

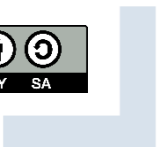


SPHN RDF Training Primer

Vasundra Touré and Philip Krauss

Data interoperability team, Personalized Health Informatics
Group, SIB Swiss Institute of Bioinformatics

A project of



Agenda

1. Strategy

2. Basic Components

- Ontology
- RDF
- SPARQL

Wishlist : F.A.I.R. and more

Schema Softness

- I want to be able to describe data but allow certain levels of freedom
- I don't want to be limited by tables

Ontologies

Accessibility

- I want that all information is stored and queried in open formats and can be read also in 10+ years

RDF + SPARQL

Interoperability

- I want that information can be created, stored, exchanged, merged, read and queried by different systems, leading to the same information
- I want that information that means the same thing is the same thing
- I want make use of information that is already available

Semantic Web

Semantic Web – a vision in 2001

Semantic Web, a visionary concept proposed in 2001 by Sir Tim Berners-Lee, the inventor of the World Wide Web

“I have a dream for the Web [in which computers] become capable of analyzing all the data on the Web – the content, links, and transactions between people and computers. ..”

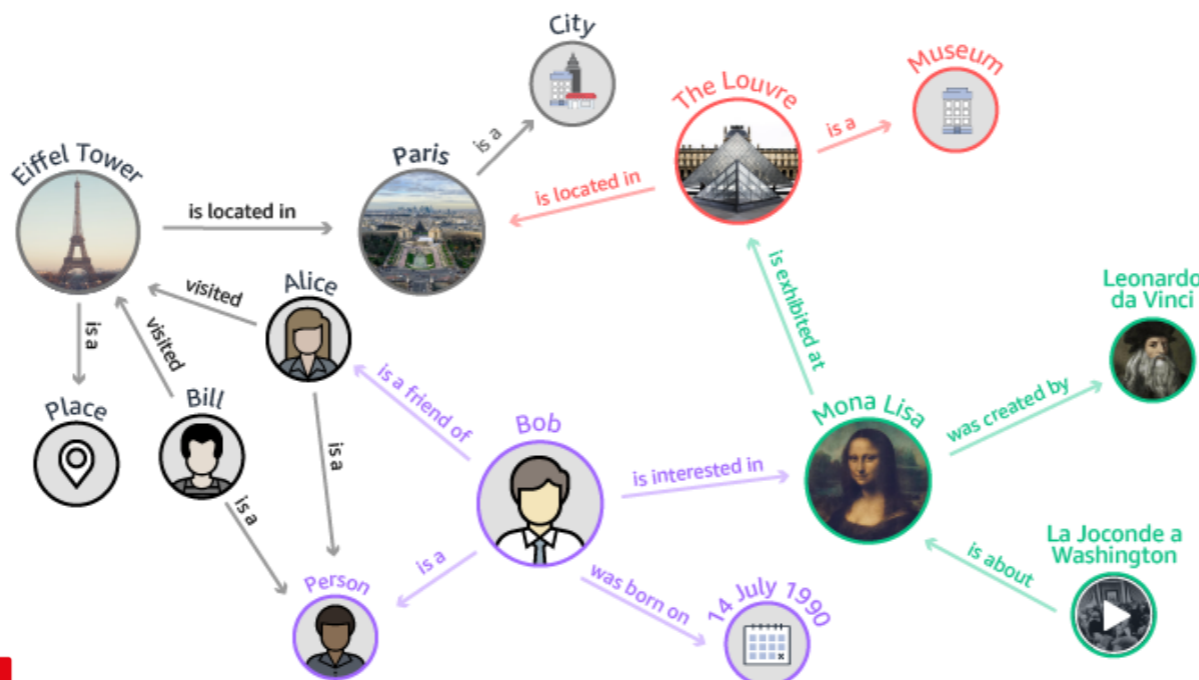


I Want To Reinvent The **Web**

Sir Tim Berners-Lee
World Wide Web Creator

FOSSBYTES

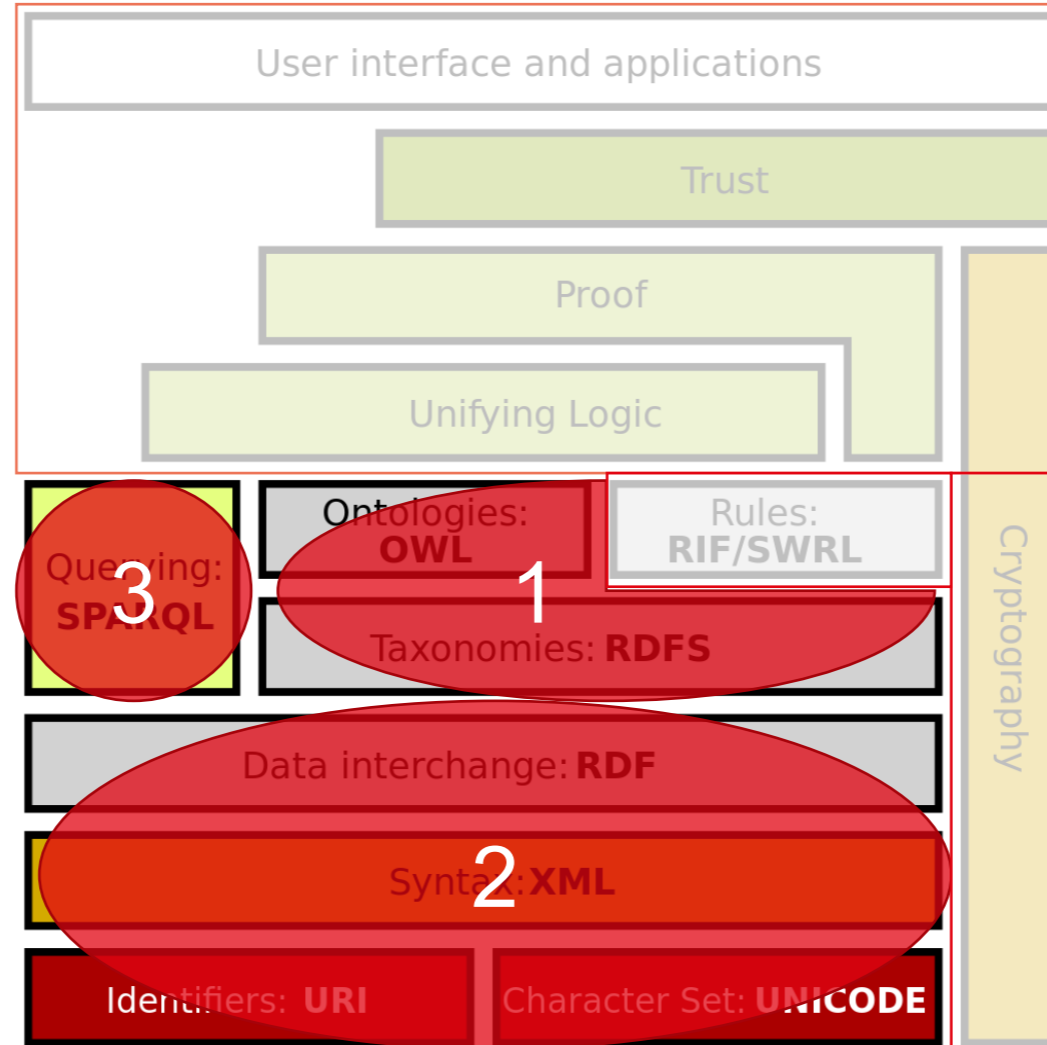
Sir Tim Berners-Lee, fossbytes.com, 2016



Amazon illustrates Knowledge Graph in Neptune

Semantic Web Layer Cake

1. Ontologies
2. RDF
3. SPARQL

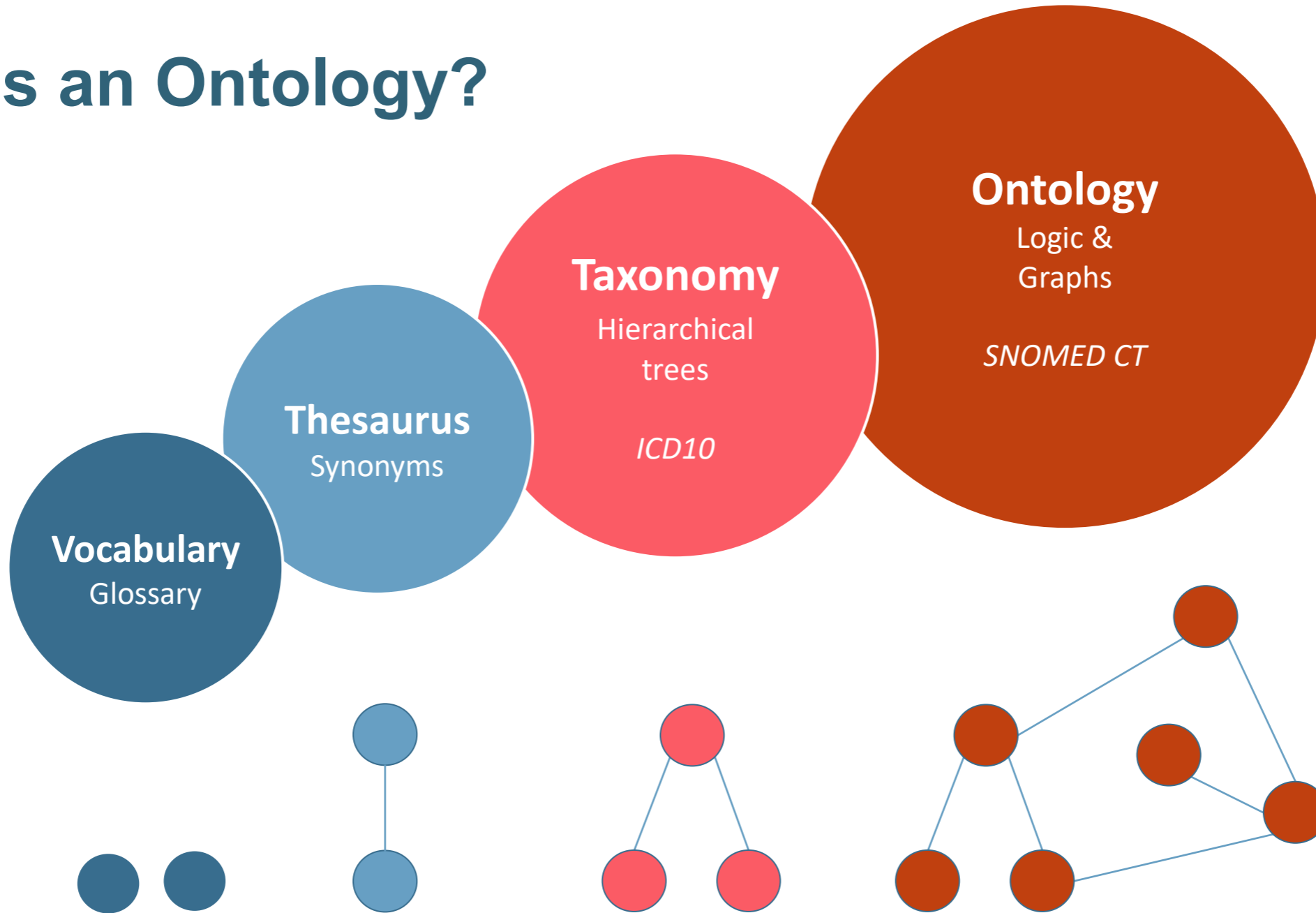


Ontologies

e.g. OWL Ontology Web Language
A W3C Standard

“How to formally describe Meaning of Information”

What is an Ontology?



What is an Ontology?

Descriptions + Rules how a thing is connected to another thing

Similar to a table definition but on graphs and

- Potentially describing a formal interpretation of a thing
- Not necessarily defining everything completely

For all the Relational Database people, the ontology can be compared to an ER (Entity Relationship) Model with some extensions.

Let's show it by browsing the SPHN Ontology

sphn (https://biomedit.ch/rdf/sphn-ontology/sphn/2021/1) : [C:\Users\Philip Schledermann\OneDrive - Trivadis AG\Documents\02_Projects\35_SIB\new_ontology_2021-1\sphn-ontology_202...

File Edit View Reasoner Tools Refactor Window Help

< > sphn (https://biomedit.ch/rdf/sphn-ontology/sphn/2021/1) Search...

> SPHN Concept > Allergy

Active ontology x Entities x Classes x Object properties x Annotation properties x Individuals by class x DL Query x

Annotation properties Datatypes Individuals Classes Object properties Data properties

Allergy — https://biomedit.ch/rdf/sphn-ontology/sphn#Allergy

Annotations Usage Manchester syntax entity rendering

Class hierarchy: Allergy

- owl:Thing
 - SPHN Concept
 - Administrative Case
 - Administrative Gender
 - Adverse Event
 - Adverse reaction (disorder)
 - Allergy**
 - Allergy Episode
 - Assessment score (observable entity)
 - Biobanksample
 - Biosample
 - Birth Date
 - Birth date
 - Body Site
 - Body site
 - Body structure (body structure)
 - Care Handling
 - Care regime (regime/therapy)
 - Civil Status
 - Clinical specialty (qualifier value)
 - Code
 - Consent
 - Consent Document
 - Consent report (record artifact)
 - Data Determination
 - Data Provider Institute
 - Data Release
 - Date of birth (observable entity)
 - Dead (finding)
 - Death Status
 - Device (physical object)
 - Drug
 - Drug Administration Event
 - Drug or medication (substance)

Annotations: Allergy

Annotations +

- rdfs:label Allergy
- rdfs:comment risk of harmful or undesirable, physiological response which is unique to an individual and associated with exposure to a substance

Description: Allergy

Equivalent To +

- 'Propensity to adverse reactions to substance (finding)'

SubClass Of +

- 'SPHN Concept'

General class axioms +

SubClass Of (Anonymous Ancestor)

- Allergy
- 'Propensity to adverse reaction (finding)' and ("Role group (attribute) some ('Associated with (attribute)' some 'Substance (substance)'))

Instances +

Target for Key +

To use the reasoner click Reasoner > Start reasoner Show Inferences

What we can learn from the Allergy definition

An Allergy

- is connected to a patient using the hasSubjectPseudoidentifier relation
- is connected to a hospital using the hasDataProviderInstitute relation
- is connected to a substance using the hasSubstance relation
- Substances defined in SPHN are the same as substances as described SNOMED CT

This ontology does not say anything about

- Serialization/Datatypes
- Cardinalities

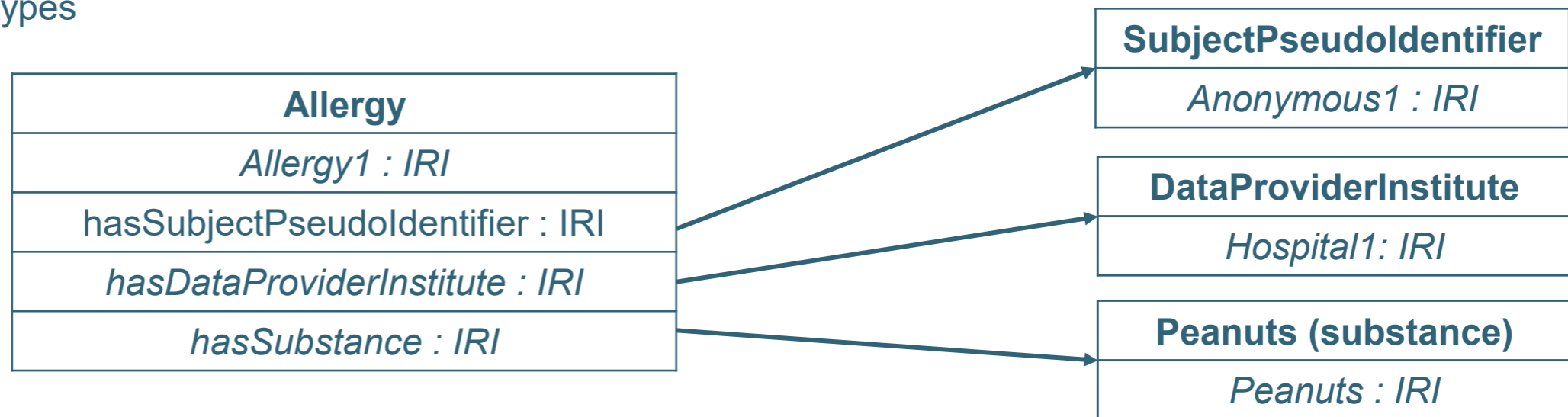
Especially for the Relational Database people

An Allergy

- is connected to a patient using the hasSubjectPseudoidentifier relation
- is connected to a hospital using the hasDataProviderInstitute relation
- is connected to a substance using the hasSubstance relation
- Substances defined in SPHN are the same as substances as described SNOMED CT

This ontology does not say anything about

- Serialization/Datatypes
- Cardinalities



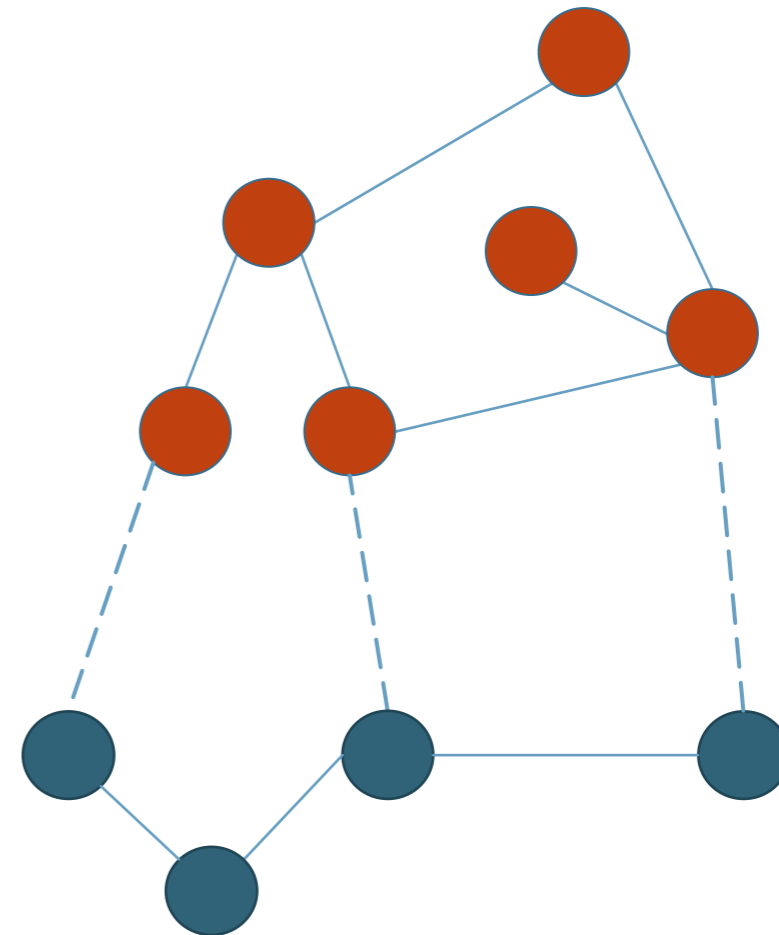
Ontology vs Instance data

Ontology

- describes how objects relates to each other
- formalizes the meaning
- stored in RDF or OWL
- SPHN
 - Provided by SPHN or adapted by projects
 - Also called “SPHN RDF schema”

Instance data

- instantiates the objects from the ontology
- Modeled data according to the ontology
- SPHN
 - Hospitals produce instance data
 - Stored in RDF and transferred to project environment



Spoiler: You will see in a few slides what the dashed lines mean...

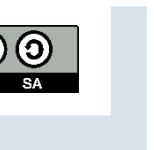


RDF

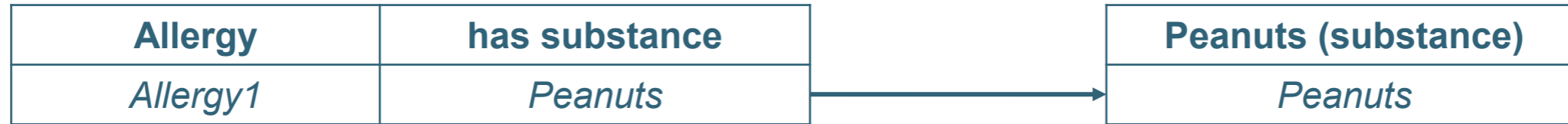
Resource Description Framework A W3C Standard

“How to represent and exchange Information”

A project of



Classical Tables to express information

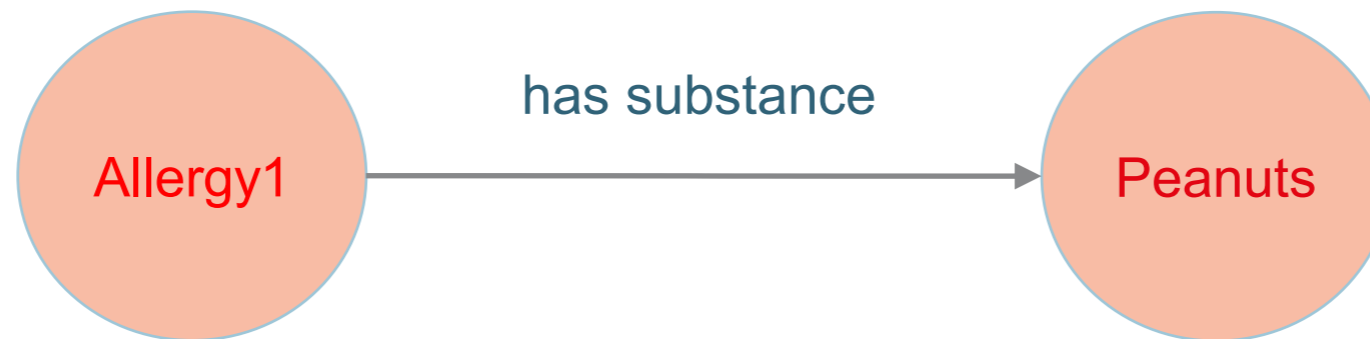
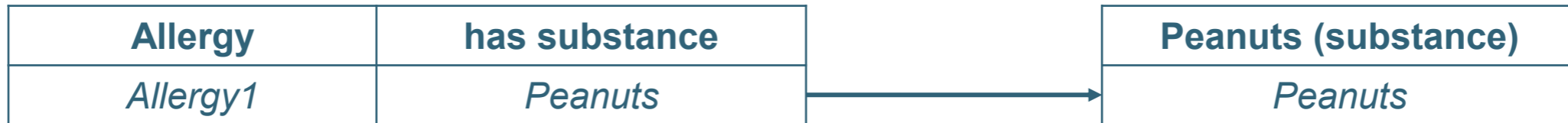


Allergy1 is an instance of the **Allergy**

Allergy1 **has substance** *Peanuts*

Peanuts is an instance of the **Peanuts substance**

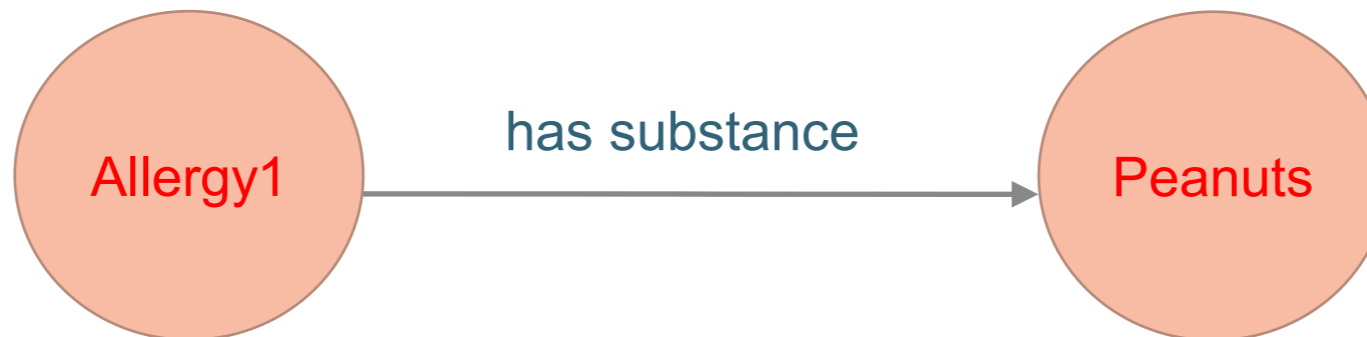
But we can also use Graphs



RDF Elements <-> How to build a Graph

IRIs	<code><http://sib.swiss/allergies/Allergy1></code>	<p>IRIs (Internationalized Resource Identifiers) are a more generic form of URIs. They identify things like keys, and not need to be resolvable.</p>
Literals	<code>"Peanut"^^<http://www.w3.org/2001/XMLSchema#string></code>	<p>Everything that is not a thing but a describing element. We use the XML Schema namespace for that.</p>
Blank Nodes	<code>_:alice</code>	<p>A key that is only valid inside the data. You can typically not query by a blank node itself.</p>

How do we combine them?

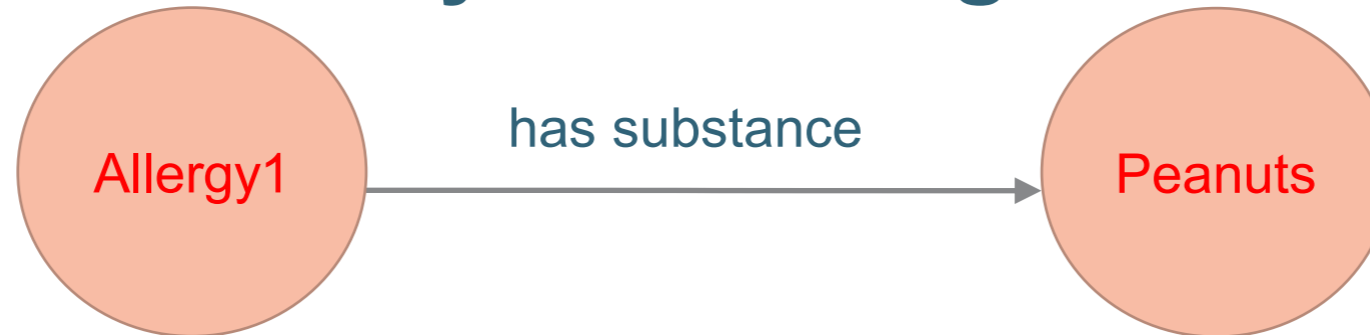


Subject	Predicate	Object
<i>IRI or Blank Node</i>	<i>IRI</i>	<i>IRI, Blank Node or Literal</i>
<http://sib.swiss/allergies/Allergy1>	<https://biomedit.ch/rdf/sphn-ontology/sphn#hasSubstance>	<http://sib.swiss/substances/Peanuts>

If you want to dig into the specification: <https://www.w3.org/TR/rdf11-concepts/>

Is there a nicer way?

Namespaces as Syntactic Sugar



	Subject	Predicate	Object
	<i>[prefixed] IRI or Blank Node</i>	<i>[prefixed] IRI</i>	<i>[prefixed] IRI, Blank Node or Literal</i>
1	<http://sib.swiss/allergies/Allergy1>	<https://biomedit.ch/rdf/sphn-ontology/sphn#hasSubstance>	<http://sib.swiss/substances/Peanuts>
2	PREFIX substances : <http://sib.swiss/substances/>		
3	PREFIX sphn : <https://biomedit.ch/rdf/sphn-ontology/sphn#>		
4	<http://sib.swiss/allergies/Allergy1>	sphn:hasSubstance	substances:Peanuts

Just be aware not all serializations support it. Turtle (ttl) for example does.

Start creating some data

We want to model the following information:

- A patient (“anonymous1”) that came to our fictive hospital (“hospital1”)
- “anonymous1” has an allergy against peanuts
- Peanuts are a substance
- We want to declare that our peanut substance is an instance of SNOMED CT Code “762952008”

Class Hierarchy

- owl:Thing
 - SPHN Concept
 - Administrative Case
 - Administrative Gender
 - Adverse Event
 - Allergy
 - Allergy Episode
 - Biobanksample
 - Biosample
 - Birth Date
 - Body Site
 - Care Handling
 - Civil Status
 - Code
 - Consent
 - Data Determination
 - Data Provider Institute
 - Data Release
 - Death Status
 - Drug
 - Drug Administration Event
 - Drug Prescription
 - Duration
 - FOPH Diagnosis
 - FOPH Procedure
 - Frequency
 - Gestational Age at Birth
 - Healthcare Encounter
 - ICD-O Diagnosis
 - Inhaled Oxygen Concentration
 - Intent
 - Lab Result
 - Location
 - Measurement
 - Measurement Method
 - Medical Device
 - Nursing Diagnosis
 - Oncology Treatment Assessment
 - Problem Condition
 - Radiotherapy Procedure
 - Simple Score
 - Subject Pseudo Identifier
 - Substance
 - Substance Amount
 - TNM Classification
 - Terminology
 - Therapeutic Area
 - Time Pattern
 - Tumor Grade
 - Tumor Specimen

Entity Usage: Allergy

Entity	Type	Axiom Type	Axiom
			<ul style="list-style-type: none"> or 'Biosample' or 'Birth Date' or 'Body Height' or 'Body Temperature' or 'Body Weight' or 'Catheter' or 'Central Venous Pressure' or 'Circumference Measure' or 'Civil Status' or 'Consent' or 'Death Status' or 'Drug Administration Event' or 'Drug Prescription' or 'FOPH Diagnosis' or 'FOPH Procedure' or 'Healthcare Encounter' or 'Heart Rate' or 'ICD-O Diagnosis' or 'Inhaled Oxygen Concentration' or 'Lab Result' or 'Nursing Diagnosis' or 'Oncology Treatment Assessment' or 'Oxygen Saturation' or 'Problem Condition' or 'Radiotherapy Procedure' or 'Respiratory Rate' or 'Simple Score' or 'Systemic Arterial Blood Pressure' or 'TNM Classification' or 'Tumor Grade' or 'Tumor Stage'
has subject pseudo identifier	Object Property	Domain (rdfs:domain)	'has subject pseudo identifier' Domain ('Administrative Gender' or Allergy or Biobanksample or Biosample or 'Birth Date' or Catheter or 'Civil Status' or Consent or 'Death Status' or Drug or 'Drug Administration Event' or 'Drug Prescription' or Duration or 'FOPH Diagnosis' or 'FOPH Procedure' or Frequency or 'Healthcare Encounter' or 'ICD-O Diagnosis' or 'Lab Result' or Measurement or 'Nursing Diagnosis' or 'Oncology Treatment Assessment' or 'Problem Condition' or 'Radiotherapy Procedure' or 'Simple Score' or 'Substance Amount' or 'TNM Classification' or 'Time Pattern' or 'Tumor Grade' or 'Tumor Stage')
has substance	Object Property	Domain (rdfs:domain)	'has substance' Domain (Allergy or 'Allergy Episode')

Class: Allergy

IRI
https://biomedit.ch/rdf/sphn-ontology/sphn#Allergy

Annotations

- rdfs:label Allergy lang
- rdfs:comment risk of harmful or undesirable, physiological response which is unique to an individual and associated with exposure to a substance lang
- Ent: Enter value lang

Parents

- SPHN Concept
- Enter a class name

Relationships

- Ent: Enter value lang

Have a look at the ontology first

The Allergy seems to be the central point

- connects to the Person via the hasSubjectPseudoidentifier
- connects to the Hospital via the hasDataProviderInstitute
- connects to the Substance via the hasSubstance

Create IRIs for the instances

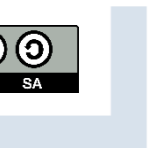
	Name	IRI
Patient	anonymous1	http://sib.swiss/fictivePatients/ anonymous1
Hospital	hospital1	http://sib.swiss/ hospital1
Allergy	allergy1	http://sib.swiss/allergies/ allergy1
Peanuts	peanuts1	http://sib.swiss/substances/ peanuts1

- anonymous1
- hospital1
- allergy1
- peanuts1



What are we missing?

A project of



Connecting elements! -> Predicates

Name	IRI
hasSubstance	https://biomedit.ch/rdf/sphn-ontology/sphn#hasSubstance
hasSubjectPseudoidentifier	https://biomedit.ch/rdf/sphn-ontology/sphn#hasSubjectPseudoidentifier
hasDataProviderInsitute	https://biomedit.ch/rdf/sphn-ontology/sphn#hasDataProviderInsitute

hasSubstance →

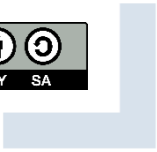
hasSubjectPseudoidentifier →

hasDataProviderInsitute →



We are still missing something...

A project of



Semantic types: Instantiating Things of the Ontology

short form	IRI
rdf:type	http://www.w3.org/1999/02/22-rdf-syntax-ns#type

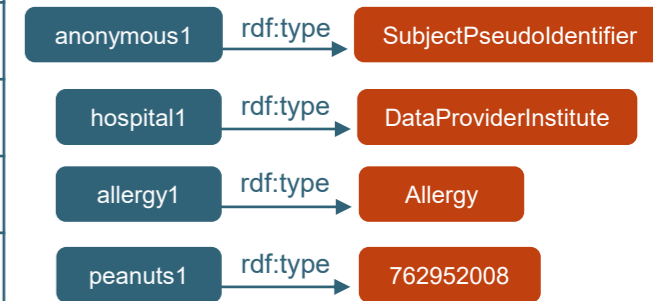
rdf:type →

That's the dashed lines you saw earlier!

Putting it all together

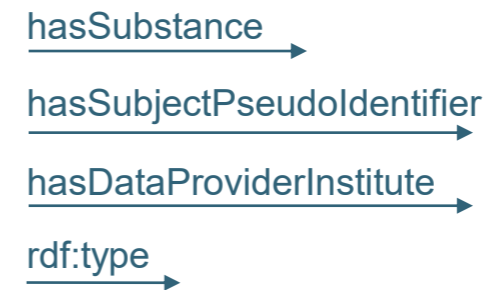
Subjects or Objects:

Name	IRI	Type
anonymous1	http://sib.swiss/fictivePatients/ anonymous1	sphn: SubjectPseudoidentifier
hospital1	http://sib.swiss/ hospital1	sphn: DataProviderInstitute
peanuts1	http://sib.swiss/allergies/ allergy1	sphn: Allergy
peanuts1	http://sib.swiss/substances/ peanuts1	snomed: 762952008



Predicates:

Name	IRI
hasSubstance	sphn: hasSubstance
hasSubjectPseudoidentifier	sphn: hasSubjectPseudoidentifier
hasDataProviderInstitute	sphn: hasDataProviderInstitute
type	rdf: type



sphn, snomed, rdf Prefixes are omitted on this page

A valid Turtle file

```

@prefix allergies: <http://sib.swiss/allergies/> .
@prefix patients: <http://sib.swiss/fictivePatients/> .
@prefix substances: <http://sib.swiss/substances/> .
@prefix sib: <http://sib.swiss/> .
@prefix sphn: <https://biomedit.ch/rdf/sphn-ontology/sphn#> .
@prefix snomed: <http://snomed.info/id/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

```

```

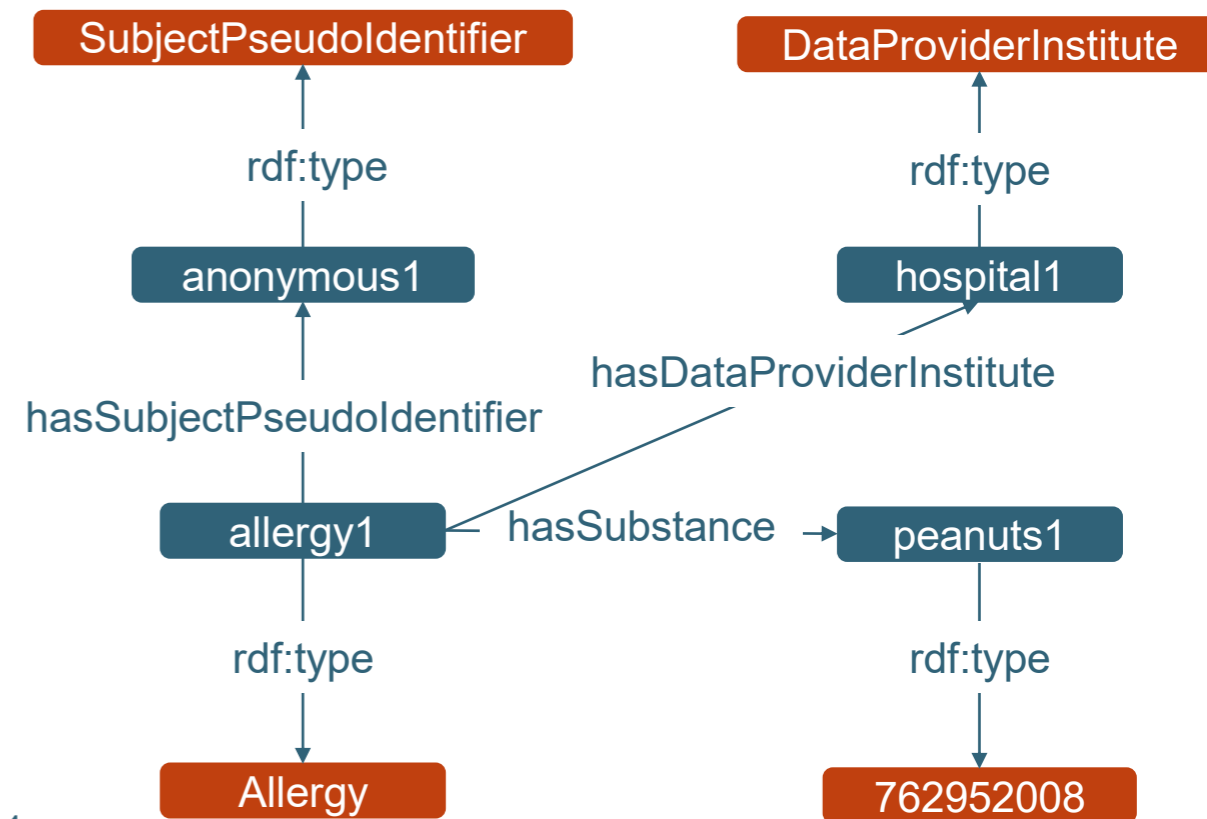
# types
patients:anonymous1 rdf:type sphn:SubjectPseudoidentifier .
sib:hospital1 rdf:type sphn:DataProviderInstitute .
allergies:allergy1 rdf:type sphn:Allergy .
substances:peanuts1 rdf:type snomed:762952008 .

```

```

# relations to the allergy
allergies:allergy1 sphn:hasSubjectPseudoidentifier patients:anonymous1 .
allergies:allergy1 sphn:hasDataProviderInstitute sib:hospital1 .
allergies:allergy1 sphn:hasSubstance substances:peanuts1 .

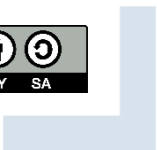
```





Loading into a DB

DBs for storing RDF are triplestores

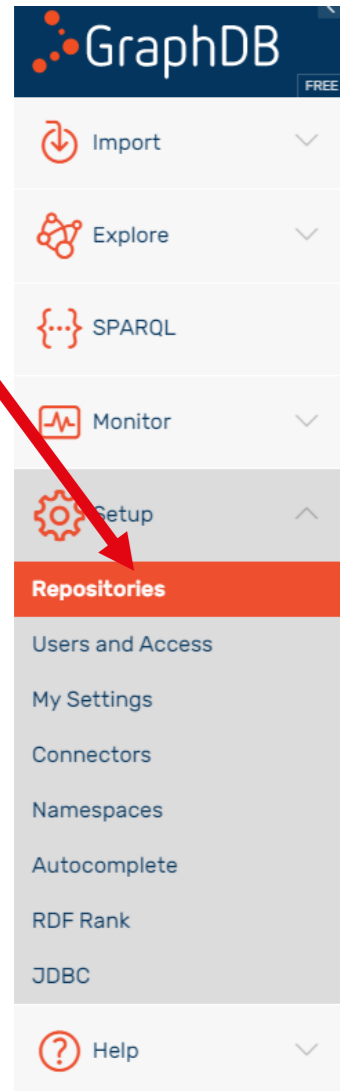


Examples of Triplestores



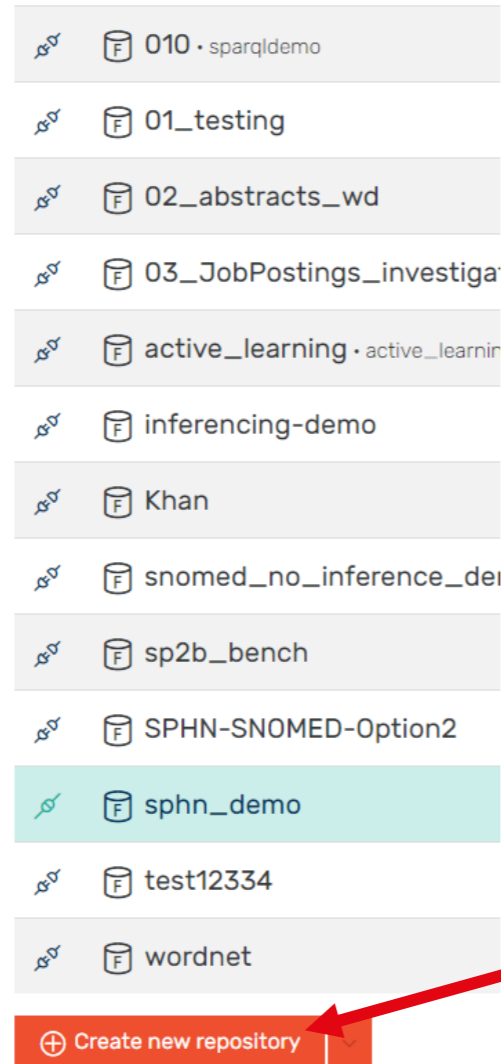
+ some more

Example with GraphDB Free (1/8)



GraphDB FREE

- Import
- Explore
- SPARQL
- Monitor
- Setup
- Repositories**
- Users and Access
- My Settings
- Connectors
- Namespaces
- Autocomplete
- RDF Rank
- JDBC
- Help



- 010 · sparqldemo
- 01_testing
- 02_abstracts_wd
- 03_JobPostings_investiga
- active_learning · active_learnir
- inferencing-demo
- Khan
- snomed_no_inference_dei
- sp2b_bench
- SPHN-SNOMED-Option2
- sphn_demo**
- test12334
- wordnet
- Create new repository

Create Repository

Repository properties

Repository ID*

Repository title

Type

Storage folder

Ruleset

Disable owl:sameAs

Supports SHACL validation

Base URL

Entity index size

Use predicate indices Cache literal language tags

Use context index Enable literal index

1

2

3

Example with GraphDB Free (2/8)

The screenshot shows the GraphDB Free 'Import' interface. A sidebar on the left contains navigation options: Import, RDF (highlighted), Tabular (OntoRefine), Explore, SPARQL, and Monitor. The main area is titled 'Import' and has two tabs: 'User data' and 'Server files'. Three import methods are listed in boxes: 'Upload RDF files' (All RDF formats, up to 200 MB), 'Get RDF data from a URL' (All RDF formats), and 'Import RDF text snippet' (Type or paste RDF data). The top right shows the user 'admin' and the database 'sphn_demo'. Three red circles with arrows point to: 1. The 'sphn_demo' dropdown, 2. The 'RDF' sidebar item, and 3. The 'Import RDF text snippet' box.

Example with GraphDB Free (3/8)

```
@prefix allergies: <http://sib.swiss/allergies/> .
@prefix patients: <http://sib.swiss/fictivePatients/> .
@prefix substances: <http://sib.swiss/substances/> .
@prefix sib: <http://sib.swiss/> .
@prefix sphn: <https://biomedit.ch/rdf/sphn-ontology/sphn#> .
@prefix snomed: <http://snomed.info/id/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
```

```
# types
patients:anonymous1 rdf:type sphn:SubjectPseudoidentifier .
sib:hospital1 rdf:type sphn:DataProviderInstitute .
allergies:allergy1 rdf:type sphn:Allergy .
substances:peanuts1 rdf:type snomed:762952008 .
```

```
# relations to the allergy
allergies:allergy1 sphn:hasSubjectPseudoidentifier patients:anonymous1 .
allergies:allergy1 sphn:hasDataProviderInstitute sib:hospital1 .
allergies:allergy1 sphn:hasSubstance substances:peanuts1 .
```

Import RDF data from a text snippet

```
# types
patients:anonymous1 rdf:type sphn:SubjectPseudoidentifier .
sib:hospital1 rdf:type sphn:DataProviderInstitute .
allergies:allergy1 rdf:type sphn:Allergy .
substances:peanuts1 rdf:type snomed:762952008 .

# relations to the allergy
allergies:allergy1 sphn:hasSubjectPseudoidentifier patients:anonymous1 .
allergies:allergy1 sphn:hasDataProviderInstitute sib:hospital1 .
allergies:allergy1 sphn:hasSubstance substances:peanuts1 .
```

Start import automatically

Cancel Format: Turtle ▾ **Import**

Example with GraphDB Free (4/8)

Import settings ✕

Base IRI ⓘ

Target graphs ⓘ From data The default graph Named graph

Enable replacement of existing data


[Show advanced settings](#) ▾

[Restore defaults](#)



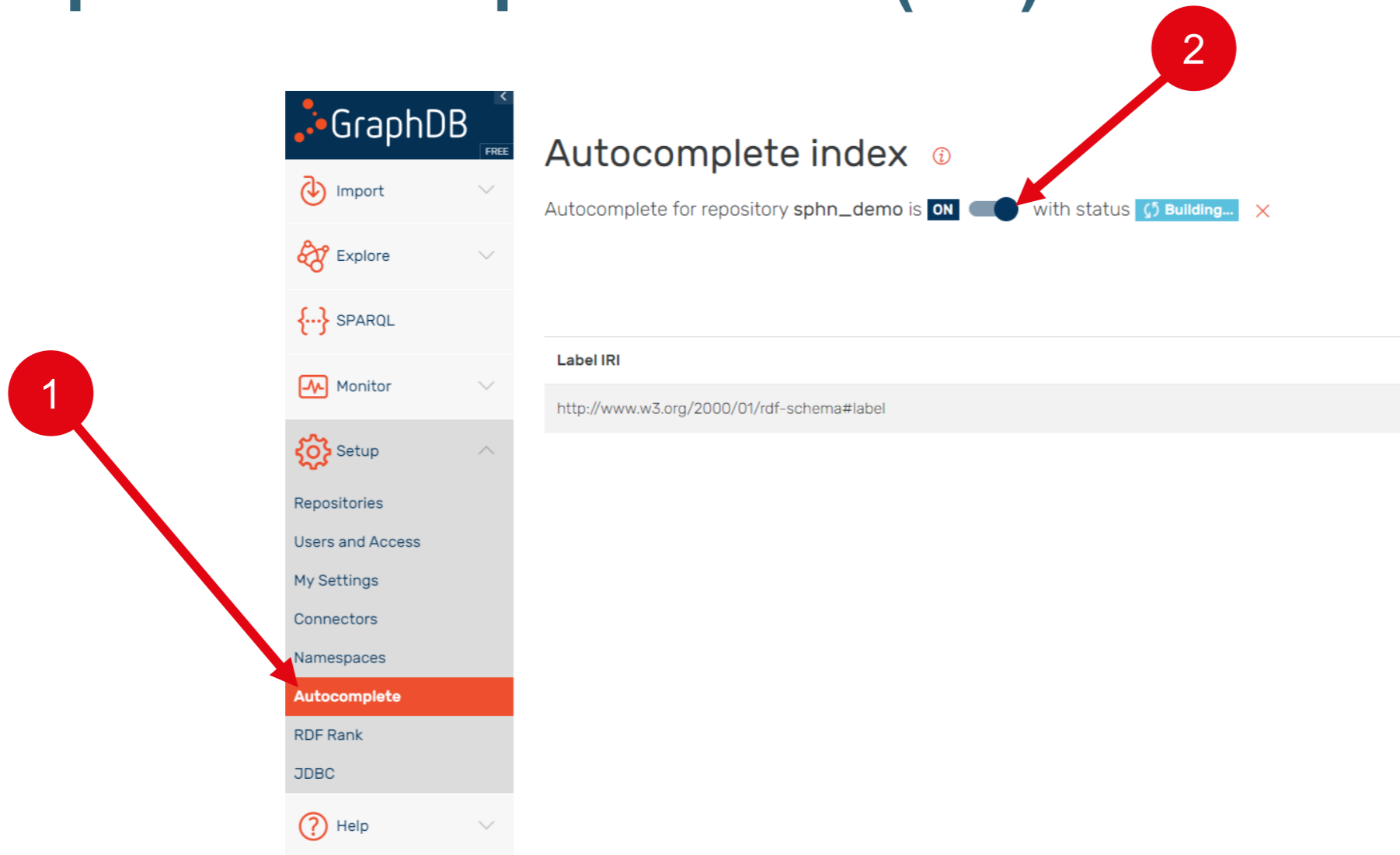
Example with GraphDB Free (5/8)



 Text snippet 2021-04-23 11:50:29.113

   Imported successfully in less than a second.

Example with GraphDB Free (6/8)



GraphDB FREE

- Import
- Explore
- SPARQL
- Monitor
- Setup**
 - Repositories
 - Users and Access
 - My Settings
 - Connectors
 - Namespaces
 - Autocomplete**
 - RDF Rank
 - JDBC
- Help

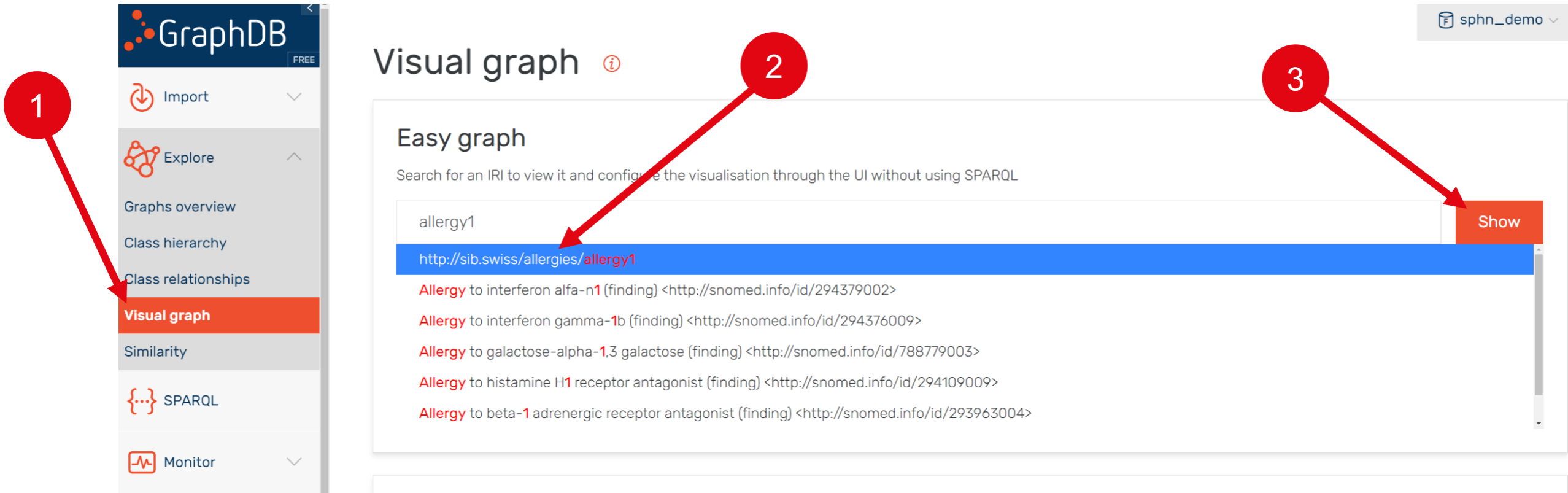
Autocomplete index ⓘ

Autocomplete for repository sphn_demo is **ON** with status **Building...** ✕

Label IRI

http://www.w3.org/2000/01/rdf-schema#label

Example with GraphDB Free (7/8)



1 Visual graph

2 Search for an IRI to view it and configure the visualisation through the UI without using SPARQL

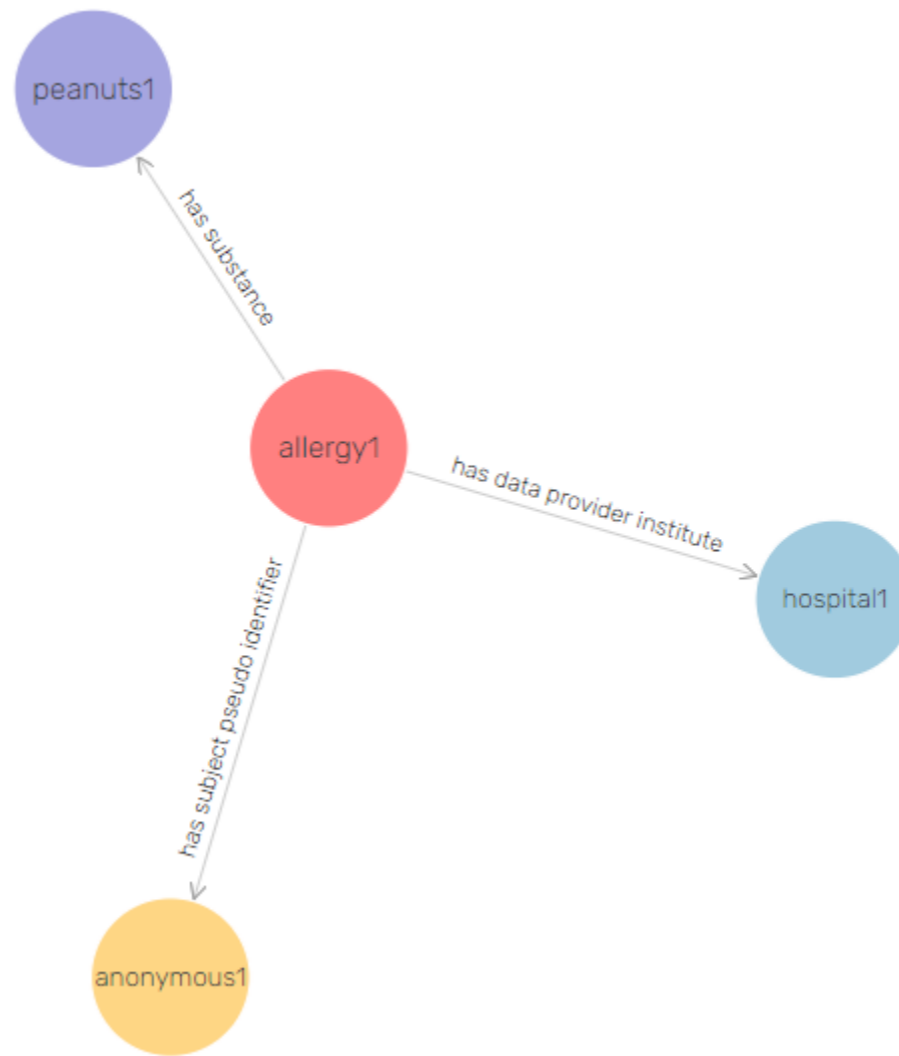
3 Show

allergy1

<http://sib.swiss/allergies/allergy1>

- Allergy to interferon alfa-n1 (finding) <http://snomed.info/id/294379002>
- Allergy to interferon gamma-1b (finding) <http://snomed.info/id/294376009>
- Allergy to galactose-alpha-1,3 galactose (finding) <http://snomed.info/id/788779003>
- Allergy to histamine H1 receptor antagonist (finding) <http://snomed.info/id/294109009>
- Allergy to beta-1 adrenergic receptor antagonist (finding) <http://snomed.info/id/293963004>

Example with GraphDB Free (8/8)



Some Information

I loaded the ontology beforehand ;-)
You should do that as well in your projects so you will be able to get all the information about types etc!

We have shown the easiest way for the demo to get data inside a nearly fresh GraphDB instance here. This is mainly for testing. There exist other options.

Have a look at the documentation of your database e.g. <https://graphdb.ontotext.com/documentation/free/index.html>

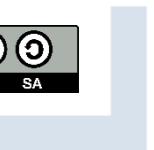
GraphDB Free is a Free of use for personal usage. It is not open-source. You can apply for a license using <https://www.ontotext.com/products/graphdb/graphdb-free/>



SPARQL

SPARQL Protocol and RDF Query Language A W3C Standard

“How to query Information”



The SPARQL Syntax

prefix dc: <...> ← Prefix declarations
prefix uni: <...>

SELECT ... ← Declare type of query: SELECT, ASK, DESCRIBE, CONSTRUCT

FROM <...> ← Define dataset
FROM NAMED <...>

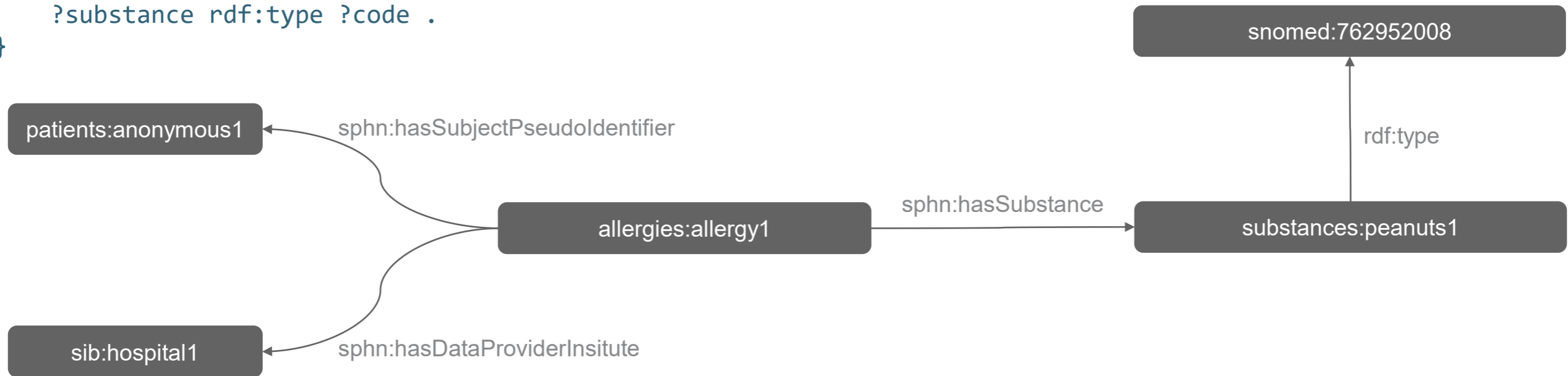
WHERE { ... } ← Graph pattern

ORDER BY ... ← Query modifiers
HAVING ...
GROUP BY ...
LIMIT ...
OFFSET ...
BINDINGS ...

Example Query Evaluation (1/5)

```
PREFIX patients: <http://sib.swiss/fictivePatients/>
PREFIX sphn: <https://biomedit.ch/rdf/sphn-ontology/sphn#>
PREFIX sib: <http://sib.swiss/>
```

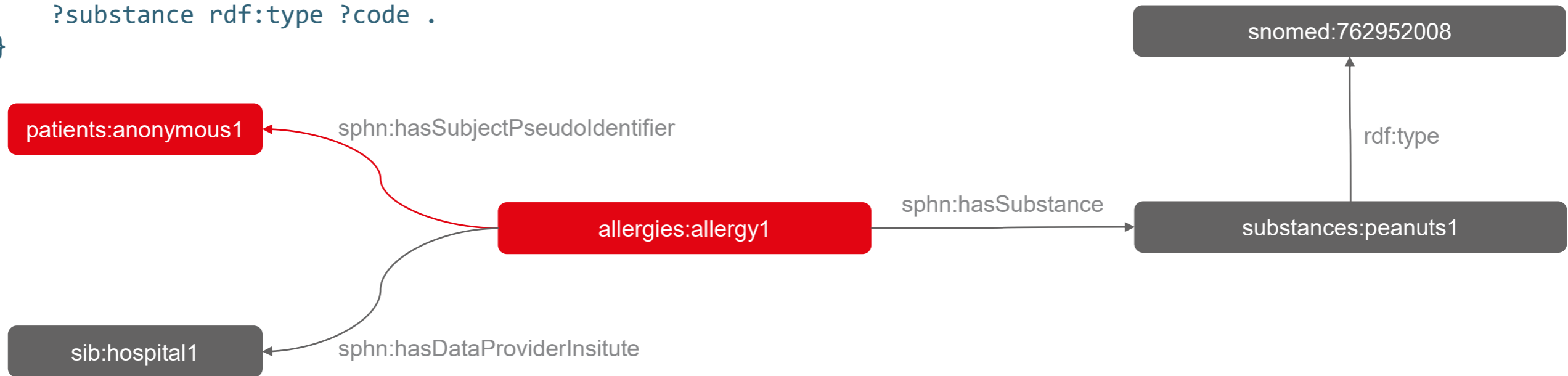
```
SELECT ?code
WHERE {
  ?allergy sphn:hasSubjectPseudoIdentifier patients:anonymous1 .
  ?allergy sphn:hasSubstance ?substance .
  ?substance rdf:type ?code .
}
```



Example Query Evaluation (2/5)

```
PREFIX patients: <http://sib.swiss/fictivePatients/>
PREFIX sphn: <https://biomedit.ch/rdf/sphn-ontology/sphn#>
PREFIX sib: <http://sib.swiss/>
```

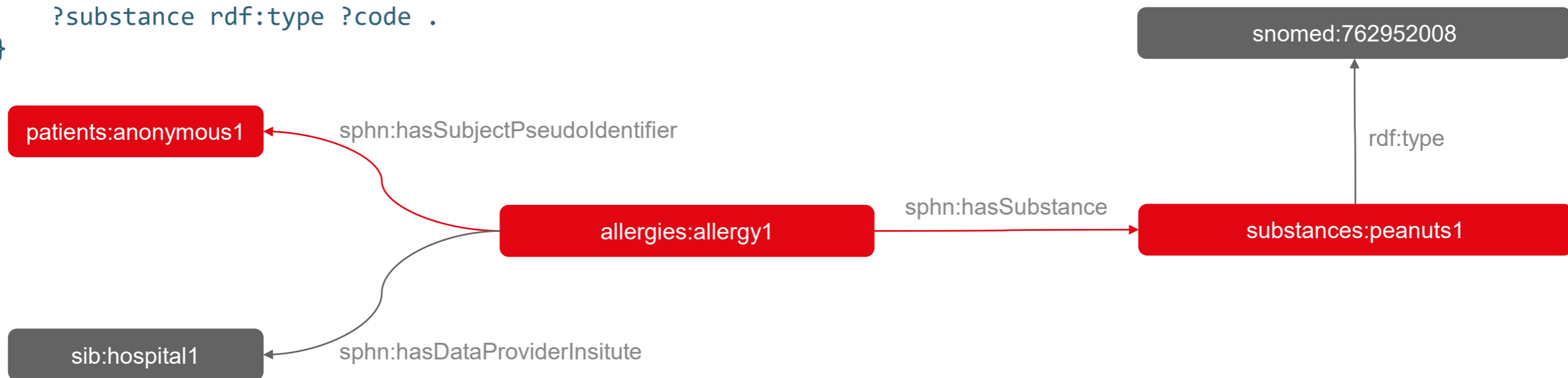
```
SELECT ?code
WHERE {
  ?allergy sphn:hasSubjectPseudoIdentifier patients:anonymous1 .
  ?allergy sphn:hasSubstance ?substance .
  ?substance rdf:type ?code .
}
```



Example Query Evaluation (3/5)

```
PREFIX patients: <http://sib.swiss/fictivePatients/>
PREFIX sphn: <https://biomedit.ch/rdf/sphn-ontology/sphn#>
PREFIX sib: <http://sib.swiss/>
```

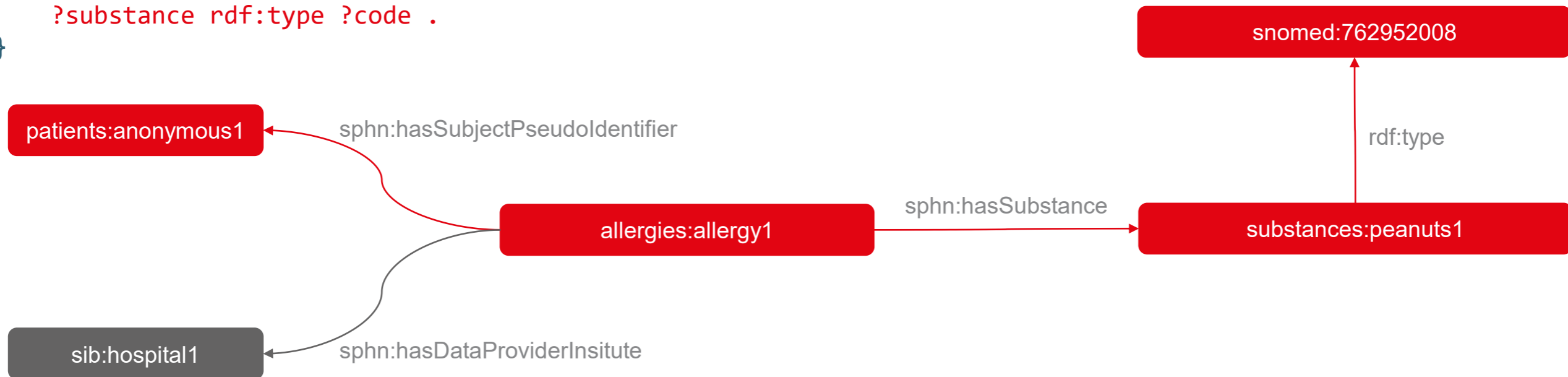
```
SELECT ?code
WHERE {
  ?allergy sphn:hasSubjectPseudoIdentifier patients:anonymous1 .
  ?allergy sphn:hasSubstance ?substance .
  ?substance rdf:type ?code .
}
```



Example Query Evaluation (3/5)

```
PREFIX patients: <http://sib.swiss/fictivePatients/>
PREFIX sphn: <https://biomedit.ch/rdf/sphn-ontology/sphn#>
PREFIX sib: <http://sib.swiss/>
```

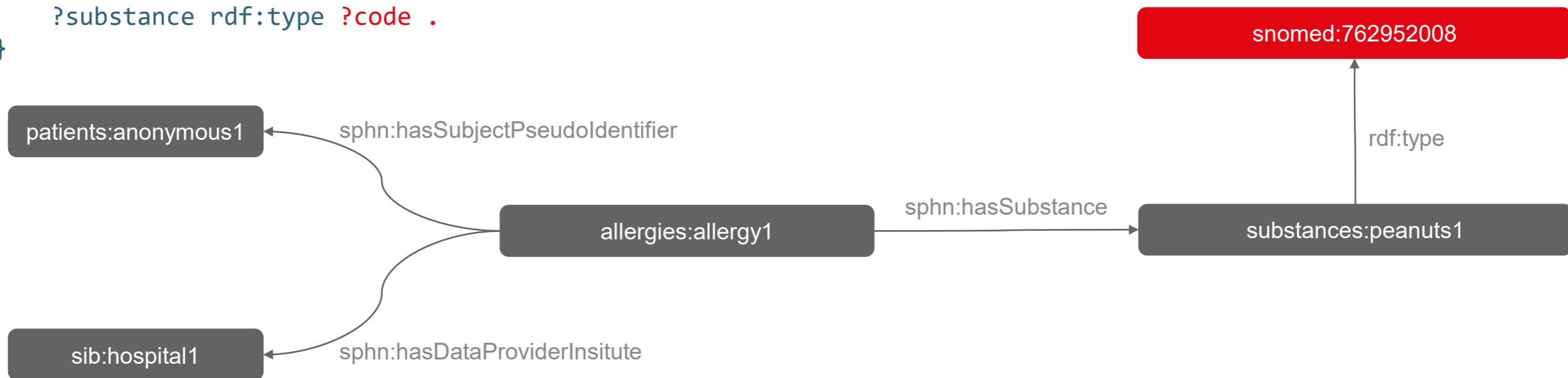
```
SELECT ?code
WHERE {
  ?allergy sphn:hasSubjectPseudoIdentifier patients:anonymous1 .
  ?allergy sphn:hasSubstance ?substance .
  ?substance rdf:type ?code .
}
```



Example Query Evaluation (3/5)

```
PREFIX patients: <http://sib.swiss/fictivePatients/>
PREFIX sphn: <https://biomedit.ch/rdf/sphn-ontology/sphn#>
PREFIX sib: <http://sib.swiss/>
```

```
SELECT ?code
WHERE {
  ?allergy sphn:hasSubjectPseudoIdentifier patients:anonymous1 .
  ?allergy sphn:hasSubstance ?substance .
  ?substance rdf:type ?code .
}
```



Let's do it live (1/2)

SPARQL Query & Update

sphn_demo admin

Editor only Editor and results Results only

Unnamed × Unnamed × Unnamed × Unnamed × Unnamed × Unnamed × Unnamed × Unnamed × Unnamed × Unnamed ×

Unnamed × Unnamed × Unnamed × Unnamed × Unnamed × ⊕

```
1 PREFIX patients: <http://sib.swiss/fictivePatients/>
2 PREFIX sphn: <https://biomedit.ch/rdf/sphn-ontology/sphn#>
3 PREFIX sib: <http://sib.swiss/>
4
5 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
6 SELECT ?code
7 WHERE {
8   ?allergy sphn:hasSubjectPseudoIdentifier patients:anonymous1 .
9   ?allergy sphn:hasSubstance ?substance .
10  ?substance rdf:type ?code .
11 }
12
```

3 Run

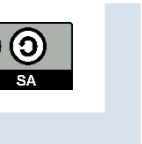
Let's do it live (2/2)

	code
1	snomed:762952008



Test

A project of



What do we need to change in the previous query to get the IRI of the Hospital?

```
PREFIX patients: <http://sib.swiss/fictivePatients/>
PREFIX sphn: <https://biomedit.ch/rdf/sphn-ontology/sphn#>
PREFIX sib: <http://sib.swiss/>
```

```
SELECT ?code
WHERE {
    ?allergy sphn:hasSubjectPseudoIdentifier patients:anonymous1 .
    ?allergy sphn:hasSubstance ?substance .
    ?substance rdf:type ?code .
}
```

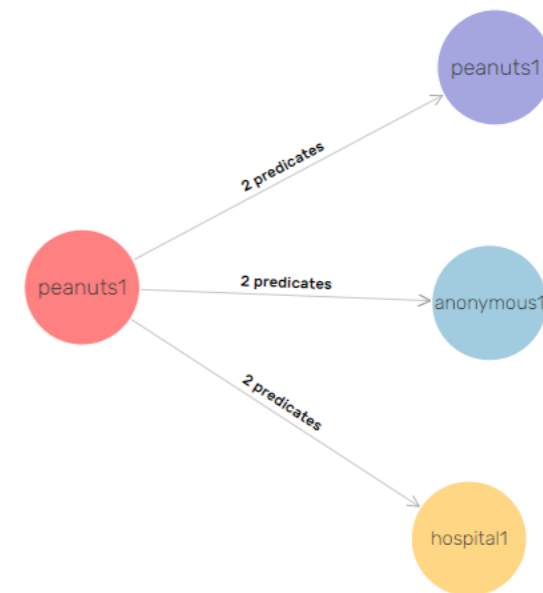
What do we need to change in the previous query to get the IRI of the Hospital?

```
PREFIX patients: <http://sib.swiss/fictivePatients/>
PREFIX sphn: <https://biomedit.ch/rdf/sphn-ontology/sphn#>
PREFIX sib: <http://sib.swiss/>
```

```
SELECT ?code
WHERE {
  ?allergy sphn:hasSubjectPseudoIdentifier patients:anonymous1 .
  ?allergy sphn:hasSubstance ?substance .
  ?substance rdf:type ?code .
}
```

Possible relations:

rdf:type
 sphn:hasSubjectPseudoIdentifier
 sphn:hasDataProviderInstitute
 sphn:hasSubstance



Slide to edit live in the session

This one






```
PREFIX patients: <http://sib.swiss/fictivePatients/>  
PREFIX sphn: <https://biomedit.ch/rdf/sphn-ontology/sphn#>  
PREFIX sib: <http://sib.swiss/>
```

```
SELECT ?hospital  
WHERE {  
    ?allergy sphn:hasSubjectPseudoIdentifier patients:anonymous1 .  
    ?allergy sphn:hasDataProviderInstitute ?hospital .  
}
```

```

1 PREFIX patients: <http://sib.swiss/fictivePatients/>
2 PREFIX sphn: <https://biomedit.ch/rdf/sphn-ontology/sphn#>
3 PREFIX sib: <http://sib.swiss/>
4
5 SELECT ?hospital
6 WHERE {
7     ?allergy sphn:hasSubjectPseudoIdentifier patients:anonymous1 .
8     ?allergy sphn:hasDataProviderInstitute ?hospital .
9 }
10

```

Run

Table
Raw Response
Pivot Table
Google Chart

Download as ▼

Showing results from 1 to 1 of 1. Query took 0.1s, moments ago.

	hospital ▼
1	sib:hospital1

Let's do something fancy:

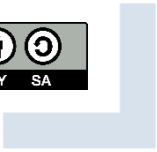
Give me all patients that have an allergy against something in family of Pulse vegetables

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX snomed: <http://snomed.info/id/>
PREFIX sphn: <https://biomedit.ch/rdf/sphn-ontology/sphn#>
```

```
SELECT ?patients
WHERE {
    ?direct_descendants rdfs:subClassOf snomed:227313005 .
    ?substance rdf:type ?direct_descendants .
    ?allergy sphn:hasSubstance ?substance .
    ?allergy sphn:hasSubjectPseudoIdentifier ?patients .
}
```



How do you download as flat data?



For “short” queries

```

3 PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
4 PREFIX snomed: <http://snomed.info/id/>
5 PREFIX sphn: <https://biomedit.ch/rdf/sphn-ontology/sphn#>
6
7 SELECT distinct ?patients
8 WHERE {
9   ?direct_descendants rdfs:subClassOf snomed:227313005 .
10  ?substance rdf:type ?direct_descendants_and_pulse_vegetables .
11  ?allergy sphn:hasSubstance ?substance .
12  ?allergy sphn:hasSubjectPseudoIdentifier ?patients .
13  # Either it is directly the Pulse Vegetables or it is one of its direct subclasses.
14  FILTER(?direct_descendants_and_pulse_vegetables = snomed:227313005 || ?direct_descendants_and_pulse_vegetables in (?direct_descendants))
15 } LIMIT 100
16
  
```

Table Raw Response Pivot Table Google Chart

Filter query results Showing results from 1 to 1 of 1. Query took

	patients
1	patients:anonymous1

- Download as
- JSON
- JSON*
- XML
- CSV
- TSV
- TSV*



For “long” queries

GraphDB FREE

- Import
- Explore
- SPARQL
- Monitor
- Setup
- Help
- REST API**
- Documentation
- Developer Hub
- Support
- System information

REST API documentation i

KDF4J API

repositories : Repository management

sparql : SPARQL

GET /repositories/{repositoryID}

Implementation Notes
The main endpoint that is responsible for sending queries to a particular repository

Response Class (Status 200)
integer

Response Content Type

Parameters

Parameter	Value	Description
repositoryID	<input type="text" value="(required)"/>	The repository ID
query	<input type="text" value="(required)"/>	The query to evaluate
queryLn	<input type="text"/>	Specifies the query line number denoting the query and 'sparql' for SPARQL query
infer	<input type="text" value=""/>	Specifies whether to infer statements (ignoring case)
\$<varname>	<input type="text"/>	Specifies variable name

2

3

Further Information

W3C References

- SPARQL Reference : <https://www.w3.org/TR/sparql11-query/>
- RDF Primer : <https://www.w3.org/TR/rdf11-primer/>
- RDF Concepts & Abstract Syntax : <https://www.w3.org/TR/rdf11-concepts/>

Books

- Bob DuCharme : **Learning SPARQL : Querying and Updating with SPARQL 1.1** : ISBN: 978-1449371432
> *Getting your Hands dirty with SPARQL and RDF*
- Andreas Blumenauer, Helmut Nagy (SWC) : **The Knowledge Graph Cookbook** : ISBN: 978-3-902796-70-7
> *More modern and business focused*
- Jeffrey T. Pollok : **Semantic Web for Dummies** : ISBN: 978-0-470-39679-7
> *“old” but covers all necessary concepts needed, quite good beginner book*

Internet Pages or Papers

- SPARQL 1.1. Cheat Sheet : http://www.iro.umontreal.ca/~lapalme/ift6281/sparql-1_1-cheat-sheet.pdf
- Knowledge Graphs : <https://arxiv.org/pdf/2003.02320.pdf>

Thank you

We'd value feedback on your training experience:

<https://www.sib.swiss/form/view.php?id=89324>