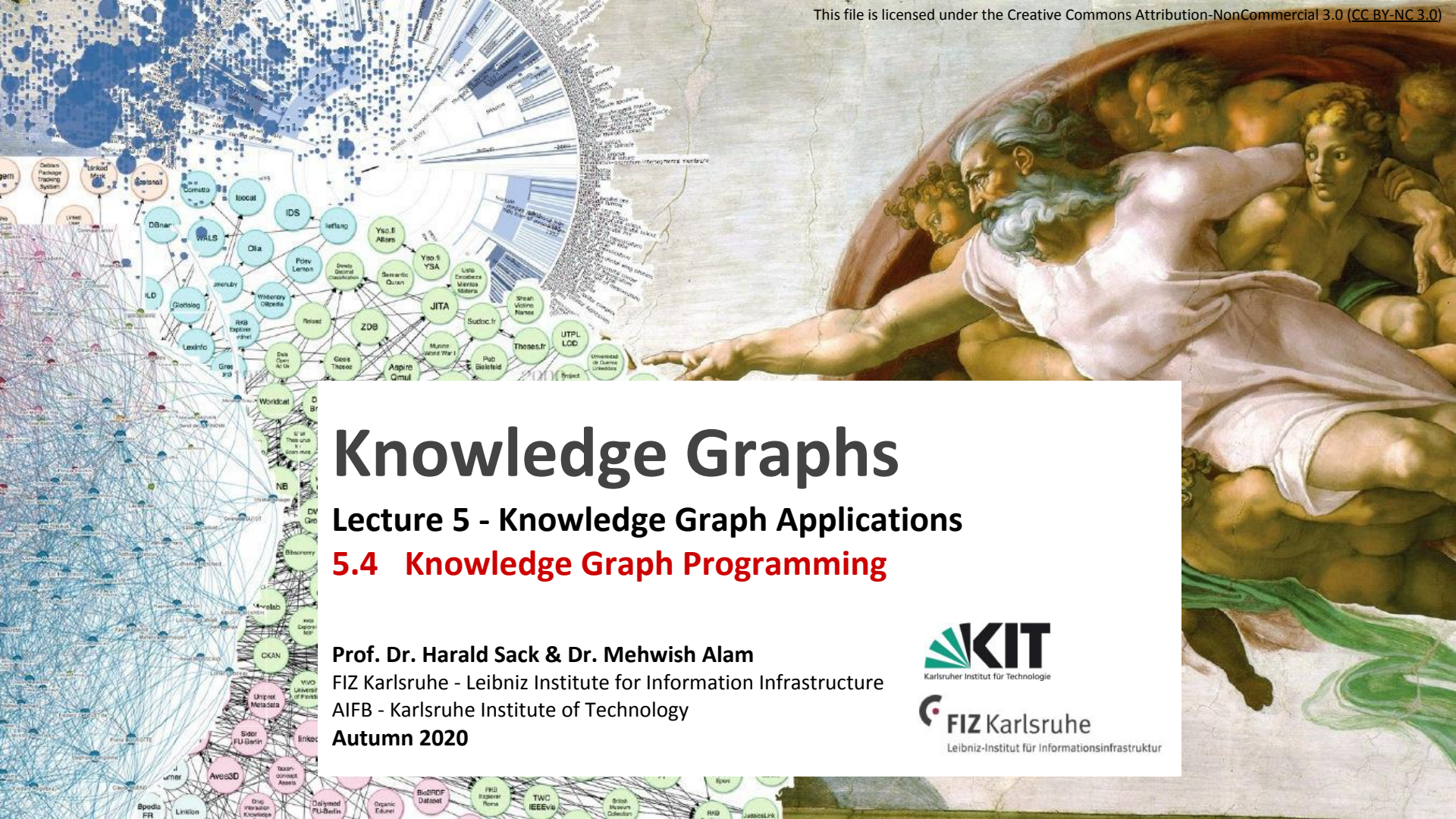


Knowledge Graphs

Lecture 5 - Knowledge Graph Applications

5.4 Knowledge Graph Programming

Prof. Dr. Harald Sack & Dr. Mehwish Alam
FIZ Karlsruhe - Leibniz Institute for Information Infrastructure
AIFB - Karlsruhe Institute of Technology
Autumn 2020

Karlsruher Institut für Technologie

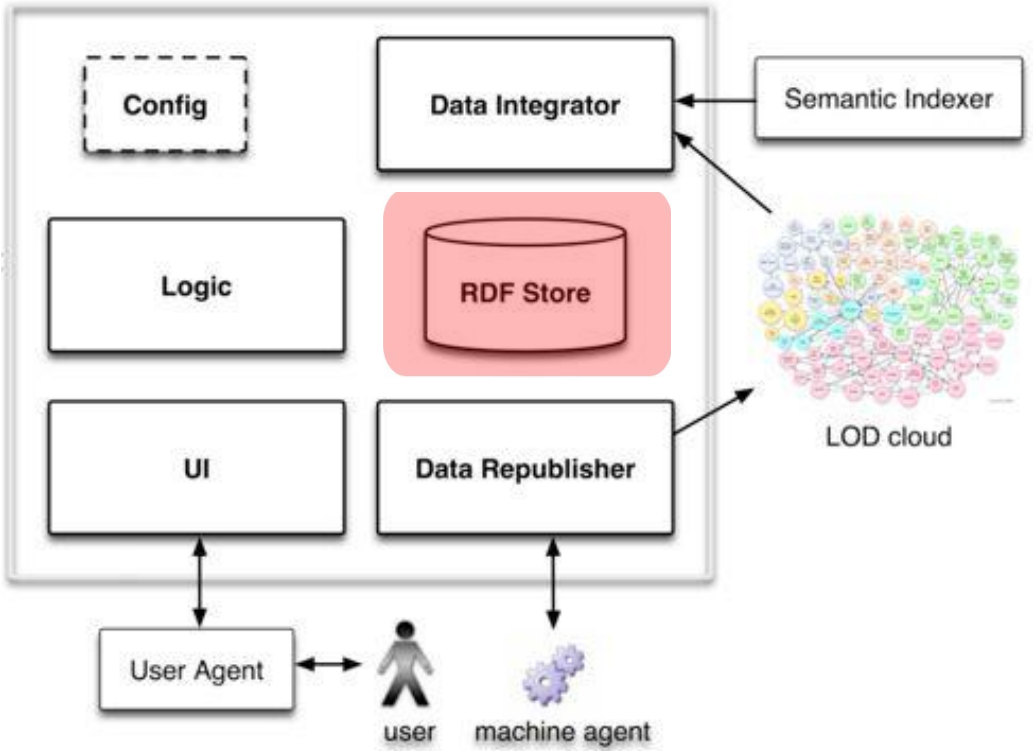
FIZ Karlsruhe
Leibniz-Institut für Informationsinfrastruktur

Knowledge Graphs

Lecture 5: Knowledge Graph Applications

Knowledge Graph Driven (Web) Applications

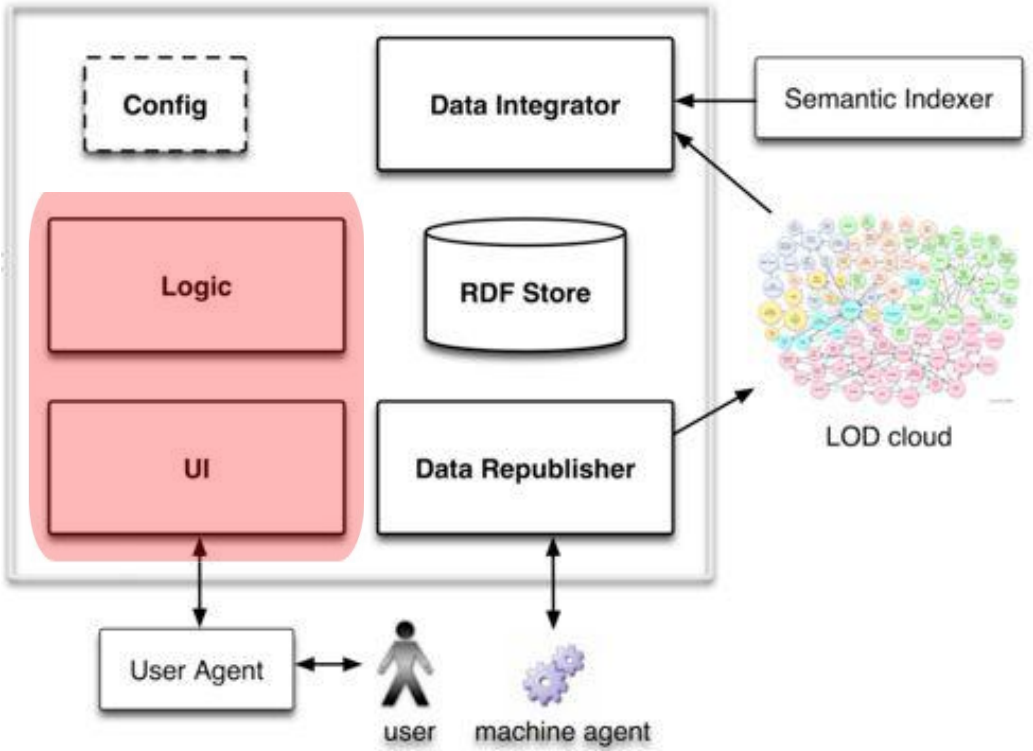
- Required Components:
  - **Local RDF Store**
    - caching of results
    - permanent storage



M.Hausenblas: Linked Data Applications, DERI Technical Report, 2009

Knowledge Graph Driven (Web) Applications

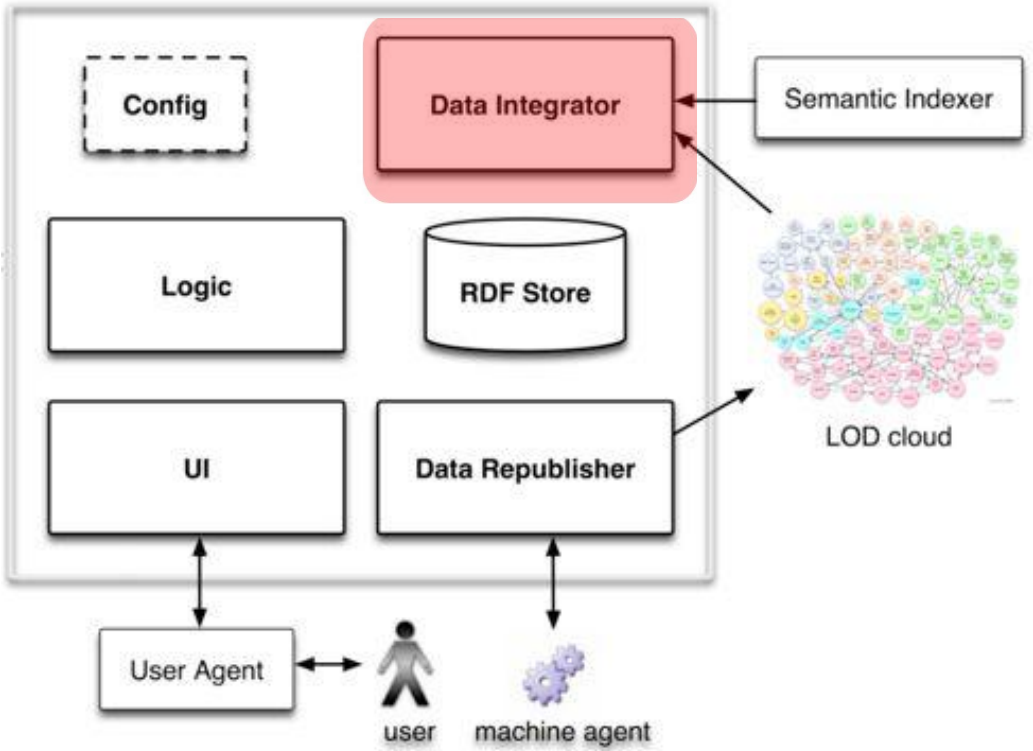
- Required Components:
  - **Logic** (Controller) and
  - **User Interface**
    - (=Business Logic)
    - (not KG specific)



M.Hausenblas: Linked Data Applications, DERI Technical Report, 2009

Knowledge Graph Driven (Web) Applications

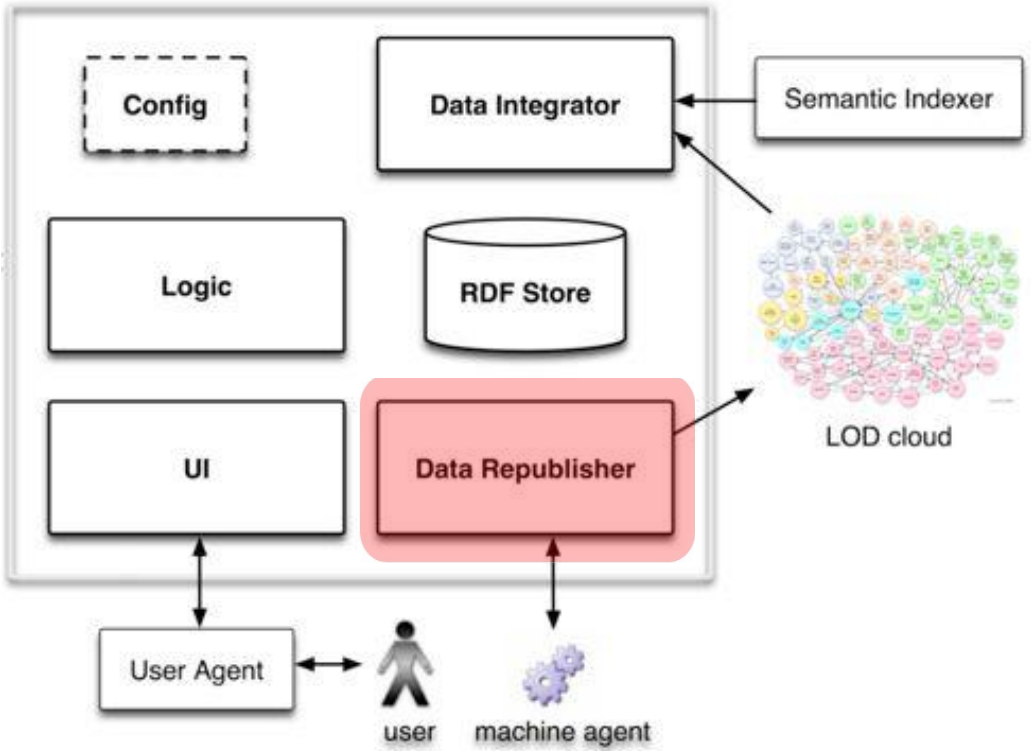
- Required Components:
- **Data Integration component**
  - get data directly from Knowledge Graph or
  - via Semantic Indexer



M.Hausenblas: Linked Data Applications, DERI Technical Report, 2009

Knowledge Graph Driven (Web) Applications

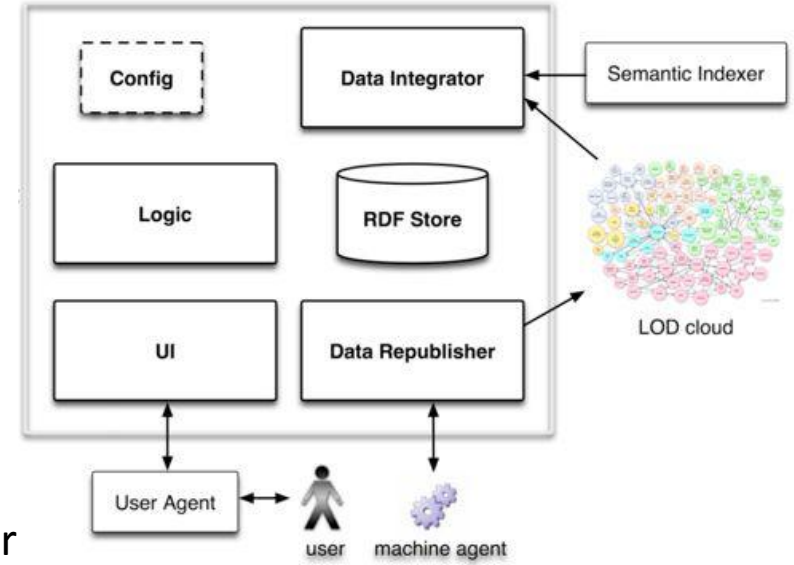
- Required Components:
- **Data Re-/Publishing component**
  - write back application dependent data into the Web of Data



M.Hausenblas: Linked Data Applications, DERI Technical Report, 2009

Knowledge Graph Driven (Web) Applications

- Required Components:
  - **Local RDF Store**
    - caching of results
    - permanent storage
  - **Logic** (Controller) and
  - **User Interface** (=Business Logic)
    - (not KG specific)
  - **Data Integration component**
    - get data directly from Web of Data or
    - via Semantic Indexer
  - **Data Re-/Publishing component**
    - write back application dependent data into the Web of Data



M.Hausenblas: Linked Data Applications, DERI Technical Report, 2009

Knowledge Graph Driven (Web) Applications

- The easiest way is to make use of a suitable library:
  - SPARQL Javascript Library
    http://www.thefigtrees.net/lee/blog/2006/04/sparql_calendar_demo_a_sparql.html
  - ARC for SPARQL (PHP)
    https://github.com/semsol/arc2/wiki
  - dotNetRDF (C#)
    https://dotnetrdf.github.io/
  - Jena/ARQ (Java)
    http://jena.apache.org/
  - Sesame (Java)
    http://rdf4j.org/
  - **RDFLib / SPARQL Wrapper (Python)**
    http://rdflib.github.io/sparqlwrapper/

Knowledge Graph Driven (Web) Applications

- The easiest way is to make use of a suitable library:
  - **SPARQL Wrapper (Python)**
    http://rdflib.github.io/sparqlwrapper/

- Access to Linked Data via **SPARQL endpoints**
  - Let‘s choose **DBpedia** (just for simplicity...)
    http://dbpedia.org/sparql

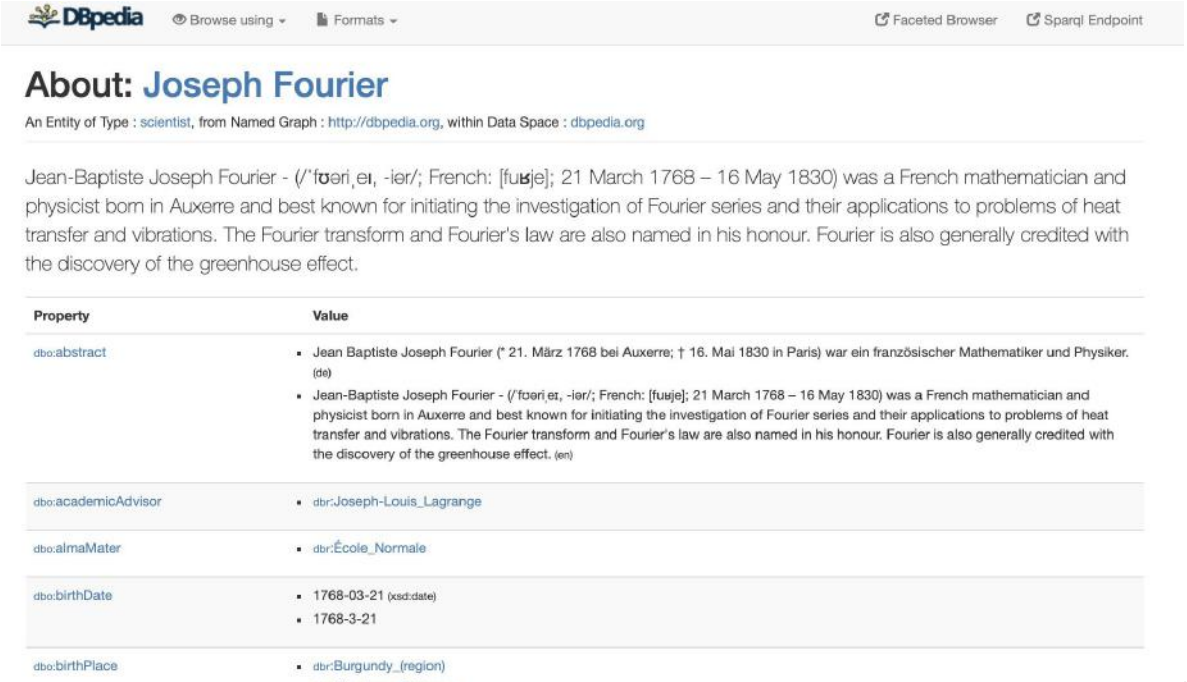
- ...now we have to think of a simple example...

Knowledge Graph Programming Example

- **Example Application:**
  - Build a simple application that looks for famous people having their birthdays **today(!)**, as e.g. **scientists**, based on **DBpedia**
  - Create a **list of scientists**, whose birthday is today including some **additional information**, as e.g.
    - Year of Birth
    - Short description, picture (if available), etc.
  - Let‘s create a simple (local) **web page** for the task (i.e. encode results in HTML), which can be displayed in the browser
  - We use **Python** and the **SPARQL Wrapper library**

Prerequisites - (Manual) Data Analysis

- Choose a representative example:
  - E.g. **Joseph Fourier** from **DBpedia**



DBpedia Browse using Formats Faceted Browser Sparql Endpoint

About: Joseph Fourier

An Entity of Type : scientist, from Named Graph : http://dbpedia.org, within Data Space : dbpedia.org

Jean-Baptiste Joseph Fourier - (/ˈfʊəriˌeɪ, -iər/; French: [fuʁje]; 21 March 1768 – 16 May 1830) was a French mathematician and physicist born in Auxerre and best known for initiating the investigation of Fourier series and their applications to problems of heat transfer and vibrations. The Fourier transform and Fourier's law are also named in his honour. Fourier is also generally credited with the discovery of the greenhouse effect.

Property	Value
dbo:abstract	▪ Jean Baptiste Joseph Fourier (* 21. März 1768 bei Auxerre; † 16. Mai 1830 in Paris) war ein französischer Mathematiker und Physiker. (de) ▪ Jean-Baptiste Joseph Fourier - (/ˈfʊəriˌeɪ, -iər/; French: [fuʁje]; 21 March 1768 – 16 May 1830) was a French mathematician and physicist born in Auxerre and best known for initiating the investigation of Fourier series and their applications to problems of heat transfer and vibrations. The Fourier transform and Fourier's law are also named in his honour. Fourier is also generally credited with the discovery of the greenhouse effect. (en)
dbo:academicAdvisor	▪ dbr:Joseph-Louis_Lagrange
dbo:almaMater	▪ dbr:École_Normale
dbo:birthDate	▪ 1768-03-21 (xsd:date) ▪ 1768-3-21
dbo:birthPlace	▪ dbr:Burgundy_(region) ▪ dbr:Kingdom_of_France

http://dbpedia.org/page/Joseph_Fourier

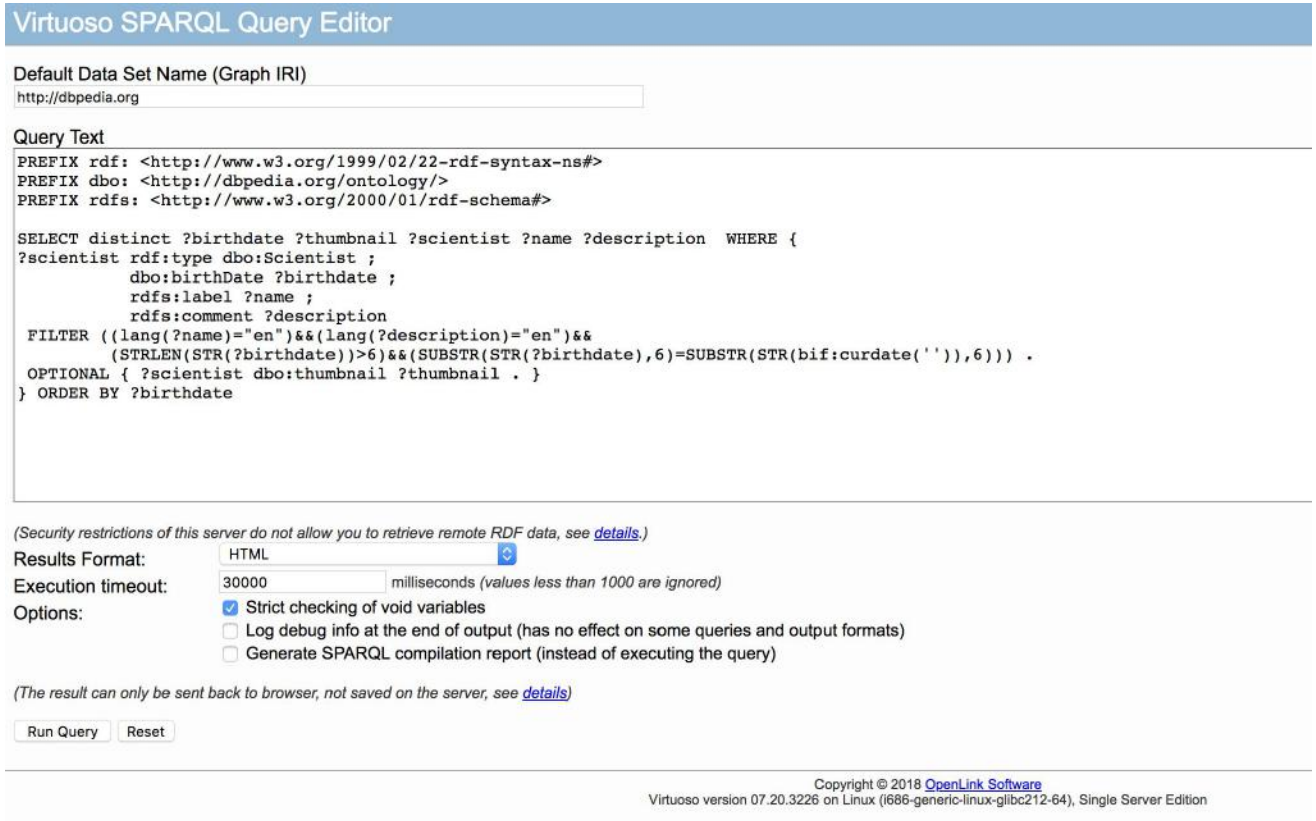
Prerequisites - (Manual) Data Analysis

- Data Analysis via **SPARQL**
  - What kind of entities are you looking for?
    - `?scientist rdf:type dbo:Scientist .`
  - What information do you need?
    - `?scientist dbo:birthDate ?birthdate .`
    - `?scientist rdfs:label ?name .`
    - `?scientist rdfs:comment ?description .`
    - `OPTIONAL { ?scientist dbo:thumbnail ?thumbnail }`
  - Any filter criteria?
    - `(lang(?name)="en") && (lang(?description)="en")`
    - `(SUBSTR(STR(?birthdate),6)="18-03")`
    - More sophisticated: `(SUBSTR(STR(bif:curdate('')),6)`

Virtuoso triple store builtin function for current date

[4]

Prerequisites - SPARQL Queries



SPARQL query at dbpedia.org

[4]

Python Code in a Collaborative Notebook

Karlsruher Institut für Technologie

FIZ Karlsruhe
Leibniz Institute for Information Infrastructure

[Link to collaborative Notebook]

Scientific Birthdays Example

This is the python notebook example for lecture 3.4 Knowledge Graph Programming, of the OpenHPI lecture "Knowledge Graphs 2020".

Please make a copy of this notebook to try out your own adaptions via "File -> Save Copy in Drive"

First, we have to install the **sparqlwrapper library** before we can use it with the notebook.

```
[ ] !pip install -q sparqlwrapper    #install SPARQLwrapper
```

We are going to use a few libraries:

- **datetime** for date formatting and interpretation
- **SPARQLWrapper** to execute SPARQLE queries and to import the results into python

Thus, we will import them now.

```
[ ] from datetime import datetime
    from SPARQLWrapper import SPARQLWrapper, JSON, XML, N3, RDF
```

We will use DBpedia (http://dbpedia.org/sparql) as our SPARQL endpoint

```
[ ] sparql = SPARQLWrapper("http://dbpedia.org/sparql") #determine SPARQL endpoint
```

Next comes the query example from the lecture and its execution

```
[ ] #SPARQL query to be executed
    sparql.setQuery("""
    PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
    PREFIX dbo: <http://dbpedia.org/ontology/>
    PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
    PREFIX dc:  <http://purl.org/dc/elements/1.1/>

    Select distinct ?birthdate ?thumbnail ?scientist ?name ?description  WHERE {
    ?scientist rdf:type dbo:Scientist ;
           dbo:birthDate ?birthdate ;
           rdfs:label ?name ;
           rdfs:comment ?description
     FILTER ((lang(?name)="en")&&(lang(?description)="en")&&(STRLEN(STR(?birthdate))>6)&&(SUBSTR(STR(?birthdate),6)=SUBSTR(STR(bif:curdate('')),6))) .
     OPTIONAL { ?scientist dbo:thumbnail ?thumbnail . }
    } ORDER BY ?birthdate
    """)

    sparql.setReturnFormat(JSON)   # Return format is JSON
    results = sparql.query().convert()   # execute SPARQL query and write result to "results"
```

Knowledge Graph Programming Example

Scientific Birthdays of Thursday January 30

1720 

Charles De Geer, Baron Charles de Geer (the family is usually known as De Geer with a capitalized "De"; Finspång in Risinge 30 January 1720 – Stockholm 7 March 1778) was a Swedish industrialist and entomologist.

1883 ?

Andrew Robertson (engineer), He was the son of a marine engineer. He was a graduate of the University of Manchester, with a first-class honours degree

1898 ?

George F C Griss, George François Cornelis Griss (30 January 1898, Amsterdam – 2 August 1953, Blaricum), usually cited as G. F. C. Griss, was a Dutch mathematician and philosopher, who was occupied with hegelian idealism and Brouwers intuitionism and stated a negationless mathematics.

1898 ?

Russell Ohl, Russell Shoemaker Ohl (January 30, 1898 – March 20, 1987) was an American engineer who is generally recognized for patenting the modern solar cell (US Patent 2402662, "Light sensitive device").Ohl was a notable semiconductor researcher prior to the invention of the transistor. He was also known as R.S. Ohl.

1899 

Max Theiler, Max Theiler (30 January 1899 – 11 August 1972) was a South African-American virologist and doctor. He was awarded the Nobel Prize in Physiology or Medicine in 1951 for developing a vaccine against yellow fever in 1937, becoming the first African-born Nobel laureate.

1903 ?

G. Evelyn Hutchinson, George Evelyn Hutchinson ForMemRS (January 30, 1903 – May 17, 1991), was an American ecologist sometimes described as the "father of modern ecology." He contributed for more than sixty years to the fields of limnology, systems ecology, radiation ecology, entomology, genetics, biogeochemistry, a mathematical theory of population growth, art history, philosophy, religion, and anthropology. He worked on the passage of phosphorus through lakes, the chemistry and biology of lakes, the theory of interspecific competition, and on insect taxonomy and genetics, zoo-geography and African water bugs. He is known as one of the first to combine ecology with mathematics. He became an international expert on lakes and wrote the four-volume Treatise on Limnology in 1957.

1905 

Emilio Segrè, Emilio Gino Segrè (30 January 1905 – 22 April 1989) was an Italian physicist and Nobel laureate who discovered the elements technetium and astatine, and the antiproton, a sub-atomic antiparticle, for which he was awarded the Nobel Prize in Physics in 1959.From 1943 to 1946 he worked at the Los Alamos National Laboratory as a group leader for the Manhattan Project. He found in April 1944 that Thin Man, the proposed plutonium gun-type nuclear weapon, would not work because of the presence of plutonium-240 impurities.

1911 ?

Alexander George Ogston, Alexander George Ogston FRS (30 January 1911 – 29 June 1996) was a biochemist who specialised in the thermodynamics of biological systems. He was particularly interested in connective tissue and the use of physico-chemical methods to study the size, weight and structure of molecules. He made the "three-point attachment" contribution to stereochemistry. His grandfather was Sir Alexander Ogston, a Scottish surgeon.

1915 ?

Horst Feistel, Horst Feistel (January 30, 1915 – November 14, 1990) was a German-born cryptographer who worked on the design of ciphers at IBM, initiating research that culminated in the development of the Data Encryption Standard (DES) in the 1970s.

Waman Dattatreya Patwardhan, Waman Dattatreya Patwardhan (January 30, 1917 - July 27, 2007) was an Indian nuclear chemist, defence scientist and an expert in the science of Explosives engineering. He was the founder director of the Explosives Research and Development

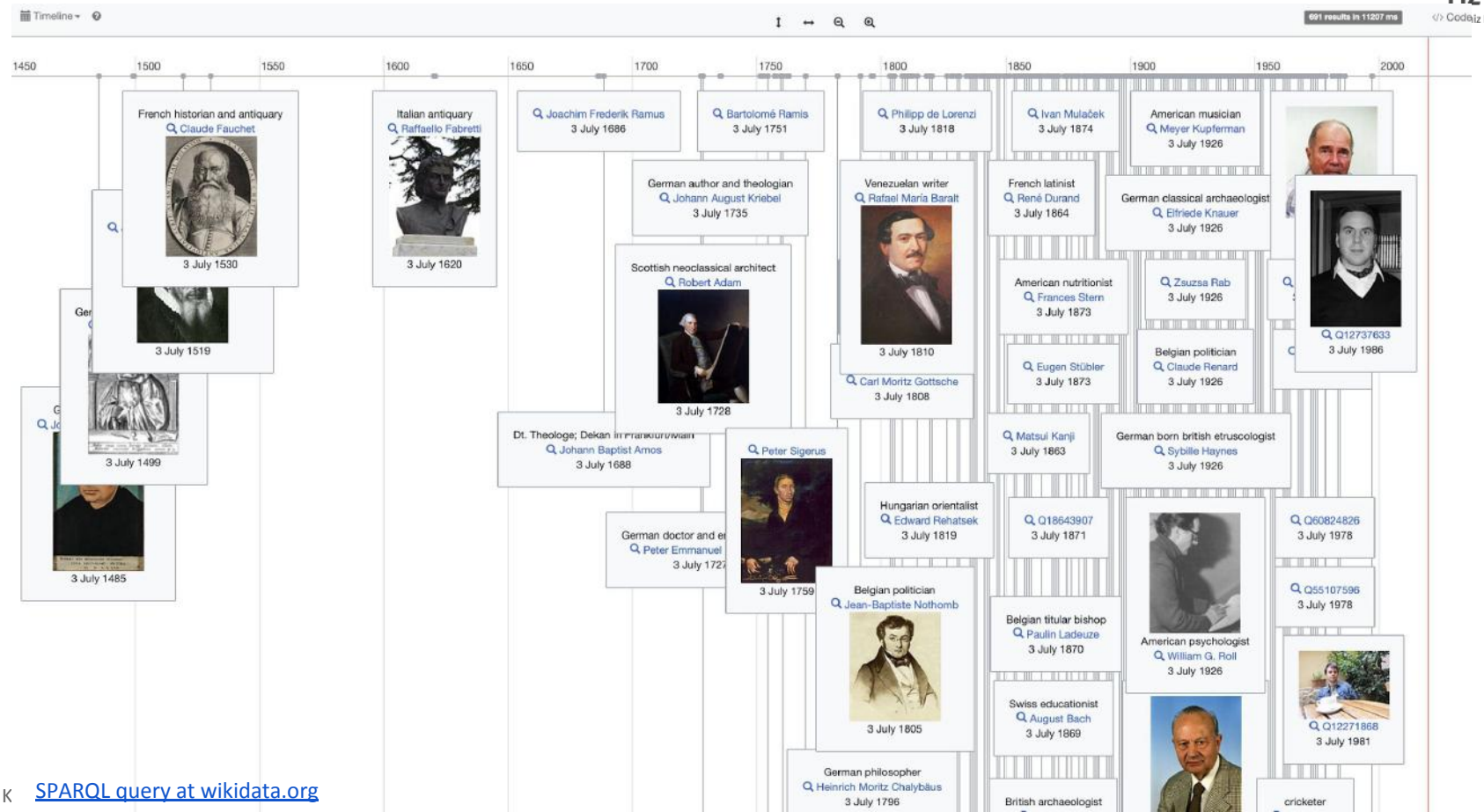
You may also try out Wikidata

```
PREFIX wdt: <http://www.wikidata.org/prop/direct/> # Properties
PREFIX wd: <http://www.wikidata.org/entity/>       # Entities

#defaultView:Timeline
SELECT distinct ?birthdate ?thumbnail ?author ?authorLabel ?authorDescription
WHERE {
  BIND(MONTH(NOW()) AS ?nowMonth) # NOW() is current date, create new variable for current month
  BIND(DAY(NOW()) AS ?nowDay)     # create new variable for current day
  ?author wdt:P106/wdt:P279* wd:Q901 ;      # (P106) occupation (Q901) scientist
          wdt:P569 ?birthdate               # (P569) birthdate
  # compare birthdate and current date
  FILTER (MONTH(?birthdate) = ?nowMonth && DAY(?birthdate) = ?nowDay) .
  OPTIONAL { ?author wdt:P18 ?thumbnail . } # (P18) image
  SERVICE wikibase:label
    { bd:serviceParam wikibase:language "en, de, fr, it" } # provide labels in English
}
ORDER BY ?birthdate
```

SPARQL query at wikidata.org

You may also try out Wikidata



K [SPARQL query at wikidata.org](#)

[4]

Next Lecture...

Picture References:

- [1] Utagawa Kuniyoshi, The three tiles in Takiyasha the Witch and the Skeleton Spectre, [Public Domain] https://commons.wikimedia.org/wiki/File:Takiyasha_the_Witch_and_the_Skeleton_Spectre.jpg