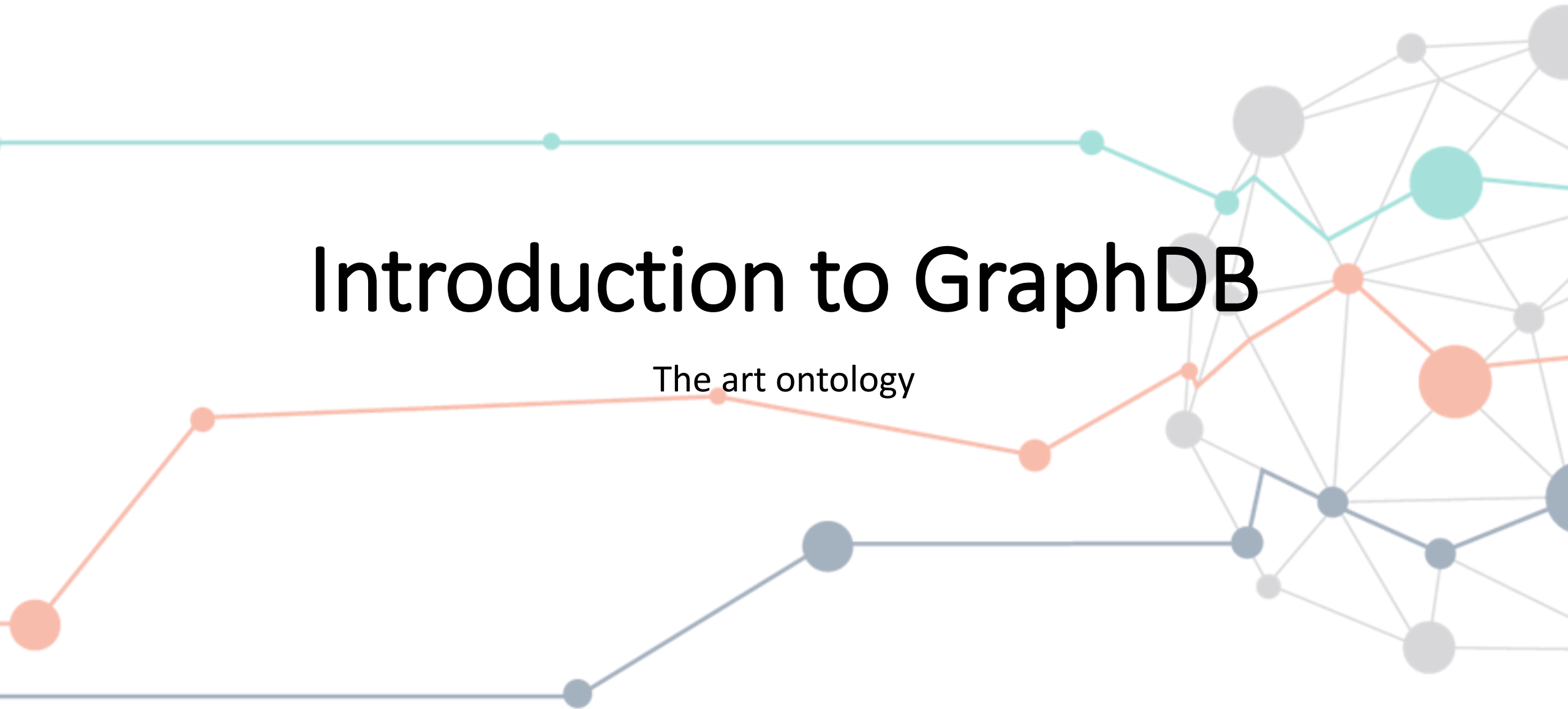


# Introduction to GraphDB

The art ontology



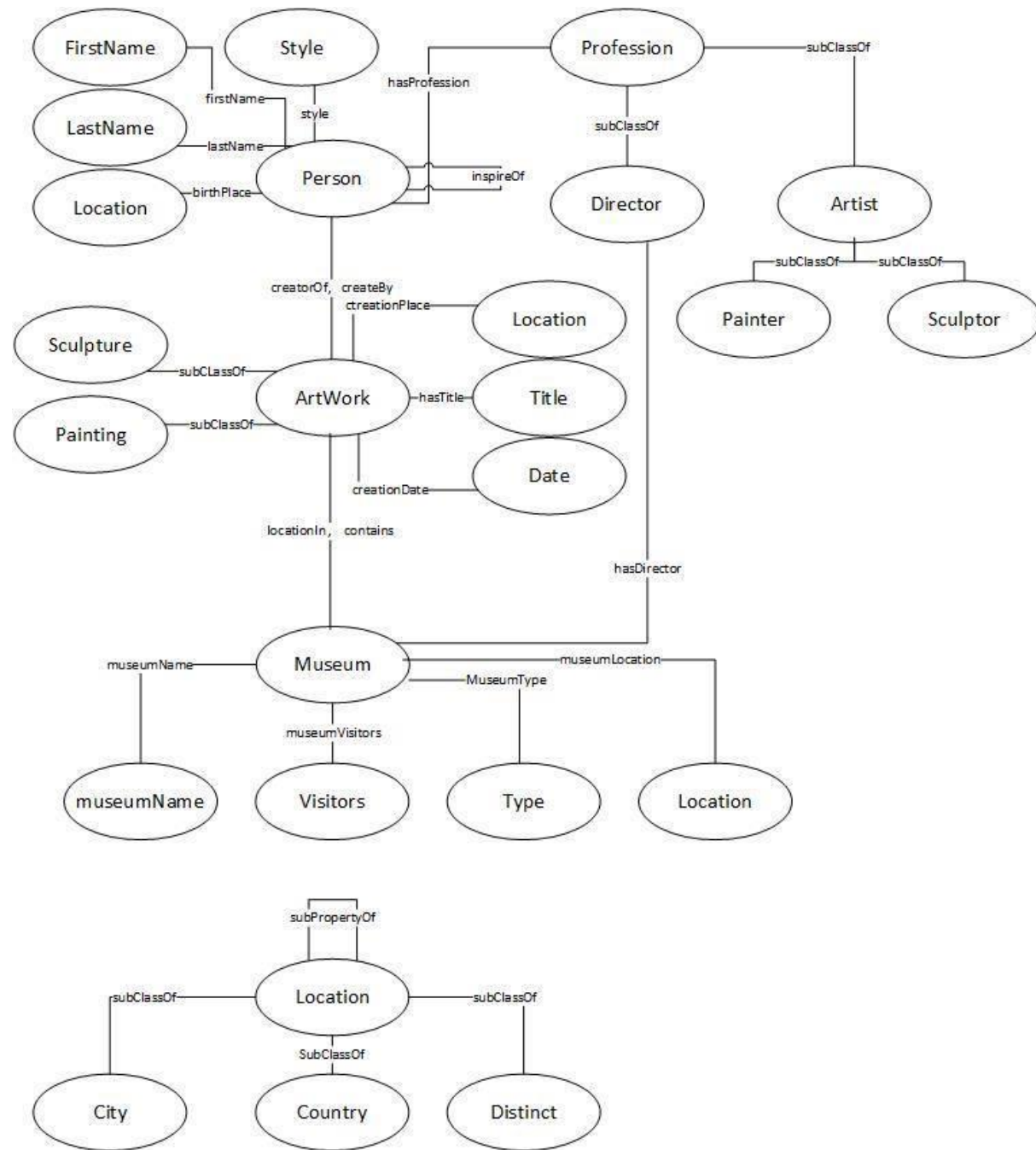
# Information about the KB

The knowledge base contains information about:

- artists,
- their works (paintings and sculptures),
- museums.

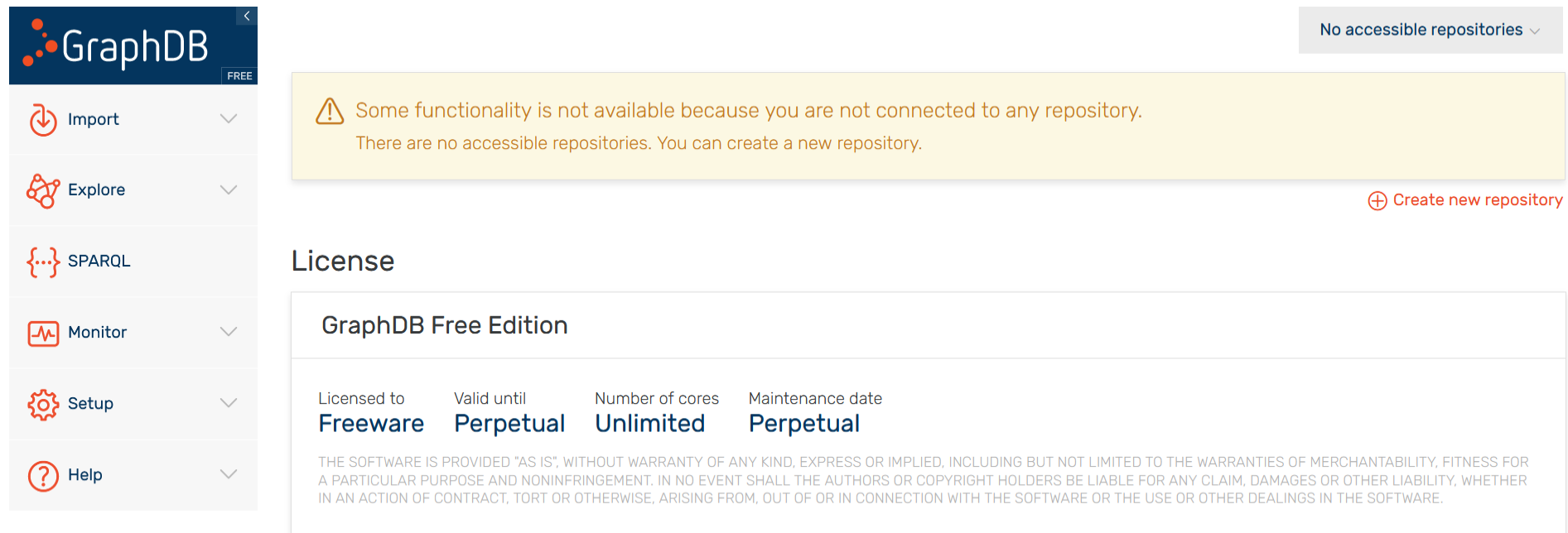
There is also additional information about:

- the location of each museum,
- when and where the individual works were created,
- the names and birthplaces of the artists.



## 2. Creating a repository in GraphDB

For the realization of this project, we use GraphDB Free. After installation and launch of the product, its graphical view opens in the default browser.



GraphDB FREE

No accessible repositories ▾

⚠ Some functionality is not available because you are not connected to any repository.  
There are no accessible repositories. You can create a new repository.

[+ Create new repository](#)

### License

#### GraphDB Free Edition

Licensed to	Valid until	Number of cores	Maintenance date
<b>Freeware</b>	<b>Perpetual</b>	<b>Unlimited</b>	<b>Perpetual</b>

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

We select "Create new repository". We choose the name of the repository, select the reasoning rules to be OWL-Horst (Optimized), and click the “Create” button in the end.

GraphDB

FREE

Import

Explore

SPARQL

Monitor

Setup

Help

No accessible repositories

## Create Repository

### Repository properties

Repository ID\*

art-ontology-repository

Repository title

Type

GRAPHDB-FREE

Storage folder

storage

Ruleset

RDFS-Plus (Optimized)

Upload custom ruleset

☒ Disable owl:sameAs

Base URL

http://example.org/owlim#

Entity index size

10000000

☒ Use predicate indices

☒ Cache literal language tags

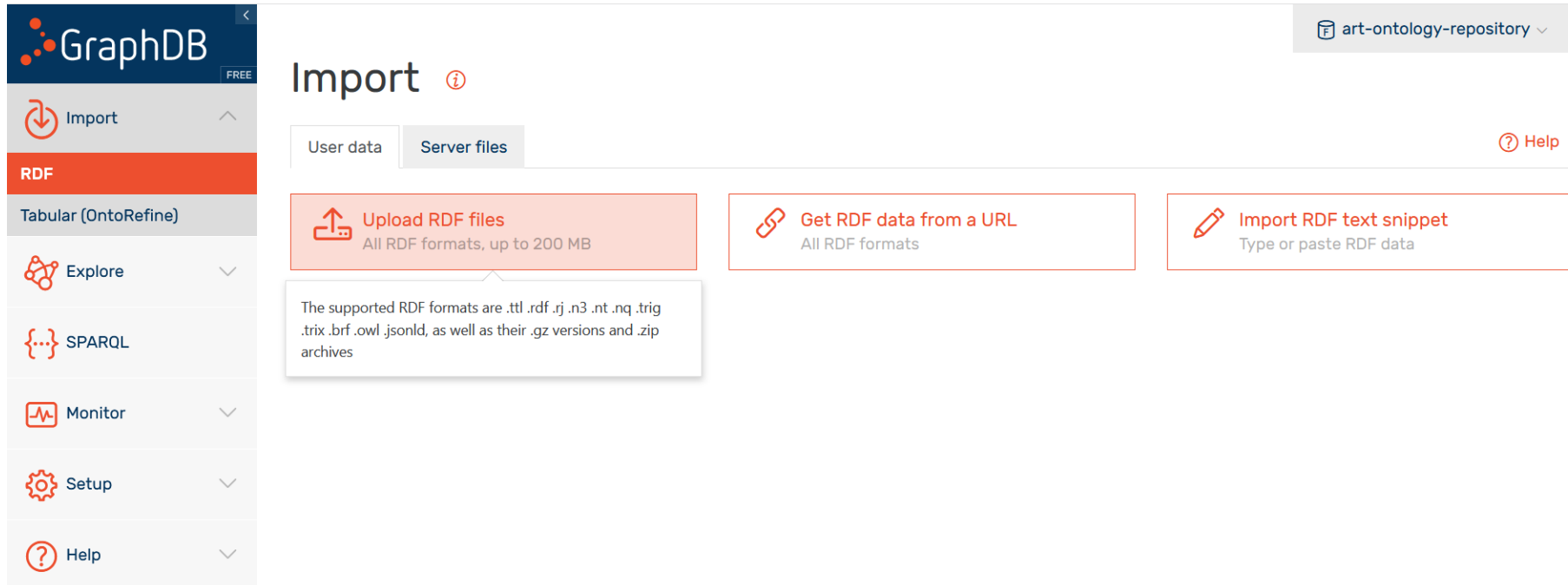
☐ Use context index

☒ Enable literal index

☐ Check for inconsistencies

☐ Throw exception on query time-out

To upload the required file to the repository, we select the menu Import -> RDF -> Upload RDF files:



We select to upload the required file and click the “Import” button for the file. Select the Base URI to be <http://www.art-ontology.com/fmi#>, and we again select “Import”.

## Import settings



Base IRI ⓘ

http://www.art-ontology.com/fmi#

Target graphs ⓘ

☒ From data ☐ The default graph ☐ Named graph

http://example.com/graph...

☐ Enable replacement of existing data

Show advanced settings ▾

Restore defaults

Cancel

Import

If everything is all right, we will see the following:



📎 art.ttl

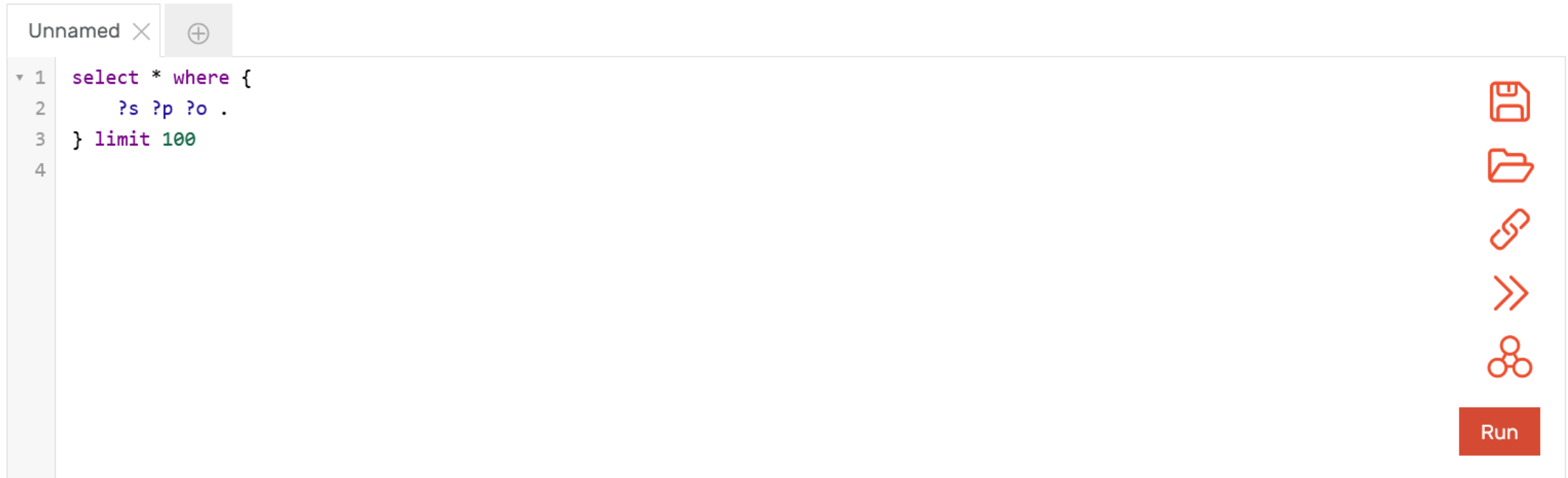
✕ ⓘ ✓ Imported successfully in less than a second.



🔄 Import

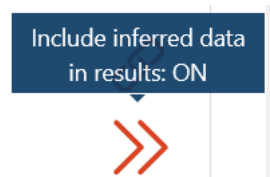
# 3. Queries

In GraphDB we write queries after we select “SPARQL” form the menu.

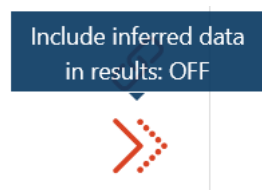




The request is written in the specified window. It is possible to specify whether reasoning should be used in its implementation based on a logical conclusion on the knowledge and data from the ontology or not. If not, the result of the execution of the application will be based only on the knowledge and the data, that are included explicitly in the ontology.



- If two arrows are visible in the right bar of the window, this means that the reasoning option is on.



- if only one arrow is visible, this is an indication that the execution of the query does not include the results of reasoning.

## 3.1. Queries without reasoning

- Only information that is explicit in the ontology is searched.
- Example: All paintings by Italian artists in the High Renaissance style, located in the Louvre:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX art: <http://www.art-ontology.com/fmi#>
select ?person ?artWork where{
    ?person art:hasProfession art:Painter;
        art:style ?style .
    filter( str(?style) = "High Renaissance" ).
    ?person art:birthPlace ?loc.
    ?loc art:locationName ?locname . filter( str(?locname) = "Italy" ).
    ?person art:creatorOf ?artWork.
    ?artWork rdf:type art:Painting;
        art:locationIn ?museum.
    ?museum art:museumName ?mname. filter( str(?mname) = "Louvre" ).
}
```

The artists and their paintings are shown as a result. The requirements are: the style is High Renaissance, the person has a profession painter and is born in Italy; the artwork he created is created a is a painting, and that painting is in the Louvre.

In the ontology there is only one pair that meets the requirements, and it is art: LeonardoDaVinci - art: MonaLisa.

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX art: <http://www.art-ontology.com/fmi#>
3 select ?person ?artWork where{
4   ?person art:hasProfession art:Painter;
5   art:style ?style .
6   filter( str(?style) = "High Renaissance" ).
7   ?person art:birthPlace ?loc.
8   ?loc art:locationName ?locname . filter( str(?locname) = "Italy" ).
9   ?person art:creatorOf ?artWork.
10  ?artWork rdf:type art:Painting;
11  art:locationIn ?museum.
12  ?museum art:museumName ?mname. filter( str(?mname) = "Louvre" ).
13 }
```

Table

Raw Response

Pivot Table

Google Chart

Download as

Filter query results

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

	person	artWork
1	art:LeonardoDaVinci	art:MonaLisa

## 2. Queries with reasoning

- We search for information, gathered with reasoning on the ontology.
- Example: All paintings, created by Leonardo Da Vinci in Italy:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX art: <http://www.art-ontology.com/fmi#>
select ?artWork where{
    ?person art:firstName ?fname;
           art:lastName ?sname; art:hasProfession art:Painter;
           art:creatorOf ?artWork.
    filter( str(?fname) = "Leonardo" ). filter( str(?sname) = "Da
    Vinci").
    ?artWork rdf:type art:Painting;
           art:creationPlace ?loc.
    ?loc art:locationName ?crlocname; art:subRegionOf art:Italy.
}
```

When the request is executed without reasoning, the result is the picture *art:LadyWithAnErmine*, which is directly set to be created in Milan (direct *subRegionOf* Italy).

If reasoning is enabled, two results are obtained: *LadyWithAnErmine* and *TheLastSupper*. The second is derived from reasoning, as according to the data in the database, *TheLastSupper* is painted in SanVittore, which is a *subRegionOf* Milan, which is a *subRegionOf* Italy. In the ontology, the *subRegionOf* property is defined as **transitive**.

Result without reasoning:

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX art: <http://www.art-ontology.com/fmi#>
3 select ?artWork where{
4   ?person art:firstName ?fname;
5   art:lastName ?sname;  art:hasProfession art:Painter;  art:creatorOf ?artWork.
6   filter( str(?fname) = "Leonardo" ).  filter( str(?sname) = "Da Vinci").
7   ?artWork rdf:type art:Painting;
8   art:creationPlace ?loc.
9   ?loc art:locationName ?crlocname;  art:subRegionOf art:Italy.
10 }
11
```



Run

- Table
- Raw Response
- Pivot Table
- Google Chart

Download as

Filter query results

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

	artWork
1	art:LadyWithAnErmine

# Result with reasoning:

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX art: <http://www.art-ontology.com/fmi#>
3 select ?artWork where{
4   ?person art:firstName ?fname;
5   art:lastName ?sname; art:hasProfession art:Painter; art:creatorOf ?artWork.
6   filter( str(?fname) = "Leonardo" ). filter( str(?sname) = "Da Vinci").
7   ?artWork rdf:type art:Painting;
8   art:creationPlace ?loc.
9   ?loc art:locationName ?crlocname; art:subRegionOf art:Italy.
10 }
11
```



Run

Table Raw Response Pivot Table Google Chart

Download as

Filter query results

Showing results from 1 to 2 of 2. Query took 0.1s, moments ago.

	artWork
1	art:LadyWithAnErmine
2	art:TheLastSupper

keyboard shortcuts

# One more example

- All sculptures by Michelangelo

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX art: <http://www.art-ontology.com/fmi#>
select ?artWork where{
    ?artWork rdf:type art:Sculpture; art:createBy
    ?sculptor.
    ?sculptor art:firstName ?fname; filter( str(?fname) =
    "Michelangelo" ).
}
```

- The result is a list of all the sculptures that were created by a sculptor named Michelangelo. Without reasoning the list is empty, as the link between the sculpture and its creator is set with the inverse property *creatorOf* of *createBy*. With reasoning included, the result of the execution of the request is art: Pieta.



Result without reasoning:

▼ 1

2

▼ 3

4

5

6

7

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX art: <http://www.art-ontology.com/fmi#>
select ?artWork where{
?artWork rdf:type art:Sculpture; art:createBy ?sculptor.
?sculptor art:firstName ?fname; filter( str(?fname) = "Michelangelo" ).
}
```



Run

TableRaw ResponsePivot TableGoogle Chart

Download as ▼

Filter query results

No results. Query took 0.1s, moments ago.

	artWork	
No data available in table		

Result with reasoning:

▼ 1

2

▼ 3

4

5

6

7

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX art: <http://www.art-ontology.com/fmi#>
select ?artWork where{
?artWork rdf:type art:Sculpture; art:createBy ?sculptor.
?sculptor art:firstName ?fname; filter( str(?fname) = "Michelangelo" ).
}
```



Run

TableRaw ResponsePivot TableGoogle Chart

Download as ▼

Filter query results

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

	artWork	
1	art:Pieta	

Questions?