# Semantically Enhancing OLAP Cubes:

## Integrating SPARQL and SQL for Next-Generation Data Publication & Business Intelligence

**Benjamin Cogrel and Adrian Gschwend**

Knowledge Graph Forum, Basel, Switzerland, May 30th 2024

zazuko

ONTOPIC

# CSV, XLS, …; let's call it table

| | A | B | C | D |
|---|---|---|---|---|
| 1 | id | cityId | temp | timestamp |
| 2 | 1 | Q425415 | -17 | 2009-05-19T06:32:08 |
| 3 | 2 | Q671288 | 19.9 | 2003-11-26T11:33:18 |
| 4 | 3 | Q956602 | -26.9 | 2011-08-29T20:02:38 |
| 5 | 4 | Q131638 | 44.8 | 2001-11-05T22:01:03 |
| 6 | 5 | Q13025347 | 12 | 2011-12-03T08:09:07 |
| 7 | 6 | Q10976 | 42.9 | 2006-03-10T05:17:37 |

| | A | B | C |
|---|---|---|---|
| 1 | city | cityLabel | cityId |
| 2 | http://www.wikidata.org/entity/Q44214 | Formosa | Q44214 |
| 3 | http://www.wikidata.org/entity/Q41252 | Bydgoszcz | Q41252 |
| 4 | http://www.wikidata.org/entity/Q43433 | Chandigarh | Q43433 |
| 5 | http://www.wikidata.org/entity/Q38811 | Nasiriyah | Q38811 |
| 6 | http://www.wikidata.org/entity/Q42763 | Santiago de los Caballeros | Q42763 |
| 7 | http://www.wikidata.org/entity/Q43509 | Guayaquil | Q43509 |
| 8 | http://www.wikidata.org/entity/Q39984 | Cannes | Q39984 |
| 9 | http://www.wikidata.org/entity/Q40921 | Honiara | Q40921 |
| 10 | http://www.wikidata.org/entity/Q44162 | San Fernando del Valle de Catamarca | Q44162 |
| 11 | http://www.wikidata.org/entity/Q44237 | Mendoza | Q44237 |
| 12 | http://www.wikidata.org/entity/Q44239 | Neuquén | Q44239 |

# Tables

| | A | B | C | D |
|---|---|---|---|---|
| 1 | id | cityId | temp | timestamp |
| 2 | 1 | Q425415 | -17 | 2009-05-19T06:32:08 |
| 3 | 2 | Q671288 | 19.9 | 2003-11-26T11:33:18 |
| 4 | 3 | Q956602 | -26.9 | 2011-08-29T20:02:38 |
| 5 | 4 | Q131638 | 44.8 | 2001-11-05T22:01:03 |
| 6 | 5 | Q13025347 | 12 | 2011-12-03T08:09:07 |
| 7 | 6 | Q10976 | 42.9 | 2006-03-10T05:17:37 |

# Semantics of Tables?

Columns

Rows

| | A | B | C | D |
|---|---|---|---|---|
| 1 | id | cityId | temp | timestamp |
| 2 | 1 | Q425415 | -17 | 2009-05-19T06:32:08 |
| 3 | 2 | Q671288 | 19.9 | 2003-11-26T11:33:18 |
| 4 | 3 | Q956602 | -26.9 | 2011-08-29T20:02:38 |
| 5 | 4 | Q131638 | 44.8 | 2001-11-05T22:01:03 |
| 6 | 5 | Q13025347 | 12 | 2011-12-03T08:09:07 |
| 7 | 6 | Q10976 | 42.9 | 2006-03-10T05:17:37 |

Cell

# Implicit Semantics

Implicit semantics refer to the unspoken or inferred meaning that **you** derive from context.

ID of a Row?

A reference to a City

Probably Temperature

A row is a temperature measurement at a specific time and place.

| | A | B | C | D |
|---|---|---|---|---|
| 1 | id | cityId | temp | timestamp |
| 2 | 1 | Q425415 | -17 | 2009-05-19T06:32:08 |
| 3 | 2 | Q671288 | 19.9 | 2003-11-26T11:33:18 |
| 4 | 3 | Q956602 | -26.9 | 2011-08-29T20:02:38 |
| 5 | 4 | Q131638 | 44.8 | 2001-11-05T22:01:03 |
| 6 | 5 | Q13025347 | 12 | 2011-12-03T08:09:07 |
| 7 | 6 | Q10976 | 42.9 | 2006-03-10T05:17:37 |

Header Row

Date Time

Probably °C

# The City Table

Implicit semantics refer to the unspoken or inferred meaning that **you** derive from context.

An URL called City ?

The name of the City

The ID of a City Probably the same ID as in the previous Table

A row represents a city with its name and IDs

| | A | B | C |
|---|---|---|---|
| 1 | city | cityLabel | cityId |
| 2 | http://www.wikidata.org/entity/Q44214 | Formosa | Q44214 |
| 3 | http://www.wikidata.org/entity/Q41252 | Bydgoszcz | Q41252 |
| 4 | http://www.wikidata.org/entity/Q43433 | Chandigarh | Q43433 |
| 5 | http://www.wikidata.org/entity/Q38811 | Nasiriyah | Q38811 |
| 6 | http://www.wikidata.org/entity/Q42763 | Santiago de los Caballeros | Q42763 |
| 7 | http://www.wikidata.org/entity/Q43509 | Guayaquil | Q43509 |
| 8 | http://www.wikidata.org/entity/Q39984 | Cannes | Q39984 |
| 9 | http://www.wikidata.org/entity/Q40921 | Honiara | Q40921 |
| 10 | http://www.wikidata.org/entity/Q44162 | San Fernando del Valle de Catamarca | Q44162 |
| 11 | http://www.wikidata.org/entity/Q44237 | Mendoza | Q44237 |
| 12 | http://www.wikidata.org/entity/Q44239 | Neuquén | Q44239 |

# Stop guessing semantics; make them explicit

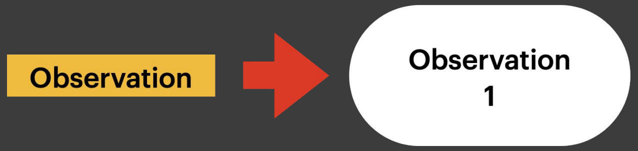
Dimensions

Observation

| | A | B | C | D |
|---|---|---|---|---|
| 1 | id | cityId | temp | timestamp |
| 2 | 1 | Q425415 | -17 | 2009-05-19T06:32:08 |
| 3 | 2 | Q671288 | 19.9 | 2003-11-26T11:33:18 |
| 4 | 3 | Q956602 | -26.9 | 2011-08-29T20:02:38 |
| 5 | 4 | Q131638 | 44.8 | 2001-11-05T22:01:03 |
| 6 | 5 | Q13025347 | 12 | 2011-12-03T08:09:07 |
| 7 | 6 | Q10976 | 42.9 | 2006-03-10T05:17:37 |

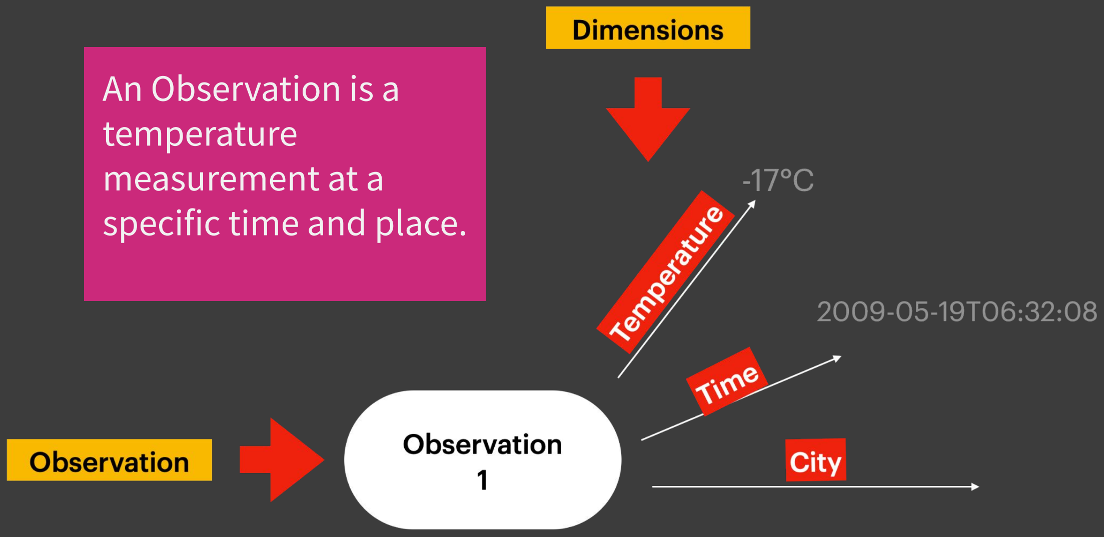
# Stop guessing semantics; make them explicit

An Observation is a temperature measurement at a specific time and place.



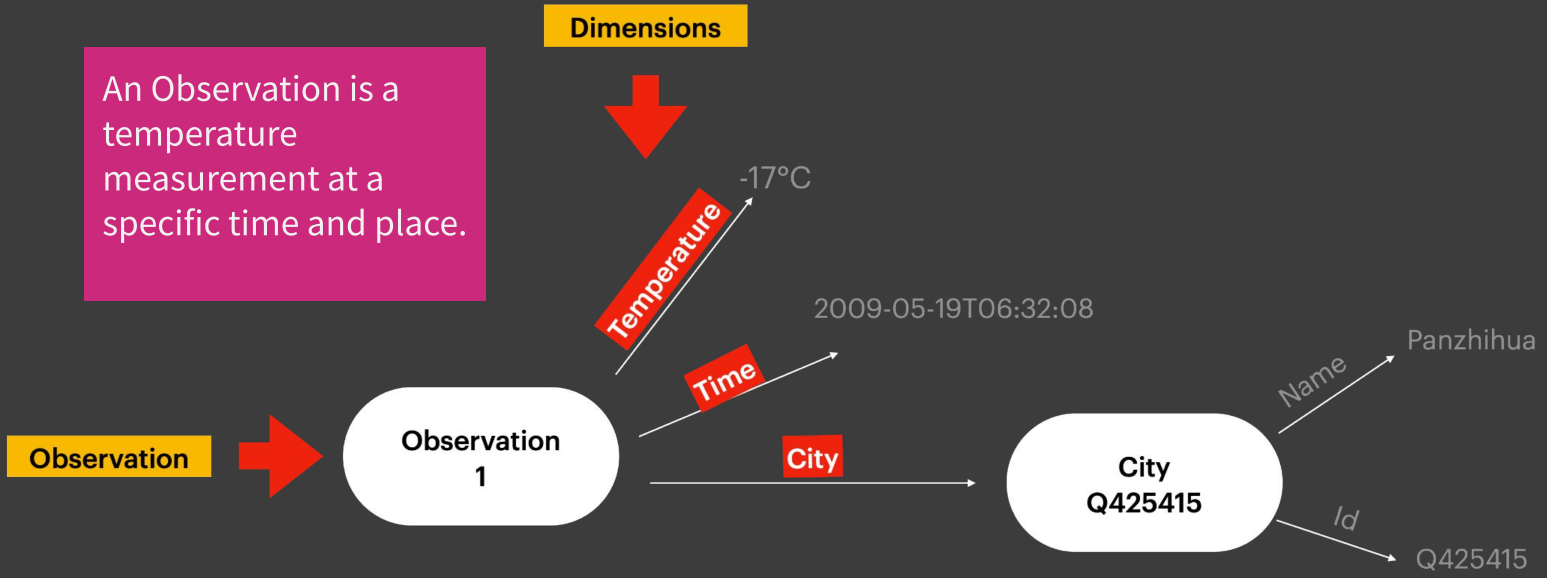
# Stop guessing semantics; make them explicit

An Observation is a temperature measurement at a specific time and place.



# Stop guessing semantics; make them explicit

An Observation is a temperature measurement at a specific time and place.



# Stop guessing semantics; make them explicit

The city where it was measured

Temperature in °C between -65 and 55.

Time when it was matured between 2009 and 2020

Dimension Metadata
- Name
- Description
- Unit
- Levels (Scale)
- Datatype
- …

Cube Metadata
- Name
- Date
- Source
- …

| | A | B | C | D |
|---|---|---|---|---|
| 1 | id | cityId | temp | timestamp |
| 2 | 1 | Q425415 | -17 | 2009-05-19T06:32:08 |
| 3 | 2 | Q671288 | 19.9 | 2003-11-26T11:33:18 |
| 4 | 3 | Q956602 | -26.9 | 2011-08-29T20:02:38 |
| 5 | 4 | Q131638 | 44.8 | 2001-11-05T22:01:03 |
| 6 | 5 | Q13025347 | 12 | 2011-12-03T08:09:07 |
| 7 | 6 | Q10976 | 42.9 | 2006-03-10T05:17:37 |

Observations

# All this is a (OLAP) Cube

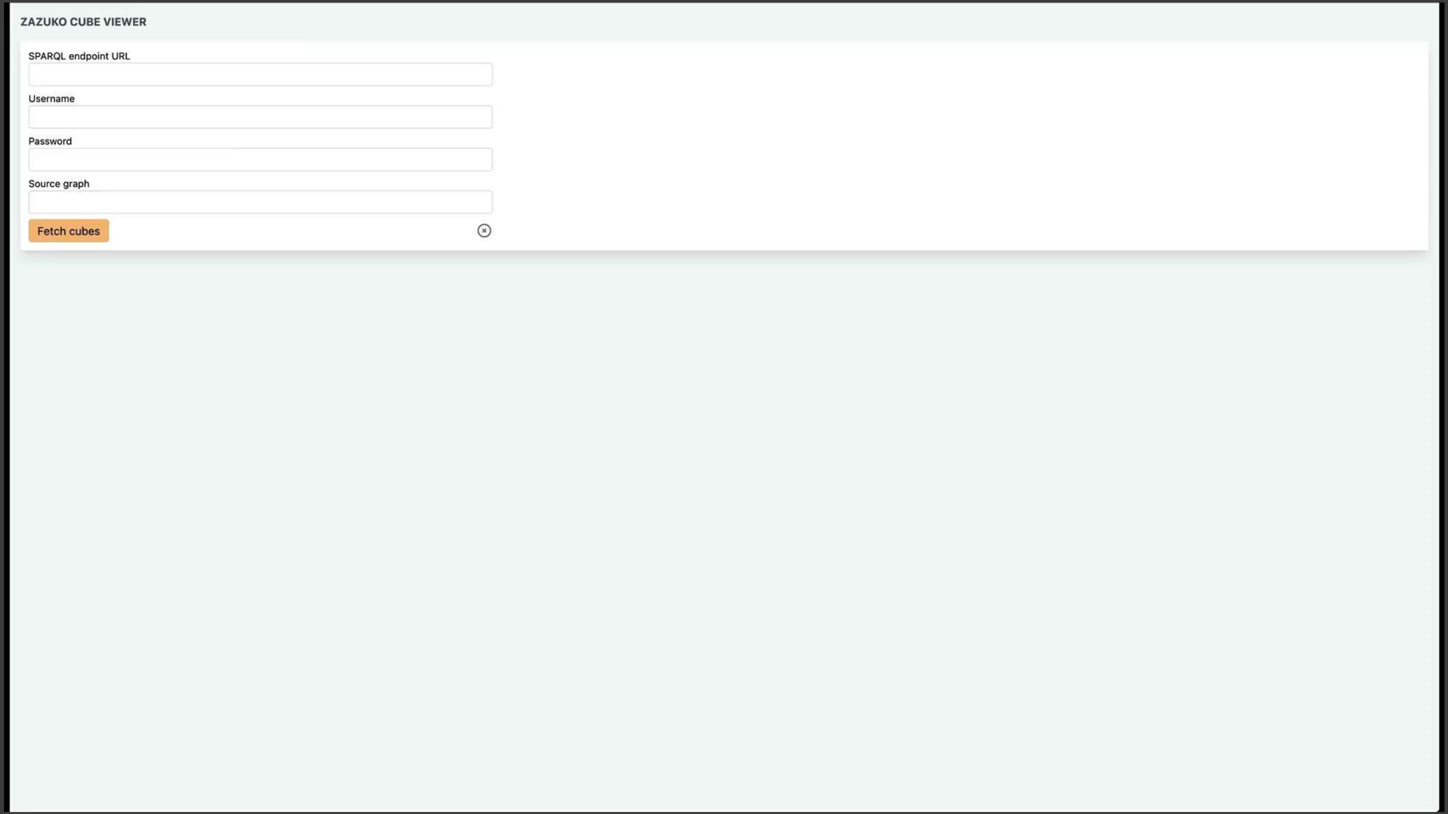
Online Analytical Processing with a multi-dimensional array (hypercube)

# RDF Cube Vocabularies

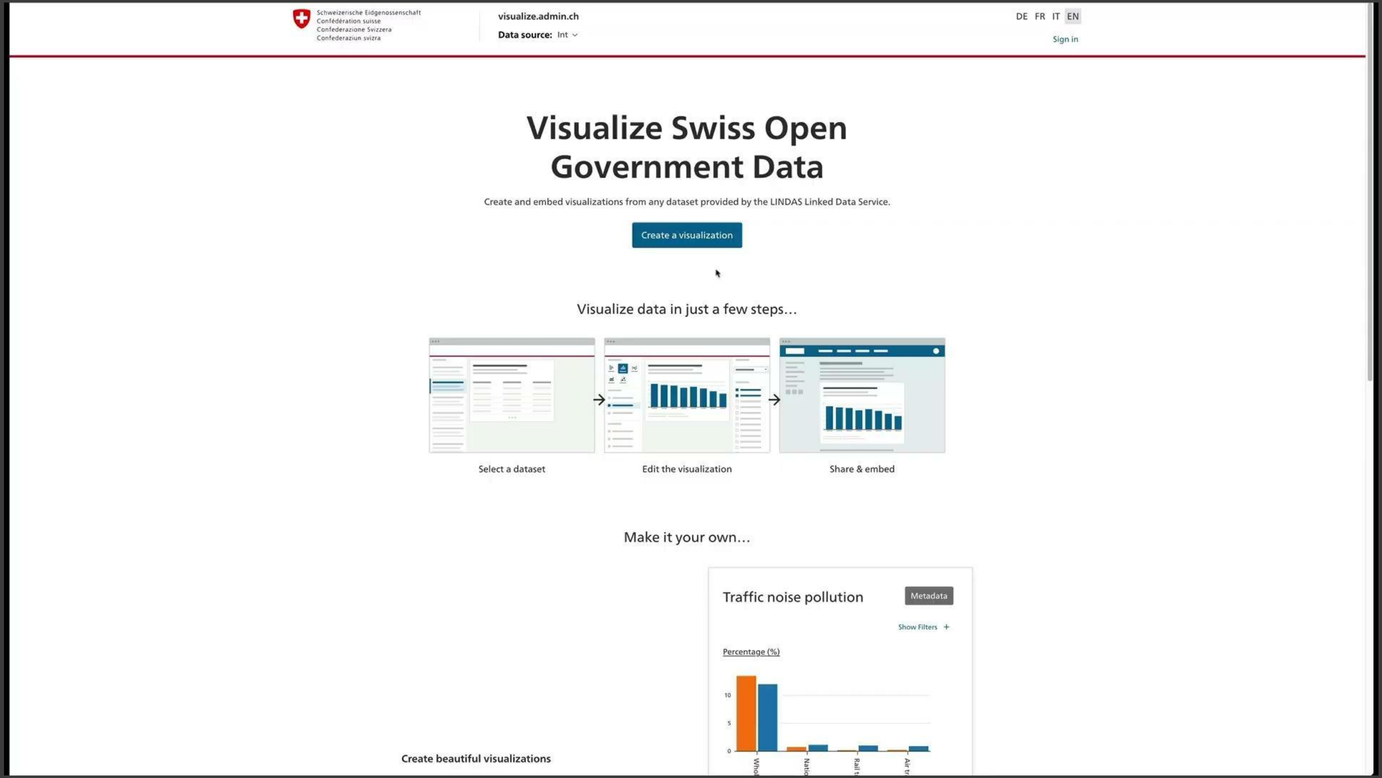
- Cube.link
  - Swiss government/Zazuko
  - used in all the demos shown before
- schema.org
  - schema:Observation
- Semantic Sensor Network Ontology
  - sosa:Observation
- RDF Data Cube Vocabulary
  - The original

“Observation” is the common theme, details differ

# But Why?

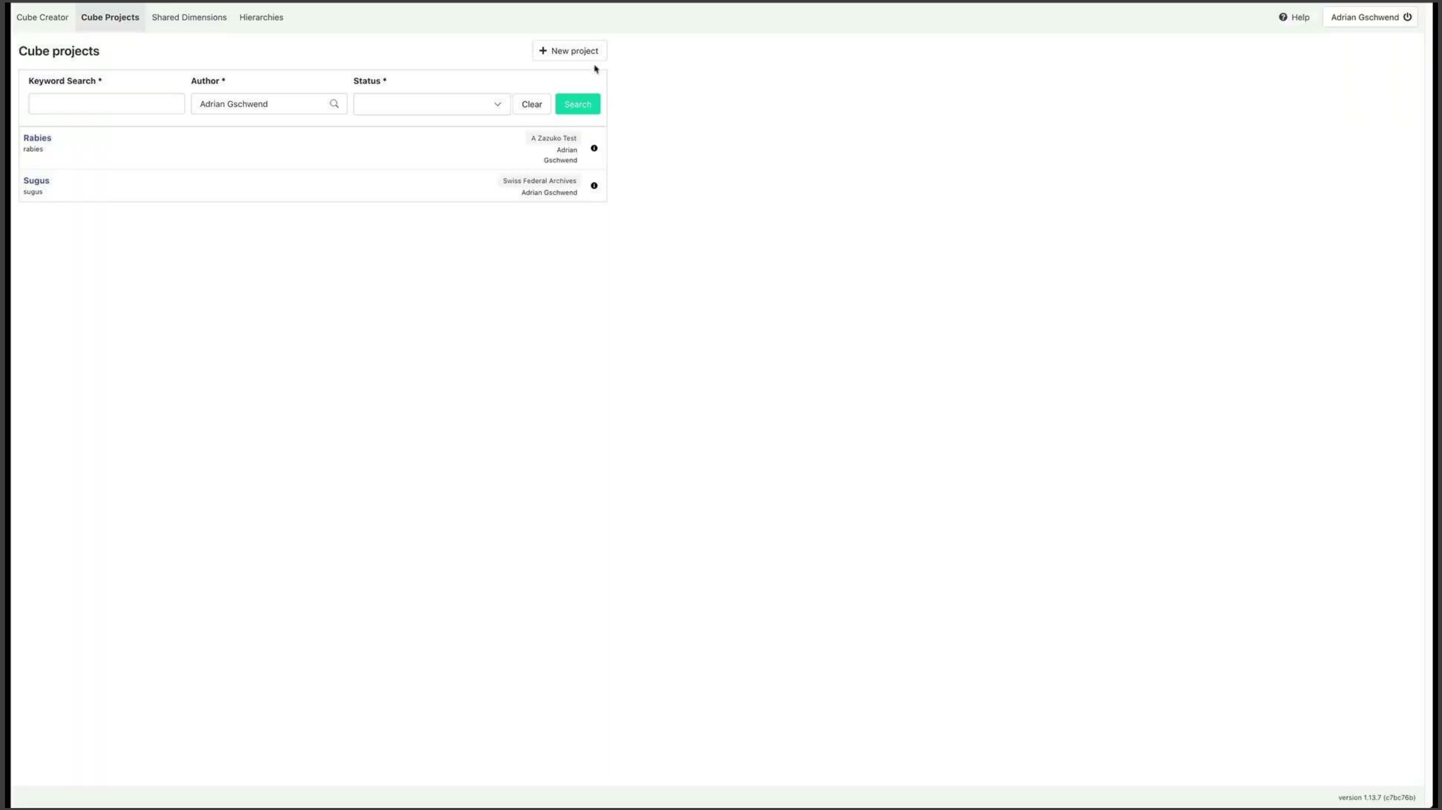

ZAZUKO CUBE VIEWER
SPARQL endpoint URL
Username
Password
Source graph
Fetch cubes

# But Why?



Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
visualize.admin.ch
Data source: Int
DE FR IT EN
Sign in
Visualize Swiss Open Government Data
Create and embed visualizations from any dataset provided by the LINDAS Linked Data Service.
Create a visualization
Visualize data in just a few steps...
Select a dataset
Edit the visualization
Share & embed
Make it your own...
Traffic noise pollution
Metadata
Show Filters +
Percentage (%)
10
5
0
Create beautiful visualizations

# How?



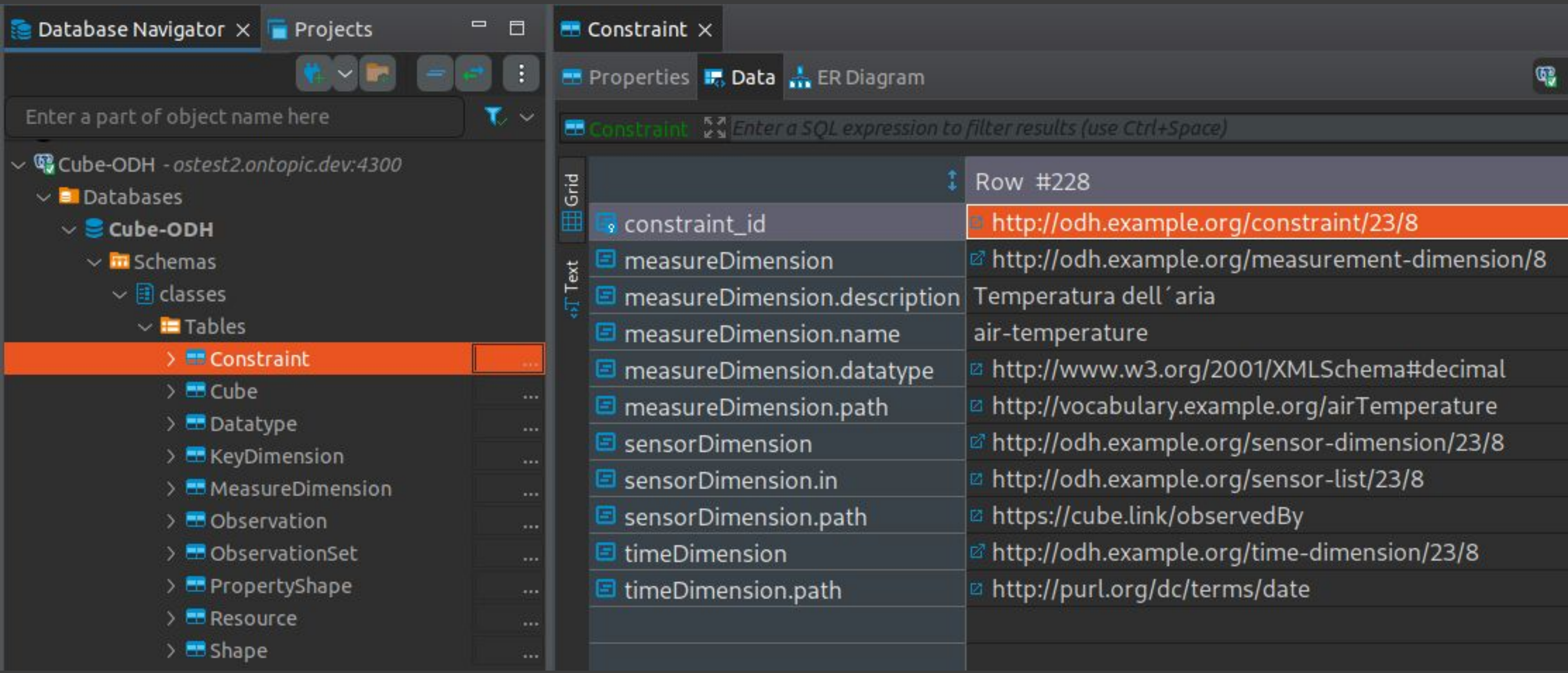
Cube Creator
Cube Projects
Shared Dimensions
Hierarchies
Help
Adrian Gschwend
Cube projects
New project
Keyword Search *
Author *
Status *
Adrian Gschwend
Clear
Search
Rabies
rabies
A Zazuko Test
Adrian Gschwend
Sugus
sugus
Swiss Federal Archives
Adrian Gschwend
version 1.13.7 (c7bc76b)

# RDF cubes with virtualization

- If your data in CSV/JSON/Parquet files or in a relational database
  - You can map it to RDF using an R2RML mapping
  - Systems like Dremio, Trino and DuckDB allow you to query files in SQL
- R2RML mappings enable 2 kinds of deployments
  - Materialization: data is transformed in RDF and loaded in a triplestore
  - Virtualization: user queries are translated into SQL and send to the data source
- 2 languages to query the KG
  - SPARQL
    - Ontop if virtual, triplestore otherwise
  - SQL (new!)
    - For Virtual KGs only (as of now)

# Querying KGs in SQL

- All the default graph data is automatically available through virtual relational tables
    - Don’t have to create custom SQL views (using a SPARQL query)
    - No extra-modelling effort, on a par with the SPARQL interface
    - One table per class, one per object property, one per data property
- Virtual tables contain as many columns as possible
    - They are denormalized so as to save users from writing joins
    - While still guaranteeing that there is no more than 1 row per entity
        - Safe to run aggregates over
    - Made possible by analyzing the structures of the mapping and of the source
    - Backed by many optimizations to run fast
- With foreign keys to pursue the linked data experience
    - For the 1-1 and n-1 relationships


Database Navigator
Projects
Enter a part of object name here
Cube-ODH - ostest2.ontopic.dev:4300
Databases
Cube-ODH
Schemas
classes
Tables
Constraint
Cube
Datatype
KeyDimension
MeasureDimension
Observation
ObservationSet
PropertyShape
Resource
Shape
Constraint
Properties
Data
ER Diagram
Constraint
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text
Row #228
constraint_id
http://odh.example.org/constraint/23/8
measureDimension
http://odh.example.org/measurement-dimension/8
measureDimension.description
Temperatura dell´aria
measureDimension.name
air-temperature
measureDimension.datatype
http://www.w3.org/2001/XMLSchema#decimal
measureDimension.path
http://vocabulary.example.org/airTemperature
sensorDimension
http://odh.example.org/sensor-dimension/23/8
sensorDimension.in
http://odh.example.org/sensor-list/23/8
sensorDimension.path
https://cube.link/observedBy
timeDimension
http://odh.example.org/time-dimension/23/8
timeDimension.path
http://purl.org/dc/terms/date

# Accessing the KG in BI tools

- The SQL connector emulates a PostgreSQL database
  - It works with its standard drivers (JDBC, ODBC, etc.)
- Works with popular tools
  - Tableau, PowerBI, Metabase (open-source)
  - Excel, Pandas, Veezoo

Let's demo it with Metabase (great at data exploration)

Search...

New

# Browse data

Learn about our data

DATABASES > CUBE-ODH

classes

data_properties

object_properties

# Performance and scalability

With TimescaleDB

- Demo made on a cheap 8 GB server with 4 cores (Hetzner CPX31)
- With 480M observations in the same hypertable
  - With default partitioning (7 days)
  - And indexing for quickly retrieving the values of a given time series
- Having the same data in a standard Postgres table on a large RDS instance was painful (both at ingestion and querying times)

# Scaling with DuckDB

DuckDB

- “ DuckDB is a fast in-process analytical database”
- “SQLite” for large tables/time series
- No daemon, JDBC to file

Process described in this post:

https://www.linkedin.com/pulse/scaling-sparql-querying-billion-observations-ontop-duckdb-gschwend-myghf/

# Scaling with DuckDB: Setup

Synthetic CSV file

- 1 billion rows in observation table, ~41GB
- ~ 7’000 cities from Wikidata in city table (join)

Ontop 5.2.0 beta 2

- **No** keys in SQL schema for DuckDB (performance)
- R2RML file
- Ontop Lenses instead of keys
- MacBook M3 Pro (2023), 36 GB memory

# Scaling with DuckDB: Results

Synthetic CSV file

- DuckDB load from CSV: A few minutes
- Written into an index file on disk

SPARQL queries

- Count
- AVG temp by city for a particular year

**Ontop SPARQL endpoint**

endpoint address: http://localhost:8080/sparql | ontop v5.2.0-SNAPSHOT

Observations x | S P O x | AVG x | Filter City x | Query x | Count x | +

```sparql
PREFIX cube: <https://cube.link/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX dimension: <https://example.org/1b/dimension/>

SELECT ?cityName (AVG(?temp) as ?avgTemp) WHERE {
  ?obs a cube:Observation ;
       dimension:temperature ?temp ;
       dimension:time ?time ;
       dimension:city ?city .

  ?city schema:name ?cityName .

  FILTER(year(?time) = 2014)

}
GROUP BY ?cityName
ORDER BY ?cityName
```

Table | Response | Pivot Table | Google Chart | Geo | </>

## Ontop SPARQL endpoint

endpoint address: http://localhost:8080/sparql | ontop v5.2.0-SNAPSHOT

Observations x | S P O x | AVG x | Filter City x | Query x | Query 1 x | +

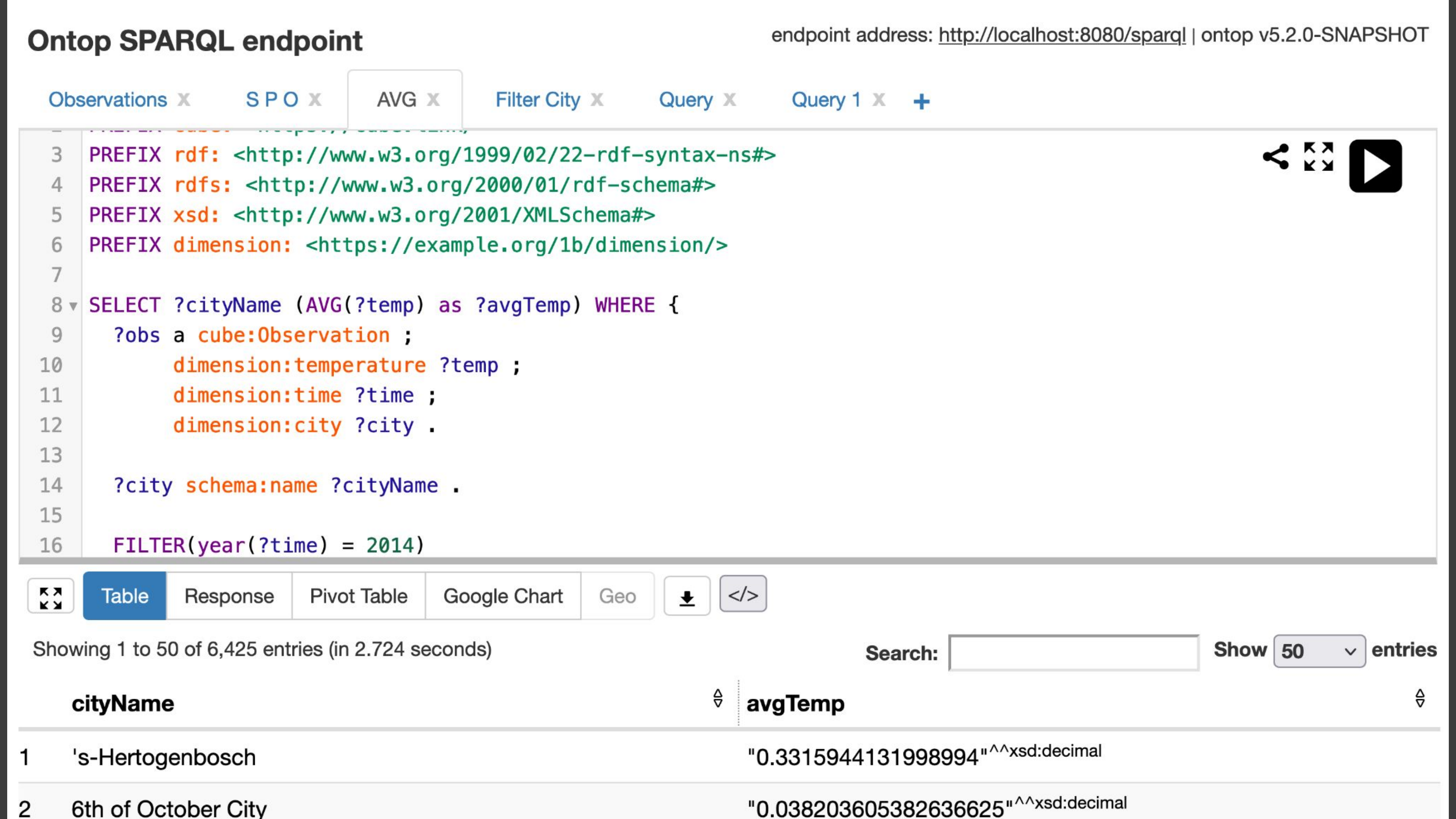
```
PREFIX schema: <http://schema.org/>
PREFIX cube: <https://cube.link/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX dimension: <https://example.org/1b/dimension/>

SELECT ?cityName (AVG(?temp) as ?avgTemp) WHERE {
  ?obs a cube:Observation ;
       dimension:temperature ?temp ;
       dimension:time ?time ;
       dimension:city ?city .

  ?city schema:name ?cityName .
```

Table | Response | Pivot Table | Google Chart | Geo | </>

Showing 1 to 50 of 6,425 entries (in 7.054 seconds)

Search: Show 50 entries

| | cityName | avgTemp |
|---|---|---|
| 1 | 's-Hertogenbosch | "0.06506800886003021"^^xsd:decim |
| 2 | 6th of October City | "0.328946688710063"^^xsd:decima |


Ontop SPARQL endpoint
endpoint address: http://localhost:8080/sparql | ontop v5.2.0-SNAPSHOT
Observations x
S P O x
AVG x
Filter City x
Query x
Query 1 x
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX dimension: <https://example.org/1b/dimension/>
SELECT ?cityName (AVG(?temp) as ?avgTemp) WHERE {
?obs a cube:Observation ;
dimension:temperature ?temp ;
dimension:time ?time ;
dimension:city ?city .
?city schema:name ?cityName .
FILTER(year(?time) = 2014)
Table
Response
Pivot Table
Google Chart
Geo
</>
Showing 1 to 50 of 6,425 entries (in 2.724 seconds)
Search:
Show 50 entries
cityName
avgTemp
1 's-Hertogenbosch "0.3315944131998994"^^xsd:decimal
2 6th of October City "0.038203605382636625"^^xsd:decimal

## Ontop SPARQL endpoint

endpoint address: http://localhost:8080/sparql | ontop v5.2.0-SNAPSHOT

Observations X | S P O X | AVG X | Filter City X | Query X | Count X | +

```
PREFIX cube: <https://cube.link/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT (COUNT(?sub) AS ?count) WHERE {
  ?sub a cube:Observation .
}
LIMIT 10
```

Table | Response | Pivot Table | Google Chart | Geo | </>

Showing 1 to 1 of 1 entries (in 0.191 seconds)

**Search:** **Show** 50 **entries**

| | count |
|---|---|
| 1 | "1000000000"^^xsd:integer |

Showing 1 to 1 of 1 entries (in 0.191 seconds)

# Scaling with DuckDB: Materialization

Same hardware, MacBook M3 Pro

- Ontop materialized the whole DB in 5h 15m
- 590GB N-Triples file
- Gzipped: 26GB
- 4'000'046'284 Triples

Triples per second: > **210’000**!

# How about RDF triplestores?

- No known option that reaches this speed (DuckDB + Ontop)
- Discussions started with QLever team
- Check out presentation by Hannah Bast later today

# Conclusions: Why RDF cubes?

Beyond CSV & co

- Clear semantics for **everyone else**
- Make implicit knowledge explicit
- Virtual SQL tables also benefits!

Data becomes discoverable!

# Conclusions: Why RDF cubes?

RDF & SPARQL is on the web (or Intranet):

- Same dimensions can be truly shared
  - “Set of terms”
  - Taxonomies
- Relate concepts
  - owl:sameAs
  - Or any other relation that makes sense
- Virtual SQL tables also benefits!

Data becomes discoverable!

# Thank you!

Questions?