



Vorlesung Open Data

Linked Data, Wikidata und SPARQL

Termin 11, 17. Mai 2018

Benedikt Hitz-Gamper

Forschungsstelle Digitale Nachhaltigkeit


Institut für Wirtschaftsinformatik

Universität Bern

Terminübersicht Vorlesung

22. Februar 2018	Informationen zur Vorlesung, Einführung ins Thema Open Data und Open Government
01. März 2018	Open Government Data und das Impact Measuring Framework
08. März 2018	Digitale Nachhaltigkeit
15. März 2018	Entstehung und Anwendung des Öffentlichkeitsgesetz
22. März 2018	Geistiges Eigentum und die Open-Bewegung
29. März 2018	Open Data Journalism
05. April 2018	Semesterferien
12. April 2018	Open Finance und Participatory Budgeting
19. April 2018	Open Transport Data
26. April 2018	Open Geodata
03. Mai 2018	Aid Transparency und Open Aid
10. Mai 2018	Auffahrt
17. Mai 2018	Linked Data und Semantic Web
24. Mai 2018	Open Corporate Data
31. Mai 2018	Abschlusspräsentationen Open Data Apps

Heutiger Ablauf

1. Wikidata
 2. SPARQL und Linked Data
 3. Federated SPARQL Queries
- 

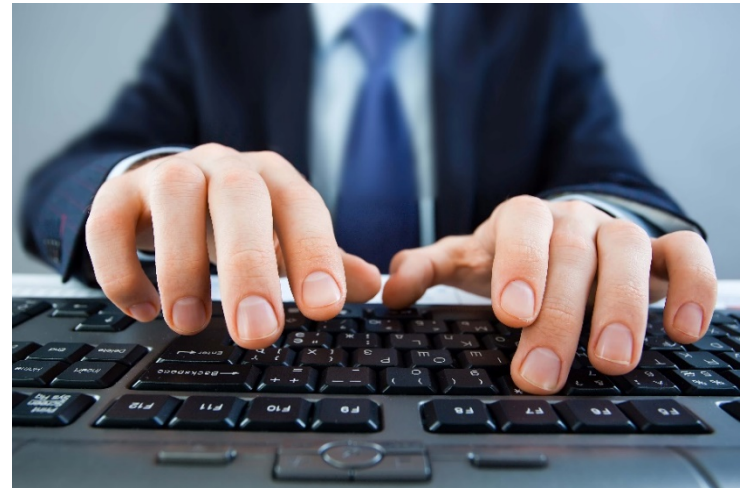
Wikidata

- > Ein Projekt von Wikimedia (Betreiber u.a. von Wikipedia)
- > Freie und offene «Knowledge Base» für strukturierte Daten
- > Für Menschen UND Computer gemacht
- > Dient als Datenbank für Wikipedia
- > Daten können als Linked Data abgerufen werden



Wikidata Hands-On

- > Was weiss Wikidata über die Stadt Bern?
- > <https://www.wikidata.org/wiki/Q70>



Was macht Linked Data interessant?

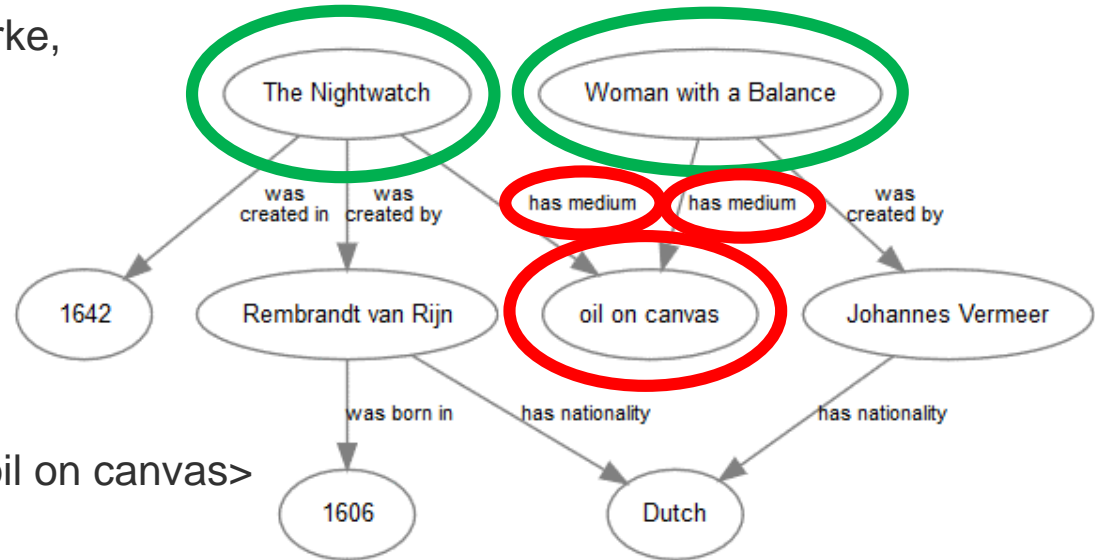
- > RDF/Linked Data beinhaltet eine mächtige Abfragesprache
 - SPARQL: **S**PARQL **P**rotocol **A**nd **R**DF **Q**uery **L**anguage
- > Linked Data aus unterschiedlichen Quellen lassen sich vergleichsweise einfach kombinieren
 - sog. “Federated Queries”



SPARQL Kurzeinführung 1

> Grundmechanismus: Pattern-Matching

- Bsp: Gesucht werden Werke, die in der Technik «Öl auf Leinwand» erstellt wurden



- Subject <has medium> <oil on canvas>

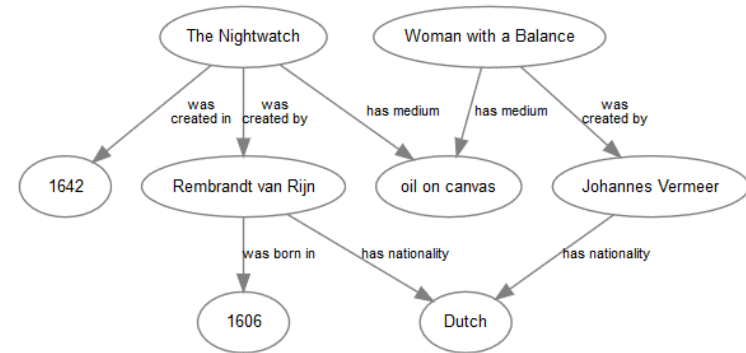
SPARQL Kurzeinführung 1

> Grundmechanismus: Pattern-Matching

Fragezeichen
bedeutet Variable



```
SELECT ?painting WHERE {
  ?painting <has medium> <oil on canvas>.
}
```



SPARQL Kurzeinführung 2

> Grundmechanismus: Pattern-Matching

- Bsp: Gesucht werden Werke, die von einem Künstler erstellt wurden, der Niederländer ist



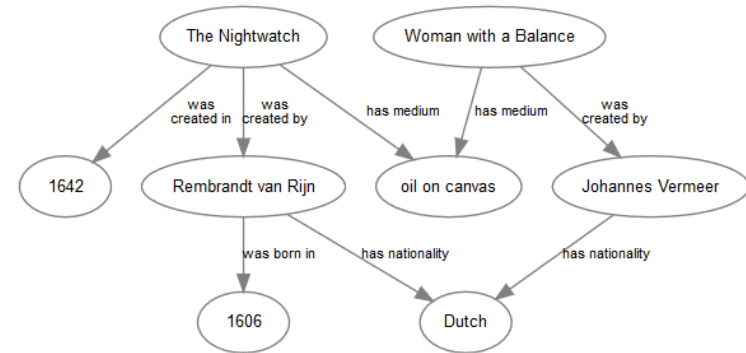
- Subject <was created by> Object
Object <has nationality> <Dutch>

SPARQL Kurzeinführung 2

> Grundmechanismus: Pattern-Matching

Variablen mit gleichem Namen müssen jeweils den gleichen Wert haben

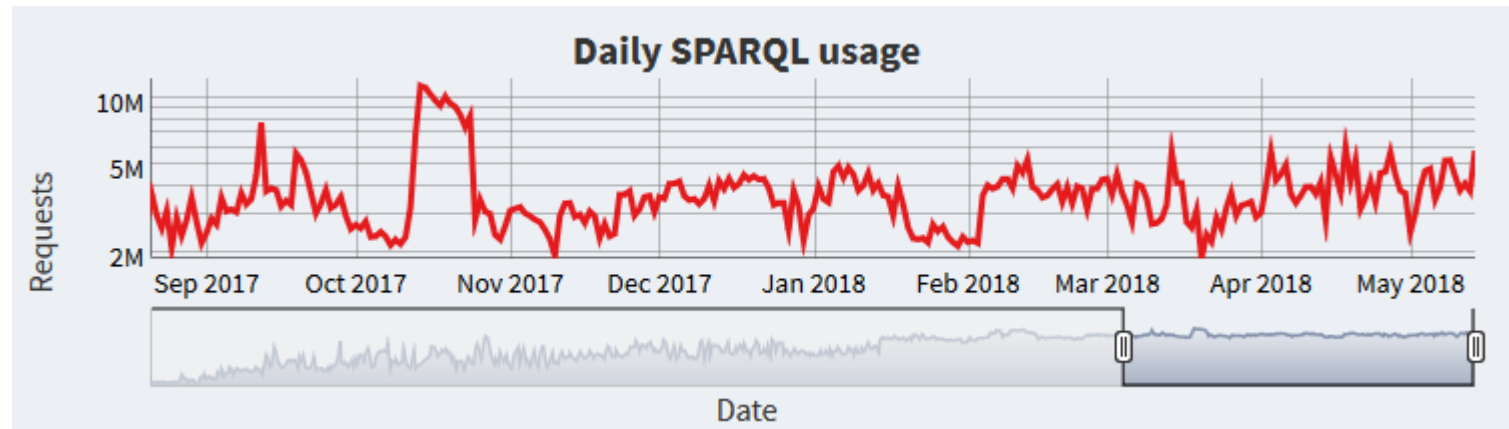
```
SELECT ?painting WHERE {
  ?painting <was created by> ?artist.
  ?artist <has nationality> <Dutch>.
}
```



Punkt bedeutet mehrere Pattern

Wikidata und SPARQL

> Braucht das überhaupt irgendwer in der «realen» Welt?



Wikidata und SPARQL

- > Wikidata ist von Grund auf multilingual und hat keine bevorzugte Sprache
 - Subjekte und Objekte haben keine «sprechenden» Namen, sondern Nummern
 - Subjekte/Objekte: Universum = Q1, Bern = Q70, Olympi Rudakov = Q9999999
 - Prädikate: Geburtsdatum = P569, Einwohneranzahl = P1082
- > In SPARQL Queries müssen Subjekt und Objekt normalerweise jeweils mit **wd:Qxxx** eingegeben werden, Prädikate mit **wdt:Pxxx**

SPARQL Hands-On 1

- > Gesucht: Alle Berge in der Schweiz, die höher als 4000m sind



SPARQL Hands-On 1

> Gesucht: Alle Berge in der Schweiz, die höher als 4000m sind

```
SELECT ?mountain
WHERE {
    ?mountain wdt:P31 wd:Q8502 .
    ?mountain wdt:P17 wd:Q39 .
}
```

SPARQL Hands-On 1

> Gesucht: Alle Berge in der Schweiz, die höher als 4000m sind

```
SELECT ?mountain ?mountainLabel
WHERE {
    ?mountain wdt:P31 wd:Q8502 .
    ?mountain wdt:P17 wd:Q39 .

    SERVICE wikibase:label { bd:serviceParam wikibase:language "de" }
}
```

SPARQL Hands-On 1

> Gesucht: Alle Berge in der Schweiz, die höher als 4000m sind

```
SELECT ?mountain ?mountainLabel ?height
```

```
WHERE {
```

```
  ?mountain wdt:P31 wd:Q8502 .
```

```
  ?mountain wdt:P17 wd:Q39 .
```

```
  ?mountain wdt:P2044 ?height .
```

```
  SERVICE wikibase:label { bd:serviceParam wikibase:language "de" }
```

```
}
```


SPARQL Hands-On 1

> Gesucht: Alle Berge in der Schweiz, die höher als 4000m sind

```
SELECT ?mountain ?mountainLabel ?height
WHERE {
  ?mountain wdt:P31 wd:Q8502 .
  ?mountain wdt:P17 wd:Q39 .
  ?mountain wdt:P2044 ?height filter (?height>4000) .

  SERVICE wikibase:label { bd:serviceParam wikibase:language "de" }
}
```

SPARQL Hands-On 1

> Gesucht: Alle Berge in der Schweiz, die höher als 4000m sind

```
#defaultView:Map
```

```
SELECT ?mountain ?mountainLabel ?height ?location
```

```
WHERE {
```

```
  ?mountain wdt:P31 wd:Q8502 .
```

```
  ?mountain wdt:P17 wd:Q39 .
```

```
  ?mountain wdt:P2044 ?height filter (?height>4000) .
```

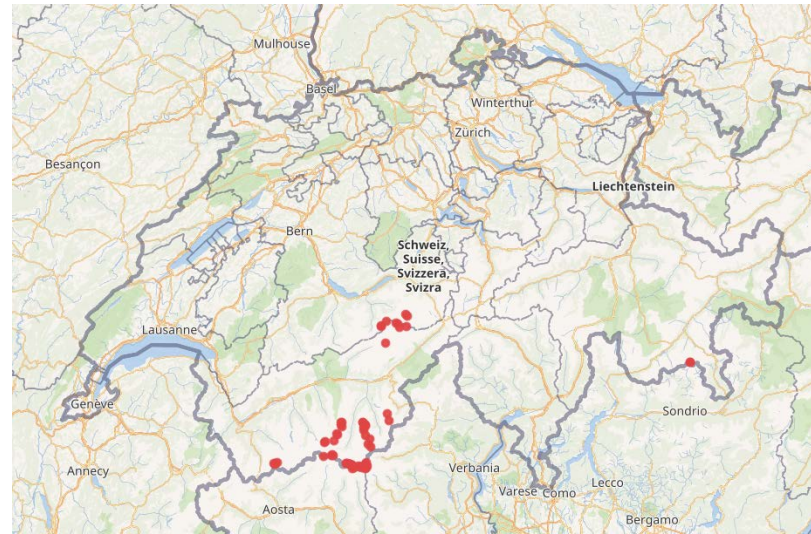
```
  ?mountain wdt:P625 ?location .
```

```
SERVICE wikibase:label { bd:serviceParam wikibase:language "de" }
```

```
}
```

SPARQL Hands-On 1

- > Gesucht: Alle Berge in der Schweiz, die höher als 4000m sind
- > <http://tinyurl.com/hhwfgd9>



SPARQL Hands-On 2

> Gesucht: Alle Gewässer, die schlussendlich in die Aare münden

```
#defaultView:Map
SELECT ?river ?riverLabel ?location WHERE {
  ?river wdt:P403 wd:Q1675.
  ?river wdt:P625 ?location.

  SERVICE wikibase:label{bd:serviceParam wikibase:language "de".}
}
```

SPARQL Hands-On 2

> Gesucht: Alle Gewässer, die schlussendlich in die Aare münden

```
#defaultView:Map
SELECT ?river ?riverLabel ?location WHERE {
  ?river wdt:P403+ wd:Q1675.
  ?river wdt:P625 ?location.

  SERVICE wikibase:label{bd:serviceParam wikibase:language "de".}
}
```

SPARQL Hands-On 2

- > Gesucht: Alle Gewässer, die schlussendlich in die Aare münden
- > Bsp: Tschentebach
 - Tschentebach (Q2457722) mündet in (P403)
 - Engstlige (Q1120135) mündet in (P403)
 - Kander (Q668478) mündet in (P403)
 - Thunersee(Q14426) Abfluss (P201)
 - Aare (Q1675)
- > Tschentebach → P403/P403/P403/P201 → Aare



SPARQL Hands-On 2

- > Gesucht: Alle Gewässer, die schlussendlich in die Aare münden

```
#defaultView:Map
SELECT DISTINCT ?river ?riverLabel ?location WHERE {
  ?river wdt:P403*/wdt:P201*/wdt:P403* wd:Q1675.
  ?river wdt:P625 ?location.

  SERVICE wikibase:label{bd:serviceParam wikibase:language "de".}
}
```

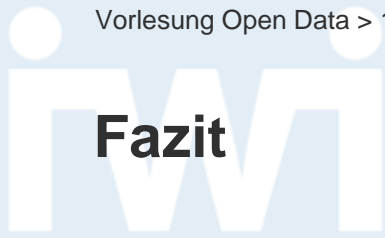


sog. Property-Path

SPARQL Hands-On 2

- > Gesucht: Alle Gewässer, die schlussendlich in die Aare münden
- > <http://tinyurl.com/y8kfzy4y>





Fazit

u^b

b
UNIVERSITÄT
BERN

- > Zum Datenformat Linked Data gehört mit SPARQL eine mächtige Abfragesprache, mit der die Daten umfangreich gefiltert, aggregiert und kombiniert werden können (bspw. CSV/XML haben keine solchen Sprachen)
- > Linked Data ist gut geeignet, wenn die Daten schon inhärent netzwerkartig aufgebaut sind (Flüsse, die in Flüsse fließen, die in Flüsse fließen)

Federated SPARQL Queries

- > Daten liefern häufig spannende Einsichten, wenn diese aus verschiedenen Quellen kombiniert werden
- > Das Problem dabei: Verknüpfung aufwändige Handarbeit

Gemeinde	Einwohner
Neuenburg	34'000
Aeschlen	305
Davos	11'060

Gemeinde	Höhe [m ü. M.]
Neuchâtel	434
Oberdiessbach	605
Davos-Platz	1560

- > Sprache, Eingemeindungen, Teilgemeinden, etc...

Federated SPARQL Queries

- > Linked Data / RDF liefert die Möglichkeiten, um Daten einfach zu kombinieren
 - URI, sameAs, seeAlso, etc
- > Data Provider muss diese allerdings auch verwenden
 - Ein Aspekt von Data Quality
- > Gutes Beispiel: Linked Data Dienst des Bundesamts für Landestopografie
 - <https://ld.geo.admin.ch/boundaries/municipality/351>

Federated SPARQL Queries

- > Verweis auf weitere Infos zur Gemeinde Bern bei Wikidata (Q70)

Bern

<https://ld.geo.admin.ch/boundaries/municipality/351>

a <http://www.opengis.net/ont/geosparql#Feature>, <http://www.geonames.org/ontology#ADM3>, <http://schema.org/AdministrativeArea>



type	Feature
type	A ADM3
type	AdministrativeArea
hasVersion	351.2017
hasVersion	351.2016
hasVersion	351.2018
hasMap	?ch.swisstopo.swissboundaries3d-gemeinde-flaeche-fill-351
name	Bern
sameAs	351?geometryFormat=geojson
seeAlso	351
seeAlso	Q70
bfsNumber	351


Federated SPARQL Queries Hands-On

- > **Gesucht:**
 - Schweizer Gemeinden mit ihren geografischen Grenzen,
 - politischem Oberhaupt, dessen Parteizugehörigkeit und
 - das entsprechende Gemeindewappen
- > **Quellen:**
 - <https://ld.geo.admin.ch/sparql/>
 - <https://query.wikidata.org/>
- > **Resultat:** <https://tinyurl.com/y9o7bf2v>

Federated SPARQL Queries Hands-On

> Ausschnitt aus SPARQL Query auf <https://ld.geo.admin.ch/sparql>

```
SELECT ?wikiURI WHERE {
  ?muni rdfs:seeAlso ?wikiURI;
  {
    SELECT ?wikiURI WHERE {
      SERVICE <https://query.wikidata.org/bigdata/namespace/wdq/sparql> {
        ?wikiURI wdt:P31 wd:Q70208;
        wdt:P6 ?mayor;
        wdt:P94 ?flagLink.
      }
    }
  }
}
```



Weitere (Federated) SPARQL Sample Queries

🔗 eCH0071 sample SPARQL queries

If you need help with additional queries please open an issue in this Github repository.

- All (not versioned) municipality IDs in Switzerland
- All versioned municipalities
- Versioned municipalities that are still active
- Political municipalities that are still active
- Political municipalities that are still active, including admission date
- All events on 01.01.2015
- Details of all events on 01.01.2015
- How many municipalities in Switzerland changed canton?
- Federated Query: Get shapes of municipalities that stopped existing 2016
- Federated query: Show all municipality changes in 2016