

SPARQLSnap!

querying Knowledge Graphs via Snap!



What we will talk about?

Linked Open Data
and how to query them
via a block based programming environment based on Snap!

Which is the structure of the lesson?

Familiarization

Step-by-step tutorial

Hands-on

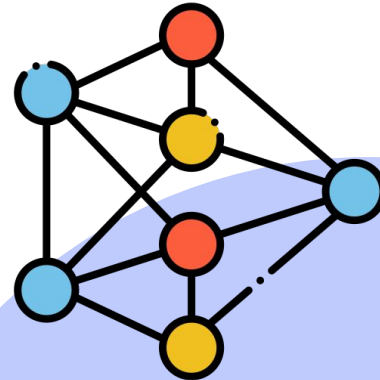
Familiarization

Step-by-step tutorial

Hands-on

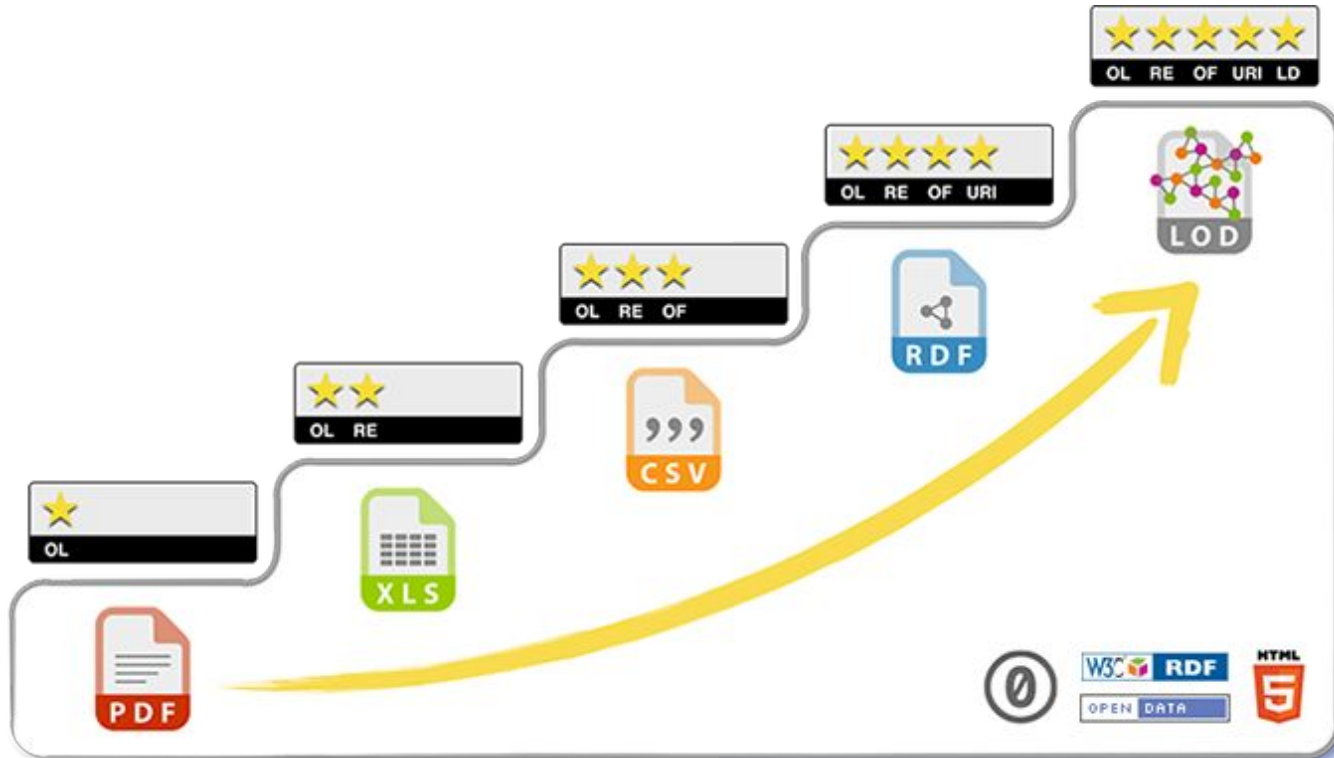
How to define (Linked) Open Data?

They are data...



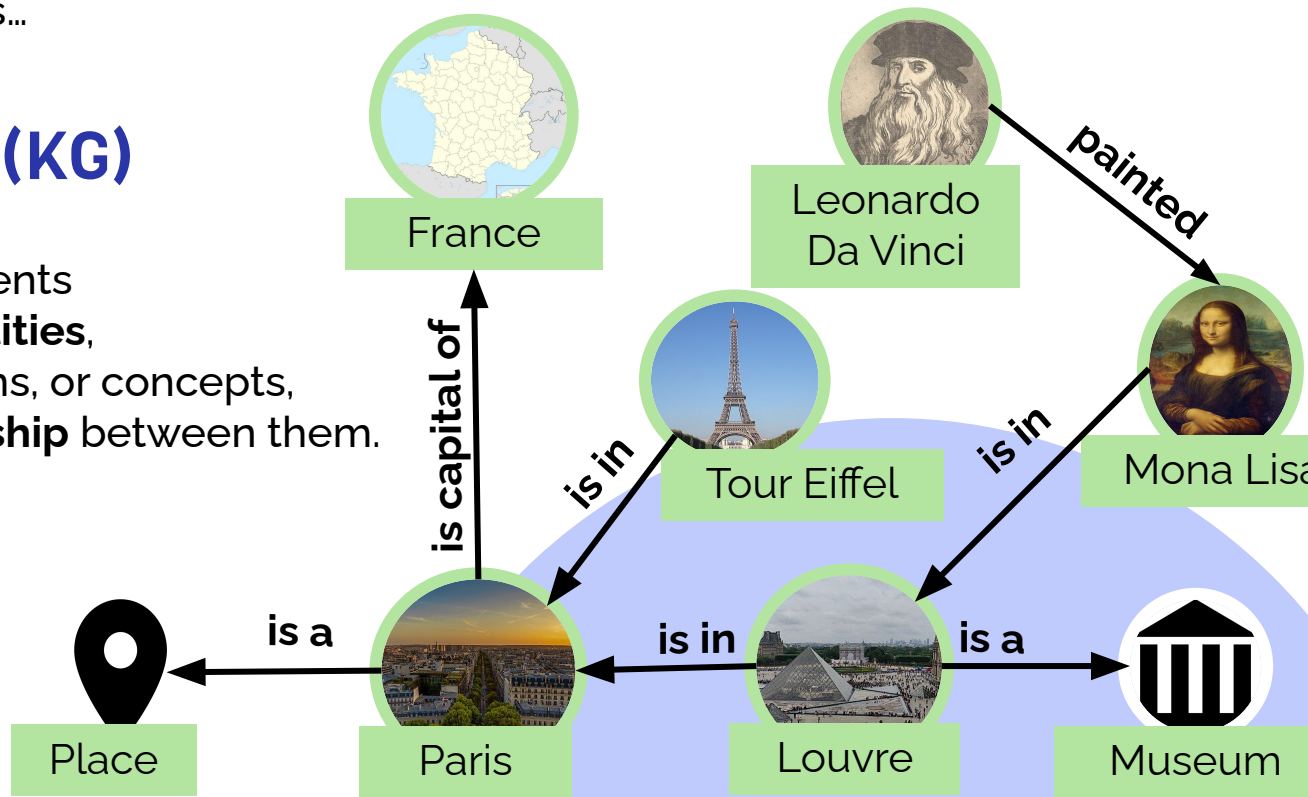
... published with an OPEN license... ... interlinked with other available data.

How to define (Linked) Open Data?



Knowledge Graph (KG)

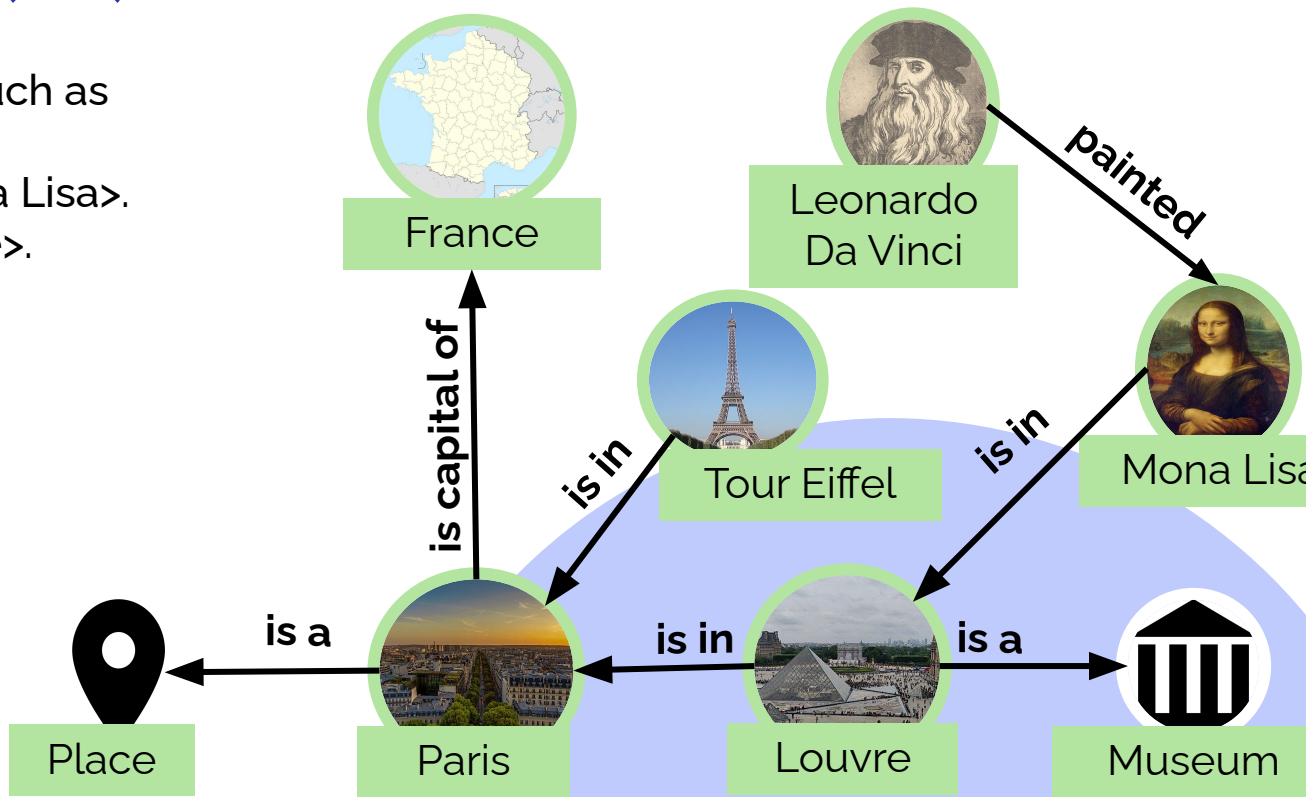
A knowledge graph represents a network of real-world **entities**, i.e. objects, events, situations, or concepts, and illustrates the **relationship** between them.



Knowledge Graph (KG)

KGs are based on triples, such as

<Da Vinci> <Painted> <Mona Lisa>.
<Mona Lisa> <is in> <Louvre>.
[...]



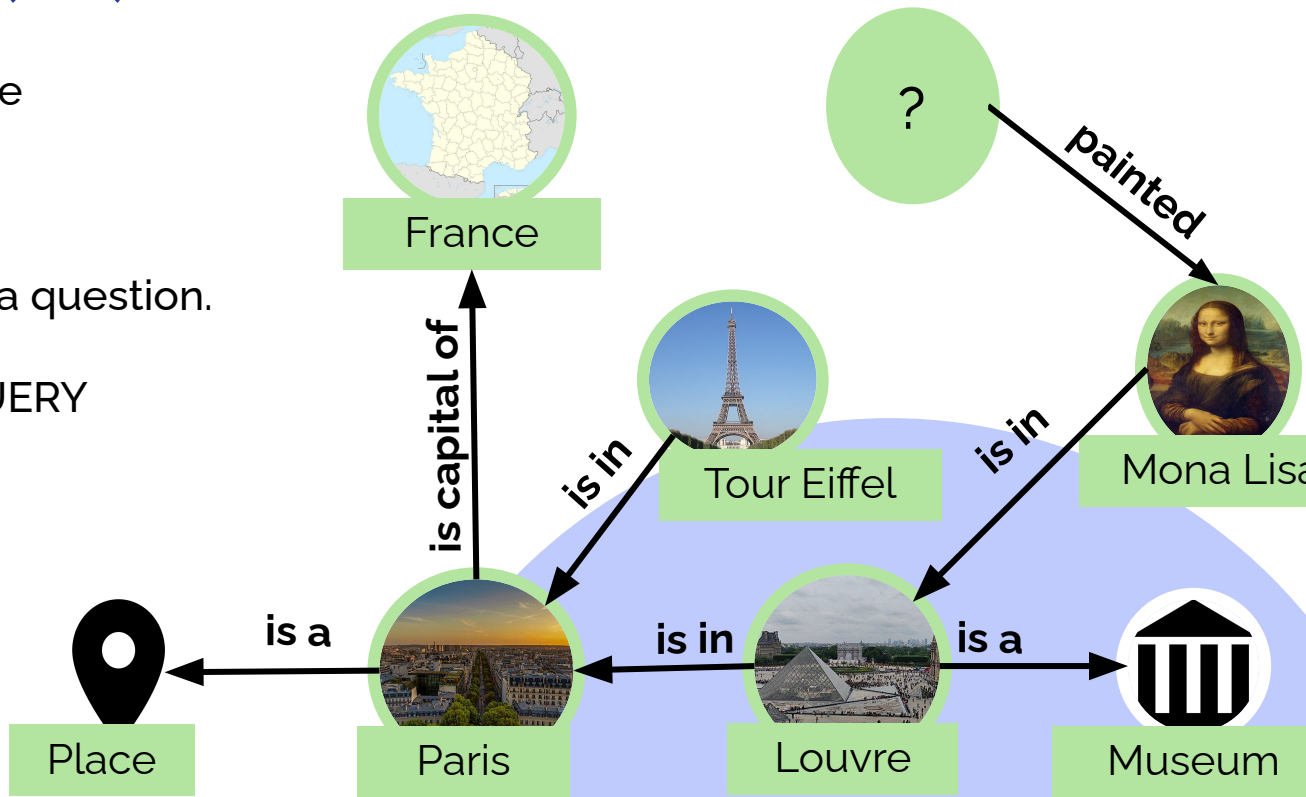
Knowledge Graph (KG)

Suppose we want to retrieve the painter of Mona Lisa.

Objective:

Retrieve data that answers a question.

INTERROGATE = make a QUERY



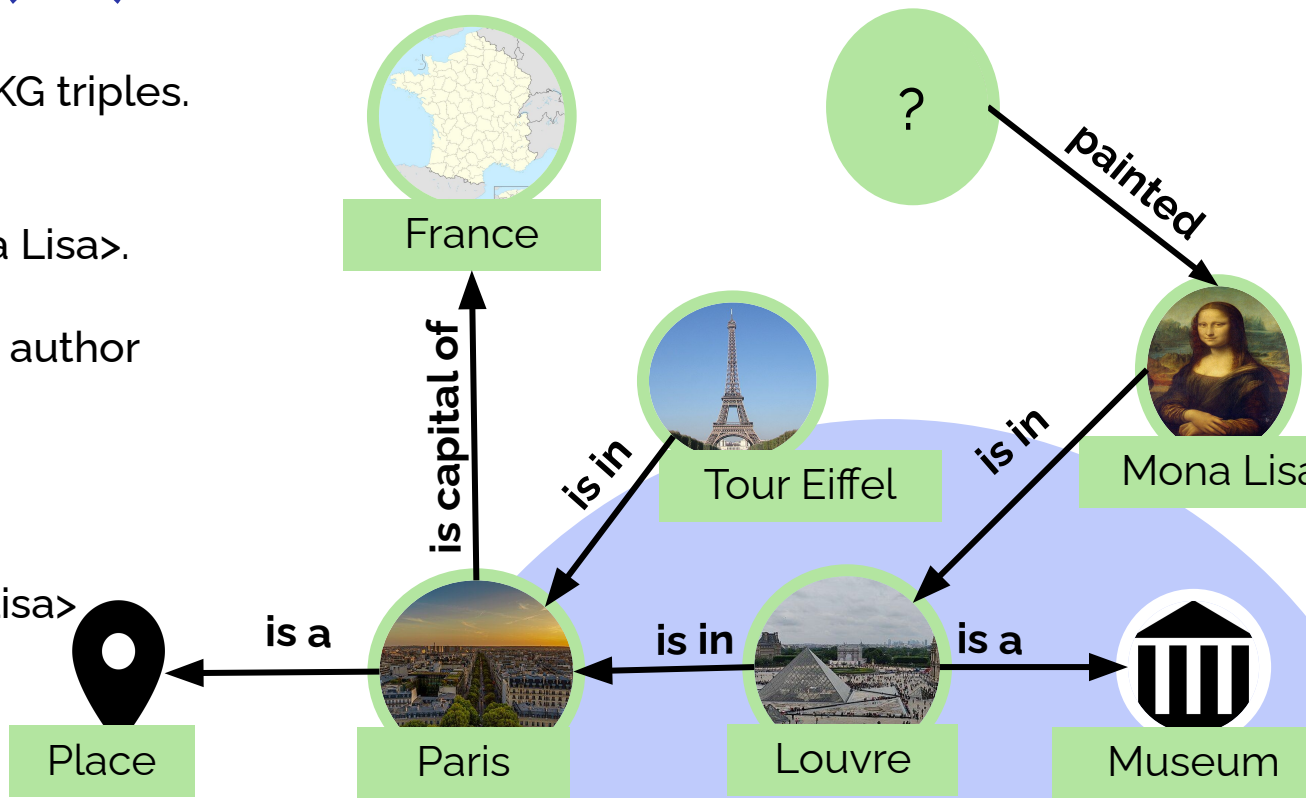
Knowledge Graph (KG)

SPARQL queries resemble KG triples.

If in the KG we have that
<Da Vinci> <Painted> <Mona Lisa>.

If we wanted to retrieve the author
of the Mona Lisa, we must
perform the query:


```
SELECT ?painter WHERE{  
  ?painter <Painted> <Mona Lisa>  
}
```





Using SPARQL is
challenging!

Use a block based
programming environment
to create queries to pose
questions over Knowledge
Graphs.



Knowledge Graph (KG) - Warning!

Pay attention to **QUESTION MARKS** to name variable the query has to retrieve.

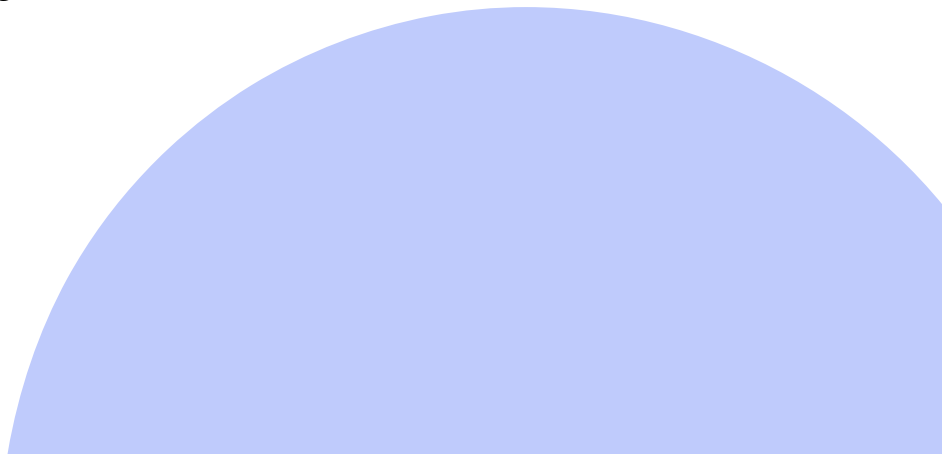
```
SELECT ?painter WHERE{  
  ?painter <painting> <Mona Lisa>  
}
```

Knowledge Graph (KG) - Warning!

URIs not words.

The entities (i.e. the nodes)
and predicates (i.e. relations)
are referred to via URIs.

So, before making a query, we need to retrieve
via SPARQLSnap! the URI of the entities e
predicates we are interested in.



SPARQLSnap!

<https://anonymous.4open.science/w/Snap-10CA/snap.html>

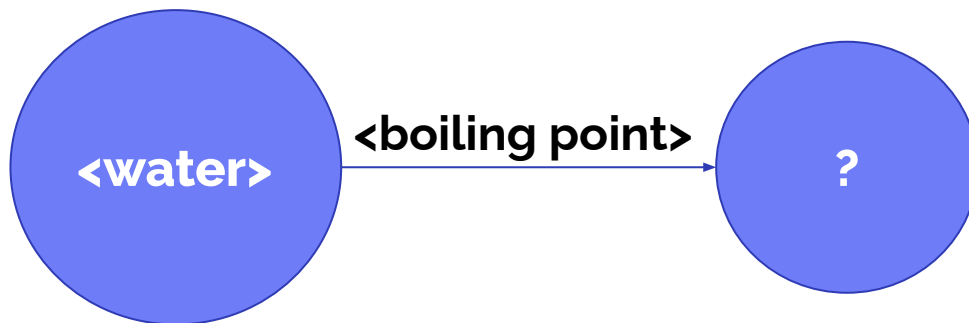
Familiarization

Step-by-step tutorial

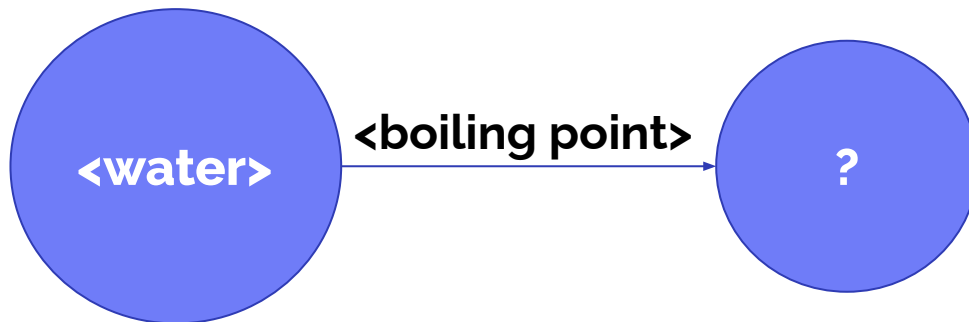
Hands-on

Which is the boiling point of water?

Which is the **boiling point** of **water**?



Which is the **boiling point** of **water**?



Process: Retrieve the URI of **water**, the URI of **boiling point** and then make the query to retrieve the boiling point value.

KGQueries tab > create a block > Monitor > SEARCH FOR A NODE/ARC RELATIVE TO > + variable > add blocks for entity/property search. WARNING: Be careful to surround the query with a ring (right click on the "search [...]" block and click on "insert into a ring")



Which is the boiling point of water?

The screenshot shows the SPARQLSnap! interface with a query script and its results.

Query Script:

- when clicked
- set **water** to look for a node with label **water**
- set **boiling_temperature** to look for an edge with label **boiling point**
- run
 - execute query block:
 - select distinct **?item**
 - from **WikiData**
 - where
 - subject: **water**
 - predicate: **boiling_temperature** object: **?item**
 - order by
 - limit

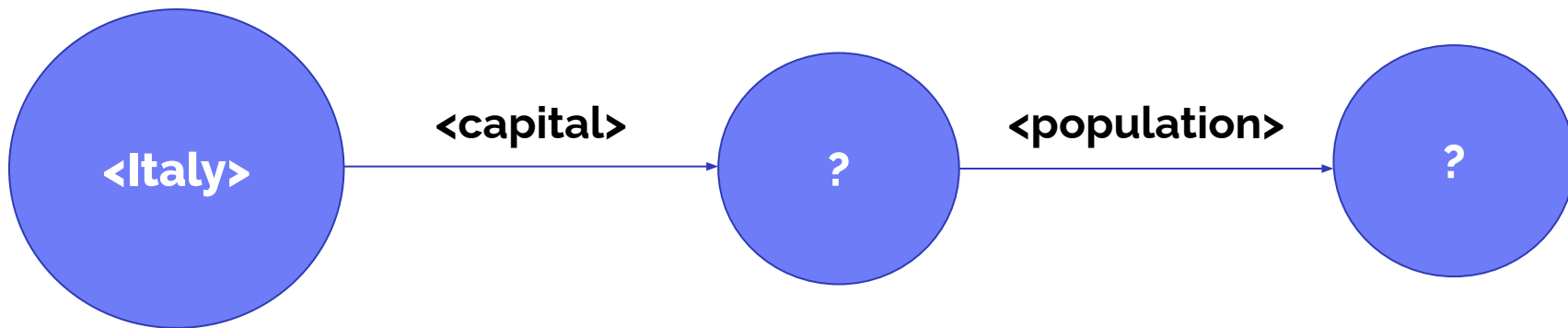
Table view:

	item
1	99.9839

OK

What is the population of the capital of Italy?

What is the population of the capital of Italy?



What is the population of the capital of Italy?

The screenshot shows the SPARQLSnap! interface. The main workspace contains a query block with the following code:

```
select distinct ?population  
from WikiData  
where  
  subject: Italy  
  predicate: capital object: ?capital  
  subject: ?capital  
  predicate: population object: ?population  
order by  
limit
```

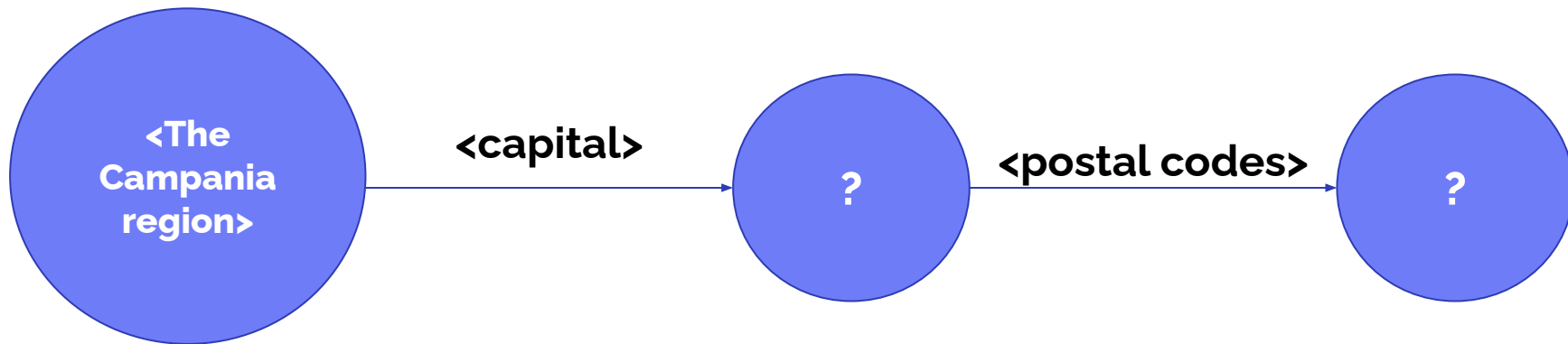
To the right of the query block is a 'Table view' window displaying the results:

	population
1	2748109

The 'Table view' window also includes an 'OK' button and a close icon.

Which are the postal codes of the capital of the Campania region?

Which are the postal codes of the capital of the Campania region?



Which are the postal codes of the capital of the Campania region?

The image shows a SPARQLSnap! blockly interface. It starts with a 'when clicked' block, followed by three 'set' blocks: 'set the_Campania_region to look for a node with label Campania', 'set capital to look for an edge with label capital', and 'set postal_codes to look for an edge with label postal_code'. These are followed by a 'run execute query block:' which contains a SPARQL query. The query is: 'select distinct ?codes from WikiData where subject: the_Campania_region predicate: capital object: ?capital subject: ?capital predicate: postal_codes object: ?codes order by ?codes ASC limit 10'. The 'limit' block has a slider set to 10.

```
select distinct ?codes
from WikiData
where
  subject: the_Campania_region
  predicate: capital object: ?capital
  subject: ?capital
  predicate: postal_codes object: ?codes
order by ?codes ASC
limit 10
```

25	codes
1	80121
2	80122
3	80123
4	80124
5	80125
6	80126
7	80127
8	80128
9	80129
10	80131
11	80132
12	80133
13	80134
14	80135
15	80136
16	80137
17	80138
18	80139
19	80141
20	80142
21	80143
22	80144
23	80145
24	80146
25	80147

It's your turn!

<https://anonymous.4open.science/w/Snap-10CA/snap.html>

Familiarization

Step-by-step tutorial

Hands-on

Suggested tasks:

- 1- Which is the construction date of the Royal Palace of Caserta?
- 2- What is the date of birth of the mayor of Naples?
- 3- How many brothers does the mayor of Naples have?

Thank you!

- Autore:
 - *Details are omitted due to anonymity requirement.*
- Icons created by Freepik - Flaticon
- License: This Work is licensed with Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)
<https://creativecommons.org/licenses/by-sa/4.0>

