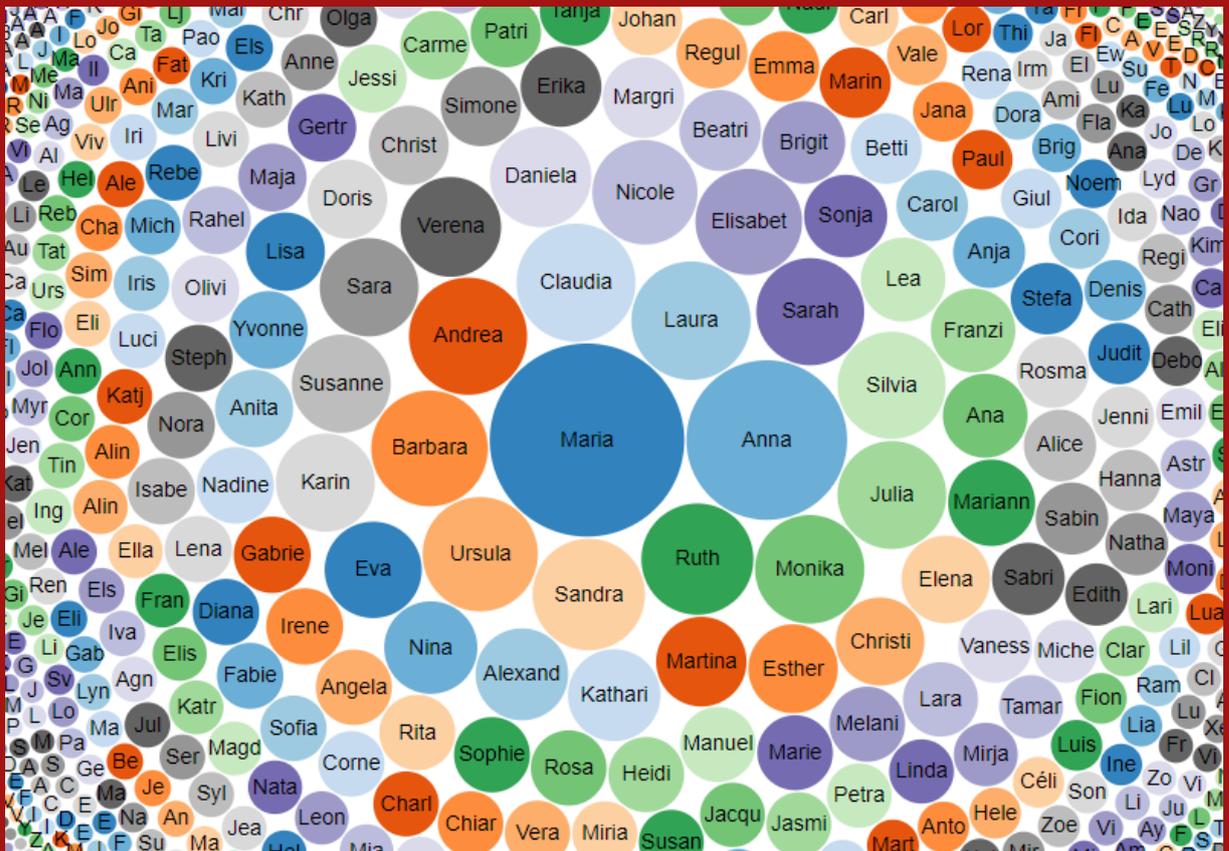




Linked Data Workshop



Judith Riegelnic
Klemens Rosin
Rolf Schenker
Hansjörg Stalder
Daniel Truttmann

Zürich, 22. November 2023

Impressum

Herausgeberin

Stadt Zürich

Statistik

Postfach, 8022 Zürich

stadt-zuerich.ch/statistik

T +41 44 412 08 00

Inhaltsverzeichnis

1	Einführung	4
2	Theorie	5
2.1	Konzepte	5
2.2	Begriffe	6
2.3	SPARQL	7
2.4	Datenstrukturen	10
3	Praxis	12
3.1	Cube suchen	13
3.2	Startpunkt: Observations	14
3.3	Follow your nose	15
3.4	Zeit	16
3.5	Raum	20
3.6	Vorname	21
3.7	Vorsicht	22
3.8	Geburtsjahr	23
3.9	Grafik	24
3.10	Zusätzliches	26
4	Anhang	40
4.1	Wer macht Linked Data?	40
4.2	LinkedDataGPT	40
4.3	Glossar	41
4.4	Bildquellen	42

1 Einführung

Beginner-Level

Der Linked Data Workshop richtet sich an diejenigen, die bisher wenig oder keine Kenntnisse zu Linked Data haben.

Vorgehen und Daten

Nach einem kurzen Theorie-Block dürfen Sie selbst üben. Dabei werden Sie von Tutor*innen unterstützt. Wir arbeiten heute mit Vornamensdaten der Stadt Zürich.

2 Theorie

2.1 Konzepte

Semantik

Für Maschinen ist die Bedeutung von Texten, die **Semantik**, schwer erfassbar. Das ist insbesondere dann der Fall, wenn weitere Informationen zum **Kontext** fehlen. So ist beispielsweise für Maschinen unklar, was bei den Wörtern «Bank», «Erde» und «Jaguar» gemeint ist (Homonyme).



Abbildung 1: Beispiele zu unklarer Semantik: Bank, Erde und Jaguar

Triples

Bei Linked Data werden Beziehungen oder Eigenschaften mit sogenannten Triples beschrieben. Ein Triple besteht aus **Subjekt**, **Prädikat** und **Objekt**.

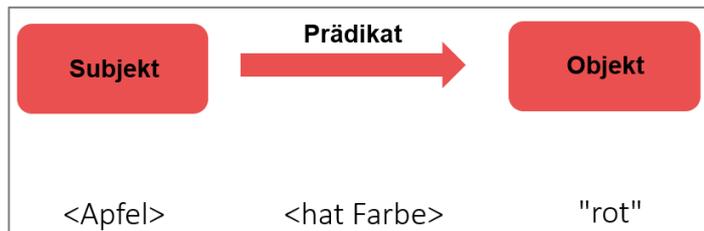


Abbildung 2: Triple

Graphen

Mehrere Triples können verbunden werden; das ergibt einen **Graphen**.

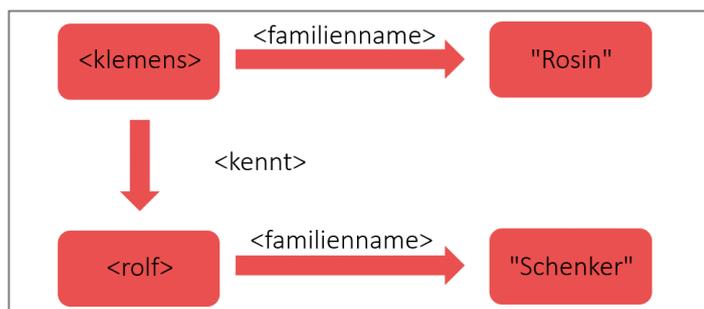


Abbildung 3: Graph

2.2 Begriffe

URI

Bei Linked Data ist die eindeutige Identifizierung wichtig. Dazu werden **Uniform Resource Identifiers (URI)** verwendet. Damit können sowohl reale wie auch abstrakte Sachen eindeutig identifiziert werden; zum Beispiel Webseiten, Personen, Produkte, Standorte, Eigenschaften oder Beziehungen.

Auch in anderen Bereichen spielt die eindeutige Identifizierung eine zentrale Rolle. Das ist beispielsweise bei Literatur der Fall; dort wird die **ISBN** (International Standard Book Number) verwendet. Ein anders Beispiel sind Webseiten und die **URL** (Uniform Resource Locator). URI sind jedoch umfassender als URL (Abbildung 4).



Abbildung 4: URI vs. URL

RDF

Die **Struktur von Linked Data** sind nicht Tabellen wie bei klassischen Datenbanken, sondern **RDF**. Diese Abkürzung steht für Resource Description Framework.

- **Resource**: Alles, was eindeutig mit einem URI identifiziert werden kann.
- **Description**: Beschreibung der Ressourcen: Attribute, Beziehungen
- **Framework**: Gerüst für diese Beschreibungen

SPARQL

SPARQL ist die **Abfragesprache für Linked Data**. Die Abkürzung steht für SPARQL Protocol And RDF Query Language (SPARQL Protokoll- und RDF-Abfragesprache). Es ist also eine rekursive gebildetes Wort. SPARQL beinhaltet ähnliche Elemente wie die Programmiersprache SQL.

2.3 SPARQL

Einführung

Dieses Kapitel enthält keine umfassende SPARQL-Schulung. Es beinhaltet nur **elementare Code-Elemente**, sodass Sie im Praxisteil mit SPARQL arbeiten können.

Weitere Informationen zu SPARQL

Weiterführende Informationen zu SPARQL können Sie unter anderem hier finden:

- SPARQL auf Wikibooks ([Link](#)), u.a. Funktionen ([Link](#))
- SPARQL auf W3C ([Link](#))
- Wikidata SPARQL-Tutorial (Deutsch; [Link](#))
- data.world SPARQL-Tutorial ([Link](#))
- Linked Data Engineering: SPARQL (Video-Serie, [Link](#))

SELECT, WHERE

In SPARQL wird bei Variablennamen jeweils ein Fragezeichen vorangestellt. Das WHERE-Statement beinhaltet die Tripels. Welche Variablen ausgegeben werden, wird im SELECT-Statement angegeben. Ein Stern (*) bedeutet die Ausgabe aller Variablen. Mit SELECT DISTINCT werden nur unterschiedliche Records (also Zeilen) ausgegeben.

```
SELECT ?frucht
WHERE
{
  ?frucht <hatFarbe> "gelb" .
  ?frucht <schmeckt> "bitter" .
}
```

Abbildung 5: SELECT, WHERE

Satzzeichen

Ein Triple wird jeweils mit einem **Punkt** abgeschlossen. Zwecks besserer Lesbarkeit ist es üblich, dass in SPARQL vor den Satzzeichen ein Leerschlag verwendet wird.

```
?frucht <hatFarbe> "gelb" .
?frucht <schmeckt> "bitter" .
?frucht <schmeckt> "fruchtig" .
```

Abbildung 6: Punkt am Triple-Ende

Falls mehrere Triples dasselbe **Subjekt** haben, kann im Code nur das erste Subjekt geschrieben und dafür zwischen den Triples ein **Strichpunkt** verwendet werden.

Wenn **Subjekt und Prädikat** gleich sind, können diese in den folgenden Triples weggelassen werden; die Triples werden dafür mit einem **Komma** getrennt.

```
?frucht <hatFarbe> "gelb" ;
      <schmeckt> "bitter" ,
      "fruchtig" .
```

Abbildung 7: Strichpunkt, Komma, Punkt

Ausgabe: LIMIT, ORDER BY

Mit **LIMIT** wird die Anzahl Records (Zeilen) in der Ausgabe eingeschränkt. Zum Beispiel werden mit LIMIT 10 bloss zehn Zeilen ausgegeben. Das ist insbesondere bei Tests hilfreich. MIT **ORDER BY** können die ausgegebenen Zeilen sortiert werden.

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

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

Abbildung 8: LIMIT, ORDER BY

BIND, FILTER

Mit BIND werden neue Spalten hinzugefügt. Die YEAR-Funktion wandelt ein Datum zu einer Jahreszahl um. Mit dem FILTER-Statement werden gewisse Zeilen ausgewählt.

```
SELECT *
WHERE
{
    BIND(YEAR(?datum) AS ?jahr)

    FILTER(?jahr = 2023)
}
```

Abbildung 9: BIND, YEAR, FILTER

PREFIX

URIs können aus relativ vielen Zeichen bestehen. Daher lohnt es sich, einen Teil des URI als **Prefix** (Deutsch: Präfix) zu Beginn des Codes festzulegen. Die Präfixe können anschliessend im SPARQL-Code wieder verwendet werden. Der Code wird dadurch lesbarer.

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

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

```
SELECT *
WHERE
{
    ?obs property:TIME ?zeit ;
    property:RAUM ?raum .
}
```

Abbildung 10: PREFIX

2.4 Datenstrukturen

Datenpunkte beschreiben Im Grundangebot von Statistik Stadt Zürich ([Webseite](#) und [Open Data Plattform](#)) sind Daten zu den Vornamen der Wohnbevölkerung nach Jahrgang zu finden. Auf der Webseite ist beispielsweise ersichtlich, dass es gemäss Bevölkerungsregister Ende 2022 in der Stadt Zürich fünf männliche Personen mit dem 1. Vornamen «Charles» und Jahrgang 1997 gab, die zur [wirtschaftlichen Wohnbevölkerung](#) zählen (Abbildung 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	0
Chi	0	0	0	1	0
Chris	0	1	0	1	2
Christian	29	22	14	10	10
Christof	1	0	0	0	0

Abbildung 11: Bisherige Daten

Bei Linked Data werden die einzelnen **Datenpunkte** und deren **Eigenschaften** beschrieben (Abbildung 12).

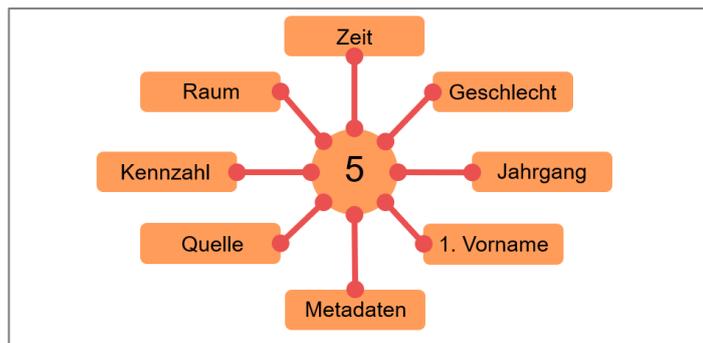


Abbildung 12: Datenpunkte beschreiben

Hierarchische Datenstrukturen

Damit Sie sich in den Daten orientieren können, hilft es, deren Strukturen zu kennen. Linked Data sind hierarchisch geordnet: Innerhalb eines **Graphs** befinden sich mehrere **Cubes**. Die **Cubes** können mehrere **Observation Sets** beinhalten; allerdings hat es aktuell bei Statistik Stadt Zürich in jedem Cube nur ein Observation Set. In den Observation Sets befinden sich die einzelnen **Observations** (Beobachtungen; Datenpunkte).

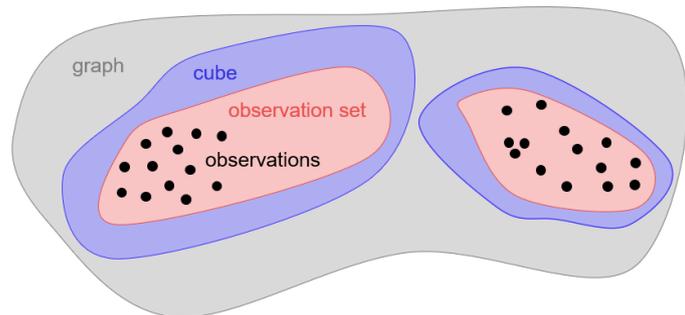


Abbildung 13: Datenstrukturen

3 Praxis

Praxisteil

Der Praxisteil folgt einer bestimmten «Route»: Zuerst wird ein Cube ausgewählt, die Auswertungen werden verfeinert und letztendlich die Ergebnisse grafisch dargestellt.

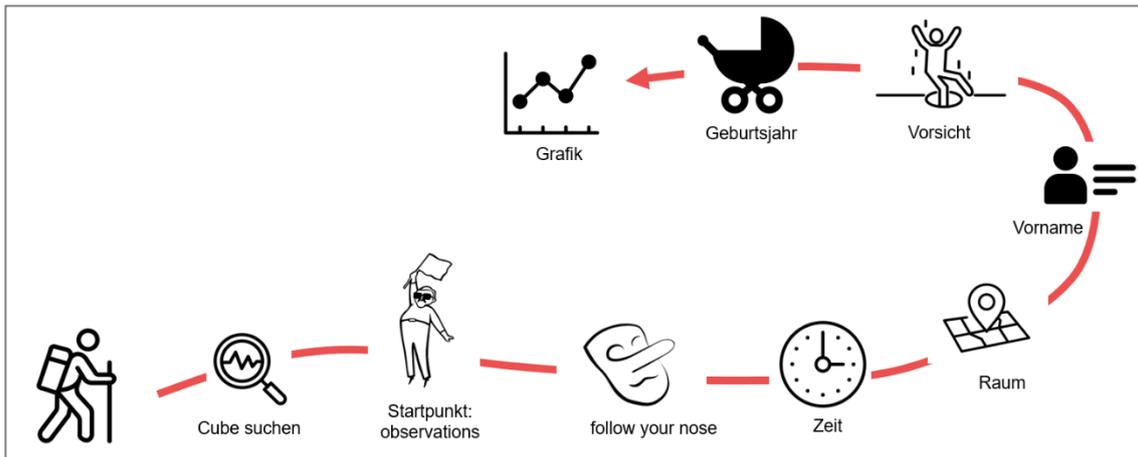


Abbildung 14: Praxis

Zusätzliches

Falls Sie möchten, können Sie anschliessend eigene Ideen verfolgen und weitere Auswertungen durchführen. Zudem haben wir zusätzliche Aufgaben bereitgestellt:

- Geschlecht
- Ranking
- Federated Query: Wikidata Bubble Chart
- Federated Query: HAVING
- Federated Query: Wikidata Porträts



3.1 Cube suchen

Im Praxisteil wird mit Vornamen-Daten der Stadt Zürich gearbeitet. Es gibt drei Cubes, die Informationen zu Namen enthalten.

Aufgabe: Mit dem [Link](#) gelangen Sie zum sogenannten SPARQL-Editor der Stadt Zürich. Von dort können Abfragen auf dem SPARQL-Endpoint der Stadt Zürich ausgeführt werden. Lassen Sie nun das Query laufen (schwarze Pfeiltaste oben rechts).

Wir möchten mit Daten zu den ersten Vornamen der Wohnbevölkerung arbeiten. Wie lautet die Cube-Nummer?

Hinweis: Es ist nicht notwendig, dass Sie diesen SPARQL-Code verstehen. Sie werden später Codes detailliert anschauen.

```

1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX schema: <http://schema.org/>
3 PREFIX schemac: <https://cube.link/>
4
5 SELECT DISTINCT ?cu ?cuLabel
6 FROM <https://lindas.admin.ch/stadtzuerich/stat>
7 WHERE {
8
9     #Cubes und Cube-Labels
10    ?cu rdf:type schemac:Cube ;
11        schema:name ?cuLabel .
12
13    #Text filtern
14    FILTER REGEX(LCASE(STR(?cuLabel)), "name") .
15
16 }
17
18 ORDER BY ?cuLabel
    
```

Lösung: Der Cube mit den Vornamen der Wohnbevölkerung heisst <https://ld.stadt-zuerich.ch/statistics/000437>.

3.2 Startpunkt: Observations



Startpunkt: Zu Beginn von Linked Data Auswertungen ist es hilfreich, einzelne Observations eines bestimmten Cubes zu betrachten.

Aufgabe: Lassen Sie [diesen Code](#) laufen, um zehn Observations des Cubes 000437 anzuzeigen.

Versuchen Sie möglichst viele Elemente des Codes zu verstehen. Wie ist dieser Code aufgebaut?

```

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): ObservationSet
10    datacubes:000437 schemac:observationSet ?obsSet .
11
12    #Observations des ObservationSets
13    ?obsSet schemac:observation ?obs .
14
15 }
16
17 LIMIT 10
    
```

Lösung: Aufbau des Codes.

- PREFIXES: zu Beginn werden einige Prefixes festgelegt, damit im Code die URIs nicht zu lange sind
- SELECT DISTINCT *: alle Variablen ausgeben (aber nur unterschiedliche Zeilen)
- FROM: Graph der Stadt Zürich
- WHERE: Beinhaltet Triples
- Zeile 10: Cube 000437 beinhaltet Observation Set (Variablenname: ?obsSet)
- Zeile 13: Das Observation Set beinhaltet Observations (Variablenname: ?obs)
- LIMIT 10: zehn Observations ausgeben

3.3 Follow your nose



In Linked Data lautet ein hilfreiches Prinzip: Immer der Nase nach («follow your nose»). So lernen Sie beispielsweise die Eigenschaften der Observations kennen.

Aufgabe: Im vorangehenden Code haben Sie Observations ausgegeben. Wählen Sie zufällig eine Beobachtung aus; zum Beispiel [diese](#).

Was steht in der linken Spalte? Was in der rechten Spalte? Welche Eigenschaften hat dieser Datenpunkt?

Hinweis: in neuem Tab öffnen mit «ctrl + klick»

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

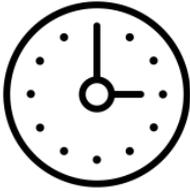
Lösung: In der linken Spalte stehen die URIs der Prädikate der Triples; in der rechten Spalte die URIs der Objekte.

Ein Beispiel zur linken Spalte: BEW steht für <https://ld.stadt-zuerich.ch/statistics/measure/BEW> (hat Wohnbevölkerung)

Ein Beispiel zur rechten Spalte: 2

Das Triple lautet: ?obs <https://ld.stadt-zuerich.ch/statistics/measure/BEW> 2

In Worten: Der Datenpunkt hat Wohnbevölkerung zwei.



3.4 Zeit

Es sollen bloss die Observations des Jahres 2021 gewählt werden. Dabei wird schrittweise vorgegangen:

- Zeit ausgeben (Prädikat: lange URI)
- Prädikat mit Prefix
- Jahr: aus der Zeit eine neue Variable erstellen
- Jahr 2021 auswählen

Zeit ausgeben

Aufgabe: Mit dem Prinzip von «follow your nose» haben Sie verschiedene Prädikate und Objekte einer Observation kennengelernt.

Wie lautet das Prädikat für ZEIT? Erstellen Sie damit ein Triple, um die Zeit der Observations auszugeben.

Lösung: Prädikat für ZEIT: <https://ld.stadt-zuerich.ch/statistics/property/TIME>

Triple: siehe [Code](#), Zeile 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
    
```

Prädikat mit Prefix

Aufgabe: Verwenden Sie beim Prädikat für ZEIT einen Prefix.

Lösung: [Code](#)

Zeile 3: es kann ein bereits bestehender Prefix genutzt werden

Zeile 15: Angepasstes 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
```

Neue Variable erstellen

Aufgabe: Erstellen Sie aus der Zeit eine neue Variable mit dem Jahr.

Hinweis: Es gibt die Funktion YEAR().

Lösung: [Code](#)

Zeile 18: Neue Variable mit dem Jahr

```
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 ausgewählten 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
```

**Jahr
auswählen**

Aufgabe: Wählen Sie die Beobachtungen des Jahres 2021 aus.

Hinweis: Verwenden Sie die FILTER-Funktion.

Lösung: [Code](#)

Zeile 21: Jahr 2021 auswählen

```
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 ausgewählten 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 #Jahr auswaehlen
21   FILTER(?year = 2021)
22
23 }
24
25 LIMIT 10
```



3.5 Raum

Aufgabe: Sind die Daten für verschiedene Raum-Ausprägungen vorhanden? Prüfen Sie das mit SELECT DISTINCT.

Lösung: [Code](#)

Es kommt nur eine Ausprägung vor; nämlich <https://ld.stadt-zuerich.ch/statistics/code/R30000>. Das ist der URI für die Stadt Zürich (Gebiet ab der zweiten Eingemeindung im Jahr 1934).

Zeile 16: Es wird die Variable ?raum erstellt.

Zeile 6: Nach SELECT DISTINCT wird nur die Variable ?raum ausgegeben.

```

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 Vorname

Aufgabe: Geben Sie zusätzlich zu der Observation den Namen als Text aus.

Hinweise: Mit «follow your nose» können Sie herausfinden, wie das Triple zum Namen lauten soll. Zudem können Sie damit entdecken, wie der Name als Text (!) ausgegeben wird.

Warum ist es sinnvoll in der Ausgabe zusätzlich zum Vornamen die Observation zu sehen? Dann kann mit «follow your nose» vorgegangen werden, um weitere Variablen hinzuzufügen.

Lösung: [Code](#)

Zeile 16: Es wird die Variable ?name erstellt.

Zeile 19: Damit der Text des Namens zu sehen ist, wird eine Variable mit dem Label der Variable ?name definiert.

```

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 ausgewählten 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 auswählen
25 FILTER(?year = 2021)
26
27 }
28
29 LIMIT 10

```



3.7 Vorsicht

Aufgabe: Nun kommt eine Variable mit der Anzahl Personen hinzu. Führen Sie dazu den folgenden [Code](#) aus. In den Ergebnissen ist beispielsweise zu sehen, dass es zwei Personen mit dem Namen Charles gibt. Ist das korrekt?

Lösung: Es fehlen noch Variablen zu diesem Datenpunkt. Mit «follow your nose» (auf eine Observation klicken) ist ersichtlich, dass es bei diesem Datenpunkt zum Beispiel noch das Geburtsjahr als Eigenschaft gibt.

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

```

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 ?pers ?obs
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
19     #Labels
20     ?name schema:name ?nameLabel .
21
22     #Measure: Personen
23     ?obs measure:BEW ?pers .
24
25     #Neue Variablen: Jahr
26     BIND(YEAR(?time) AS ?year)
27
28     #Jahr auswaehlen
29     FILTER(?year = 2021)
30
31 }
32
33 LIMIT 10

```



3.8 Geburtsjahr

Aufgabe: Erstellen Sie eine zusätzliche Variable für das Geburtsjahr. Und filtern Sie auf einen bestimmten Namen (z.B. Charlotte).

Lösung: [Code](#)

Zeile 21: Variable für Geburtsjahr

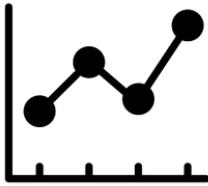
Zeile 33: Filter (Charlotte)

```

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 DISTINCT ?nameLabel ?gebjahr ?pers ?obs
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:GEJ ?geburtsjahr .
19
20     #Geburtsjahr
21     ?geburtsjahr schema:position ?gebjahr .
22
23     #Name: Label
24     ?name schema:name ?nameLabel .
25
26     #Personen
27     ?obs measure:BEW ?pers .
28
29     #Jahr
30     BIND(YEAR(?time) AS ?year)
31
32     #Auswahl
33     FILTER((?year = 2021) && (?nameLabel = 'Charlotte'))
34
35 }
36
37 LIMIT 10

```

3.9 Grafik

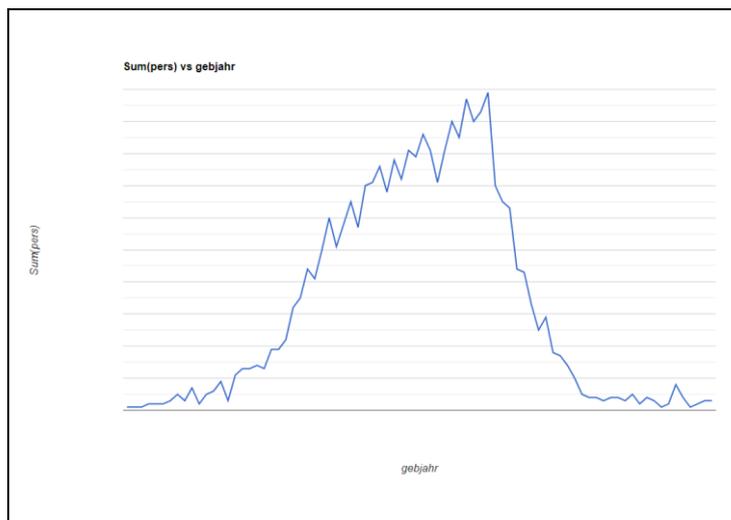


Aufgabe: Filtern Sie auf den Namen «Michael» und sortieren Sie die Ergebnisse absteigend nach Geburtsjahr.

Beim SPARQL-Editor der Stadt Zürich gibt es gewisse Grafik-Tools (limitierte Möglichkeiten). Erstellen Sie bei «Pivot Table» eine Grafik mit Geburtsjahr (x-Achse) gegen Anzahl Personen (y-Achse).

Lösung: [Code](#)

Zeile 37: Nach Geburtsjahr absteigend sortieren



```
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 DISTINCT ?nameLabel ?gebjahr ?pers ?obs
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:GEJ ?geburtsjahr .
19
20     #Geburtsjahr
21     ?geburtsjahr schema:position ?gebjahr .
22
23     #Name: Label
24     ?name schema:name ?nameLabel .
25
26     #Personen
27     ?obs measure:BEW ?pers .
28
29     #Jahr
30     BIND(YEAR(?time) AS ?year)
31
32     #Auswahl
33     FILTER((?year = 2021) && (?nameLabel = 'Michael'))
34
35 }
36
37 ORDER BY DESC(?gebjahr)
```

3.10 Zusätzliches

Eigene Ideen

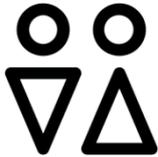


Haben Sie eigene Ideen? Versuchen Sie, weitere Auswertungen durchzuführen.

Bei Fragen können Sie sich gerne an die Tutor*innen wenden.

Wenn Sie momentan nicht an eigenen Auswertungen arbeiten möchten, haben wir im Folgenden weitere Aufgaben für Sie vorbereitet.

Geschlecht



Gewisse Vornamen tragen sowohl weibliche wie männliche Menschen (zum Beispiel «Andrea»). Die vorliegenden Daten kommen aus dem Bevölkerungsregister der Stadt Zürich; dort ist die Variable Geschlecht nur binär enthalten (also keine Kategorie «Divers»).

Aufgabe: Definieren Sie eine Variable für Geschlecht. Und erstellen Sie für den Namen «Andrea» die gleiche Grafik wie zuvor (Anzahl Personen pro Geburtsjahr) nach Geschlecht unterteilt.

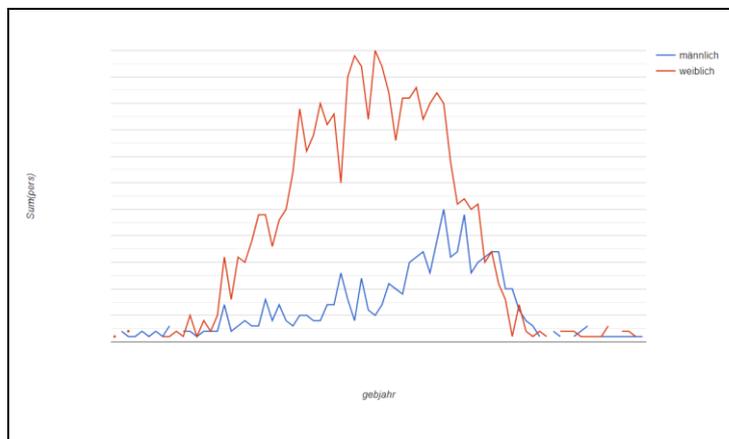
Lösung: [Code](#)

Zeile 18: Variable für Geschlecht

Zeile 26: Label (Text) für Geschlecht

The screenshot shows a data visualization tool interface with the following configuration:

- Chart Type:** Line Chart
- Available Variables:** nameLabel, pers, obs
- Columns:** gebjahr
- Cells:** Sum, pers
- Rows:** gesLabel



```
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 DISTINCT ?nameLabel ?gebjahr ?geslLabel ?pers ?obs
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         property:GEJ ?geburtsjahr .
20
21     #Geburtsjahr
22     ?geburtsjahr schema:position ?gebjahr .
23
24     #Name: Labels
25     ?name schema:name ?nameLabel .
26     ?gesl schema:name ?geslLabel .
27
28     #Personen
29     ?obs measure:BEW ?pers .
30
31     #Jahr
32     BIND(YEAR(?time) AS ?year)
33
34     #Auswahl
35     FILTER((?year = 2021) && (?nameLabel = 'Andrea'))
36
37 }
38
39 ORDER BY DESC(?gebjahr)
```

Ranking



Aufgabe: Welches sind die häufigsten Frauennamen in der Stadt Zürich?

Hinweis: Für diese Rangliste müssen Summen gebildet werden. Tipps zum Aggregieren pro Gruppe finden Sie beispielsweise bei Wikibooks.

https://en.wikibooks.org/wiki/SPARQL/Aggregate_functions

Lösung: [Code](#)

Zeile 35: Gruppe (GROUP BY)

Zeile 7: Summe

```

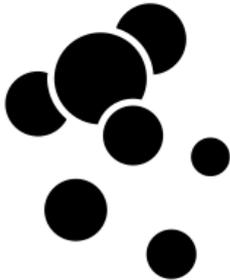
1 PREFIX schema: <http://schema.org/>
2 PREFIX schemac: <https://cube.link/>
3 PREFIX property: <https://ld.stadt-zuerich.ch/statistics/property/>
4 PREFIX measure: <https://ld.stadt-zuerich.ch/statistics/measure/>
5 PREFIX datacubes: <https://ld.stadt-zuerich.ch/statistics/>
6
7 SELECT ?nameLabel (SUM(?pers) 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

```

Linked Data Workshop

	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



Die Daten auf [Wikidata](#) sind Linked Data. Mit dem [Wikidata Query Service](#) kann aber nicht nur auf Wikidata zugegriffen werden; mit sogenannten Federated Queries können auch Linked Data anderer Endpoints verbunden und analysiert werden. Bei Federated Queries zeigt sich das Potenzial von Linked Data: Verschiedene Datenquellen können zusammengeführt und ausgewertet werden. Je mehr Institutionen Linked Data anbieten, desto grösser wird der potenzielle Nutzen.

Aufgabe: Lassen Sie die vorangehende Auswertung (häufigste Frauennamen in der Stadt Zürich) mit dem Wikidata Query Service laufen. Erstellen Sie anschliessend einen Bubble Chart (der Wikidata Query Service bietet mehr Grafik-Möglichkeiten als der SPARQL-Editor der Stadt Zürich).

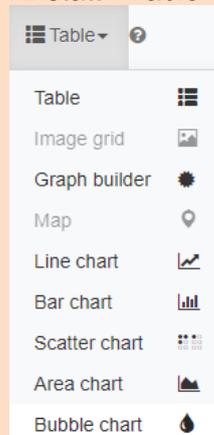
Hinweis: Suchen Sie nach Informationen zu «federated query» oder «SERVICE» ([Beispiel](#)).

Lösung: <https://w.wiki/7EeP>

Es dauert etwa 10 Sekunden, bis der Code durchgelaufen ist.

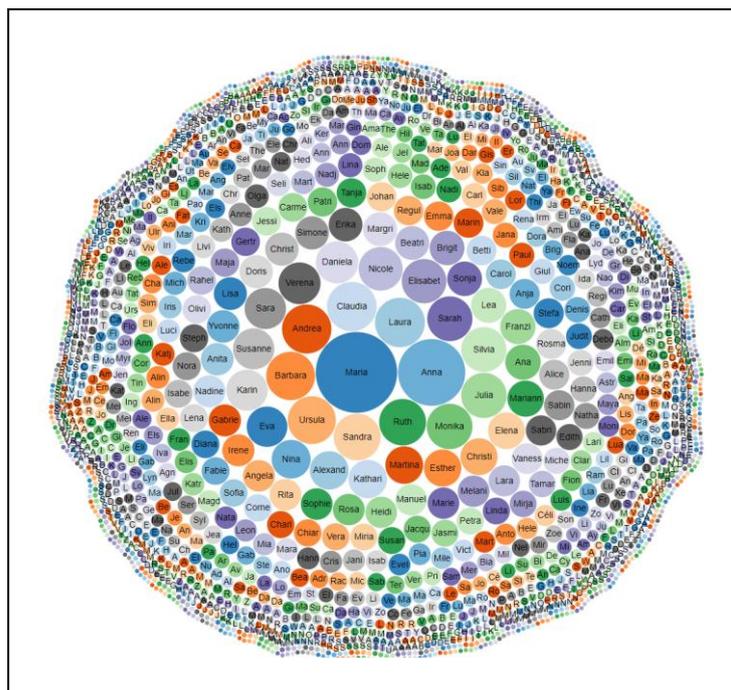
Zeile 11: Federated Query zum SPARQL-Endpoint der Stadt Zürich.

Anstatt «Table»: «Bubble chart» wählen.



Linked Data Workshop

```
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
9 WHERE {
10
11   SERVICE <https://ld.stadt-zuerich.ch/query> {
12
13     #Observation fuer ausgewaehlten Cube
14     datacubes:000437 schemac:observationSet ?obsSet .
15     ?obsSet schemac:observation ?obs .
16
17     #Properties
18     ?obs property:TIME ?time ;
19         property:NAM ?name ;
20         property:SEX ?gesl .
21
22     #Name: Labels
23     ?name schema:name ?nameLabel .
24     ?gesl schema:name ?geslLabel .
25
26     #Personen
27     ?obs measure:BEW ?pers .
28
29     #Jahr
30     BIND(YEAR(?time) AS ?year)
31
32     #Auswahl
33     FILTER((?year = 2021) && (?geslLabel = 'weiblich'))
34
35     #SERVICE Ende
36   }
37
38
39 }
40
41 GROUP BY(?nameLabel)
42 ORDER BY DESC(?total) ?nameLabel
```



Federated Query, HAVING



Aufgabe: Wählen Sie im Wikidata Query Service diejenigen Frauennamen aus, die mindestens 1000 Mal vorkommen, und erstellen Sie damit ein Säulendiagramm.

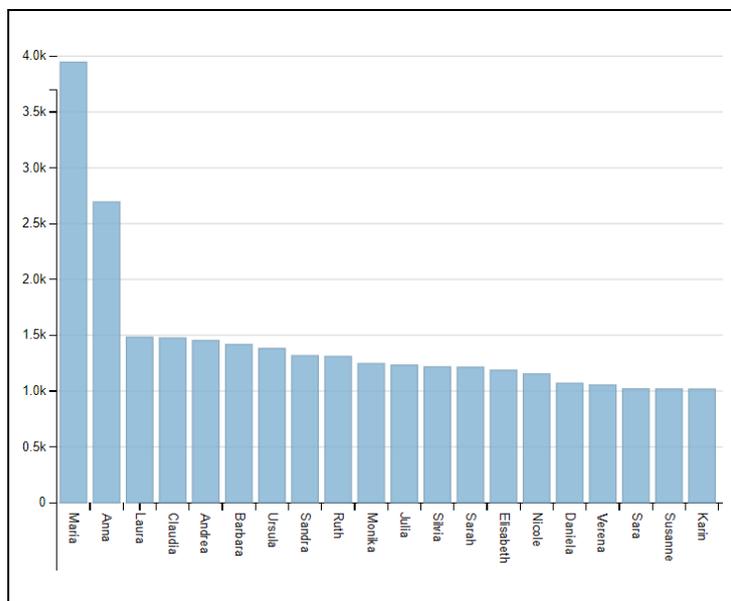
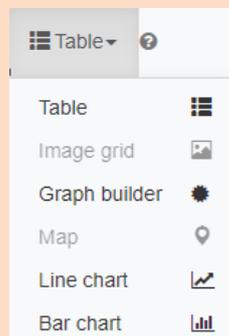
Hinweis: Es geht um das Filtern einer aggregierten Grösse. Dabei ist «HAVING» hilfreich (Beispiel auf wikibooks.org).

Lösung: <https://w.wiki/7EeS>

Im Säulendiagramm wird deutlich, dass Maria und Anna in der Stadt Zürich im Vergleich mit den anderen Frauennamen viel häufiger vorkommen.

Zeile 42: HAVING

Anstatt «Table»: «Bar chart» wählen.



```
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
9 WHERE {
10
11     SERVICE <https://ld.stadt-zuerich.ch/query> {
12
13         #Observation fuer ausgewaehlten Cube
14         datacubes:000437 schemac:observationSet ?obsSet .
15         ?obsSet schemac:observation ?obs .
16
17         #Properties
18         ?obs property:TIME ?time ;
19             property:NAM ?name ;
20             property:SEX ?gesl .
21
22         #Name: Labels
23         ?name schema:name ?nameLabel .
24         ?gesl schema:name ?geslLabel .
25
26         #Personen
27         ?obs measure:BEW ?pers .
28
29         #Jahr
30         BIND(YEAR(?time) AS ?year)
31
32         #Auswahl
33         FILTER((?year = 2021) && (?geslLabel = 'weiblich'))
34
35         #SERVICE Ende
36     }
37
38 }
39
40
41 GROUP BY(?nameLabel)
42 HAVING (?total >= 1000)
43 ORDER BY DESC(?total) ?nameLabel
```

**Federated Query,
Wikidata Porträts**

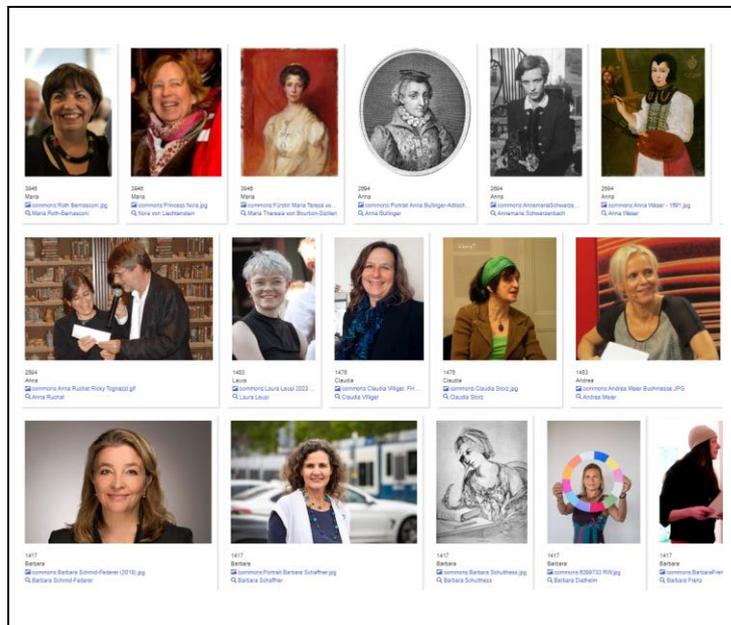
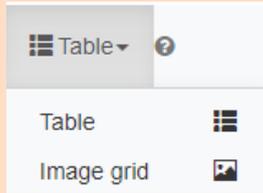


Kennen Sie Zürcherinnen, die häufige Vornamen tragen?

Aufgabe: Wählen Sie die häufigsten zehn Frauen-namen in der Stadt Zürich aus. Verlinken Sie Frauen mit diesen Vornamen, die auf Wikidata sind, Geburtsort Zürich haben und auf Wikidata ein Bild (Porträt) hinterlegt haben. Stellen Sie diese Bilder dar.

Lösung: Leider gibt es zu dieser Wikidata-Abfrage keine Kurz-URL. Kopieren Sie daher den Code auf der nächsten Seite in den [Wikidata Query Service](#).

Anstatt «Table»: «Image grid» wählen.



```
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". }
}

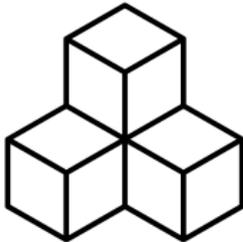
ORDER BY DESC(?total)
```

```

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 PREFIX code: <https://ld.stadt-zuerich.ch/statistics/code/>
7
8 #Wikidata
9
10 SELECT ?nameLabel ?total ?person ?personLabel ?pic
11
12 WHERE {
13
14   ?person wdt:P21 wd:Q6581072 ;
15           wdt:P19 wd:Q72 ;
16           wdt:P18 ?pic ;
17           wdt:P735 ?name .
18   ?name rdfs:label ?nameLabel .
19
20   FILTER((LANG(?nameLabel)) = "de")
21   FILTER((STR(?nameLabel)) = ?nameLabelSSZ)
22
23
24 #Stadt Zuerich
25
26 {
27
28   SELECT ?nameLabelSSZ (SUM(xsd:integer(?pers)) AS ?total)
29
30   WHERE {
31
32     SERVICE <https://ld.stadt-zuerich.ch/query> {
33
34       datacubes:000437 schemac:observationSet ?obsSet .
35       ?obsSet schemac:observation ?obs .
36       ?obs property:TIME ?time ;
37            property:NAM ?name ;
38            property:SEX code:SEX0002 ;
39            measure:BEW ?pers .
40       ?name schema:name ?nameLabelSSZ .
41
42       BIND(YEAR(?time) AS ?year)
43       FILTER(?year = 2021)
44
45     }
46   }
47   GROUP BY(?nameLabelSSZ)
48   ORDER BY DESC(?total)
49   LIMIT 10
50
51 }
52
53 SERVICE wikibase:label { bd:serviceParam wikibase:language "de, en, fr, it". }
54
55 }
56
57 ORDER BY DESC(?total)

```

**Mehrere Cubes,
Geburten und
Bevölkerung**



Wie viele **Neugeborene** tragen einen bestimmten Namen? Wie viele Menschen der **Bevölkerung** heissen entsprechend? Verbinden Sie dazu die beiden Cubes Namen der Geburten und der Wohnbevölkerung (siehe Kapitel 3.1, Cube suchen).

Aufgabe: Welches waren die 20 häufigsten männlichen Vornamen (erster Vorname) bei Neugeborenen im Jahr 2021? Berechnen Sie für diese pro Name:

- Anzahl **Geburten** im Jahr 2021
- Anzahl Personen in der **Bevölkerung** per Ende 2021
- Wieviel **Prozent** machen die Neugeborenen 2021 an der Bevölkerung aus?

Lösung: [Code](#).

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

Beispiel Theo: 20 wurden 2021 geboren. Ende Jahr hatte es in der Bevölkerung 124 Theos. Die Geburten 2021 machten 16 Prozent aus.

Beispiel Alexander: 14 wurden 2021 geboren. Ende Jahr hatte es in der Bevölkerung 1399. Die Geburten 2021 machten 1 Prozent aus.

```

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 PREFIX code: <https://ld.stadt-zuerich.ch/statistics/code/>
7
8 SELECT ?nameLabel ?geburten ?personen
9 FROM <https://lindas.admin.ch/stadtzuerich/stat>
10 WHERE {
11 {
12   SELECT ?name (SUM(?geb) AS ?geburten)
13   WHERE {
14
15     #Geburten (G): Observations
16     datacubes:000361 schemac:observationSet ?obsSetG .
17     ?obsSetG schemac:observation ?obsG .
18
19     #Properties
20     ?obsG property:TIME "2021-12-31"^^xsd:date ;
21     property:SEX code:SEX0001 ;
22     property:NAF code:NAF0001 ;
23     property:NAM ?name .
24
25     #Geburten
26     ?obsG measure:GEB ?geb .
27
28   }
29   GROUP BY ?name
30 }
31 OPTIONAL
32 {
33   SELECT ?name (SUM(?pers) AS ?personen)
34   WHERE {
35
36     #Bevoelkerung (B): Observations
37     datacubes:000437 schemac:observationSet ?obsSetB .
38     ?obsSetB schemac:observation ?obsB .
39
40     #Properties
41     ?obsB property:TIME "2021-12-31"^^xsd:date ;
42     property:SEX code:SEX0001 ;
43     property:NAM ?name .
44
45     #Personen
46     ?obsB measure:BEV ?pers .
47
48   }
49   GROUP BY ?name
50 }
51 }
52
53 #Labels
54 ?name schema:name ?nameLabel .
55
56 }
57
58 ORDER BY DESC(?geburten)

```

4 Anhang

4.1 Wer macht Linked Data?

Linked Open Data

In der Schweiz veröffentlichen verschiedene Institutionen Daten als Linked Open Data; unter anderem folgende:

- Stadt Zürich: www.stadt-zuerich.ch/lod
- Kanton Basel-Stadt ([Link](#))
- Weitere Institutionen in der Schweiz ([Link](#))

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

Abbildung 15: Wer macht Linked Open Data?

4.2 LinkedDataGPT

Seit August 2023 gibt es den [LinkedDataGPT](#) von Liip (siehe [Beschreibung](#)). Dieser verwendet Daten, die von der Stadt Zürich als Linked Open Data veröffentlicht wurden.

4.3 Glossar

Federated Query	Verbundene Abfrage. In einer Abfrage werden Daten verschiedener Datenquellen verbunden (federated: verbunden, vereinigt).
LD	Linked Data
LOD	Linked Open Data (Linked Data mit offener Lizenz)
RDF	Resource Description Framework <ul style="list-style-type: none">– Resource: Alles, was eindeutig mit einem URI identifiziert werden kann.– Description: Beschreibung der Ressourcen: Attribute, Beziehungen– Framework: Gerüst für diese Beschreibungen
Semantik	Bedeutung/Inhalt eines Wortes, Satzes oder Textes
SPARQL	SPARQL Protocol And RDF Query Language
URI	Uniform Resource Identifier. Eindeutige Identifizierung von Realem oder Abstraktem (Webseiten, Personen, Produkte, Standorte, Eigenschaften, Beziehungen usw.)
URL	Uniform Resource Locator (bei Webseiten)

4.4 Bildquellen

Abbildung 1: Beispiele zu unklarer Semantik: Bank, Erde und Jaguar	https://commons.wikimedia.org/wiki/File:The_Co-operative_Bank_-_Ealing_%289415463884%29.jpg , https://commons.wikimedia.org/wiki/File:Sitzbank_in_Gemeinde_T%C3%B6pen_20201003_DSC4822.jpg , https://en.wikipedia.org/wiki/Earth , https://de.wikipedia.org/wiki/Datei:Rendzina_flachgruendig_Heufeld_Schwaebische-Alb.jpg , https://en.wikipedia.org/wiki/Jaguar#/media/File:Standing_jaguar.jpg , https://de.wikipedia.org/wiki/Jaguar_X350
Abbildung 2: Triple	Eigene Darstellung
Abbildung 3: Graph	Eigene Darstellung
Abbildung 4: URI vs. URL	Eigene Darstellung; nach «Linked Data Engineering, 1.6. How to name Things», https://open.hpi.de/courses/semanticweb2016/items/5F6Tk3MwjAkAiNSagEwERD
Abbildung 5: SELECT, WHERE	Eigene Darstellung; nach Wikidata-Tutorial: https://www.wikidata.org/wiki/Wikidata:SPARQL_tutorial/de
Abbildung 6: Punkt am Triple-Ende	Eigene Darstellung; nach Wikidata-Tutorial: https://www.wikidata.org/wiki/Wikidata:SPARQL_tutorial/de
Abbildung 7: Strichpunkt, Komma, Punkt	Eigene Darstellung; nach Wikidata-Tutorial: https://www.wikidata.org/wiki/Wikidata:SPARQL_tutorial/de
Abbildung 8: LIMIT, ORDER BY	Eigene Darstellung; nach Wikidata-Tutorial: https://www.wikidata.org/wiki/Wikidata:SPARQL_tutorial/de

Abbildung 9: BIND, YEAR, FILTER Eigene Darstellung

Abbildung 10: PREFIX Eigene Darstellung

Abbildung 11: Bisherige Daten Eigene Darstellung

Abbildung 12: Datenpunkte beschreiben Eigene Darstellung

Abbildung 13: Datenstrukturen Eigene Darstellung

Abbildung 14: Praxis Eigene Darstellung

Abbildung 15: Wer macht Linked Open Data? Eigene Darstellung

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