} }]
```

place_of_birth

MQL: Ask the Graph

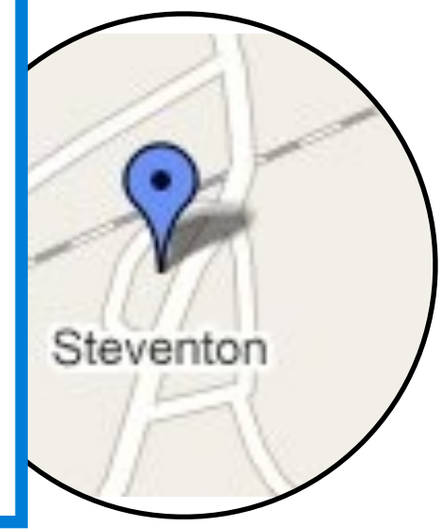
```
[{ "id":null,
   "/people/person/place_of_birth":{ "id":"/en/steventon"} }]
```

place_of_birth

MQL: Ask the Graph

```
[{ "id":null,
   "/people/person/place_of_birth":{ "id":"/en/steventon"} }]
```

```
[{
    "/people/person/place_of_birth": {
      "id": "/en/steventon"
    },
    "id": "/en/jane_austen"
  },{
    "/people/person/place_of_birth": {
      "id": "/en/steventon"
    },
    "id": "/en/francis_austen"
  },{
    "/people/person/place_of_birth": {
      "id": "/en/steventon"
    },
    "id": "/en/cassandra_austen"
}]
```



MQL: Ask the Graph

place_of_birth

MQL: Ask the Graph

```
[{ "id": "/en/steventon",
"!/people/person/place_of_birth":[{ "id":null }] }]
```

place_of_birth

MQL: Ask the Graph

```
[{ "id": "/en/steventon",
"!/people/person/place_of_birth":[{ "id":null }] }]
```

place_of_birth

MQL: Ask the Graph

```
[{ "id": "/en/steventon",
"!/people/person/place_of_birth":[{ "id":null }] }]
```

```
[{,
    "id": "/en/steventon",
    "!/people/person/place_of_birth": [
      {
        "id": "/en/jane_austen"
      },
      {
        "id": "/en/cassandra_austen"
      },
      {
        "id": "/en/francis_austen"
      }
    ]
}]
```

Google I/O 11

Schema:
What to expect

Google I/O 11

Becoming Jane (the person)

How do we know Jane Austen is a person?

What makes us think she has a "place of birth?"

Everything is an object (/type/object)

Properties

What are Properties?

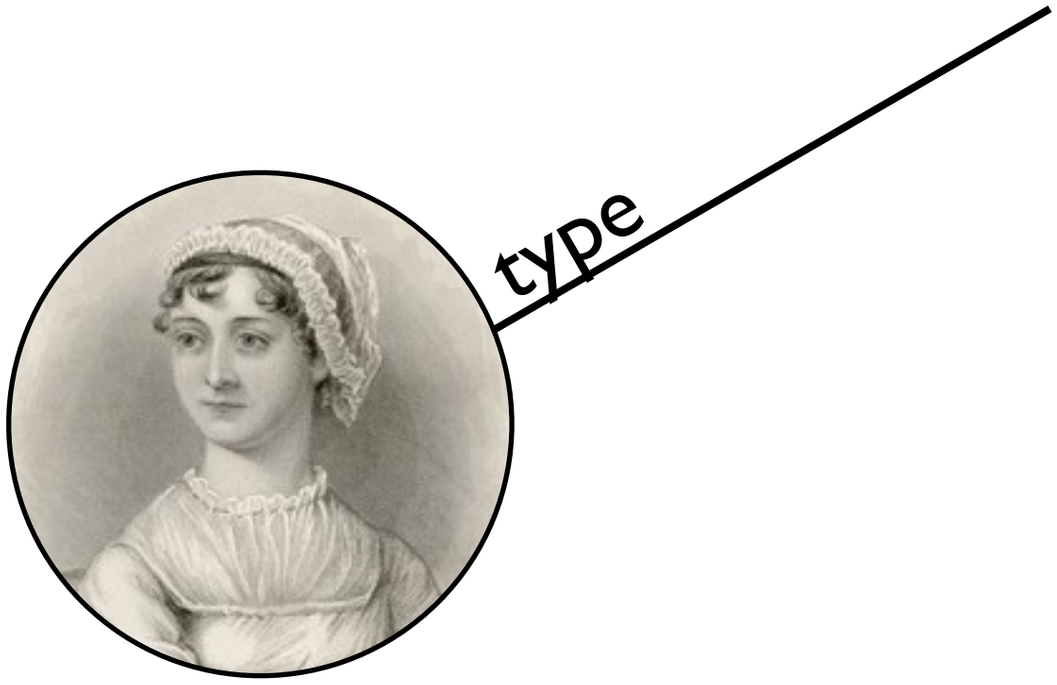
Property	ID	Expected Type	Description
id	/type/object/id	/type/id	Friendly identifier for the object. Each object has a preferred id, but may be reachable via multiple namespace paths. ***master, unique***
guid	/type/object/guid	/type/id	Unique numeric identifier of an object. Each object has only one guid. ***master, unique***
type	/type/object/type	/type/type	
name	/type/object/name	/type/text	a textual display name ***master, unique, hidden***
key	/type/object/key	/type/key	any object may be entered in a namespace ***reverse***
timestamp	/type/object/timestamp	/type/datetime	the creation time of this primitive. ***master, unique***
permission	/type/object/permission	/type/permission	the permission of this object ***master, unique***
creator	/type/object/creator	/type/user	the user who initially created this object ***master, unique***
attribution	/type/object/attribution	/type/attribution	***master, unique***
search	/type/object/search		***master***
machine id	/type/object/mid	/type/id	Machine identifier for the object. This identifier is assigned when the object is created and used to track the object throughout its lifetime. ***master***

TYPE property available to all objects

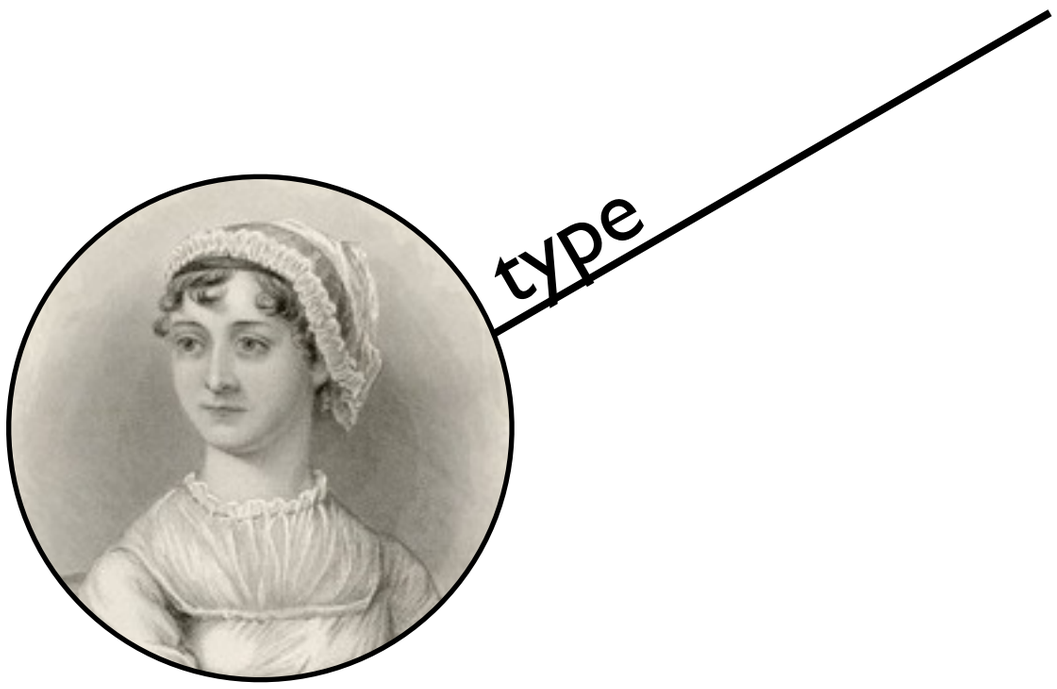
http://www.freebase.com/schema/type/object

Becoming Jane (the person)

Becoming Jane (the person)

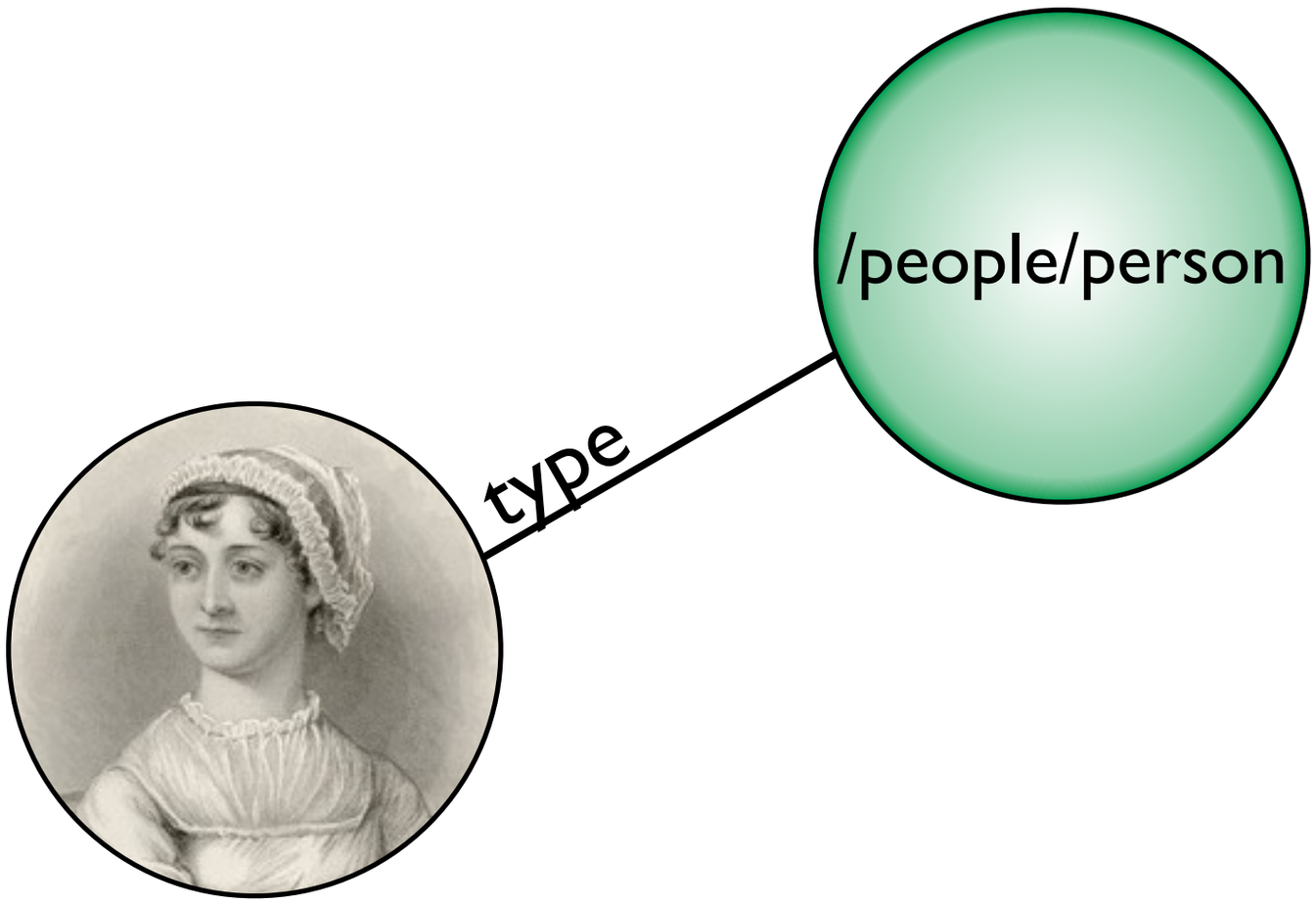


Becoming Jane (the person)




```
{ "id":"/en/jane_austen",
  "type":[{ "id":null}] }
```

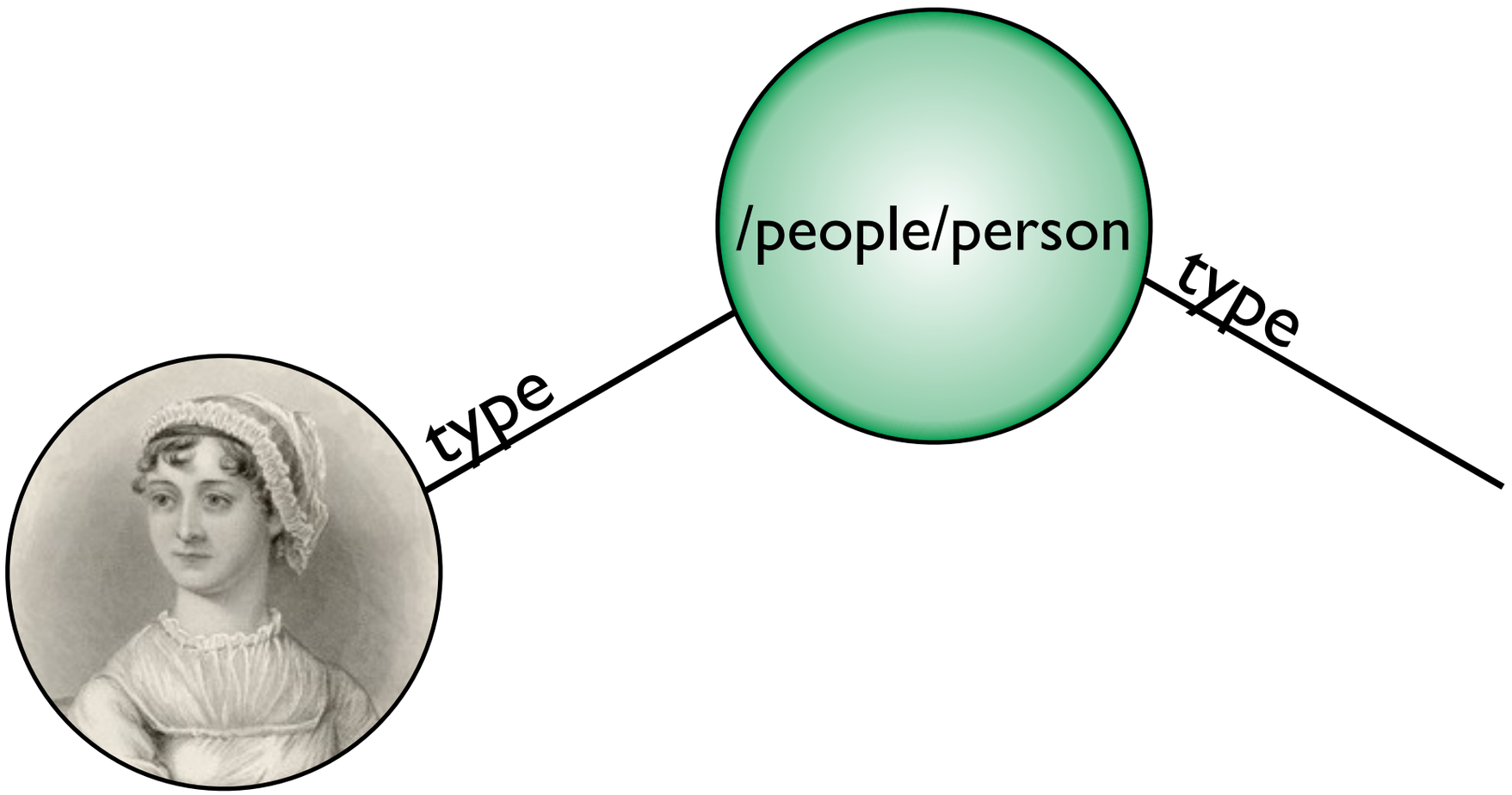
Becoming Jane (the person)




```
{ "id":"/en/jane_austen",
  "type":[{ "id":null}] }
```

Becoming Jane (the person)

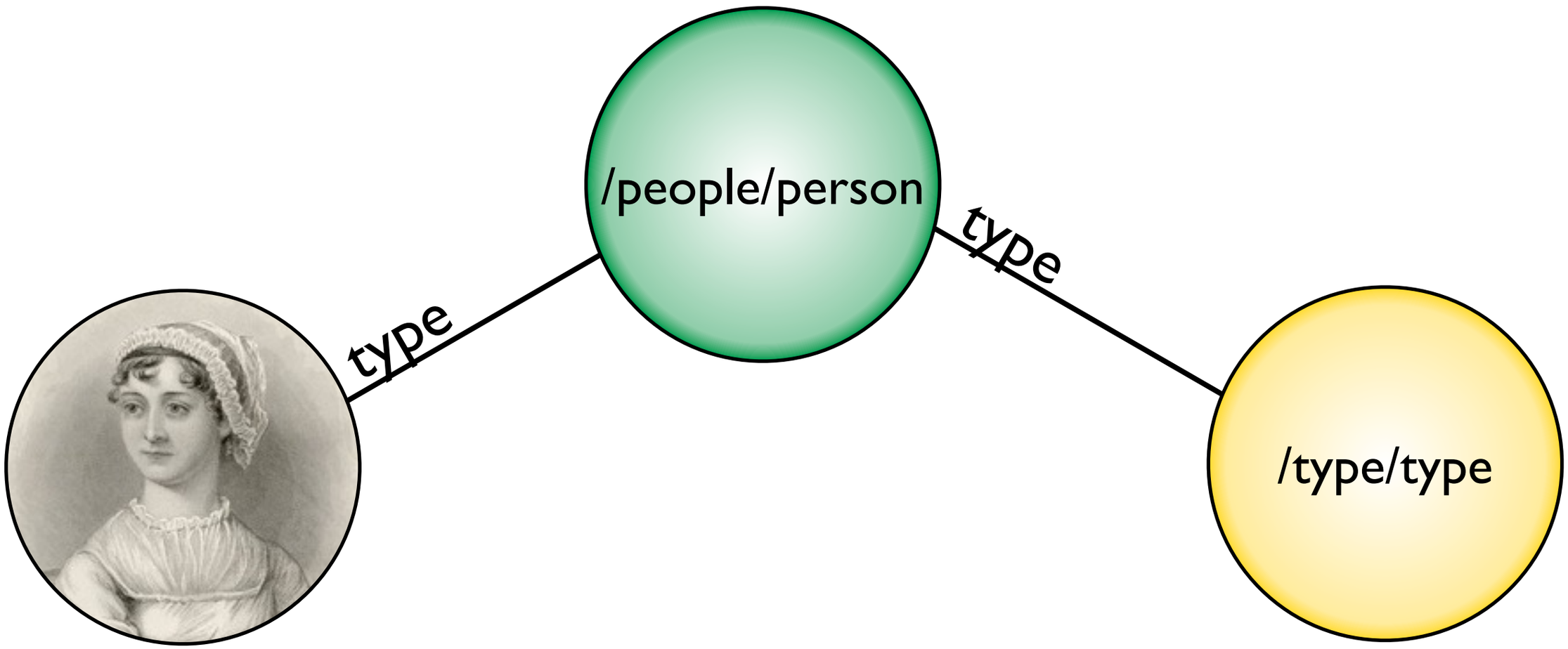
```
{ "id":"/people/person",
  "type":[{ "id":null}] }
```




```
{ "id":"/en/jane_austen",
  "type":[{ "id":null}] }
```

Becoming Jane (the person)

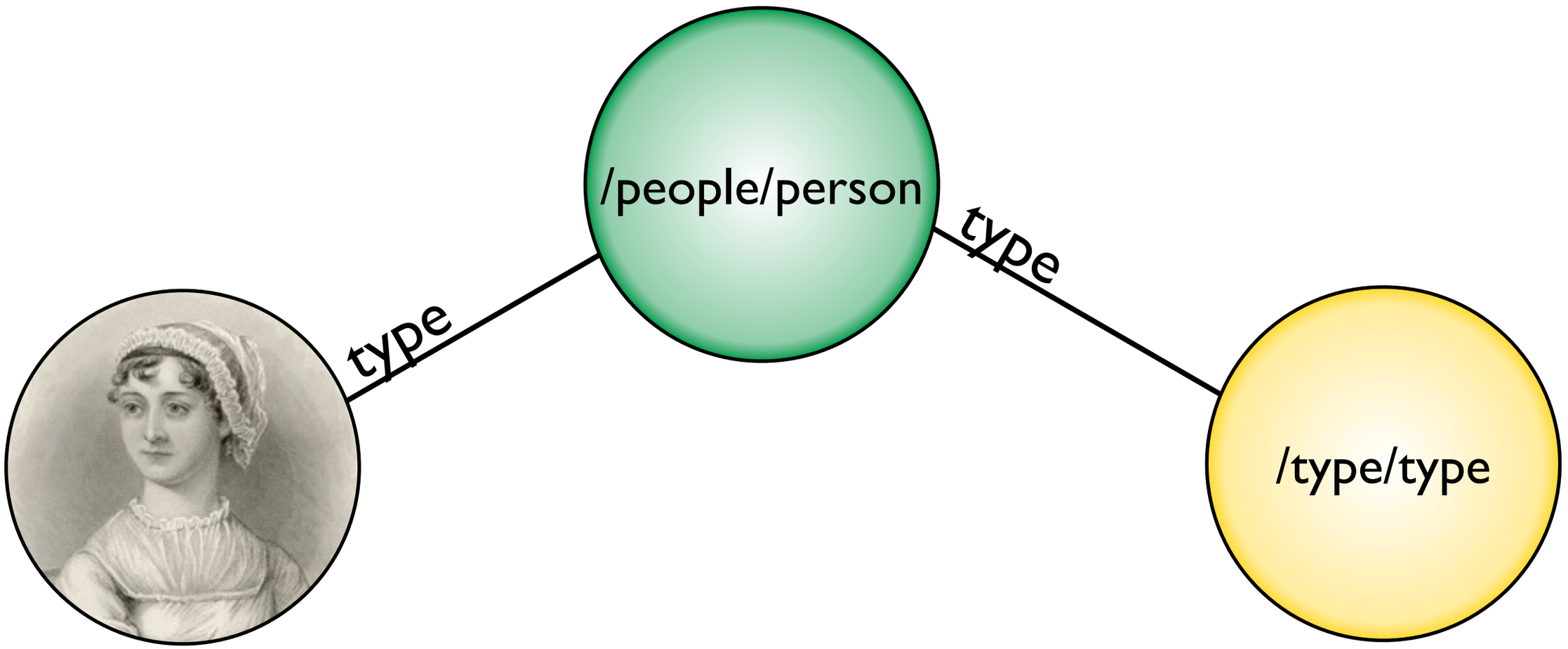
```
{ "id":"/people/person",
  "type":[{ "id":null}] }
```




```
{ "id":"/en/jane_austen",
  "type":[{ "id":null}] }
```

Becoming Jane (the person)

{ "id":"/people/person",
"type":[{ "id":null}] }

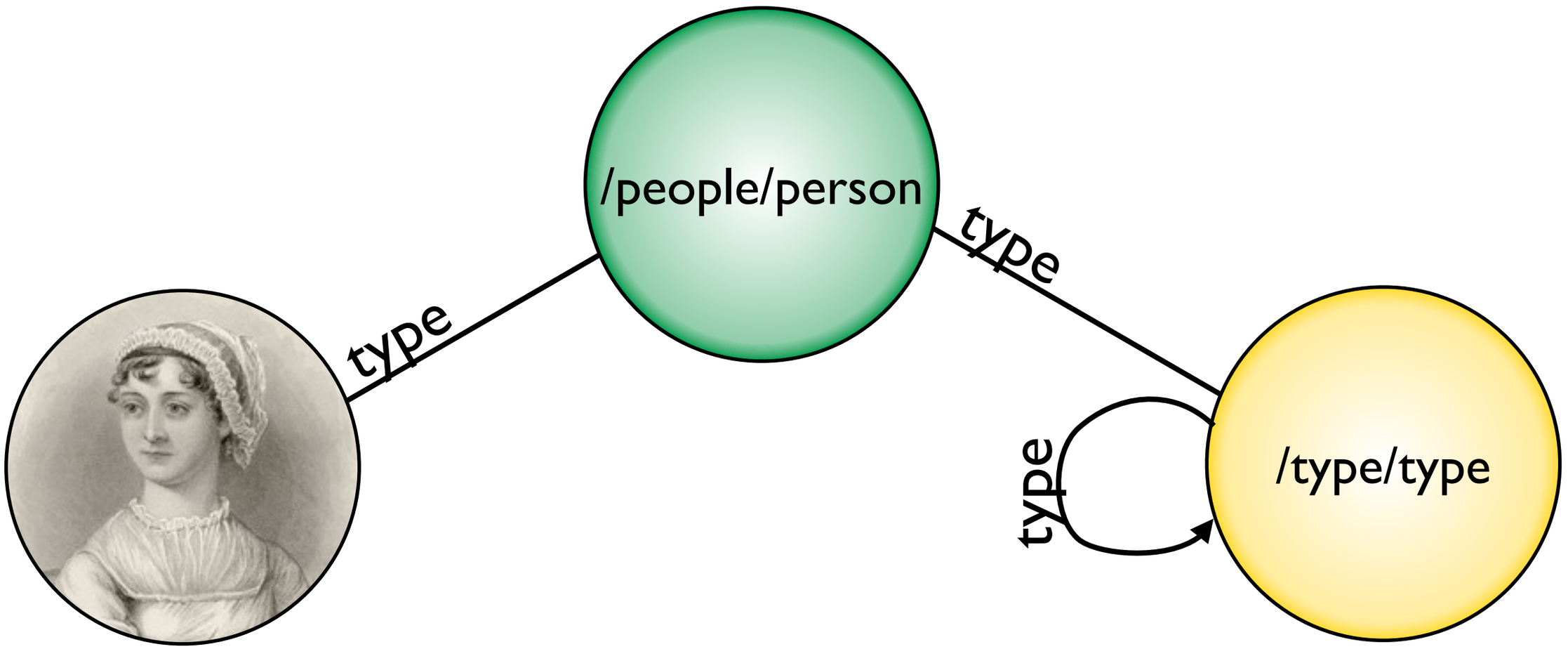


{ "id":"/en/jane_austen",
"type":[{ "id":null}] }

{ "id":"/type/type",
"type":[{ "id":null}] }

Becoming Jane (the person)

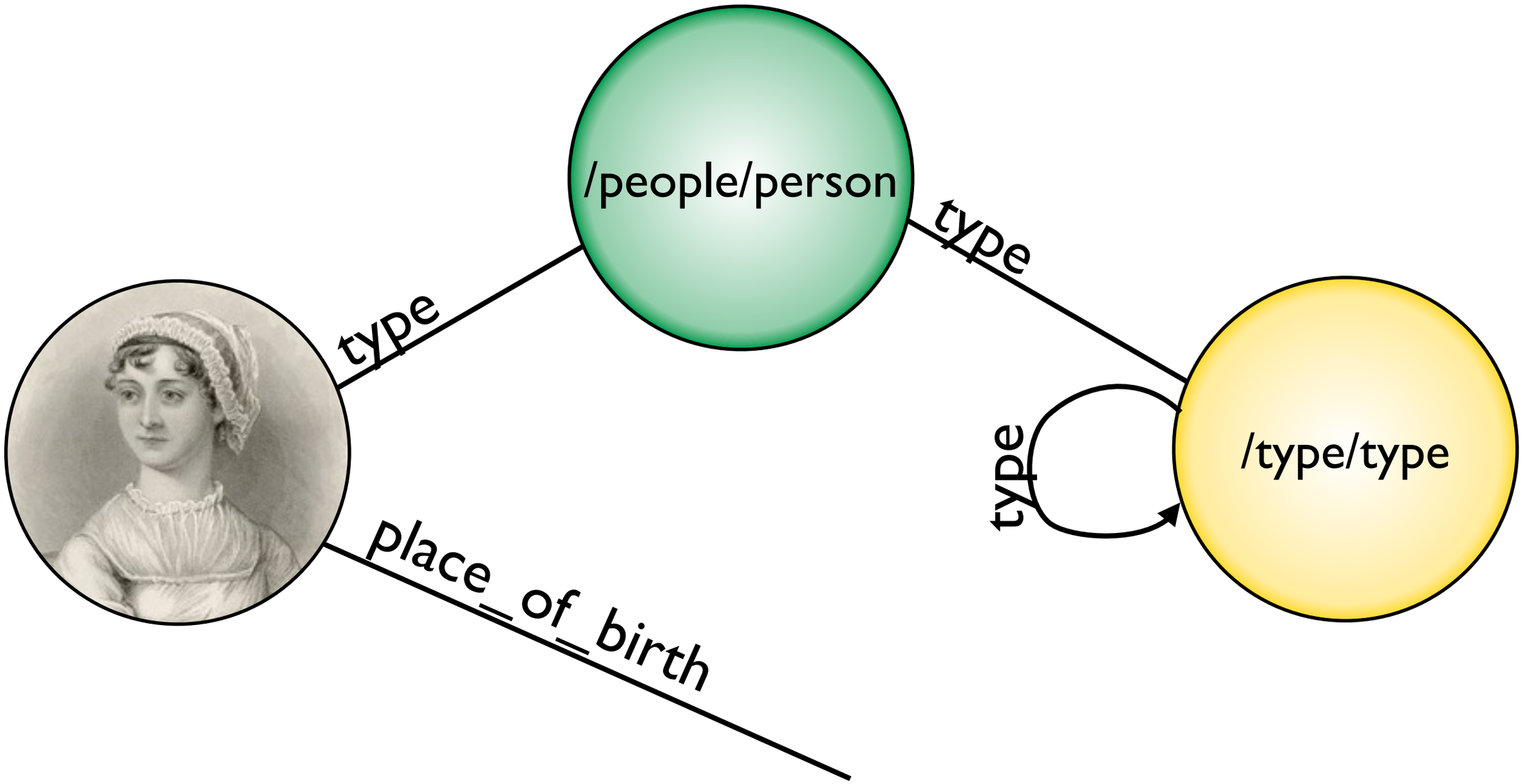
{ "id":"/people/person",
  "type":[{ "id":null}] }



{ "id":"/en/jane_austen",
  "type":[{ "id":null}] }

{ "id":"/type/type",
  "type":[{ "id":null}] }

Becoming Jane (the person)



/type/type

Type Type

Key: /type/type

Type More

Created by Freebase Staff on October 22, 2006

Table | Diagram | Browse 20,402 Instances | Build Query

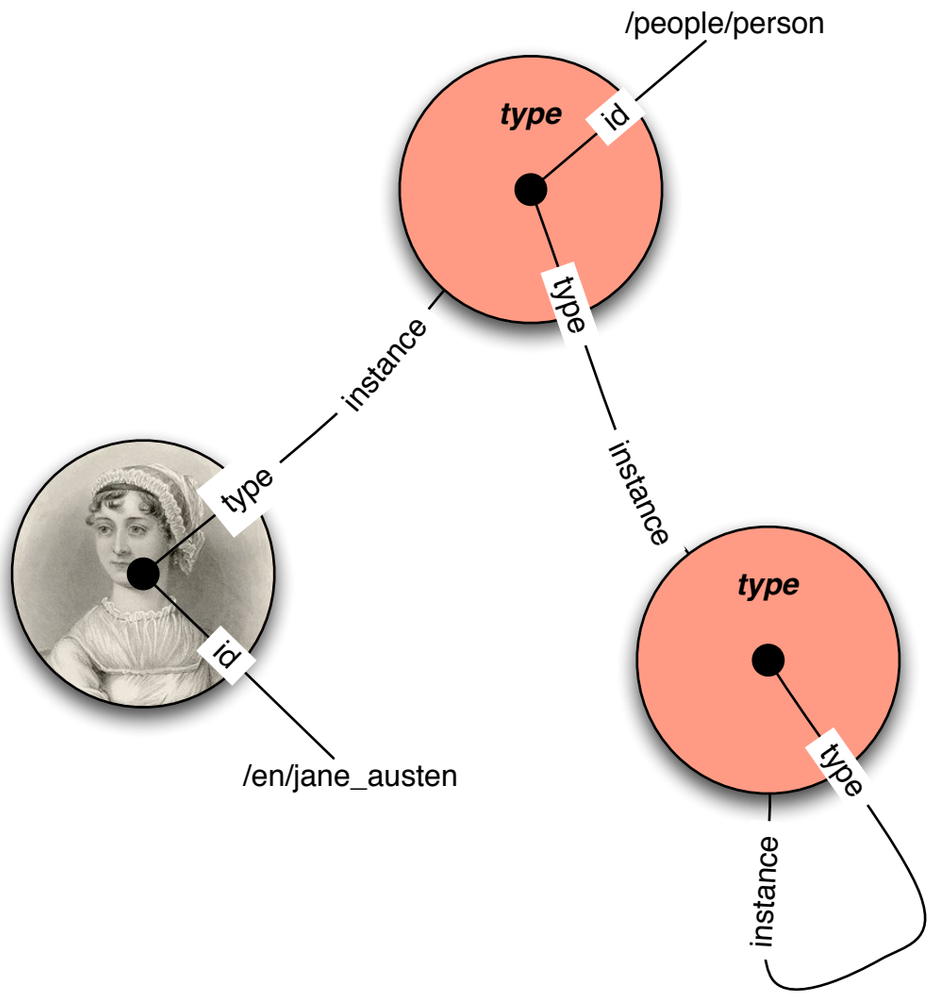
Properties

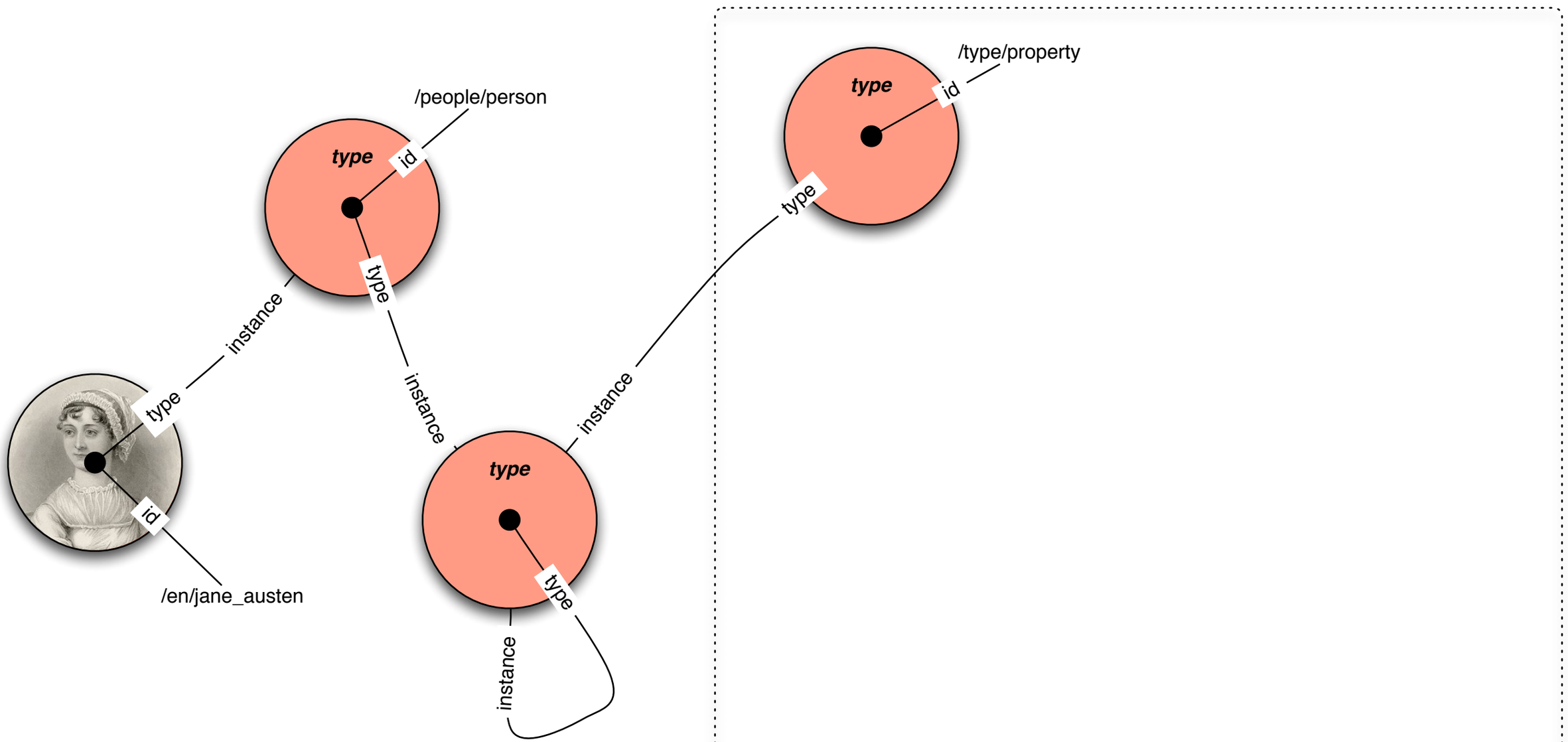
What are Properties?

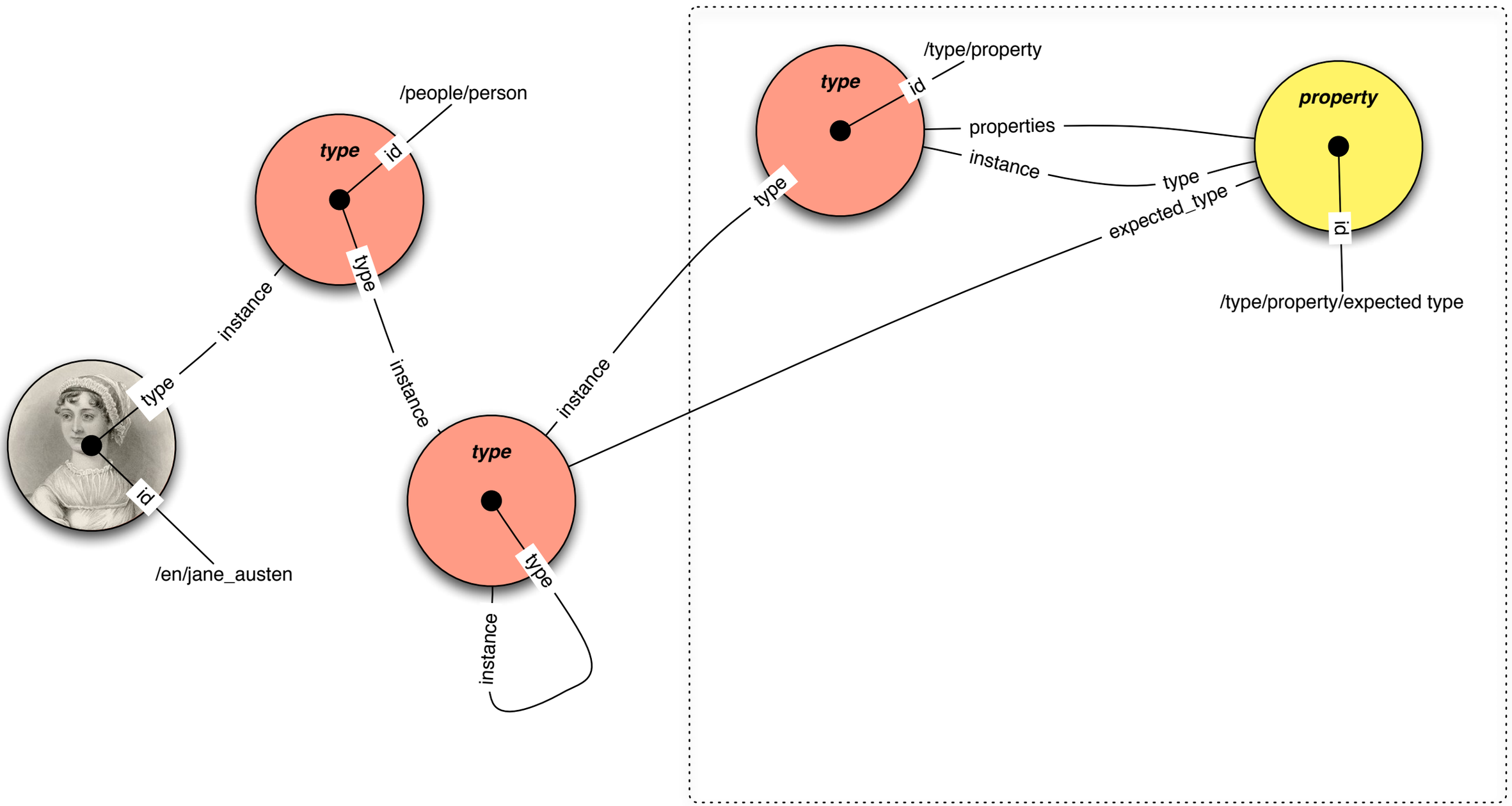
Property	ID	Expected Type	Description
properties	/type/type/properties	/type/property	
instance	/type/type/instance	/type/object	the instances of this type ***reverse***
domain	/type/type/domain	/type/domain	the domain containing this type ***master, unique***
expected_by	/type/type/expected_by	/type/property	properties which expect this type as their value ***reverse***
default_property	/type/type/default_property	/type/rawstring	this is the property used when a single value is expected out of the type. It may be name, id, or value. ***master, unique***
extends	/type/type/extends	/type/type	used internally by hi json ***master***

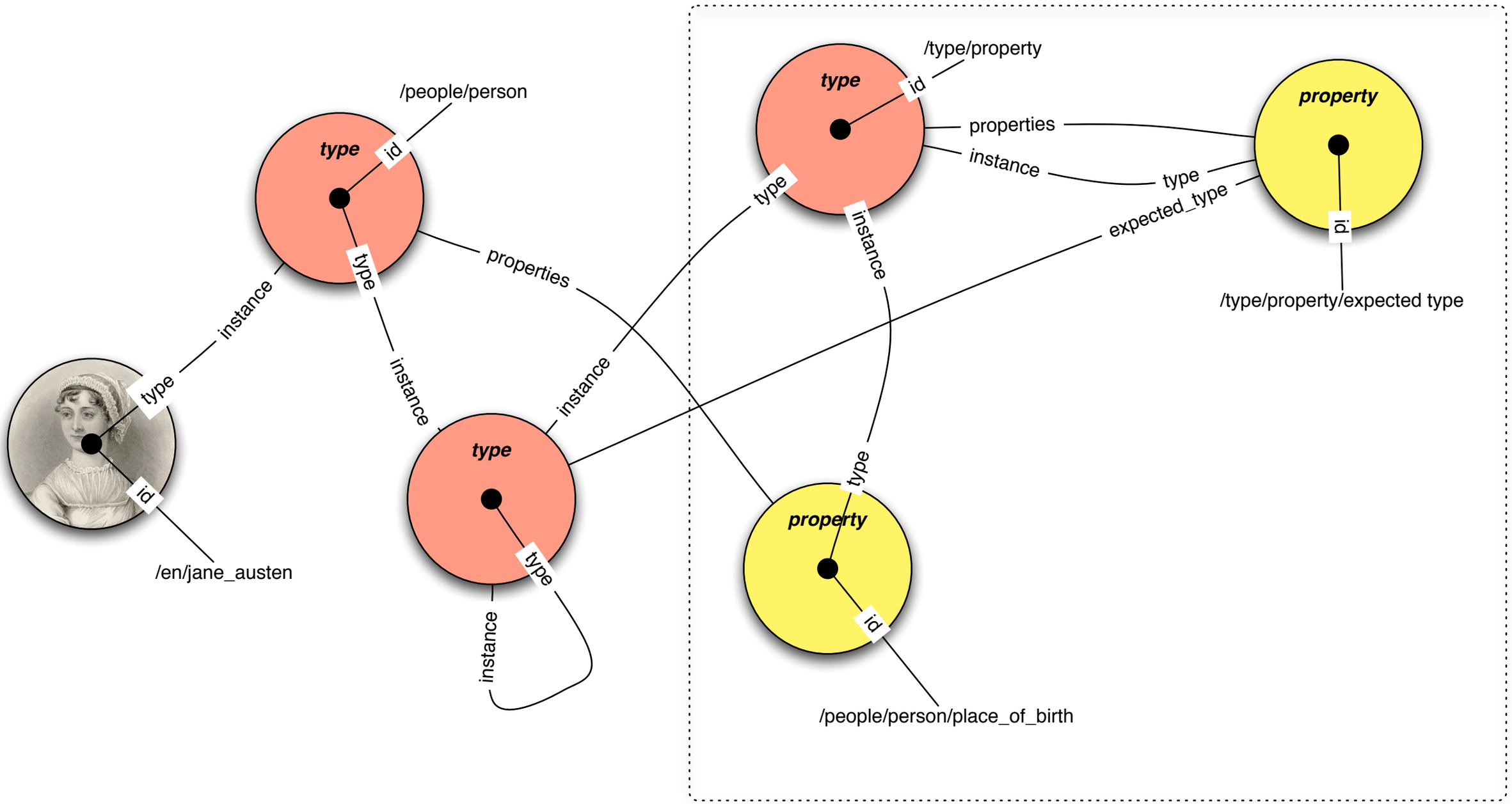
Every TYPE has PROPERTY properties that expect /type/property

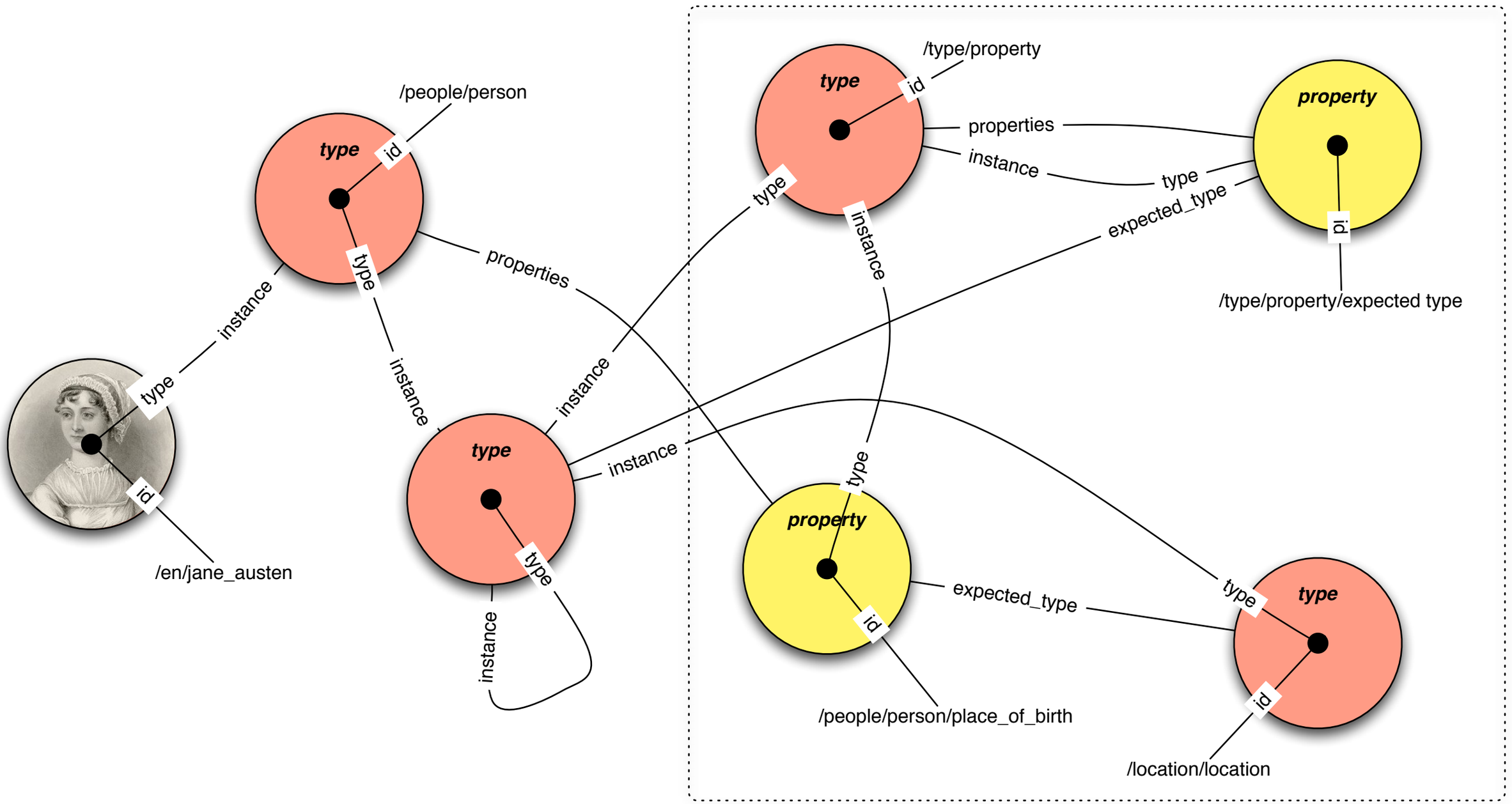
http://www.freebase.com/schema/type/type


/people/person
type
id
type
instance
type
id
/en/jane_austen
instance
type
type
instance


/people/person
type
id
type
instance
type
id
/en/jane_austen
/type/property
type
id
type
instance
instance
type
type
instance




/people/person
type
id
instance
type
properties
/en/jane_austen
type
id
instance
instance
type
type
instance
/type/property
type
id
properties
instance
type
expected_type
property
id
/type/property/expected type
type
property
id
/people/person/place_of_birth



SCHEMA described by objects IN THE GRAPH
(so we can query/explore it just like Topics using MQL)

Find all the properties that are defined for people:

```
{
  "id":    "/people/person,
  "type": "/type/type",
  "properties": []
}
```

Find all the properties that are defined for people:

```
{
  "id":   "/people/person,
  "type": "/type/type",
  "properties": []
}
```

"/people/person/date_of_birth",
"/people/person/place_of_birth",
"/people/person/gender",
"/people/person/profession",
"/people/person/places_lived",
"/people/person/languages"
....

What is the expected type for place of birth?

```
{
  "id": "/people/person/place_of_birth",
  "type": "/type/property",
  "expected_type":null
}
```

What is the expected type for place of birth?

```
{
  "id": "/people/person/place_of_birth",
  "type": "/type/property",
  "expected_type":null
}
```

"/location/location"

Find all the types that have a property that has an expected_type of /location/location:

```
[{
  "id":   null,
  "type": "/type/type",
  "properties": [{
    "id": null,
    "expected_type": {
      "id": "/location/location"
    }
  } ]
} ]
```

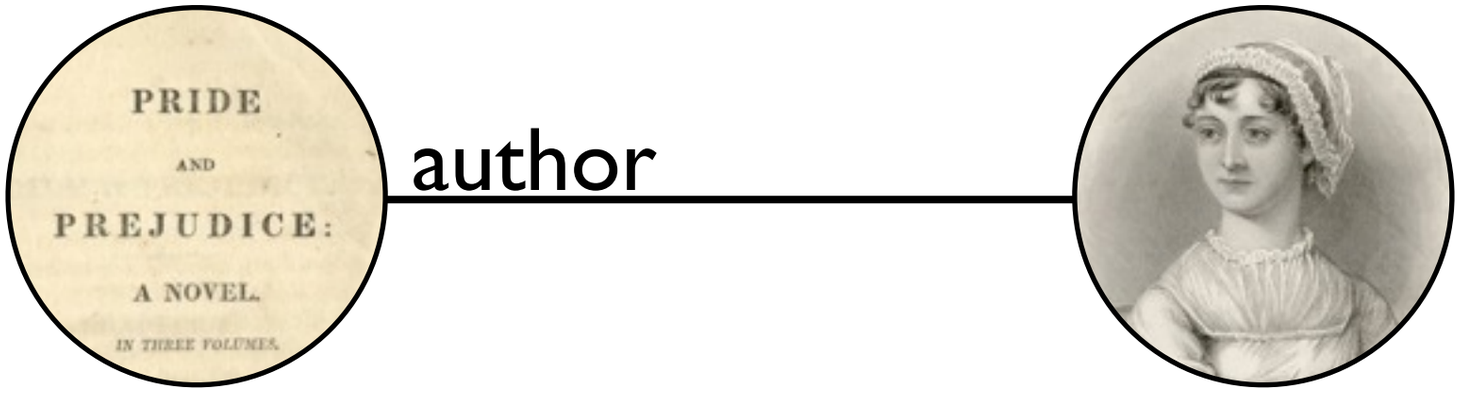
Find all the types that have a property that has an expected_type of /location/location:

```
[{
  "id":   null,
  "type": "/type/type",
  "properties": [{
    "id": null,
    "expected_type": {
      "id": "/location/location"
    }
  } ]
} ]
```

Phylogeny!

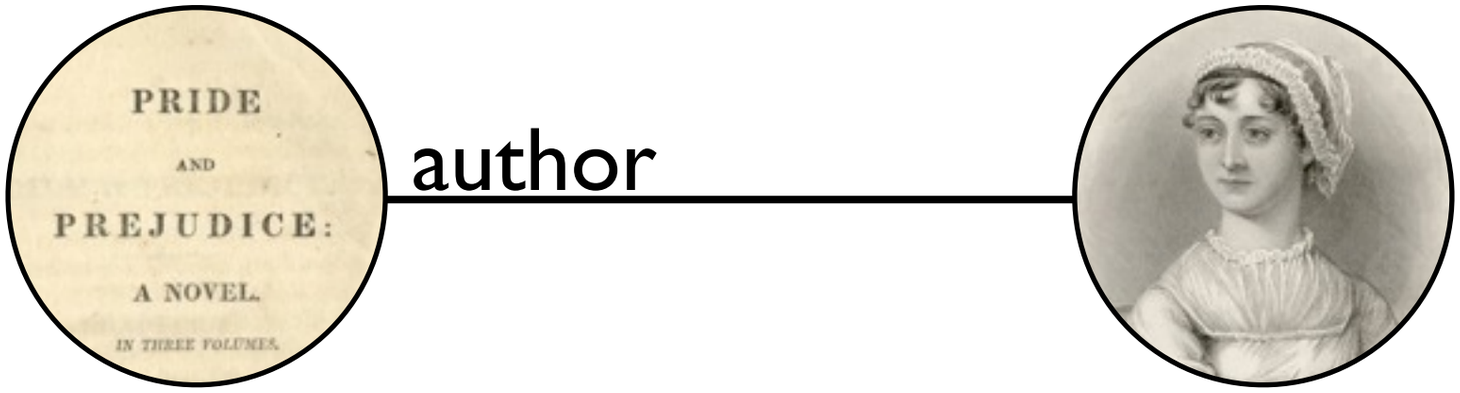
/location/location (contains)
/location/location (containedby)
/language/human_language (region)
/people/person (place_of_birth)
....

Master Properties (and Reverse Properties)




```
{
  "id":"/book/written_work/author",
  "type":"/type/property",
  "master_property":null,
  "reverse_property":null
}
```

Master Properties (and Reverse Properties)




```
{
  "id":"/book/written_work/author",
  "type":"/type/property",
  "master_property":null,
  "reverse_property":null
}
```

```
{
  "id": "/book/written_work/author",
  "type": "/type/property"
  "master_property": "/book/author/works_written",
  "reverse_property": null,
}
```

Google I/O 11

Link:
Looking into connections

A simple query

```
{ "id":"/en/jane_austen",
  "/people/person/languages":[{ "id":null }] }
```

languages

A simple query

```
{ "id":"/en/jane_austen",
  "/people/person/languages":[{ "id":null }] }
```

languages

A simple query

```
{ "id":"/en/jane_austen",
  "/people/person/languages":[{ "id":null }] }
```

languages

A simple query

Link Directive

```
{ "id":"/en/jane_austen",
  "/people/person/languages":[{"link":{"*":null}, "id":null }] }
```

languages

A simple query

Link Directive

```
{ "id":"/en/jane_austen",
  "/people/person/languages":[{"link":{"*":null}, "id":null }] }
```

languages

A simple query

Link Directive

```
{ "id":"/en/jane_austen",
  "/people/person/languages":[{"link":{"*":null}, "id":null }] }
```

languages

/type/link
- creator
- timestamp
- master_property
- reverse

.....

/type/link: "Used to access advanced features of MQL"

Freebase | Find topics... | Data | Schema | Apps | Docs | Sign In or Sign Up

Schema > Metaweb System Types > Link

Link Type

Key: /type/link

Created by Freebase Staff on November 26, 2006

Link is a pseudo type, used to access advanced features of MQL.

Table | Diagram | Build Query

Properties

What are Properties?

Property	ID	Expected Type	Description
type	/type/link/type	/type/type	*master, unique*
timestamp	/type/link/timestamp	/type/datetime	*master, unique*
creator	/type/link/creator	/type/user	*master, unique*
master_property	/type/link/master_property	/type/property	*master, unique*
reverse	/type/link/reverse	/type/boolean	*master, unique*
operation	/type/link/operation	/type/rawstring	*master, unique*
valid	/type/link/valid	/type/boolean	*master, unique*
attribution	/type/link/attribution	/type/attribution	*master, unique*
source	/type/link/source	/type/object	*master, unique*
target	/type/link/target	/type/object	*master, unique*

http://www.freebase.com/schema/type/link

/type/link: "Used to access advanced features of MQL"

/type/link: "Used to access advanced features of MQL"

Properties of /type/link:

timestamp

creator

operation

valid

master_property

reverse

source

target

target_value

/type/link: "Used to access advanced features of MQL"

Properties of /type/link:

timestamp
creator
operation
valid

History

master_property
reverse
source
target
target_value

/type/link: "Used to access advanced features of MQL"

Properties of /type/link:

History

timestamp
creator
operation
valid

Schema

master_property
reverse

source

target

target_value

/type/link: "Used to access advanced features of MQL"

Properties of /type/link:

History

timestamp
creator
operation
valid

Schema

master_property
reverse

Reflection

source
target
target_value

Who made the link?


```
{ "id":"/en/jane_austen",
  "/people/person/languages":[{"link":{"creator":null}, "id":null }] }
```


languages

Who made the link?

{ "id":"/en/jane_austen",
  "/people/person/languages":[{"link":{"creator":null}, "id":null }] }

"/user/jamie"

languages

The history of Jane (Austen's Types)

By an impartial, unprejudiced & informed historian

```
{ "id": "/en/jane_austen",
   "type": [{
     "link": {
       "timestamp":null,
       "operation":null,
       "valid":null
     },
     "id": null,
     "sort":"link.timestamp"
}] }
```

The history of Jane (Austen's Types)

By an impartial, unprejudiced & informed historian

```
{ "id": "/en/jane_austen",
  "type": [{
    "link": {
      "timestamp":null,
      "operation":null,
      "valid":null
    },
    "id": null,
    "sort":"link.timestamp"
}] }
```

```
{ "id": "/en/jane_austen",
   "type": [
    {
     "id": "/common/topic",
     "link": {
       "operation": "insert",
       "timestamp": "2006-10-22T09:07:06.0063Z",
       "valid":     true
     }},
    {
     "id": "/people/person",
    "link": {
      "operation": "insert",
      "timestamp": "2006-11-14T18:31:13.0003Z",
      "valid":     true
    }},
```

The history of Jane (Austen's Types)

By an impartial, unprejudiced & informed historian

```
{ "id": "/en/jane_austen",
   "type": [{
     "link": {
      "timestamp":null,
      "operation":null,
      "valid":null
     },
     "id": null,
     "sort":"link.timestamp"
}] }
```

```
{ "id": "/en/jane_austen",
    "type": [
     {
      "id": "/common/topic",
      "link": {
       "operation": "insert",
       "timestamp": "2006-10-22T09:07:06.0063Z",
       "valid":     true
      }},
     {
      "id": "/people/person",
      "link": {
       "operation": "insert",
       "timestamp": "2006-11-14T18:31:13.0003Z",
       "valid":     true
      }},
{
      "id": "/film/writer",
      "link": {
       "operation": "insert",
       "timestamp": "2006-11-30T18:50:40.0093Z",
       "valid":     false
}}
```

The history of Jane (Austen's Types)

By an impartial, unprejudiced & informed historian

```
{
  "id": "/film/writer",
  "link": {
    "operation": "delete",
    "timestamp": "2010-07-12T21:34:29.0000Z",
    "valid":     false
}},
```

/type/link: Used to access advanced features of MQL

Properties of /type/link:

History

timestamp
creator
operation
valid

Schema

master_property
reverse

Reflection

source
target
target_value

About the property for this link....

```
{
 "id": "/en/jane_austen",
 "/people/person/languages": [{
  "link": {
    "master_property": null
    "reverse":null
  },
  "id": null
 }]
}
```

languages

About the property for this link....

```
{
 "id": "/en/jane_austen",
 "/people/person/languages": [{
  "link": {
   "master_property": "/people/person/languages"
   "reverse": false
  },
  "id": null
 }]
}
```

languages

About the property for this link....

```
{
  "id": "/en/jane_austen",
  "/people/person/languages": [{
    "link": {
      "master_property": {
        "name":null,
        "expected_type": null
      },
      "reverse":null
    },
    "id": null
  }]
}
```

languages

About the property for this link....

```
{
  "id": "/en/jane_austen",
  "/people/person/languages": [{
    "link": {
      "master_property": {
        "name": "Languages"
        "expected_type": "/language/human_language"
      },
      "reverse": false
    },
    "id": null
  }]
}
```

languages

Cheating?!

```
{
  "id": "/en/ridley_scott",
  "/film/director/film": [
    {
      "link": {
        "master_property": {
          "name":          null,
          "expected_type": null,
          "reverse_property": {
            "name":null,
            "expected_type":null
          }
        },
        "reverse": null
      },
      "id": null
    },
  ]
}
```

Cheating?!

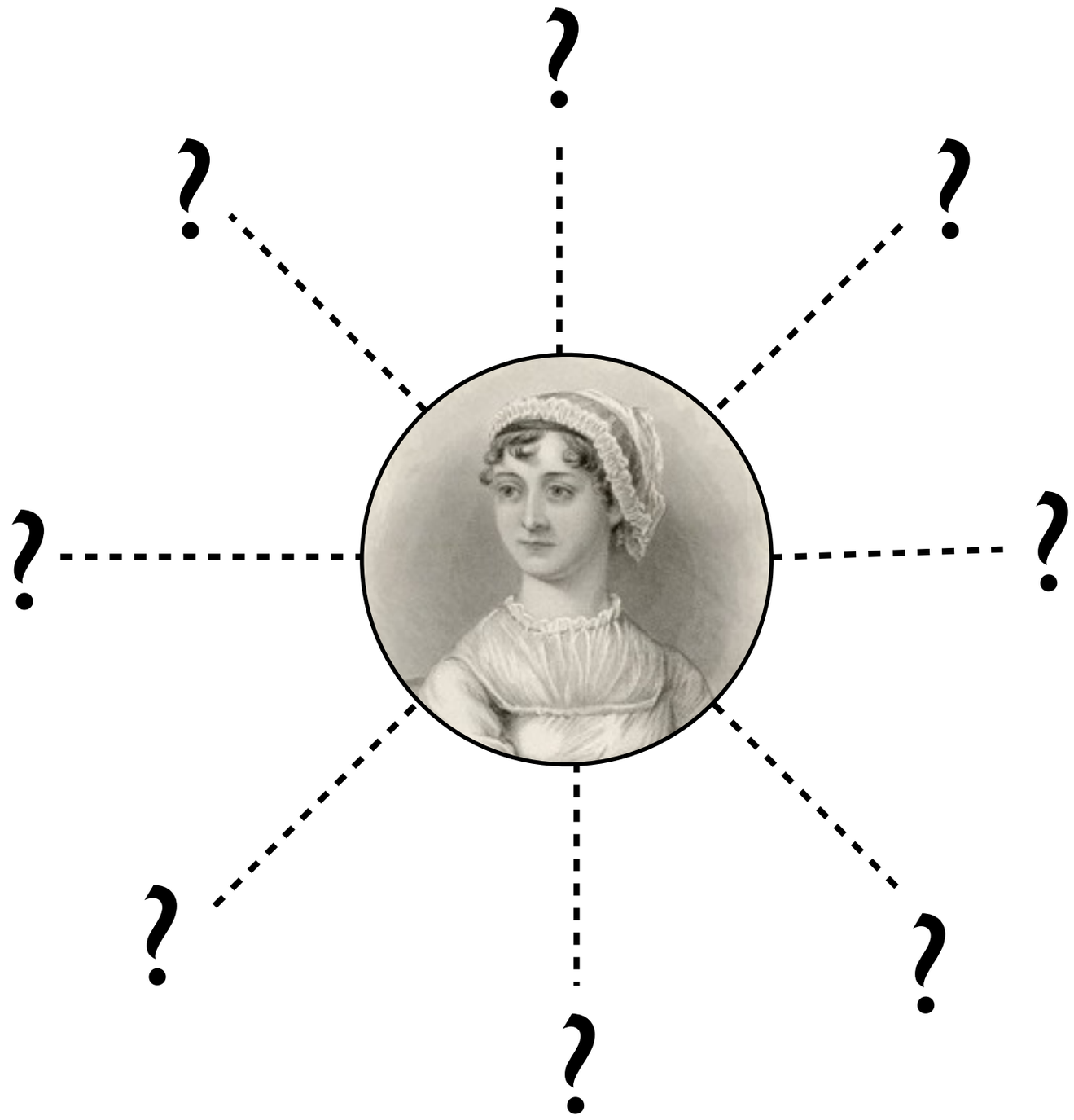
```
"/film/director/film": [
    {
      "id": "/en/1492_conquest_of_paradise",
      "link": {
        "master_property": {
          "expected_type": "/film/director",
          "name":          "Directed by",
          "reverse_property": {
            "expected_type": "/film/film",
            "name":          "Films directed"
          }
        },
        "reverse": true
      }
    }
```

Google I/O 11

Reflection:
Tell me about yourself

Google I/O 11

What's Connected Here?



Freebase | Find topics... | Data | Schema | Apps | Docs | jamie | Settings | Sign Out

You're in **Edit Mode**. Customize edit settings. | Show Topic Summary

Jane Austen Rename

Jane Austen (16 December 1775 – 18 July 1817) was an English novelist whose works of romantic fiction, set among the landed gentry, earned her a place as one of the most widely read writers in English literature, her realism and biting social commentary cementing her historical importance among scholars and critics. Austen lived her entire life as part of a close-knit family located on the lower fringes of the English landed gentry. She was educated primarily by her father and older brothers as... More

Edit Description | Read article at Wikipedia

Types Person , Deceased Person , Author , Person Or Being In Fiction , Fictional Character Creator , Influence Node , Film story contributor , Literature Subject , Ontology Instance , KWTopic + Add a type

Aliases Edit Aliases

People

Person /people/person

Date of birth:	Dec 16, 1775		
Place of birth:	location	contained by	
	Steventon	United Kingdom Hampshire Basingstoke and Deane	
Country of nationality:	United Kingdom, England		detail view »
Gender:	Female		
Profession:	Writer, Novelist		detail view »
Religion:	Anglicanism		
Ethnicity:	English		
Parents:	person	date of birth	country of nationality
	Cassandra Austen	1739	United Kingdom

Top Contributors to this topic:

Created by **Freebase Staff** Oct 22, 2006
Last edited by **jamie** 2 days ago

See Topic History

Web IDs What are Web IDs?

- Freebase 2
  - namespace /en, key jane_austen
  - namespace /m, key /m/040dv
- Library of Congress 1
- The New York Times 1
- Internet Movie Database 1
- Wikipedia 12
- Daylife 1
- Ellerdale 1
- Open Library 1
- TVRage 1
- Virtual International Authority File 1
- /user/jamie/nytdataid 1
- /wikipedia/ru_id 1

http://www.freebase.com/view/en/jane_austen

/type/link: Used to access advanced features of MQL

Properties of /type/link:

History

timestamp
creator
operation
valid

Schema

master_property
reverse

Reflection

source
target
target_value

What's Connected Here


```
[{
 "type": "/type/link",
 "source": {
  "id": "/en/jane_austen"
 },
 "master_property": null,
 "target": {
  "type": "/common/topic",
  "id": null
 }
}]
```

What's Connected Here

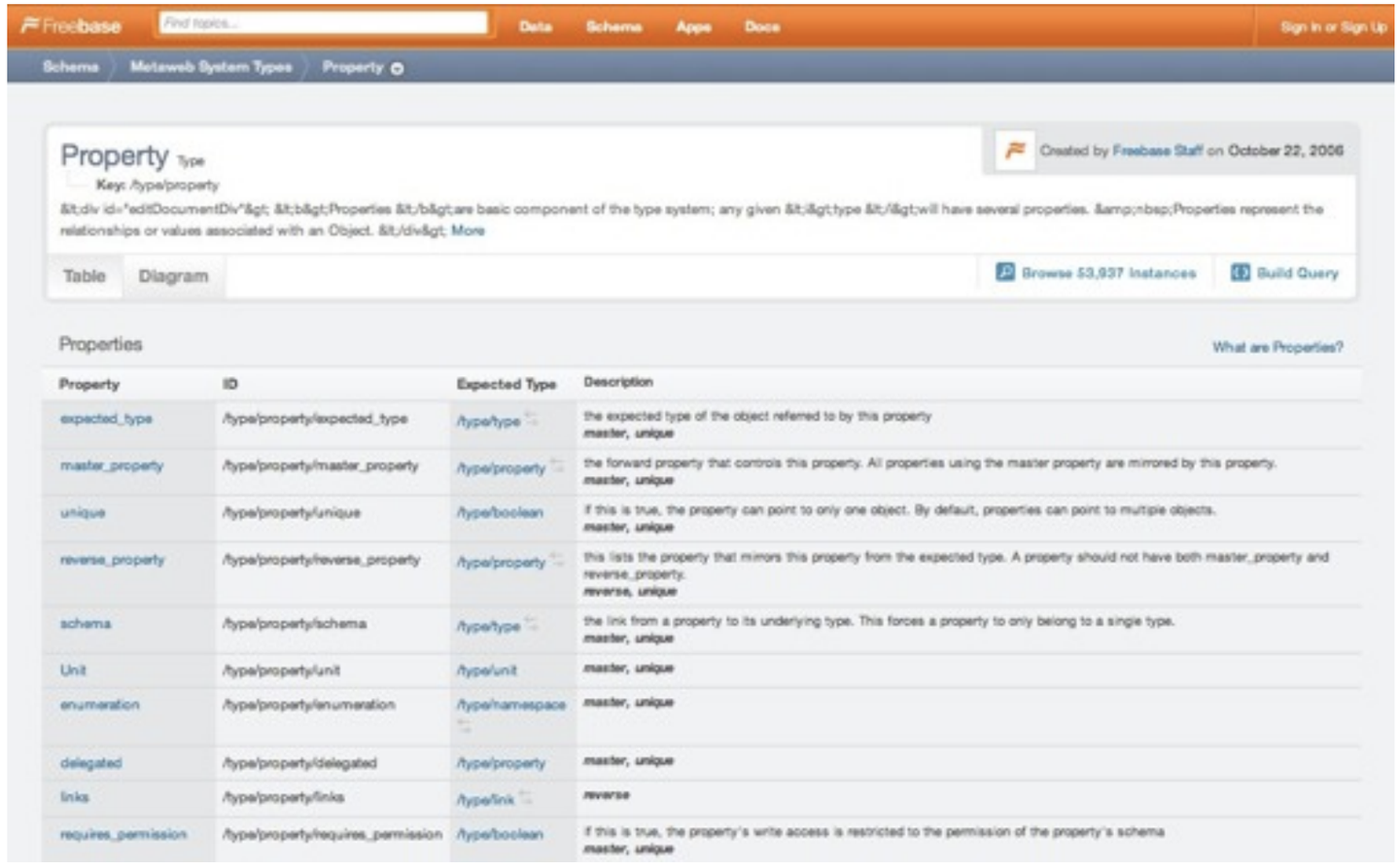
```
[{
 "type": "/type/link",
 "source": {
  "id": "/en/jane_austen"
 },
 "master_property": null,
 "target": {
  "type": "/common/topic",
  "id": null
 }
}]
```

```
{
    "master_property": "/people/person/place_of_birth",
    "source": {
      "id": "/en/jane_austen"
    },
    "target": {
      "id":   "/en/steventon",
      "type": "/common/topic"
    },
    "type":         "/type/link"
   },{
    "master_property": "/people/deceased_person/place_of_death",
    "source": {
      "id": "/en/jane_austen"
    },
    "target": {
      "id":   "/en/winchester",
      "type": "/common/topic"
    },
    "type":         "/type/link"
  },{
    "master_property": "/people/person/gender",
    "source": {
      "id": "/en/jane_austen"
    },
    "target": {
      "id":   "/en/female",
      "type": "/common/topic"
    },
    "type":         "/type/link"
  }.....
```

Property "Links"

Only works on Master Properties!

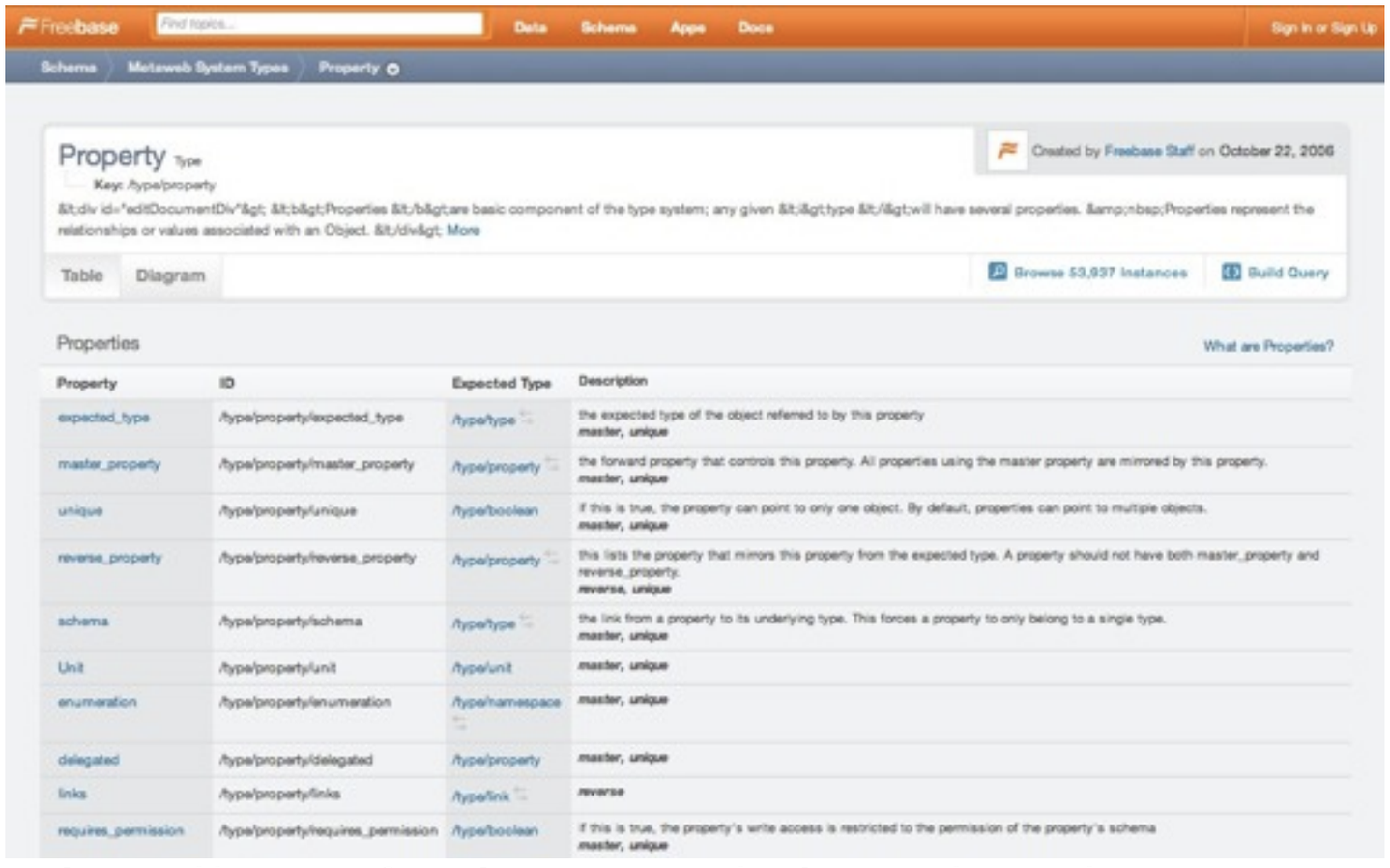
Property "Links"



http://www.freebase.com/schema/type/property

Only works on Master Properties!

Property "Links"

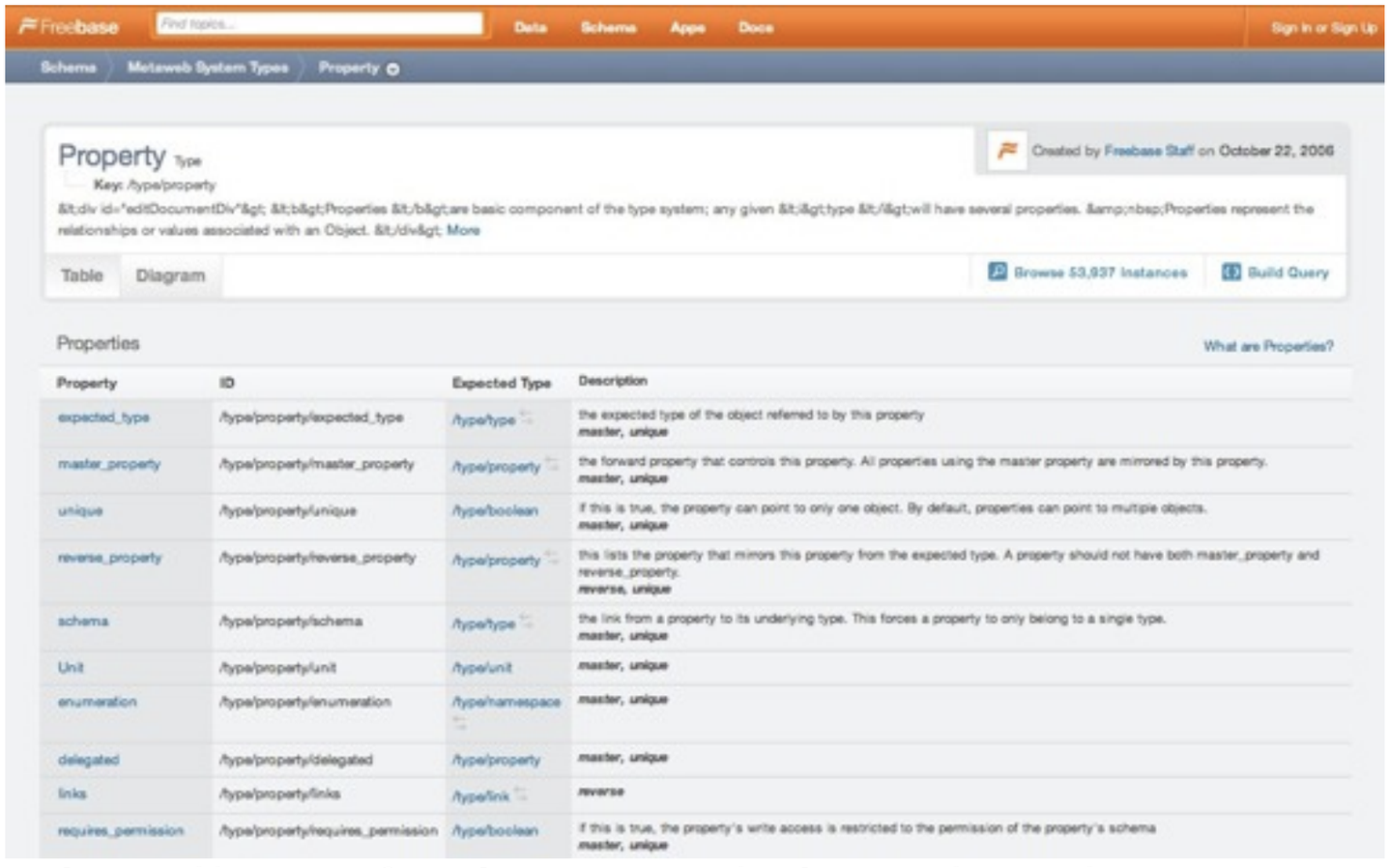


http://www.freebase.com/schema/type/property

```
{
  "id": "/type/property",
  "type":"/type/type",
  "properties":[{
      "id": null,
      "expected_type": "/type/link"
  }]
}
```

Only works on Master Properties!

Property "Links"



http://www.freebase.com/schema/type/property

```
{
  "id": "/type/property",
  "type":"/type/type",
  "properties":[{
      "id": "/type/property/links"
      "expected_type": "/type/link"
  }]
}
```

Only works on Master Properties!

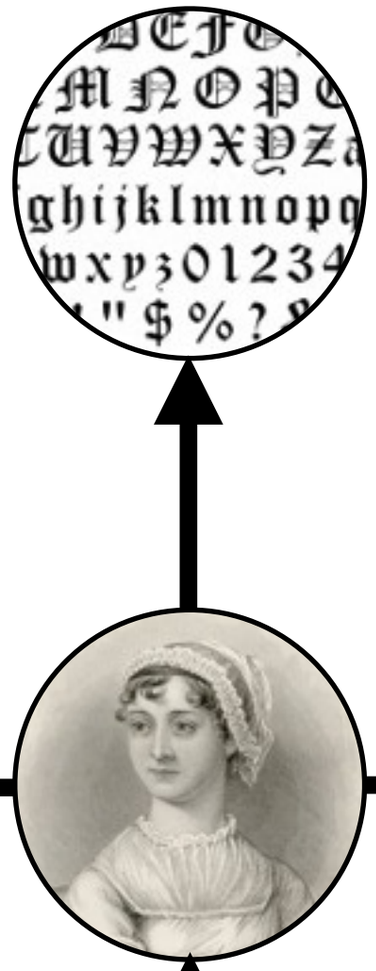
Property "Links"

```
{
  "id": "/people/person/place_of_birth",
  "type": "/type/property",
  "links":[{"*":null}]
}
```

```
{
  "id":   "/people/person/place_of_birth",
  "links": [
    {
      "attribution":   "/user/robert",
    "creator":       "/user/robert",
    "master_property": "/people/person/place_of_birth",
    "operation":     "insert",
    "reverse":       false,
    "source":        "Steve Martin",
    "target":        "Waco",
    "target_value":  null,
    "timestamp":     "2006-12-02T01:12:05.0000Z",
    "type":          "/type/link",
    "valid":         true
  }.....
```

What's Connected Here

What's Connected Here



Two Query Approach

```
[{
  "type": "/type/link",
  "source": {
      "id": "/en/jane_austen"
  },
  "master_property": null,
  "target": {
      "type":"/common/topic",
      "id": null
  }
}]
```

Two Query Approach

```
[{
 "type": "/type/link",
 "source": {
     "id": "/en/jane_austen"
 },
 "master_property": null,
 "target": {
     "type":"/common/topic",
     "id": null
 }
}]
```

```
[{
 "type": "/type/link",
 "source": {
  "id": null,
  "type":"/common/topic"
 },
 "master_property": null,
 "target": {
   "id":"/en/jane_austen"
 }
}]
```

What's Connected Here v.2

```
{
  "id": "/en/jane_austen",
  "type": [{
    "id": null,
    "properties": [{
      "id": null
    }]
  }]
}
```

What's Connected Here v.2

```
{
  "id": "/en/jane_austen",
  "type": [{
    "id": null,
    "properties": [{
      "id": null,
      "links":[{
          "source":{"id":"/en/jane_austen"},
          "target":{"id":null}
      }]
    }]
  }]
}
```

What's Connected Here v.2

```
 {
"id": "/en/jane_austen",
"type": [{
  "id": null,
  "properties": [{
    "id": null,

    "links": [{
      "source": {
        "id": "/en/jane_austen"
      },
      "target": {
        "id":       null,
        "optional": true
      },
      "optional":     true,
      "target_value": null
    }],

    "master_property": {
      "optional": true,
      "links": [{
        "source": {
          "id": null
        },
        "target": {
          "id": "/en/jane_austen"
        }
      }]
    }
  }]
}]
}
```

Google I/O 11

Property driven programming:

Acting on meaning

Google I/O 11

The typical approach

The typical approach

```
{ "id":"/en/jane_austen",
  "/people/person/place_of_birth":null }
```

The typical approach

```
{ "id":"/en/jane_austen",
  "/people/person/place_of_birth":null }

{ "id":"/en/jane_austen",
  "/people/person/languages":[{"id":null }] }
```

The typical approach

```
{ "id":"/en/jane_austen",
  "/people/person/place_of_birth":null }

{ "id":"/en/jane_austen",
  "/people/person/languages":[{"id":null }] }

{ "id":"/en/jane_austen",
  "/book/author/works_written":[{"id":null }] }

{ "id":"/en/jane_austen",
  "...................":[{"id":null }] }

{ "id":"/en/jane_austen",
  "...................":[{"id":null }] }
```

What queries for Liszt?

Reflect and React

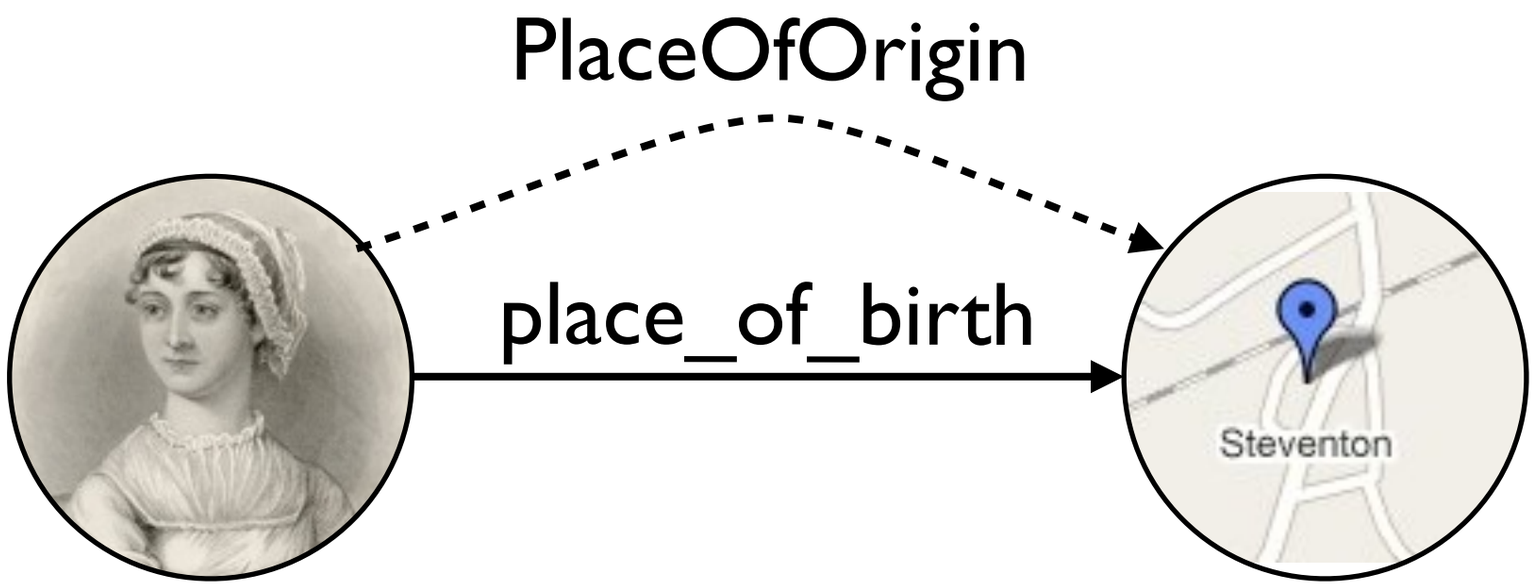
```
{
  "id": YOUR TOPIC HERE,
  "type": [{
    "id": null,
    "properties": [{
      "id": null,

      "links": [{
        "source": {
          "id": YOUR TOPIC HERE
        },
        "target": {
          "id":       null,
          "optional": true
        },
        "optional":     true,
        "target_value": null
      }],

      "master_property": {
        "optional": true,
        "links": [{
          "source": {
            "id": null
          },
          "target": {
            "id": YOUR TOPIC HERE
          }
        }]
} }] }] }
```

Google I/O 11

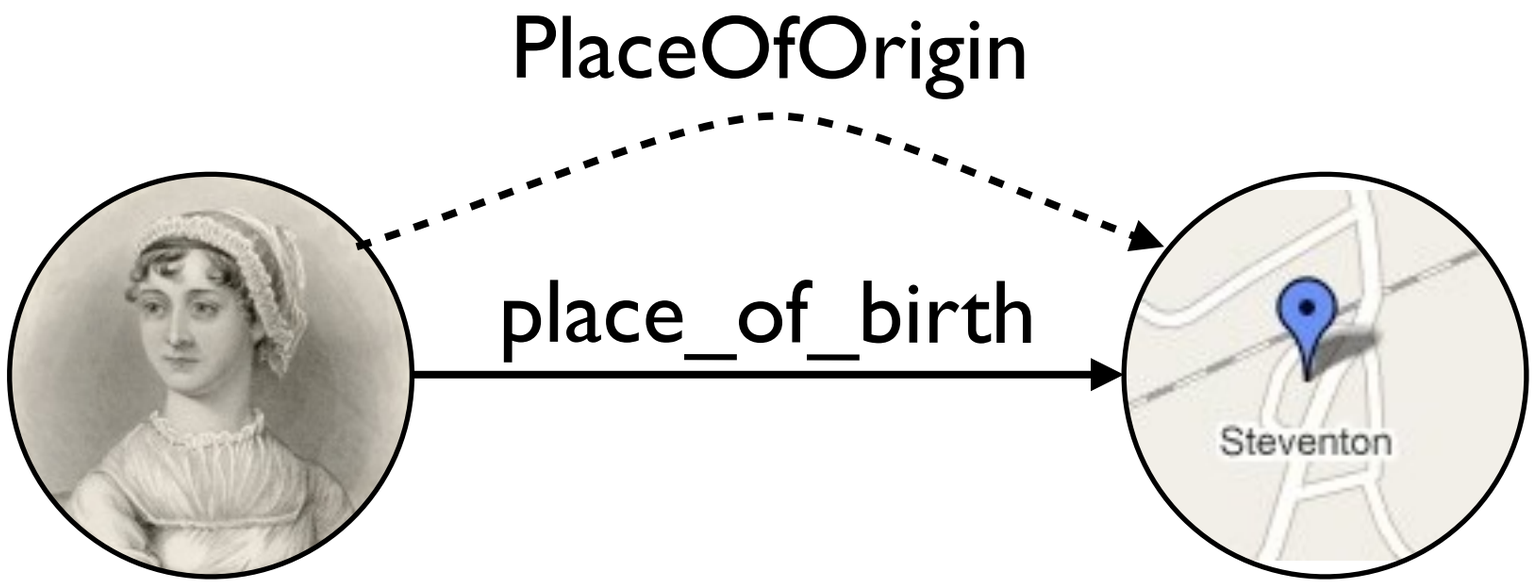
Metaschema:
Doing more with less

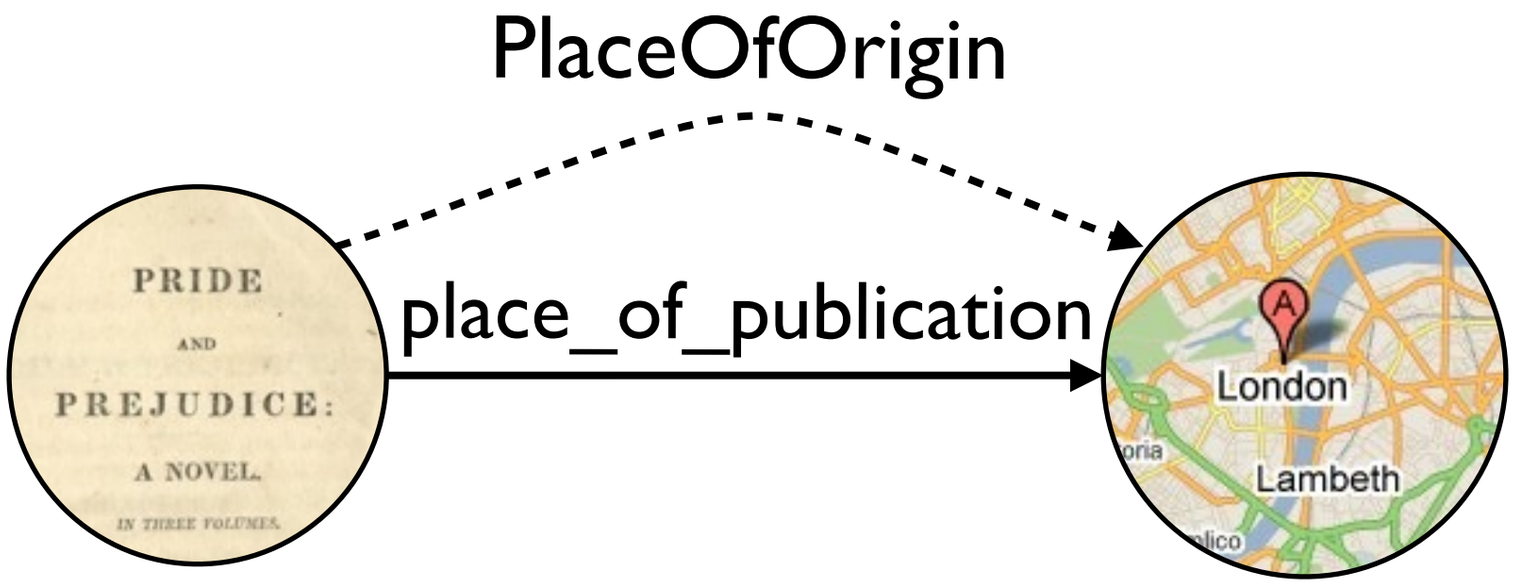
Google I/O 11

Generalized relationships



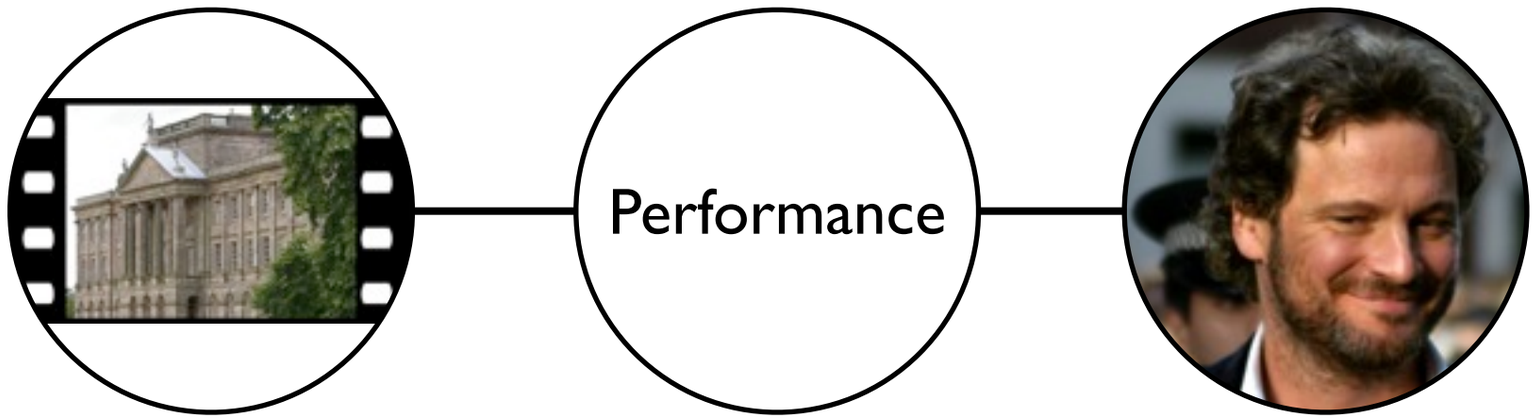
Generalized relationships



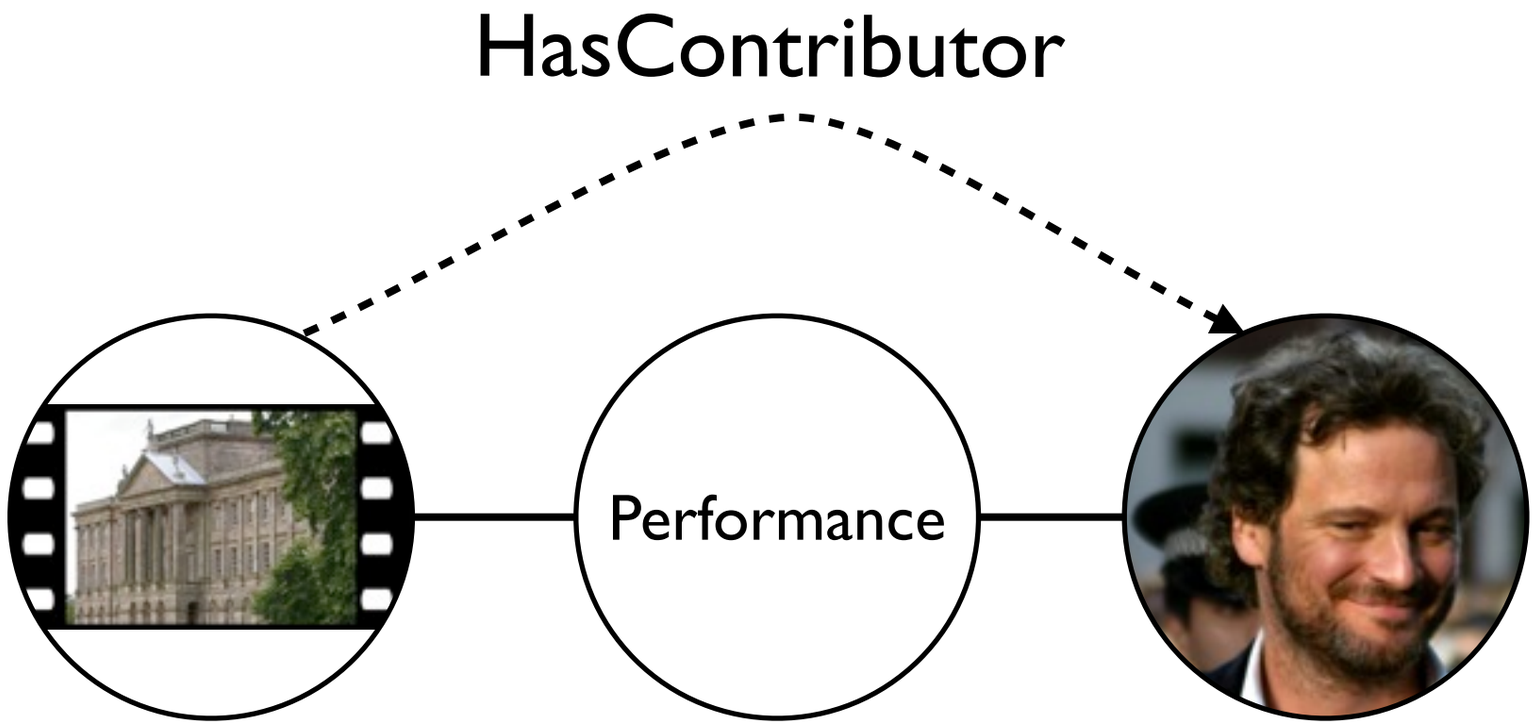


Generalized relationships

Generalized relationships



Generalized relationships



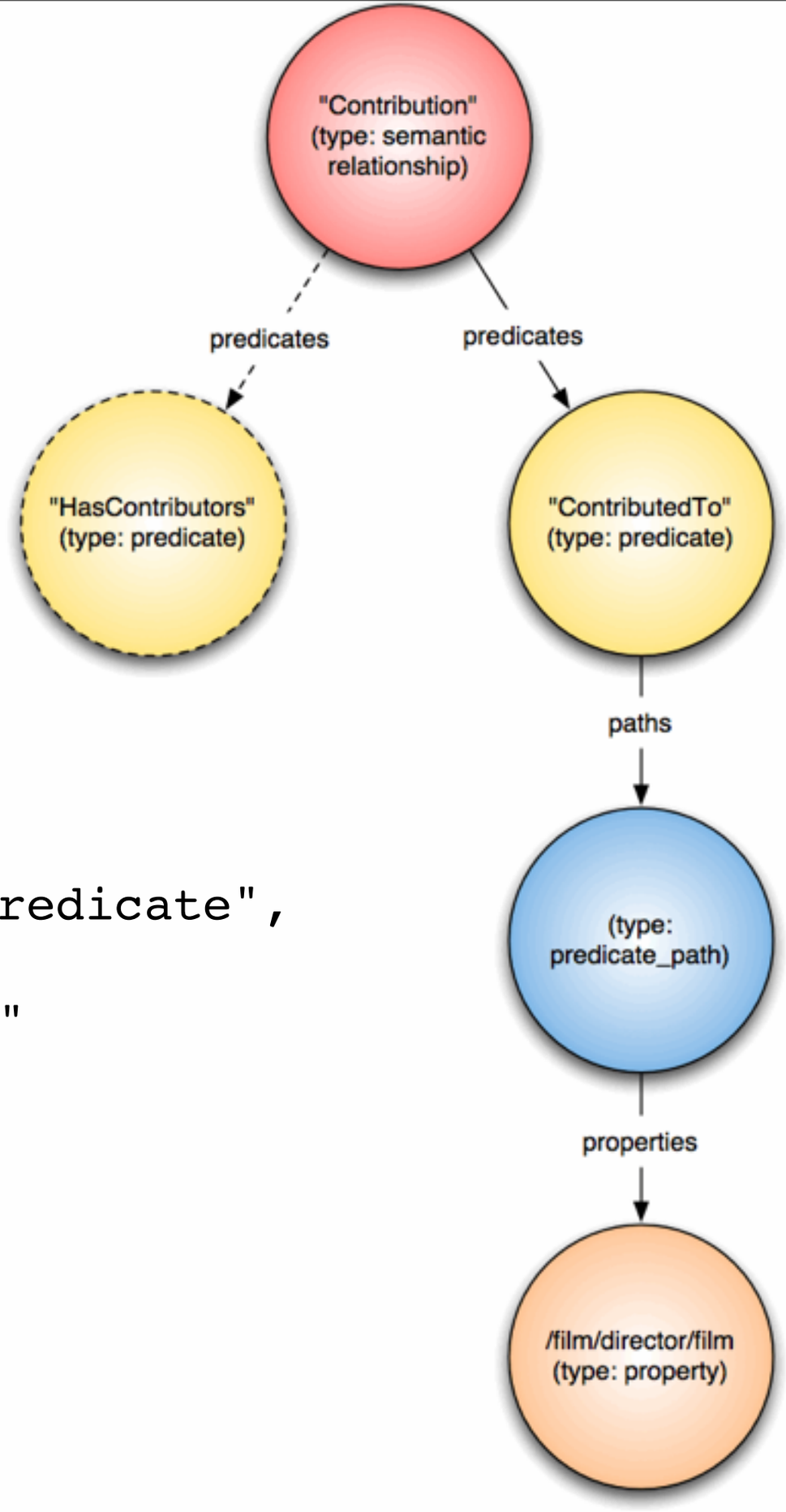
Higher Order Relationships

Participation
Practitioner
Creation
Service Area
Place of Origin
Means of Demise
Character Portrayal
Categorical
Administration
Place of Occurrence
Production
Location
Title
Whole/Part
Composition
Discovery

Contribution
Broader/Narrower
Peer
Time Point
Exhibition
Distribution
Name
Superclass/Subclass
Status
Symbol
Parent/Child
Publication
Membership
Leadership
Ownership
Adaptation

Event/Location
Fictional
Genre
Succession
Permitted Use
Identifier
Organizational Center
Character Appearance
Means of Expression
Certification
Series
Measurement
Abstract/Concrete
Subject

Metaschema Schema




```
[{
  "name": null,
  "type": "/base/fbontology/semantic_predicate",
  "paths": {
    "properties": "/film/director/film"
  }
}]
```

Metaschema Reflection

```
 {
"id": "/en/jane_austen",
"type": [{
  "properties": [{
    "id":null,
    "!/base/fbontology/predicate_path/properties": [{
      "predicate": null
    }]
  }]
 }]
}
```

Metaschema Reflection

/en/robert_redford

```
"/people/person"
"/film/actor"
"/film/producer"
"/film/director"
"/theater/theater_actor"
"/tv/tv_actor"
"/tv/tv_producer"
"/award/award_winner"
"/award/award_nominee"
"/influence/influence_node"
"/business/board_member"
```

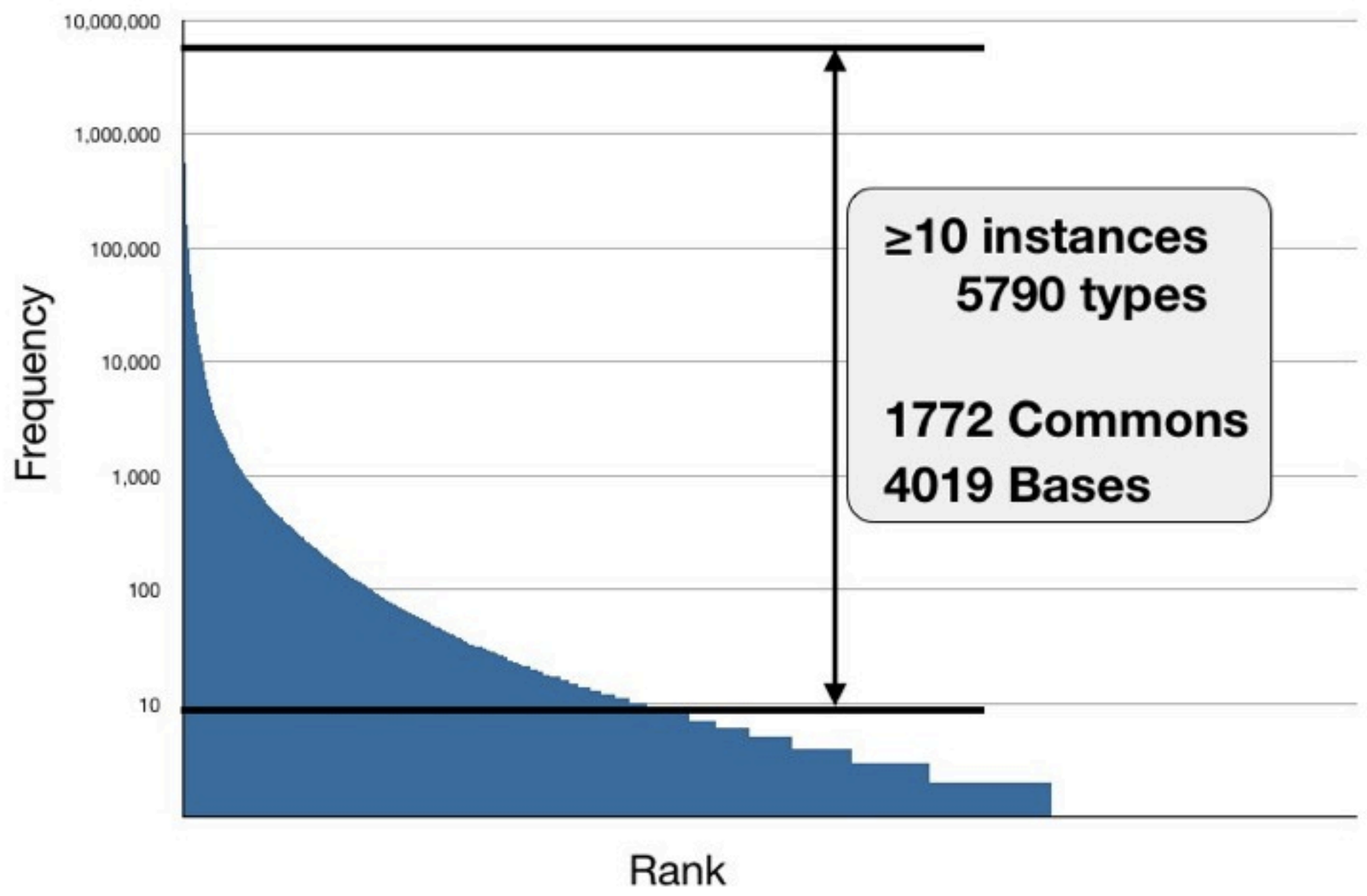
Metaschema Reflection

```
 {
"id": "/en/robert_redford",
"type": [{
  "properties": [{
    "id":null,
    "!/base/fbontology/predicate_path/properties": [{
      "predicate": null
    }]
  }]
 }]
}
```

Metaschema Reflection

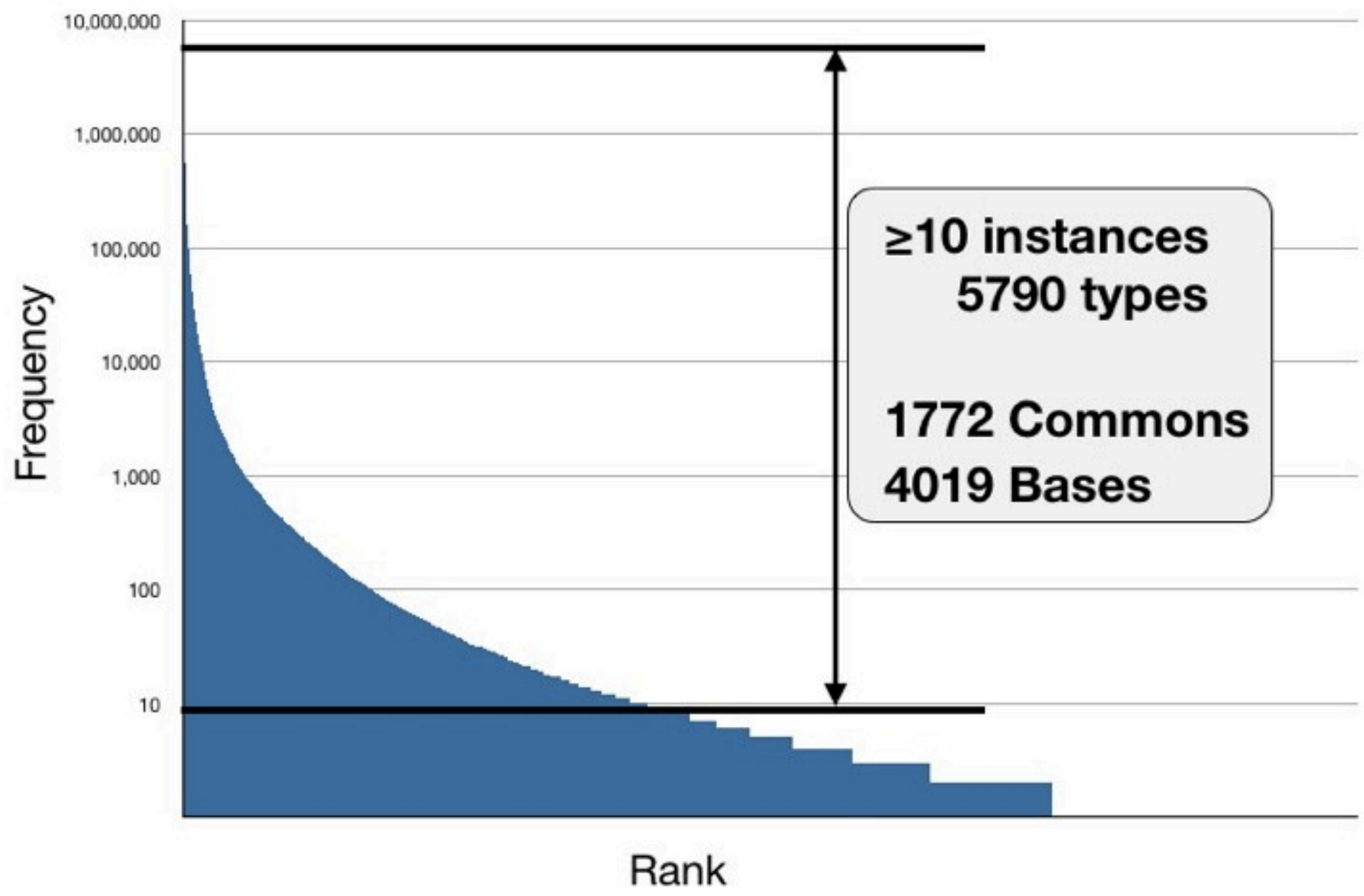
```
[{
 "type": "/type/link",
 "source": {
  "id": "/en/jane_austen"
 },
 "master_property": {
  "!/base/fbontology/predicate_path/properties": {
   "predicate": null
  }
 },
 "target": {
  "type": "/common/topic",
  "name": null
 }
}]
```

Reduce the number of properties your program needs to "understand" with Metaschema!



Reduce the number of properties your program needs to "understand" with Metaschema!

- React to the properties in 1772 Types?
- React to 46 properties?



http://io2011.freebaseapps.com

All the Queries from the talk, links, etc.
Property Driven Demo App

http://io2011.freebaseapps.com

All the Queries from the talk, links, etc.
Property Driven Demo App

http://www.freebase.com/docs

http://lists.freebase.com

More information is just a click away

Link Queries

Information about Jane Austen's language link

```
{ "id":"/en/jane_austen",
"/people/person/languages":[{"link":{"*":null}, "id":null }] }
```

Try It!

Who made Jane Austen's language link?

```
{ "id":"/en/jane_austen",
"/people/person/languages":[{"link":{"creator":null}, "id":null }] }
```

Try It!

The history of Jane Austen's Types

```
{ "id": "/en/jane_austen",
   "type": [{
     "link": {
       "timestamp":null,
       "operation":null,
       "valid":null
     },
     "id": null,
     "sort":"link.timestamp"
}] }
```

Try It!

Info about the language property

```
1  //returns a chunk of HTML to display for the id
2  function reflection(id){
3    output = "";
4
5    //FOR METASCHEMA: change "reflect" to "metareflect" in this next line
6    var reflectquery = acre.require("reflect").query;
7
8    //the metareflect query is almost exactly the same as the reflect query.
9    //the one small change lets us see the property as a metaschema relationship
10
11   reflectquery = acre.freebase.extend_query(reflectquery, {"id":id});
12
13   //we can also extend the query structure ourselves
14   reflectquery["type"][0]["properties"][0]["links"][0]["source"]["id"] = id;
15   reflectquery["type"][0]["properties"][0]["master_property"]["links"][0]["target"]["id"] =
   id;
16
17   var rez = acre.freebase.mqlread(reflectquery).result;
18   console.log(rez); //for console debug mode
19
20   //iterate through the result structure
21   for(typeindex in rez["type"]){
22     for(propindex in rez["type"][typeindex]["properties"]){
23       var propstruc = rez["type"][typeindex]["properties"][propindex];
24
25       //what property is this
26       var theprop = propstruc["id"];
27
28       //the metaschema version of this query has one more structure to inspect
29       //but using propstruc["!/base/fbontology/predicate_path/properties"][0] is a CHEAT!
30       //this version will only get you single property path relations
31       //to do this fully we need to change the metareflect query to dig a little deeper
32
33       if(propstruc["!/base/fbontology/predicate_path/properties"] != undefined){
34         theprop = propstruc["!/base/fbontology/predicate_path/properties"][0]["predicate"];
35       }
36
```

Apps ▾ | property driven example ▾ | jamie | Settings | Sign Out

Files | New File | Save | Save All | Save As | Undo | Redo | Indent | File ▾ | Editor ▾ | Help ▾ | Line 6 | View | View with Console

- ▼ Templates
  - index
- ▼ Scripts
  - doreflect
  - prop_actions
- ▼ Queries
  - metareflect
  - reflect

```
1  //returns a chunk of HTML to display for the id
2  function reflection(id){
3    output = "";
4
5    //FOR METASCHEMA: change "reflect" to "metareflect" in this next line
6    var reflectquery = acre.require("reflect").query;
7
8    //the metareflect query is almost exactly the same as the reflect query.
9    //the one small change lets us see the property as a metaschema relationship
10
11   reflectquery = acre.freebase.extend_query(reflectquery, {"id":id});
12
13   //we can also extend the query structure ourselves
14   reflectquery["type"][0]["properties"][0]["links"][0]["source"]["id"] = id;
     rties"][0]["master_property"]["links"][0]["target"]["id"] =

     d(reflectquery).result;
     e debug mode

     tructure

     [typeindex]["properties"]){
     ][typeindex]["properties"][propindex];


     d"];

     f this query has one more structure to inspect
     ase/fbontology/predicate_path/properties"][0] is a CHEAT!
     et you single property path relations
      to change the metareflect query to dig a little deeper

     ology/predicate_path/properties"] != undefined){
     ase/fbontology/predicate_path/properties"][0]["predicate"];
```

Not a very pretty app - but clone this app and try adding prop_action functions.
You can switch to a Metaschema representation by following the comment on line 6 of "doreflect."

To clone this app (so you have edit rights):
1) click the "view source" link at the bottom of this page,
2) click the title of the app ("Property Driven Example") and then click the "Clone this App" but n.

Search:

Alias: The Father of Our Country
Alias: The Father of the United States

Image 

Image 

Image 

Image 

Find

Presentation Feedback http://goo.gl/LGT8c

Lyme Park (/en/lyme_park)
Attribution: Harlequeen
http://www.flickr.com/photos/harlequeen/4204033771/

Colin Firth photo (/en/colin_firth)
Attribution: Caroline Bonarde Ucci
http://en.wikipedia.org/wiki/File:ColinFirth05.jpg

Creative Commons Attribution Icon
Attribution: Creative Commons
http://creativecommons.org/about/downloads

Head shot of /user/jamie
Attribution: Freebase.com
http://www.freebase.com/view/m/01x375d

CC-BY Image Attribution