Introduction to GraphDB

The art ontology

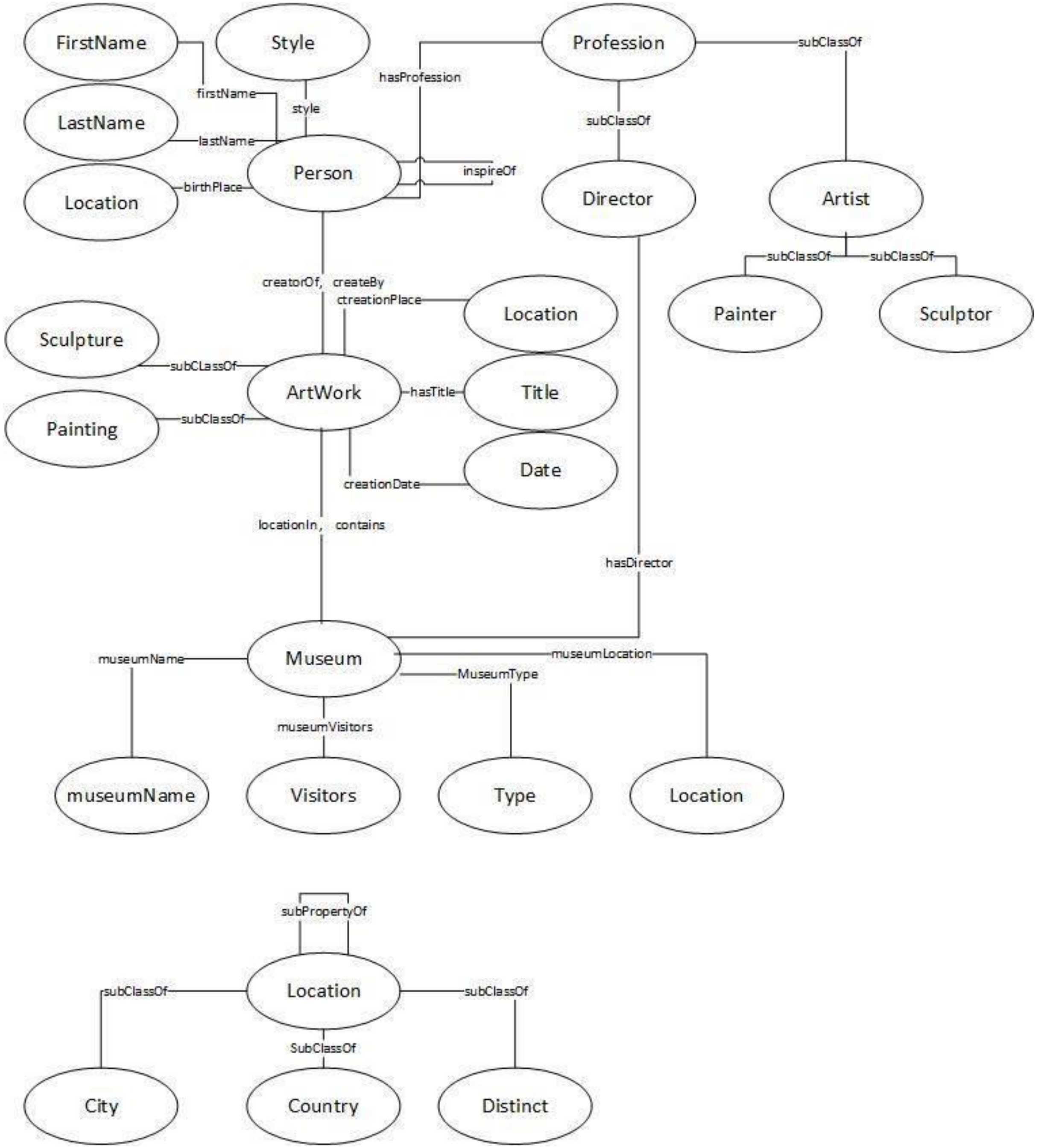
Information about the KB

The knowledge base contains information about:

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

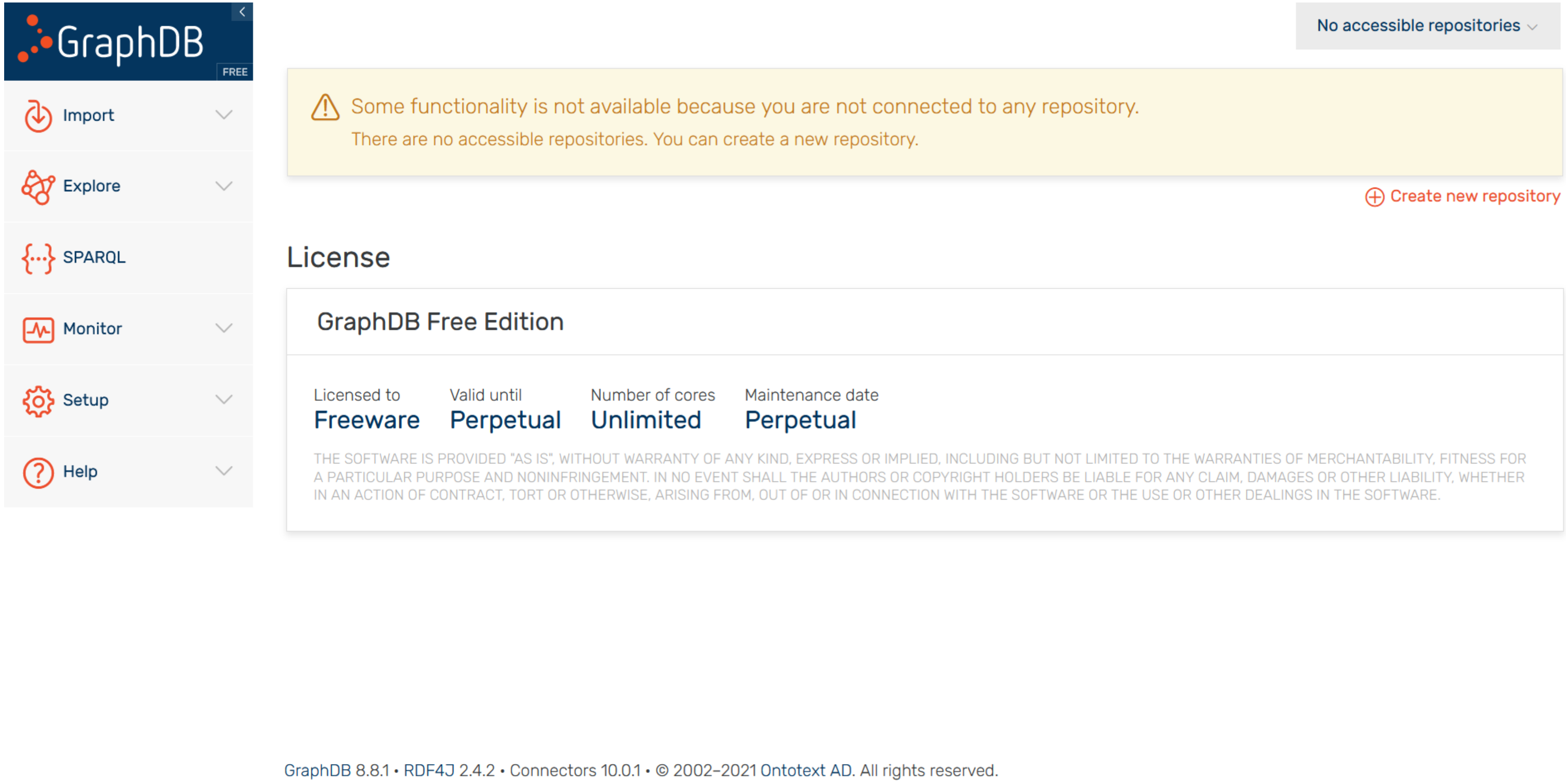
There is also additional information about:

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


FirstName
Style
Profession
subClassOf
hasProfession
firstName
style
LastName
lastName
Person
inspireOf
Director
Artist
Location
birthPlace
subClassOf
subClassOf
subClassOf
creatorOf, createBy
ctreationPlace
Location
Painter
Sculptor
Sculpture
subCLassOf
ArtWork
hasTitle
Title
Painting
subClassOf
Date
creationDate
locationIn, contains
hasDirector
Museum
museumName
museumLocation
MuseumType
museumVisitors
museumName
Visitors
Type
Location
subPropertyOf
Location
subClassOf
subClassOf
SubClassOf
City
Country
Distinct

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.



Some functionality is not available because you are not connected to any repository.

There are no accessible repositories. You can create a new repository.

License

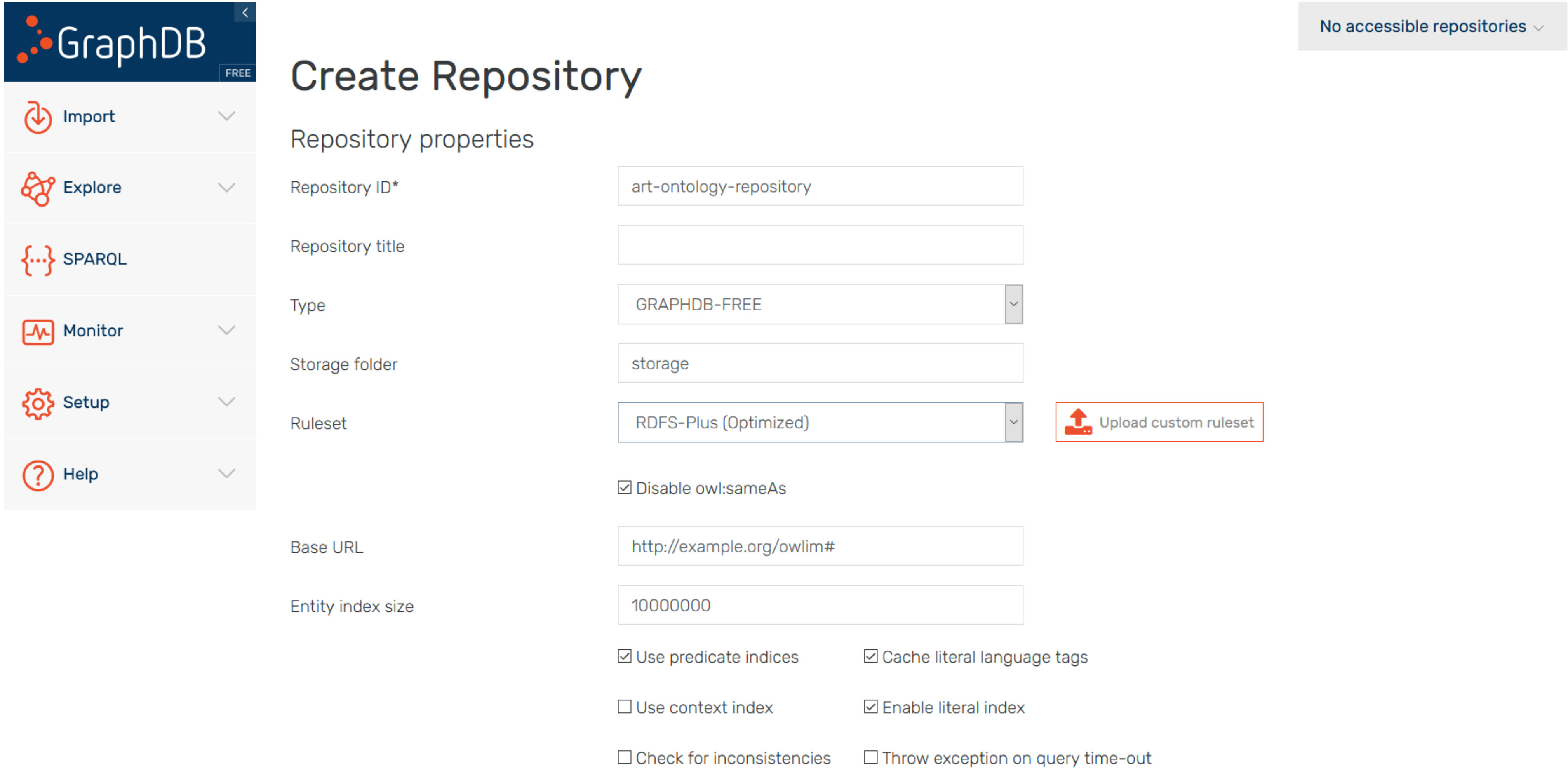
GraphDB Free Edition

Licensed to	Valid until	Number of cores	Maintenance date
Freeware	Perpetual	Unlimited	Perpetual

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.

GraphDB 8.8.1 • RDF4J 2.4.2 • Connectors 10.0.1 • © 2002–2021 Ontotext AD. All rights reserved.

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.



Create Repository

Repository properties

Repository ID*	art-ontology-repository
Repository title	
Type	GRAPHDB-FREE
Storage folder	storage
Ruleset	RDFS-Plus (Optimized)

☑ Disable owl:sameAs

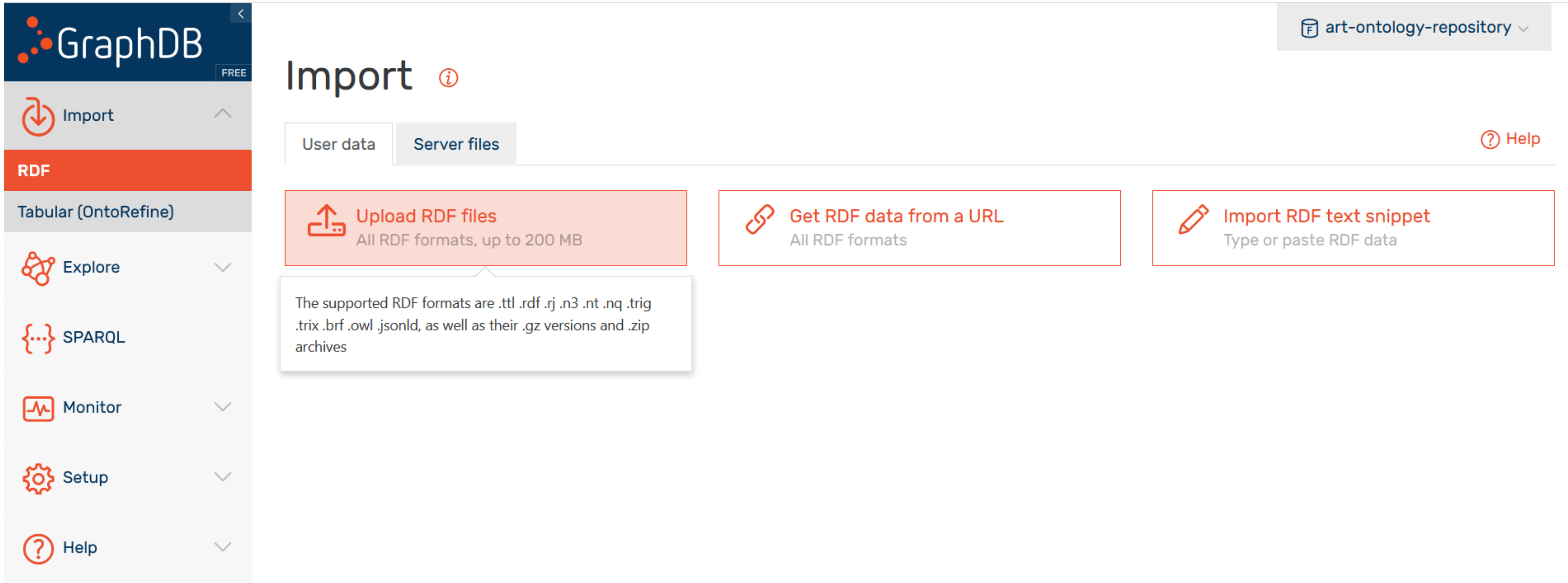
Base URL	http://example.org/owlim#
Entity index size	10000000

☑ Use predicate indices ☑ Cache literal language tags

☐ Use context index ☑ Enable literal index

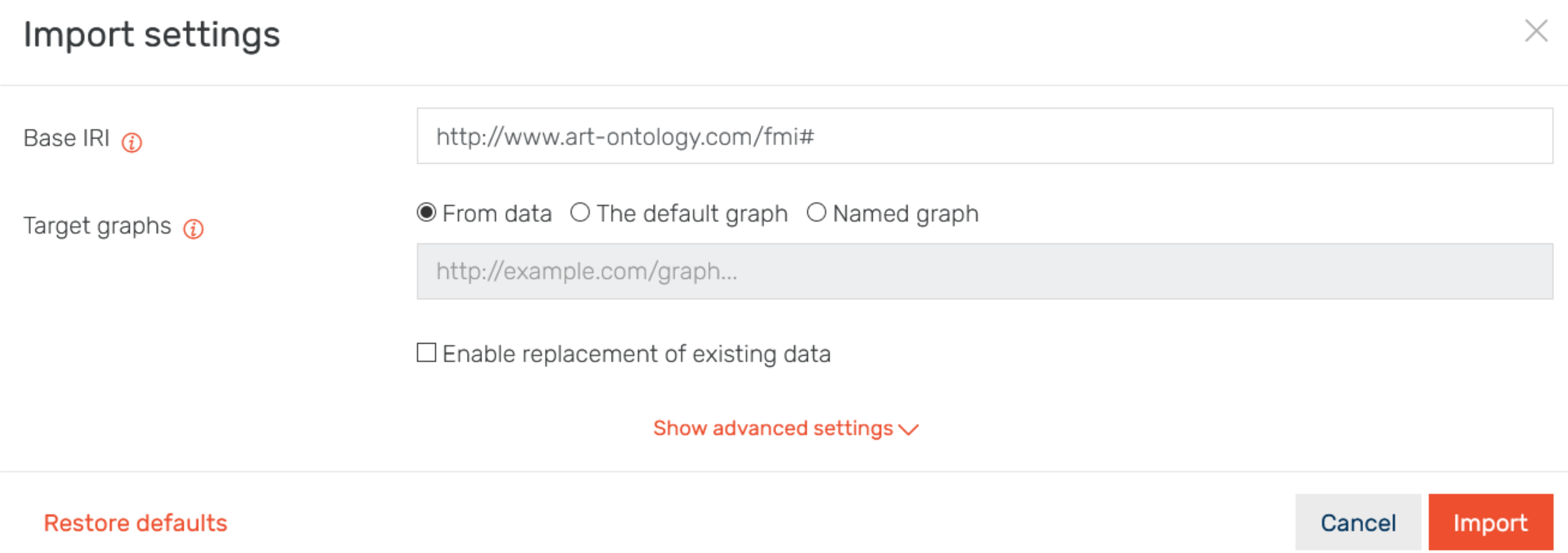
☐ Check for inconsistencies ☐ Throw exception on query time-out

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



The supported RDF formats are .ttl .rdf .rj .n3 .nt .nq .trig .trix .brf .owl .jsonld, as well as their .gz versions and .zip archives

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

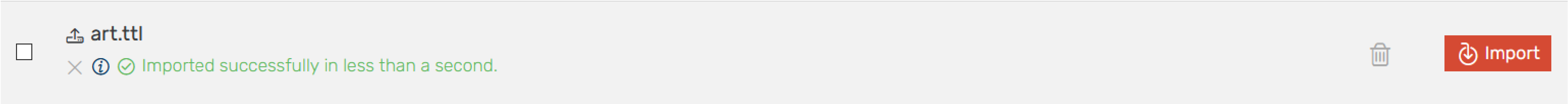
Base IRI: http://www.art-ontology.com/fmi#

Target graphs: ● From data ○ The default graph ○ Named graph

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

□ Enable replacement of existing data

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

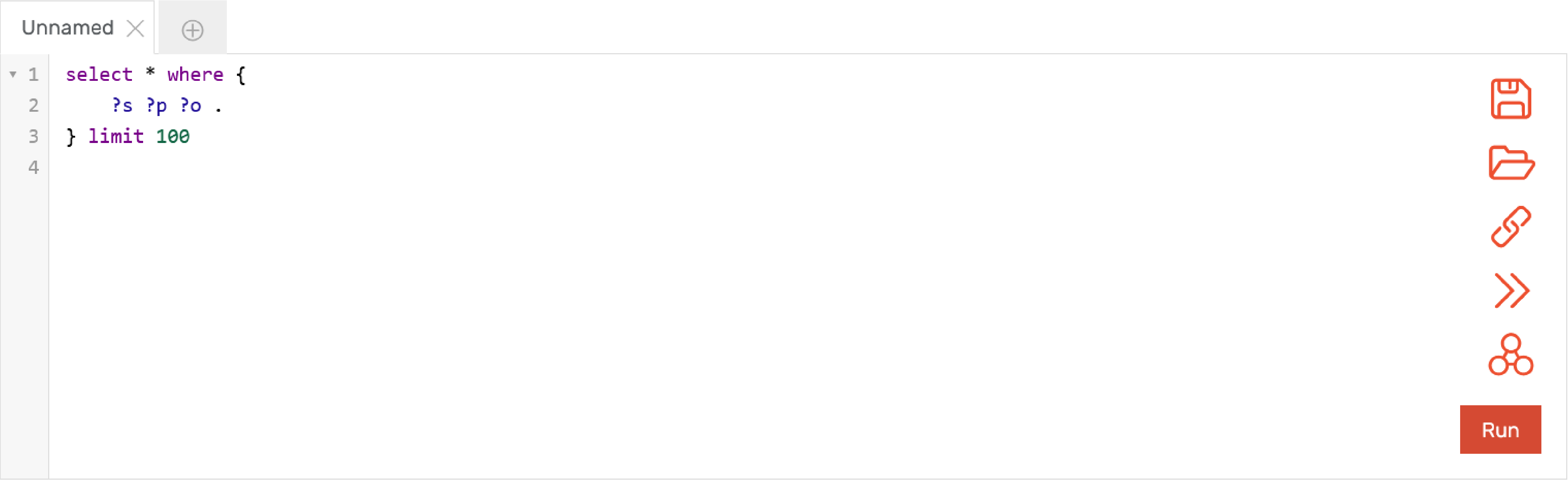


□ art.ttl

Imported successfully in less than a second.

3. Queries

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



```
select * where {
    ?s ?p ?o .
} limit 100
```

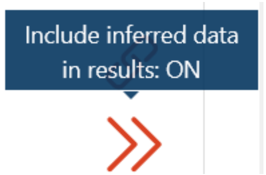
Basic SPARQL

We will discuss SPARQL in detail in the next exercises. For now, you should know that:

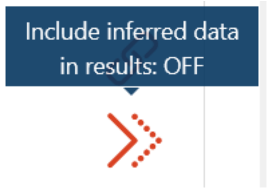
- SELECT clauses are used for querying specific variables and expressions
- Statements inside the SELECT clause match the *subject-property-object* pattern.
- Variables start with *?.*
- Semicolon *(;)* means repetition of the subject, and *c*omma (,) means repetition of both the subject and the property.

Reasoning

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.

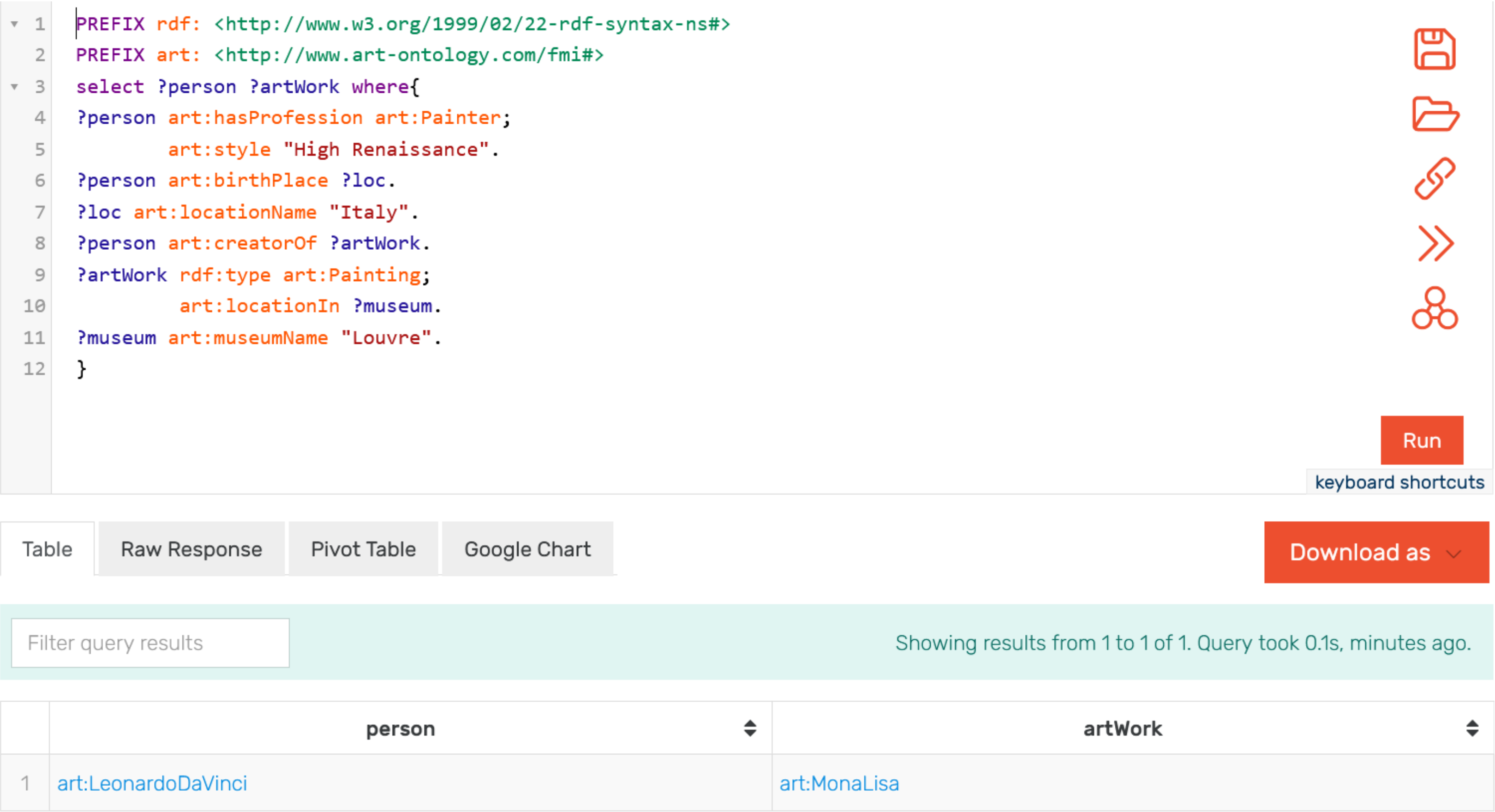
3.1. Queries without reasoning

- 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 "High Renaissance".
?person art:birthPlace ?loc.
?loc art:locationName "Italy".
?person art:creatorOf ?artWork.
?artWork rdf:type art:Painting;
         art:locationIn ?museum.
?museum art:museumName "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 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.



```
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 "High Renaissance".
?person art:birthPlace ?loc.
?loc art:locationName "Italy".
?person art:creatorOf ?artWork.
?artWork rdf:type art:Painting;
         art:locationIn ?museum.
?museum art:museumName "Louvre".
}
```

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

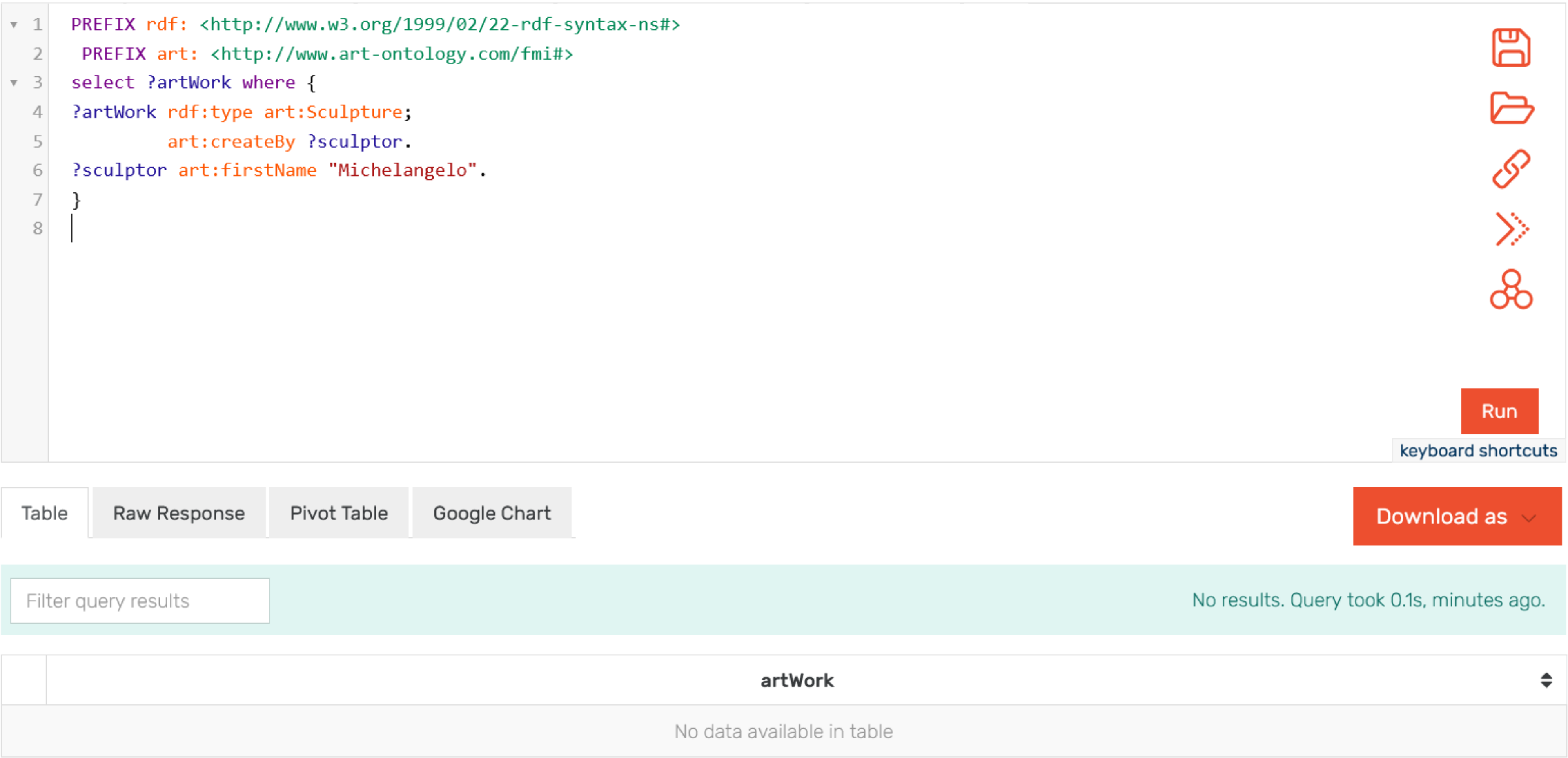
	person	artWork
1	art:LeonardoDaVinci	art:MonaLisa

2. Queries with reasoning

- 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 "Michelangelo".
}
```

Result without reasoning:

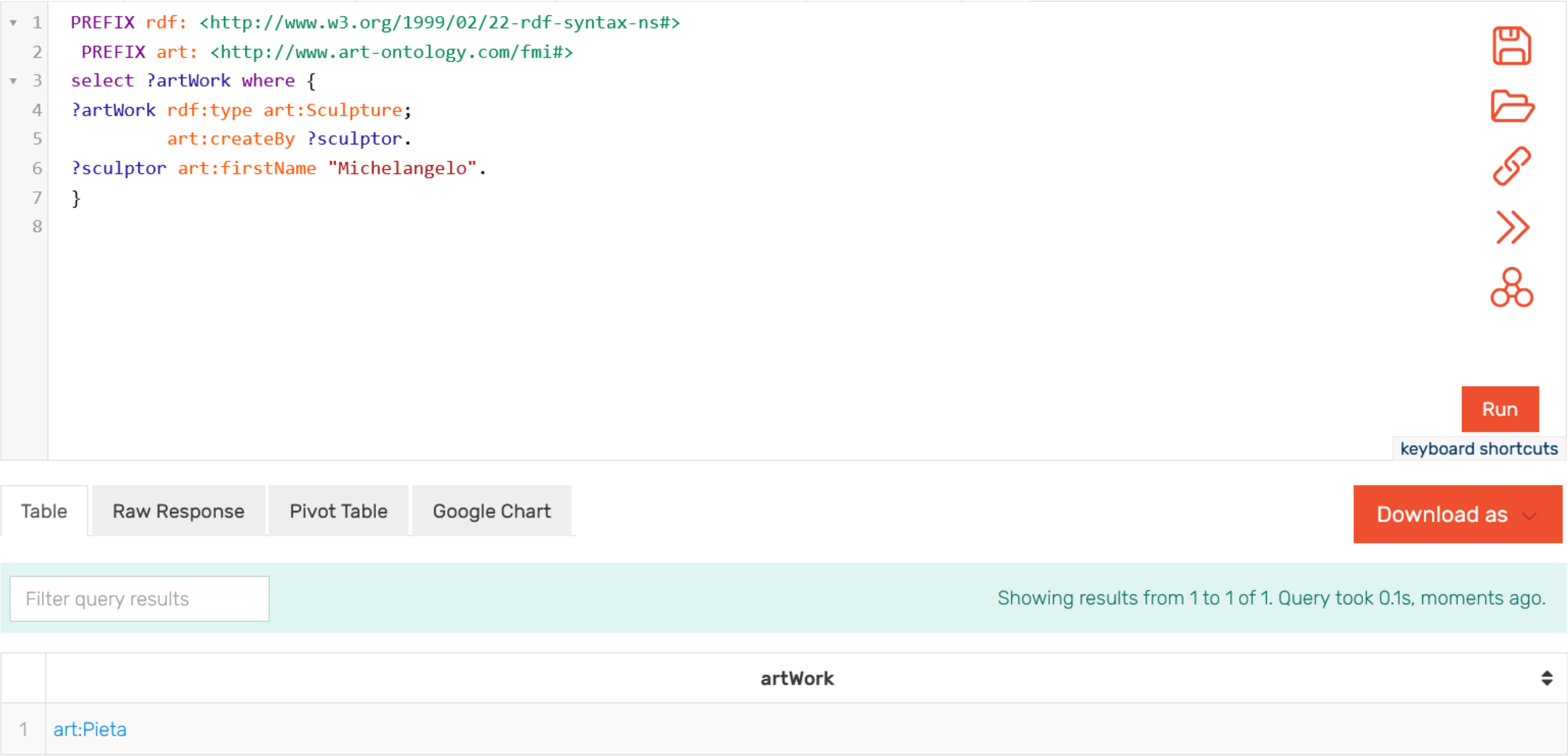


```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
 PREFIX art: <http://www.art-ontology.com/fmi#>
select ?artWork where {
?artWork rdf:type art:Sculpture;
         art:createBy ?sculptor.
?sculptor art:firstName "Michelangelo".
}
```

No results. Query took 0.1s, minutes ago.

artWork
No data available in table

Result with reasoning:



```
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 "Michelangelo".
}
```

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

	artWork
1	art:Pieta

The result is a list of all sculptures that were created by a sculptor named Michelangelo.

When the reasoning us turned off, the list is empty, because in the ontology the relationship between the sculpture and its creator is set with the property *creatorOf* (which is an inverse property of the *createBy* property).

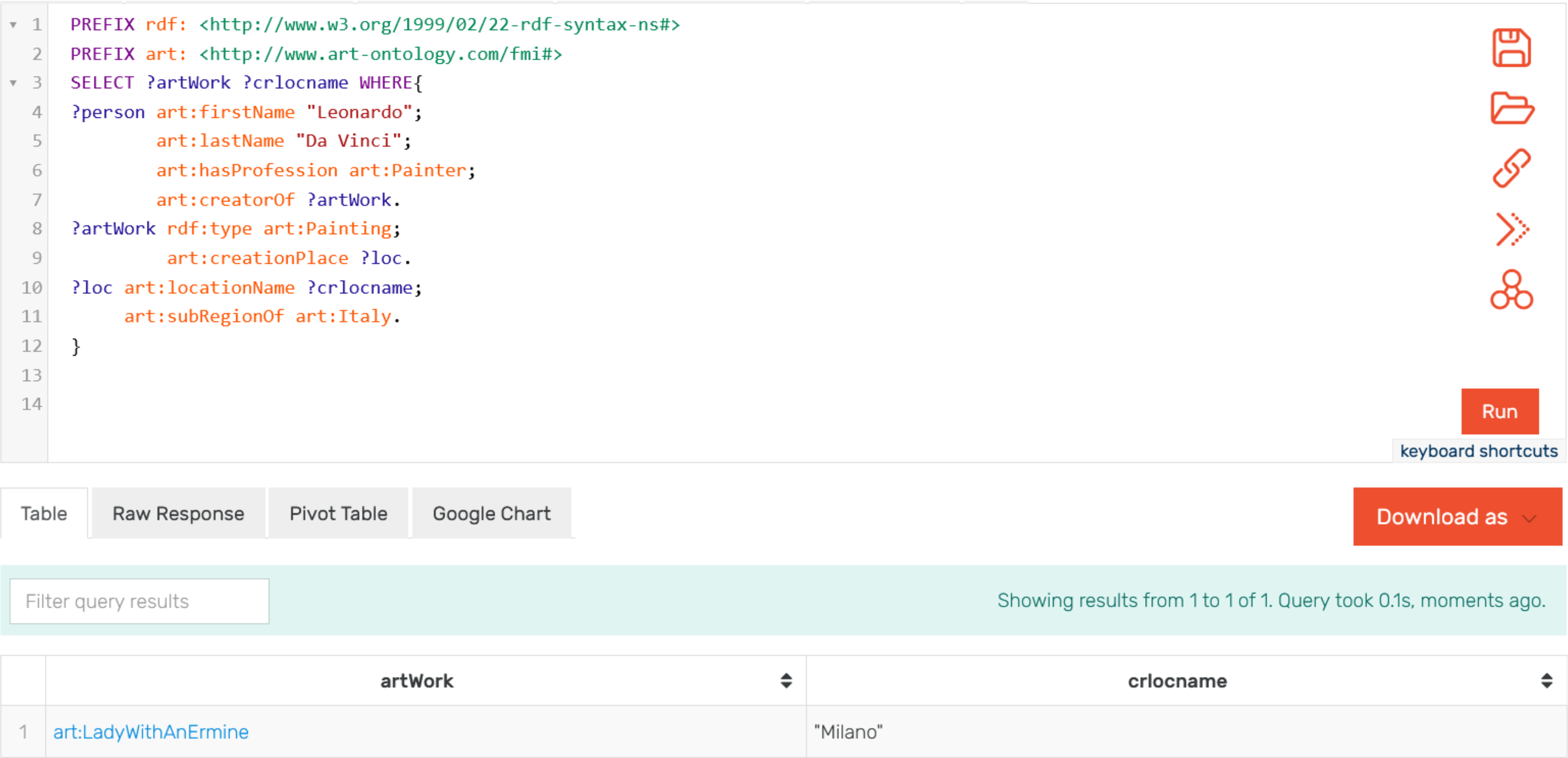
When the reasoning is enabled, the result of executing the query is *art:Pieta*.

One more example

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 ?crlocname WHERE{
?person art:firstName "Leonardo";
        art:lastName "Da Vinci";
        art:hasProfession art:Painter;
        art:creatorOf ?artWork.
?artWork rdf:type art:Painting;
         art:creationPlace ?loc.
?loc art:locationName ?crlocname;
     art:subRegionOf art:Italy.
}
```

Result without reasoning:

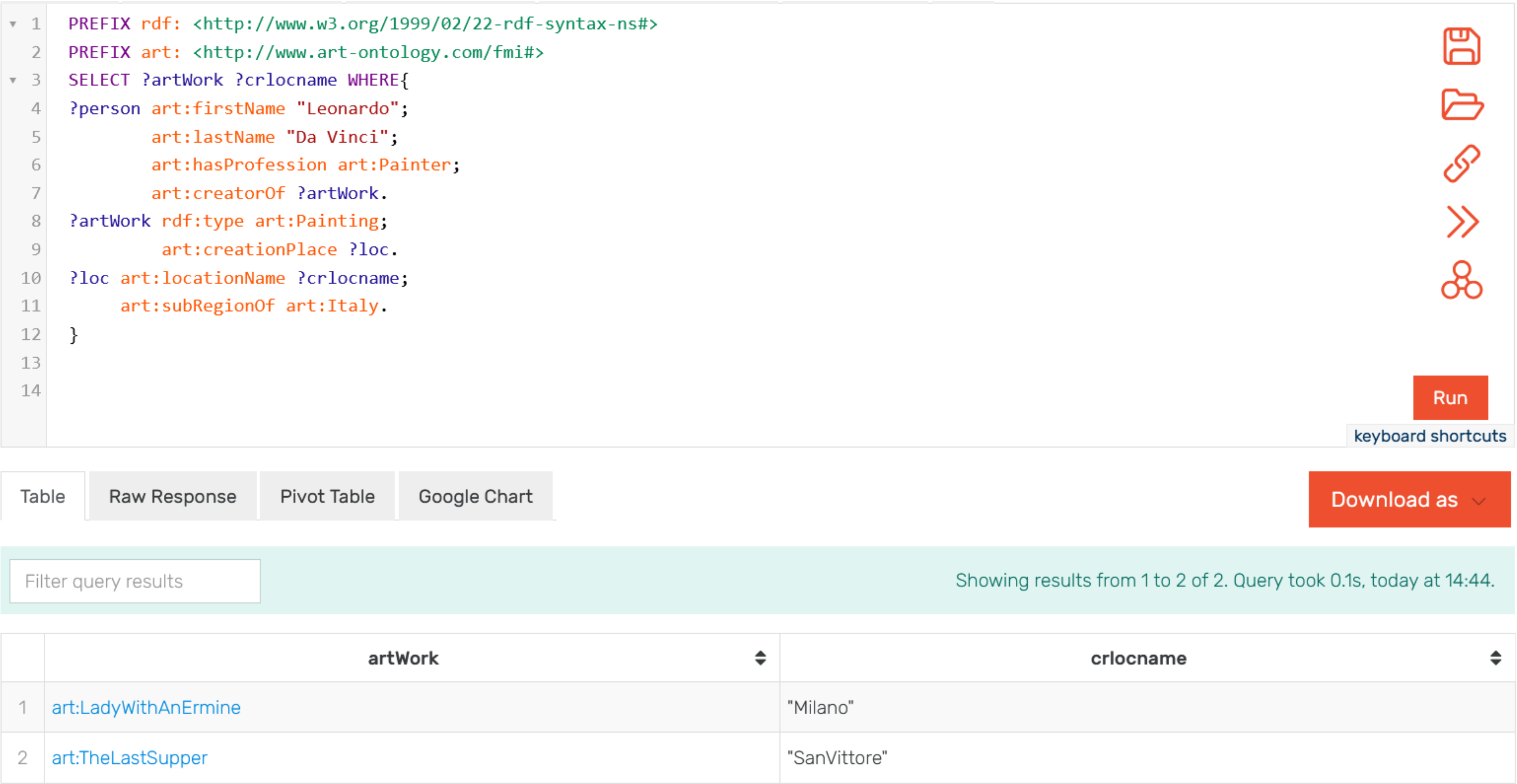


```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX art: <http://www.art-ontology.com/fmi#>
SELECT ?artWork ?crlocname WHERE{
?person art:firstName "Leonardo";
        art:lastName "Da Vinci";
        art:hasProfession art:Painter;
        art:creatorOf ?artWork.
?artWork rdf:type art:Painting;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

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

	artWork	crlocname
1	art:LadyWithAnErmine	"Milano"

Result with reasoning:



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

Showing results from 1 to 2 of 2. Query took 0.1s, today at 14:44.

	artWork	crlocname
1	art:LadyWithAnErmine	"Milano"
2	art:TheLastSupper	"SanVittore"

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**.

Questions?