



Università degli Studi di Roma “Tor Vergata”

GraphDB

Esercizi Reasoning

Manuel Fiorelli

fiorelli@info.uniroma2.it

Per prima cosa dobbiamo creare un repository:

- **RepositoryID:** *TestOWL*
- **Ruleset:** *OWL2-RL*
 - Si noti che dopo aver selezionato questo ruleset, *la spunta sull'opzione disable owl:sameAs viene tolta automaticamente* (cioè l'ottimizzazione viene attivata)

Creazione di un repository (2/6)

The screenshot shows the GraphDB Workbench interface in a browser window. The address bar shows 'localhost:7200'. The left sidebar contains navigation options: Import, Explore, SPARQL, Monitor, Setup, and Help. The main content area displays a warning message: 'Some functionality is not available because you are not connected to any repository. Click one of the repositories below to connect to it or create a new repository.' Below the message is a list of available repositories:

DBpedia_Ontology_cor...	EuroVoc_core · Core reposi...
GDB_TEST1	OpenMultilingualWordne...
RemoteHistory_core · C...	RemoteHistory_support...
Test	TestOWL2

At the bottom right, there is a red button labeled 'Create new repository' with a plus icon. A hand cursor is pointing at this button. At the bottom left, there is a blue button labeled 'License'.

Creazione di un repository (3/6)

GraphDB **FREE**

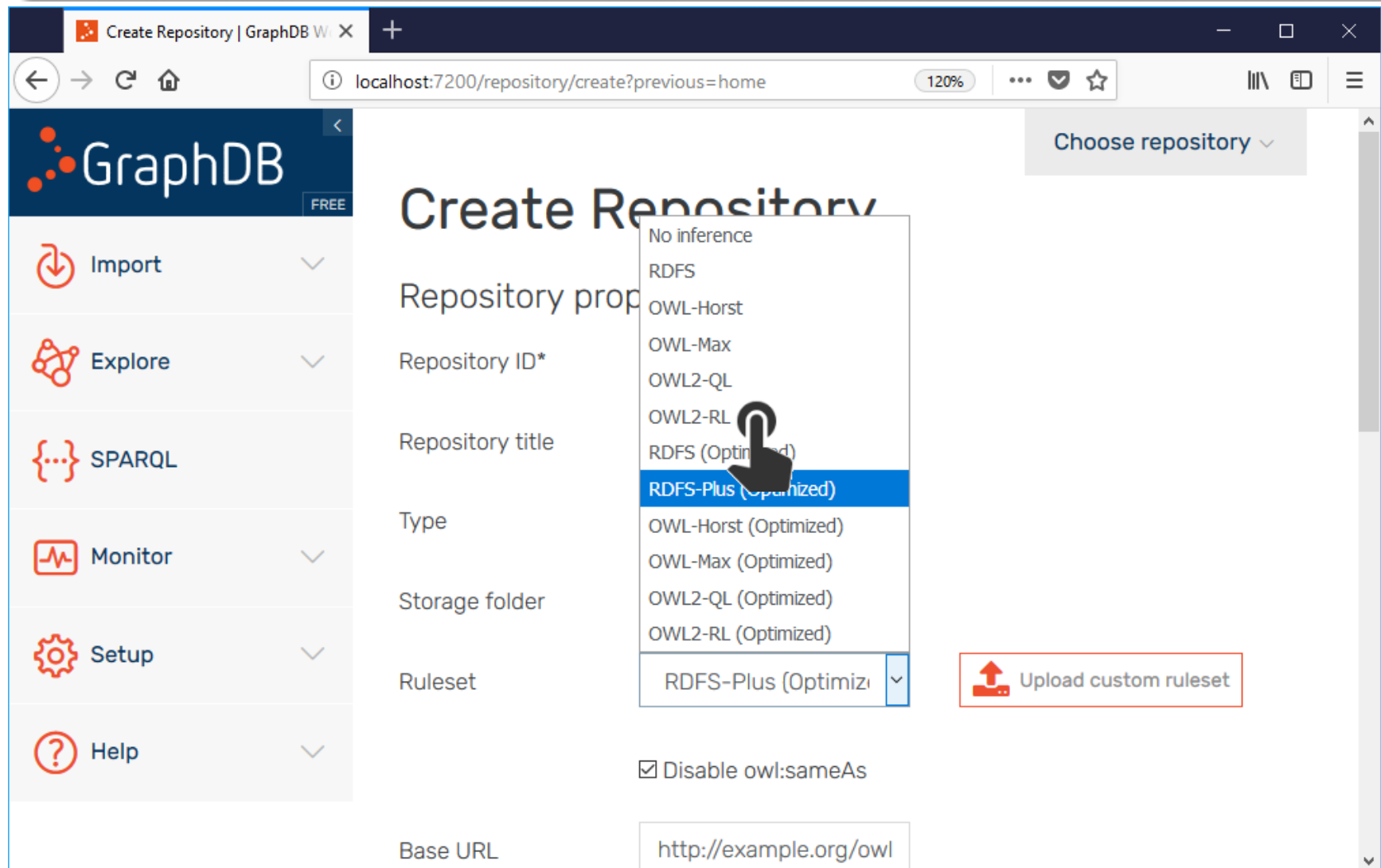
Choose repository ▾

Create Repository

Repository properties

Repository ID*	<input type="text" value="TestOWL"/>
Repository title	<input type="text"/>
Type	<input type="text" value="GRAPHDB-FREE"/>
Storage folder	<input type="text" value="storage"/>
Ruleset	<input style="border: 1px solid #ccc;" type="text" value="RDFS-Plus (Optimiz...)"/>
	<input checked="" type="checkbox"/> Disable owl:sameAs
Base URL	<input type="text" value="http://example.org/owl"/>

Creazione di un repository (4/6)



GraphDB FREE

Import

Explore

SPARQL

Monitor

Setup

Help

Choose repository

Create Repository

Repository properties

Repository ID*

Repository title

Type

Storage folder

Ruleset

Disable owl:sameAs

Base URL

http://example.org/owl

RDFS-Plus (Optimized)

Upload custom ruleset

Creazione di un repository (5/6)

GraphDB FREE

Import

Explore

SPARQL

Monitor

Setup

Help

Create Repository

Choose repository ▾

Repository properties

Repository ID*

Repository title

Type

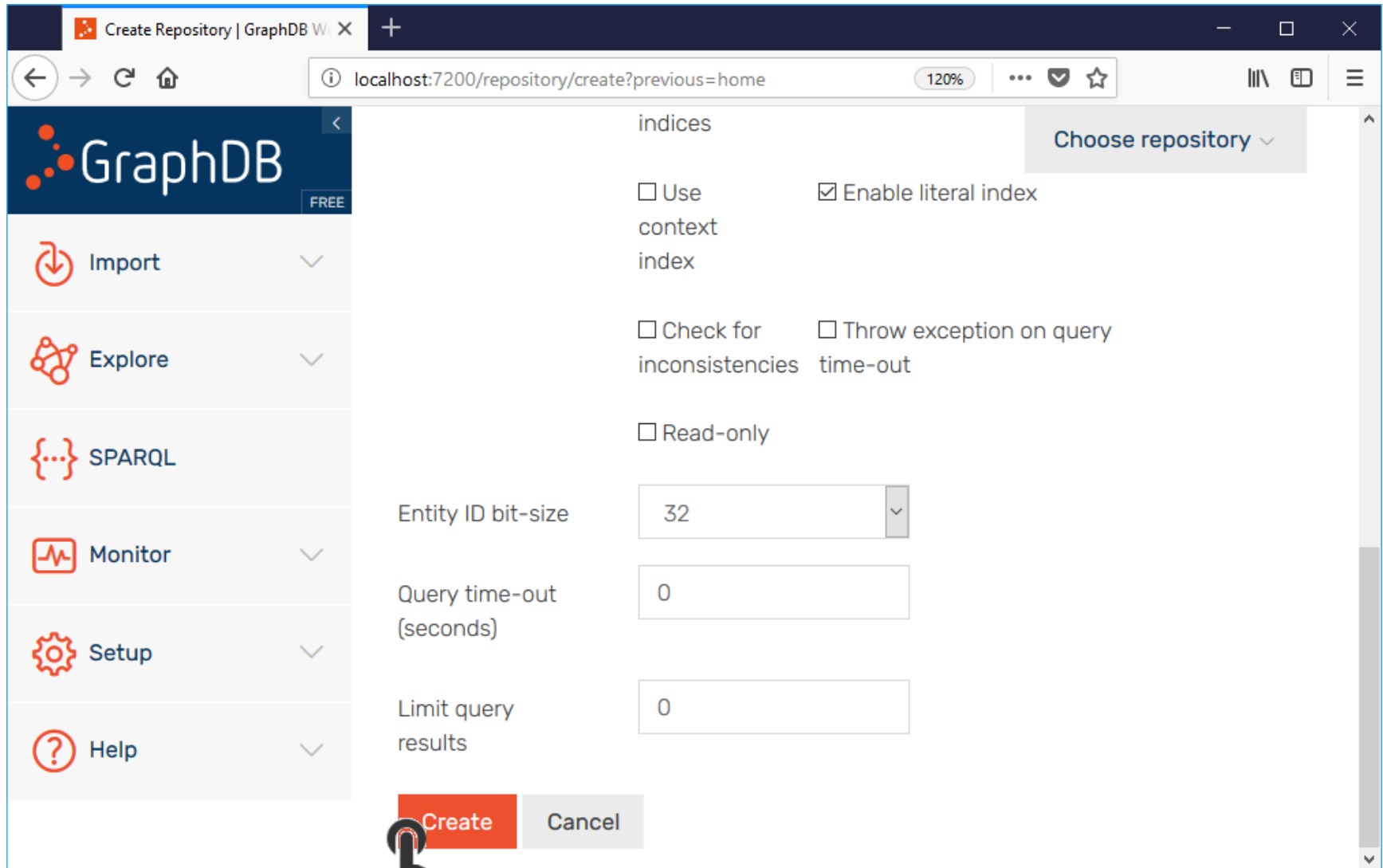
Storage folder

Ruleset

Disable owl:sameAs

Base URL

Creazione di un repository (6/6)



GraphDB FREE

Import

Explore

SPARQL

Monitor

Setup

Help

indices

Choose repository

Use context index

Enable literal index

Check for inconsistencies

Throw exception on query time-out

Read-only

Entity ID bit-size: 32

Query time-out (seconds): 0

Limit query results: 0

Create Cancel

Selezione del repository

The screenshot shows the GraphDB Workbench interface in a browser window. The address bar shows 'localhost:7200'. A yellow warning banner at the top states: 'Some functionality is not available because you are not connected to any repository. Click one of the repositories below to connect to it or create a new repository.' Below this is a table of available repositories:

DBpedia_Ontology_core...	EuroVoc_core • Core reposit...
GDB_TEST1	OpenMultilingualWordnet...
RemoteHistory_core • Cor...	RemoteHistory_support ...
Test	TestOWL
TestOWL2	

A mouse cursor is pointing at the 'TestOWL' repository. At the bottom right, there is a '+ Create new repository' button. The left sidebar contains navigation options: Import, Explore, SPARQL, Monitor, Setup, and Help. The 'GraphDB FREE' logo is visible in the top left of the interface.

License

Aggiunta di triple RDF con INSERT DATA (1/2)

The screenshot shows the GraphDB SPARQL Query & Update interface. The browser address bar shows `localhost:7200/sparql`. The interface includes a sidebar with navigation options: Import, Explore, SPARQL (highlighted), Monitor, Setup, and Help. The main area displays a SPARQL query in a code editor with line numbers 1 through 15. The query is as follows:

```

1 PREFIX owl: <http://www.w3.org/2002/07/owl#>
2 prefix : <http://example.org/>
3 INSERT DATA {
4     :ClassA a owl:Class .
5     :ClassB a owl:Class .
6
7     :instance1 a :ClassA .
8     :instance2 a :ClassB .
9
10    :Intersection a owl:Class ;
11        owl:intersectionOf (:ClassA :ClassB) .
12
13    :instance1 owl:sameAs :instance2 .
14 }
15

```

At the bottom right of the code editor, there is a red "Run" button with a mouse cursor hovering over it. Below the code editor, a message states: "No results from previous run. Click Run or press Ctrl/Cmd-Enter to execute keyboard shortcuts".

Aggiunta di triple RDF con INSERT DATA (2/2)

The screenshot shows the GraphDB SPARQL Query & Update interface. The left sidebar contains navigation options: Import, Explore, SPARQL (highlighted), Monitor, Setup, and Help. The main area displays a SPARQL query in a code editor with line numbers 1 through 15. The query is as follows:

```

1 PREFIX owl: <http://www.w3.org/2002/07/owl#>
2 prefix : <http://example.org/>
3 INSERT DATA {
4     :ClassA a owl:Class .
5     :ClassB a owl:Class .
6
7     :instance1 a :ClassA .
8     :instance2 a :ClassB .
9
10    :Intersection a owl:Class ;
11        owl:intersectionOf (:ClassA :ClassB) .
12
13    :instance1 owl:sameAs :instance2 .
14 }
15

```

Below the code editor, a status bar indicates: "Added 11 statements. Update took 0.2s, moments ago." This message is circled in purple. To the right of the status bar, there is a "Run" button and a tooltip that says "Press Alt+Enter to autocomplete".

Spiegazione dei dati appena aggiunti

- Abbiamo definito due classi (*ClassA* e *ClassB*) e fornito un'istanza per ciascuna di esse (rispettivamente, *instance1* e *instance2*).
- Abbiamo quindi definito la classe *Intersection* come l'intersezione delle due classi citate di sopra
- Abbiamo asserito che le due istanze sono in effetti la stessa cosa (usando il predicato *owl:sameAs*)

Istanze di *Intersection* senza inferenza (1/2)

SPARQL Query & Update | GraphDB

localhost:7200/sparql

GraphDB FREE

Import

Explore

```
1 prefix : <http://example.org/>
2 select ?s where {
3   ?s a :Intersection
4 }
5
```

Table | Raw Response | Pivot Table | Google Chart

Download as

Run

Filter query results

No results. Query took 0.1s, minutes ago.

s

No data available in table

keyboard shortcuts

Istanze di *Intersection* senza inferenza (2/2)

Nessuna risorsa è *esplicitamente dichiarata* essere un'istanza di *Intersection*.

Istanze di *Intersection* con inferenza e espansione owl:sameAs (1/2)

SPARQL Query & Update | GraphDB

localhost:7200/sparql

```

1 prefix : <http://example.org/>
2 select ?s where {
3   ?s a :Intersection
4 }

```

Table | Raw Response | Pivot Table | Google Chart

Download as

Filter query results

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

	s
1	http://example.org/instance1
2	http://example.org/instance2

Run

keyboard shortcuts

Istanze di *Intersection* con inferenza e espansione owl:sameAs (2/2)

instance1 è dichiarata esplicitamente di tipo *ClassA*, ma essendo stata dichiarata coincidente con *instance2*, che è di tipo *ClassB*, possiamo inferire che *instance1* è anche di tipo *ClassB*. Pertanto, si può inferire che *instance1* appartiene alla classe *Intersection*.

Un ragionamento analogo ci permette di inferire che *instance2* appartiene alla classe *Intersection*.

Istanze di *Intersection* con inferenza e senza espansione owl:sameAs (1/2)

SPARQL Query & Update | GraphDB

localhost:7200/sparql

```
1 prefix : <http://example.org/>
2 select ?s where {
3   ?s a :Intersection
4 }
```

Download as

Filter query results

ing results from 1 to 1 of 1. Query took 0.2s, minutes ago.

	s
1	http://example.org/instance1

Run

keyboard shortcuts

Istanze di *Intersection* con inferenza e senza espansione owl:sameAs (2/2)

GraphDB possiede un'implementazione non rule-based del predicato *owl:sameAs*.

- Tiene traccia delle classi di equivalenza generate da *owl:sameAs*, associando a ciascuna di esse un *rappresentante* (tra le risorse che vi appartengono)
- Rappresenta ciascuno statement una sola volta sostituendo a ciascun componente i rappresentati delle classi di equivalenza cui appartengono
- Durante la valutazione di una query, GraphDB utilizza una sorta di backward-chaining enumerando tutti gli IRI equivalenti, garantendo la completezza dell'inferenza e dei risultati.
- L'implementazione di questa feature è tale da permettere di distinguere gli statement espliciti da quelli impliciti.

Per maggiori dettagli si consulti:

<http://graphdb.ontotext.com/documentation/free/sameas-optimisation.html>

Da notare l'uso della clausola *FROM onto:disable-sameAs* per disabilitare da dentro la query SPARQL l'espansione di *owl:sameAs* (utile quando si sottomette la query tramite API REST)

Aggiunta di altre triple (vedi esercizi OWL 9 e 10)

The screenshot shows the GraphDB SPARQL Query & Update interface. The browser address bar is localhost:7200/sparql. The left sidebar contains navigation options: Import, Explore, SPARQL (selected), Monitor, Setup, and Help. The main area displays a SPARQL query in a code editor with line numbers 1 through 16. The query defines two prefixes and inserts data for a family tree.

```

1 PREFIX owl: <http://www.w3.org/2002/07/owl#>
2 prefix : <http://example.org/>
3 INSERT DATA {
4     :grandparent a owl:ObjectProperty ;
5         owl:propertyChainAxiom (:parent :parent) .
6     :parent a owl:ObjectProperty .
7     :philipp a owl:Thing ;
8     :parent :john , :mary .
9     :alice a owl:Thing ;
10        :parent :luke , :janet .
11    :mary a owl:Thing ;
12        :parent :frank , :jessica .
13    :luke a owl:Thing ;
14        :parent :frank , :jessica .
15 }
16

```

At the bottom of the editor, there is a "Run" button and a tooltip that says "Press Alt+Enter to autocomplete". Below the editor, a message states: "No results from previous run. Click Run or press Ctrl/Cmd-Enter to execute keyboard shortcuts".

Query su property chain (senza inferenza)

The screenshot shows the GraphDB SPARQL Query & Update interface. The browser address bar is localhost:7200/sparql. The interface includes a sidebar with 'Import' and 'Explore' options. The main area contains a SPARQL query editor with the following code:

```
1 prefix : <http://example.org/>
2
3 select * where {
4   ?s :grandparent ?o .
5 }
```

Below the query editor, there are tabs for 'Table', 'Raw Response', 'Pivot Table', and 'Google Chart'. A 'Download as' button is highlighted with a red box and a purple arrow. Below the query editor, there is a 'Run' button with a hand cursor icon. The results area shows 'No results. Query took 0.1s, moments ago.' and a table with the headers 's' and 'o'. A 'keyboard shortcuts' button is visible at the bottom right.

Query su property chain (con inferenza)

The screenshot shows the GraphDB SPARQL Query & Update interface. The query is:

```

prefix : <http://example.org/>
select * where {
  ?s :grandparent ?o .
}

```

The results table shows the following data:

	s	o
1	http://example.org/philipp	http://example.org/frank
2	http://example.org/philipp	http://example.org/jessica
3	http://example.org/alice	http://example.org/frank
4	http://example.org/alice	http://example.org/jessica

A purple arrow points from the 'Run' button to the 'o' column header in the results table. The 'Run' button is a red rectangle with the text 'Run' and a hand icon pointing to it. The interface also shows a 'Filter query results' input field and a status bar indicating '4 results from 1 to 4 of 4. Query took 0.1s, moments ago.'

Query *DESCRIBE* su *grandparent* (1/2)

SPARQL Query & Update | GraphDB

localhost:7200/sparql

GraphDB FREE

- Import
- Explore
- SPARQL**
- Monitor
- Setup
- Help

```

1 prefix : <http://example.org/>
2
3 describe :grandparent
    
```

Run

Table Raw Response Pivot Table Google Chart keyboard shortcuts

Query *DESCRIBE* su *grandparent* (2/2)

Filter query results

Showing results from 1 to 6 of 6. Query took 0.1s, minutes ago.

	subject	predicate	object
1	_:node10	rdf:first	http://example.org/parent
2	_:node10	rdf:rest	rdf:nil
3	_:node9	rdf:first	http://example.org/parent
4	_:node9	rdf:rest	_:node10
5	http://example.org/grandparent	rdf:type	owl:ObjectProperty
6	http://example.org/grandparent	owl:propertyChainAxiom	_:node9

L'oggetto della tripla col predicato *owl:propertyChainAxiom* è un blank node che rappresenta una RDF Collection attraverso un approccio molto simile a quello usato da Prolog: una lista contiene un elemento (*rdf:first*) e un riferimento al resto della lista (*rdf:rest*) (si noti che *rdf:nil* è la lista vuota)