SensorThings in QGIS

Inhaltsverzeichnis

Links

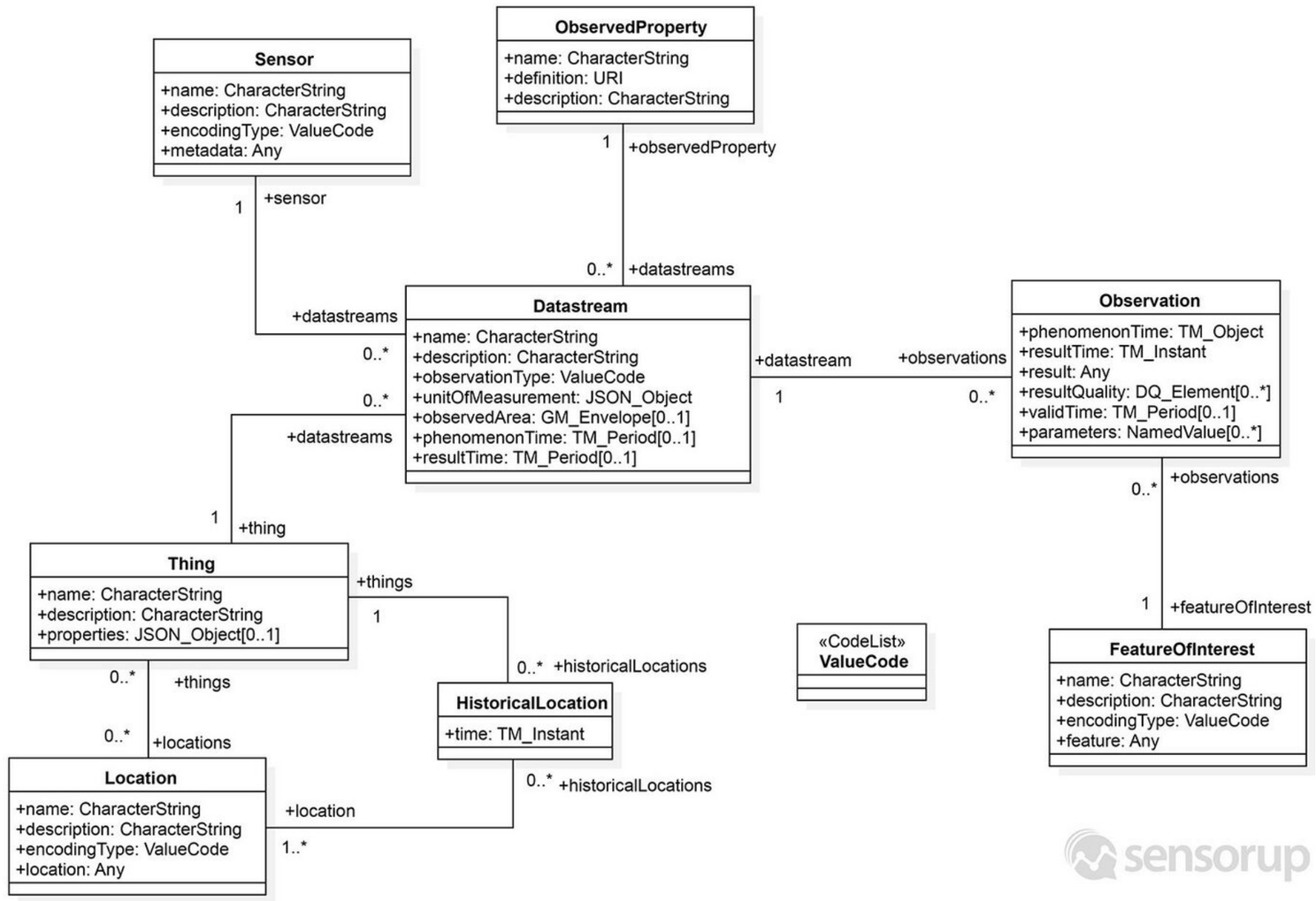
- Dokumentation in QGIS: https://docs.qgis.org/testing/en/docs/user_manual/working_with_ogc/ogc_client_support.html#sensorthings
- Sensor things Data Provider, pt 1 (Request in QGIS) #8866: https://github.com/qgis/QGIS-Documentation/issues/8866#issuecomment-2713647039
- Filtering
  - https://datacoveeu.github.io/API4INSPIRE/sensorthingsapi/requestingData/STA-Filtering.html
  - 9.3.3.5.2 Built-in query functions
    - OGC SensorThings API Part 1: Sensing Version **1.0** -> https://docs.ogc.org/is/15-078r6/15-078r6.html
    - OGC SensorThings API Part 1: Sensing Version **1.1** -> https://docs.ogc.org/is/18-088/18-088.html

verfügbare Endpoints

- Demo mit 50 Punkten: https://ogc-demo.k8s.ilt-dmz.iosb.fraunhofer.de/FROST-SoilThings/v1.1/
- Hamburg : https://iot.hamburg.de/v1.1/
- Listen mit Endpoints
  - https://github.com/opengeospatial/sensorthings/blob/master/PublicEndPoints.md

- https://github.com/AirBreak-UIA/SensorThingsAPI_QGIS-plugin?tab=readme-ov-file#readme

OGC



Offene Fragen

- Unterschied Locations/Orte zu Points of Interest/Objekte von Interessse
  - Hamburg 5.343 zu 10.895.800; Ingolstadt 770 zu 731

Filtern für die Einbindung

- Grundsätzlich gibt es vier Attribut-Typen

Id	Name	Alias	Typ	Typname
abc 0	id		Text (string)	
{} 4	properties		Karte	json
17	Thing_Datastream_phenomenonTimeStart		Datum & Zeit	
[a] 27	Thing_Datastream_Observation_resultQuality		Zeichenkettenliste	

Texte

- substringof('für',description) trifft zu, wenn im Attribut description irgendwo der String 'für' enthalten ist
- endswith(description,'tt') trifft zu, wenn das Attribut description mit 'tt' endet
- startswith(description,'Mittel') trifft zu, wenn dass Attribut description mit dem String 'Mittel' beginnt
- startswith(tolower(description),'mittel') trifft zu, wenn dass Attribut description mit dem String 'mittel' beginnt
- startswith(toupper(description),'MITTEL') trifft zu, wenn dass Attribut description mit dem String 'MITTEL' beginnt
- length(description) gt 13 trifft zu, wenn dass Attribut description mehr als 13 Zeichen hat
- indexof(description,'i') eq 2 trifft zu, wenn im Attribut description das Zeichen Nr. 2 den wert 'i' hat (i<>I)

Karte oder {}

- properties/adresse eq 'Gilbertstraße 63, 22767 Hamburg' filtert alle Objekte, bei den properties/Adresse (Typ:Karte) den Wert 'Gilbertstraße 63, 22767 Hamburg' hat
- substringof('Hamburg',properties/adresse) filtert alle Objekte, in denen bei properties/Adresse (Typ:Karte) den Wert 'Hamburg' enthalten ist

Datum & Zeit

- year(resultTime) lt year(now()) filtert alles aus den Jahren vor dem aktuellen Jahr
- year(resultTime) eq 2025 filtert alles aus dem Jahren 2025
- month(resultTime) eq '6' filtert alles einem Juni
- resultTime gt 2025-06-01T00:00:00Z filtert alle Werte nach dem 01.06.2025 00:00 Uhr heraus
- resultTime gt now() sub duration'P6D' filtert alle Werte der letzen 6 Tage heraus (now - 6 Tage)

Filtern der Ergebnisse

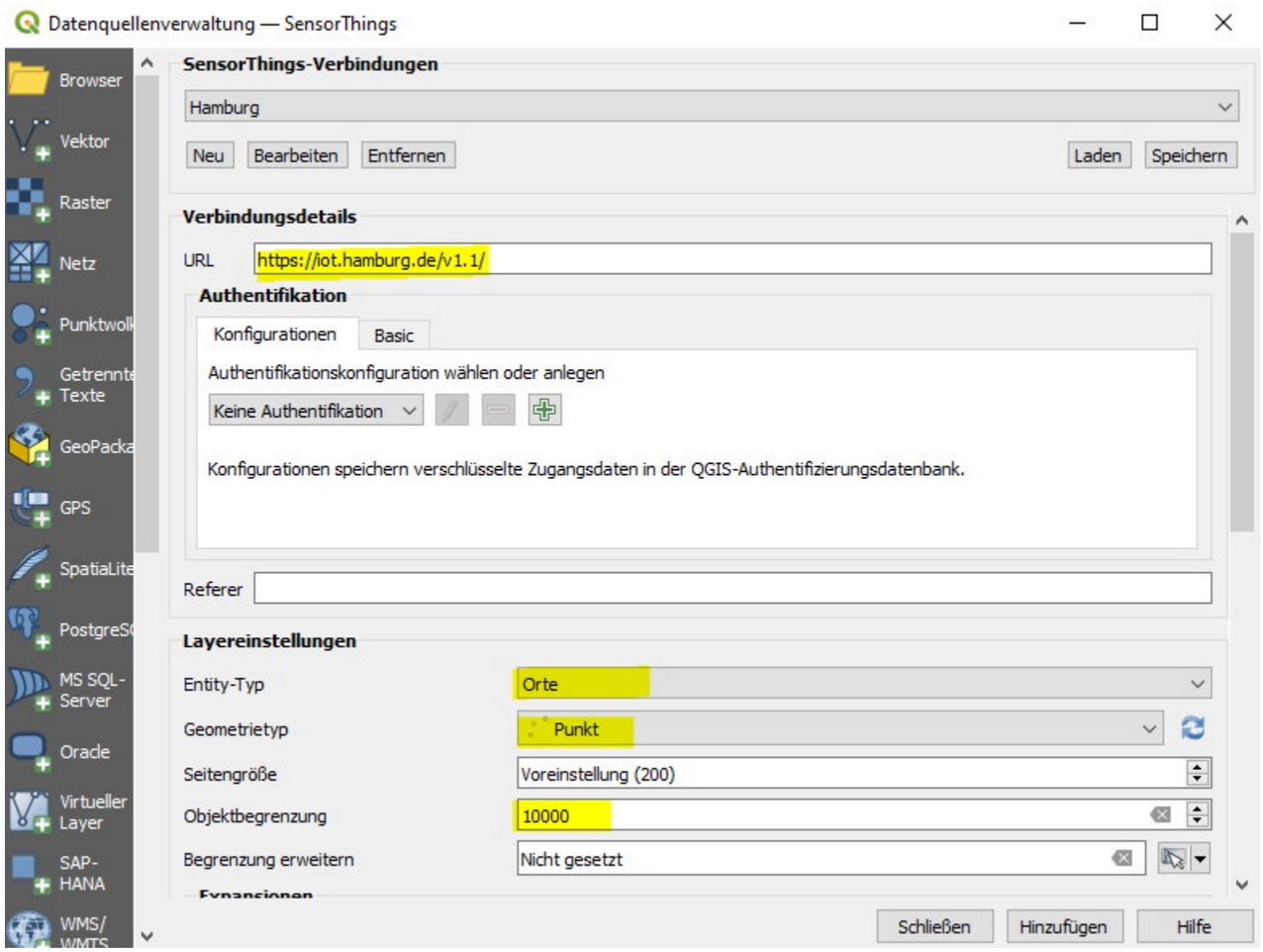
- Karte oder {}
  - map_get("properties", 'adresse') liest aus dem Attribut properties das Feld adresse aus, sofern es vorhanden ist
  - year(to_datetime(map_get("properties", 'infoLastUpdate'))) liest aus dem Attribut properties das Feld infoLastUpdate aus, sofern es vorhanden ist und erzeugt aus dem Zeitwert das Jahr
  - array_to_string(array_foreach(map_akeys("properties"),format('%1: %2', @element, map_get("properties", @element))),', ') erzeugt aus der "Karte oder {}" einen flachen String mit allen Key/Values, egal wie viele oder unterschiedlich
- Zeit
  - umwandeln von UTC in Ortszeit unter Berücksichtigung der Sommerzeit, wobei die UTC-Zeit im Feld "Thing_Datastream_Observation_resultTime" steht, der Output erfolgt als Text

```
case when
"Thing_Datastream_Observation_resultTime" between
format_date(to_datetime(concat(year(now()), '-03-',31 - (day_of_week(concat(year(now()), '-03-31')) %7),'T01:59:59')),'yyyy-MM-dd"T"hh:mm:ss')
and
format_date(to_datetime(concat(year(now()), '-10-',31 - (day_of_week(concat(year(now()), '-10-31')) %7),'T03:00:00')),'yyyy-MM-dd"T"hh:mm:ss')
then
```

```
format_date(to_datetime("Thing_Datastream_Observation_resultTime") + make_interval(0, 0, 0, 0, 2),'dd.MM.yyyy hh:mm:ss' )
else
format_date(to_datetime("Thing_Datastream_Observation_resultTime") + make_interval(0, 0, 0, 0, 1),'dd.MM.yyyy hh:mm:ss' )
end
```

Arbeitsweise

Sichten der Locations / Orte

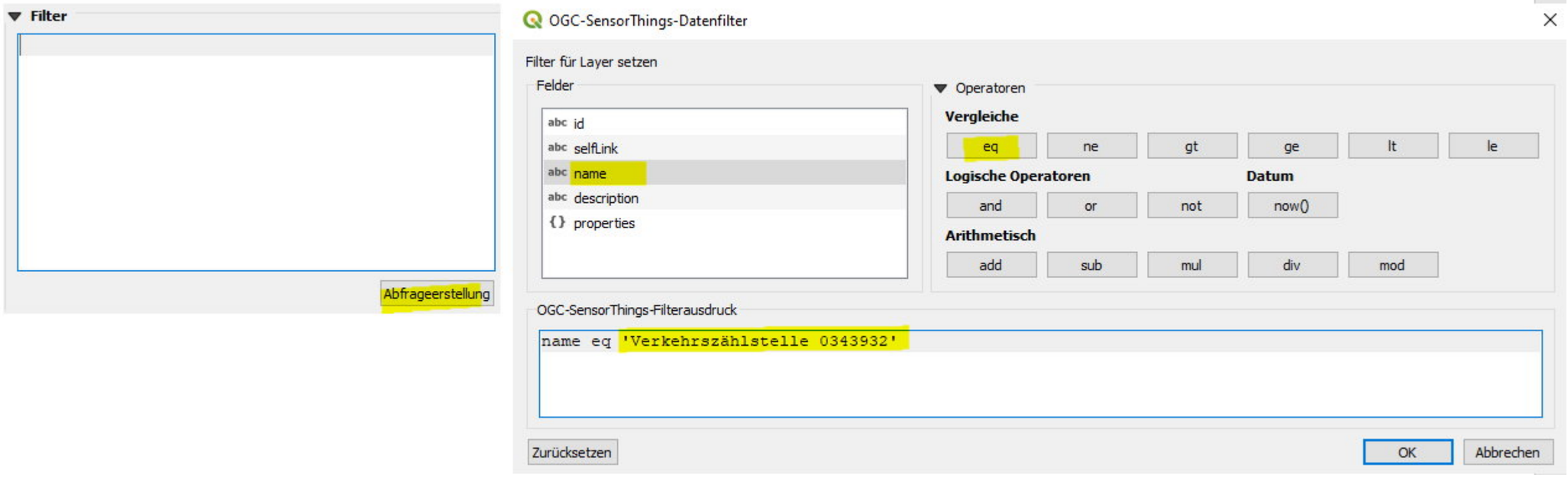


- Beispiel: https://iot.hamburg.de/v1.1/
- hier geht es um die Gesamtmenge der Sensoren mit dem Zweck, um sich einen auszusuchen
- das Ergebnis sind dann n Geometrieobjekte (hier 5342 Punkte)
- ein Beispielobjekt
  - **id**: 26518972
  - **selfLink**: https://iot.hamburg.de/v1.1/Locations(26518972)
  - **name**: Verkehrszählstelle 0343932
  - **description**: Mittelpunkt des Streckenverlaufes bezüglich Zählstelle 0343 auf dem Richtungsarm Nordost von Südwest nach Nordost
  - **properties**: NULL

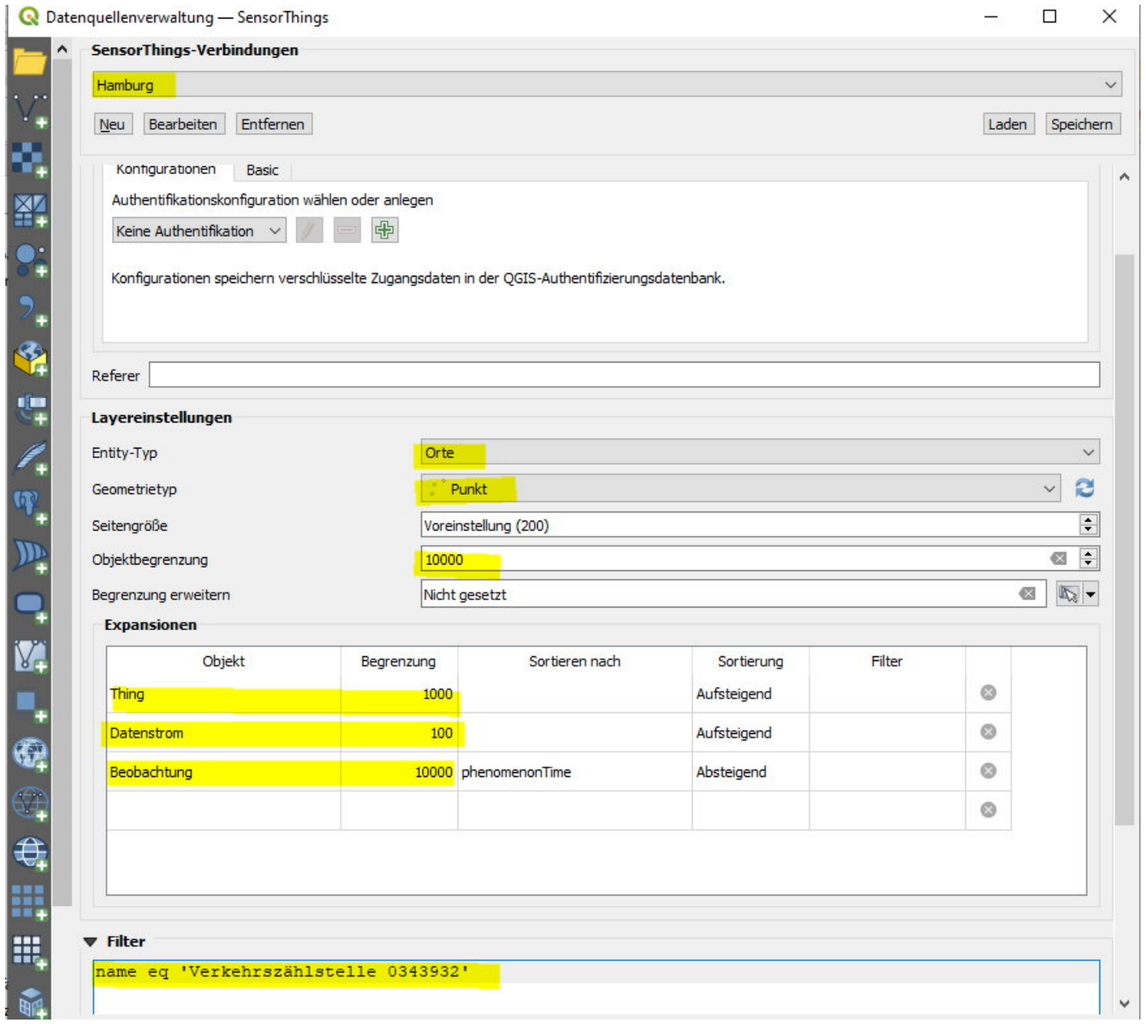
Beschränkung auf ein Objekt

- Zur Beschränkung auf nur ein (oder wenige) Objekte müssen diese im vorherigen Schritt am besten aus der Datentabelle ausgewählt werden. Dazu wird der Name exakt in der Schreibweise verwendet wie angegeben

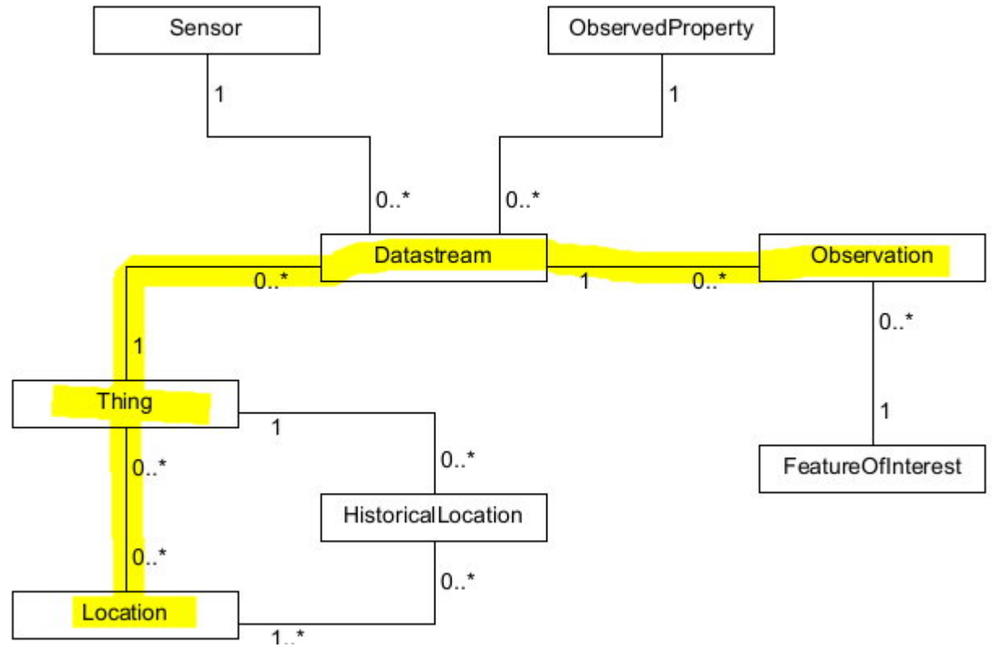
```
name eq 'Verkehrszählstelle 0343932'
```

- 

Verknüpfung von Location/Ort mit Observation /Beobachtungen



- auf Grund der Datenstrukur muss die Verbindung Location-Thing-Datstream-Observation gebildet werden
-

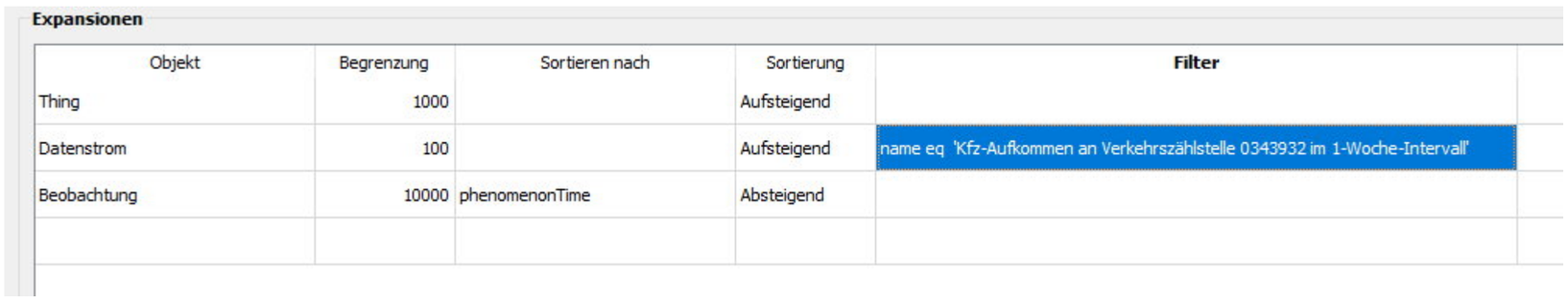


- das Ergebnis sind 3176 Observation /Beobachtungen in vier verschiedenen Datenströmen ("Thing_Datastream_name")
  - 'Kfz-Aufkommen an Verkehrszählstelle 0343932 im 1-Stunde-Intervall'
  - 'Kfz-Aufkommen an Verkehrszählstelle 0343932 im 1-Tag-Intervall'
  - 'Kfz-Aufkommen an Verkehrszählstelle 0343932 im 1-Woche-Intervall'
  - 'Kfz-Aufkommen an Verkehrszählstelle 0343932 im 15-Min-Intervall'

Filtern auf einen Datenstrom

```
name eq  'Kfz-Aufkommen an Verkehrszählstelle 0343932 im 1-Woche-Intervall'
```

-



- beschränkt das Ergebnisa auf 176 Stück

Filtern auf einen Zeitabschnitt

```
year(resultTime) eq 2025
```

- filtert alle Ergebnisse aus dem Jahr 2025 (hier 21 Stück)
-

Objekt	Begrenzung	Sortieren nach	Sortierung	**Filter**
Thing	1000		Aufsteigend	
Datenstrom	100		Aufsteigend	name eq 'Kfz-Aufkommen an Verkehrszählstelle 0343932 im 1-Woche-Intervall'
Beobachtung	10000	phenomenonTime	Absteigend	year(resultTime) eq 2025

- als Eigenschaften des Punktlayers ergibt sich jetzt

```
type=PointZ entity='Location' expandTo='Thing:limit=1000;Datastream:limit=100:filter=name eq \'Kfz-Aufkommen an Verkehrszählstelle 0343932 im 1-Woche-Intervall\' ;Observation:orderby=phenomenonTime,desc:limit=10000:filter=year(resultTime) eq
2025' featureLimit='10000' url='https://iot.hamburg.de/v1.1/' sql=name eq 'Verkehrszählstelle 0343932'
```

Ausgabe der Ergebnisse

Data Plotly

- Data Plotly ist das von QGIS (https://docs.qgis.org/testing/en/docs/user_manual/working_with_ogc/ogc_client_support.html#sensorthings) empfohlene Plug-in für diese Aufgabe

---

Diese Seite wurde zuletzt am 10. Juni 2025 um 11:11 Uhr bearbeitet.