Università degli Studi di Roma “Tor Vergata”

Gestione di dati RDF con Ontotext GraphDB
…ed un pizzico di LOV

Manuel Fiorelli

<fiorelli@info.uniroma2.it>

Ontotext GraphDB

Ontotext GraphDB è una *famiglia di repository semantici* affidabili, scalabili e ricchi di funzionalità.

- **GraphDB Free**: gratuito da usare, ma limitato a 2 query concorrenti
- **GraphDB SE**: nessun limite al numero di query concorrenti
- **GraphDB EE**: modalità cluster per garantire alta disponibilità (availability), resilienza (resilience) and throughput

Possibilità di deployment con Docker o Puppet.

GraphDB Free

GraphDB Free può essere **usato gratuitamente**, ma è necessario registrarsi sul sito di Ontotext:

- Il link per il download viene fornito in una email spedita all'indirizzo indicato nel form di registrazione

È **equivalente a GraphDB SE**, fatta eccezione per il fatto di essere limitato a due query concorrenti.

È implementato come un Sail RDF4J (ulteriori informazioni nelle prossime lezioni)

GraphDB Free – standalone server

Noi useremo lo **standalone server**:

- È una *distribuzione autosufficiente*: nel richiede un servlet container esterno (es. Apache Tomcat); tuttavia, necessità che nel sistema sia installato Java 8+
- Può essere lanciato *eseguendo un solo script* (*graphdb[.cmd]*)
- Il *workbench grafico* è raggiungile all'indirizzo http://localhost:7200

GraphDB Free – Funzionalità (1/4)

- Usabile gratuitamente;
- Gestisce decine di miliardi di statement RDF su un singolo server;
- Esegue operazioni di query e ragionamento usando indici basati su file;
- Pieno supporto a SPARQL 1.1;
- Deployment e portabilità semplificate dall'uso di JAVA;
- Scalabilità, sia in termini di volume dei dati sia di velocità di caricamento e inferenza;
- Compatibile con RDF4J 2.0;
- Compatible con Jena attraverso un adattatore incluso;

Fonte: http://graphdb.ontotext.com/documentation/free/free/graphdb-free.html

GraphDB Free – Funzionalità (2/4)

- Ragionamento pienamente compatibile con gli standard per RDFS, OWL 2 RL e QL;
- Supporto per regole personalizzate; regole ottimizzate per le prestazioni;
- Supporto ottimizzato per l'integrazione dei dati attraverso owl:sameAs;
- Indici speciali per vincoli geo-spaziali efficienti (near-by, within, distance);
- Ricerca full-text, basata su Lucene;

Source: http://graphdb.ontotext.com/documentation/free/free/graphdb-free.html

GraphDB Free – Funzionalità (3/4)

- Ritrazione efficiente di statement inferita dopo aggiornamenti;
- Affidabile nel preservare i dati, la loro consistenza ed integrità;
- Import/export di sintassi RDF attraverso RDF4J: XML, N3, N-Triples, N-Quads, Turtle, TriG, TriX;
- API plugin framework, public classes and interfaces;
- Ottimizzatore di query che permette di valutare diverse piani di interrogazione;

Source: http://graphdb.ontotext.com/documentation/free/free/graphdb-free.html

GraphDB Free – Funzionalità (4/4)

- RDF rank per ordinare i risultati delle query per rilevanza o altre misure;
- Notifiche permettono ai client di reagire a statement nel flusso degli aggiornamenti;
- Connettore Lucene per ricerche normali e faceted (aggregazione) estremamente veloci; automaticamente aggiornato con i dati in GraphDB;
- GraphDB Workbench - lo strumento predefinito di amministrazione basato sul web;
- LoadRDF per creare repository molto velocemente da dataset grandi;

Source: http://graphdb.ontotext.com/documentation/free/free/graphdb-free.html

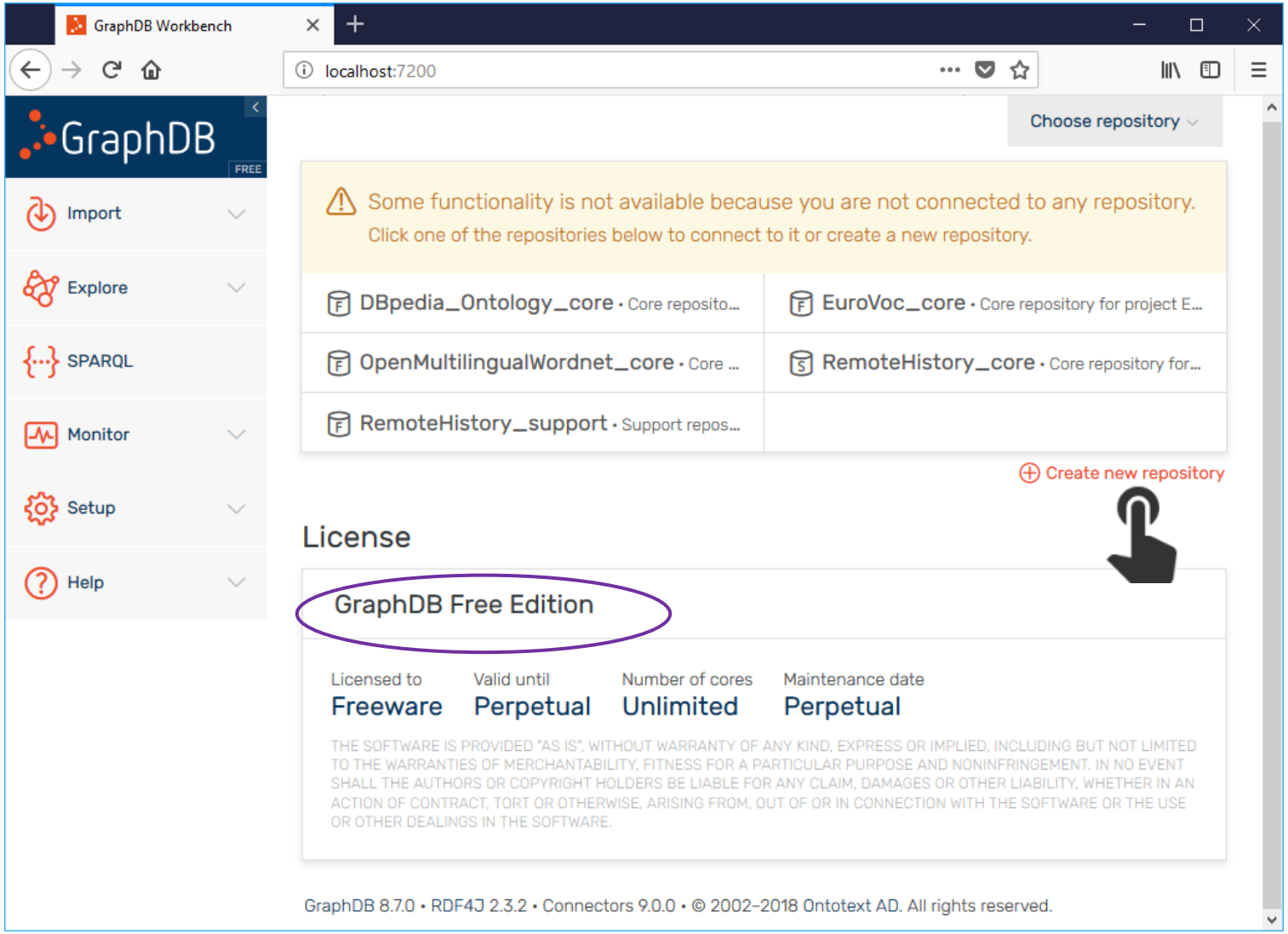
Avviare GraphDB

Avviare il Worbench di GraphDB è davvero semplice:

- Scompattare l'archivio relativo alla distruzione standalone
- Avviare lo script *bin/graphdb* (Linux) o *bin/graphdb.cmd* (Windows)
- Puntare il browser web all'indirizzo http://localhost:7200

Per impostazione predefinita, GraphDB parte in maniera non sicura; tuttavia, esso supporta l'**autenticazione** degli utenti, **controllo di accesso** basato su ruoli e l'uso della **crittografia in transito**. Inoltre, possibilità di specificare **timeout e limit sui risultati delle query**.

Workench a colpo d'occhio



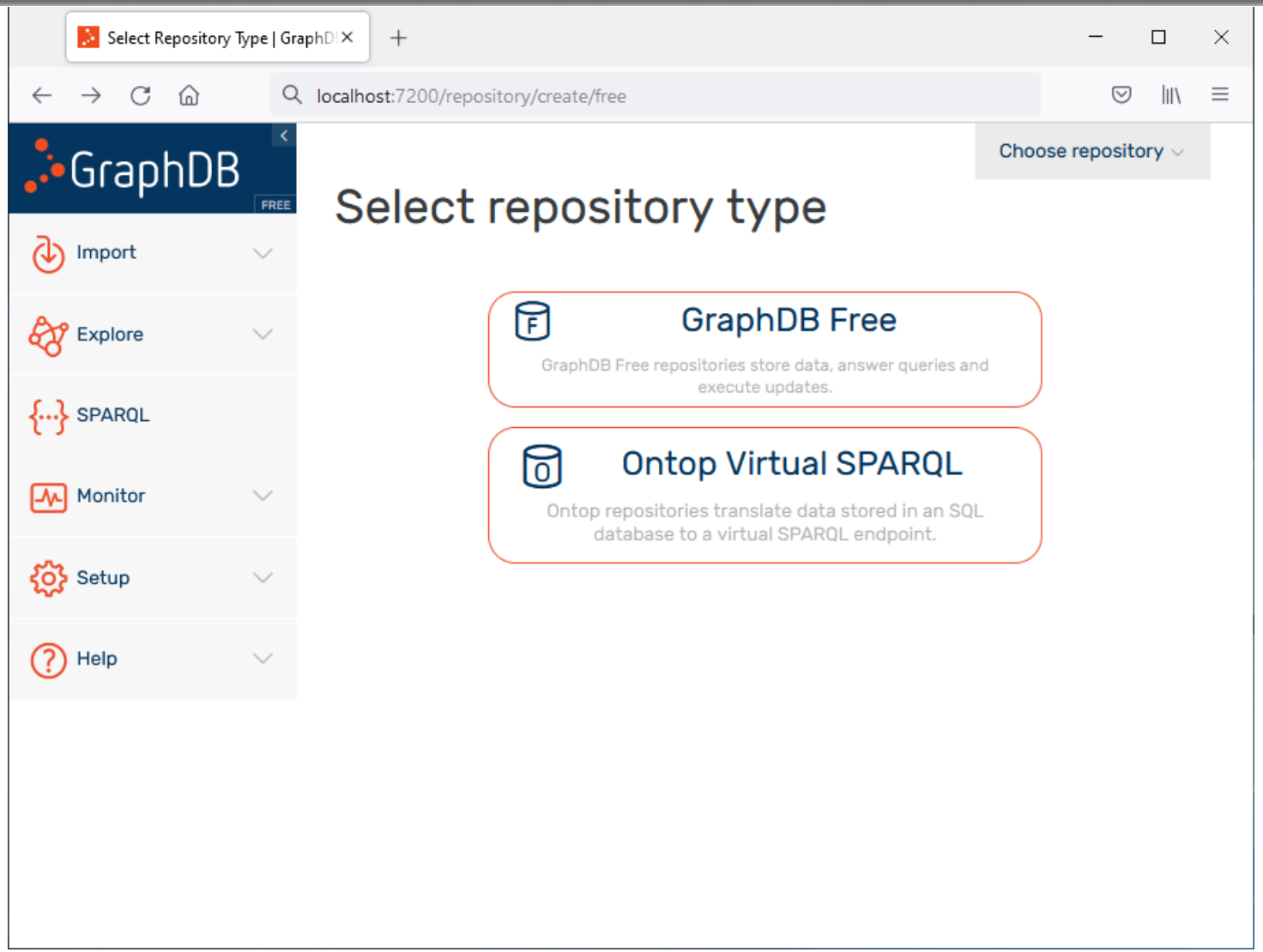
Creazione di un repository (1/4)

La *creazione di un nuovo repository* è il **primo passo** indispensabile per usare GraphDB.

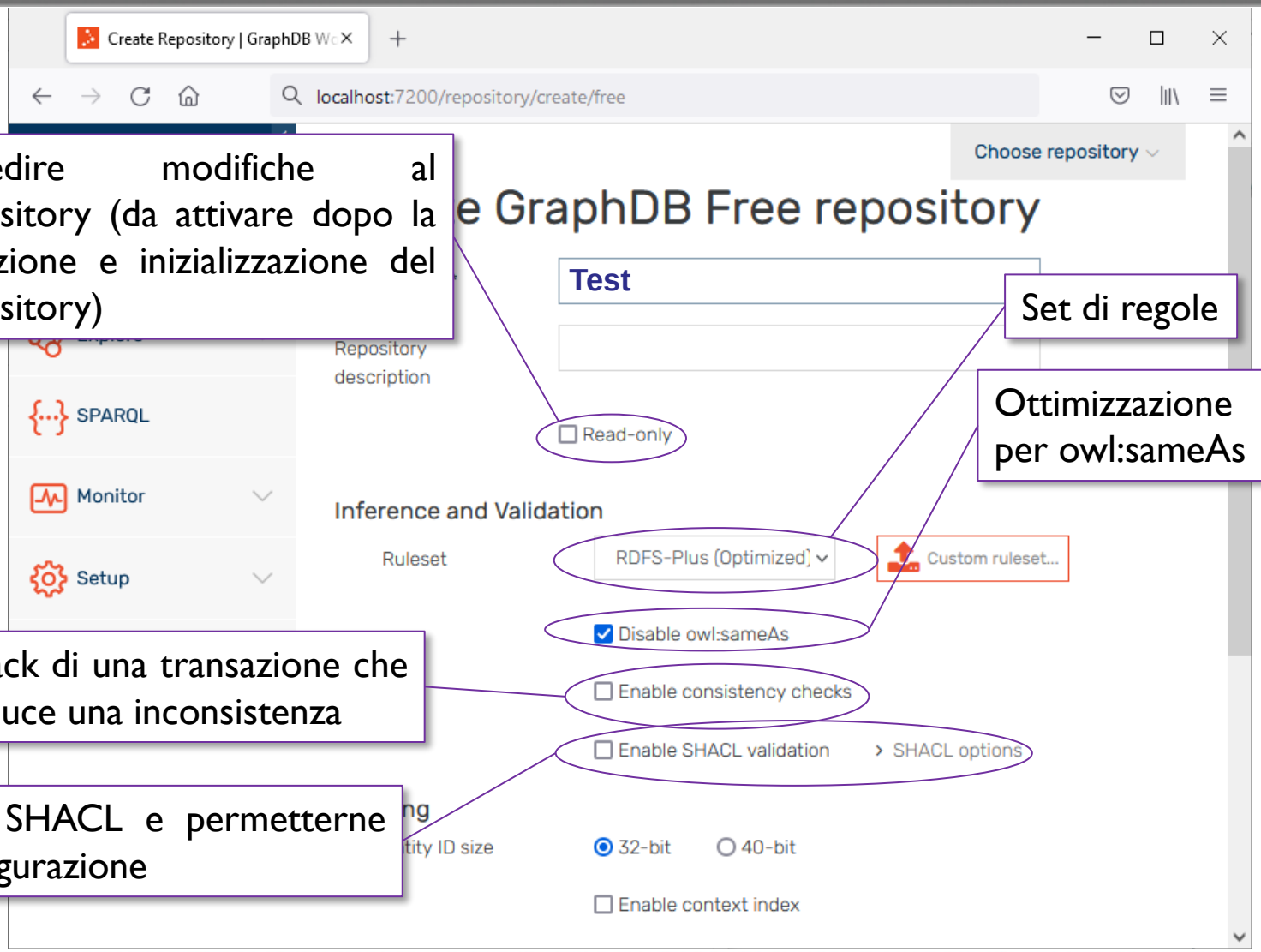
- Un repository è il *luogo in cui possiamo caricare le triple RDF* e la maggior parte delle operazioni sono eseguite su un certo repostory.
- Quando creiamo un repository, possiamo specificare diversi *parametri di configurazione* (alcuni dei quali possono essere cambiati in seguito): http://graphdb.ontotext.com/documentation/free/configuring-a-repository.html#configuring-a-repository-configuration-parameters

GraphDB può gestire più repository.

Creazione di un repository (2/4)



Creazione di un repository (3/4)



Impedire modifiche al repository (da attivare dopo la creazione e inizializzazione del repository)

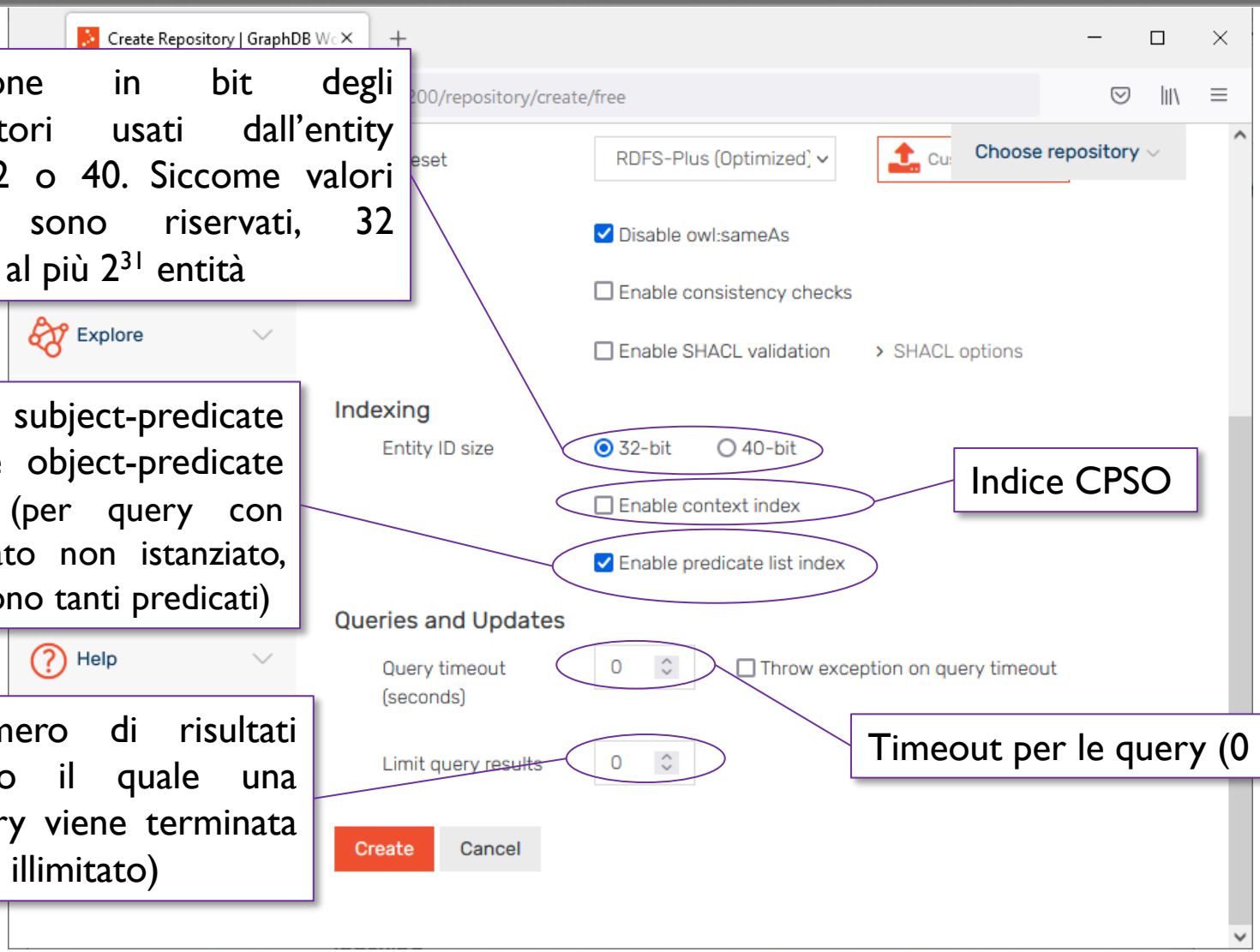
Set di regole

Ottimizzazione per owl:sameAs

Rollback di una transazione che introduce una inconsistenza

Abilita SHACL e permetterne la configurazione

Creazione di un repository (4/4)



Dimensione in bit degli identificatori usati dall'entity index: 32 o 40. Siccome valori negativi sono riservati, 32 supporta al più 2^{31} entità

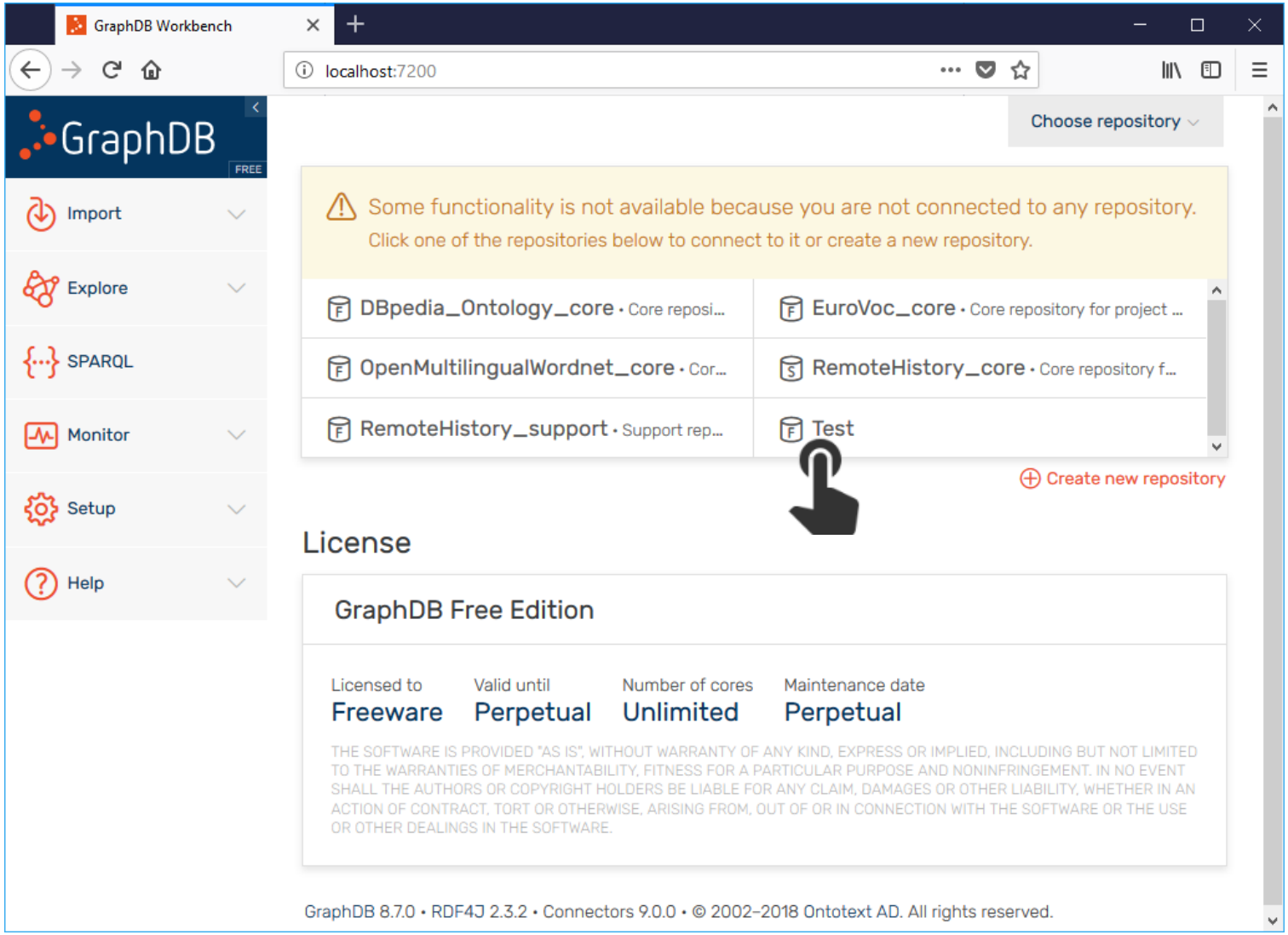
Indici subject-predicate (SP) e object-predicate (OP) (per query con predicato non istanziato, se ci sono tanti predicati)

Indice CPSO

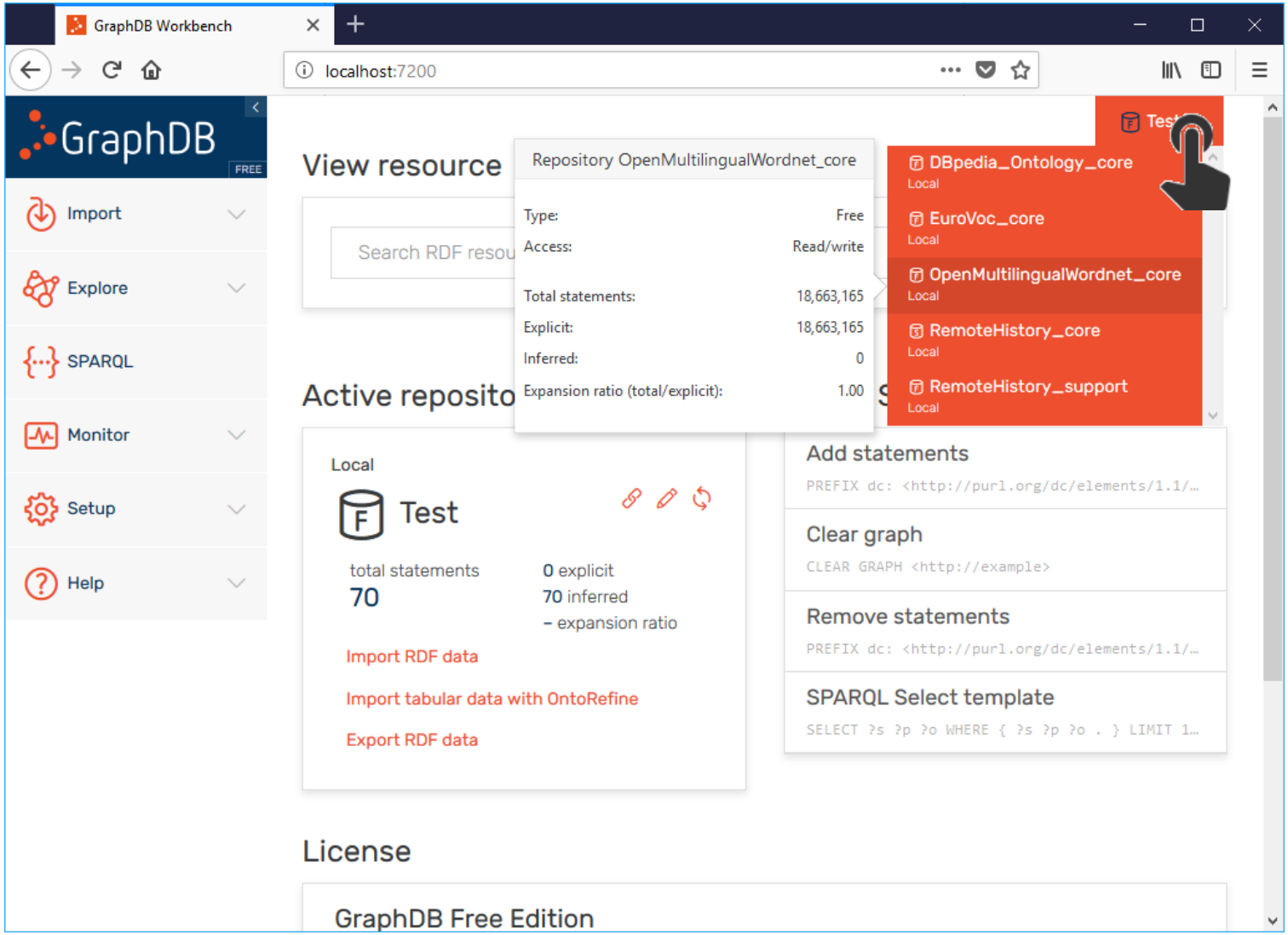
Numero di risultati dopo il quale una query viene terminata (0 = illimitato)

Timeout per le query (0 = illimitato)

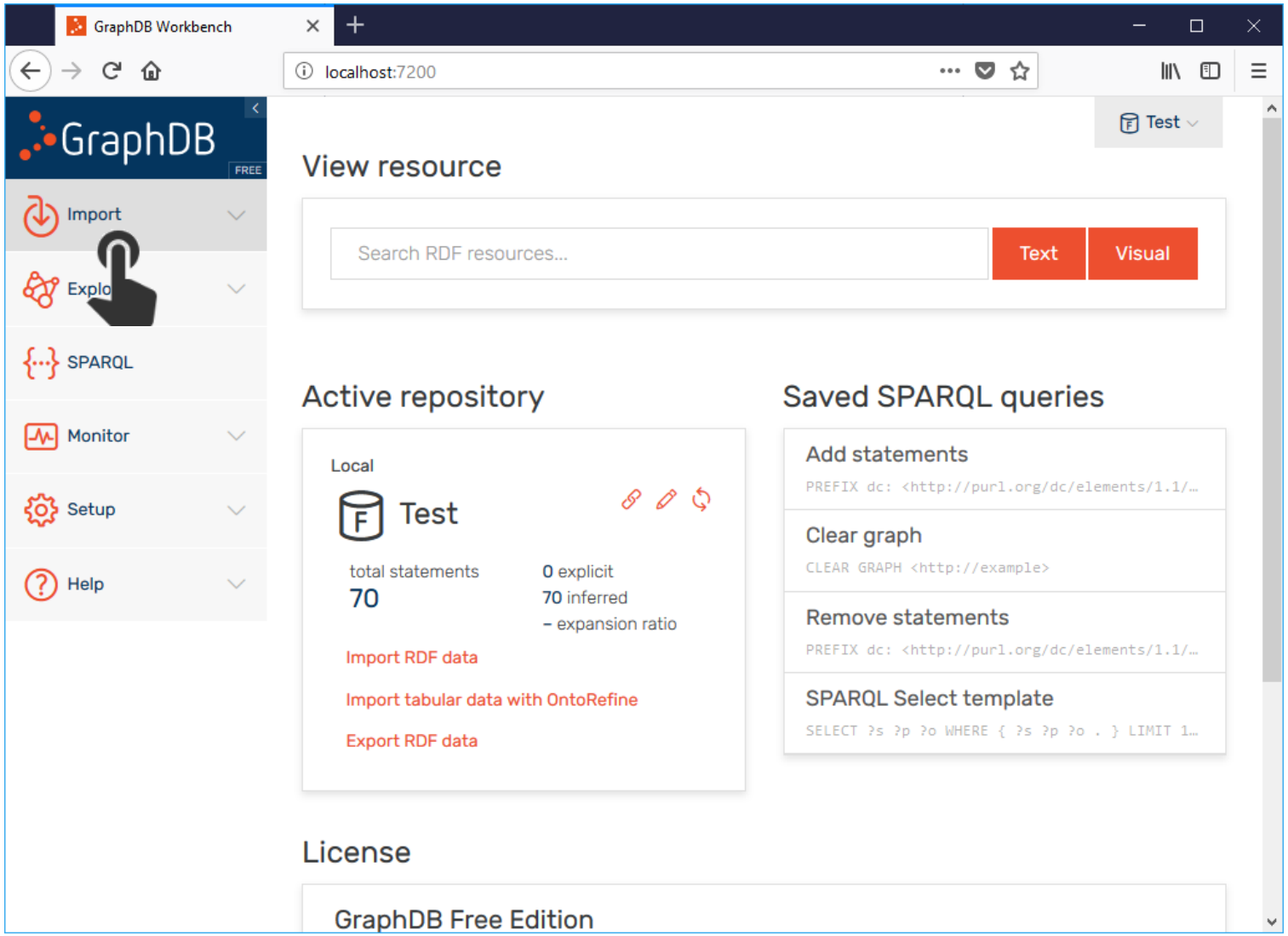
Selezione di un repository (1/2)



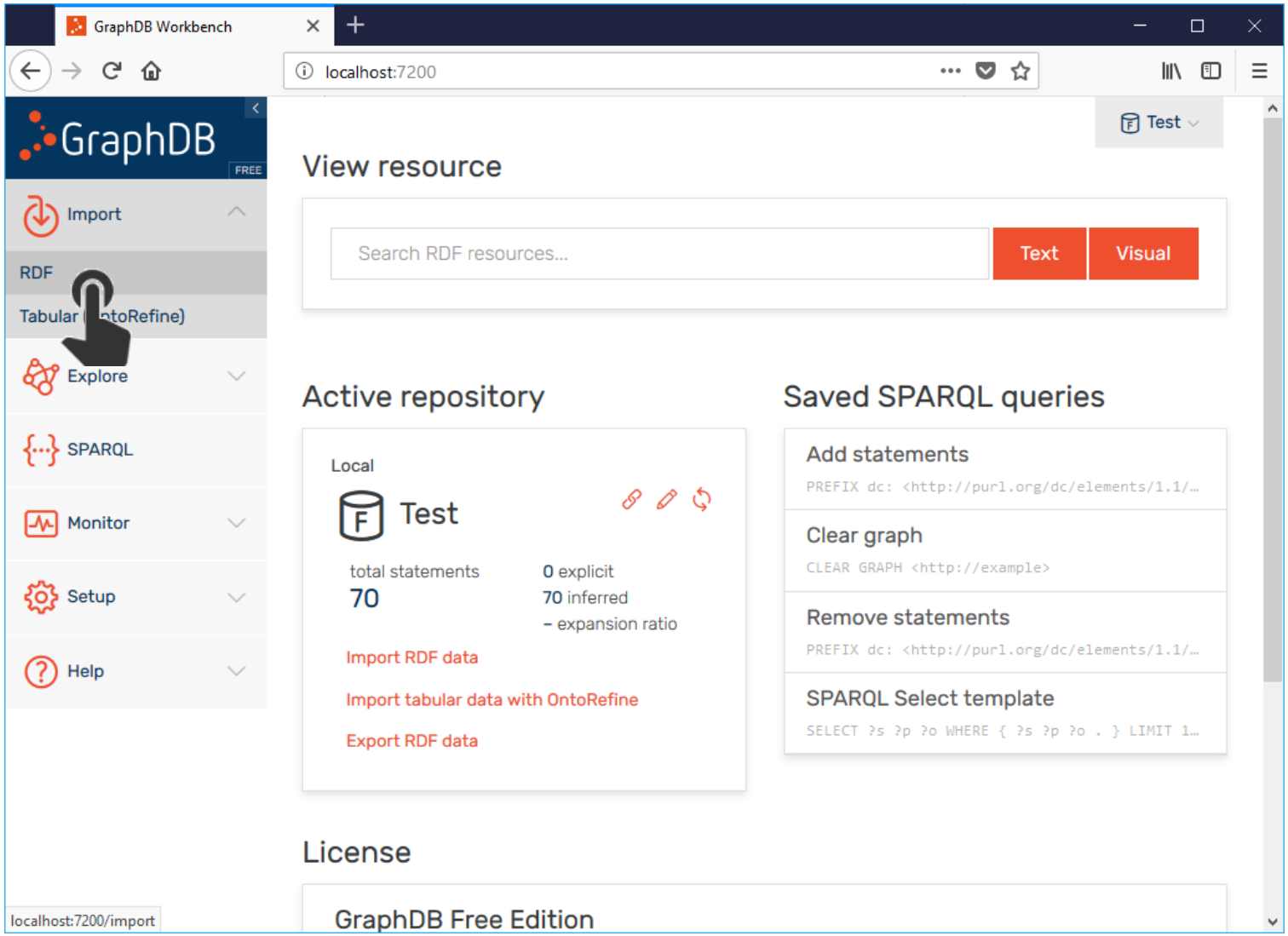
Selezione di un repository (2/2)



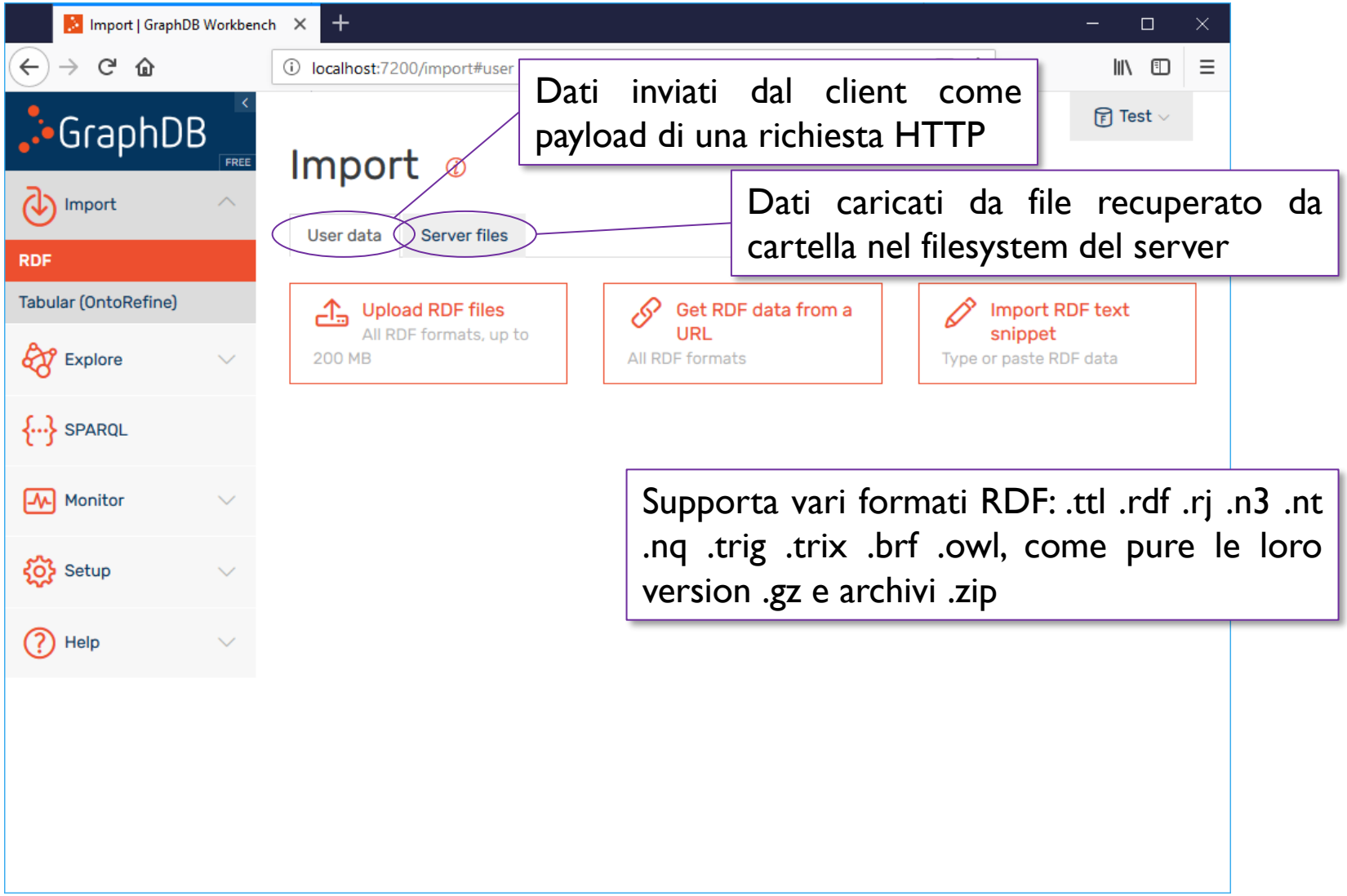
Import (1/2)


GraphDB Workbench
localhost:7200
GraphDB
FREE
Import
Explo
SPARQL
Monitor
Setup
Help
Test
View resource
Search RDF resources...
Text
Visual
Active repository
Local
Test
total statements
70
0 explicit
70 inferred
– expansion ratio
Import RDF data
Import tabular data with OntoRefine
Export RDF data
Saved SPARQL queries
Add statements
PREFIX dc: <http://purl.org/dc/elements/1.1/…
Clear graph
CLEAR GRAPH <http://example>
Remove statements
PREFIX dc: <http://purl.org/dc/elements/1.1/…
SPARQL Select template
SELECT ?s ?p ?o WHERE { ?s ?p ?o . } LIMIT 1…
License
GraphDB Free Edition

Import (2/2)


GraphDB Workbench
localhost:7200
GraphDB
FREE
Import
RDF
Tabular (OntoRefine)
Explore
SPARQL
Monitor
Setup
Help
Test
View resource
Search RDF resources...
Text
Visual
Active repository
Local
Test
total statements
70
0 explicit
70 inferred
– expansion ratio
Import RDF data
Import tabular data with OntoRefine
Export RDF data
Saved SPARQL queries
Add statements
PREFIX dc: <http://purl.org/dc/elements/1.1/…
Clear graph
CLEAR GRAPH <http://example>
Remove statements
PREFIX dc: <http://purl.org/dc/elements/1.1/…
SPARQL Select template
SELECT ?s ?p ?o WHERE { ?s ?p ?o . } LIMIT 1…
License
GraphDB Free Edition
localhost:7200/import

Import RDF


Import | GraphDB Workbench
localhost:7200/import#user
GraphDB
FREE
Import
RDF
Tabular (OntoRefine)
Explore
SPARQL
Monitor
Setup
Help
Test
Import
User data
Server files
Upload RDF files
All RDF formats, up to 200 MB
Get RDF data from a URL
All RDF formats
Import RDF text snippet
Type or paste RDF data
Dati inviati dal client come payload di una richiesta HTTP
Dati caricati da file recuperato da cartella nel filesystem del server
Supporta vari formati RDF: .ttl .rdf .rj .n3 .nt .nq .trig .trix .brf .owl, come pure le loro version .gz e archivi .zip

Caricamento della prima ontologia

Vogliamo caricare la nostra prima ontologia…

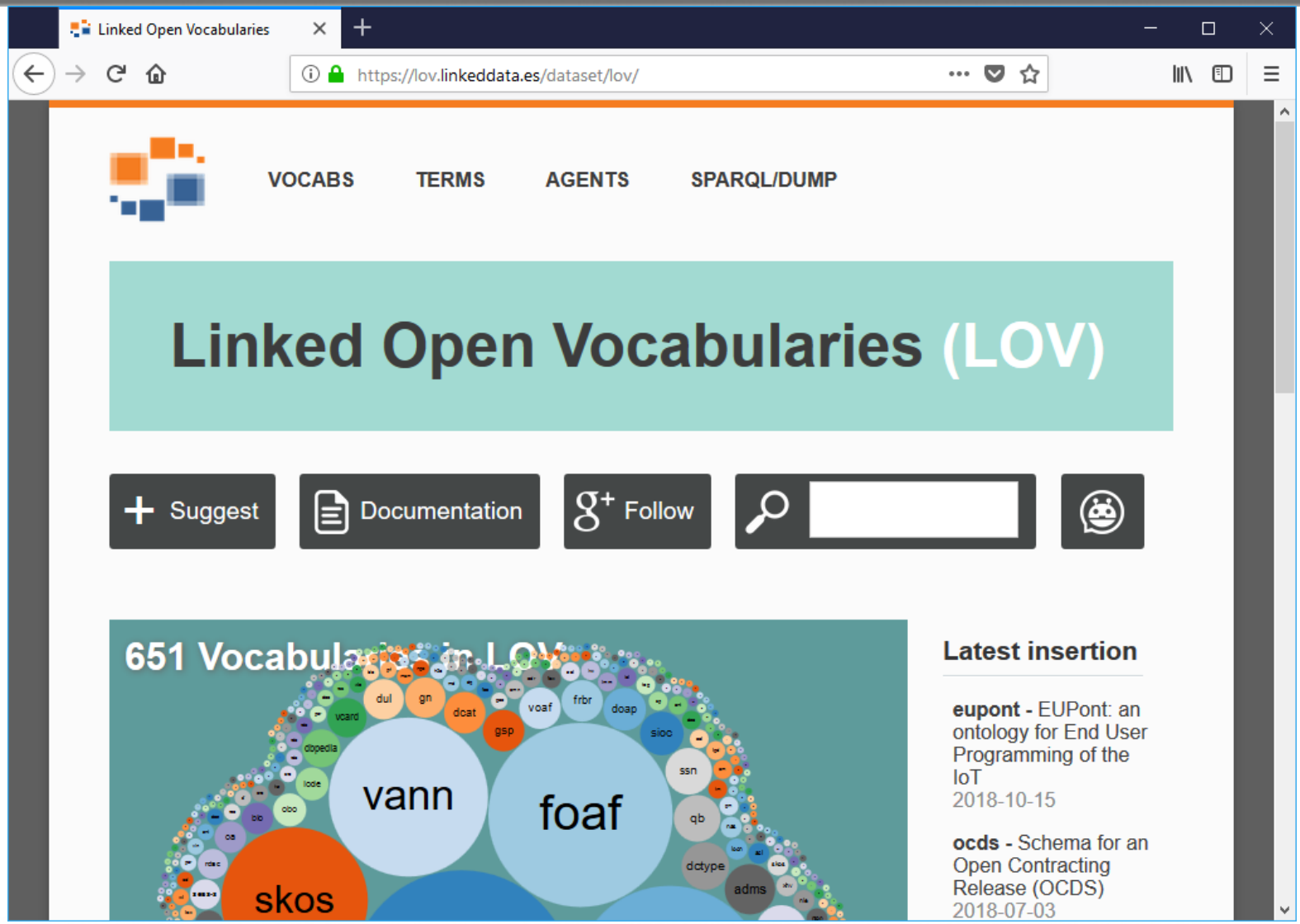
…in particolare, una *ontologia che parli delle persone e delle loro relazioni sociali.*

Andiamo su Linked Open Vocabularies (LOV):

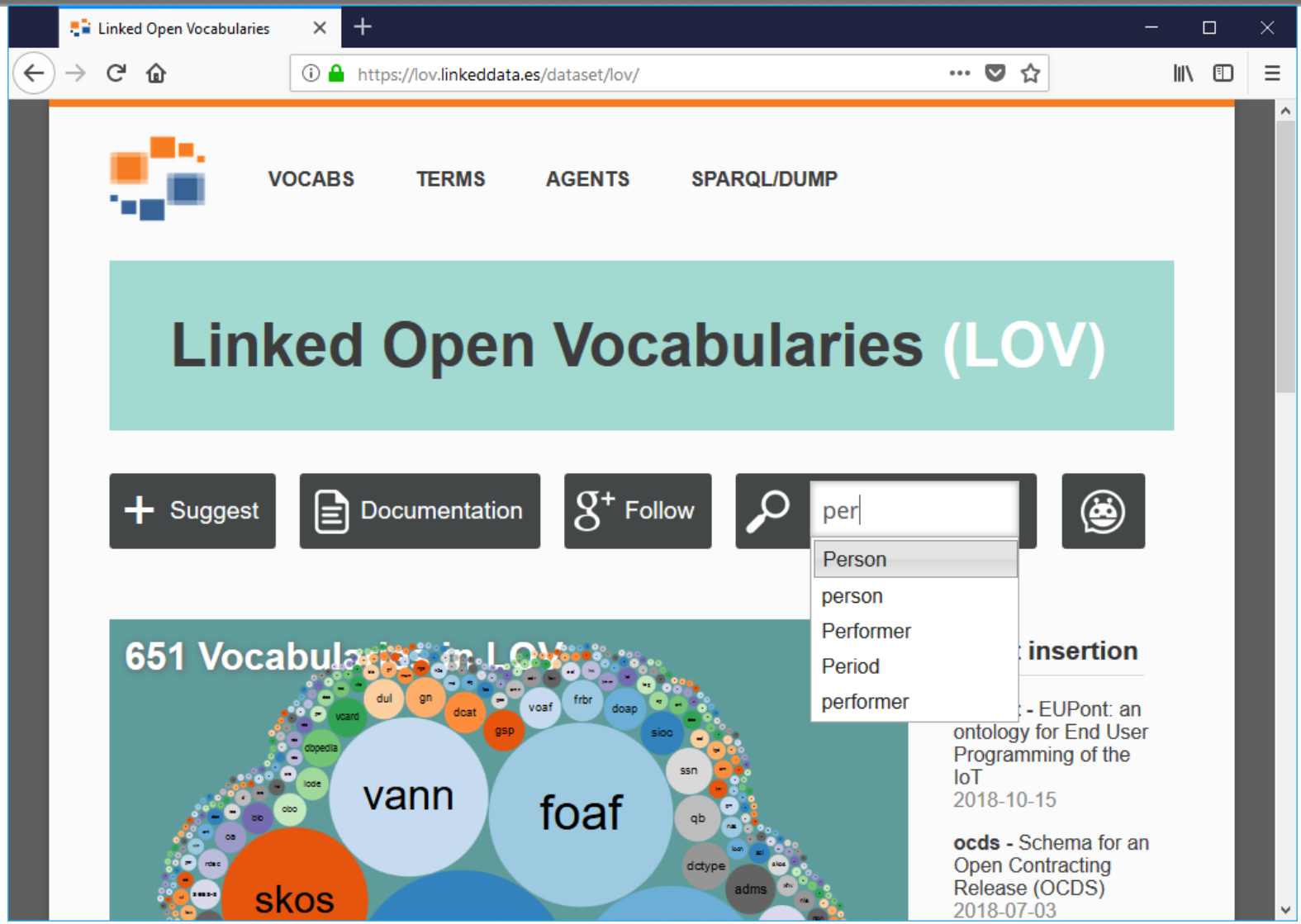
https://lov.linkeddata.es/

- Un catalogo (curato!) di vocabolari
- Metadati rappresentati in RDF usando VoID, VOAF ed altri vocabolari
- Copia cache dei vocabolari
- Catalogo accessibile via dump, SPARQL endpoint o API
- Fulltext search sui termini definiti dai vocabolari indicizzati

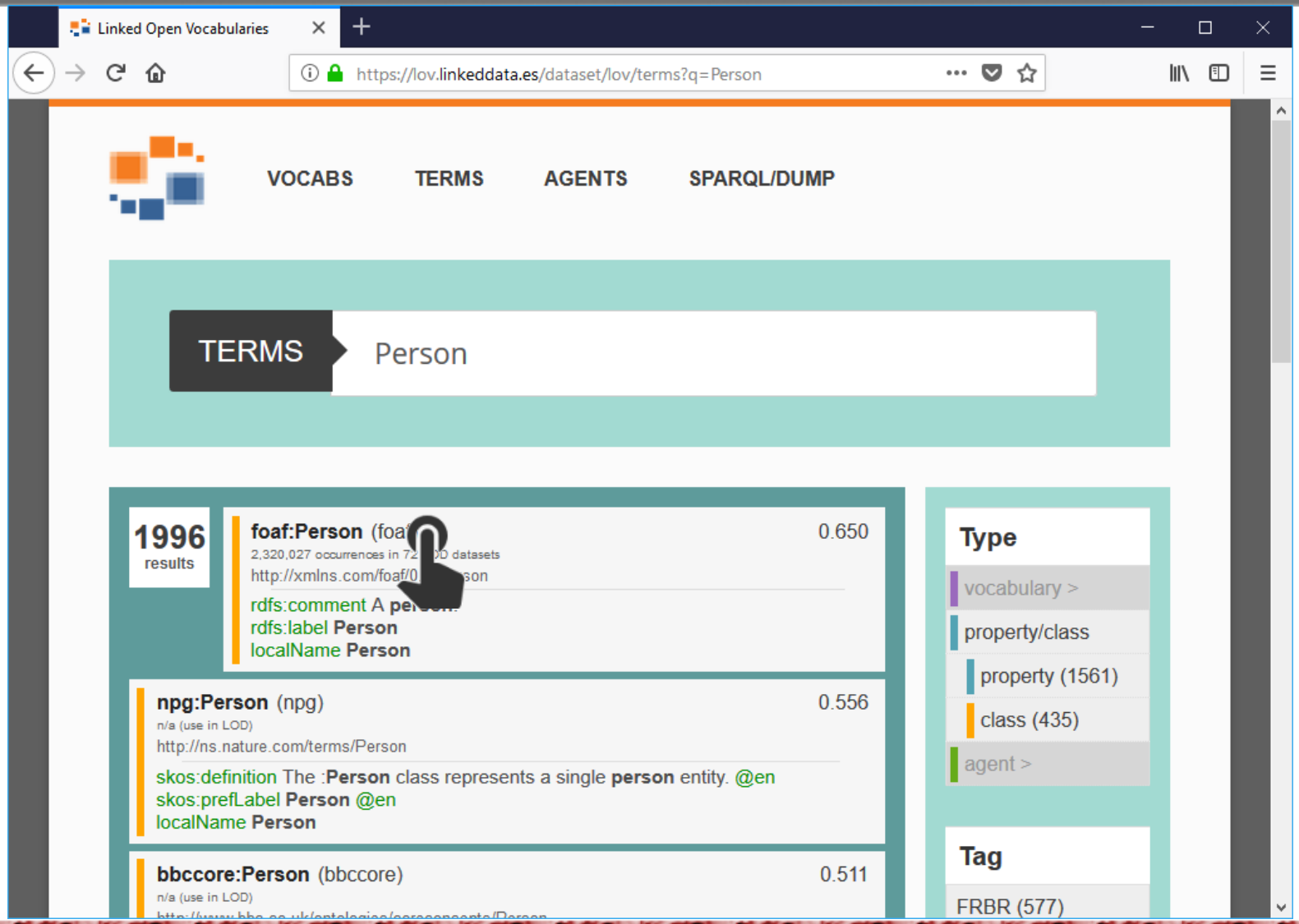
LOV - Homepage



LOV – Ricerca (con completamento)



LOV – Risultati della ricerca



LOV – FOAF (1)


Linked Open Vocabularies
https://lov.linkeddata.es/dataset/lov/vocabs/foaf
VOCABS
TERMS
AGENTS
SPARQL/DUMP
Friend of a Friend vocabulary
(foaf)
Metadata
URI
http://xmlns.com/foaf/0.1/
Namespace
http://xmlns.com/foaf/0.1/
homepage
http://www.foaf-project.org/
Description
FOAF is a project devoted to linking people and information using the Web. Regardless of whether information is in people's heads, in physical or digital documents, or in the
n3
Statistics
Classes 13
Properties 62
Datatypes 0
Instances 0
Statistiche sul contenuto del vocabolario

LOV – FOAF (2)

LOV integra diversi servizi esterni:

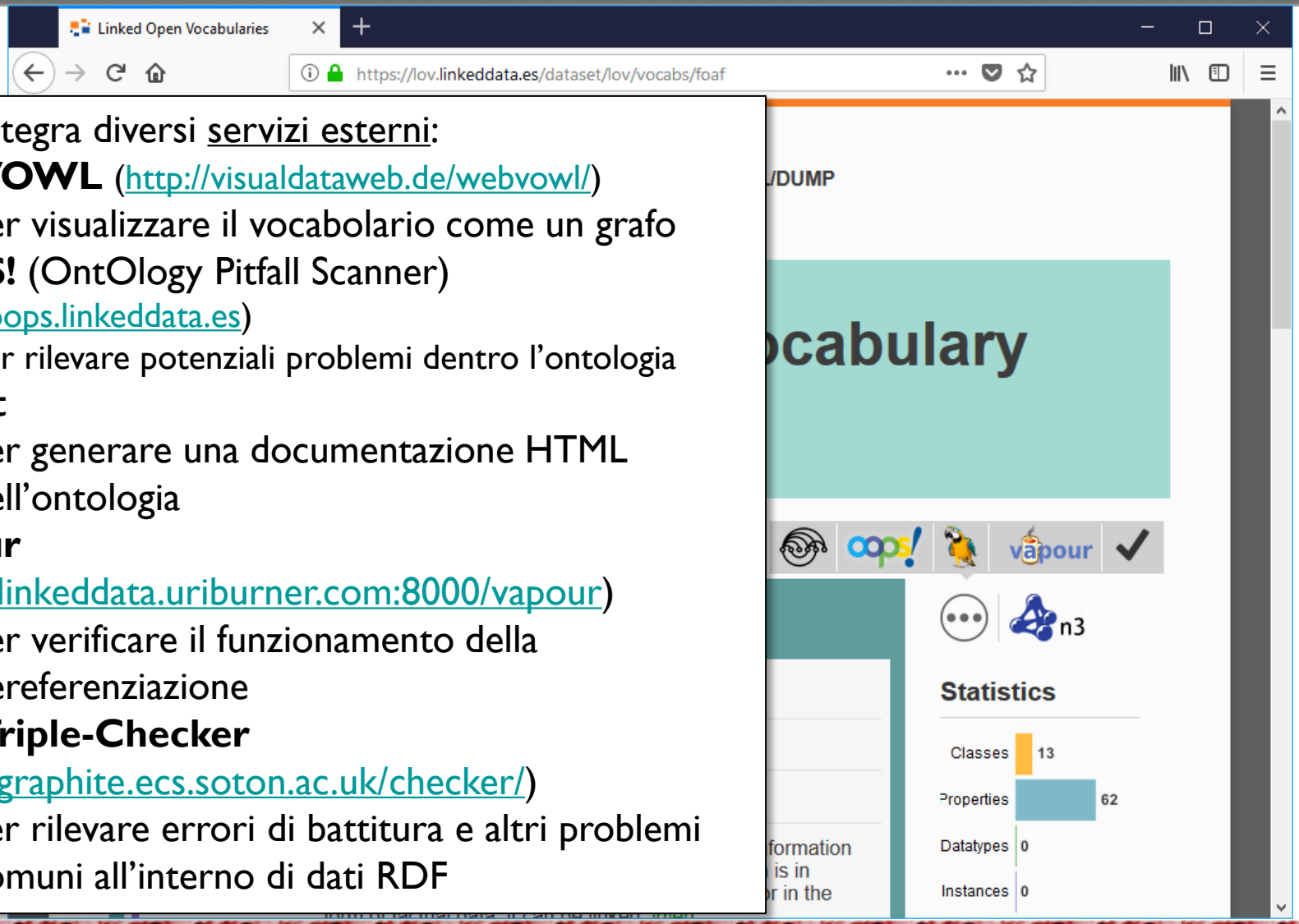
WebVOWL (http://visualdataweb.de/webvowl/)
per visualizzare il vocabolario come un grafo

OOPS! (OntOlogy Pitfall Scanner)
(http://oops.linkeddata.es)
per rilevare potenziali problemi dentro l'ontologia

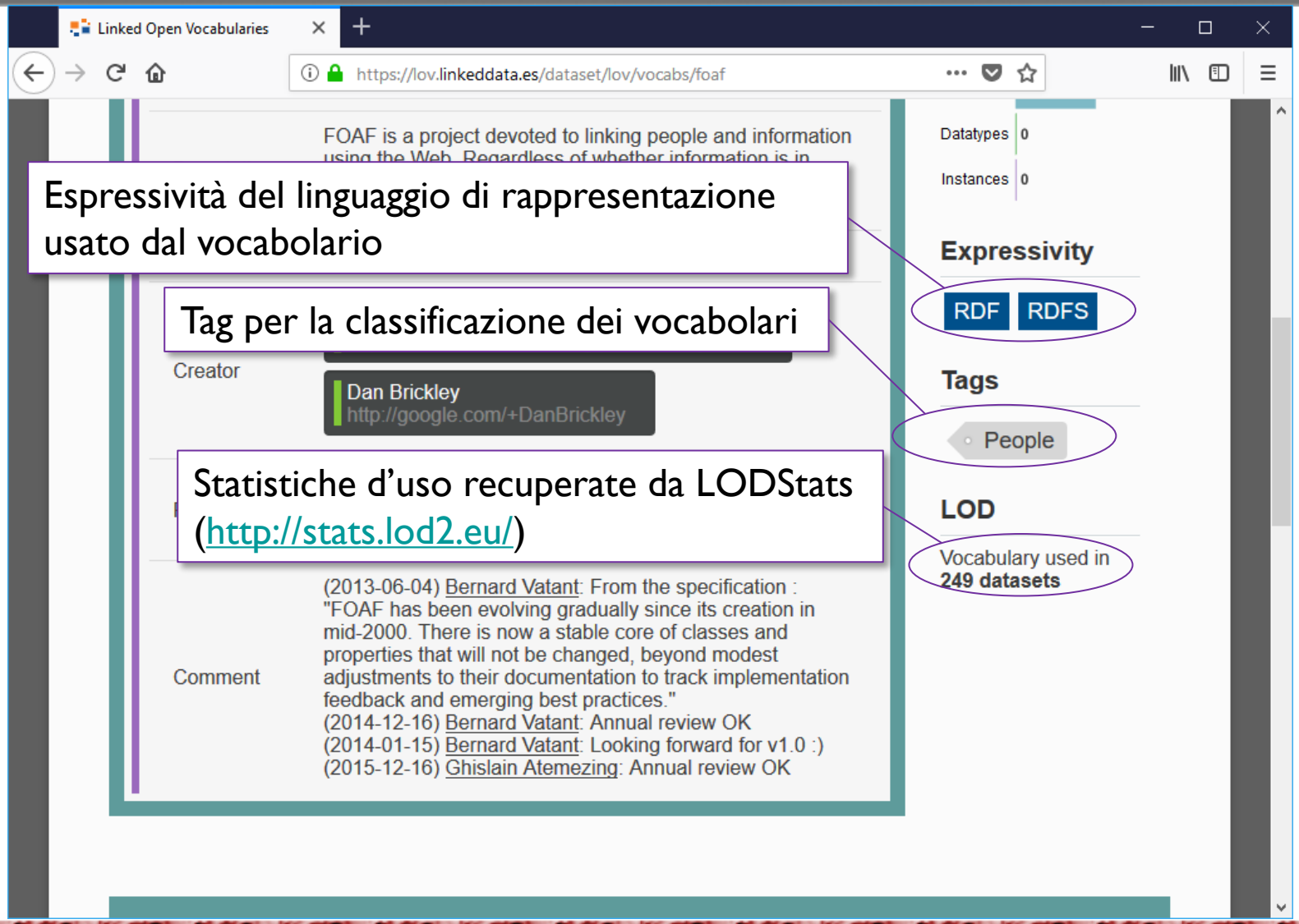
Parrot
per generare una documentazione HTML dell'ontologia

Vapour
(http://linkeddata.uriburner.com:8000/vapour)
per verificare il funzionamento della dereferenziazione

RDF Triple-Checker
(http://graphite.ecs.soton.ac.uk/checker/)
Per rilevare errori di battitura e altri problemi comuni all'interno di dati RDF



LOV – FOAF (3)



Espressività del linguaggio di rappresentazione usato dal vocabolario

Tag per la classificazione dei vocabolari

Statistiche d'uso recuperate da LODStats (http://stats.lod2.eu/)

LOV – FOAF (4)

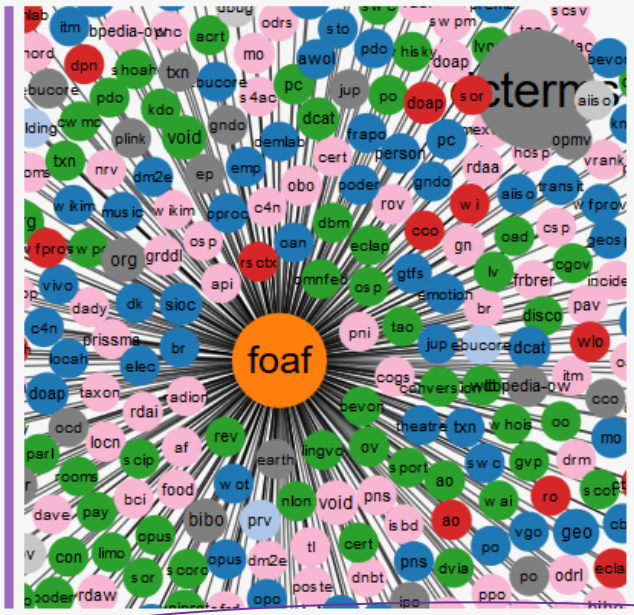
Linked Open Vocabularies

https://lov.linkeddata.es/dataset/lov/vocabs/foaf

Collegamenti tra vocabolari

373 Incoming Links

13 Outgoing Links



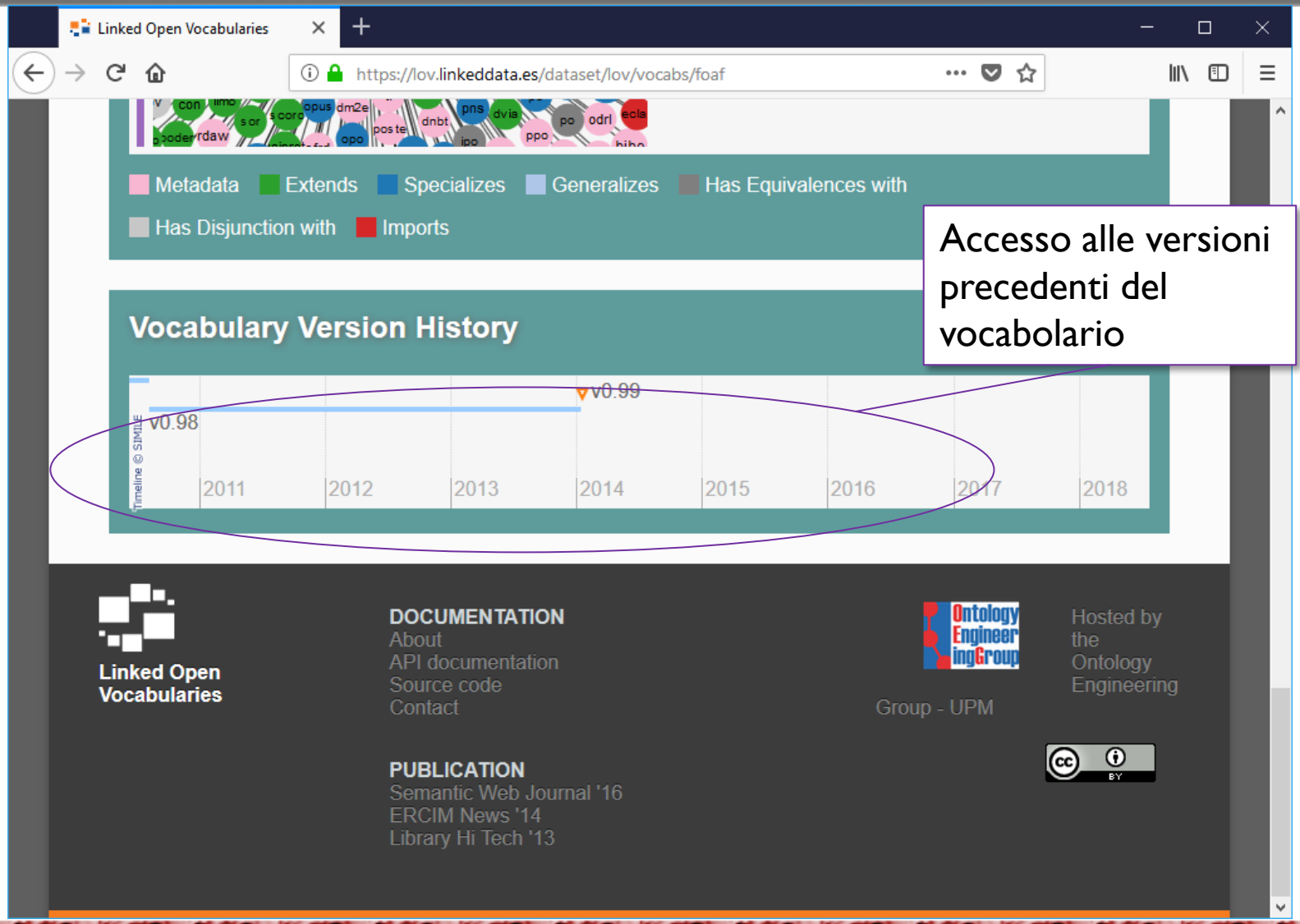
Tipi di collegamento

Metadata Extends Specializes Generalizes Has Equivalences with

Has Disjunction with Imports

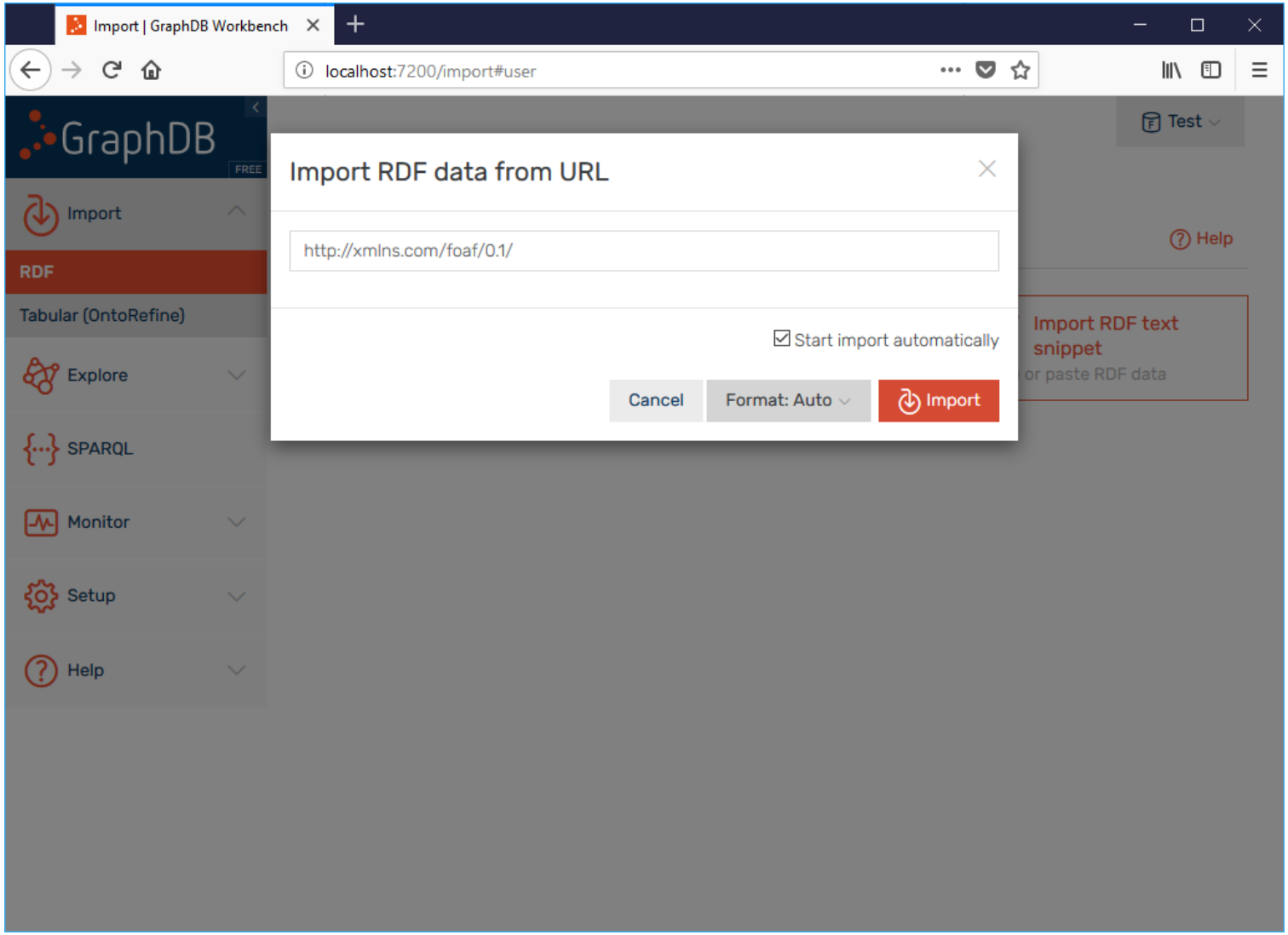
Manuel Fiorelli fiorelli@info.uniroma2.it
http://art.uniroma2.it/fiorelli

LOV – FOAF (5)

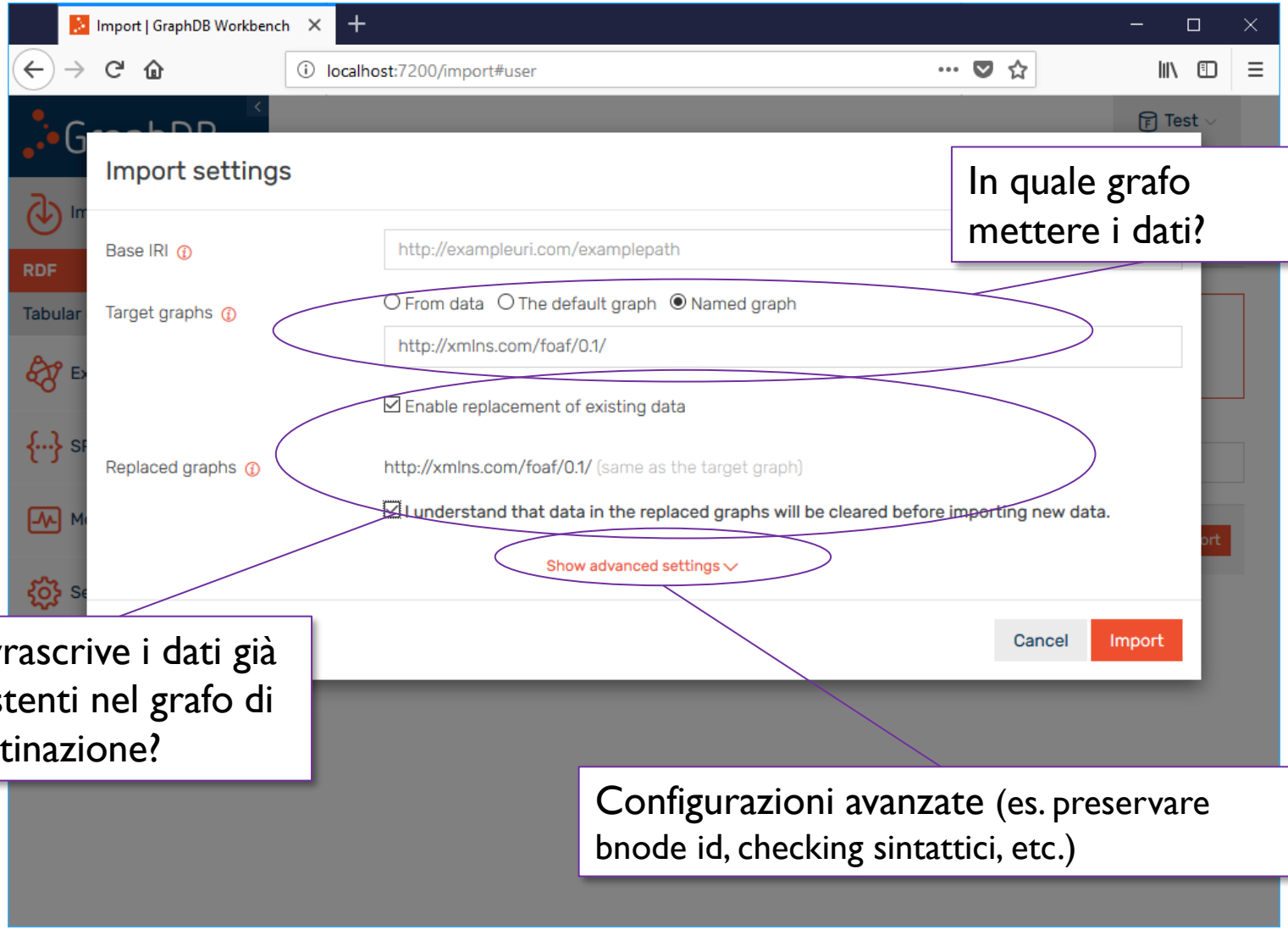


Accesso alle versioni precedenti del vocabolario

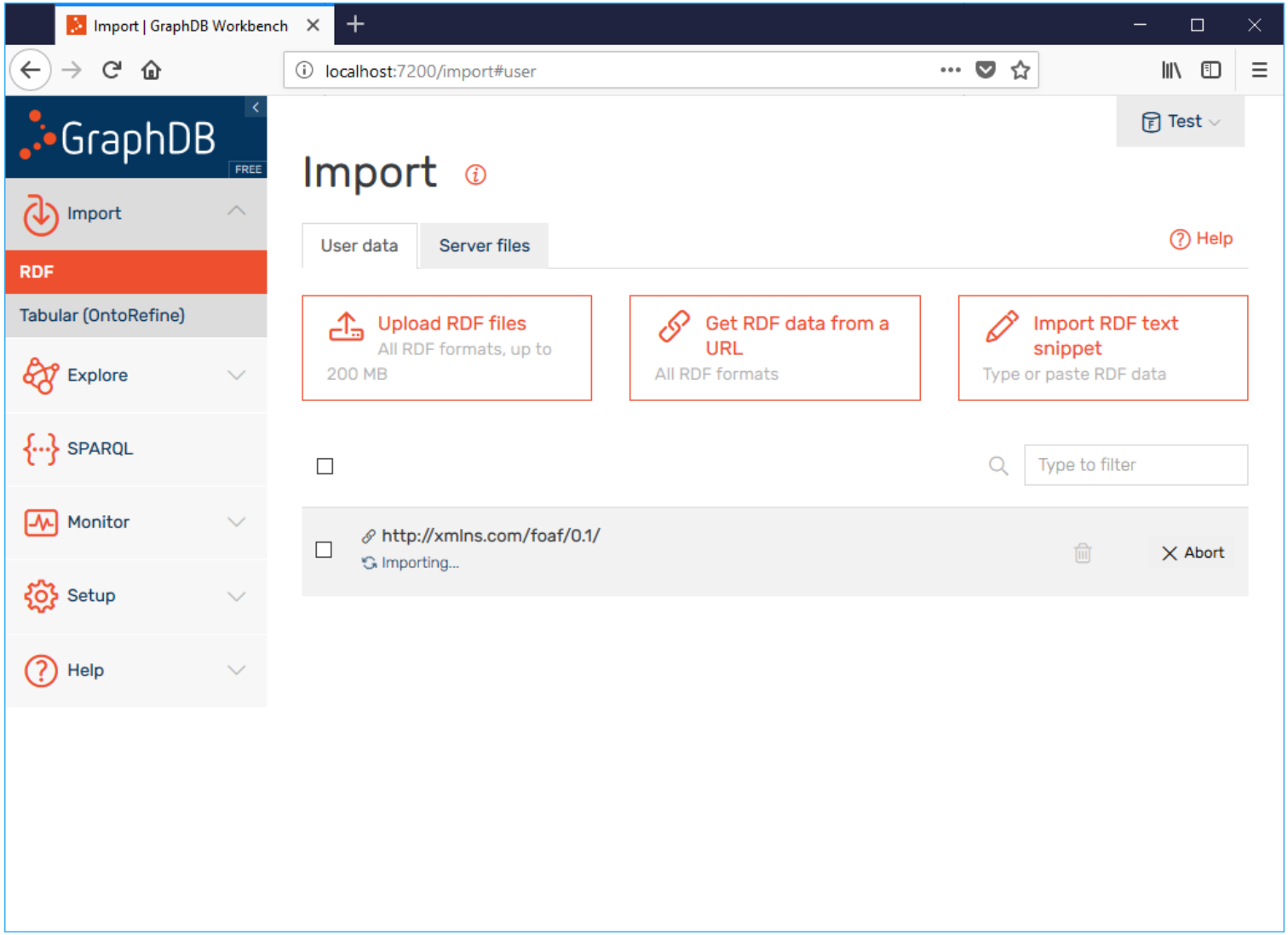
Import di dati RDF da URL (1/3)



Import di dati RDF da URL (2/3)



Import di dati RDF da URL (3/3)



Visualizzazione

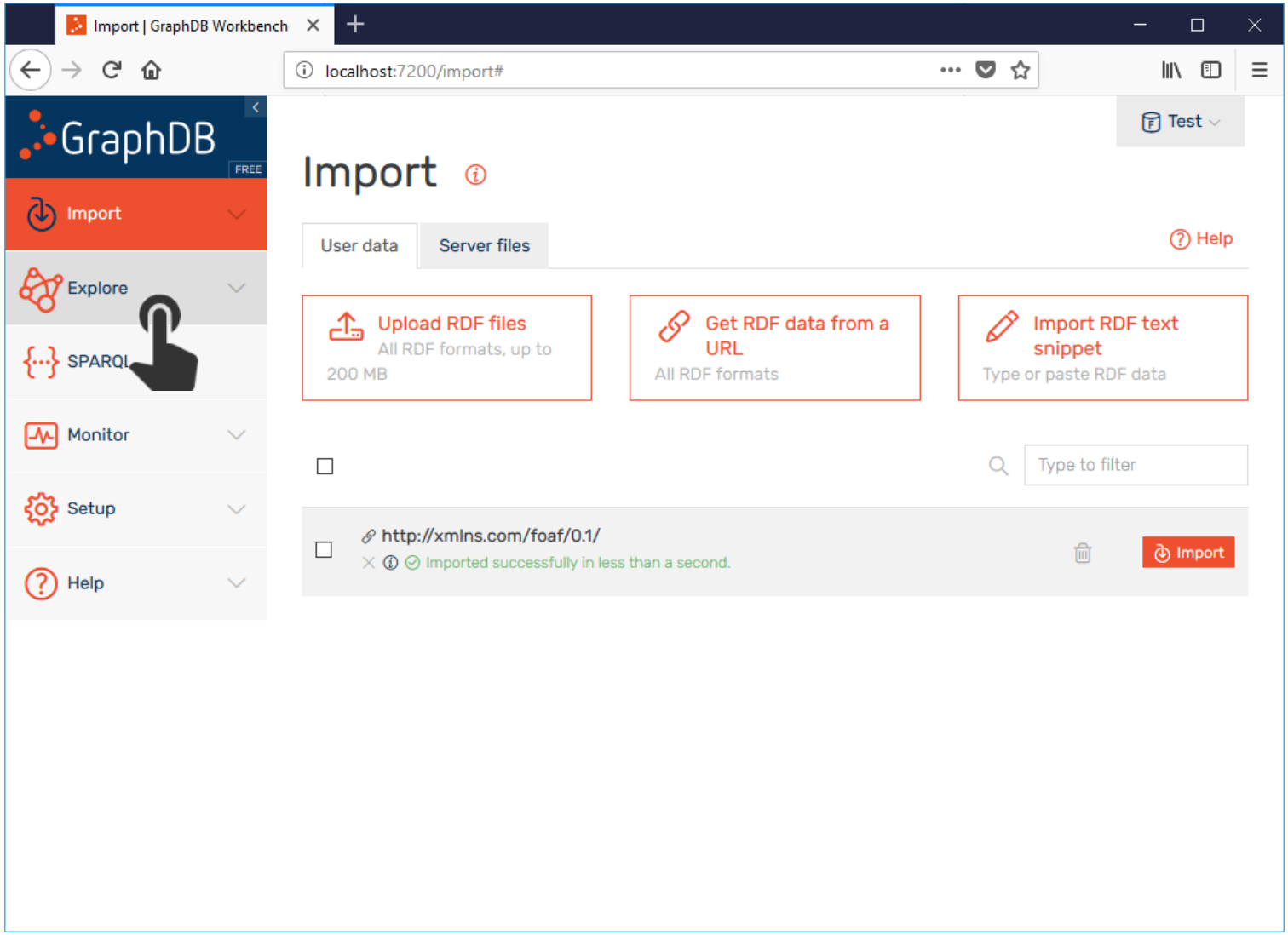
Disporre di **opportune visualizzazioni** è indispensabile per comprendere i dati e generare approfondimenti (insight)

GraphDB offre diverse visualizzazioni:

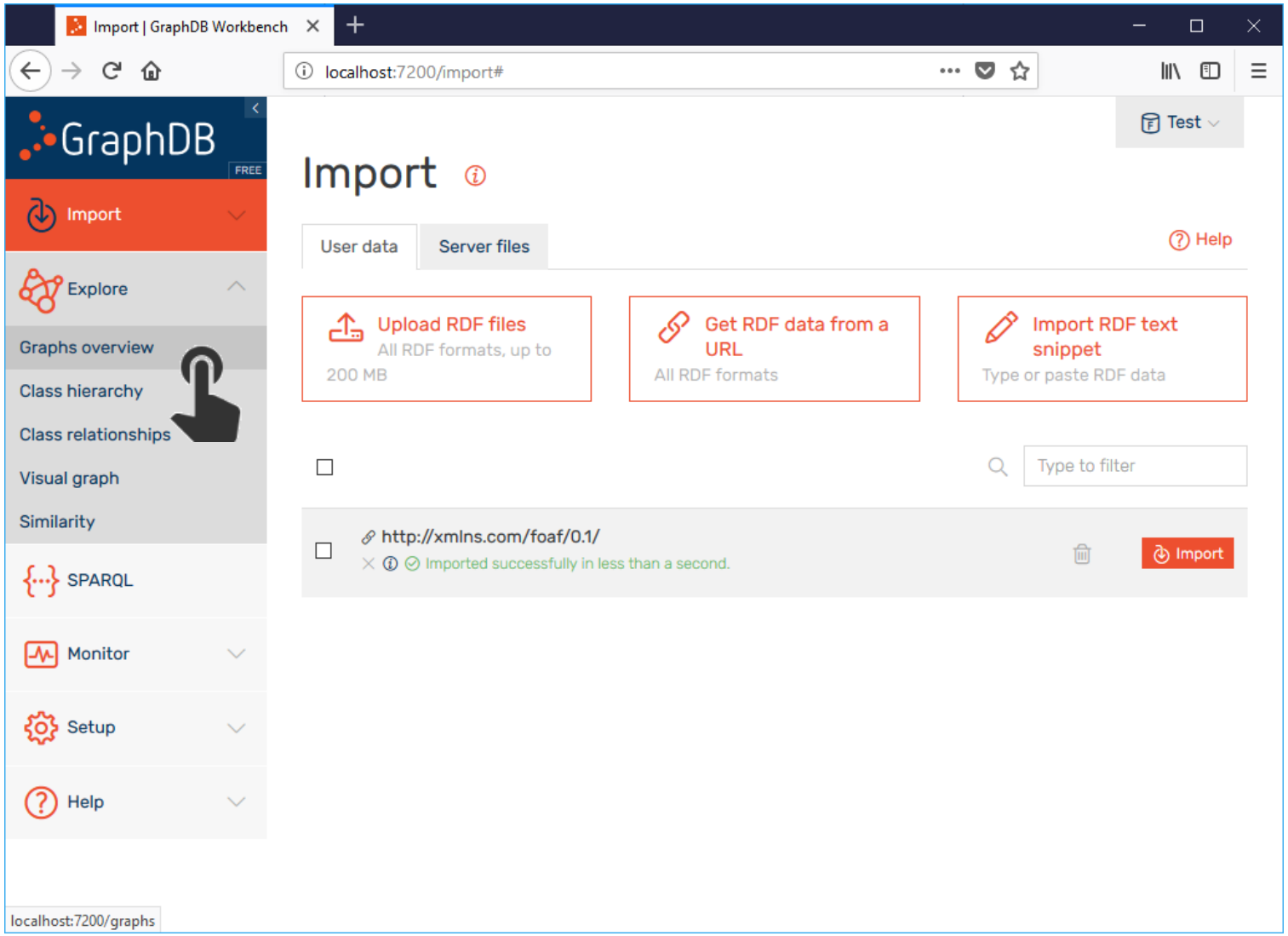
- resource view (visualizzazione/editing delle triple)
- visual graph (connettività delle risorse)
- domain-range graph (relazioni tra classi)
- class relationships (relazioni tra istanze delle classi)

RDF Rank permette di determinare l'«importanza» delle risorse, e ciò influisce sulle visualizzazioni esaltando certe parti dei dati e tagliandone altre.

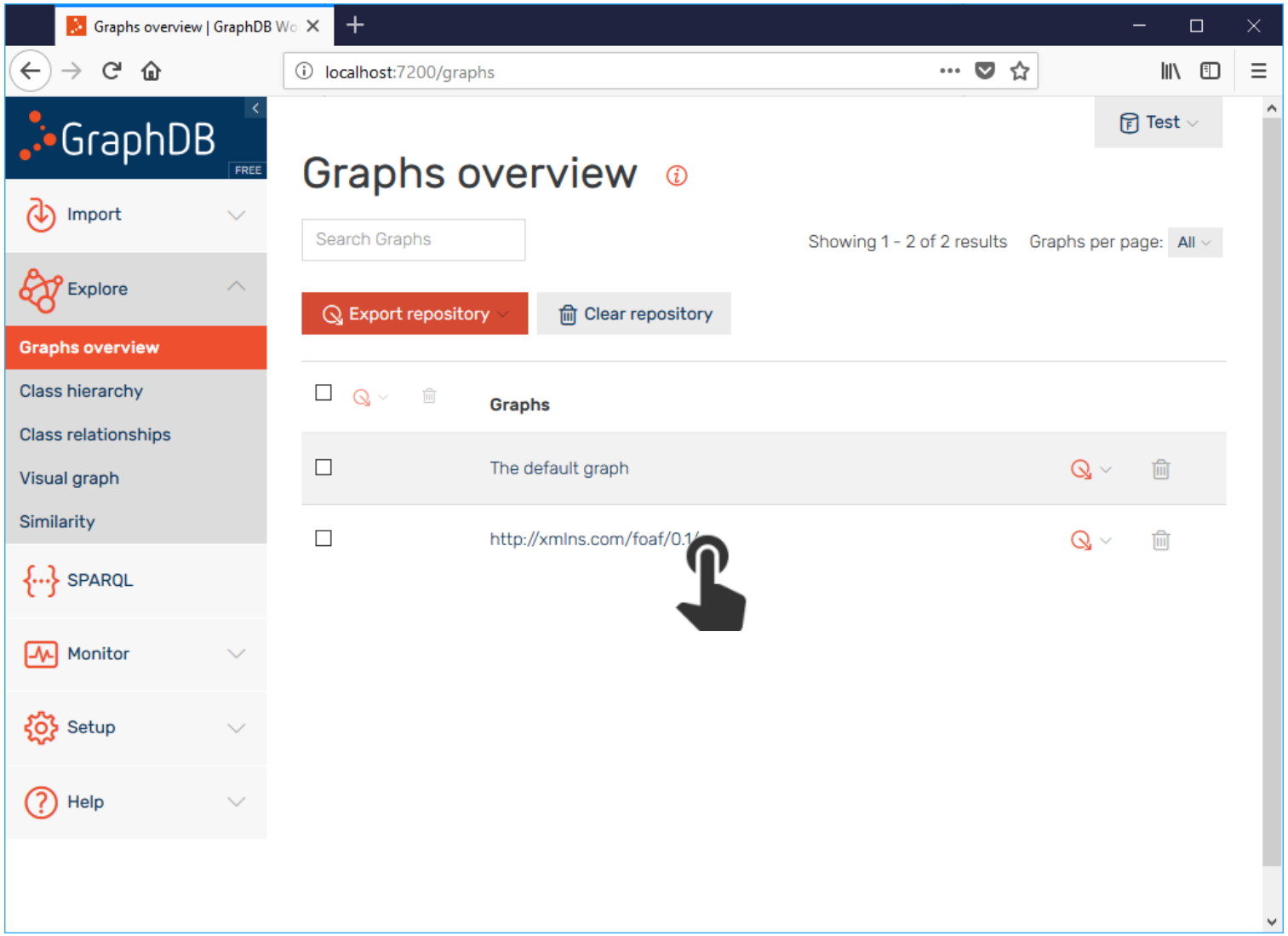
Explore


Import | GraphDB Workbench
localhost:7200/import#
GraphDB
FREE
Import
Explore
SPARQL
Monitor
Setup
Help
Test
Import
User data
Server files
Help
Upload RDF files
All RDF formats, up to 200 MB
Get RDF data from a URL
All RDF formats
Import RDF text snippet
Type or paste RDF data
Type to filter
http://xmlns.com/foaf/0.1/
Imported successfully in less than a second.
Import

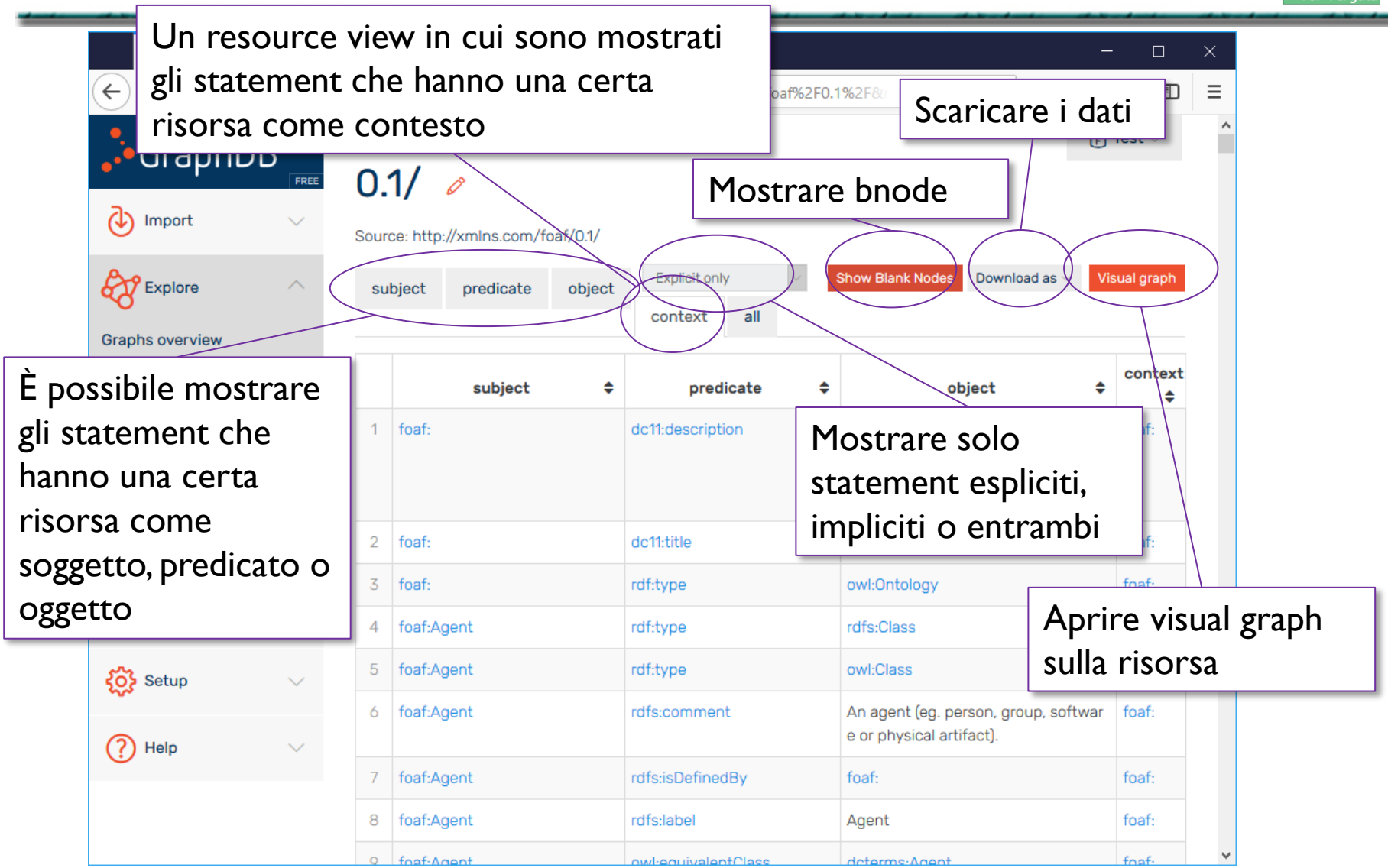
Explore – Graph Overview (1/2)



Explore – Graph Overview (2/2)



Explore – Resource view (1/2)



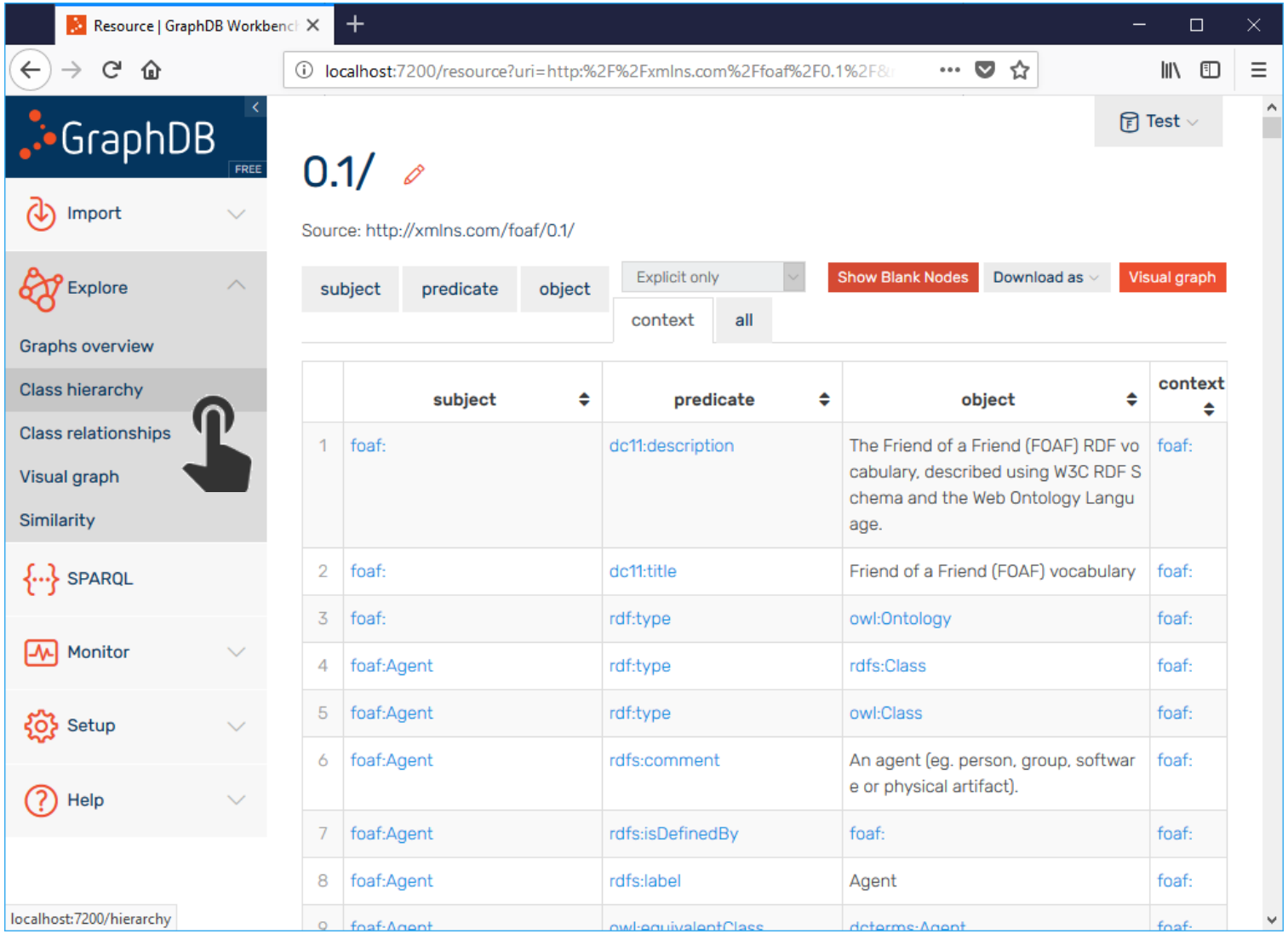
Explore – Resource view (2/2)

Il *resource view* è una visualizzazione ad elevata risoluzione, utile per vedere nel dettaglio:

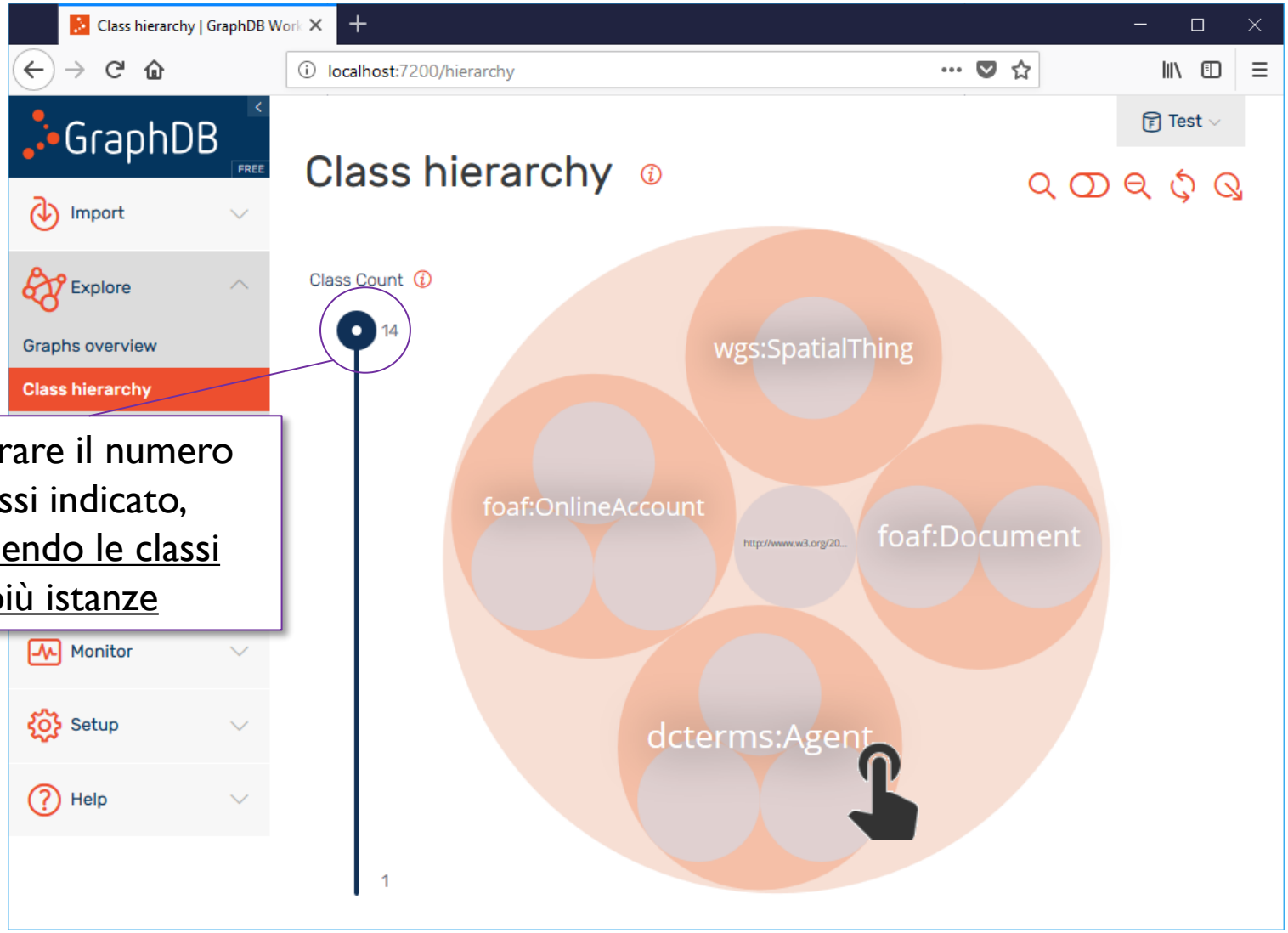
- Ciò che si conosce su una risorsa (*come soggetto*)
- Riferimenti ad una risorsa (*come oggetto*)
- Usi di una risorsa *come predicato*

Può anche essere usato per *editare gli statement* e *definire nuove risorse*.

Explore – Class Hierarchy (1/5)

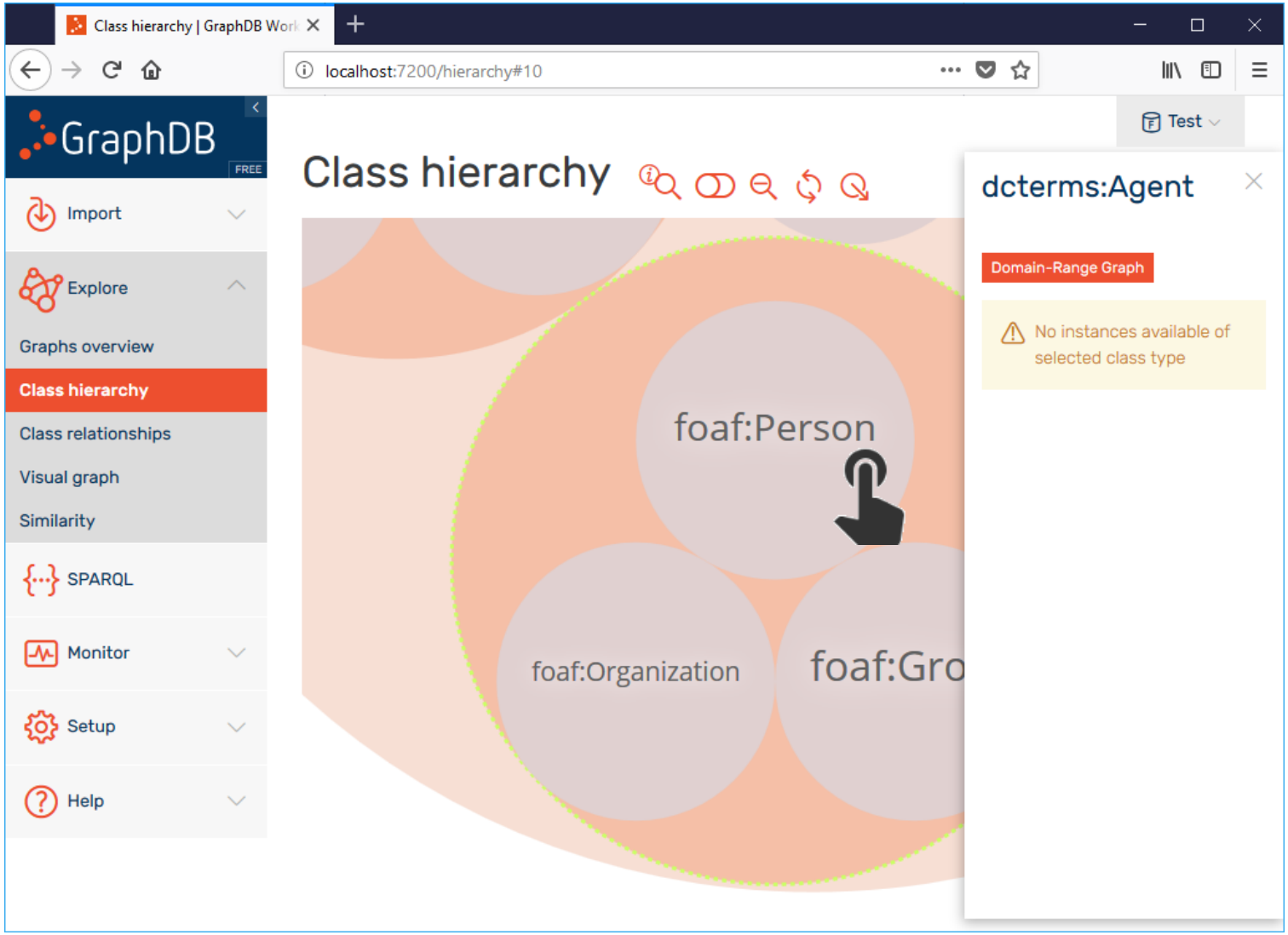


Explore – Class Hierarchy (2/5)

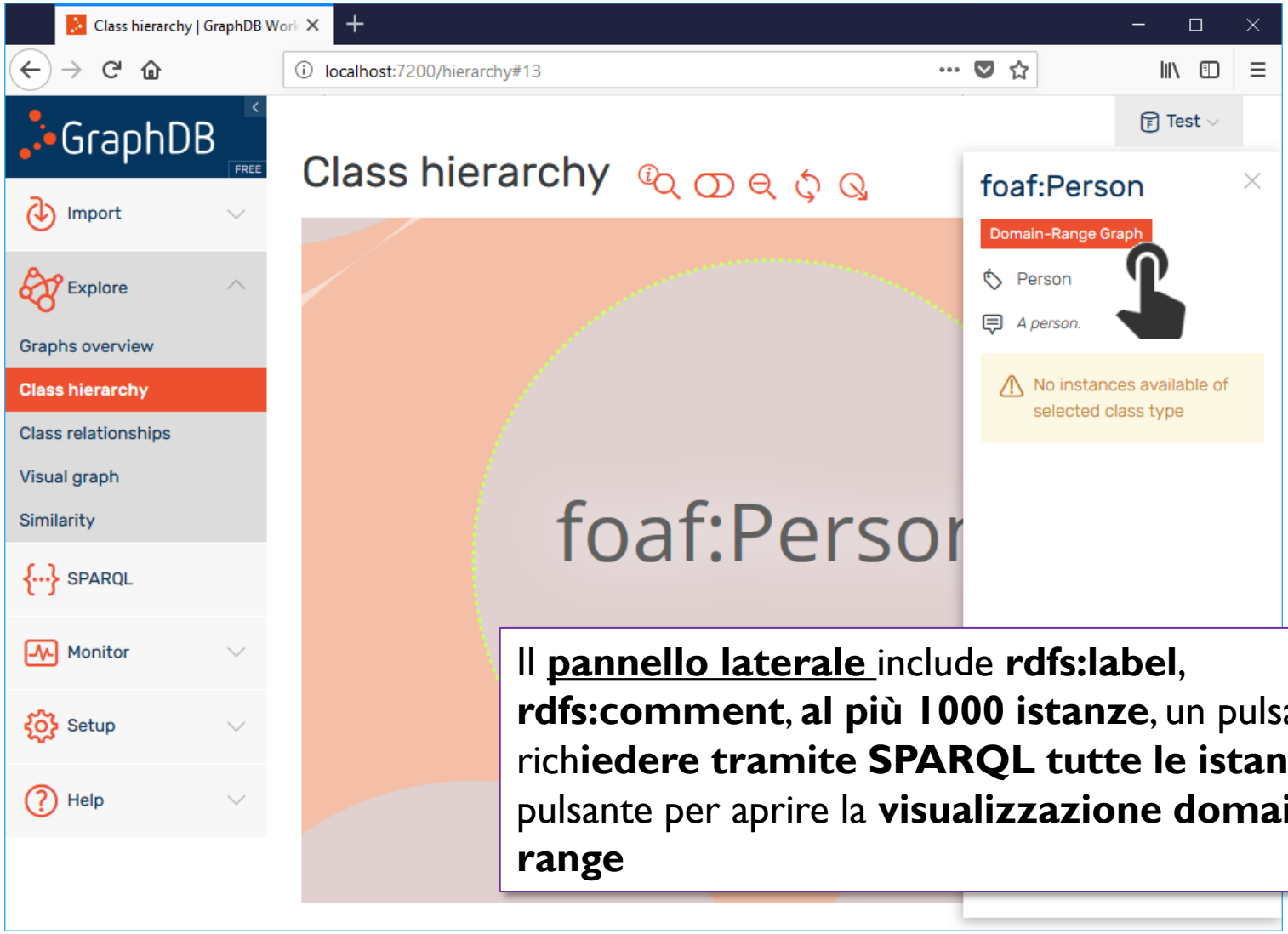


Mostrare il numero di classi indicato, scegliendo le classi con più istanze

Explore – Class Hierarchy (3/5)



Explore – Class Hierarchy (4/5)



Il **pannello laterale** include **rdfs:label**, **rdfs:comment**, **al più 1000 istanze**, un pulsate per richiedere **tramite SPARQL tutte le istanze**, ed il pulsante per aprire la **visualizzazione domain-range**

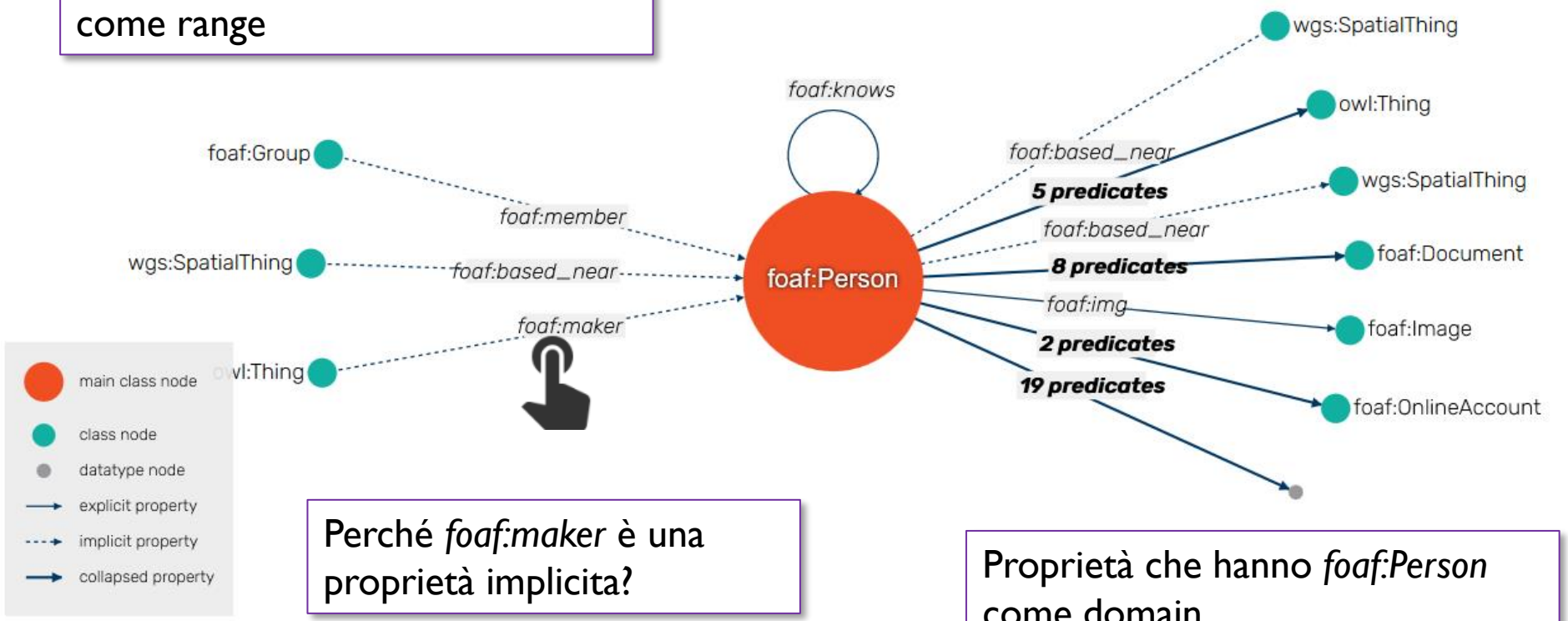
Explore – Class Hierarchy (5/5)

La visualizzazione *class hierarchy* ci permette di visualizzare ed esplorare la gerarchia delle classi.

Ci permette di limitare il numero di classi visualizzate, dando la priorità a quelle con più istanze.

Explore – Domain-Range Graph (1/3)

Proprietà che hanno *foaf:Person* come range

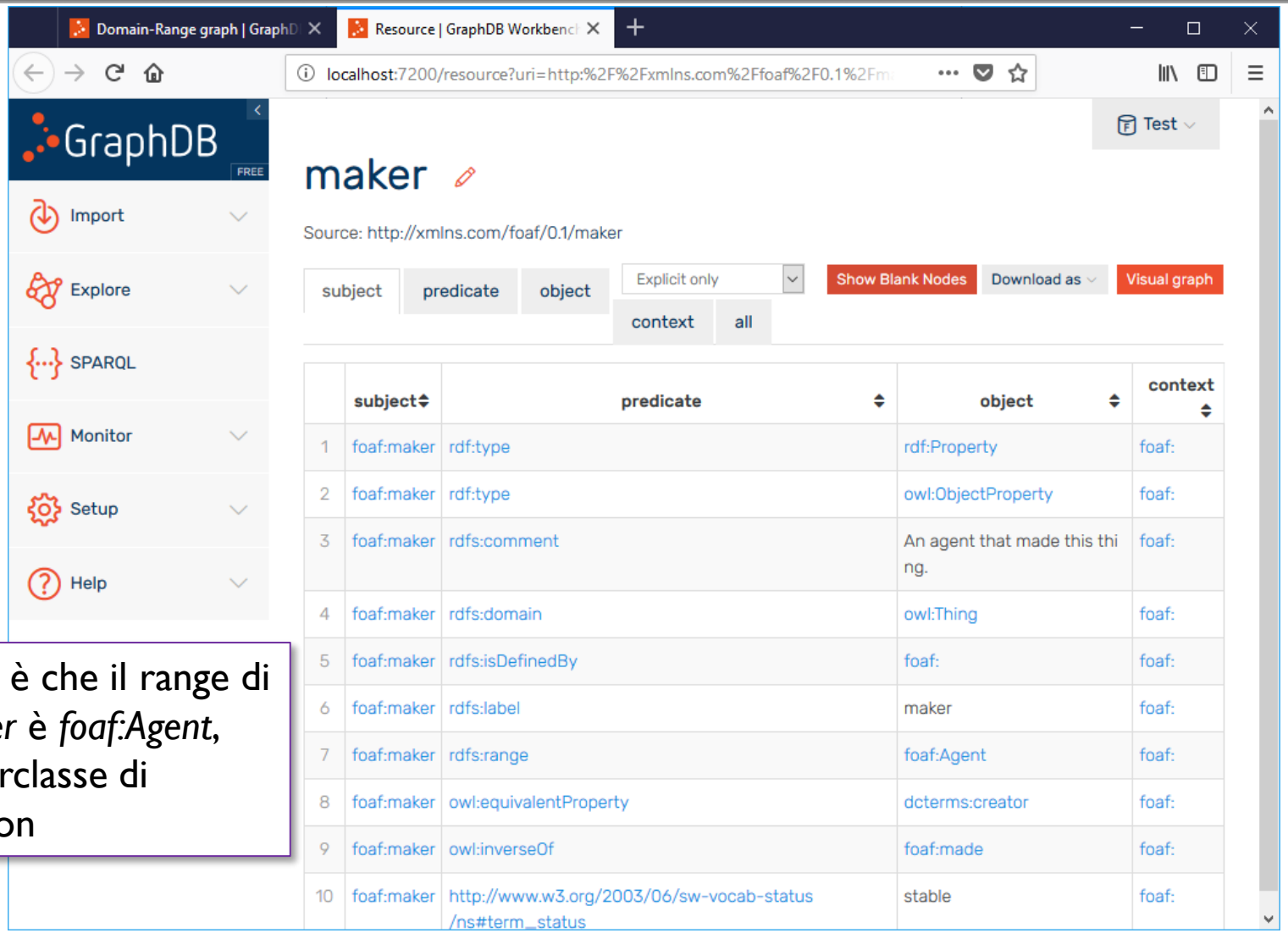


Perché *foaf:maker* è una proprietà implicita?

Proprietà che hanno *foaf:Person* come domain.

Gli elementi in grassetto indicano gruppi di proprietà con una certa combinazione di domain e range

Explore – Domain-Range Graph (2/3)



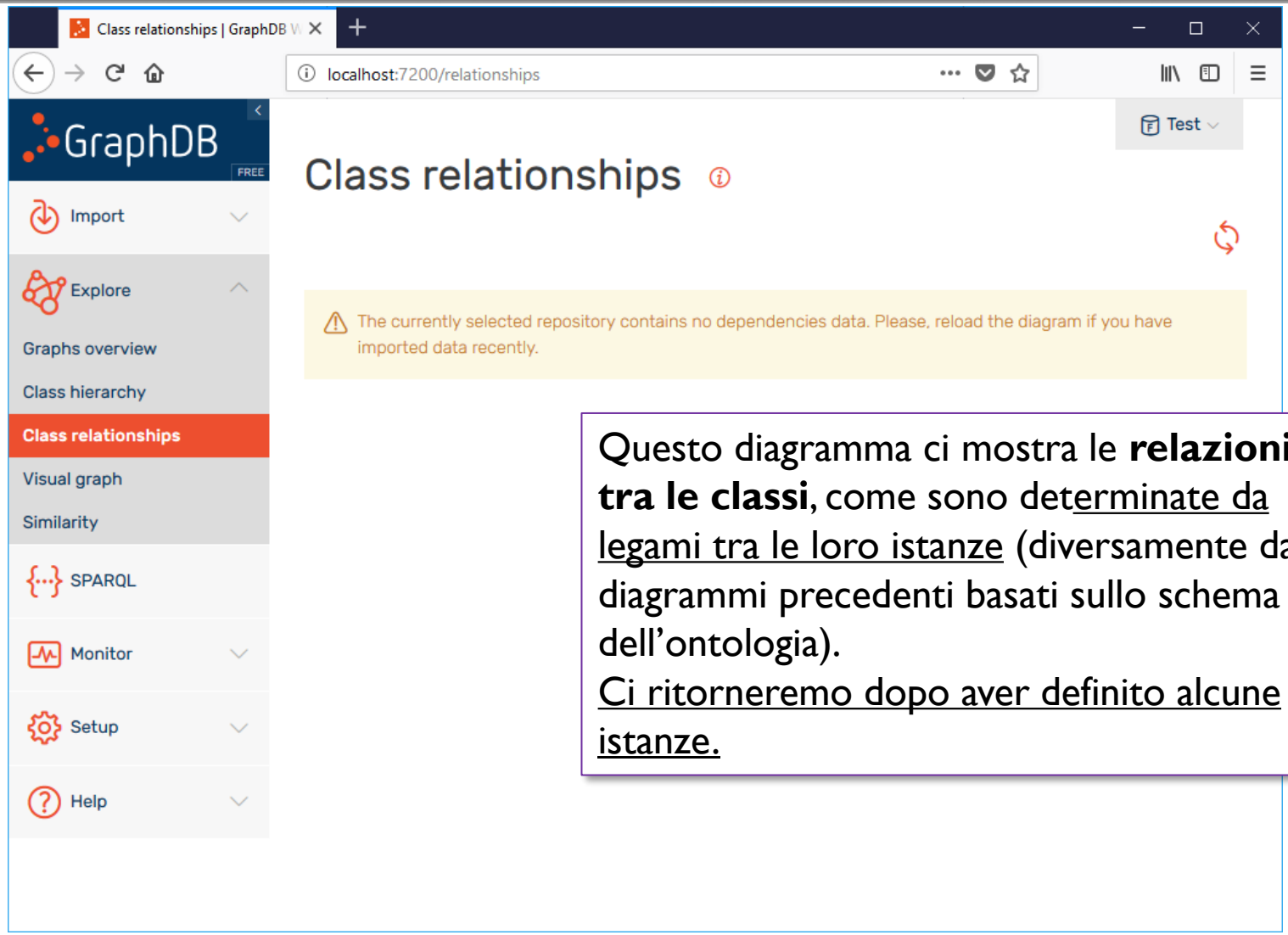
Il motivo è che il range di *foaf:maker* è *foaf:Agent*, una superclasse di foaf:Person

Explore – Domain-Range Graph (3/3)

Il *domain-range graph* ci permette di analizzare le relazioni tra le classi (a livello assiomatico).

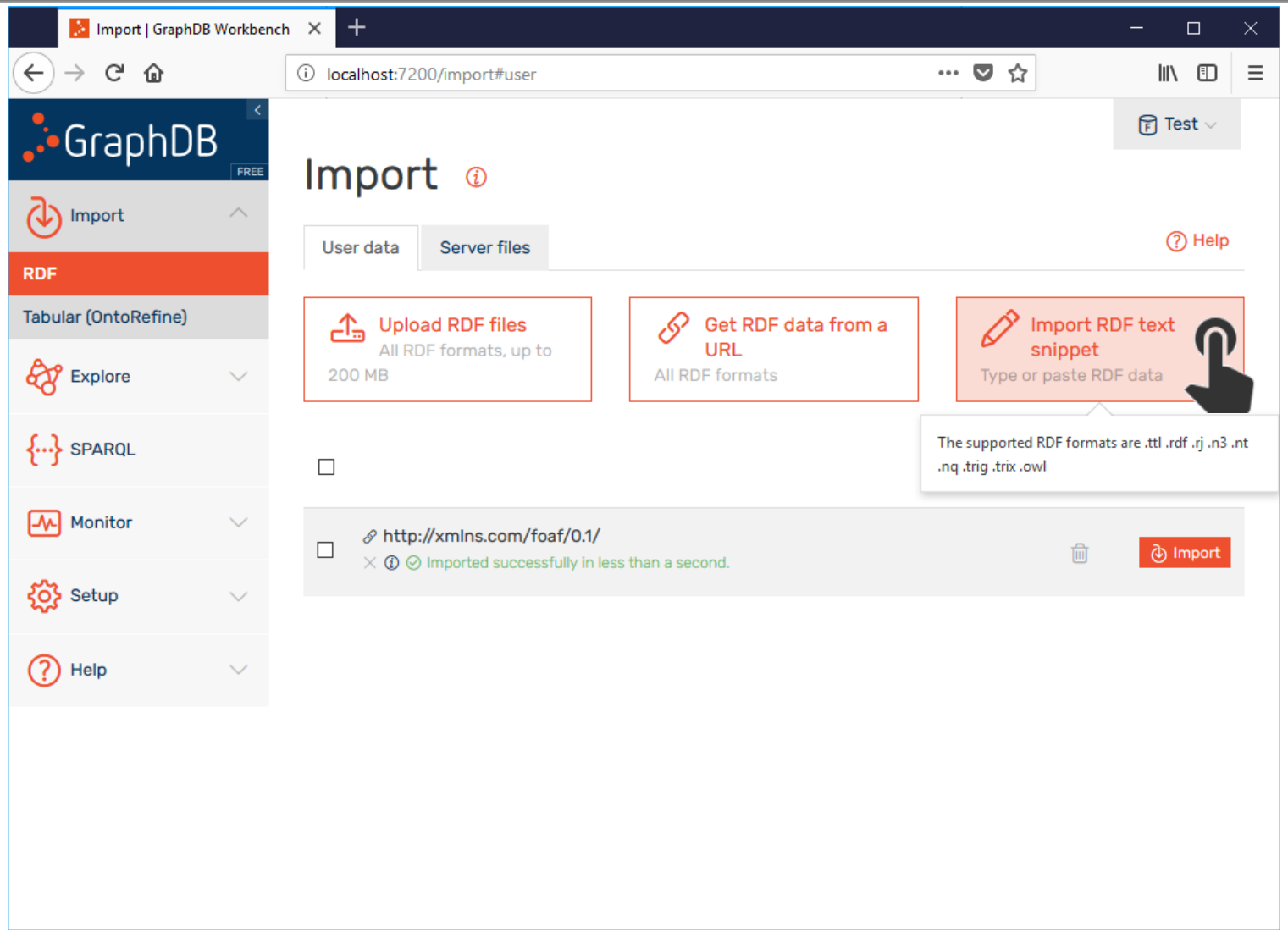
Il concetto di proprietà implicita ci aiuta a gestire meglio l'interazione tra dichiarazioni di domain/range e la gerarchia delle classi.

Explore – Graph Relationships – senza istanze

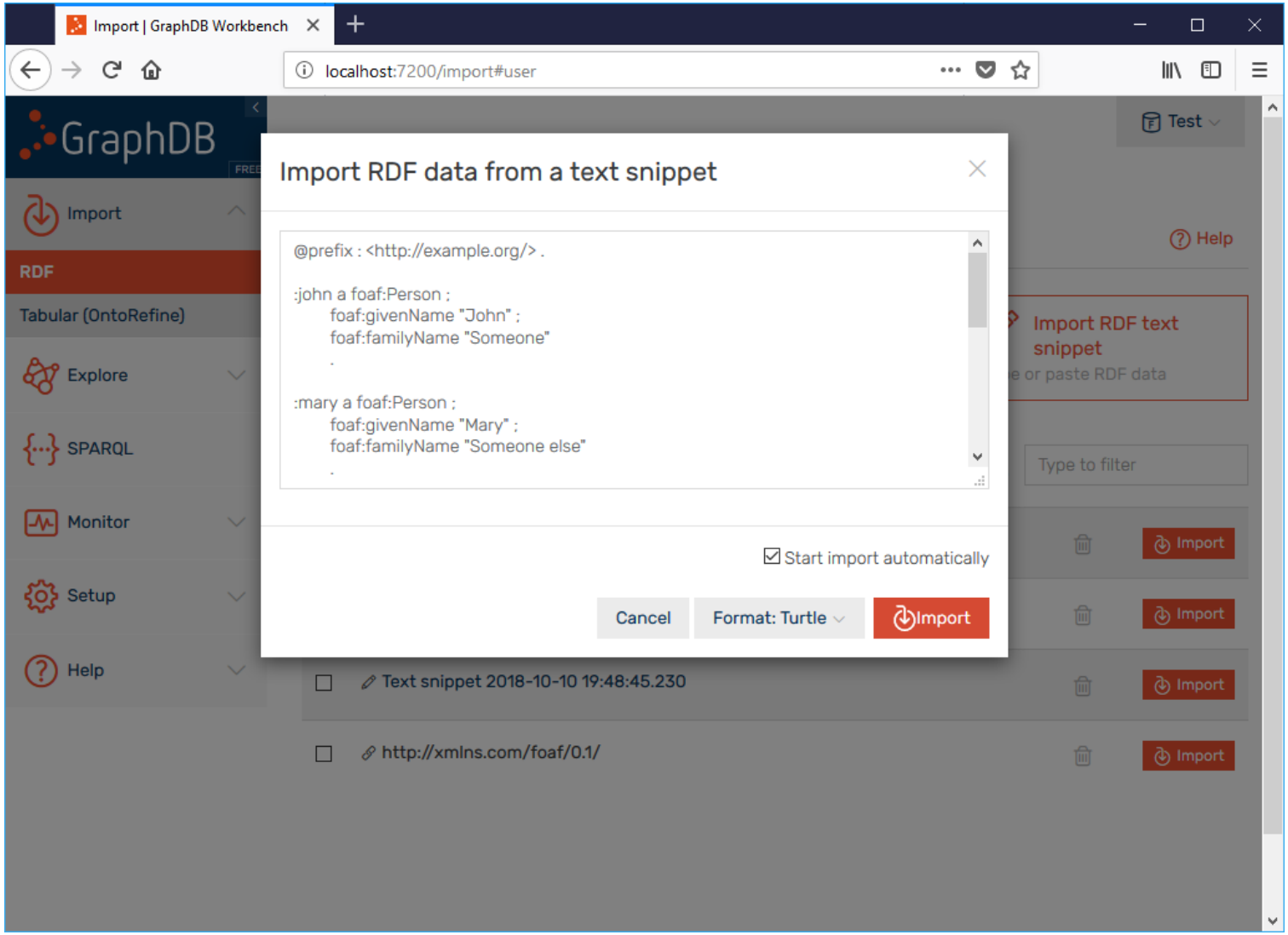


Questo diagramma ci mostra le **relazioni tra le classi**, come sono determinate da legami tra le loro istanze (diversamente dai diagrammi precedenti basati sullo schema dell'ontologia).
Ci ritorneremo dopo aver definito alcune istanze.

Import RDF text snippet (1/4)



Import RDF text snippet (2/4)



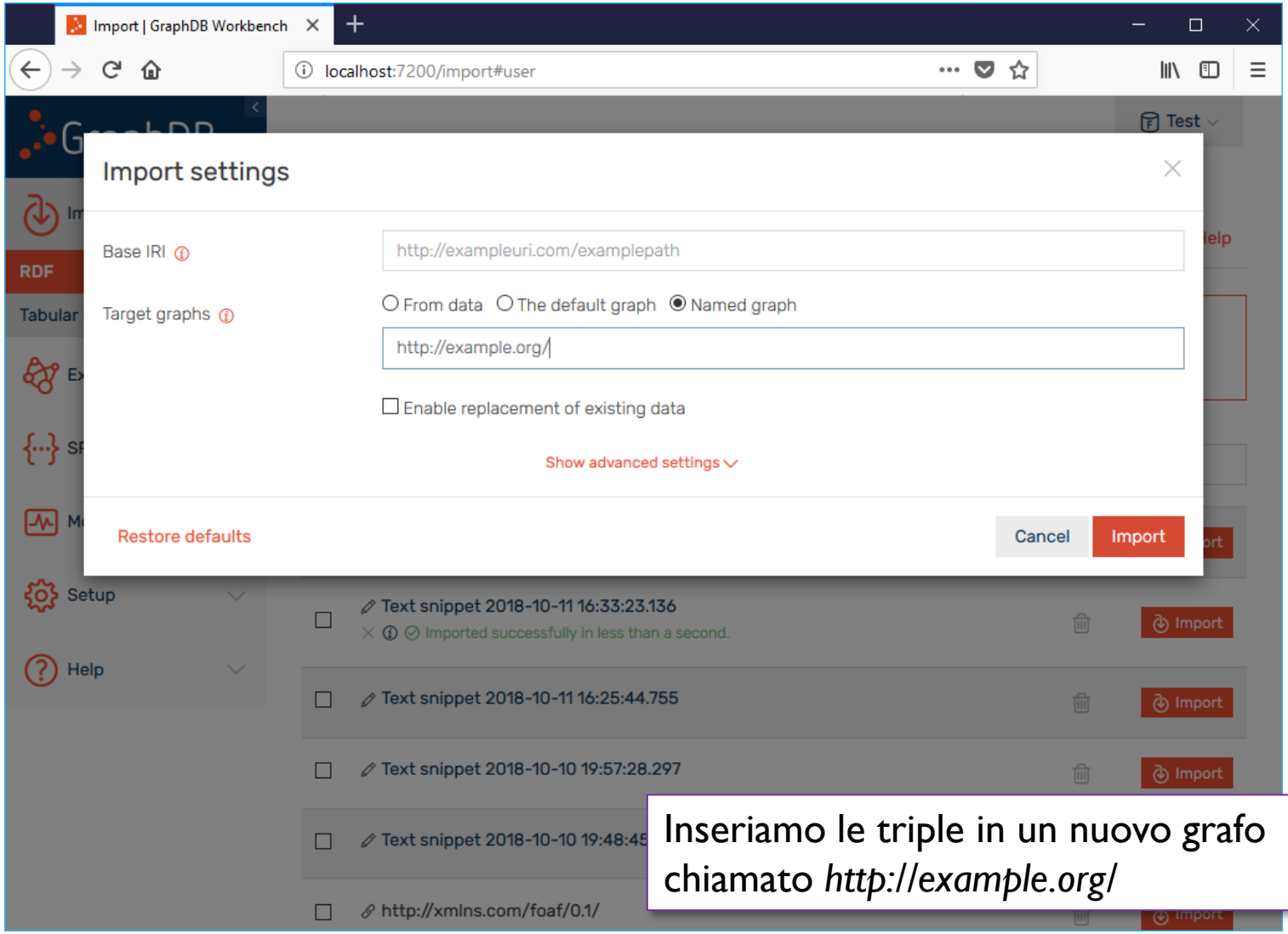
Import RDF text snippet (3/4)

```
@prefix : <http://example.org/> .

:john a foaf:Person ;
   foaf:givenName "John" ;
   foaf:familyName "Someone"
   .
:mary a foaf:Person ;
   foaf:givenName "Mary" ;
   foaf:familyName "Someone else"
   .
:philip a foaf:Person ;
   foaf:givenName "Philip" ;
   foaf:familyName "The One"
   .
```

```
:alice a foaf:Person ;
   foaf:givenName "Alice" ;
   foaf:familyName "Cetara"
   .

:john foaf:knows :mary, :philip, :alice .
:mary foaf:knows :john, :philip .
:philip foaf:knows :john, :mary .
:alice foaf:knows :john .
```

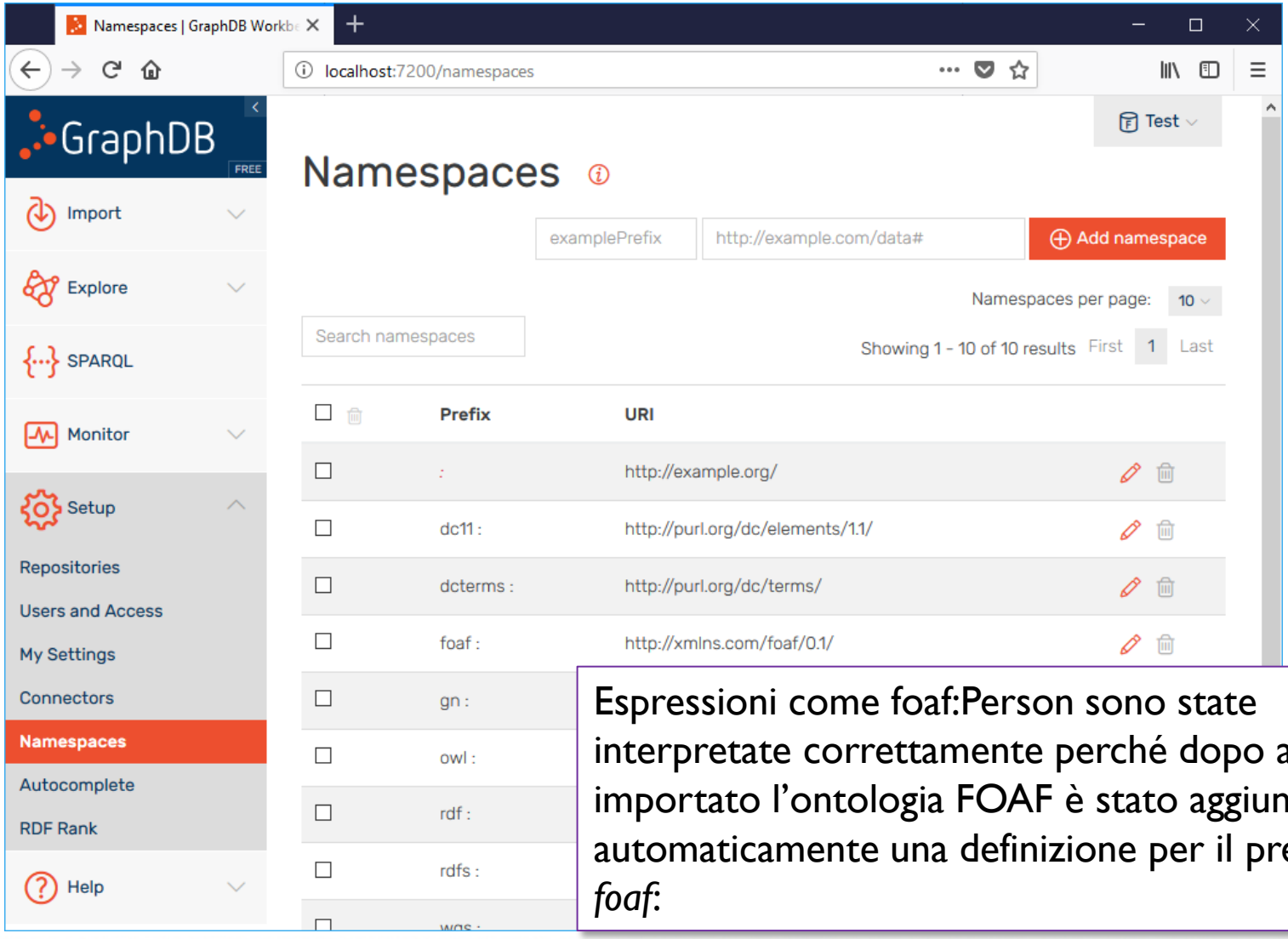
Definiamo alcune istanze di fantasia della classe foaf:Person, ed immaginiamo che alcune si conoscano (*foaf:knows*) tra di loro.

Import RDF text snippet (4/4)



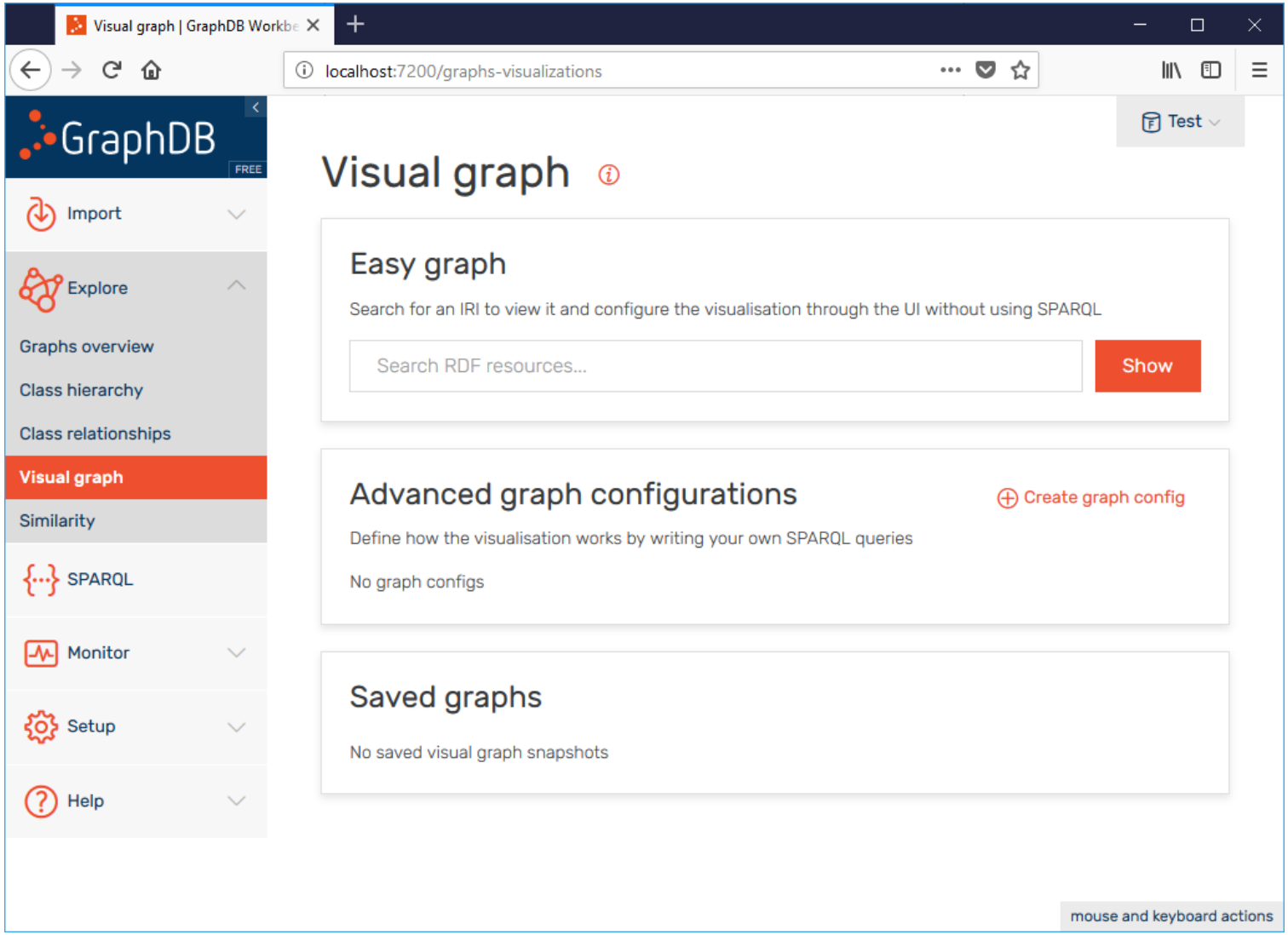
Inseriamo le triple in un nuovo grafo chiamato *http://example.org/*

Setup - Namespaces

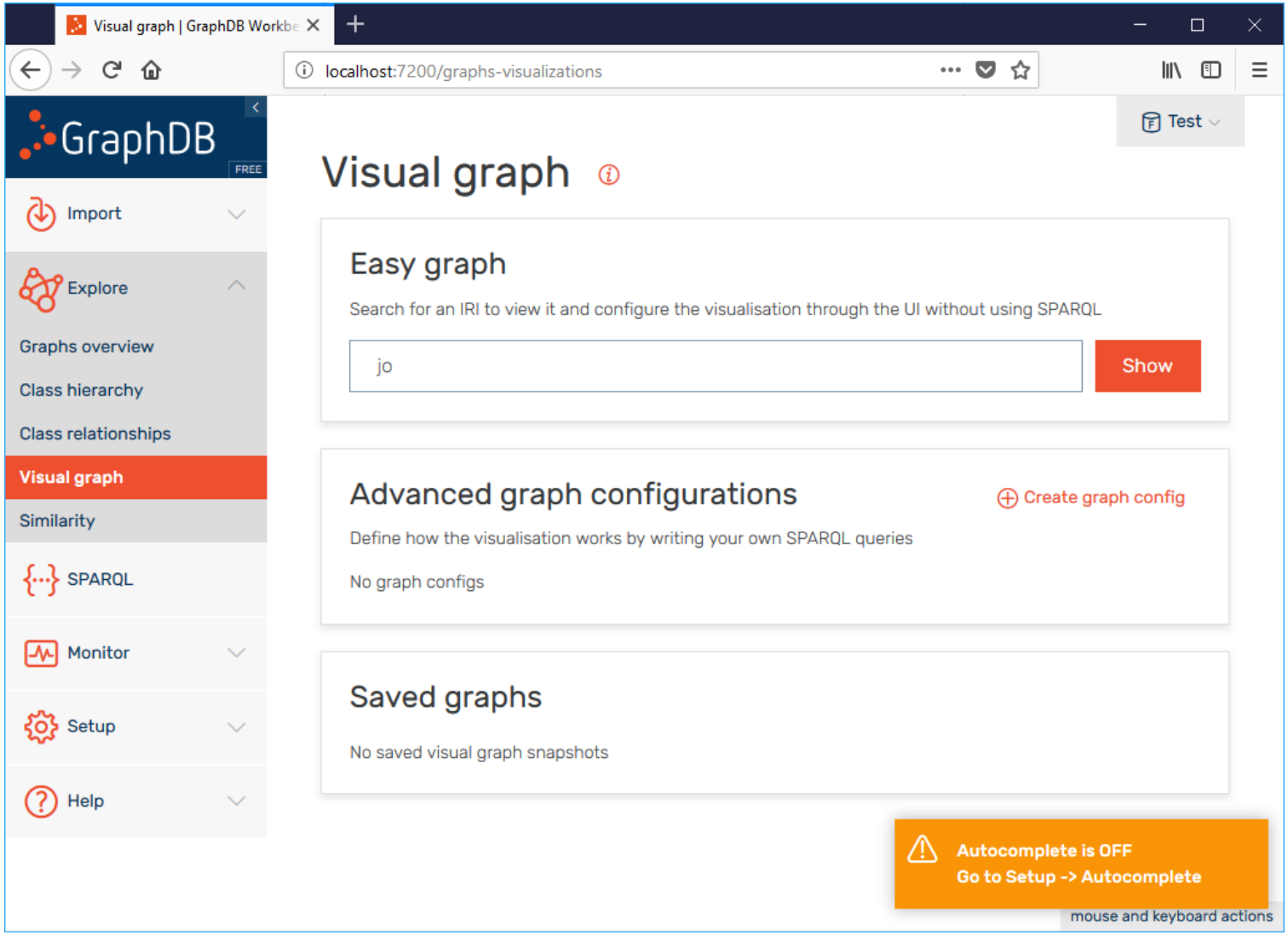


Espressioni come foaf:Person sono state interpretate correttamente perché dopo aver importato l'ontologia FOAF è stato aggiunta automaticamente una definizione per il prefisso *foaf*:

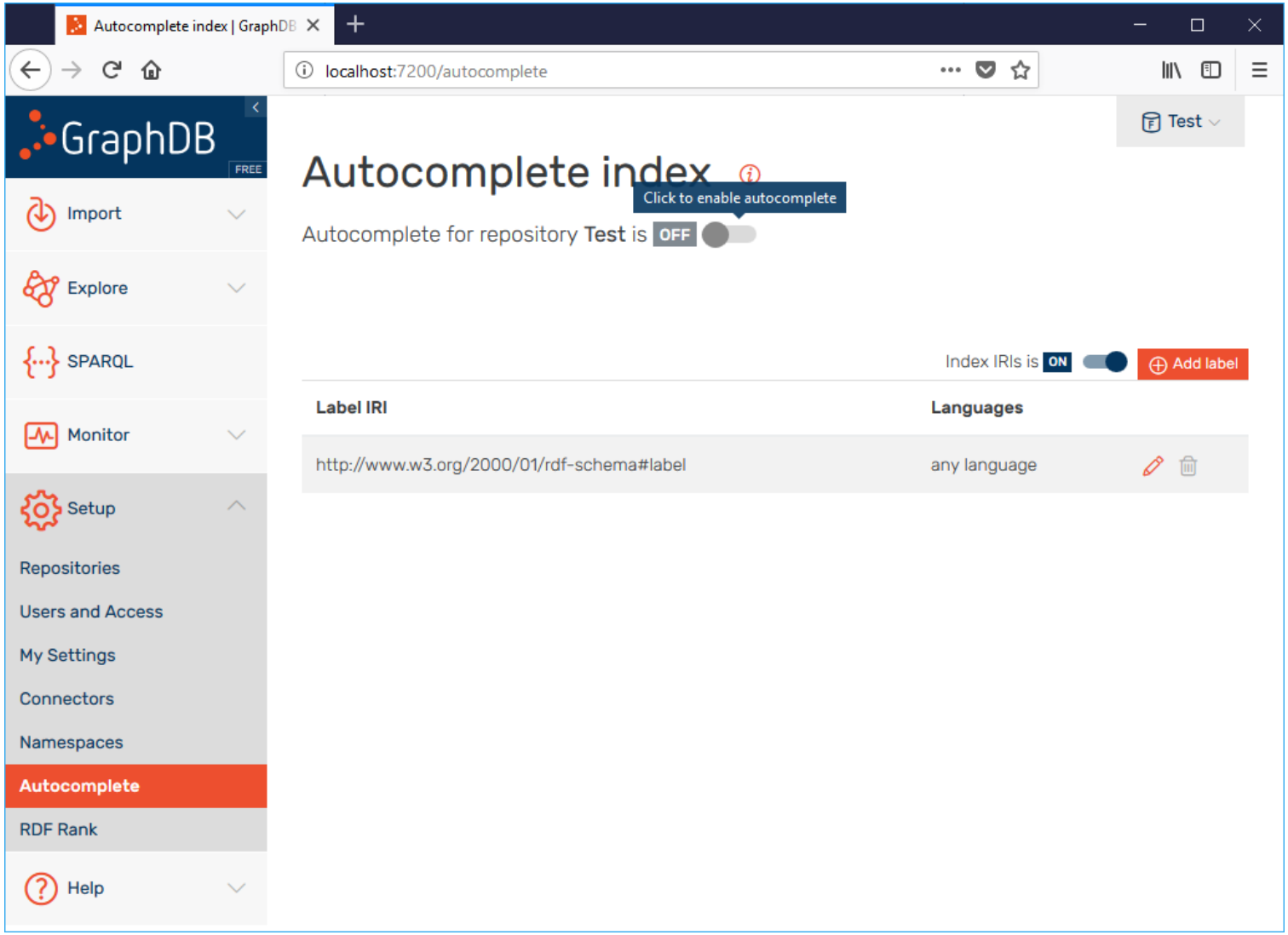
Visual graph (1/10)



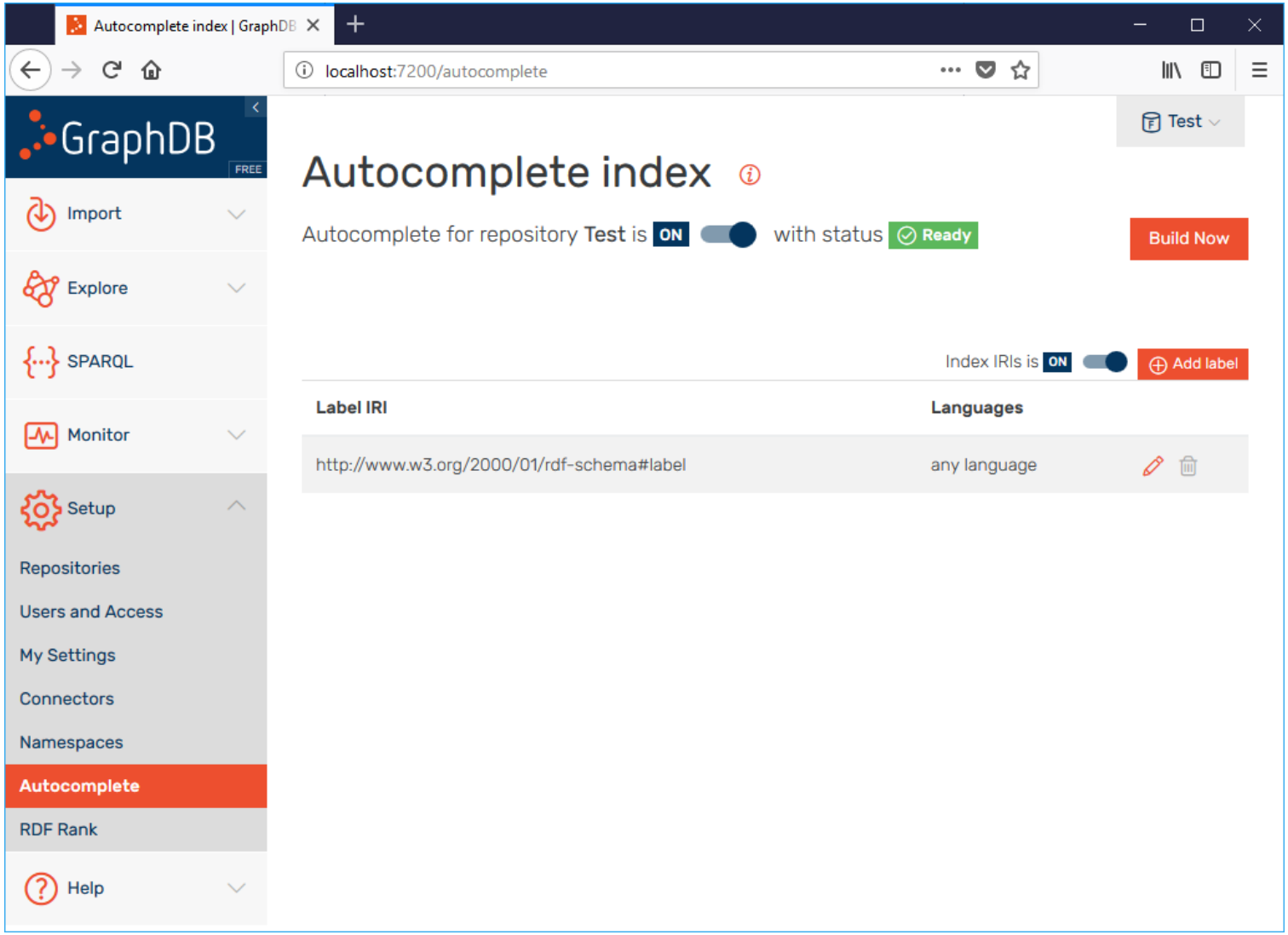
Visual graph (2/10)



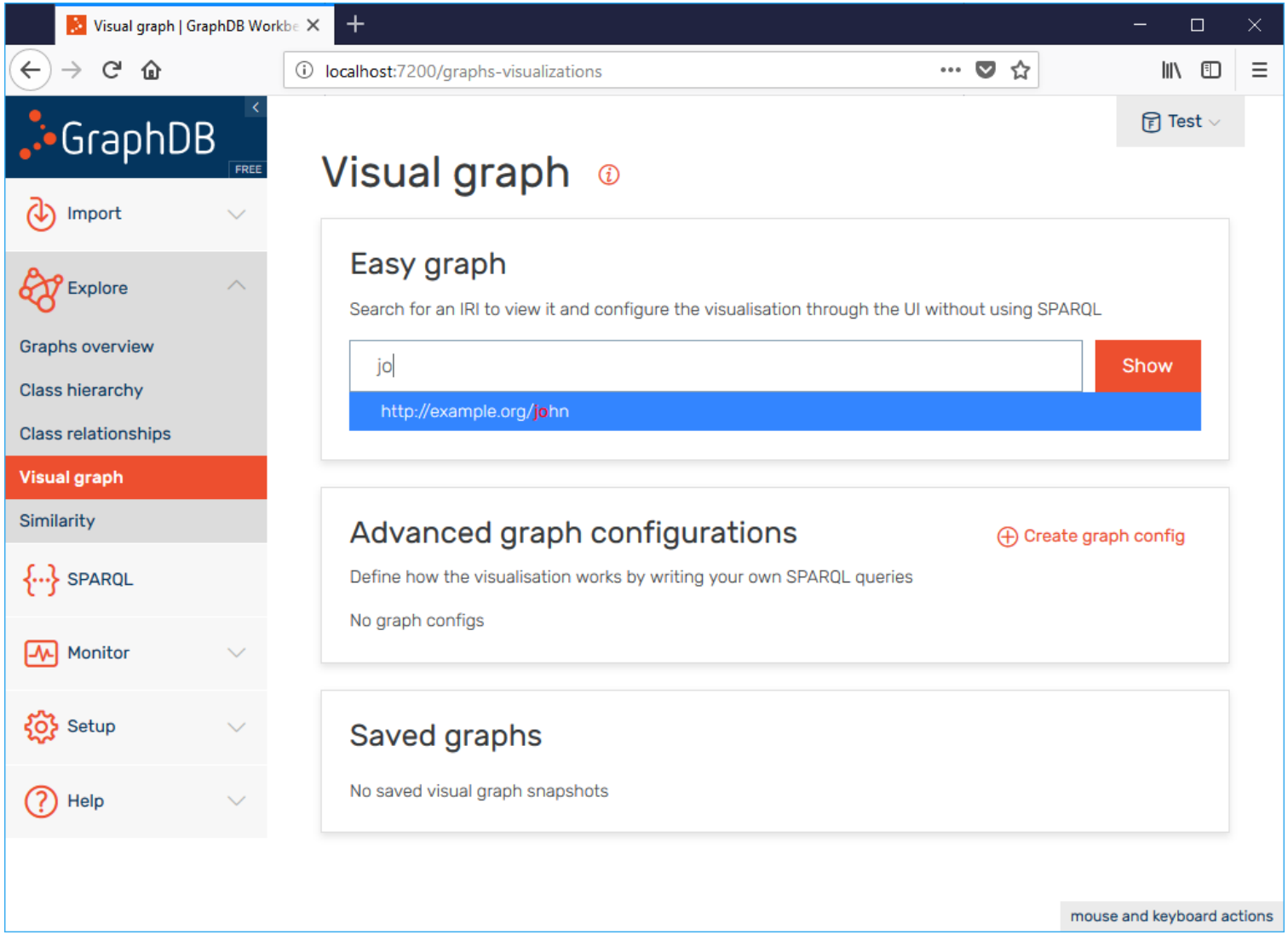
Autocompletamento (1/2)



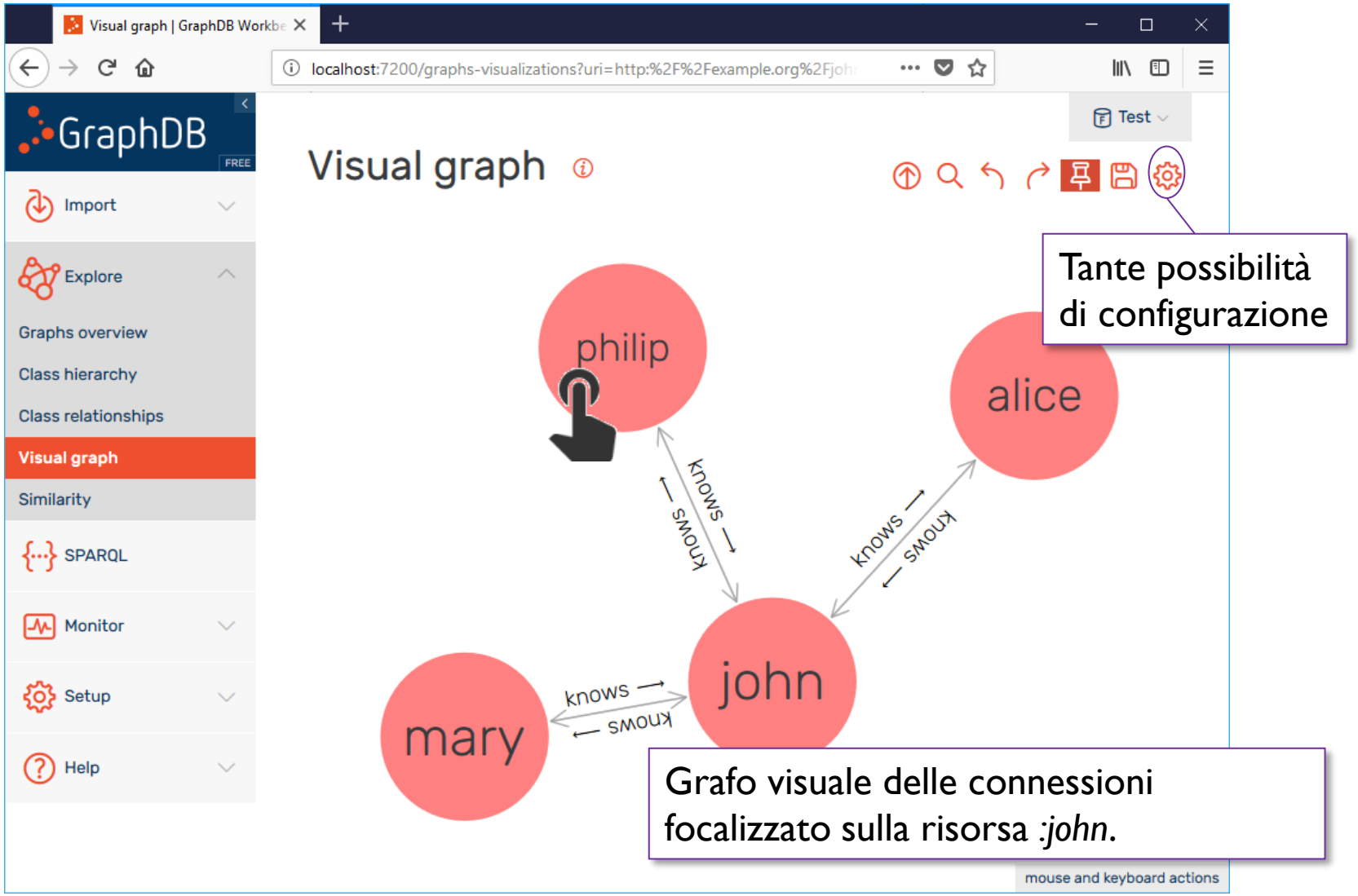
Autocompletamento (2/2)



Visual graph (3/10)



Visual graph (4/10)



Visual graph (5/10)

Il *visual graph* ci permette di visualizzare ed esplorare le relazioni tra le risorse.

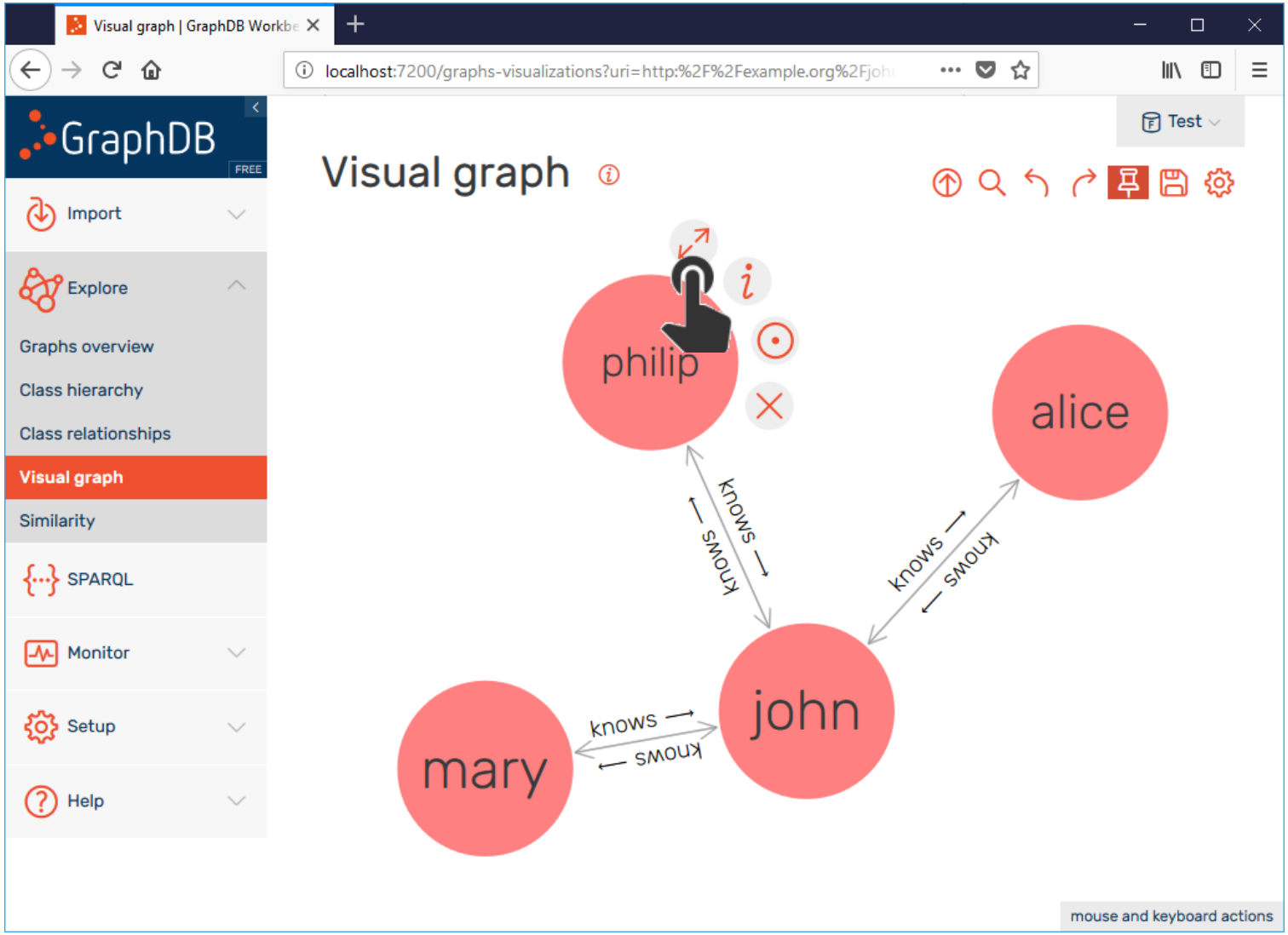
Per default, mostra al più 20 connessioni ad altre risorse ordinate per RDF Rank (se calcolato)

→ concentrarsi sulle connessioni più importanti (o meglio, alle risorse più importanti)

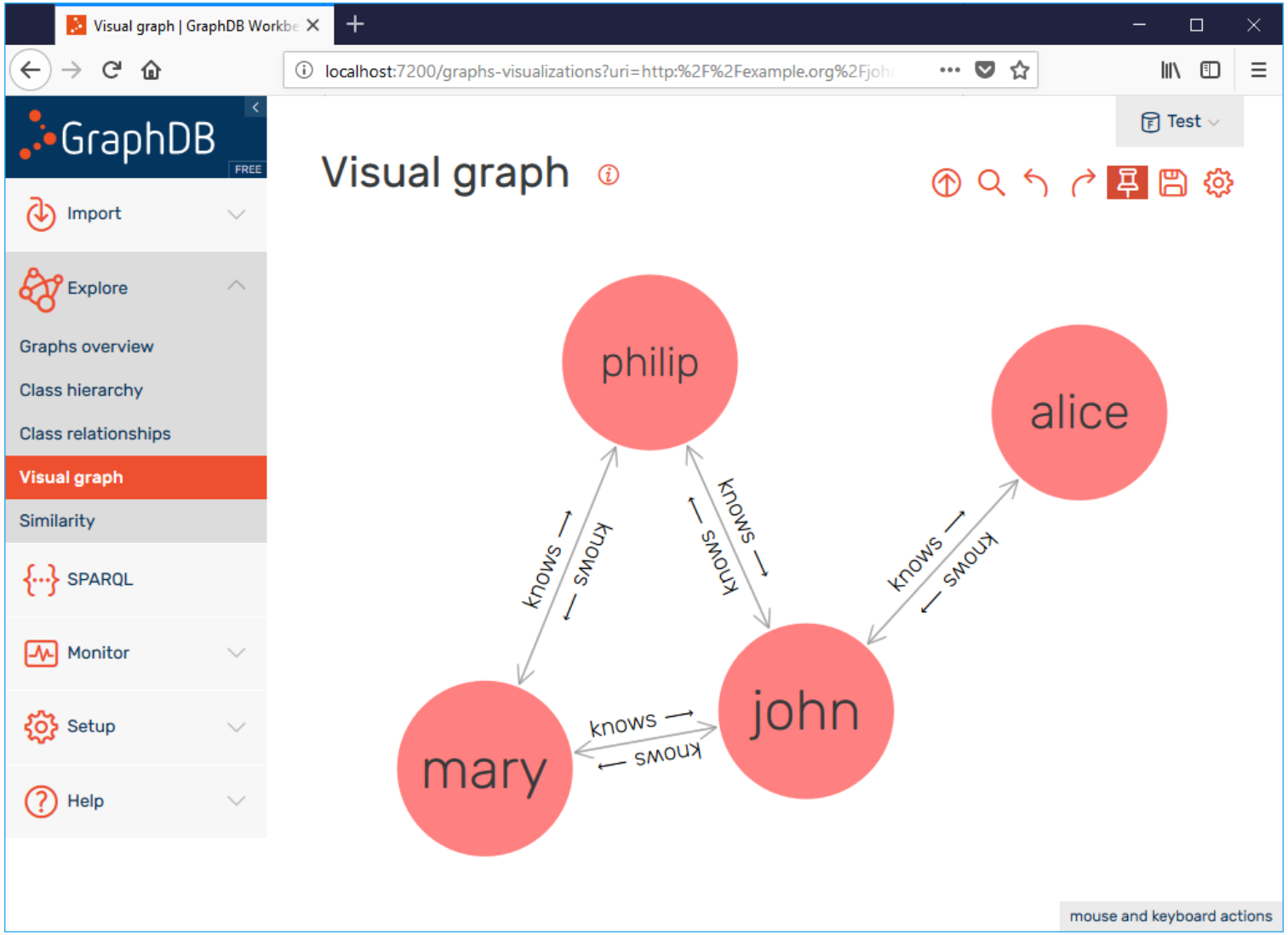
I nodi associati alle diverse risorse hanno un colore determinato dalla classe di appartenenza, ed hanno una dimensione che esprime il loro RDF Rank (se calcolato)

→ Discriminare l'importanza di una risorsa

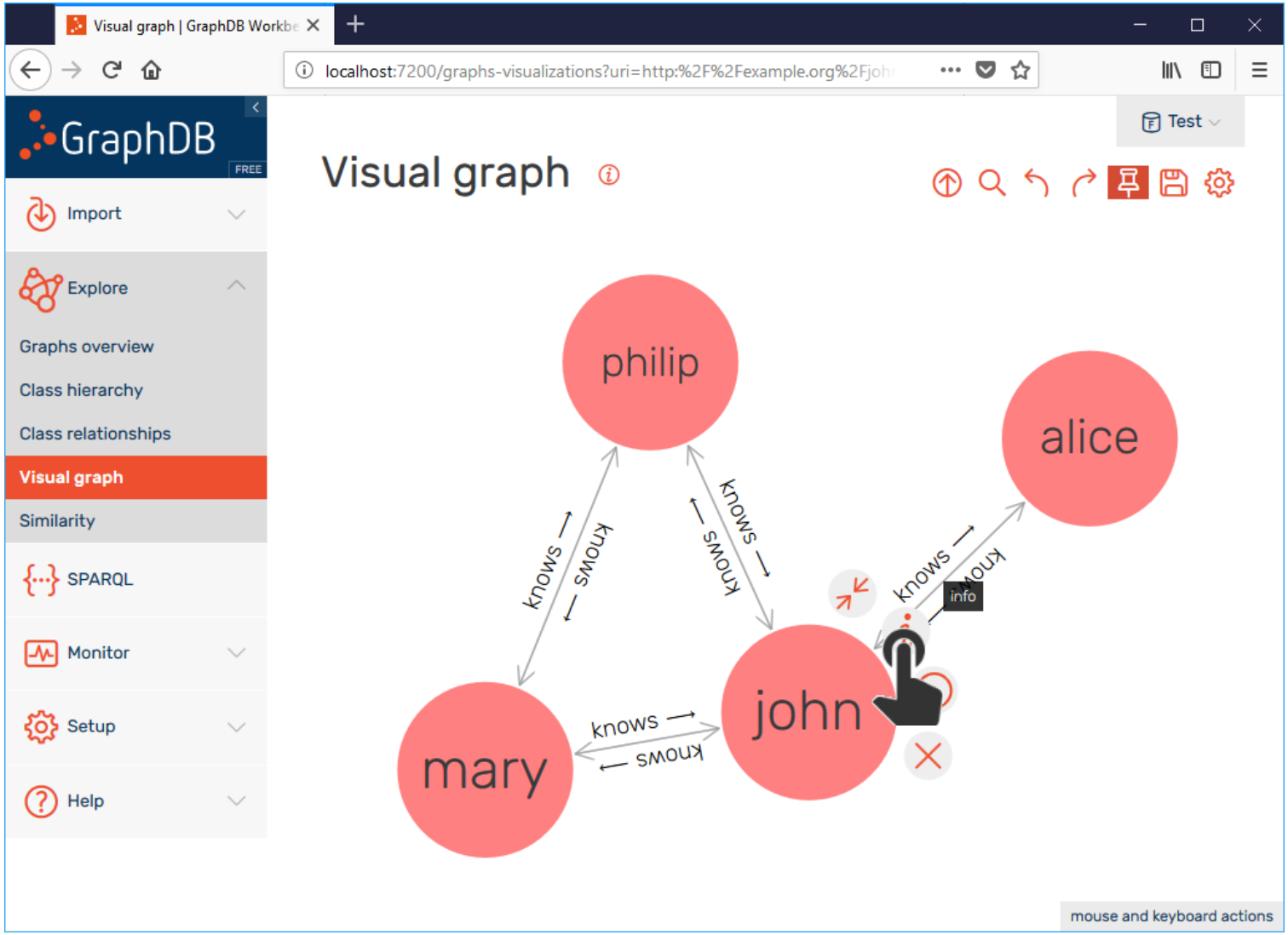
Visual graph (6/10)



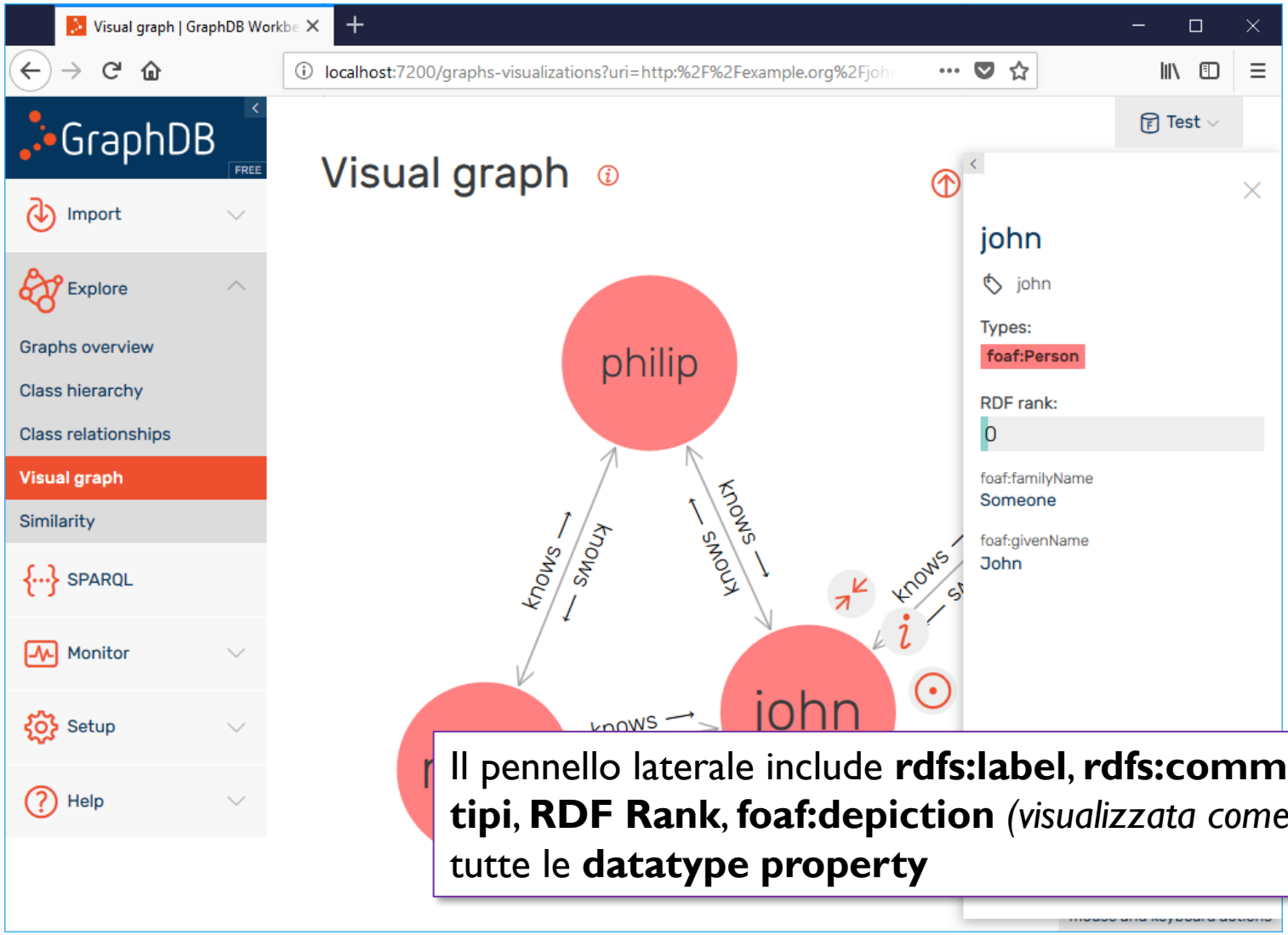
Visual graph (7/10)



Visual graph (8/10)

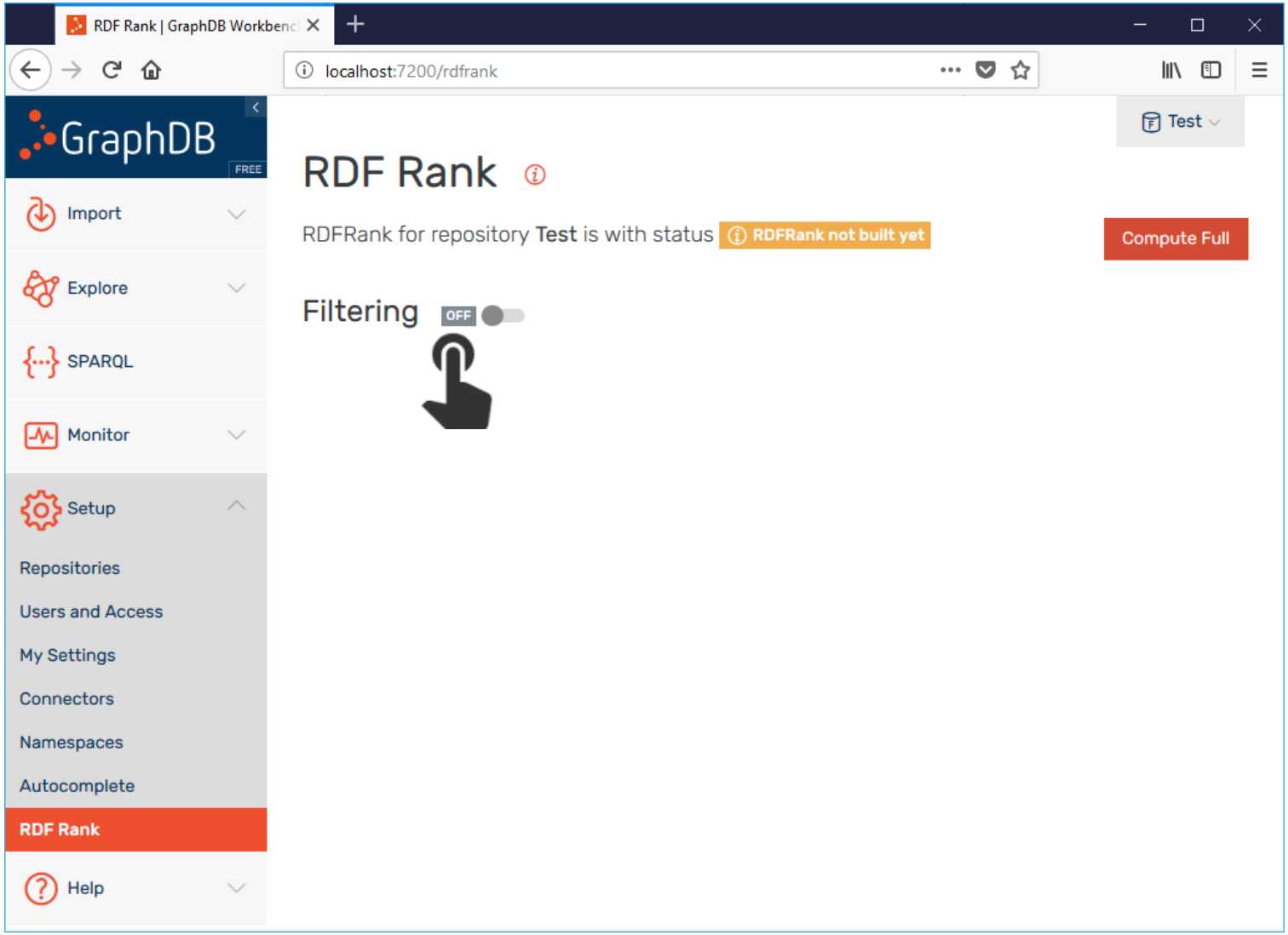


Visual graph (9/10)

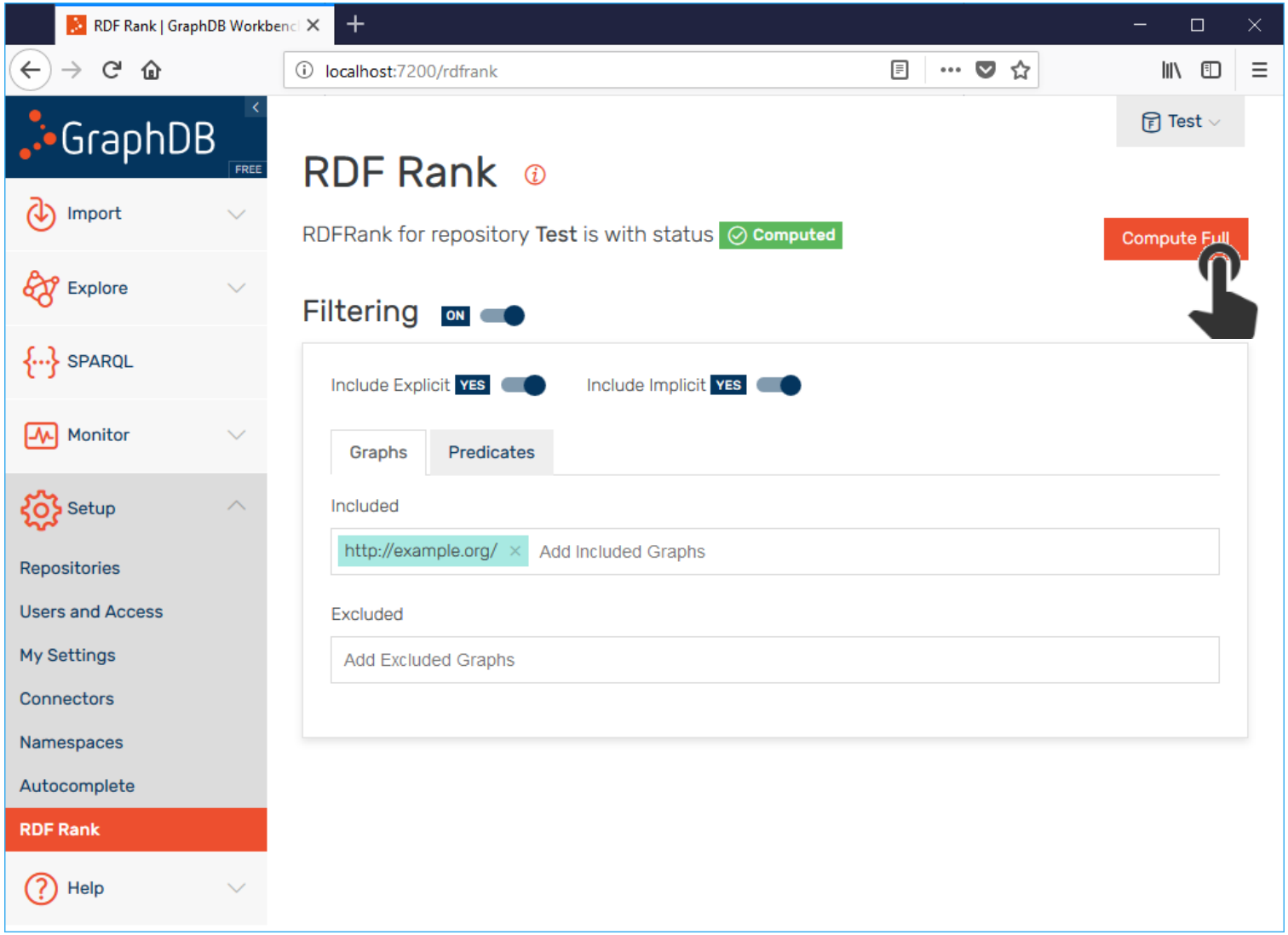


Il pennello laterale include **rdfs:label**, **rdfs:comment**, i **tipi**, **RDF Rank**, **foaf:depiction** *(visualizzata come figura)*, tutte le **datatype property**

RDF Rank (1/4)


RDF Rank | GraphDB Workbench
localhost:7200/rdfrank
GraphDB
FREE
Import
Explore
SPARQL
Monitor
Setup
Repositories
Users and Access
My Settings
Connectors
Namespaces
Autocomplete
RDF Rank
Help
Test
RDF Rank
RDFRank for repository Test is with status
RDFRank not built yet
Compute Full
Filtering
OFF

RDF Rank (2/4)


RDF Rank | GraphDB Workbench
localhost:7200/rdfrank
GraphDB
FREE
Import
Explore
SPARQL
Monitor
Setup
Repositories
Users and Access
My Settings
Connectors
Namespaces
Autocomplete
RDF Rank
Help
Test
RDF Rank
RDFRank for repository Test is with status Computed
Compute Full
Filtering ON
Include Explicit YES
Include Implicit YES
Graphs
Predicates
Included
http://example.org/
Add Included Graphs
Excluded
Add Excluded Graphs

RDF Rank (3/4)

RDF Rank determina l'importanza di un nodo in un grafo RDF esaminando le sue interconnessioni:

- Molto simile all'algoritmo PageRank utilizzato da Google per ordinare i risultati di una ricerca (assieme a molti altri segnali!)

Il PageRank di una pagina Web è il suo *long-term visit rate* di un utente che naviga sul Web seguendo con equiprobabilità i link uscenti da ciascuna pagina

- Questo processo è modellato come una catena di Markov a tempo discreto. Affinché il long-term visit rate sia definito è necessario che la catena di Markov sia *ergodica*.
- A tale scopo, la catena è modificata in modo che i) giunto a un dead end (pagina senza link uscenti), l'utente salti con equiprobabilità in una qualsiasi pagina del web, ii) in ogni istante, c'è una probabilità (teleportation rate) che l'utente salti ad una qualunque pagina Web (teleporting) anziché seguire uno dei link uscenti

RDF Rank (4/4)

Dietro le quinte RDF Rank viene **configurato inviando opportuni UPDATE SPARQL** al repository.

In aggiunta al **ricomputo completo**, viene offerta la possibilità di un **ricomputo incrementale**:

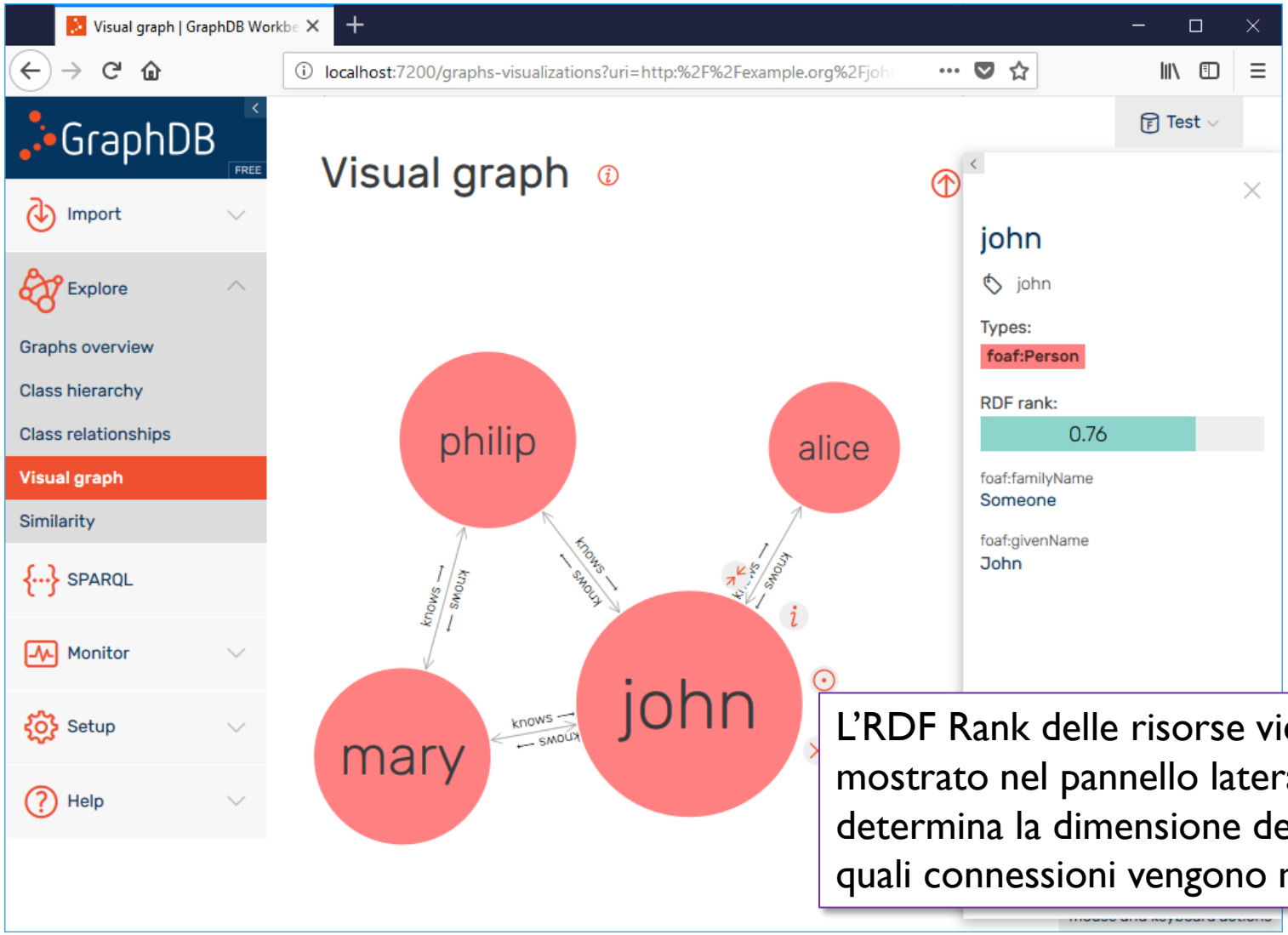
Utilizza un algoritmo può veloce rispetto a quello usato nel ricomputo completo dell'RDF Rank, che introduce però alcune approssimazioni. In un contesto di aggiornamenti frequenti alla base di conoscenza, l'algoritmo incrementale permette un aggiornamento più efficiente del rango delle risorse.

Utilizzo di algoritmi per grafi

RDF Rank ci mostra che è possibile applicare comuni algoritmi per grafi ai dati RDF.

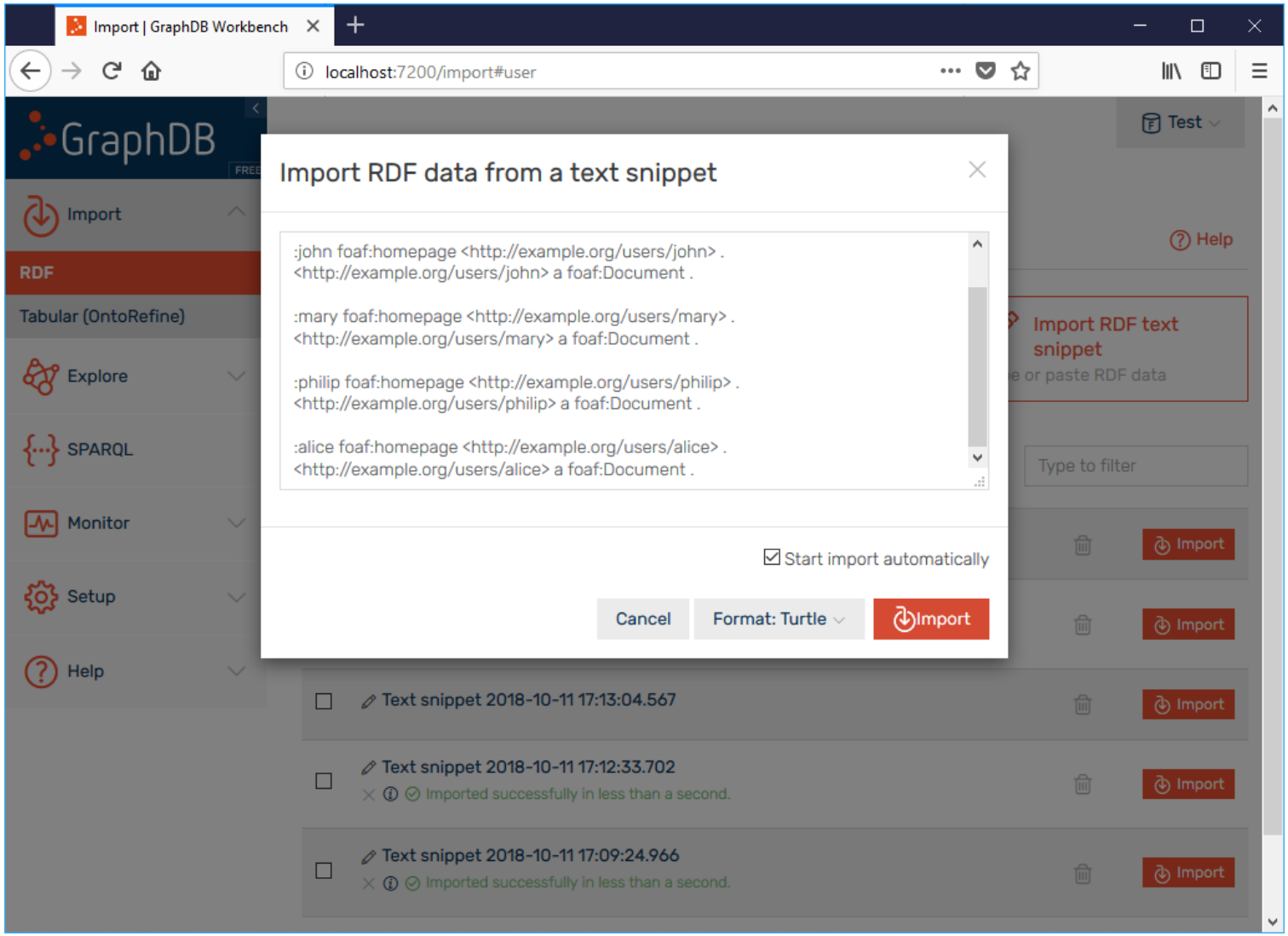
In altre parole, RDF può non solo essere visualizzato come un grafo, ma anche elaborato come un grafo.

Visual graph (10/10)



L'RDF Rank delle risorse viene mostrato nel pannello laterale, determina la dimensione dei nodi e quali connessioni vengono mostrate.

Rendiamo più interessante la descrizione delle istanze (1/3)

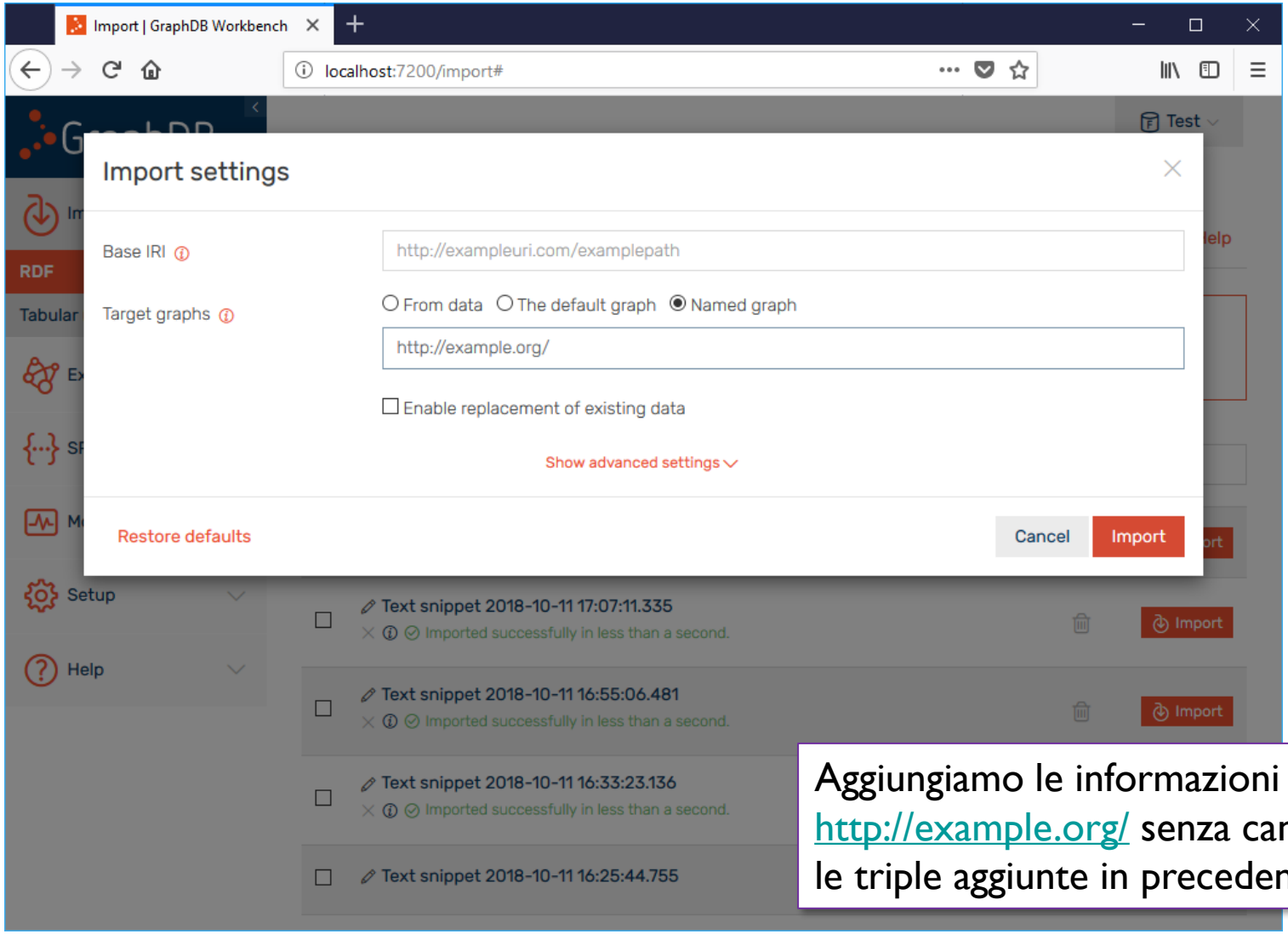


Rendiamo più interessante la descrizione delle istanze (2/3)

```
@prefix : <http://example.org/> .

:bigGroup a foaf:Group ;
   foaf:name "Big Group" ;
   foaf:member :john, :mary, :philip .

:mediumGroup a foaf:Group ;
   foaf:name "Medium Group" ;
   foaf:member :john, :alice .

:john foaf:homepage
<http://example.org/users/john> .
<http://example.org/users/john> a
foaf:Document .
```

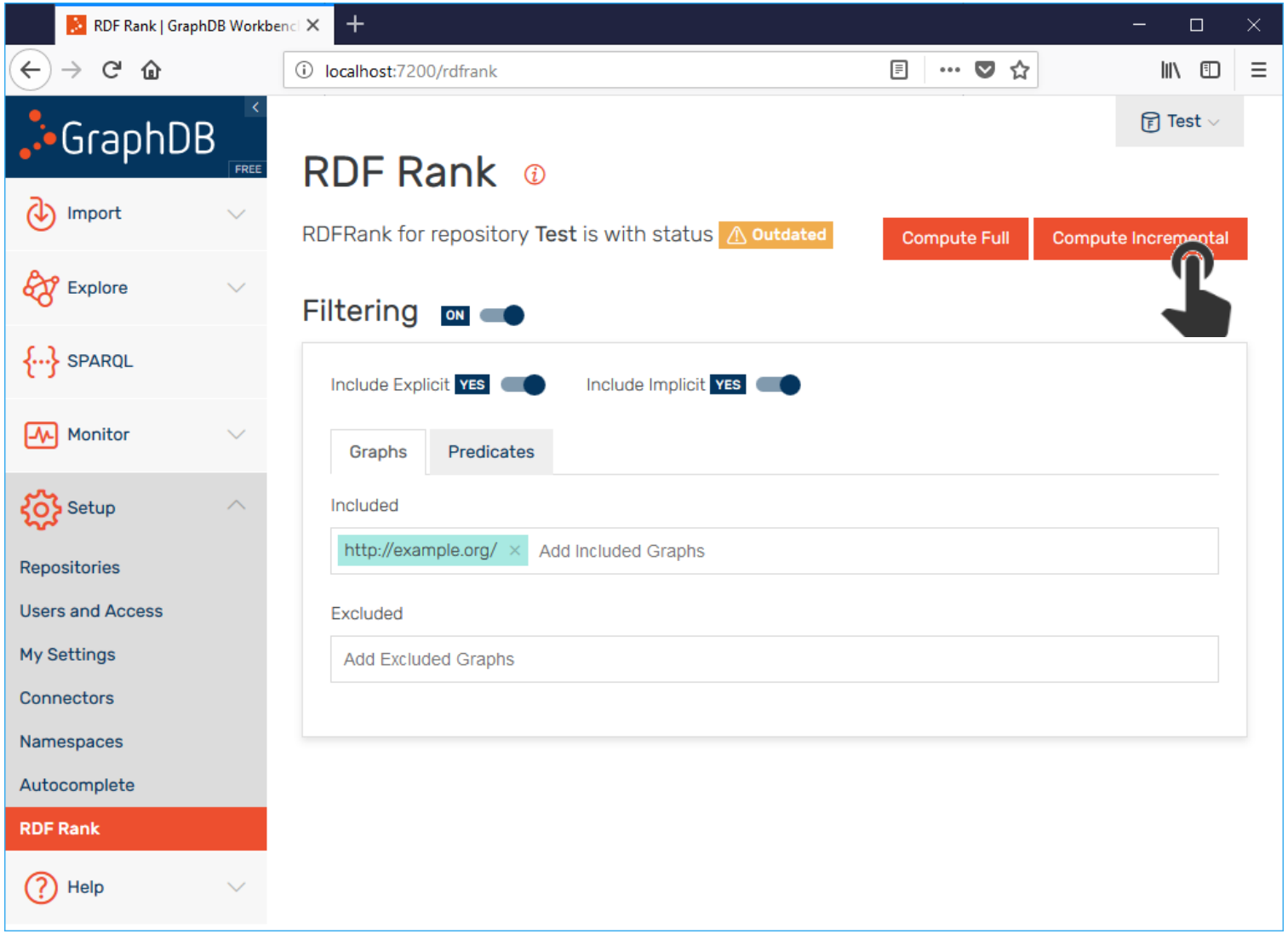
```
:mary foaf:homepage
<http://example.org/users/mary> .
<http://example.org/users/mary> a
foaf:Document .

:philip foaf:homepage
<http://example.org/users/philip> .
<http://example.org/users/philip> a
foaf:Document .

:alice foaf:homepage
<http://example.org/users/alice> .
<http://example.org/users/alice> a
foaf:Document .
```

Rendiamo più interessante la descrizione delle istanze (3/3)

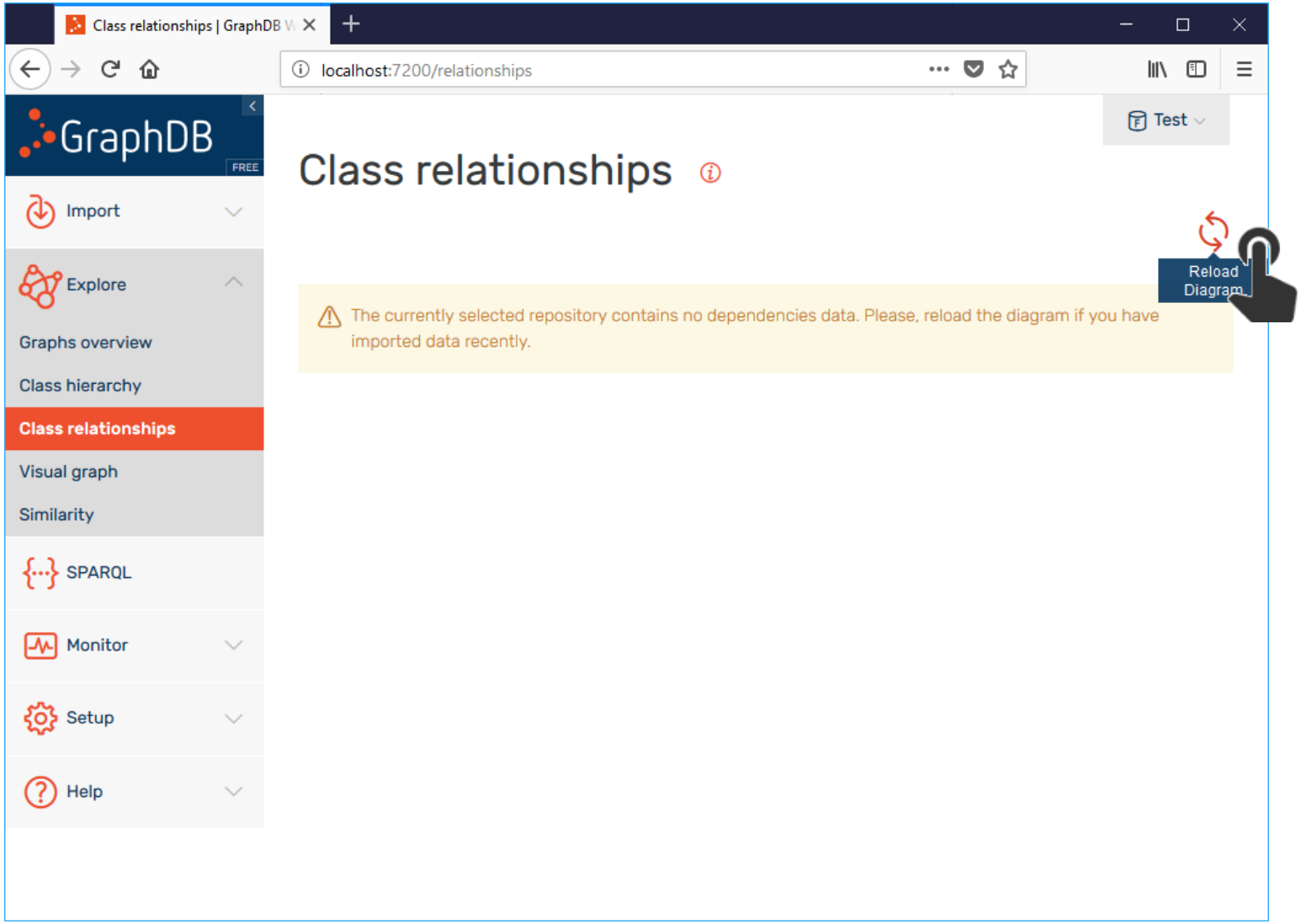


Aggiungiamo le informazioni al grafo http://example.org/ senza cancellare le triple aggiunte in precedenza

Calcolo incrementale dell'RDF Rank

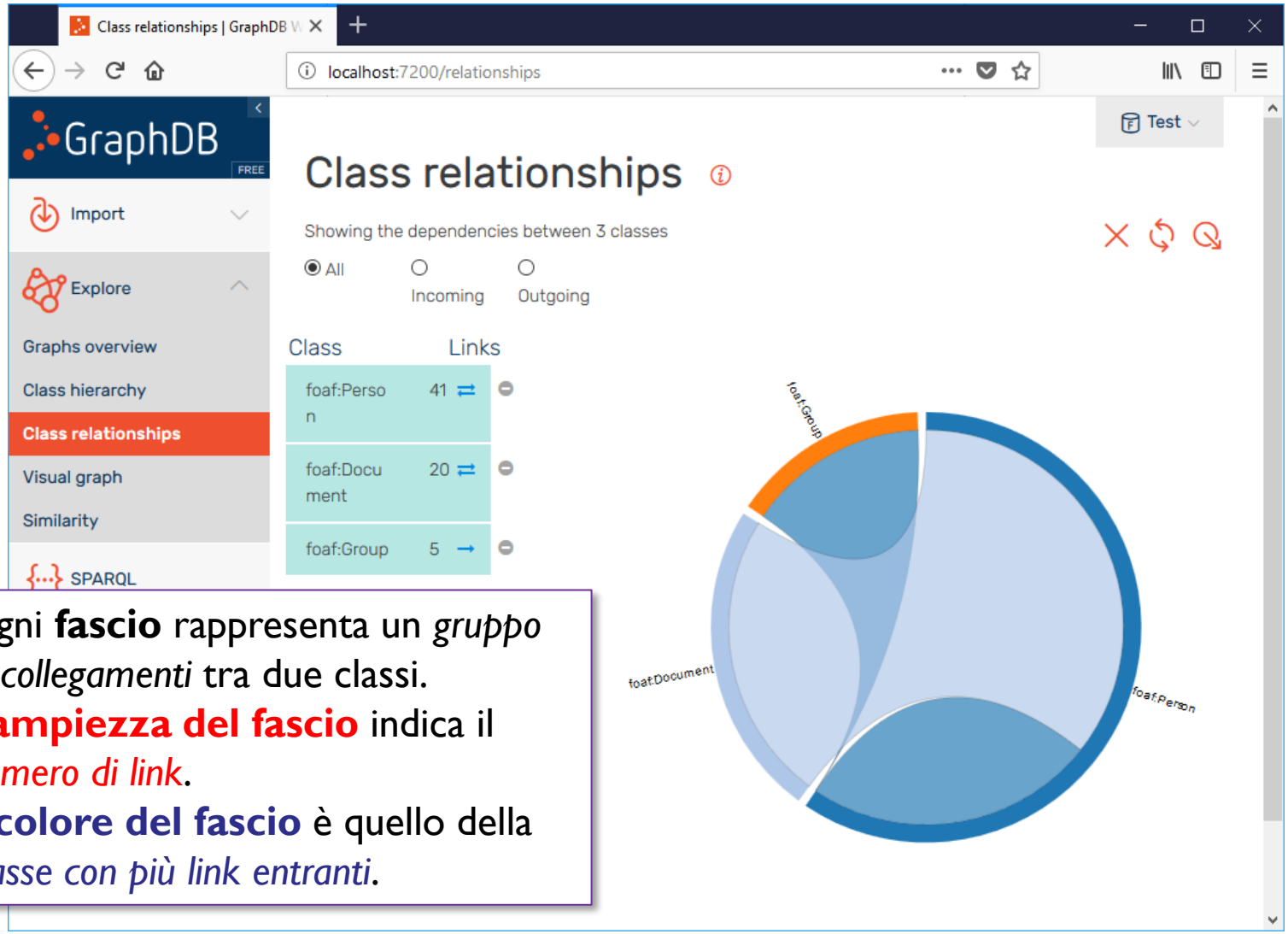


Explore – Class Relationships - Aggiornamento



Manuel Fiorelli fiorelli@info.uniroma2.it
http://art.uniroma2.it/fiorelli

Explore – Class Relationships (1/2)

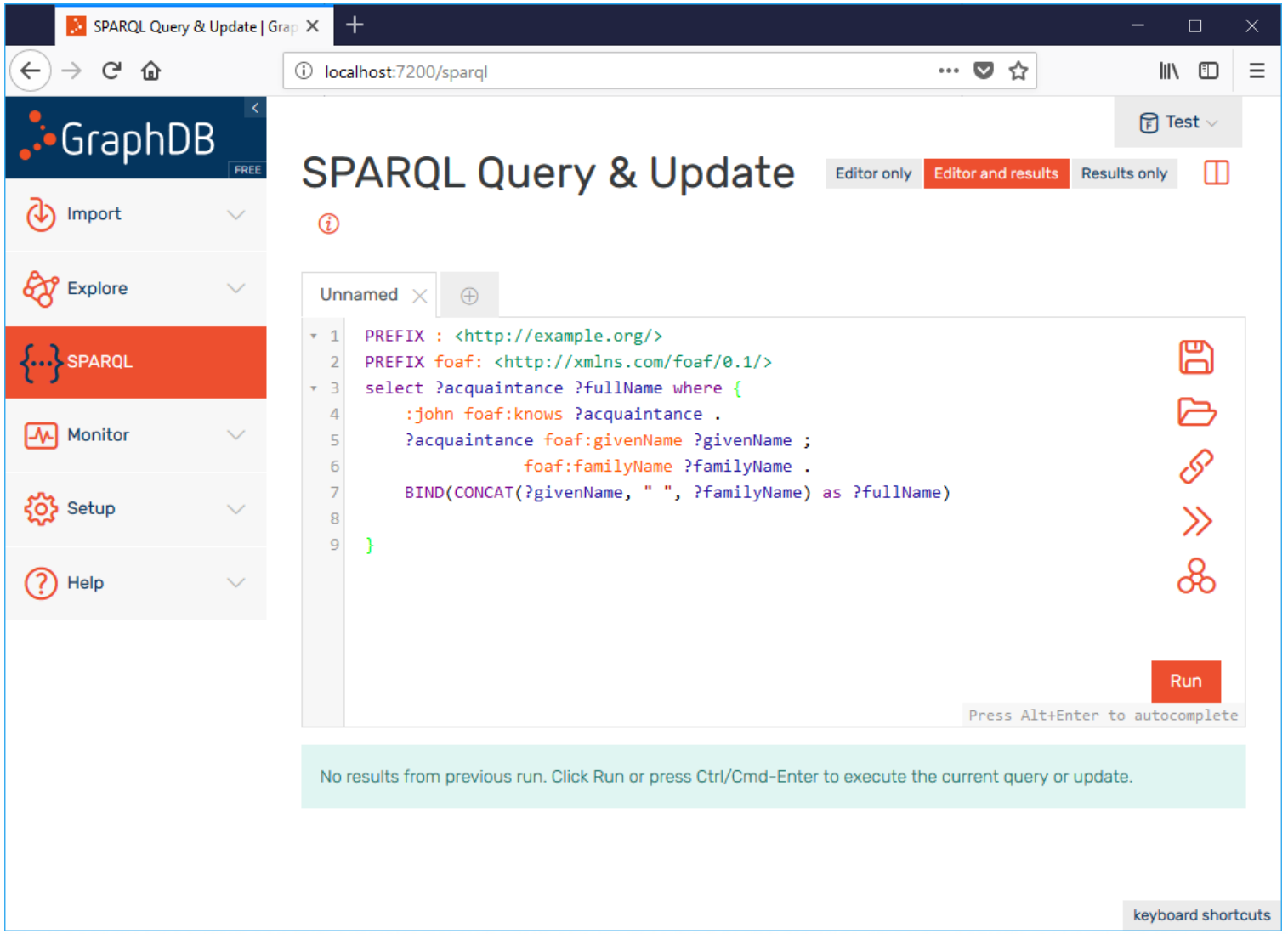


Ogni **fascio** rappresenta un *gruppo di collegamenti* tra due classi.
L'**ampiezza del fascio** indica il *numero di link*.
Il **colore del fascio** è quello della *classe con più link entranti*.

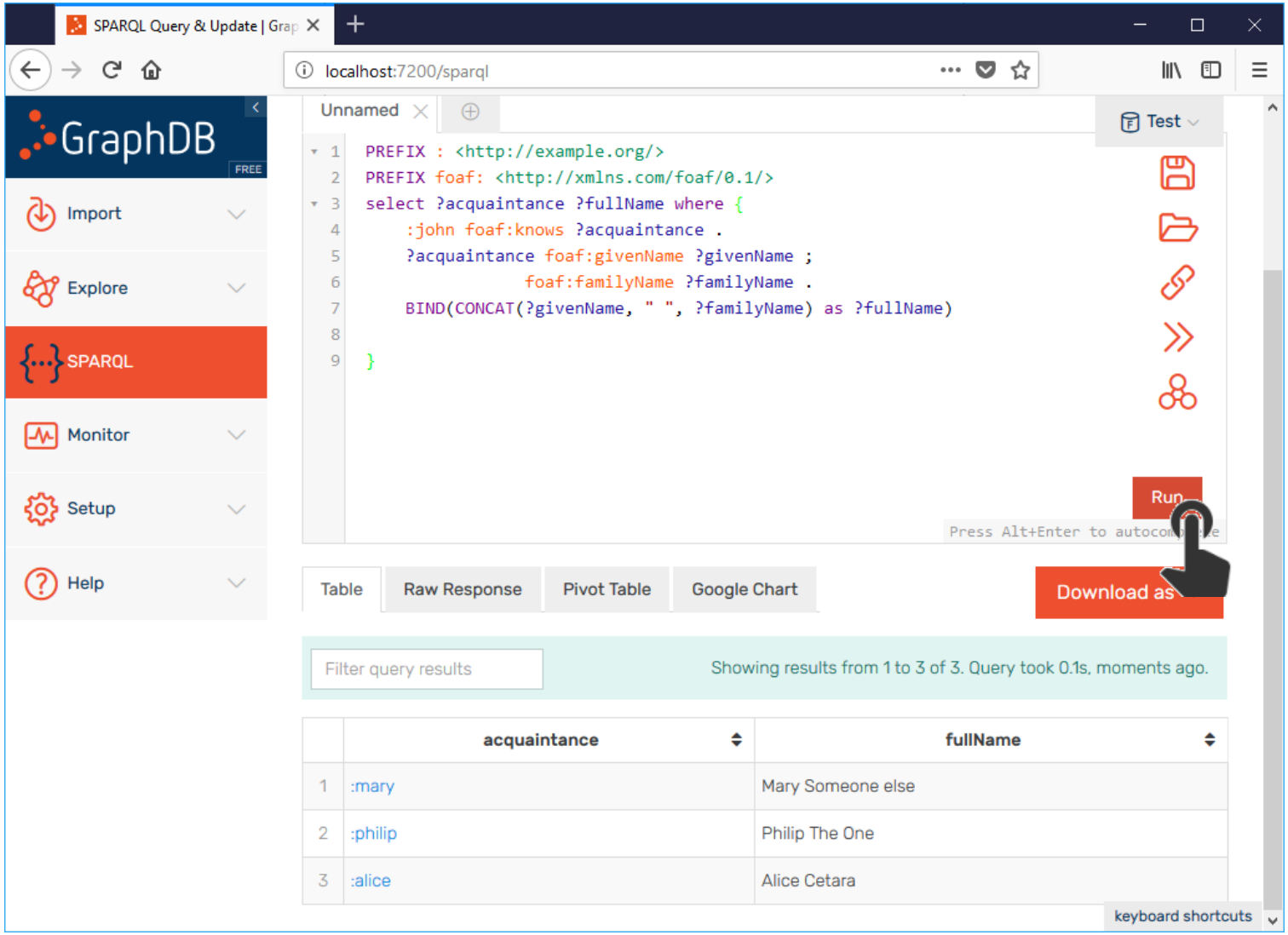
Explore – Class Relationships (2/2)

Il diagramma *class relationships* visualizza ed aiuta a comprendere le relazioni tra le classi come sono determinate da collegamenti effettivi tra le loro istanze.

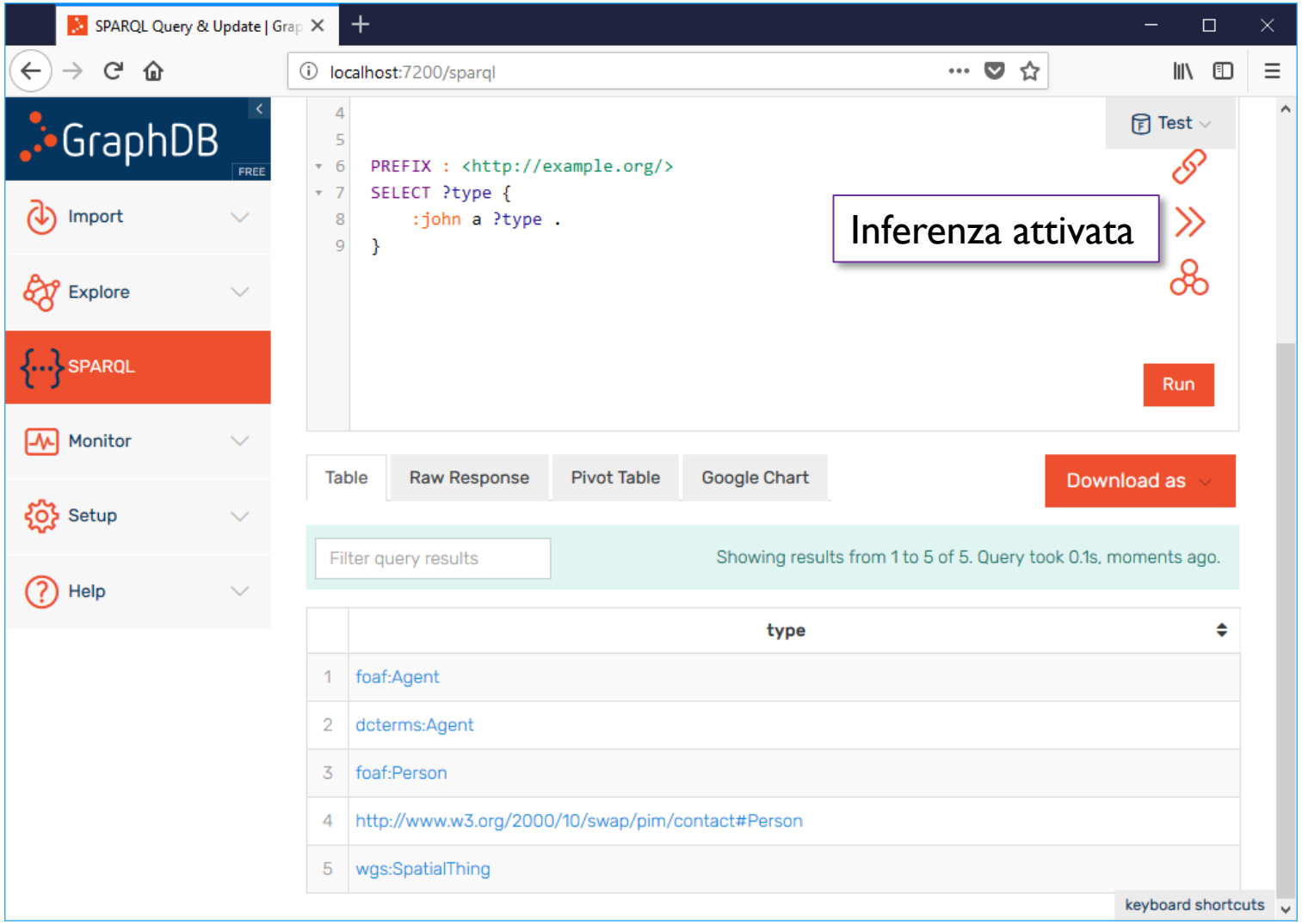
Query SPARQL (1/6)



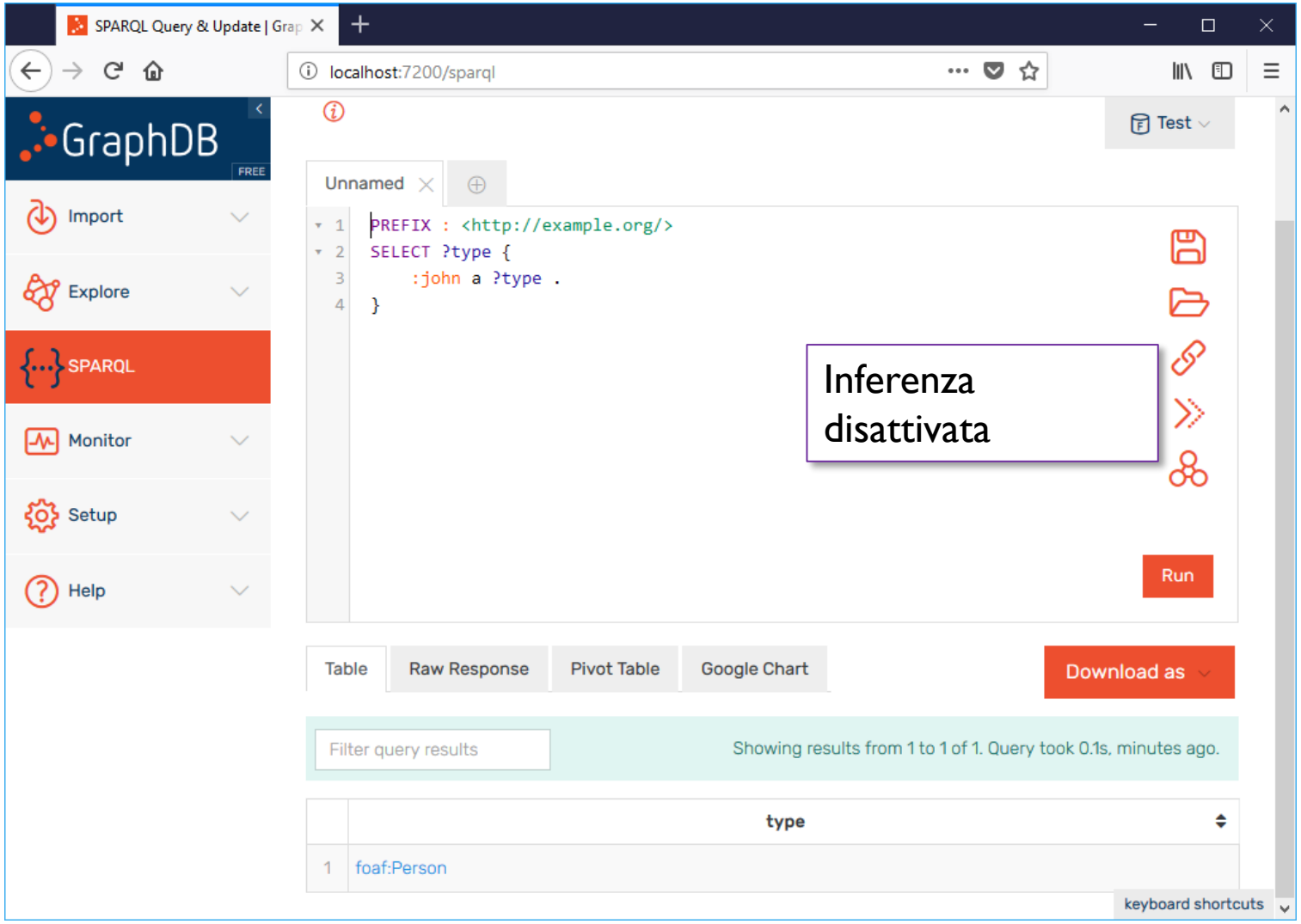
Query SPARQL (2/6)



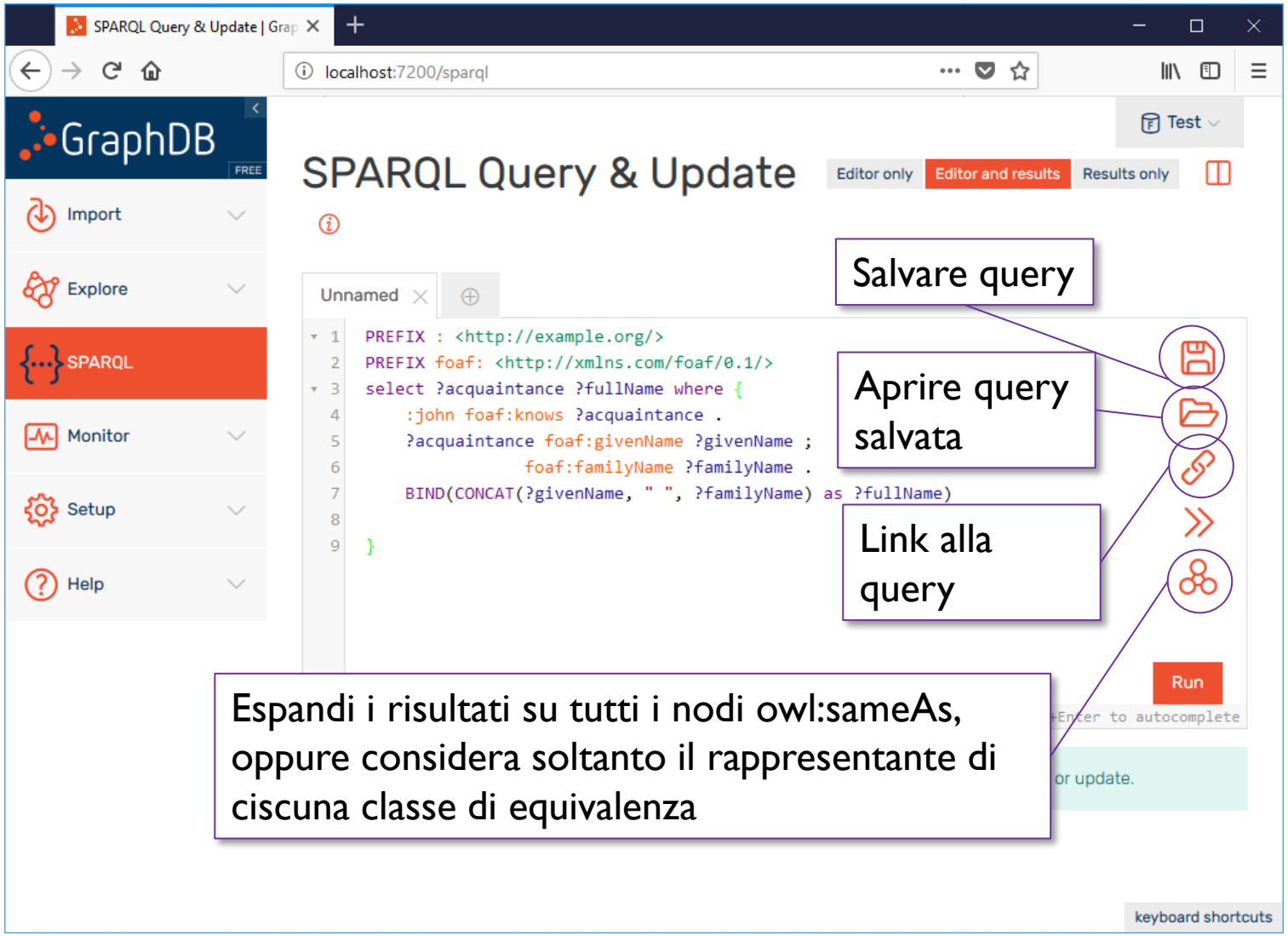
Query SPARQL (3/6)



Query SPARQL (4/6)

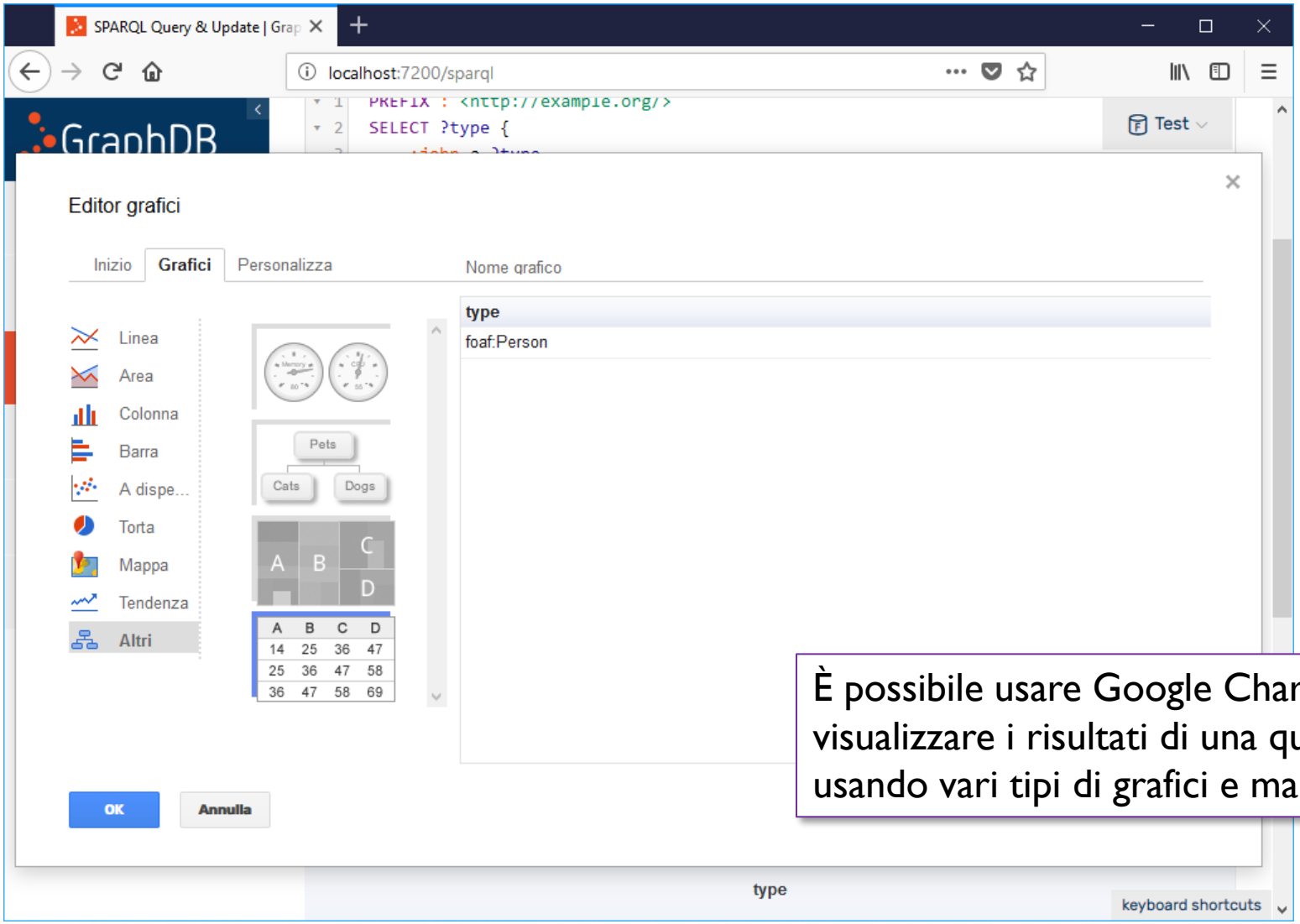


Query SPARQL (5/6)



Espandi i risultati su tutti i nodi owl:sameAs, oppure considera soltanto il rappresentante di ciscuna classe di equivalenza

Query SPARQL (6/6)



È possibile usare Google Chart per visualizzare i risultati di una query usando vari tipi di grafici e mappe

Reasoning (1/2)

GraphDB implementa il reasoning usando la strategia di **materializzazione completa**:

> Applica iterativamente tutte le regole definite nel ruleset selezionato finché non riesce più ad inferire nuove triple.

Il costo del reasoning viene trasferito sulle transazioni in scrittura (poiché tutto il processo di inferenza viene eseguito dopo le modifiche), mentre le transazioni in lettura sono avvantaggiate (poiché le triple inferite si trovano già pronte negli indici)

Reasoning (2/2)

La semantica sottostante è **monotona**.

L'**aggiunta di una tripla** non può invalidare inferenze precedenti, ma soltanto permetterne di nuove:

- è sufficiente proseguire con nuove iterazioni del processo di inferenza (partendo dalle regole attivate dalla nuova tripla)

La **rimozione di una tripla** può invalidare inferenze precedenti, ed una tripla inferita deve essere rimossa se non esiste nessun altro modo per derivarla senza usare la tripla rimossa:

- La soluzione naive consiste nel riavviare il processo di reasoning da zero, ma ciò significa che il costo di una rimozione sarà sempre maggiore man mano che la taglia del repository cresce. GraphDB supporta anche qui un approccio incrementale: i) cerca in forward-chaining le triple che si possono inferire dalla tripla che è stata rimossa, ii) poi cerca in backward-chaining derivazioni per queste triple che non usano la tripla cancellata

Ottimizzazione di owl:sameAs

Nella semantica di OWL **owl:sameAs** è una proprietà riflessiva, simmetrica e transitiva, cioè una **relazione di equivalenza**.

Ciò significa che se ho N risorse *sameAs* tra di loro, avrò N^2 *sameAs* ed ogni statement che coinvolge ciascuna risorsa viene ripetuto per tutte le altre.

GraphDB offre una implementa non rule-based di owl:sameAs, per cui questi gruppi di risorse sono rappresentati esplicitamente come classi di equivalenza, ciascuna dotata di un rappresentate che viene usato come identificatore della classe, e per rappresentare in maniera più compatta tutti gli statement che coinvolgono i membri della classe.

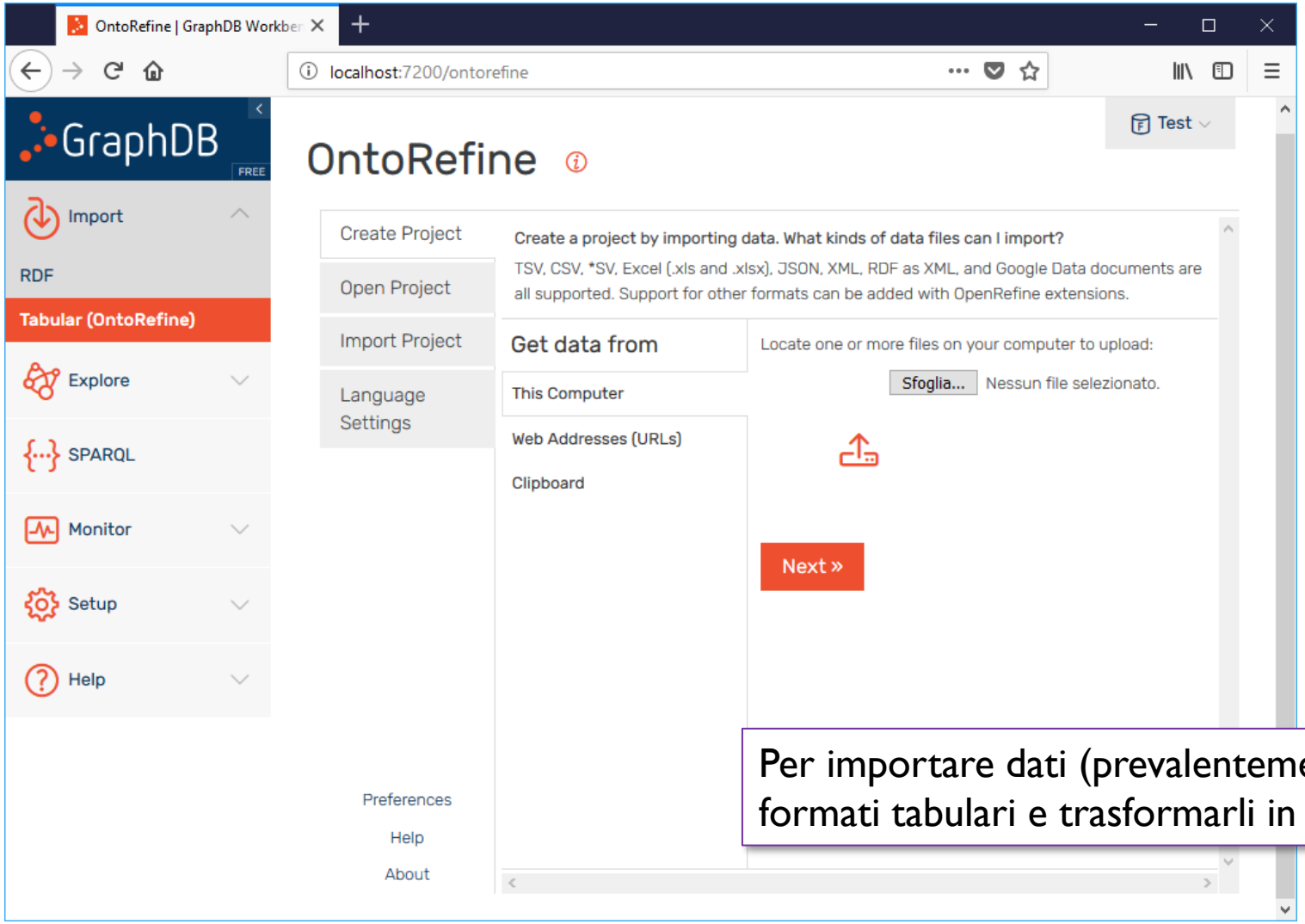
GEOSPARQL & Full text search

GraphDB supporta query geospaziali e fulltext search, (usando opportuni indici):

- http://graphdb.ontotext.com/documentation/free/full-text-search.html
- http://graphdb.ontotext.com/documentation/free/geosparql-support.html

Anche nel caso della fulltext search, è possibile aggiornare gli indici in maniera incrementale dopo cambiamenti dei dati.

OntoRefine



Per importare dati (prevalentemente) in formati tabulari e trasformarli in RDF

Manuel Fiorelli fiorelli@info.uniroma2.it
http://art.uniroma2.it/fiorelli

References

1. Ontotext GraphDB. http://graphdb.ontotext.com/
2. Bishop, B., Kiryakov, A., Ognyanoff, D., Peikov, I., Tashev, Z., & Velkov, R. (2011). OWLIM: A family of scalable semantic repositories. Semantic Web, 2(1), 33-42.