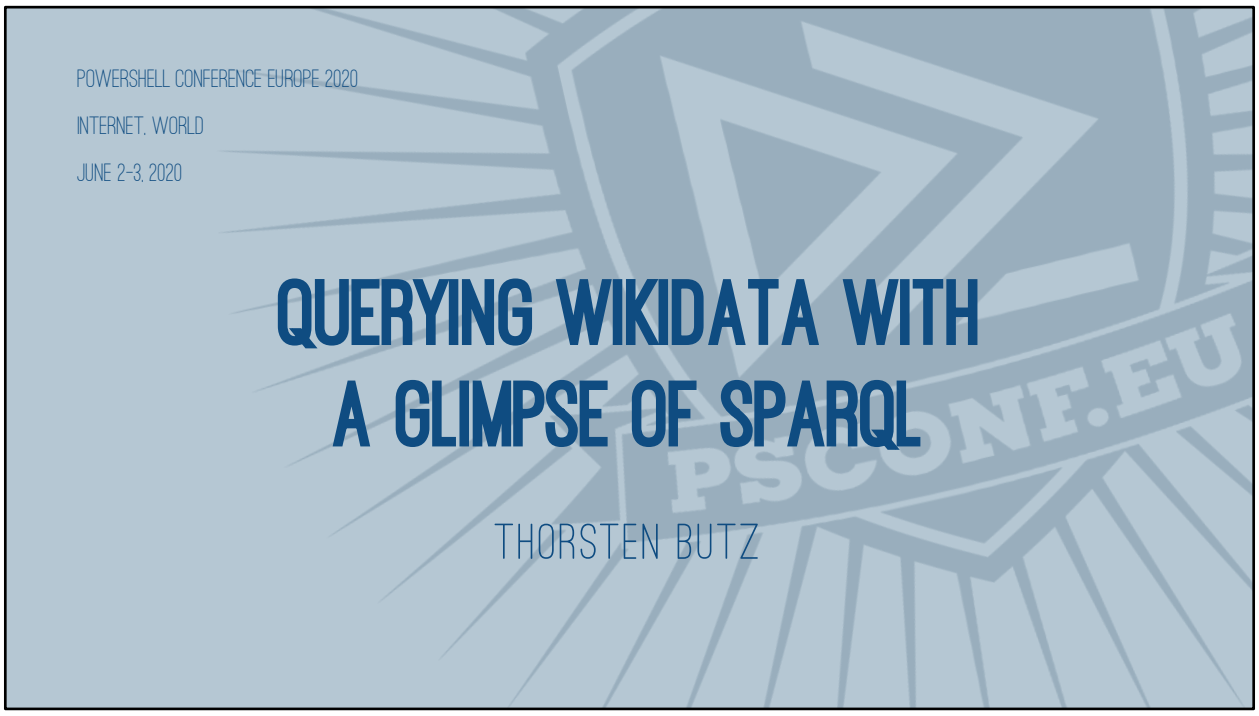


Abstract

Querying Wikidata with a glimpse of SPARQL

Wikidata represents the underlying basement of Wikipedia, it's the collaboratively edited knowledge base filling up Wikipedia with data.
Querying Wikidata enables you to gain access to to a universe of information,
just in case you know how to answer the right questions.

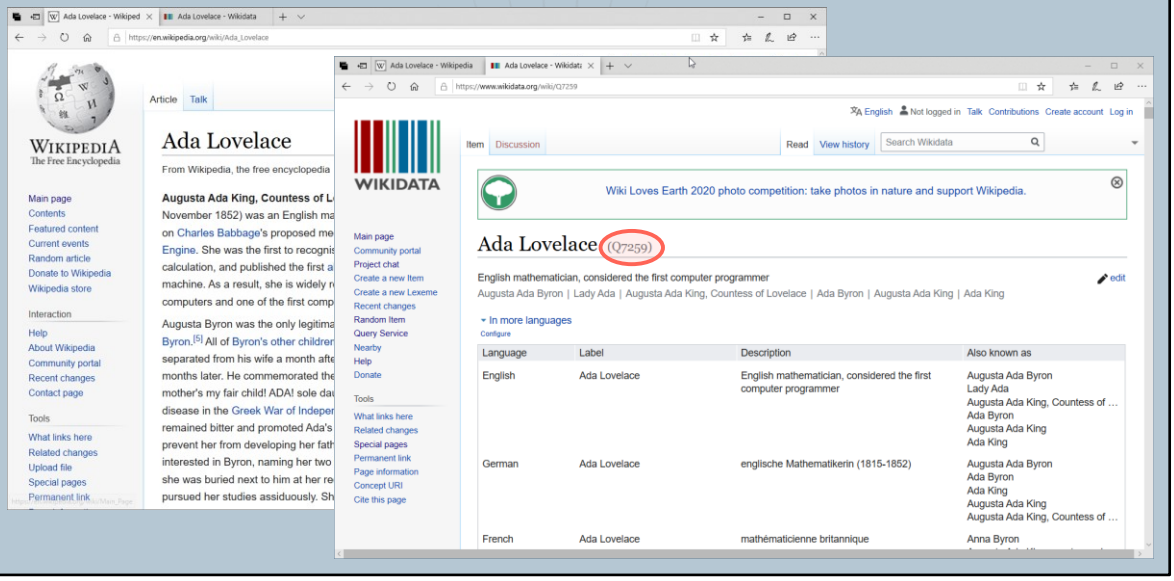
This session provides an insight to wikidata/wikimedia queries by SPARQL, the semantic query language from W3C.

After you watched this, you will be able to embed SPARQL queries inside your PowerShell code.


PSCONF.EU 2020 WOULDN'T BE POSSIBLE WITHOUT OUR AWESOME SPONSORS
Microsoft
System Frontier
ScriptRunner
powershell.one

Thank you!

GET-HELP ABOUT_WIKIDATA



Screenshots:
https://en.wikipedia.org/wiki/Ada_Lovelace
https://www.wikidata.org/wiki/Q7259

What is Wikidata?
"Wikidata is a knowledge database. It contains millions of statements, such as “the capital of Canada is Ottawa”, or “the Mona Lisa is painted in oil paint on poplar wood”, or “gold has a melting point of 1,064.18 degrees Celsius”."

https://www.wikidata.org/wiki/Wikidata:SPARQL_tutorial#Grouping

```
# Searching for someone like Ada ...
SELECT ?programmer ?date_of_birth ?programmerLabel WHERE {
  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en". }

  ?programmer wdt:P106 wd:Q5482740;
              wdt:P31 wd:Q5;
              wdt:P21 wd:Q6581072;
              wdt:P569 ?date_of_birth.

  FILTER("1801-01-01"^^xsd:dateTime <= ?date_of_birth &&
         ?date_of_birth <= "1900-12-31"^^xsd:dateTime).
}
```

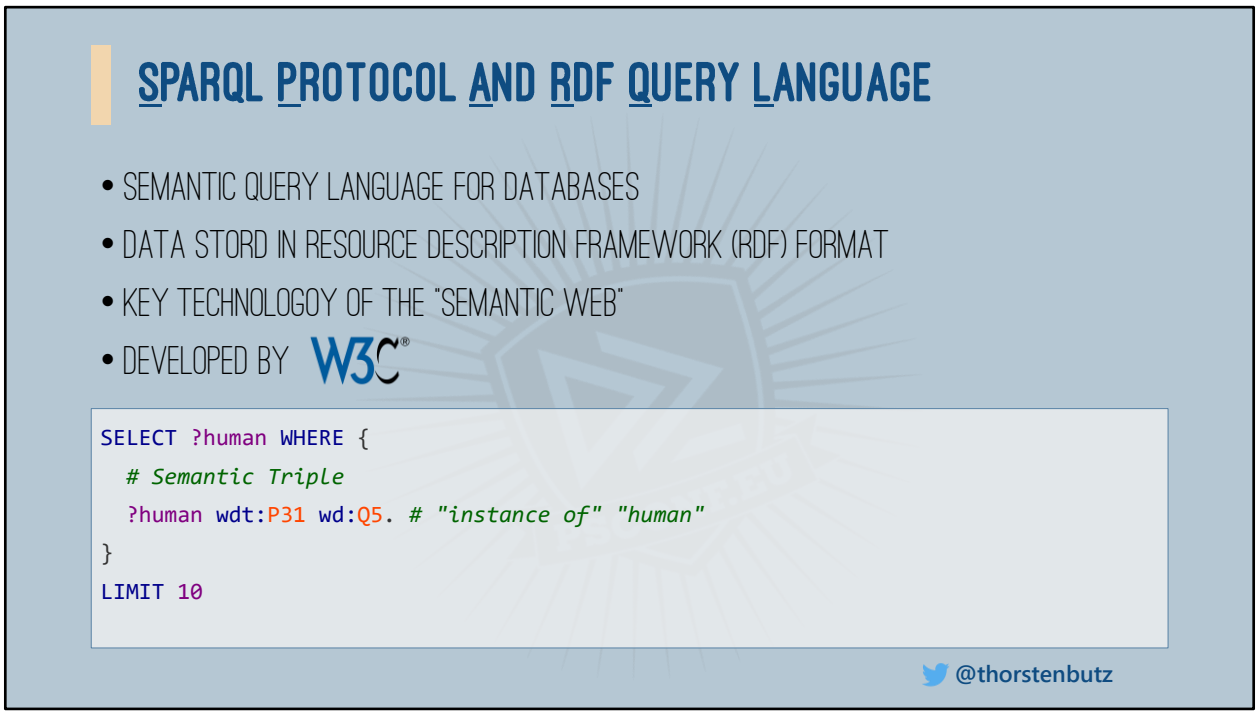
@thorstenbutz

The query:
https://w.wiki/RWj

```
# Searching for someone like Ada ...
SELECT ?programmer ?date_of_birth ?programmerLabel WHERE {
  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en". }

  # Who has   "occupation" "programmer"?
  ?programmer wdt:P106 wd:Q5482740;
  #           "instance of" "human"
              wdt:P31 wd:Q5;
  #           "sex or gender" "female"
              wdt:P21 wd:Q6581072;
  #           "date of birth"
              wdt:P569 ?date_of_birth.


  # Show only programmers from the 19th century
  FILTER("1801-01-01"^^xsd:dateTime <= ?date_of_birth &&
         ?date_of_birth <= "1900-12-31"^^xsd:dateTime).
}
```

@thorstenbutz

The query:
https://w.wiki/RWf



What is SPARL?
"SPARQL is a language to formulate questions (queries) for knowledge databases. With the right database, a SPARQL query could answer questions like “what is the most popular tonality in music?” or “which character was portrayed by the most actors?” or “what’s the distribution of blood types?” or “which authors’ works entered the public domain this year?”."

Reference:
https://en.wikipedia.org/wiki/SPARQL

Tutorial:
https://www.wikidata.org/wiki/Wikidata:SPARQL_tutorial

BASICS

- VARIABLE => ?human
- ITEM => Q5 (human)
- PROPERTY => P31 (instance of)
- PREFIX => wdt: (wikidata property)
  wd: (wikidata item)

```
SELECT ?human WHERE {
  # Semantic Triple
  ?human wdt:P31 wd:Q5. # "instance of" "human"
}
LIMIT 10
```

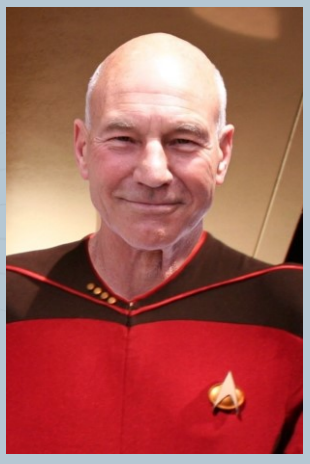
@thorstenbutz

Reference:
https://www.wikidata.org/wiki/Wikidata:SPARQL_tutorial


```
# Demo B: Star Trek

# Who performed Jean-Luc Picard?
SELECT ?item ?itemLabel WHERE {
  SERVICE wikibase:label {
    bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en".
  }
  # "Dixon Hill" has performer JLP.
  ?item wdt:P175 wd:Q16276.
  # "Jean-Luc Picard" has performer Patrick Stewart.
  wd:Q16276 wdt:P175 ?item .
}

# What items have been created by Gene Roddenberry?
SELECT ?item ?itemLabel ?languages_spoken__written_or_signed
  ?languages_spoken__written_or_signedLabel ?date_of_birth
  ?country_of_citizenship ?country_of_citizenshipLabel
WHERE {
    SERVICE wikibase:label {
      bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en".
    }
    ?item wdt:P170 wd:Q191716.
    ?item wdt:P569 ?date_of_birth.

    OPTIONAL { ?item wdt:P1412 ?languages_spoken__written_or_signed. }
    OPTIONAL { ?item wdt:P27 ?country_of_citizenship. }

  }
```

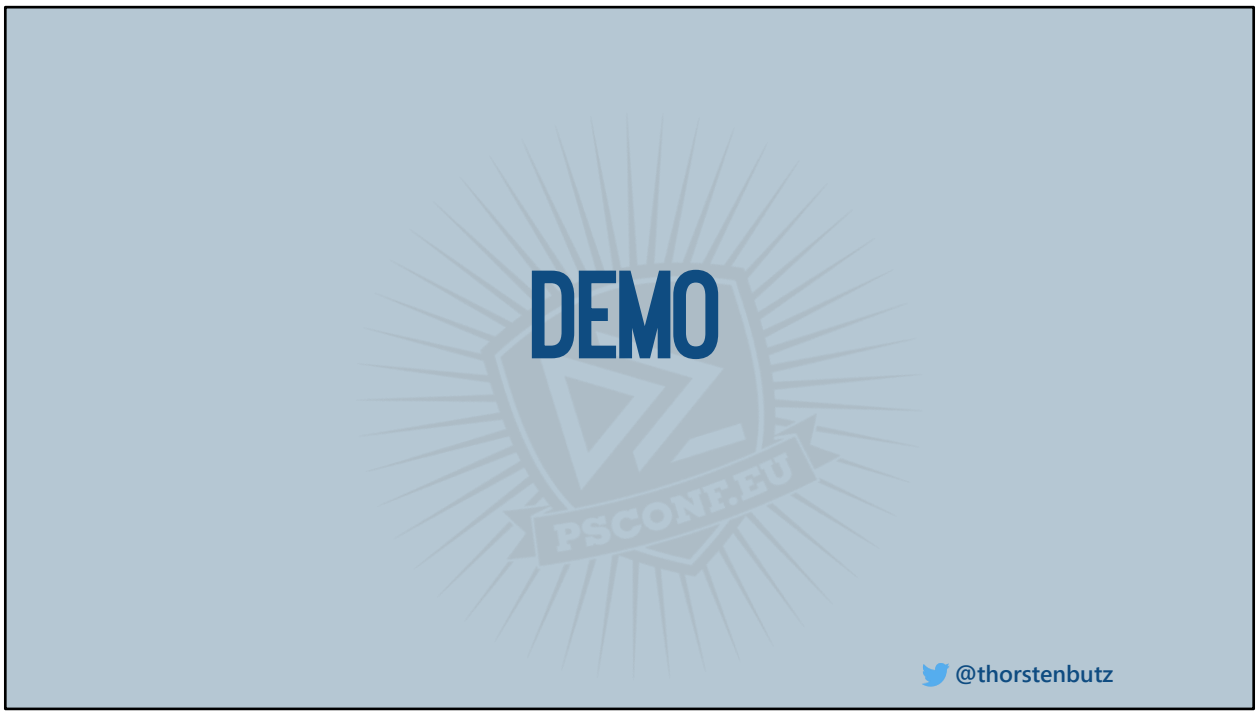
SUBJECT - PREDICATE - OBJECT

```
SELECT ?item ?itemLabel WHERE {
   SERVICE wikibase:label {..}
   # Dixon Hill has "performer" "Jean-Luc Picard".
   ?item wdt:P175 wd:Q16276.
}

SELECT ?item ?itemLabel WHERE {
   SERVICE wikibase:label {..}
   # Jean-Luc Picard has "performer" "Patrick Stewart".
   wd:Q16276 wdt:P175 ?item.
}
```



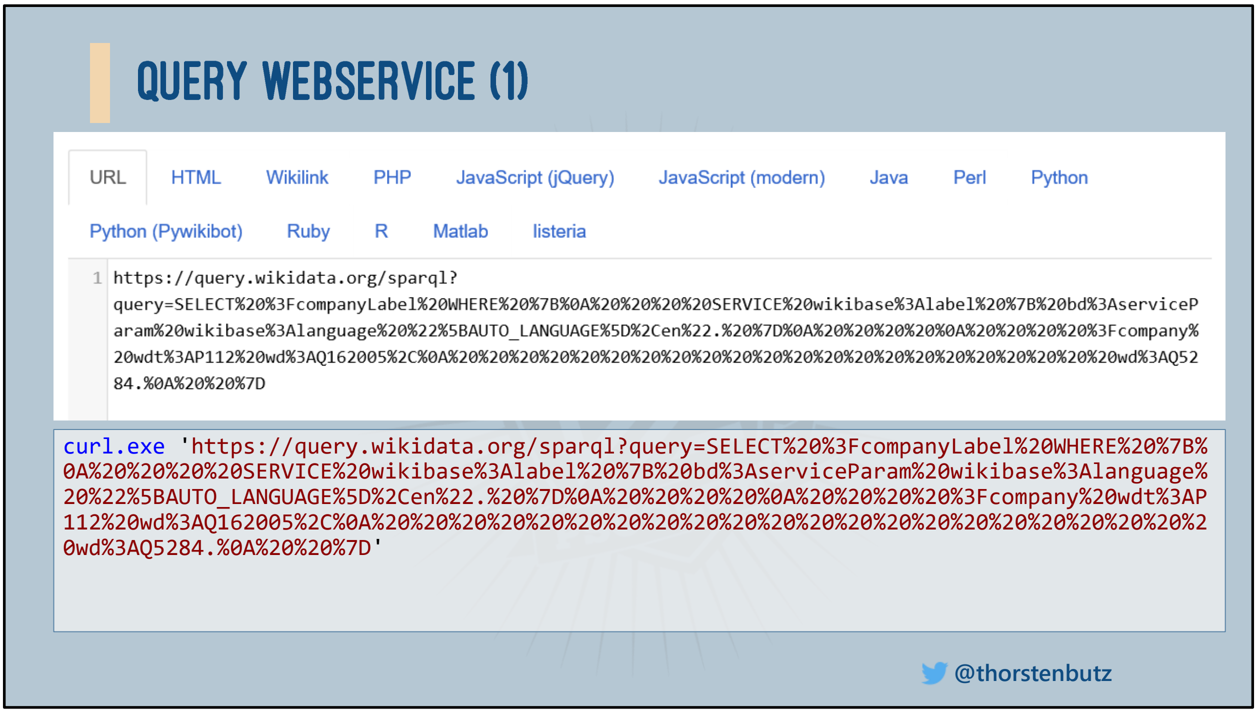
@thorstenbutz

Reference:
https://en.wikipedia.org/wiki/Semantic_triple




```
# Demo C: People at Microsoft

# What was founded by Paul and Bill?
SELECT ?companyLabel WHERE {
    SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGU
AGE],en". }
    ?company wdt:P112 wd:Q162005,
                      wd:Q5284.
}

# Microsoft employees
SELECT ?item ?itemLabel ?GitHub_username ?notable_workLabel WHERE {
    SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGU
AGE],en". }
    ?item wdt:P108  wd:Q2283;            # ? has employer MSFT
          wdt:P2037 ?GitHub_username;    # ? has a Github username
          wdt:P800  ?notable_work.       # ? has a notable work
  }
```

You will find the complete set of example files here:
https://www.thorsten-butz.de/psconfeu2020/
https://github.com/psconfeu/2020




```
# The founder of MSFT. We had that before …

SELECT ?companyLabel WHERE {
    SERVICE wikibase:label {
      bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en".
    }
    ?company wdt:P112 wd:Q162005,
                      wd:Q5284.
}
```

QUERY WEBSERVICE (2)

```
$sparql = @'
SELECT ?companyLabel WHERE {
    SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en". }
    ?company wdt:P112 wd:Q162005,
                      wd:Q5284.
}
'@

# XML
Invoke-RestMethod -Uri "https://query.wikidata.org/sparql?query=$sparql"

#CSV
Invoke-RestMethod -Uri "https://query.wikidata.org/sparql?query=$sparql"
       -Headers @{'Accept' = 'text/csv' } | ConvertFrom-Csv
```

@thorstenbutz

The query:
https://w.wiki/Ryp

GROUPING

```
SELECT ?item ?itemLabel ?GitHub_username ?notable_workLabel WHERE {
    SERVICE wikibase:label {..}
    ?item wdt:P108  wd:Q2283;            # ? has employer MSFT
          wdt:P2037 ?GitHub_username;    # ? has a Github username
          wdt:P800  ?notable_work.       # ? has a notable work
  }
```

item	itemLabel	GitHub_username	notable_workLabel
wd:Q62849	Miguel de Icaza	migueldeicaza	GNOME
wd:Q62849	Miguel de Icaza	migueldeicaza	Mono
wd:Q92838	Erich Gamma	egamma	Design Patterns
wd:Q3299582	Paul Buchheit	paulbuchheit	Gmail
wd:Q3299582	Paul Buchheit	paulbuchheit	FriendFeed
wd:Q9335956	Shawn Hargreaves	shawnhar	Allegro

The query:
https://w.wiki/RaH

```
# Grouping results
SELECT ?item ?itemLabel ?GitHub_username
  (group_concat(?notable_workLabel;separator=",") as ?arrNotableWork)
WHERE {
    SERVICE wikibase:label {
      bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en".
      ?item rdfs:label ?itemLabel.
      ?notable_work rdfs:label ?notable_workLabel.
    }
    ?item wdt:P108 wd:Q2283;            # ? has employer MSFT
          wdt:P2037 ?GitHub_username;   # ? has a Github username
          wdt:P800 ?notable_work.       # ? has a notable work
  }
  GROUP BY ?item ?itemLabel ?GitHub_username having(count(?notable_workLabel) >= 2)
```

@thorstenbutz

The query:
https://w.wiki/Ryq

```
# Grouping results
SELECT ?item ?itemLabel ?GitHub_username
  (group_concat(?notable_workLabel;separator=",") as ?arrNotableWork)
WHERE {
    SERVICE wikibase:label {
      bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en".
      ?item rdfs:label ?itemLabel.
      ?notable_work rdfs:label ?notable_workLabel.
    }
    ?item wdt:P108 wd:Q2283;            # ? has employer MSFT
          wdt:P2037 ?GitHub_username;   # ? has a Github username
          wdt:P800 ?notable_work.       # ? has a notable work
 }
 GROUP BY ?item ?itemLabel ?GitHub_username having(count(?notable_workLabel) >= 2)
```

WIKIDATA

@thorstenbutz

The query:
https://w.wiki/RaM
You can also click on the WIKIDTA on the slide.

item	itemLabel	GitHub_username	arrNotableWork
wd:Q62849	Miguel de Icaza	migueldeicaza	GNOME,Mono
wd:Q3299582	Paul Buchheit	paulbuchheit	Gmail,FriendFeed

SUMMARY

- WIKIDATA IS A GREAT RESOURCE
- POWERSHELL IS NOT (YET) A 1ST CLASS CITIZEN ON THE WEB
- AN OBJECT BASED SHELL ALIGNS PERFECTLY WITH WEBSERVICES
- GET-DEMOFILE | START-TRYING

SPARQL 1.1 OVERVIEW (W3C RECOMMENDATION)
HTTPS://WWW.W3.ORG/TR/SPARQL11-OVERVIEW/

WIKIDATA QUERY SERVICE/USER MANUAL
HTTPS://WWW.MEDIAWIKI.ORG/WIKI/WIKIDATA_QUERY_SERVICE/USER_MANUAL

WIKIDATA SPARQL TUTORIAL
HTTPS://WWW.WIKIDATA.ORG/WIKI/WIKIDATA:SPARQL_TUTORIAL

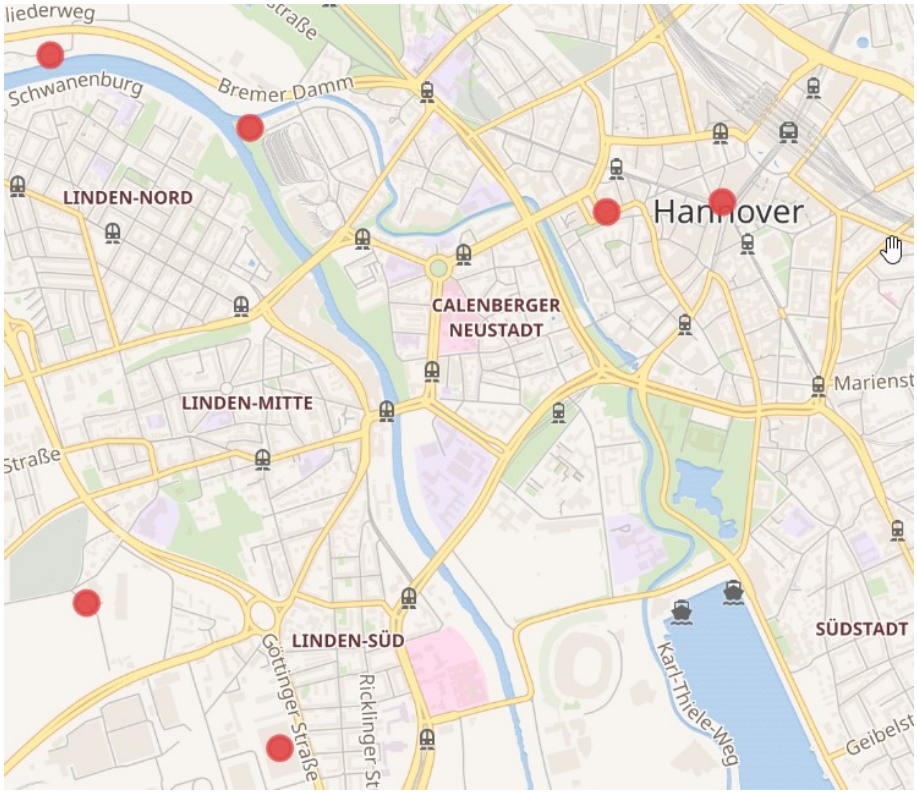
@thorstenbutz

SPARQL Examples:
https://www.wikidata.org/wiki/Wikidata:SPARQL_query_service/queries/examples
https://www.w3.org/2009/Talks/0615-qbe/

Reference:
https://en.wikipedia.org/wiki/Stevenote

```
# Pubs in the neighborhood of HCC (Hannover Congress Centrum)
SELECT ?place ?placeLabel ?location ?distance
WHERE
{
  SERVICE wikibase:label {  bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en" . }
  SERVICE wikibase:around {
    ?place wdt:P625 ?location .
    bd:serviceParam wikibase:center ?hccLocation .
    bd:serviceParam wikibase:radius "5" .
    bd:serviceParam wikibase:distance ?distance.
  }
  # HCC coordinates
  wd:Q1576239 wdt:P625 ?hccLocation .
  # Q5307737: "drinking establishment"
  FILTER EXISTS { ?place wdt:P31/wdt:P279* wd:Q5307737 } .
}
ORDER BY ASC(?distance)
```

The query:
https://w.wiki/Rd4



ABOUT_SPEAKER

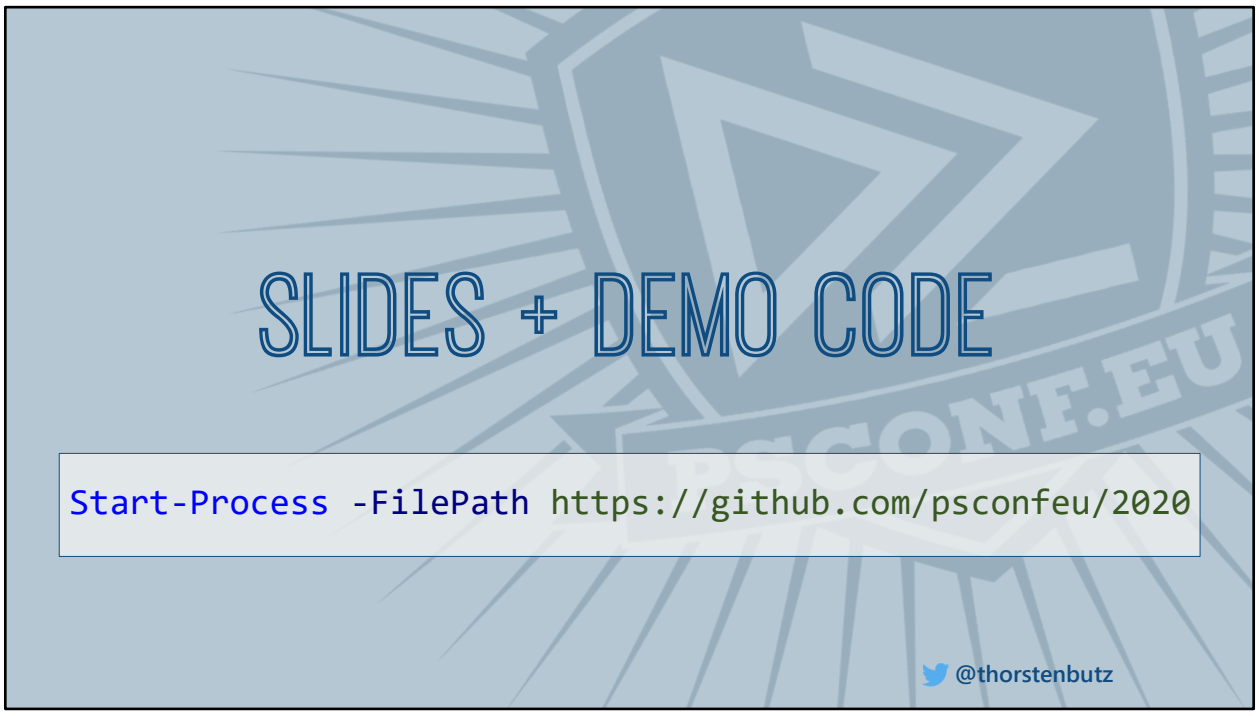
```
{
    "Speaker": "Thorsten Butz",
    "Uri"    : "thorsten-butz.de",
    "Twitter": "@thorstenbutz",
    "Podcast": "slidingwindows.de"
}
```

@thorstenbutz

Thanks for your attention.
I am professional writer, trainer and consultant.
You will find me an (almost any) social network.
Ping me there …

You're welcome to contact me if you'd like to engage me.
https://thorsten-butz.de

Subscribe to my audiocast/podcast:
http://www.slidingwindows.de/?feed=slw-aac
http://www.slidingwindows.de/?feed=slw-mp3
.. or simply search in the iTunes directory or your podcast client for "Thorsten Butz".



All sessions:
https://github.com/psconfeu/2020

My session:
https://www.thorsten-butz.de/psconfeu2020/

Stay in touch:
https://psconf.eu
https://twitter.com/psconfeu