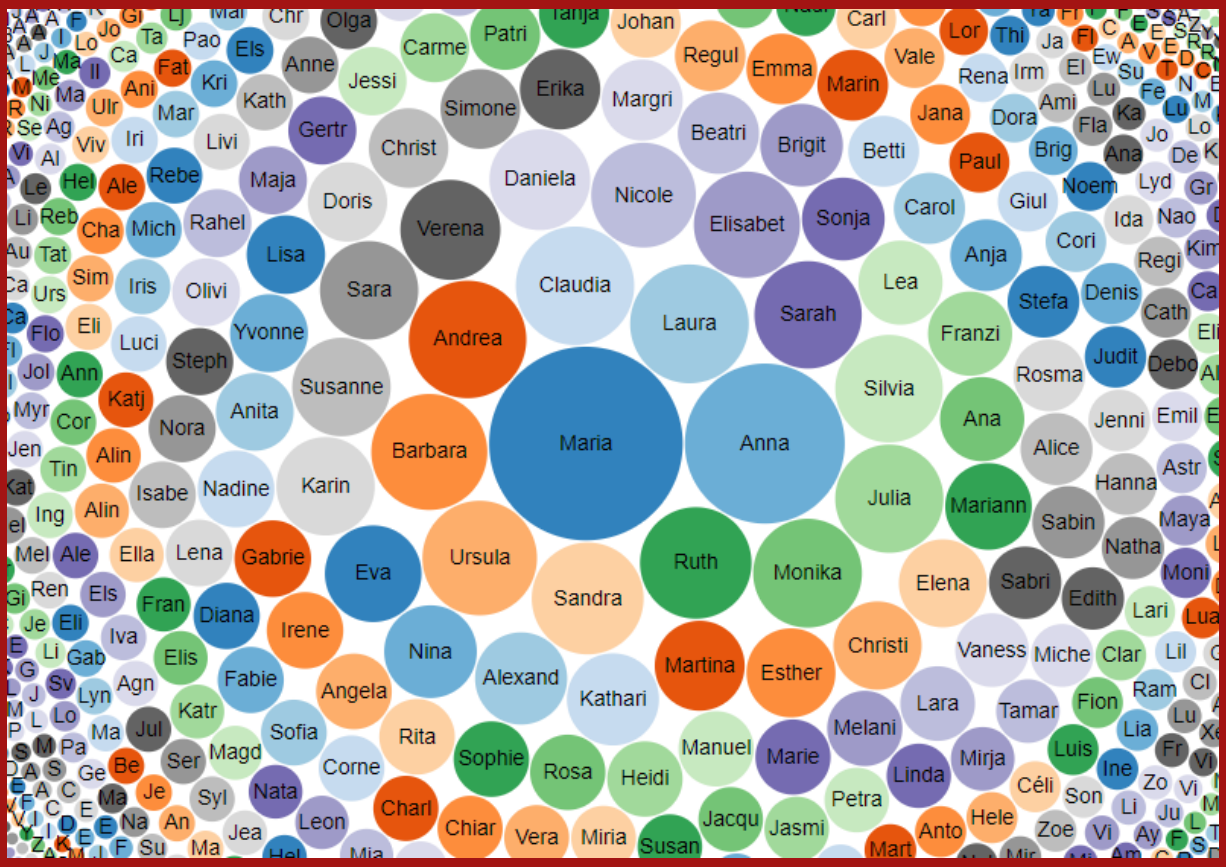
Stadt Zürich
Statistik

Linked Data Workshop

Swiss Statistics Meeting 2023



Judith Riegelnig
Klemens Rosin
Rolf Schenker
Hansjörg Stalder
Daniel Truttmann

Basel, August 30, 2023

Imprint

Publisher
Stadt Zürich
Statistik
Postfach, 8022 Zürich
stadt-zuerich.ch/statistik

T +41 44 412 08 00

Table of Contents

1 Introduction

Beginner-Level

The linked Data Workshop is designed for those who have little or no knowledge of linked data.

Procedure and data

After a short theory session, you will practice on your own. You will be supported by tutors. We will work with first name data of the city of Zurich.

2 Theory

2.1 Concepts

Semantics

For machines, the meaning of texts, the **semantics**, is difficult to understand; in particular, when further information about the **context** is missing. For example, it is unclear to machines what is meant by the words «palm», «jaguar», and «bat» (homonyms).

Figure 1: examples of unclear semantics: palm, jaguar, and bat

Triples

In linked data, relationships or properties are described with so-called triples. A triple consists of **subject**, **predicate,** and **object**.

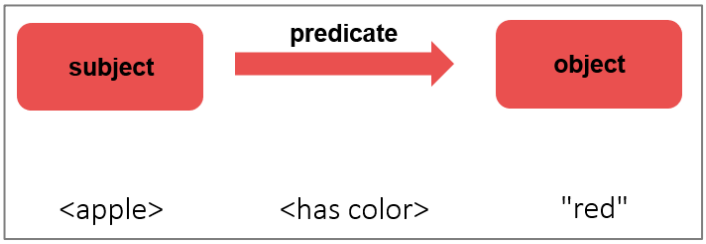


Figure 2: triple

Graph

Several triples can be connected; this results in a **graph**.

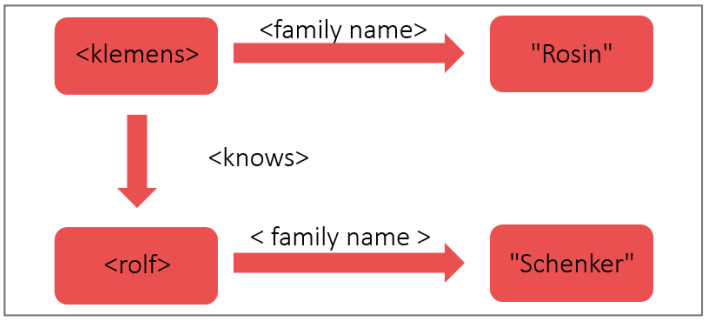


Figure 3: graph

2.2 Terminology

URI

In linked data, unique identification is important. Hence, there exist **Uniform Resource Identifiers** (**URI**). They can be both real and abstract things to be uniquely identified. Examples for URIs are webpages, people, products, locations, properties, or relationships.

Unique identification also plays a central role in other areas. This is the case, for example, with literature; the **ISBN** (International Standard Book Number) is used there. Another example are web pages and the **URL** (Uniform Resource Locator). However, URIs are more comprehensive than URLs.

Figure 4: URI vs. URL

RDF

The structure of linked data are not tables as in classical databases, but RDF. This abbreviation stands for Resource Description Framework.

- **Resource**: Everything that can be uniquely identified with a URI
- **Description**: Description of the resources: attributes, properties
- **Framework**: Framework for these descriptions

SPARQL

SPARQL is the **query language for linked data**. The abbreviation stands for SPARQL Protocol And RDF Query Language. Thus, it is a recursive created word. SPARQL contains similar elements as the programming language SQL.

2.3 SPARQL

Introduction

This chapter does not contain comprehensive SPARQL training. It covers only **elementary code elements** enabling you to work with SPARQL in the practical part.

Additional information about SPARQL

More information about SPARQL can be found here:

- SPARQL on Wikibooks (Link), including functions (Link)
- SPARQL on W3C (Link)
- Wikidata SPARQL tutorial (Link)
- data.world SPARQL-Tutorial (Link)
- Linked Data Engineering: SPARQL (Link, video series)

SELECT, WHERE

In SPARQL, variable names are preceded by a question mark. The WHERE statement contains the triples. Which variables are output is specified in the SELECT statement. An asterisk (*) stands for the output of all variables. With SELECT DISTINCT only different records (i.e. lines) are returned.

```
SELECT ?fruit
WHERE
{
   ?fruit <hasColor> "yellow" .
   ?fruit <tastes> "bitter" .
}
```

Figure 5: SELECT, WHERE

Punctuation

A triple is always finished with a **dot**. For better readability it is common that in SPARQL a space is used in front of the punctuation marks.

```
?fruit <hasColor> "yellow" .
?fruit <tastes>   "bitter" .
?fruit <tastes>   "fruity" .
```

Figure 6: dots at the end of the triples

If several triples have the same **subject**, only the first subject can be written in the code and a **semicolon** is used between the triples.

If **subject and predicate** are the same, they can be omitted in the subsequent triples; the triples are separated by a **comma**.

```
?fruit <hasColor> "yellow" ;
       <tastes>   "bitter" ,
                  "fruity" .
```

Figure 7: semicolon, comma, dot

LIMIT, ORDER BY

LIMIT restricts the number of records (lines) in the output. For example, with LIMIT 10 only ten lines are printed. This is especially useful for tests. With **ORDER BY** the output lines are sorted.

```
SELECT *
WHERE
{
  x y ?a .
  m n ?b .
}
LIMIT 10
```

```
SELECT *
WHERE
{
  x y ?a .
  m n ?b .
}
ORDER BY  ?a  DESC(?b)
```

Figure 8: LIMIT, ORDER BY

BIND, FILTER

New columns are added with BIND. The YEAR function converts dates to years. With the FILTER statement certain rows are selected.

```
SELECT *
WHERE
{
        BIND(YEAR(?date) AS ?year)

        FILTER(?year = 2023)
}
```

Figure 9: BIND, YEAR, FILTER

PREFIX

URIs can consist of a relatively large number of characters. Therefore, it is worthwhile to specify a part of the URI (a so-called **prefix**) at the beginning of the code. The prefix can then be reused in the SPARQL code. This makes the code more readable.

```
SELECT *
WHERE
{
  ?obs  <https://ld.stadt-zuerich.ch/statistics/property/TIME>   ?time ;
        <https://ld.stadt-zuerich.ch/statistics/property/RAUM>  ?space .
}
```

```
PREFIX property: <https://ld.stadt-zuerich.ch/statistics/property/>

SELECT *
WHERE
{
  ?obs  property:TIME    ?time ;
        property:RAUM    ?space .
}
```

Figure 10: PREFIX

2.4 Data Structures

Describe data points

The basic services of Statistik Stadt Zürich (web page and open data platform) contain data on the first names of the resident population by year of birth. For example, there are (according to the population register at the end of 2022) five males in the city of Zurich with the first name «Charles» that were born in 1997 who belong to the population present (Figure 11).

T_1
Vornamen der wirtschaftlichen Wohnbevölkerung der Stadt Zürich, 2022
mit mindestens 10 Personen
1. Vorname, männlich

Quelle: Statistik Stadt Zürich, BVS

Vorname	1996	1997	1998	1999	2000
Charles	2	5	2	3	1
Charlie	0		0	0	0
Chi	0		0	1	0
Chris	0	1	0	1	2
Christian	29	22	14	10	10
Christof	1	0	0	0	0

Figure 11: current data structure

Linked data describes the individual **data points** and their **properties** (Figure 12).

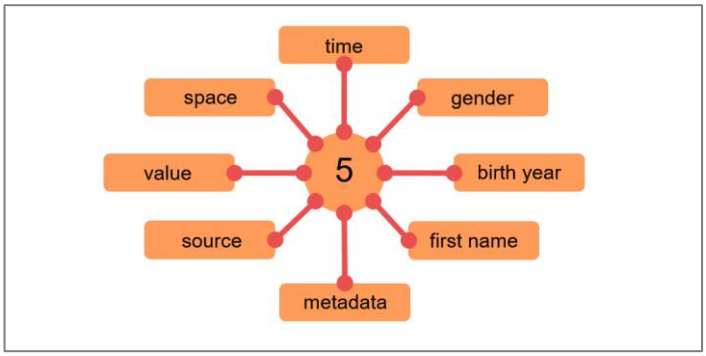


Figure 12: describe data points

Hierarchical data structures

To be able to navigate through the data, it helps to know its structures. Linked data is arranged hierarchically: Within a **graph** there are several **cubes**. The cubes can contain multiple **observation sets**: However, in the Zurich Data there is currently only one observation set in each cube. The observation sets consist of the individual **observations** (data points).

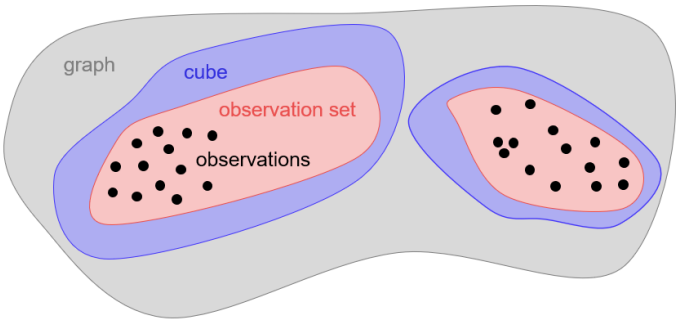


Figure 13: data structures

3 Exercices

Application

The practical part follows a certain «path»: First, a cube is selected; the evaluations are refined. Finally, the results are presented graphically.

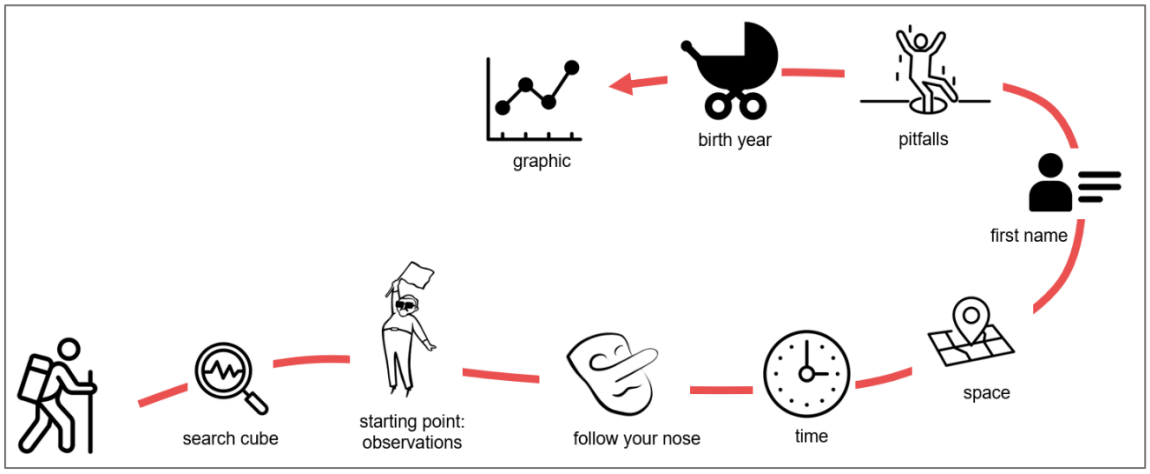


Figure 14: practical part

Additional exercises

You may continue with your own ideas and run further queries. Moreover, we have provided additional exercises:

- Gender
- Ranking
- Federated Query: Wikidata bubble chart
- Federated Query: HAVING
- Federated Query: Wikidata portraits

3.1 Search Cube

In the practical part the name data from the city of Zurich is used. Three cubes contain information about names.

Task: This link takes you to the so-called SPARQL editor of the city of Zurich. There, you can execute queries on the SPARQL endpoint of the city of Zurich. Now you can run the query (black arrow, top right).

We want to work with the data of the first names of persons belonging to the population. What is the cube number?

Hint: It is not necessary that you understand this SPARQL code. You will look at SPARQL codes in detail later.

```
PREFIX schema: <http://schema.org/>
PREFIX schemac: <https://cube.link/>

SELECT DISTINCT ?cu ?cuLabel
FROM <https://lindas.admin.ch/stadtzuerich/stat>
WHERE {

    #Cubes und Cube-Labels
      ?cu a schemac:Cube ;
          schema:name ?cuLabel .

    #Text filtern
        FILTER REGEX(LCASE(STR(?cuLabel)), "name") .

}

ORDER BY ?cuLabel
```

Solution: The cube with the first names of the resident population is called https://ld.stadt-zuerich.ch/statistics/000437.

3.2 Starting Point: Observations

Starting point: At the beginning of linked data analyses it is helpful to look at individual observations of a specific cube.

Task: Run this code to display ten observations of cube 000437.

Try to understand as many elements of the code as possible. How is this code structured?

```
1  PREFIX schema: <http://schema.org/>
2  PREFIX schemac: <https://cube.link/>
3  PREFIX datacubes: <https://ld.stadt-zuerich.ch/statistics/>
4
5  SELECT DISTINCT *
6  FROM <https://lindas.admin.ch/stadtzuerich/stat>
7  WHERE {
8
9    #Fuer den Cube 000437 (Vornamen der Wohnbevölkerung): ObserationSet
10     datacubes:000437 schemac:observationSet ?obsSet .
11
12   #Observations des ObservationSets
13     ?obsSet schemac:observation ?obs .
14
15 }
16
17 LIMIT 10
```

Solution: Structure of the code.

- PREFIXES: at the beginning prefixes are defined, so that the URIs in the code are not too long
- SELECT DISTINCT *: return all variables in the output (but only different rows)
- FROM: graph of the city of Zurich
- WHERE: triples used
- line 10: cube 000437 contains an observation set (variable name: ?obsSet)
- line 13: the observation set contains observations (variable name: ?obs)
- LIMIT 10: ten observations returned

3.3 Follow Your Nose

In linked data analyses, a helpful principle is: «follow your nose». This is how you get to know the properties of the observations.

Task: In the preceding code you have output observations. Randomly select an observation; for example, this one.

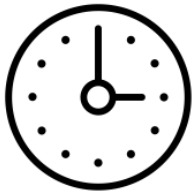
What is in the left column? What is in the right column? What are the properties of this data point?

Hint: open in new tab with «ctrl + click».

BEW	2
GEJ	GEJ1952
NAF	NAF0001
NAM	NAM0BA0
RAUM	R30000
SEX	SEX0001
TIME	2021-12-31 (date)
ZEIT	Z31122021

Solution: The left column contains the URIs of the predicates of the triples; the right column contains the URIs of the objects.

An example about the left column: BEW stands for https://ld.stadt-zuerich.ch/statistics/measure/BEW (has population).
An example to the right column: 2
The triple is: ?obs https://ld.stadt-zuerich.ch/statistics/measure/BEW 2
In words: The data point has population two.



3.4 Time

Only the observations of the year 2021 should be selected. This is done step by step:

- return time (predicate: long URI).
- predicate with prefix
- year: create a new variable from the time
- select year 2021

Return time

Task: With the principle of «follow your nose» you have learned about different predicates and objects of an observation.

What is the predicate for TIME? Create a triple to return the time of the observations

Solution: predicate for TIME: https://ld.stadt-zuerich.ch/statistics/property/TIME

Triple: see code, line 15

```
1  PREFIX schema: <http://schema.org/>
2  PREFIX schemac: <https://cube.link/>
3  PREFIX property: <https://ld.stadt-zuerich.ch/statistics/property/>
4  PREFIX datacubes: <https://ld.stadt-zuerich.ch/statistics/>
5
6  SELECT DISTINCT *
7  FROM <https://lindas.admin.ch/stadtzuerich/stat>
8  WHERE {
9
10   #Observation fuer ausgewaehlten Cube
11     datacubes:000437 schemac:observationSet ?obsSet .
12     ?obsSet schemac:observation ?obs .
13
14   #Zeit
15     ?obs <https://ld.stadt-zuerich.ch/statistics/property/TIME> ?time .
16
17 }
18
19 LIMIT 10
```

Predicate with prefix

Task: Use a prefix for TIME in the predicate.

Solution: Code

Line 3: an already existing prefix can be used

Line 15: modified triple

```
1  PREFIX schema: <http://schema.org/>
2  PREFIX schemac: <https://cube.link/>
3  PREFIX property: <https://ld.stadt-zuerich.ch/statistics/property/>
4  PREFIX datacubes: <https://ld.stadt-zuerich.ch/statistics/>
5
6  SELECT DISTINCT *
7  FROM <https://lindas.admin.ch/stadtzuerich/stat>
8  WHERE {
9
10   #Observation fuer ausgewaehlten Cube
11     datacubes:000437 schemac:observationSet ?obsSet .
12     ?obsSet schemac:observation ?obs .
13
14   #Zeit
15     ?obs property:TIME ?time .
16
17 }
18
19 LIMIT 10
```

Create new variable

Task: Use time to create a new variable containing the year.

Hint: There is a function YEAR().

Solution: Code

Line 18: New variable with the year

```
1  PREFIX schema: <http://schema.org/>
2  PREFIX schemac: <https://cube.link/>
3  PREFIX property: <https://ld.stadt-zuerich.ch/statistics/property/>
4  PREFIX datacubes: <https://ld.stadt-zuerich.ch/statistics/>
5
6  SELECT DISTINCT *
7  FROM <https://lindas.admin.ch/stadtzuerich/stat>
8  WHERE {
9
10   #Observation fuer ausgewaehlten Cube
11     datacubes:000437 schemac:observationSet ?obsSet .
12     ?obsSet schemac:observation ?obs .
13
14   #Zeit
15     ?obs property:TIME ?time .
16
17   #Neue Variablen: Jahr
18     BIND(YEAR(?time) AS ?year)
19
20 }
21
22 LIMIT 10
```

Select year

Task: Select the observations of the year 2021.

Solution: Code

Line 21: Select year 2021

```
PREFIX schema: <http://schema.org/>
PREFIX schemac: <https://cube.link/>
PREFIX property: <https://ld.stadt-zuerich.ch/statistics/property/>
PREFIX datacubes: <https://ld.stadt-zuerich.ch/statistics/>

SELECT DISTINCT *
FROM <https://lindas.admin.ch/stadtzuerich/stat>
WHERE {

  #Observation fuer ausgewaehlten Cube
    datacubes:000437 schemac:observationSet ?obsSet .
    ?obsSet schemac:observation ?obs .

  #Zeit
    ?obs property:TIME ?time .

  #Neue Variablen: Jahr
    BIND(YEAR(?time) AS ?year)

  #Jahr auswaehlen
    FILTER(?year = 2021)

}

LIMIT 10
```

3.5 Space

Task: Is the data available for different space definitions? Check this with SELECT DISTINCT.

Solution: Code

There is only one space attribute; namely https://ld.stadt-zuerich.ch/statistics/code/R30000. This is the URI for the entire city of Zurich (area from the second incorporation in 1934).

Line 16: The variable ?raum is created.

Line 6: After SELECT DISTINCT only the variable ?raum is returned.

```
1  PREFIX schema: <http://schema.org/>
2  PREFIX schemac: <https://cube.link/>
3  PREFIX property: <https://ld.stadt-zuerich.ch/statistics/property/>
4  PREFIX datacubes: <https://ld.stadt-zuerich.ch/statistics/>
5
6  SELECT DISTINCT ?raum
7  FROM <https://lindas.admin.ch/stadtzuerich/stat>
8  WHERE {
9
10   #Observation fuer ausgewaehlten Cube
11     datacubes:000437 schemac:observationSet ?obsSet .
12     ?obsSet schemac:observation ?obs .
13
14   #Zeit und Raum
15     ?obs property:TIME ?time ;
16         property:RAUM ?raum .
17
18   #Neue Variablen: Jahr
19     BIND(YEAR(?time) AS ?year)
20
21   #Jahr auswaehlen
22     FILTER(?year = 2021)
23
24 }
```

3.6 First Name

Task: In addition to the observation, return the first name as text.

Hints: With «follow your nose» you can find out the triple components. In addition, you can use this concept to discover how the name is returned as text (!).

Why does it make sense to see the observation in the output in addition to the first name? Then you can always proceed with «follow your nose» to add further variables.

Solution: Code

Line 16: The variable ?name is created.

Line 19: To display the text of the name, a variable is defined with the label of the variable ?name.

```
1  PREFIX schema: <http://schema.org/>
2  PREFIX schemac: <https://cube.link/>
3  PREFIX property: <https://ld.stadt-zuerich.ch/statistics/property/>
4  PREFIX datacubes: <https://ld.stadt-zuerich.ch/statistics/>
5
6  SELECT ?nameLabel ?obs
7  FROM <https://lindas.admin.ch/stadtzuerich/stat>
8  WHERE {
9
10   #Observation fuer ausgewaehlten Cube
11     datacubes:000437 schemac:observationSet ?obsSet .
12     ?obsSet schemac:observation ?obs .
13
14   #Properties
15     ?obs property:TIME ?time ;
16         property:NAM ?name .
17
18   #Labels
19     ?name schema:name ?nameLabel .
20
21   #Neue Variablen: Jahr
22     BIND(YEAR(?time) AS ?year)
23
24   #Jahr auswaehlen
25     FILTER(?year = 2021)
26
27 }
28
29 LIMIT 10
```

3.7 Pitfalls

Task: Now add a variable with the number of persons. For this purpose, run the following code. In the results you can see that there are for example two people named Charles. Is that correct?

Solution: Certain variables are still missing for this data point. With «follow your nose» (click on an observation) you can see that this data point still has, for example, the year of birth as a property.

nameLabel	pers	obs
Charles	"2.0"^^xsd:double	https://ld.stadt-zuerich.ch/statistics/000437/obse
Charlotte	"4.0"^^xsd:double	https://ld.stadt-zuerich.ch/statistics/000437/obse
Christa	"8.0"^^xsd:double	https://ld.stadt-zuerich.ch/statistics/000437/obse

```
PREFIX schema: <http://schema.org/>
PREFIX schemac: <https://cube.link/>
PREFIX property: <https://ld.stadt-zuerich.ch/statistics/property/>
PREFIX measure: <https://ld.stadt-zuerich.ch/statistics/measure/>
PREFIX datacubes: <https://ld.stadt-zuerich.ch/statistics/>

SELECT ?nameLabel ?pers ?obs
FROM <https://lindas.admin.ch/stadtzuerich/stat>
WHERE {

  #Observation fuer ausgewaehlten Cube
    datacubes:000437 schemac:observationSet ?obsSet .
    ?obsSet schemac:observation ?obs .

  #Properties
    ?obs property:TIME ?time ;
         property:NAM ?name .

  #Labels
    ?name schema:name ?nameLabel .

  #Measure: Personen
    ?obs measure:BEW ?pers .

  #Neue Variablen: Jahr
    BIND(YEAR(?time) AS ?year)

  #Jahr auswaehlen
    FILTER(?year = 2021)

}

LIMIT 10
```

3.8 Birth Year

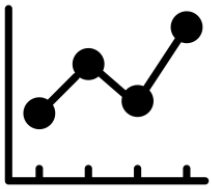
Task: Create an additional variable for the year of birth. And filter on a specific name (e.g. Charlotte).

Solution: Code

Line 21: variable for birth year

Line 33: filter (Charlotte)

```
PREFIX schema: <http://schema.org/>
PREFIX schemac: <https://cube.link/>
PREFIX property: <https://ld.stadt-zuerich.ch/statistics/property/>
PREFIX measure: <https://ld.stadt-zuerich.ch/statistics/measure/>
PREFIX datacubes: <https://ld.stadt-zuerich.ch/statistics/>

SELECT DISTINCT ?nameLabel ?gebjahr ?pers ?obs
FROM <https://lindas.admin.ch/stadtzuerich/stat>
WHERE {

  #Observation fuer ausgewaehlten Cube
    datacubes:000437 schemac:observationSet ?obsSet .
    ?obsSet schemac:observation ?obs .

  #Properties
    ?obs property:TIME ?time ;
         property:NAM ?name ;
         property:GEJ ?geburtsjahr .

  #Geburtsjahr
    ?geburtsjahr schema:position ?gebjahr .

  #Name: Label
    ?name schema:name ?nameLabel .

  #Personen
    ?obs measure:BEW ?pers .

  #Jahr
    BIND(YEAR(?time) AS ?year)

  #Auswahl
    FILTER((?year = 2021) && (?nameLabel = 'Charlotte'))

}

LIMIT 10
```



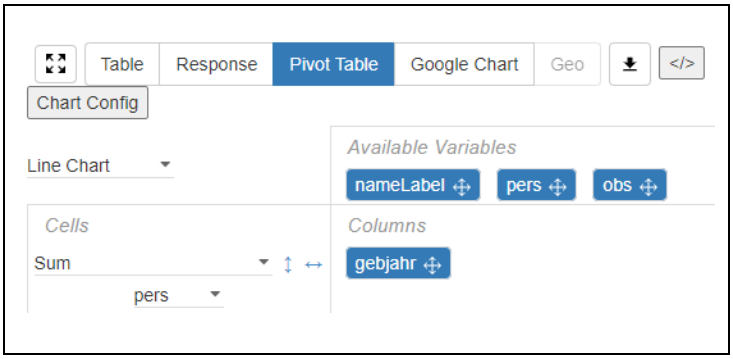
3.9 Graphic

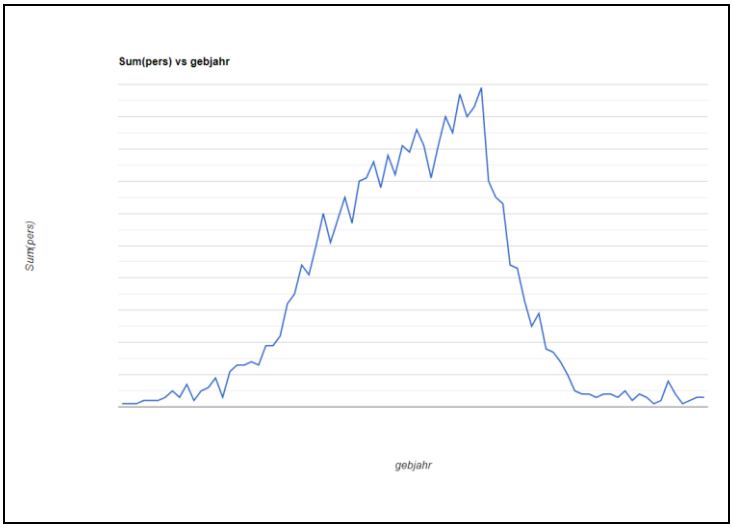
Task: Filter the name «Michael» and sort the results descending by year of birth.

In the SPARQL editor of the city of Zurich there are certain graphical tools (limited features). Create a graph with year of birth (x-axis) against number of persons (y-axis) at «Pivot Table».

Solution: Code

Line 37: Sorted by year of birth in descending order





```
PREFIX schema: <http://schema.org/>
PREFIX schemac: <https://cube.link/>
PREFIX property: <https://ld.stadt-zuerich.ch/statistics/property/>
PREFIX measure: <https://ld.stadt-zuerich.ch/statistics/measure/>
PREFIX datacubes: <https://ld.stadt-zuerich.ch/statistics/>

SELECT DISTINCT ?nameLabel ?gebjahr ?pers ?obs
FROM <https://lindas.admin.ch/stadtzuerich/stat>
WHERE {

  #Observation fuer ausgewaehlten Cube
    datacubes:000437 schemac:observationSet ?obsSet .
    ?obsSet schemac:observation ?obs .

  #Properties
    ?obs property:TIME ?time ;
        property:NAM ?name ;
        property:GEJ ?geburtsjahr .

  #Geburtsjahr
    ?geburtsjahr schema:position ?gebjahr .

  #Name: Label
    ?name schema:name ?nameLabel .

  #Personen
    ?obs measure:BEW ?pers .

  #Jahr
    BIND(YEAR(?time) AS ?year)

  #Auswahl
    FILTER((?year = 2021) && (?nameLabel = 'Michael'))

}

ORDER BY DESC(?gebjahr)
```

3.10 Additional Exercises

Own ideas

Do you have your own ideas? Try to program more analyses.

If you have any questions, please feel free to contact our tutors.

If you prefer not to work on your own queries, we have prepared more examples for you (see below).

Gender

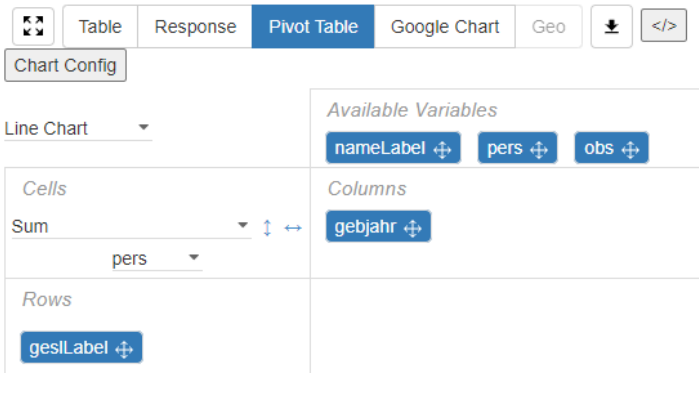
Certain first names are used by both female and male people (for example, «Andrea»). The used data come from the population register of the city of Zurich; there, the variable gender is only available in binary form (i.e. no category «diverse»).

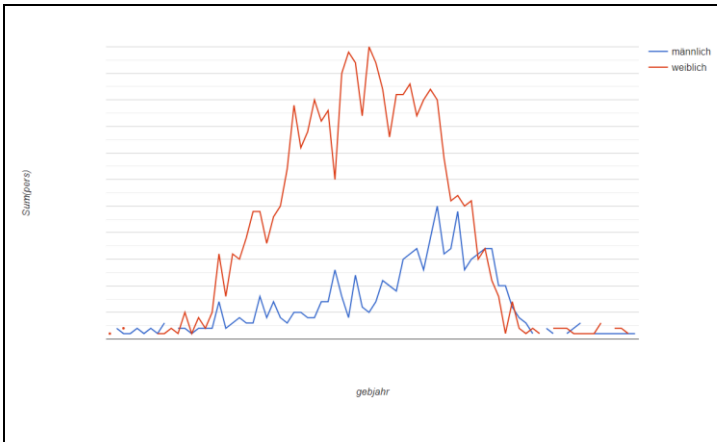
Task: Define a variable for gender. And create the same graph as before for the name «Andrea» (number of people per birth year) separated by gender.

Solution: Code

Line 18: variable for gender

Line 26: label (text) for gender





```
PREFIX schema: <http://schema.org/>
PREFIX schemac: <https://cube.link/>
PREFIX property: <https://ld.stadt-zuerich.ch/statistics/property/>
PREFIX measure: <https://ld.stadt-zuerich.ch/statistics/measure/>
PREFIX datacubes: <https://ld.stadt-zuerich.ch/statistics/>

SELECT DISTINCT ?nameLabel ?gebjahr ?geslLabel ?pers ?obs
FROM <https://lindas.admin.ch/stadtzuerich/stat>
WHERE {

  #Observation fuer ausgewaehlten Cube
    datacubes:000437 schemac:observationSet ?obsSet .
    ?obsSet schemac:observation ?obs .

  #Properties
    ?obs property:TIME ?time ;
        property:NAM ?name ;
        property:SEX ?gesl ;
        property:GEJ ?geburtsjahr .

  #Geburtsjahr
    ?geburtsjahr schema:position ?gebjahr .

  #Name: Labels
    ?name schema:name ?nameLabel .
    ?gesl schema:name ?geslLabel .

  #Personen
    ?obs measure:BEW ?pers .

  #Jahr
    BIND(YEAR(?time) AS ?year)

  #Auswahl
    FILTER((?year = 2021) && (?nameLabel = 'Andrea'))

}

ORDER BY DESC(?gebjahr)
```

Ranking

Task: What are the most frequent female names in the city of Zurich?

Hint: Sums must be calculated for this ranking. Suggestions for aggregating per group can be found at Wikibooks, for example. https://en.wikibooks.org/wiki/SPARQL/Aggregate_functions

Solution: Code

Line 35: group (GROUP BY)

Line 7: sum

```
1  PREFIX schema: <http://schema.org/>
2  PREFIX schemac: <https://cube.link/>
3  PREFIX property: <https://ld.stadt-zuerich.ch/statistics/property/>
4  PREFIX measure: <https://ld.stadt-zuerich.ch/statistics/measure/>
5  PREFIX datacubes: <https://ld.stadt-zuerich.ch/statistics/>
6
7  SELECT ?nameLabel (SUM(?pers) AS  ?total)
8  FROM <https://lindas.admin.ch/stadtzuerich/stat>
9  WHERE {
10
11   #Observation fuer ausgewaehlten Cube
12     datacubes:000437 schemac:observationSet ?obsSet .
13     ?obsSet schemac:observation ?obs .
14
15   #Properties
16     ?obs property:TIME ?time ;
17          property:NAM ?name ;
18          property:SEX ?gesl .
19
20   #Name: Labels
21     ?name schema:name ?nameLabel .
22     ?gesl schema:name ?geslLabel .
23
24   #Personen
25     ?obs measure:BEW ?pers .
26
27   #Jahr
28     BIND(YEAR(?time) AS ?year)
29
30   #Auswahl
31     FILTER((?year = 2021) && (?geslLabel = 'weiblich'))
32
33 }
34
35 GROUP BY(?nameLabel)
36 ORDER BY DESC(?total) ?nameLabel
```

	nameLabel	total
1	Maria	"3946.0"^^xsd:double
2	Anna	"2694.0"^^xsd:double
3	Laura	"1483.0"^^xsd:double
4	Claudia	"1476.0"^^xsd:double
5	Andrea	"1453.0"^^xsd:double
6	Barbara	"1417.0"^^xsd:double
7	Ursula	"1382.0"^^xsd:double
8	Sandra	"1317.0"^^xsd:double
9	Ruth	"1310.0"^^xsd:double
10	Monika	"1246.0"^^xsd:double
11	Julia	"1233.0"^^xsd:double
12	Silvia	"1217.0"^^xsd:double
13	Sarah	"1214.0"^^xsd:double
14	Elisabeth	"1187.0"^^xsd:double
15	Nicole	"1154.0"^^xsd:double
16	Daniela	"1070.0"^^xsd:double
17	Verena	"1054.0"^^xsd:double
18	Sara	"1020.0"^^xsd:double
19	Susanne	"1019.0"^^xsd:double
20	Karin	"1018.0"^^xsd:double

Federated Query, Wikidata Bubble Chart

The data on Wikidata is linked data. With the Wikidata Query Service not only Wikidata can be accessed; with so-called federated queries linked data from other endpoints can also be connected and analyzed. Federated queries show the potential of linked data: different data sources can be merged and analyzed. The more institutions provide linked data, the greater the potential benefit.

Task: Run the previous analysis (most frequent female names in the city of Zurich) with the Wikidata Query Service. Then create a bubble chart (the Wikidata Query Service offers more graphical features than the SPARQL editor of the city of Zurich).

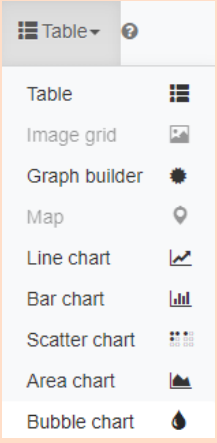
Hint: Search for information about «federated query» or «SERVICE» (example).

Solution: https://w.wiki/7EeP

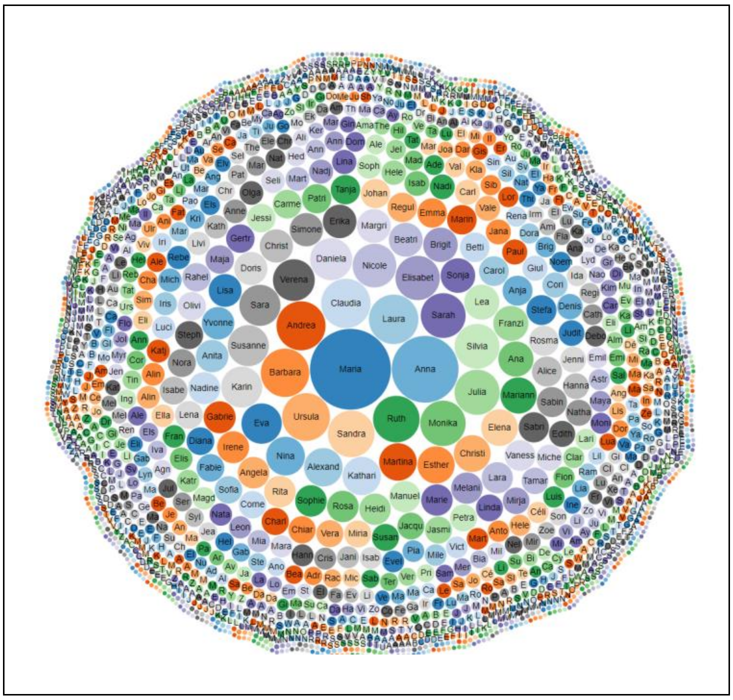
It takes about 10 seconds to run the code.

Line 11: federated query to the SPARQL endpoint of the city of Zurich.

Instead of «Table»: select «Bubble chart».



```
PREFIX schema: <http://schema.org/>
PREFIX schemac: <https://cube.link/>
PREFIX property: <https://ld.stadt-zuerich.ch/statistics/property/>
PREFIX measure: <https://ld.stadt-zuerich.ch/statistics/measure/>
PREFIX datacubes: <https://ld.stadt-zuerich.ch/statistics/>

SELECT ?nameLabel (SUM(?pers) AS  ?total)

WHERE {

  SERVICE <https://ld.stadt-zuerich.ch/query> {

  #Observation fuer ausgewaehlten Cube
    datacubes:000437 schemac:observationSet ?obsSet .
    ?obsSet schemac:observation ?obs .

  #Properties
    ?obs property:TIME ?time ;
        property:NAM ?name ;
        property:SEX ?gesl .

  #Name: Labels
    ?name schema:name ?nameLabel .
    ?gesl schema:name ?geslLabel .

  #Personen
    ?obs measure:BEW ?pers .

  #Jahr
    BIND(YEAR(?time) AS ?year)

  #Auswahl
    FILTER((?year = 2021) && (?geslLabel = 'weiblich'))

    #SERVICE Ende
  }


}

GROUP BY(?nameLabel)
ORDER BY DESC(?total) ?nameLabel
```



Federated Query, HAVING

Task: In the Wikidata Query Service, select those women's names that occur at least 1000 times and use them to create a bar chart.

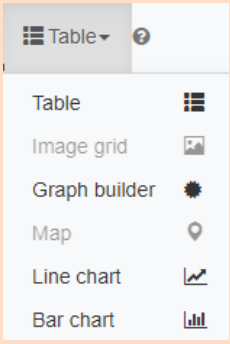
Hint: This is about filtering an aggregated quantity. HAVING is helpful for this (example on wikibooks.org).

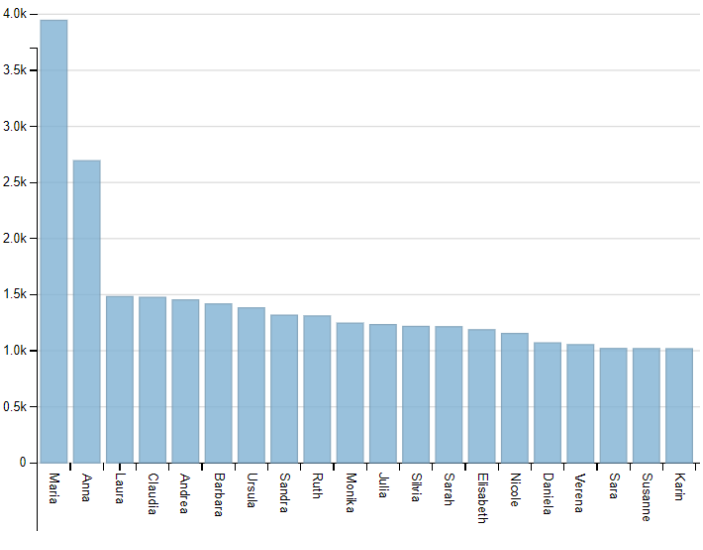
Solution: https://w.wiki/7EeS

The bar chart clearly shows that Maria and Anna are much more frequent in the city of Zurich compared to the other female names.

line 42: HAVING

Instead of «Table»: select «Bar chart».





```
PREFIX schema: <http://schema.org/>
PREFIX schemac: <https://cube.link/>
PREFIX property: <https://ld.stadt-zuerich.ch/statistics/property/>
PREFIX measure: <https://ld.stadt-zuerich.ch/statistics/measure/>
PREFIX datacubes: <https://ld.stadt-zuerich.ch/statistics/>

SELECT ?nameLabel (SUM(?pers) AS  ?total)

WHERE {

  SERVICE <https://ld.stadt-zuerich.ch/query> {

  #Observation fuer ausgewaehlten Cube
    datacubes:000437 schemac:observationSet ?obsSet .
    ?obsSet schemac:observation ?obs .

  #Properties
    ?obs property:TIME ?time ;
        property:NAM ?name ;
        property:SEX ?gesl .

  #Name: Labels
    ?name schema:name ?nameLabel .
    ?gesl schema:name ?geslLabel .

  #Personen
    ?obs measure:BEW ?pers .

  #Jahr
    BIND(YEAR(?time) AS ?year)

  #Auswahl
    FILTER((?year = 2021) && (?geslLabel = 'weiblich'))

    #SERVICE Ende
  }


}

GROUP BY(?nameLabel)
HAVING (?total >= 1000)
ORDER BY DESC(?total) ?nameLabel
```

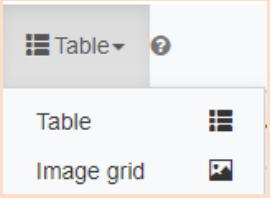
Federated Query, Wikidata Portraits

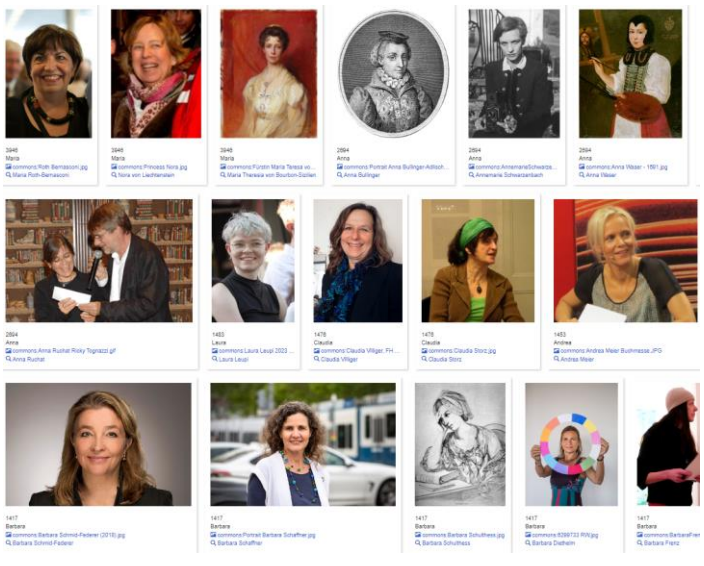
Do you know any Zurich born women who have popular first names?

Task: Select the ten most frequent female names in the city of Zurich. Connect women with these first names, who are on Wikidata, and were born in Zurich and have a picture (portrait) on Wikidata. Display these images.

Solution: Unfortunately there is no short URL for this Wikidata query. Therefore, copy the code on the next page into the Wikidata Query Service.

Instead of «Table»: select «Image grid».

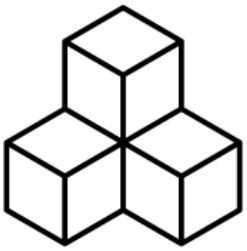




```
PREFIX schema: <http://schema.org/>
PREFIX schemac: <https://cube.link/>
PREFIX property: <https://ld.stadt-zuerich.ch/statistics/property/>
PREFIX measure: <https://ld.stadt-zuerich.ch/statistics/measure/>
PREFIX datacubes: <https://ld.stadt-zuerich.ch/statistics/>
PREFIX code: <https://ld.stadt-zuerich.ch/statistics/code/>

#Wikidata

SELECT ?nameLabel ?total ?person ?personLabel ?pic

WHERE {

  ?person wdt:P21 wd:Q6581072 ;
          wdt:P19 wd:Q72 ;
          wdt:P18 ?pic ;
          wdt:P735 ?name .
  ?name rdfs:label ?nameLabel .

  FILTER((LANG(?nameLabel)) = "de")
  FILTER((STR(?nameLabel)) = ?nameLabelSSZ)


#Stadt Zuerich

  {

    SELECT ?nameLabelSSZ (SUM(xsd:integer(?pers)) AS ?total)

      WHERE {

      SERVICE <https://ld.stadt-zuerich.ch/query> {

        datacubes:000437 schemac:observationSet ?obsSet .
        ?obsSet schemac:observation ?obs .
        ?obs property:TIME ?time ;
             property:NAM ?name ;
             property:SEX code:SEX0002 ;
             measure:BEW ?pers .
        ?name schema:name ?nameLabelSSZ .

        BIND(YEAR(?time) AS ?year)
        FILTER(?year = 2021)

      }
    }
    GROUP BY(?nameLabelSSZ)
    ORDER BY DESC(?total)
    LIMIT 10

  }

  SERVICE wikibase:label { bd:serviceParam wikibase:language "de, en,
fr, it". }

}

ORDER BY DESC(?total)
```

```
PREFIX schema: <http://schema.org/>
PREFIX schemac: <https://cube.link/>
PREFIX property: <https://ld.stadt-zuerich.ch/statistics/property/>
PREFIX measure: <https://ld.stadt-zuerich.ch/statistics/measure/>
PREFIX datacubes: <https://ld.stadt-zuerich.ch/statistics/>
PREFIX code: <https://ld.stadt-zuerich.ch/statistics/code/>

#Wikidata

SELECT ?nameLabel ?total ?person ?personLabel ?pic

WHERE {

  ?person wdt:P21 wd:Q6581072 ;
          wdt:P19 wd:Q72 ;
          wdt:P18 ?pic ;
          wdt:P735 ?name .
  ?name rdfs:label ?nameLabel .

  FILTER((LANG(?nameLabel)) = "de")
  FILTER((STR(?nameLabel)) = ?nameLabelSSZ)


#Stadt Zuerich

  {

    SELECT ?nameLabelSSZ (SUM(xsd:integer(?pers)) AS ?total)

      WHERE {

      SERVICE <https://ld.stadt-zuerich.ch/query> {

        datacubes:000437 schemac:observationSet ?obsSet .
        ?obsSet schemac:observation ?obs .
        ?obs property:TIME ?time ;
             property:NAM ?name ;
             property:SEX code:SEX0002 ;
             measure:BEW ?pers .
        ?name schema:name ?nameLabelSSZ .

        BIND(YEAR(?time) AS ?year)
        FILTER(?year = 2021)

      }
    }
    GROUP BY(?nameLabelSSZ)
    ORDER BY DESC(?total)
    LIMIT 10

  }

  SERVICE wikibase:label { bd:serviceParam wikibase:language "de, en, fr, it". }

}

ORDER BY DESC(?total)
```

Multiple Cubes, births and population



How many newborns have a certain name? How many people in the population have the same name? To answer this, connect the two cubes **births** and **population** (see chapter **Fehler! Verweisquelle konnte nicht gefunden werden.**, search cube).

Task: What were the 20 most frequenz male given names (first name) for newborns in 2021? Calculate for these per name:

- number of **births** in 2021
- number of people in the **population** at the end of 2021
- what **percentage** do newborns make up of the population in 2021?

Solution: Code.

	nameLabel	geburten	personen	prozent
1	Noah	"27"^^xsd:integer	"413"^^xsd:integer	"7"^^xsd:decimal
2	Leo	"24"^^xsd:integer	"387"^^xsd:integer	"6"^^xsd:decimal
3	Louis	"22"^^xsd:integer	"369"^^xsd:integer	"6"^^xsd:decimal
4	Theo	"20"^^xsd:integer	"124"^^xsd:integer	"16"^^xsd:decimal
5	Levi	"18"^^xsd:integer	"106"^^xsd:integer	"17"^^xsd:decimal
6	Luca	"18"^^xsd:integer	"960"^^xsd:integer	"2"^^xsd:decimal
7	Mateo	"18"^^xsd:integer	"118"^^xsd:integer	"15"^^xsd:decimal
8	Luis	"17"^^xsd:integer	"453"^^xsd:integer	"4"^^xsd:decimal
9	Maximilian	"17"^^xsd:integer	"501"^^xsd:integer	"3"^^xsd:decimal
10	Felix	"16"^^xsd:integer	"827"^^xsd:integer	"2"^^xsd:decimal
11	Finn	"16"^^xsd:integer	"183"^^xsd:integer	"9"^^xsd:decimal
12	Leonardo	"16"^^xsd:integer	"284"^^xsd:integer	"6"^^xsd:decimal
13	Liam	"16"^^xsd:integer	"202"^^xsd:integer	"8"^^xsd:decimal
14	Daniel	"15"^^xsd:integer	"3052"^^xsd:integer	"0"^^xsd:decimal
15	Elias	"15"^^xsd:integer	"324"^^xsd:integer	"5"^^xsd:decimal
16	Jan	"15"^^xsd:integer	"981"^^xsd:integer	"2"^^xsd:decimal
17	Vincent	"15"^^xsd:integer	"357"^^xsd:integer	"4"^^xsd:decimal
18	Alexander	"14"^^xsd:integer	"1399"^^xsd:integer	"1"^^xsd:decimal
19	Gabriel	"14"^^xsd:integer	"500"^^xsd:integer	"3"^^xsd:decimal
20	Leon	"14"^^xsd:integer	"331"^^xsd:integer	"4"^^xsd:decimal

Example for **Theo**: 20 were born in 2021. At the end of the year, there were 124 Theos in the population. The 2021 births accounted for 16 percent.

Example for **Alexander**: 14 were born in 2021. At the end of the year, the population had 1399. 2021 births accounted for 1 percent.

```
PREFIX schema: <http://schema.org/>
PREFIX schemac: <https://cube.link/>
PREFIX property: <https://ld.stadt-zuerich.ch/statistics/property/>
PREFIX measure: <https://ld.stadt-zuerich.ch/statistics/measure/>
PREFIX datacubes: <https://ld.stadt-zuerich.ch/statistics/>
PREFIX code: <https://ld.stadt-zuerich.ch/statistics/code/>

SELECT ?nameLabel ?geburten ?personen
FROM <https://lindas.admin.ch/stadtzuerich/stat>
WHERE {
  {
    SELECT ?name (SUM(?geb) AS ?geburten)
    WHERE {

  #Geburten (G): Observations
    datacubes:000361 schemac:observationSet ?obsSetG .
    ?obsSetG schemac:observation ?obsG .

  #Properties
    ?obsG property:TIME "2021-12-31"^^xsd:date ;
        property:SEX code:SEX0001 ;
        property:NAF code:NAF0001 ;
        property:NAM ?name .

  #Geburten
    ?obsG measure:GEB ?geb .

  }
    GROUP BY ?name
  }
  OPTIONAL
  {
    SELECT ?name  (SUM(?pers) AS ?personen)
  WHERE {

  #Bevoelkerung (B): Observations
    datacubes:000437 schemac:observationSet ?obsSetB .
    ?obsSetB schemac:observation ?obsB .

  #Properties
    ?obsB property:TIME "2021-12-31"^^xsd:date ;
        property:SEX code:SEX0001 ;
        property:NAM ?name .

  #Personen
    ?obsB measure:BEW ?pers .

}
    GROUP BY ?name

  }

  #Labels
    ?name schema:name ?nameLabel .

}

ORDER BY DESC(?geburten)
```

4 Appendix

4.1 Who makes linked data?

linked open data

In Switzerland, several institutions publish data as linked data, including the following:

- City of Zurich: www.stadt-zuerich.ch/lod
- Canton of Basel-Stadt (Link)
- Swiss Federal Railways (SFR, Link)
- Other institutions in Switzerland (Link)

	CREATOR	CREATORNAME
1	<https://register.ld.admin.ch/opendataswiss/org/bundesamt-fur-energie-bfe>	"Bundesamt für Energie BFE"@de
2	<https://register.ld.admin.ch/opendataswiss/org/bundesamt-fur-gesundheit-bag>	"Bundesamt für Gesundheit BAG"@de
3	<https://ld.admin.ch/org/fsvo>	"Bundesamt für Lebensmittelsicherheit und Veterinärwesen"@de
4	<https://register.ld.admin.ch/opendataswiss/org/bundesamt-fur-statistik-bfs>	"Bundesamt für Statistik BFS"@de
5	<https://ld.admin.ch/office/VII.1.7>	"Bundesamt für Umwelt"@de
6	<https://register.ld.admin.ch/opendataswiss/org/bundesamt-fur-umwelt-bafu>	"Bundesamt für Umwelt BAFU"@de
7	<https://register.ld.admin.ch/staatskalender/organization/10008758>	"Bundesamt für Umwelt BAFU"@de
8	<https://ld.admin.ch/FCh>	"Bundeskanzlei"@de
9	<https://register.ld.admin.ch/staatskalender/organization/20030954>	"Eidgenössische Elektrizitätskommission"@de
10	<https://register.ld.admin.ch/staatskalender/organization/10002463>	"Eidgenössisches Amt für das Handelsregister"@de
11	<https://register.ld.admin.ch/opendataswiss/org/efv_finanzstatistik>	"Finanzstatistik EFV"@de
12	<https://register.ld.admin.ch/staatskalender/organization/10004387>	"Schweizerische Nationalbibliothek"@de
13	<https://ld.admin.ch/office/II.1.4>	"Schweizerisches Bundesarchiv"@de
14	<https://register.ld.admin.ch/opendataswiss/org/schweizerisches-bundesarchiv-bar>	"Schweizerisches Bundesarchiv BAR"@de
15	<https://register.ld.admin.ch/opendataswiss/org/staatsarchiv-kanton-zuerich>	"Staatsarchiv Kanton Zürich"@de
16	<https://ld.stadt-zuerich.ch/org/552>	"Statistik Stadt Zürich"@de

Figure 15: who makes linked data?

4.2 Glossary

Federated Query	Linked query. In one query data from different data sources are combined.
LD	linked data
LOD	linked open data (with open license)
RDF	Resource Description Framework – **Resource**: Everything that can be uniquely identified with a URI – **Description**: Description of the resources: attributes, properties – **Framework**: Framework for these descriptions
Semantics	Meaning/content of a word, sentence, or text
SPARQL	SPARQL Protocol And RDF Query Language
URI	Uniform Resource Identifier. Unique identification of real or abstract things (web pages, people, products, locations, properties, relationships, etc.)
URL	Uniform Resource Locator (of web pages)

4.3 Image References

Figure 1: examples of unclear semantics: palm, jaguar, and bat	https://en.wikipedia.org/wiki/Archontophoenix_cunninghamiana https://commons.wikimedia.org/wiki/File:Human_Palm.png https://en.wikipedia.org/wiki/Jaguar#/media/File:Standing_jaguar.jpg https://de.wikipedia.org/wiki/Jaguar_X350 https://commons.wikimedia.org/wiki/File:Baseball_bat_1.jpg https://commons.wikimedia.org/wiki/File:Short-tailed_Fruit_Bat_%2816322859359%29.jpg
Figure 2: triple	Own illustration
Figure 3: graph	Own illustration
Figure 4: URI vs. URL	Own illustration; based on «Linked Data Engineering, 1.6. How to name Things», https://open.hpi.de/courses/semanticweb2016/items/5F6Tk3MwjAkAiNSaqEwERD
Figure 5: SELECT, WHERE	Own illustration; based on Wikidata tutorial: https://www.wikidata.org/wiki/Wikidata:SPARQL_tutorial/de
Figure 6: dots at the end of the triple	Own illustration; based on Wikidata tutorial: https://www.wikidata.org/wiki/Wikidata:SPARQL_tutorial/de
Figure 7: semicolon, comma, dot	Own illustration; based on Wikidata tutorial: https://www.wikidata.org/wiki/Wikidata:SPARQL_tutorial/de
Figure 8: LIMIT, ORDER BY	Own illustration; based on Wikidata tutorial: https://www.wikidata.org/wiki/Wikidata:SPARQL_tutorial/de

Stadt Zürich
Präsidialdepartement
Statistik
Napfgasse 6
8001 Zürich
T+ 41 44 412 08 00
statistik@zuerich.ch
stadt-zuerich.ch/statistik