WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

Datenorientierte Systemanalyse: RDF, Linked Data & SPARQL

26/05/2014

Axel Polleres

Datenorientierte Systemanalyse/ Datenanalyse

- **Stundengeiederholung =**
- Ein Web-Interface zu unserer Datenbank bauen
  - JSTL und andere Scriptsprachen
  - Daten darstellen
  - Daten eingeben/löschen
  - Charts erstellen

Overview

- RDF ... (= Resource Description Framework), ein standardisiertes Datenformat speziell für das Publizieren von Daten am Web

- SPARQL ... Eine Abfragesprache (ähnlich SQL) fuer RDF Daten, mit der man RDF Daten **direkt von der Quelle** abfragen kann (ohne vorher alle Daten importieren zu müssen).

Sneak preview for today:

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

- Linked Open Data & SPARQL:
- „The area and population of countries of the EU“

```
SELECT ?X ?L ?A ?P
WHERE {  ?X a <http://dbpedia.org/ontology/Country> ;
            <http://purl.org/dc/terms/subject>
      <http://dbpedia.org/resource/Category:Member_states_of_the_European_Union>;
            rdfs:label ?L  ;
            <http://dbpedia.org/property/populationCensus> ?P;
            <http://dbpedia.org/property/areaKm> ?A .
         FILTER(lang(?L) = "en" && isNumeric(?A) && isNumeric(?P) )
      }
```

- Execute this query over wikipedia: Link
- Try it out yourself at http://live.dbpedia.org/sparql/

EFMD EQUIS ACCREDITED

What we can do already with this:

- To get the results of this query in CSV via curl

```
curl "http://live.dbpedia.org/sparql" --data-urlencode 'query=SELECT ?X  ?L ?A ?P
WHERE {  ?X a <http://dbpedia.org/ontology/Country>; <http://purl.org/dc/terms/
subject> <http://dbpedia.org/resource/Category:Member_states_of_the_European_Union>;
rdfs:label ?L  ; <http://dbpedia.org/property/populationCensus> ?P; <http://
dbpedia.org/property/areaKm> ?A . FILTER(lang(?L) = "en" && isNumeric(?A) &&
isNumeric(?P) )}' -H 'Accept: text/csv' -o eu_countries.csv
```

- Import the CSV to your Database:

```
CREATE TABLE EU_countries (url varchar(100), name
varchar(50), area decimal, population integer);
```

```
\COPY EU_countries FROM 'eu_countries.csv'
WITH DELIMITER ',' CSV HEADER
```

JSP to show population density:

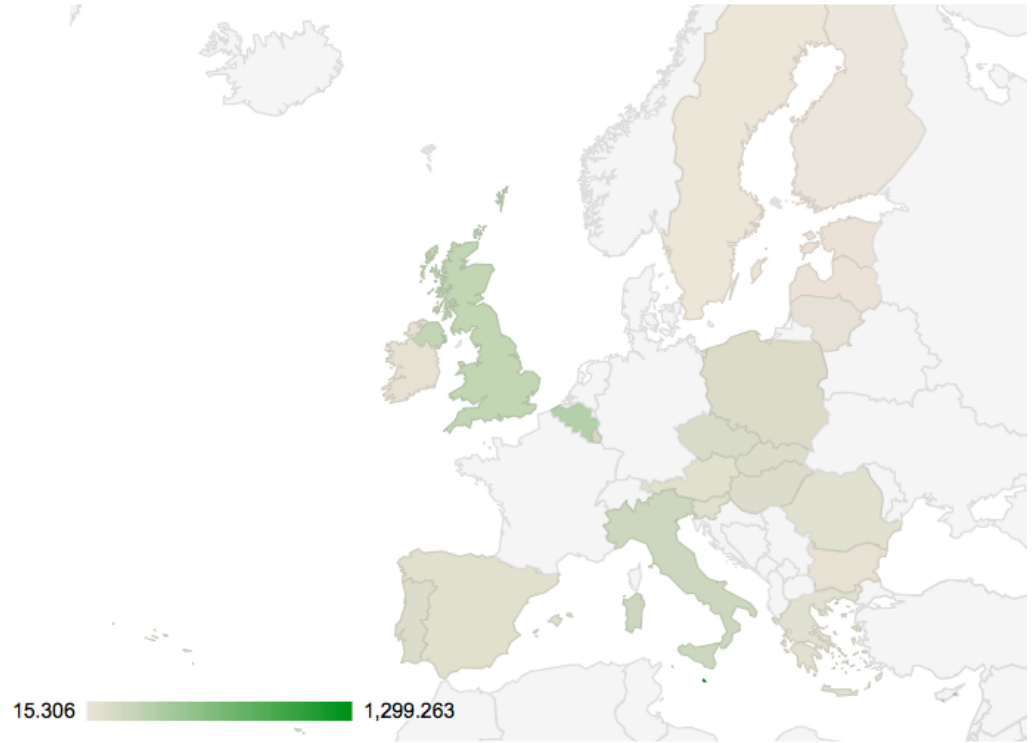
WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

- Start from this example:
- https://google-developers.appspot.com/chart/interactive/docs/gallery/geochart

- And create a map with the population densities...

EFMD EQUIS ACCREDITED

http://balrog.wu-wien.ac.at:8180/apollere/geochart.jsp



... Note that some countries are missing (e.g. France)

Linked Data and SPARQL

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

- Quick introduction to RDF
- Linked Data
- SPARQL

EFMD EQUIS ACCREDITED

Quick intro to RDF

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

- The „Resource Description Framework" = a Simple Universal Data Format for the Web.
- Represent ANY information as **triples**

Subject **P**redicate **O**bject

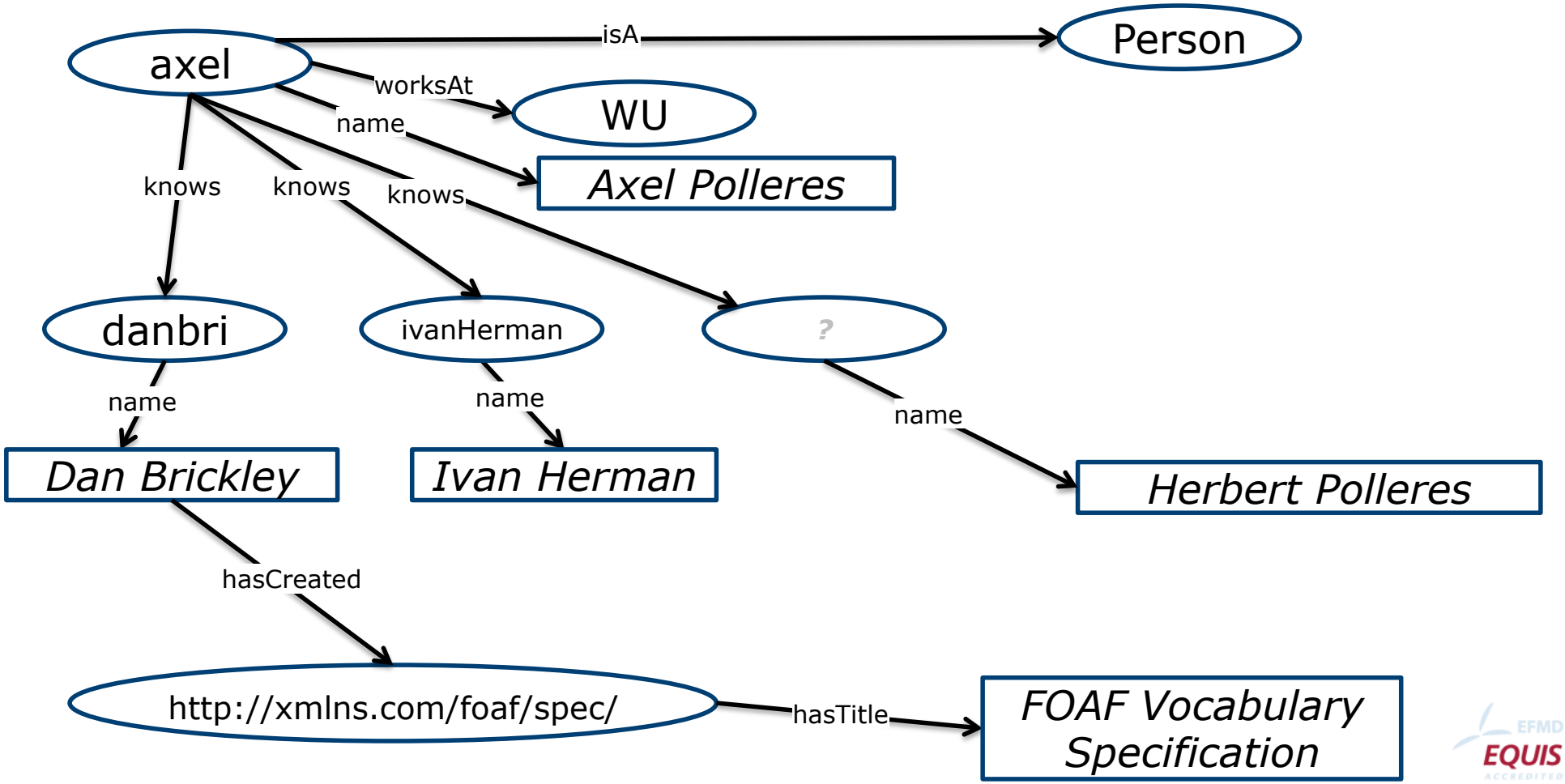
- "simplest possible database schema", data just a set of triples:

axel isA Person .
axel hasName "Axel Polleres".
axel worksAt WU .
axel knows danbri .
axel knows ivanHerman.
ivanHerman hasName "Ivan Herman".
danbri hasName "Dan Brickley" .
danbri hasCreated http://xmlns.com/foaf/spec/ .
http://xmlns.com/foaf/spec/ hasTitle *"FOAF Vocabulary Specification"* .
axel knows ***someone who*** hasName „Herbert Polleres".

EFMD EQUIS ACCREDITED

RDF triples form a graph

- RDF triples can be viewed as a directed labelled graph:



RDF triples form a graph

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

axel isA Person .
axel hasName "Axel Polleres".
axel worksAt WU .
axel knows danbri .
axel knows ivanHerman.
ivanHerman hasName "Ivan Herman".
danbri hasName "Dan Brickley" .
danbri hasCreated http://xmlns.com/foaf/spec/ .
http://xmlns.com/foaf/spec/ hasTitle "*FOAF Vocabulary Specification*" .
axel knows *someone who* hasName „Herbert Polleres".

- In RDF:
- All **nodes** are URIs, literals, or so called „blank nodes"
- **Edges** are also labelled with URIs.

EFMD EQUIS ACCREDITED

RDF

<http://polleres.net#me> isA <http://xmlns.com/foaf/0.1/Person> .
<http://polleres.net#me> hasName "Axel Polleres".
<http://polleres.net#me> worksAt <http://www.wu.ac.at>.
<http://polleres.net#me> knows <http://danbri.org/foaf.rdf#danbri>.
<http://polleres.net#me> knows <http://www.ivan-herman.net/foaf.rdf#me> .
<http://www.ivan-herman.net/foaf.rdf#me> hasName „Ivan Herman".
<http://danbri.org/foaf.rdf#danbri> hasName „Dan Brickley" .
<http://danbri.org/foaf.rdf#danbri> hasCreated <http://xmlns.com/foaf/spec/> .
<http://xmlns.com/foaf/spec/> hasTitle *"FOAF Vocabulary Specification"* .
<http://polleres.net#me> knows ***_:someone .***
_:someone hasName „Herbert Polleres".

- In RDF:
- All **nodes** (i.e. subjects, objects) are URIs, literals, or so called **„blank nodes"**
- **Edges** are also labelled with URIs.

RDF

<http://polleres.net#me> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .
<http://polleres.net#me> <http://xmlns.com/foaf/0.1/name> "Axel Polleres".
<http://polleres.net#me> <http://xmlns.com/foaf/0.1/workplaceHomepage> <http://www.wu.ac.at> .
<http://polleres.net#me> <http://xmlns.com/foaf/0.1/knows> <http://danbri.org/foaf.rdf#danbri>.
<http://polleres.net#me> <http://xmlns.com/foaf/0.1/knows> <http://www.ivan-herman.net/foaf.rdf#me> .
<http://www.ivan-herman.net/foaf.rdf#me> <http://xmlns.com/foaf/0.1/name> „Ivan Herman".
<http://danbri.org/foaf.rdf#danbri> <http://xmlns.com/foaf/0.1/name> „Dan Brickley" .
<http://danbri.org/foaf.rdf#danbri> <http://xmlns.com/foaf/0.1/made> <http://xmlns.com/foaf/spec/> .
<http://xmlns.com/foaf/spec/> <http://purl.org/dc/elements/1.1/title> "*FOAF Vocabulary Specification*" .
<http://polleres.net#me> <http://xmlns.com/foaf/0.1/knows> _:*someone* .
_:*someone* <http://xmlns.com/foaf/0.1/name> „Herbert Polleres".

- In RDF:
- All **nodes** are URIs, literals, or so called „blank nodes"
- **Edges** (i.e, predicates) are also labelled with URIs.

RDF syntax

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

- There are different serialization syntaxes for RDF

- RDF/XML
  - http://www.w3.org/TR/REC-rdf-syntax/
- Turtle
  - http://www.w3.org/TR/turtle/
- RDFa (i.e., RDF "embedded" in HTML)
  - http://www.w3.org/TR/rdfa-syntax/
- RDF in JSON
  - http://www.w3.org/TR/json-ld/

- We'll use **Turtle** syntax in this lecture:
  - sufficient to write all RDF
  - almost human-readable
  - also the basis for SPARQL

EFMD EQUIS ACCREDITED

Turtle syntax

- Triples separated by `.`,
- URIs in `<`,`>`,
- Literals in quotes,
- blank nodes labeled with prefix `_:`

<http://polleres.net#me> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .

<http://polleres.net#me> <http://xmlns.com/foaf/0.1/name> "Axel Polleres".

<http://polleres.net#me> <http://xmlns.com/foaf/0.1/workplaceHomepage> <http://www.wu.ac.at> .

<http://polleres.net#me> <http://xmlns.com/foaf/0.1/knows> <http://danbri.org/foaf.rdf#danbri>.

<http://polleres.net#me> <http://xmlns.com/foaf/0.1/knows> <http://www.ivan-herman.net/foaf.rdf#me> .

<http://www.ivan-herman.net/foaf.rdf#me> <http://xmlns.com/foaf/0.1/name> "Ivan Herman".

<http://danbri.org/foaf.rdf#danbri> <http://xmlns.com/foaf/0.1/name> "Dan Brickley".

<http://danbri.org/foaf.rdf#danbri> <http://xmlns.com/foaf/0.1/made> <http://xmlns.com/foaf/spec/> .

<http://xmlns.com/foaf/spec/> <http://purl.org/dc/elements/1.1/title> "FOAF Vocabulary Specification" .

<http://polleres.net#me> <http://xmlns.com/foaf/0.1/knows> _:someone .

_:someone <http://xmlns.com/foaf/0.1/name> "Herbert Polleres".

https://ai.wu.ac.at/~polleres/teaching/DOSA_2014/20140526/foaf_mini.ttl

Turtle Syntax Shortcuts:

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

- Declare prefixes to use shortcuts for URIs:

```
@prefix : <http://polleres.net#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .

:me rdf:type  foaf:Person .
:me foaf:name "Axel Polleres".
:me foaf:workplaceHomepage <http://www.wu.ac.at> .
:me foaf:knows <http://danbri.org/foaf.rdf#danbri>.
:me foaf:knows <http://www.ivan-herman.net/foaf.rdf#me> .
<http://www.ivan-herman.net/foaf.rdf#me> foaf:name "Ivan Herman".
<http://danbri.org/foaf.rdf#danbri> foaf:name "Dan Brickley".
<http://danbri.org/foaf.rdf#danbri> foaf:made <http://xmlns.com/foaf/spec/> .
<http://xmlns.com/foaf/spec/> dc:title "FOAF Vocabulary Specification" .
<http://polleres.net#me> foaf:knows _:someone .
_:someone foaf:made "Herbert Polleres".
```

EFMD EQUIS ACCREDITED

Turtle Syntax Shortcuts:

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

- Declare prefixes to use shortcuts for URIs:

```
@prefix : <http://polleres.net#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .

:me     a       foaf:Person .
:me foaf:name "Axel Polleres".
:me foaf:workplaceHomepage <http://www.wu.ac.at> .
:me foaf:knows <http://danbri.org/foaf.rdf#danbri>.
:me foaf:knows <http://www.ivan-herman.net/foaf.rdf#me> .
<http://www.ivan-herman.net/foaf.rdf#me> foaf:name "Ivan Herman".
<http://danbri.org/foaf.rdf#danbri> foaf:name "Dan Brickley".
<http://danbri.org/foaf.rdf#danbri> foaf:made <http://xmlns.com/foaf/spec/> .
<http://xmlns.com/foaf/spec/> dc:title "FOAF Vocabulary Specification" .
<http://polleres.net#me> foaf:knows _:someone .
_:someone foaf:made "Herbert Polleres".
```

rdf:type is a very common predicate (isA), so it has a special abbreviation 'a' in Turtle

EFMD EQUIS ACCREDITED

Turtle Syntax Shortcuts:

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

- Triples with common subject can be grouped together using `;` and `,`

```
@prefix : <http://polleres.net#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .

:me     a       foaf:Person ;
                foaf:name "Axel Polleres" ;
                foaf:workplaceHomepage <http://www.wu.ac.at> ;
                foaf:knows <http://danbri.org/foaf.rdf#danbri> ,
                                <http://www.ivan-herman.net/foaf.rdf#me> .
<http://www.ivan-herman.net/foaf.rdf#me> foaf:name "Ivan Herman".
<http://danbri.org/foaf.rdf#danbri> foaf:name "Dan Brickley";
                                        foaf:made <http://xmlns.com/foaf/spec/> .
<http://xmlns.com/foaf/spec/> dc:title "FOAF Vocabulary Specification" .
<http://polleres.net#me> foaf:knows _:someone .
_:someone foaf:made "Herbert Polleres".
```

EFMD EQUIS ACCREDITED

Turtle Syntax Shortcuts:

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

- Blank nodes don't need labels (e.g. _:someone), but you can instead use `[`,`]`:

```
@prefix : <http://polleres.net#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .

:me     a      foaf:Person ;
               foaf:name "Axel Polleres" ;
               foaf:workplaceHomepage <http://www.wu.ac.at> ;
               foaf:knows <http://danbri.org/foaf.rdf#danbri> ,
                                 <http://www.ivan-herman.net/foaf.rdf#me> .
<http://www.ivan-herman.net/foaf.rdf#me> foaf:name "Ivan Herman".
<http://danbri.org/foaf.rdf#danbri> foaf:name "Dan Brickley";
                                          foaf:made <http://xmlns.com/foaf/spec/> .
<http://xmlns.com/foaf/spec/> dc:title "FOAF Vocabulary Specification" .
<http://polleres.net#me> foaf:knows [ foaf:name "Herbert Polleres" ].
```

EFMD EQUIS ACCREDITED

RDF Syntax checking:

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

- A valid RDF Turtle file:

https://ai.wu.ac.at/~polleres/teaching/DOSA_2014/20140526/foaf_mini.ttl

Check validity of RDF syntax using the tool **rapper** (an RDF syntax checker and converter installed on our server):

```
rapper --input turtle https://ai.wu.ac.at/~polleres/teaching/
DOSA_2014/20140526/foaf_mini.ttl
```

- Actually, this is (somewhat) an excerpt of an RDF file that describes me on my Web page in RDF/XML syntax.

https://ai.wu.ac.at/~polleres/teaching/DOSA_2014/20140526/foaf.rdf

You can convert this file to Turtle syntax using:

```
rapper --input rdfxml --output turtle http://www.polleres.net/foaf.rdf
```

EFMD EQUIS ACCREDITED

Task: Create your own FOAF RDF file:

1) Generate a skeleton using http://www.ldodds.com/foaf/foaf-a-matic (creates RDF/XML)
2) Paste it in a new file, e.g. myfoaf.rdf
3) Convert it to Turtle syntax using rapper

```
rapper --input rdfxml --output turtle myfoaf.rdf > myfoaf.ttl
```

4) Add some own RDF triples

e.g.

```
:me foaf:interest <http://dbpedia.org/resource/Skiing>.
```

Linked Data

Why is RDF important as a Data format for Web Data? The "Semantic Web" promise...

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

*"If **HTML** and the Web made all the online documents look like one huge book, **RDF**, schema and inference languages will make all the data in the world look like one huge **database**"*

Tim Berners-Lee, 1999

Try the following in google:

- "*Persons who have published books related to and also have edited W3C specifications*"

EFMD EQUIS ACCREDITED

The data is all available on the Web…

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

Einführung in den Sprachkern von SQL-99 – Wolfgang Panny, Alfred Taudes – Google Books

Einführung in den Sprachkern von SQL-99

Wolfgang Panny, Alfred Taudes

Springer DE, 2000 - Business & Economics - 462 pages

0 Reviews

Dieses Buch wendet sich an Endbenutzer und Entwickler von Informationssystemen, Studierende und alle, die eine fundierte, der aktuellen Standardgeneration entsprechende Einfuhrung in den Sprachkern von SQL benotigen. Es deckt den Sprachumfang von Core SQL vollstandig ab, in mehreren Punkten geht es sogar

More »

Preview this book »

What people are saying - Write a review

We haven't found any reviews in the usual places.

Related books

SQL. Der Standard.
Chris J. Date, Hugh Darwe

SQL:1999
Jim Melton, Alan R. Simon

PostgreSQL
Jens Hartwig

Advanced SQL, 1999
Jim Melton

Jim Melton, Oracle

XQuery 1.0 and XPath 2.0 Functions and Operators (Second Edition)

W3C Recommendation

XQuery 1.0 and XPath 2.0 Functions and Operators (Second Edition)

W3C Recommendation 14 December 2010

This version:
http://www.w3.org/TR/2010/REC-xpath-functions-20101214/
Latest version:
http://www.w3.org/TR/xpath-functions/
Previous versions:
http://www.w3.org/TR/2009/PER-xpath-functions-20090421/, http://www.w3.org/TR/2007/REC-xpath-functions-20070123/
Editors:
Ashok Malhotra, Oracle Corporation <ashok.malhotra@alum.mit.edu>
Jim Melton, Oracle Corporation <jim.melton@oracle.com>
Norman Walsh, Mark Logic <Norman.Walsh@marklogic.com>
Michael Kay, Saxonica <http://www.saxonica.com/> - Second Edition

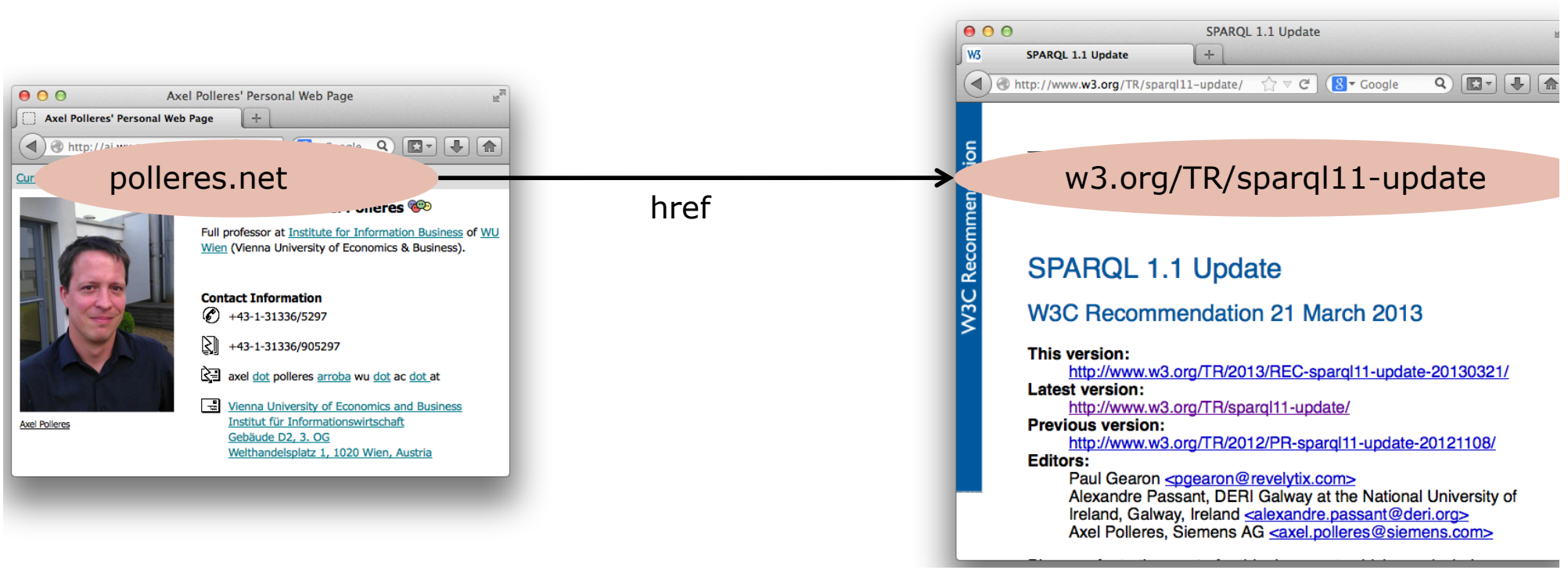
From the HTML Web…

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

- Globally Unique identifiers
- Links between **Documents** (href)
- A common protocol

URIs

HTTP



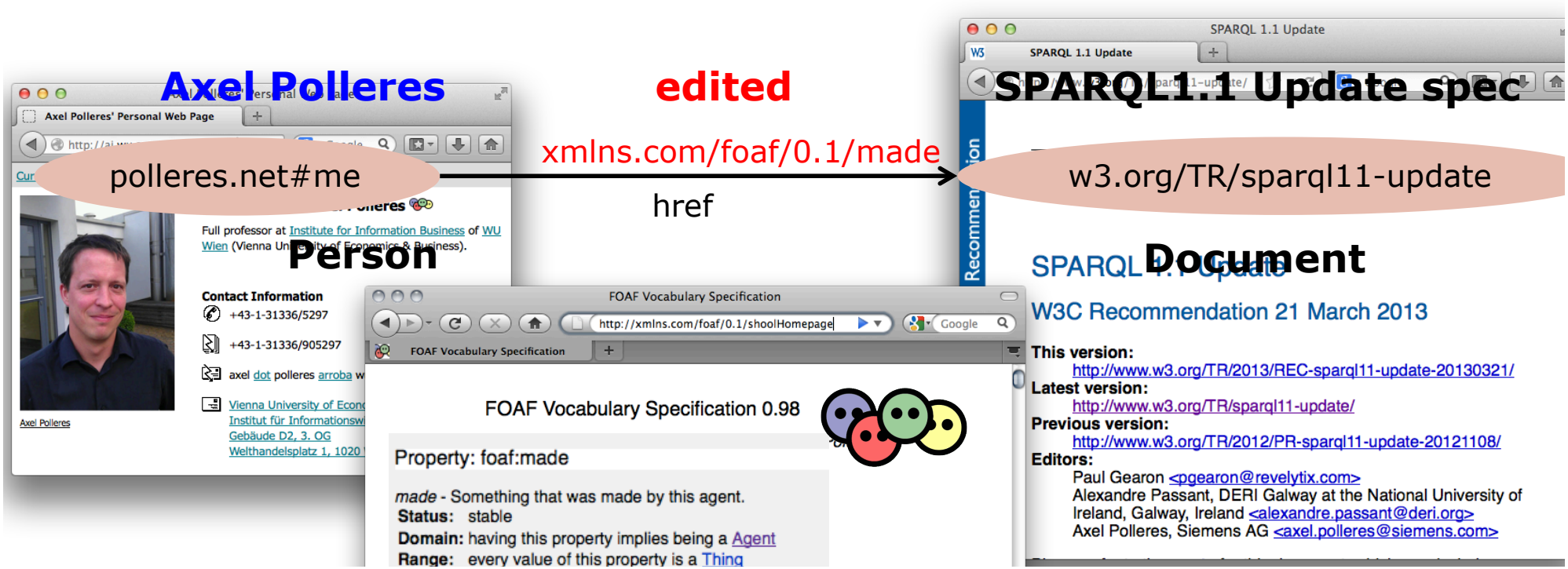
… towards a Web of (Linked) Data

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

- Globally Unique identifiers
- Typed Links (=relations) between Entities
- A common protocol

URIs
RDF
HTTP

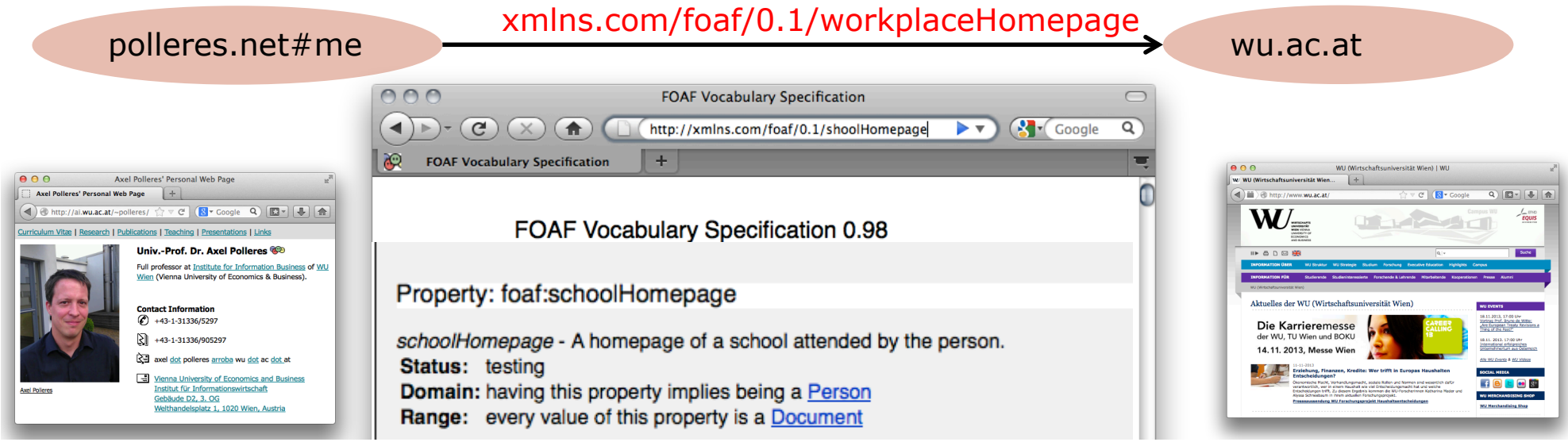
→ *RDF = a universal graph-based data format for the Web…*



Linked Data Principles

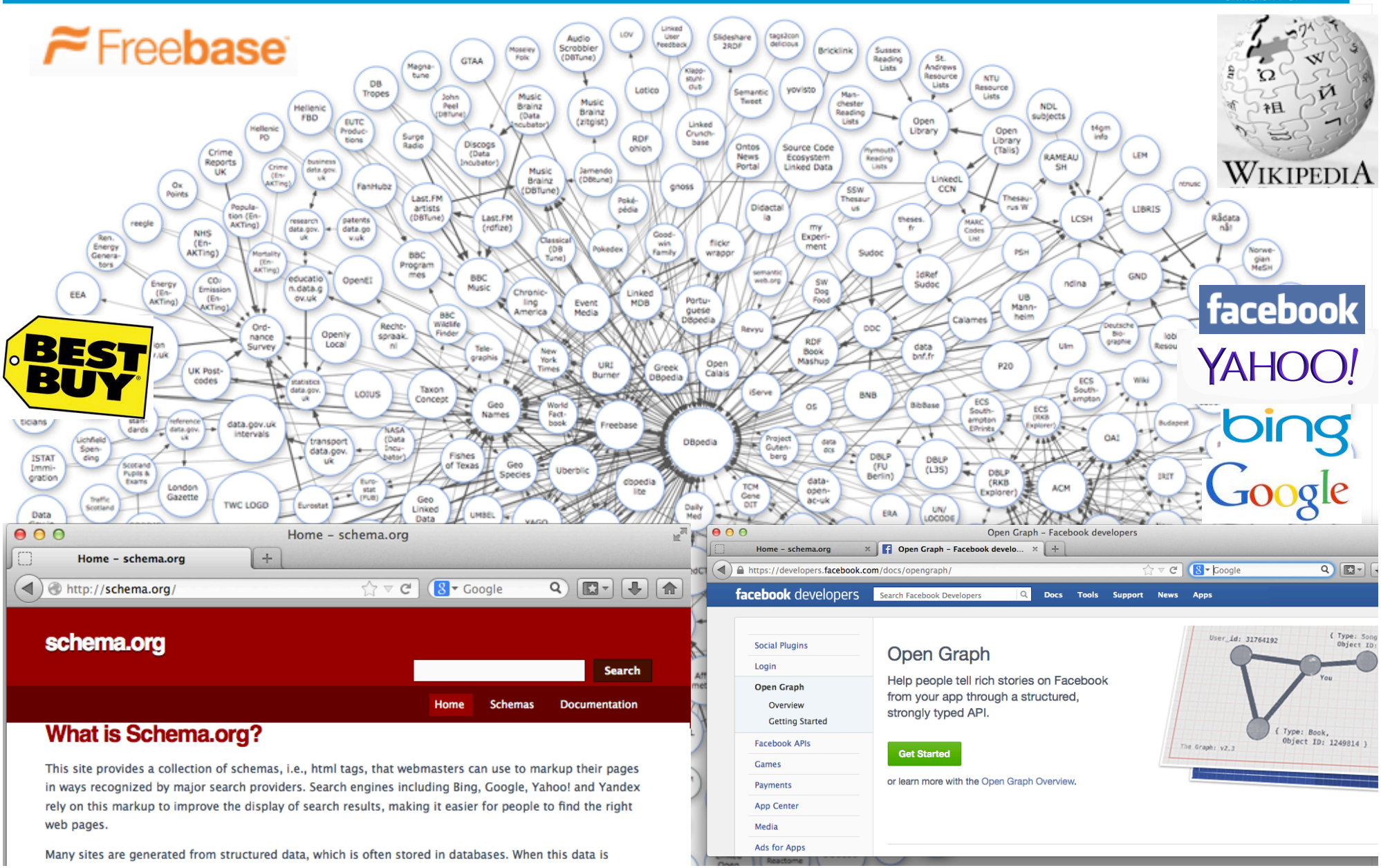
WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

1. Everything gets a URI (papers, people, talks, organizations, topics…)
2. These URIs are linked via RDF describing relations
3. Relations are URIs again (e.g. :name)
4. **When I dereference the URIs, I should find more information about them, defining them.**



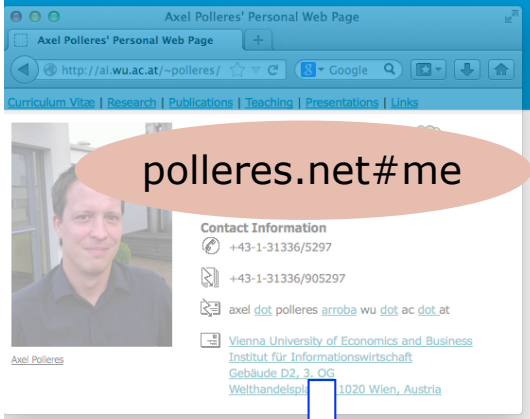
Linked Data on the Web: Adoption

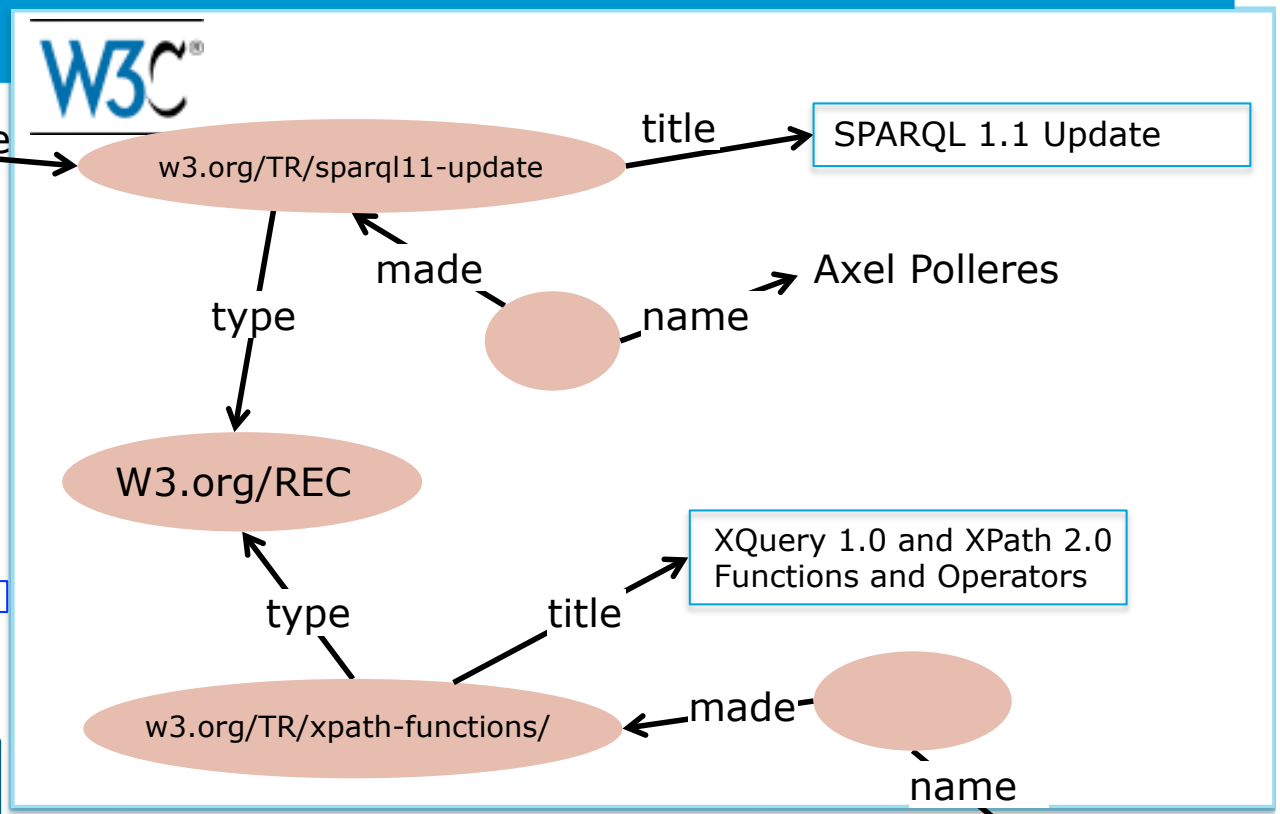
WU WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF

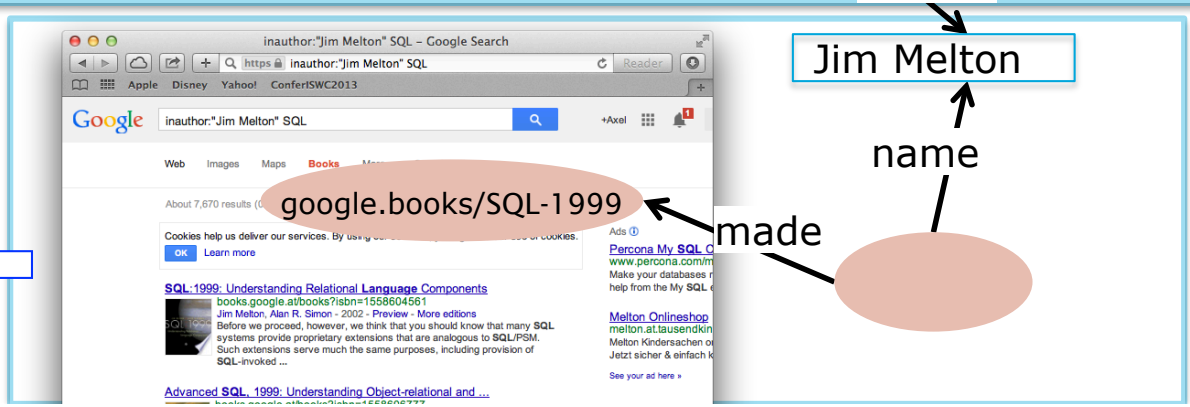


RDF + Linked Data Principles allow standard access information on the Web

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS





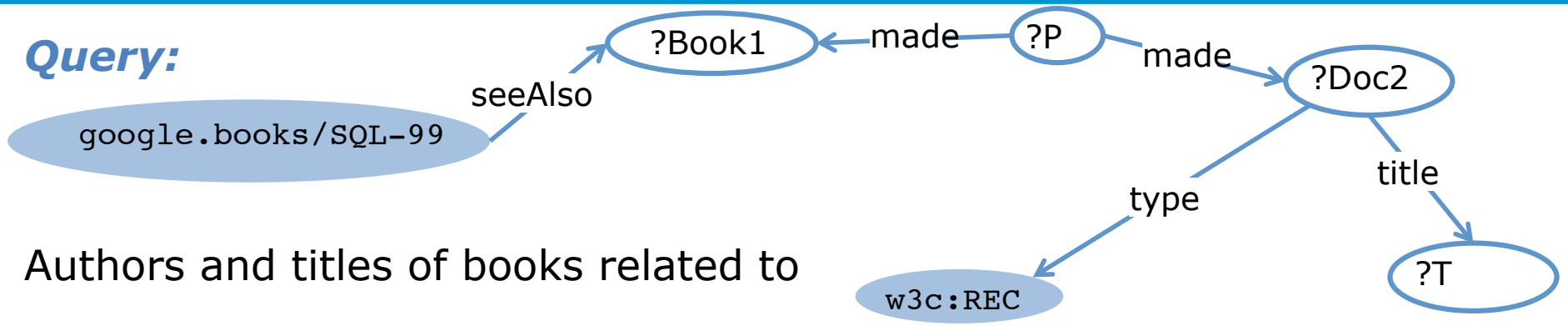


RDF Store

Subject	Predicate	Object
_:b1	foaf:made	w3.org/TR/xpath-functions/
_:b1	foaf:name	"Jim Melton"
google.books/ Jim Melton	foaf:name	"Jim Melton
google.books/ Jim Melton	foaf:made	google.books/ SQL-1999
google.books/ SQL-99	rdfs:seeAlso	google.books/ SQL-1999

SPARQL provides a standard query language

Query:



Authors and titles of books related to

RDF Store

Subject	Predicate	Object
W3c:editor1	foaf:made	w3.org/TR/xpath-functions/
W3c:editor1	foaf:name	"Jim Melton"
google.books/ Jim Melton	foaf:name	"Jim Melton
google.books/ Jim Melton	foaf:made	google.books/ SQL-1999
google.books/ SQL-99	rdfs:seeAlso	google.books/ SQL-1999

```
PREFIX g: <google.books/>
...
SELECT ?P ?T WHERE
{g:SQL-99 rdfs:seeAlso ?Book1 .
  ?P foaf:made ?Book1 .
  ?P foaf:made ?Doc2 .
  ?Doc2 rdf:type w3c:REC ;
        cd:title ?T . }
```

SPARQL = *SQL look-and-feel for Linked Data*

Could be done in SQL, couldn't it?

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

- Yes, it is known that SPARQL 1.0 has the same expressivity as Relational Algebra, i.e. within SQL-99)
- But: … syntax gets quite blown up

  … lots of self-joins…
  - encodings into SQL usually use different schemata e.g. property tables.
  - Specialized triple stores perform better than encodings on top of SQL DBs in practice
- Plus: peculiarities that make this encoding non-trivial
  - e.g. the semantics of outer-joins in SPARQL (OPTIONAL)
  - cf. SPARQL1.1 Tutorial at http://polleres.net/presentations/20101019SPARQL1.1Tutorial.pptx

triples

Subject	Predicate	Object
W3c:editor1	foaf:made	w3.org/TR/xpath-functions/
W3c:editor1	foaf:name	"Jim Melton"
google.books/ Jim Melton	foaf:name	"Jim Melton
google.books/ Jim Melton	foaf:made	google.books/ SQL-1999
google.books/ SQL-99	rdfs:seeAlso	google.books/ SQL-1999

```
SELECT T2.Subject T5.Object
FROM triples T1, triples T2, triples
T3, triples T4, triples T5
WHERE
   T1.Subject = g:SQL-99 AND
   T1.Predicate = rdfs:seeAlso AND
   T1.Object = T2.Object AND
   T2.Subject = T3.Subject AND
  T2.Predicate = foaf:made
   …
```

SPARQL step-by-step

SPARQL by examples *Step by step*:

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

- An folgendem Link finden Sie "step-by-step" Beispiele die RDF & SPARQL noch einmal im Detail erklaeren:

https://ai.wu.ac.at/~polleres/teaching/DOSA_2014/20140526/RDF_SPARQL_step-by-step/

- Aufruf des ersten Beispiels mittels:

```
arq –query query1.rq –data simple1.ttl
```

EFMD EQUIS ACCREDITED

SPARQL: more examples on how to query Wikipedia!

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

Schauen wir uns nun SPARQL-Anfragen an, die man mit Daten von einem *remote* Server, beantworten kann:

- https://ai.wu.ac.at/~polleres/teaching/DOSA_2014/20140526/RDF_SPARQL_dbpedia/

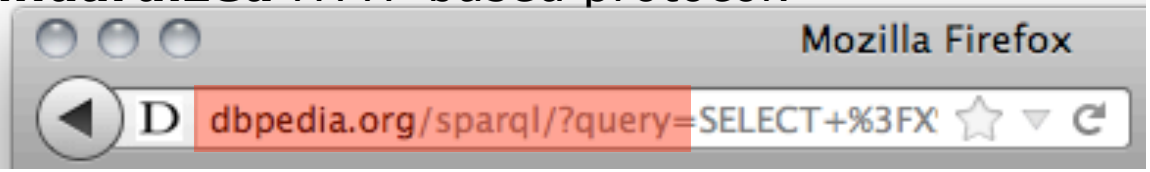
EFMD EQUIS ACCREDITED

SPARQL – Standard RDF Query Language and Protocol

- SPARQL :

```
SELECT ?X
WHERE
{
 ?X  <http://dbpedia.org/property/origin> <http://dbpedia.org/resource/Vienna>
}
```

- SQL "Look-and-feel" for the Web
- Essentially "graph matching" by *triple patterns*
- Allows conjunction (.) , disjunction (UNION), optional (OPTIONAL) patterns and filters (FILTER)
- Construct new RDF from existing RDF
- Solution modifiers (DISTINCT, ORDER BY, LIMIT, …)
- A **standardized** HTTP based protocol:



Querying RDF with SPARQL by example:

- Basic Graph Pattern = Conjunction of triple patterns („Turtle RDF Graphs with **variables**" (denoted by ‚?'):

Examples:

*„All people I know from my FOAF file **and** their names"*

```
PREFIX : <http://www.polleres.net/foaf.rdf#me>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
SELECT ?P ?N
WHERE
{ :me foaf:knows ?P .
   ?P foaf:name ?N . }
```

Try it yourself:

```
arq --query https://ai.wu.ac.at/~polleres/teaching/DOSA_2014/20140526/query1.rq --
data http://polleres.net/foaf.rdf
```

Querying RDF with SPARQL by example:

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

- Basic Graph Pattern = „Turtle RDF Graphs with **variables**" (denoted by ‚?'):

Examples:

*„All people I know from my FOAF file **and** their names"*

```
PREFIX : <http://www.polleres.net/foaf.rdf#me>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
SELECT ?P ?N
WHERE
{ :me foaf:knows ?P .
   ?P foaf:name ?N . }
```

Try it yourself:

```
arq --query https://ai.wu.ac.at/~polleres/teaching/DOSA_2014/20140526/query1.rq --
data http://polleres.net/foaf.rdf
```

EFMD EQUIS ACCREDITED

Conjunction (.) , disjunction (UNION), optional (OPTIONAL) patterns and filters (FILTER)

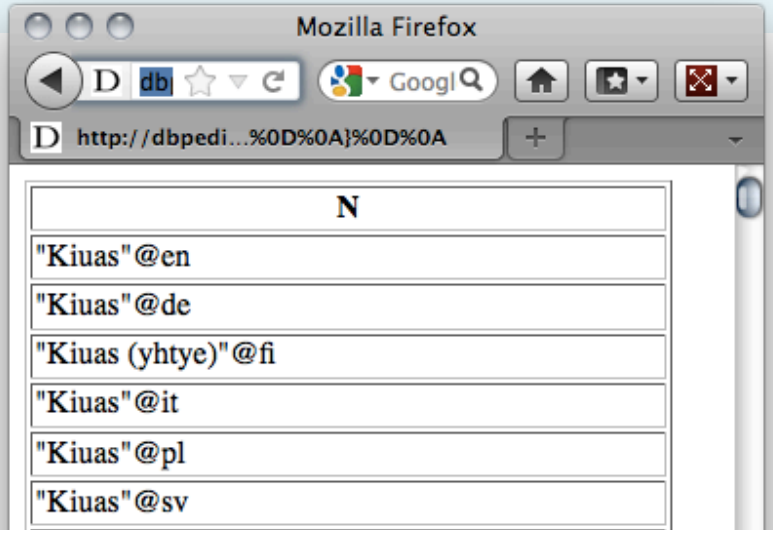
WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

***Names** of **bands** from **cities in Finland**?*

```
PREFIX : <http://dbpedia.org/resource/>
PREFIX dbprop: <http://dbpedia.org/property/>
PREFIX dbont: <http://dbpedia.org/ontology/>
PREFIX category: <http://dbpedia.org/resource/Category:>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dcterms: <http://purl.org/dc/terms/>

SELECT ?N
WHERE
{
 ?X a dbont:Band ; rdfs:label ?N ;
    dbprop:origin [ dcterms:subject category:Cities_and_towns_in_Finland] .
}
```

- *Turtle-like shortcuts can be used for namespace prefixes and to group triple patterns.*



Conjunction (.) , **disjunction (UNION)**, optional (OPTIONAL) patterns and filters (FILTER)

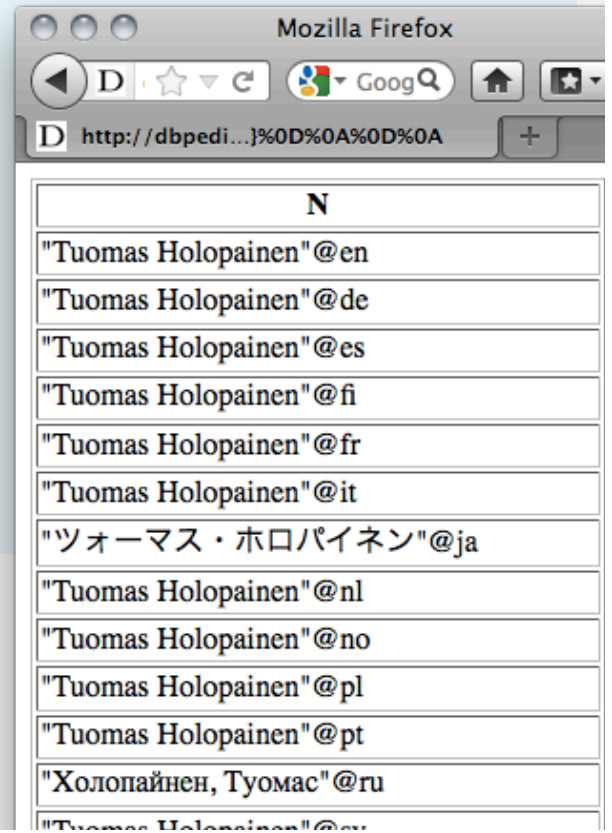
WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

*Names of things **that origin or were born in** Kitee?*

```
PREFIX : <http://dbpedia.org/resource/>
PREFIX dbprop: <http://dbpedia.org/property/>
PREFIX dbont: <http://dbpedia.org/ontology/>
PREFIX category: <http://dbpedia.org/resource/Category:>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dcterms: <http://purl.org/dc/terms/>

SELECT ?N
WHERE
{
 { ?X  dbprop:origin <http://dbpedia.org/resource/Kitee> }
 UNION
 { ?X  dbont:birthPlace <http://dbpedia.org/resource/Kitee> }

?X rdfs:label ?N
}
```



N
"Tuomas Holopainen"@en
"Tuomas Holopainen"@de
"Tuomas Holopainen"@es
"Tuomas Holopainen"@fi
"Tuomas Holopainen"@fr
"Tuomas Holopainen"@it
"ツォーマス・ホロパイネン"@ja
"Tuomas Holopainen"@nl
"Tuomas Holopainen"@no
"Tuomas Holopainen"@pl
"Tuomas Holopainen"@pt
"Холопайнен, Туомас"@ru

Conjunction (.) , disjunction (UNION), optional (OPTIONAL) patterns and **filters (FILTER)**

Cites Finland with a name starting with "H"?

```
PREFIX : <http://dbpedia.org/resource/>
PREFIX category: <http://dbpedia.org/resource/Category:>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dcterms: <http://purl.org/dc/terms/>

SELECT ?C ?N
WHERE
{
 ?C  dcterms:subject category:Cities_and_towns_in_Finland ;
     rdfs:label ?N .
 FILTER( strstarts(str(?N),"H"))
}
```

SPARQL has lots of FILTER functions to filter text with regular expressions (REGEX), filter numerics (<,>,=,+,-…), dates, etc.)

List of SPARQL functions http://www.w3.org/TR/sparql11-query/#expressions

Conjunction (.) , disjunction (UNION), **optional (OPTIONAL)** patterns and filters (FILTER)

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

*Cites in Finland with **optionally** their website*

```
PREFIX : <http://dbpedia.org/resource/>
PREFIX dbprop: <http://dbpedia.org/property/>
PREFIX dbont: <http://dbpedia.org/ontology/>
PREFIX category: <http://dbpedia.org/resource/Category:>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dcterms: <http://purl.org/dc/terms/>

SELECT ?C ?W
WHERE
{
 ?C  dcterms:subject category:Cities_and_towns_in_Finland .
 OPTIONAL { ?C dbprop:website ?W .  }
}
```

Note: variables can be **unbound** in a result!

C	W
http://dbpedia.org/resource/Kankaanp%C3%A4%C3%A4	http://w…kankaanpaa.fi/
http://dbpedia.org/resource/Kemij%C3%A4rvi	http://ww…emijarvi.fi/
http://dbpedia.org/resource/Pudasj%C3%A4rvi	http://www…dasjarvi.fi/
http://dbpedia.org/resource/Pieks%C3%A4m%C3%A4ki	http://www.p…ksamaki.fi/
http://dbpedia.org/resource/Varkaus	http://www.va…kaus.fi/
http://dbpedia.org/resource/Kask%C3%B6	
http://dbpedia.org/resource/Brahestad	
http://dbpedia.org/resource/Hang%C3%B6	
http://dbpedia.org/resource/Torne%C3%A5	
http://dbpedia.org/resource/Largest_metropolitan_areas_in_the_Nordic_countries	
http://dbpedia.org/resource/List_of_cities_and_towns_in_Finland	
http://dbpedia.org/resource/Fredrikshamn	
http://dbpedia.org/resource/Tampere	
http://dbpedia.org/resource/N%C3%A5dendal	
http://dbpedia.org/resource/Ule%C3%A5borg	
http://dbpedia.org/resource/Karleby	

Querying RDF with SPARQL by example

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

- Advanced features:
  - NOT EXISTS
  - Assignments
  - Aggregates
  - Property Paths

EFMD EQUIS ACCREDITED

Negation: NOT EXISTS

Select Persons without a homepage:

```
SELECT ?X
WHERE{ ?X rdf:type foaf:Person
          FILTER ( NOT EXISTS { ?X foaf:homepage ?H } ) }
```

- ***SPARQL1.1*** *can do NOT EXISTS in FILTERs*
  - *detect non-existence (similar to NOT EXISTS in SQL)*

Assignment

- Similar to SQL – keyword 'AS' is allowed to assign the value of an expression to a variable:

```
SELECT ?X ?L ?A ?P (?P/?A AS ?PD )
WHERE {  ?X a <http://dbpedia.org/ontology/Country> ;
            dcterms:subject
      <http://dbpedia.org/resource/
Category:Member_states_of_the_European_Union>;
            rdfs:label ?L  ;
            <http://dbpedia.org/property/
populationCensus> ?P;
            <http://dbpedia.org/property/areaKm> ?A .
         FILTER(lang(?L) = "en" && isNumeric(?A) &&
isNumeric(?P) )
      }
```

Aggregates also similar to SQL:

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

- Count the number of EU Countries and:

```
SELECT (Count(?X) AS ?EU_Countries)
WHERE {  ?X a <http://dbpedia.org/ontology/Country> ;
            dcterms:subject
<http://dbpedia.org/resource/Category:Member_states_of_the_European_Union> }
```

- Why does this one give a different result?

```
SELECT (Count(?X) AS ?EU_Countries) (sum(?P) as ?Total_Pop)
WHERE {  ?X a <http://dbpedia.org/ontology/Country> ;
            dcterms:subject
      <http://dbpedia.org/resource/Category:Member_states_of_the_European_Union> ;
      <http://dbpedia.org/property/populationCensus> ?P }
```

- ATTENTION: Not all dbpedia entities have the same attributes! This is (mostly) due to errors/ommissions in wikipedia...
- That‘s why not all countries had a population density on our map!

EFMD EQUIS ACCREDITED

Another nice feature: Path queries

- Father of Maria Theresia (on DBpedia):

```
PREFIX : <http://dbpedia.org/resource/>
PREFIX dbp: <http://dbpedia.org/property/>
SELECT ?X
WHERE {  :Maria_Theresa dbp:father  ?X }
```

- Mother of Maria Theresia (on DBpedia):

```
SELECT ?X
WHERE {  :Maria_Theresa dbp:mother  ?X }
```

- Ancestors of Maria Theresia (on DBpedia):

```
SELECT ?X
WHERE {  :Maria_Theresa (dbp:mother|dbp:father)*          ?X }
```

See also: http://en.wikipedia.org/wiki/Maria_Theresa#Ancestry

More details

- This should be enough to get you started to write some of your own queries...

- For more details on SPARQL, see e.g. these tutorials:
- http://www.cambridgesemantics.com/semantic-university/sparql-by-example
- http://ai.wu.ac.at/~polleres/presentations/20101019SPARQL1.1Tutorial.pptx

Homework

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

1) Extend your Web interface with:
   - At least one form to insert, delete, or update data.
   - At least one report that displays at least one diagram

Abgabe: bis 6.6.2014
(naechste Woche ist keine VO!)

2) Familiarize yourself with SPARQL: go through the examples...

   Come up with a few own interesting queries on DBPedia or another public SPARQL endpoint:
   - Use the SPARQL endpoint at
     - http://live.dbpedia.org/sparql
   - Other Open RDF Data SPARQL endpoints you could use:
     - http://open-data.europa.eu/en/linked-data ... Linked Data from the European Open Data Portal
     - http://worldbank.270a.info/sparql ... Linked data from WorldBank
   - Even if you don't manage to query the data you wanted: Formulate the query you intend in natural language (e.g. "*Football players born in Vienna after 1950*") and we will try to work towards the solution together next time.

3) Start thinking about the final project:
   - You should present a full Web application with a database in the back, including:
   - You should be able to present us the underlying Tables and ER diagramm
   - The Web interface should allow to manipulate Data
   - Your Database should use some external Data (imported from CSV, JSON, or from a SPARQL query
   - Your Web Interface should present some report including some visual analystics (diagrams)
- Be creative!
- In der naechsten Stundenwiederholung sollten Sie zeigen koennen:
   - Eine "draft"-Version Ihres Abschluss-Projekts, incl. Eingabemasken und Diagramm.
   - Ein eigenes validierendes RDF file
   - SPARQL-Abfragen mit den in der heutigen Stunde präsentierten features (FILTER, UNION, OPTIONAL, aggregates)

EFMD EQUIS ACCREDITED