PhUSE CSS

Linked Data for Clinical Trials

An Interactive Hands-on Workshop

EXERCISES

Version 3.0

04 March 2018

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

Contents

[Introduction 2](#_Toc505785359)

[Materials 2](#_Toc505785360)

[Symbols 2](#_Toc505785361)

[Server Login 3](#_Toc505785362)

[Exercises 4](#_Toc505785363)

[1. Create a Graph in the Graph Editor 4](#_Toc505785364)

[1.1 Open the Graph Editor 4](#_Toc505785365)

[1.2 Add a Study, Study Phase, and Treatment Arms 4](#_Toc505785366)

[1.3 Add Persons and Treatments 6](#_Toc505785367)

[1.4 Add Gender to Persons 7](#_Toc505785368)

[1.5 Demographics for Person Nodes 8](#_Toc505785369)

[1.6 Optional Activity 9](#_Toc505785370)

[1.7 Export to TTL 9](#_Toc505785371)

[1.8 Upload to Database 10](#_Toc505785372)

[2. Query your graph 11](#_Toc505785373)

[2.1 Show all triples 11](#_Toc505785374)

[2.2 How many people participate in the study? 11](#_Toc505785375)

[2.3 Find the names of people who participate in the study. 11](#_Toc505785376)

[2.4 List the names of the people randomized to each treatment type. 12](#_Toc505785377)

[3. Ontology 13](#_Toc505785378)

[3.1 Review the Ontology 13](#_Toc505785379)

[3.2 Add the Ontology to the LDWStudy database 13](#_Toc505785380)

[3.3 Explore the data 13](#_Toc505785381)

[3.4 Find the names of HumanStudySubjects in your study. 14](#_Toc505785382)

[3.5 Write a reasoner-based query to find all People associated with the study (both human study subjects and investigators) 15](#_Toc505785383)

[4. Merge data from all studies 15](#_Toc505785384)

[4.1 Obtain a copy of the pooled data. 15](#_Toc505785385)

[4.2 List Drug1 studies by phase. 16](#_Toc505785386)

[4.3 How many HumanStudySubjects participated in Drug1 studies? 16](#_Toc505785387)

[4.4 What is the gender breakdown for HumanStudySubjects across all Drug1 studies? 16](#_Toc505785388)

[4.5 Identify patients that participated in multiple studies. 16](#_Toc505785389)

[Appendix 1: Course Resources 18](#_Toc505785390)

# Introduction

Instructions in this document are specific to the PhUSE cloud server environment, Graph Editor application and workshop methods. The exercises represent one of many possible approaches to the material and make no claim to be best or recommended method. The instructor will complete the exercises during the session, along with the attendees.

Your feedback is welcomed and encouraged. Please send your comments to: **tim.williams@PhUSE.eu**

## Materials

* Laptop with Remote Desktop capability
* Printed copies of the following will be supplied by the instructor:
  + Exercises (this document)
  + Info Sheet
  + Graph Editor Reference

## Symbols

|  |  |
| --- | --- |
|  | Stop and wait. Do not proceed to the next section until instructed. |
| **!** | Cautions and warnings. Failure to follow these steps may lead to unanticipated results and problems. |
|  | The instructor presents important material at this point. |
| IdeaIcon_clean_20mm | Helpful tips and advice. |
| Image result for information icon | Additional information. |

# Server Login

Login to the cloud server using Remote Desktop. Instructions assume Windows 10 operating system.

1. In the search box on the taskbar on your laptop, type **remote desk,** then select **Remote Desktop Connection** from the Programs section.﻿
2. Click **Show Options** if needed to show the fields for **Computer:** and **User name:**

The instructor will provide the password and an IP Address for the Computer: field.

**Computer:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(See Info Sheet)

**User name:** phuseldw

**Password:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Provided by Instructor)

1. Entering values in the **Computer:** and **User name:** fields, then click **Connect**.
2. When prompted, enter the password supplied by the instructor and click **OK**.

|  |  |  |
| --- | --- | --- |
|  | Stop here and wait for the instructor.  Presentation follows |  |

# Exercises

# Create a Graph in the Graph Editor

# Open the Graph Editor

On the cloud server:

1. Open Google Chrome.
2. Click on "GraphEditor" on the shortcuts bar. 
3. The Graph Editor opens and shows two IRI nodes: One for **Drug1** and one for the drug's name, **Serum114.**



Ask for assistance if you do not see these nodes.

# Add a Study, Study Phase, and Treatment Arms

|  |  |
| --- | --- |
| **!** | When you see values like Study**(n)**, Phase**(n)**, Treatment**(n)**, etc. you must change your graph to values you find in the Info Sheet or create on your own. The values used in your graph become : Study**1**, Phase**3**, Treatment**7-1**, etc. |

Create a node that represents unique **Study** conducted for **Drug1** and the Treatments for that study. Reference the Info Sheet for values where needed.

1. Create a **Study** node with these values:

Node values:

|  |  |  |
| --- | --- | --- |
| **Prefix:** | eg |  |
| **Label:** | *Info Sheet: [Study]* | Image result for information icon*See Info Sheet for value* |
| **Type:** | IRI |  |

1. Create the relation ***from*** **Drug1** ***to*** the **Study** node:

Link values:

|  |  |
| --- | --- |
| **Prefix:** | ncit |
| **Label:** | study |

1. Create a Study **Phase** node. You may choose any of these phases: **Phase2**, **Phase3**, **Phase4.**

Node values:

|  |  |  |
| --- | --- | --- |
| **Prefix:** | ncit |  |
| **Label:** | *Choose a value:* | Image result for information icon*You choose a value for phase:* **Phase2, Phase3, Phase4**  *NO SPACE in the name!* |
| **Type:** | IRI |  |

1. Create the link ***from*** the **Study** ***to*** the Study **Phase**.

Link values:

|  |  |
| --- | --- |
| **Prefix:** | ncit |
| **Label:** | phase |

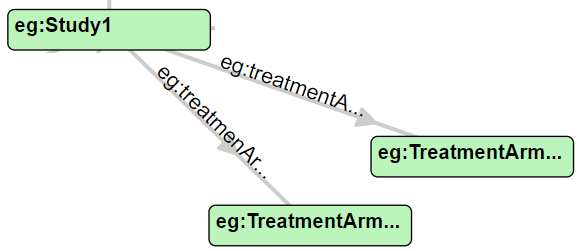
1. Create two **TreatmentArm** nodes for the **Study**.

Node values:

|  |  |  |
| --- | --- | --- |
| **Prefix:** | eg |  |
| **Label:** | *Info Sheet:*  *[First Treatment Arm]*  *[Second Treatment Arm]* | Image result for information icon*See Info Sheet for values.* |
| **Type:** | IRI |  |

Ensure you have two treatment arm nodes in your graph.

1. Create the relations ***from*** the **Study** ***to*** the two **TreatmentArm** nodes.



|  |  |  |
| --- | --- | --- |
| **Prefix:** | eg |  |
| **Label:** | treatmentArm | Image result for information icon*Label is all lowercase.* |

Link values:

There should now be one treatmentArm link to each of the **TreatmentArm** nodes.

1. Create two "treatment type" nodes to represent **PlaceboArm** and **ActiveArm** treatments

Node values:

|  |  |  |
| --- | --- | --- |
| **Prefix:** | eg |  |
| **Label:** | PlaceboArm |  |
| **Type:** | IRI |  |

Node values:

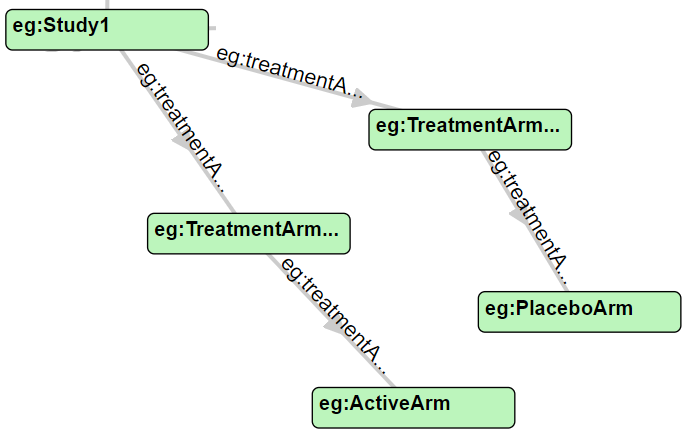
|  |  |  |
| --- | --- | --- |
| **Prefix:** | eg |  |
| **Label:** | ActiveArm |  |
| **Type:** | IRI |  |

1. Now join each **TreatmentArm** node (created in step x) to one of the **TreatmentArmType** nodes using the following relationship. It does not matter which is linked to the **PlaceboArm** or **ActiveArm**. You will create two edges:

Link values:

|  |  |  |
| --- | --- | --- |
| **Prefix:** | eg |  |
| **Label:** | treatmentArmType |  |

The **Study** and **TreatmentArm** portion of your graph should look similar to:



|  |  |
| --- | --- |
| IdeaIcon_clean_20mm | Click  to create a copy in case anything goes wrong in the next steps. |

|  |  |
| --- | --- |
| Image result for information icon | In these steps we defined two Treatment Arms that are unique to your study, then assigned those arms one of two Treatment Arm Types: PlaceboArm, ActiveArm. These "type of arm" nodes are identical across the studies in the workshop and allow identification of "all persons who received active treatment." In a real application, the Treatment Arm would be further extended to include dosage and administration information unique to that study. This graph would be too complex for the workshop. |

# Add Persons and Treatments

1. Add three **Person** nodes to the graph:

Node values:

|  |  |  |
| --- | --- | --- |
| **Prefix:** | eg |  |
| **Label:** | *Info Sheet: x.x, x.x , x.x [Person11, Person12, Person13]* | Image result for information icon*See Info Sheet for values.* |
| **Type:** | IRI |  |

1. For two of the **Person** Nodes: Create ***participatesIn*** relations ***from*** the **Person** ***to*** the **Study** node.

Link values:

|  |  |  |
| --- | --- | --- |
| **Prefix:** | eg |  |
| **Label:** | participatesIn | Image result for information icon*Note uppercase "I"* |

1. For the remaining **Person** node: Create a ***coordinatingInvestigator*** link ***from*** the **Person** node ***to*** the **Study** node.

Link values:

|  |  |  |
| --- | --- | --- |
| **Prefix:** | eg |  |
| **Label:** | coordinatingInvestigator | Image result for information icon*Note uppercase "I"* |

1. Assign one of the two **Person** nodes ***participatesIn*** the Study to one of the **TreatmentArm**s by creating the ***randomizedTo*** link ***from*** that **Person** ***to*** the **TreatArm**.

Link values:

|  |  |  |
| --- | --- | --- |
| **Prefix:** | eg |  |
| **Label:** | randomizedTo | Image result for information icon*Label is all lowercase.* |

1. Repeat for the second person node that ***participatesIn*** the Study by creating a ***randomizedTo*** link ***from*** the **Person** node ***to*** the other **TreatArm**.

Link values:

|  |  |  |
| --- | --- | --- |
| **Prefix:** | eg |  |
| **Label:** | randomizedTo | Image result for information icon*Label is all lowercase.* |

Now both **Person**s with the ***participatesIn*** relationship to the **Study** are assigned to **TreatmentArms.** The **Person** with the ***coordinatingInvestigator*** relationship is not assigned to a **Treat**ment.

# Add Gender to Persons

1. Create a node for **Male** and a node for **Female .**

Node values:

|  |  |  |
| --- | --- | --- |
| **Prefix:** | ncit |  |
| **Label:** | Male |  |
| **Type:** | IRI |  |

Node values:

|  |  |  |
| --- | --- | --- |
| **Prefix:** | ncit |  |
| **Label:** | Female |  |
| **Type:** | IRI |  |

2. Create links ***from***all 3 **Person** nodes ***to*** a Gender value using the ***gender*** relation.

Link values:

|  |  |  |
| --- | --- | --- |
| **Prefix:** | ncit |  |
| **Label:** | gender | Image result for information icon*Label is all lowercase.* |

|  |  |
| --- | --- |
| IdeaIcon_clean_20mm | Click  to save a copy of your graph. |

# Demographics for Person Nodes

A. Create one **Given Name** (First name) for each **Person** node in the graph:

1. Create **Given Name** nodes as strings:

Node values:

|  |  |  |
| --- | --- | --- |
| **Prefix:** | NOT APPLICABLE | Image result for information iconPrefix is not applicable to literals. You may ignore this field. |
| **Label:** | *<Any name you want>* |  |
| **Type:** | STRING |  |

2. Link ***from*** a **Person** ***to*** their **Given Name** using the ***givenName*** relation.

Link values:

|  |  |  |
| --- | --- | --- |
| **Prefix:** | schema |  |
| **Label:** | givenName |  |

B. Add an **Age** (in years) for each **Person** node.

1. Create **Age** nodes as INT (integers):

Node values:

|  |  |  |
| --- | --- | --- |
| **Prefix:** |  | Image result for information iconPrefix is not applicable to literals. |
| **Label:** | *<Any age integer value you want>* |  |
| **Type:** | INT |  |

2. Link ***from***a **Person** ***to*** an age using the ***age*** relation.

Link values:

|  |  |  |
| --- | --- | --- |
| **Prefix:** | eg | Image result for information icon The eg: prefix is used because age  calculation may be specific to this graph. |
| **Label:** | age |  |

|  |  |
| --- | --- |
| IdeaIcon_clean_20mm | Click  to save a copy of your graph. |

# Optional Activity

Time permitting, you may create these additional nodes while you wait for the class to catch up. Do not add too many nodes or your whiteboard may become difficult to read.

Ideas:

* Add additional Person nodes and assign them to a Treatment.
* Add additional treatments, or add details to the existing treatments.

|  |  |  |
| --- | --- | --- |
|  | Stop here and wait for the instructor. |  |

# Export to TTL

1. **As for the instructor or assistant to review your graph prior to taking the next steps. Click**  **if you make any changes.**
2. Click on  to export the graph to TTL format.
3. The graph will be saved to your downloads folder as **WhiteBoardTriples.TTL :**

**C:\Users\phuseldw\downloads\WhiteBoardTriples.TTL**

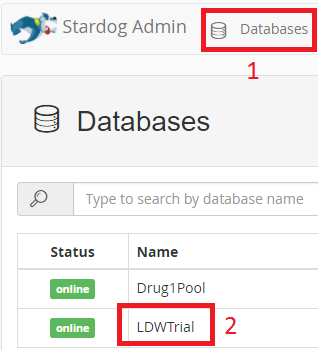
|  |  |
| --- | --- |
| **!** | Leave the graph open in Chrome. You will refer to it in the next section when creating queries. Ask for assistance if you accidentally close or refresh the Graph Editor window. |
| Image result for information icon | If you click Create TTL more than once, you will create new files that include a number in their name:  WhiteBoardTriples(1).TTL  WhiteBoardTriples(2).TTL  etc.  Make sure to use the highest numbered file in the next steps. |

1. Use Windows explorer to navigate to the TTL file.
2. Double click on the .TTL file to open it.
3. Follow along with the instructor's explanation of the TTL file content.
4. Find the various ***nodes*** and ***links*** you created in your graph.

|  |  |  |
| --- | --- | --- |
|  | Stop here and wait for the instructor. |  |

# Upload to Database

1. Open a new window in Google Chrome, keeping your graph editor window open.
2. Click the Stardog shortcut on the bookmarks bar.



1. Click "Databases" on the top menu
2. Double-click the row for **LDWStudy.**
3. In the upper right, click 



1. Select **Data | Add** from the Stardog menu:
2. Click **Browse** and navigate to, then select your WhiteBoard.TTL file:

C:\Users\phuseldw\Downloads\**WhiteBoardTriples.TTL**

1. Click **Upload**.
2. You should see the message: ***Success! Data added successfully.***

# Query your graph

# Show all triples

Find all Subject-Predicate-Object relations in your graph.

1. If the "Query Panel" is not displayed, click on the >\_Query menu item at the top of the Stardog display.
2. Select the Open Folder icon  in the Query Panel.
3. Navigate to the file C:\\_gitHub\LinkedDataWorkshop\CSS2018\scripts\SPARQL**\210-AllTriples.rq** and double-click on it to load it into the query window.
4. With the query loaded in to the window, click the  button. Scroll down to review the results.
5. Compare this view to what you expected after viewing the TTL file.

# How many people participate in the study?

Find the number persons participating in your study by counting the number of *participatesIn* relationships.

1. Load the SPARQL query **220-CountParticipants.rq**.
2. Examine the query while the instructor explains how it relates to your graph.
3. Execute the query.
4. Note how the ***coordinatingInvestigator*** is not included in the count.

# Find the names of people who participate in the study.

In this example you want to select the givenName of the Persons who participate in the study. Refer back to the your graph in the Graph Editor while creating the query.

1. The result should be the Person's name (givenName):

SELECT ?name

1. Your query must first identify Person nodes that have the ***enrolledin*** relation:

?person eg:participatesIn ?study ;

1. Then follow from that Person node using the givenName relation to find their name.

?person eg:participatesIn ?study ;

schema:givenName ?name .

1. Try writing and executing the query in the query window using the code snippets above.
2. If you have trouble, load and execute the query: **230-NameParticipants.rq**

# List the names of the people randomized to each treatment type.

Adapt your query in the previous exercise to create a list of the people within each type of treatment.

1. Change your SELECT statement to select treatment arm type and person name variables. Example:

SELECT ?treatArmType ?personName

1. Finding the ***givenName*** is the easiest part the WHERE statement.

?person schema:givenName ?personName ;

1. Refer to your graph in the Graph editor. You want to follow the path from the **?person** node along the **eg:randomizedTo** relation to a **?treatmentArm**.

?person schema:givenName ?personName ;

eg:randomizedTo ?treatmentArm .

1. Next, follow from the **?treatmentArm** along the **eg:treatmentArmType** relation to the **?treatmentArmType**.

?person schema:givenName ?personName ;

eg:randomizedTo ?treatmentArm .

?treatmentArm eg:treatmentArmType ?treatmentArmType .

1. Optionally you can add an ORDER BY statement after closing } to order your output.

ORDER BY ?treatmentArmType ?personName

SOLUTION: **240-NamesInTreatmentType.rq**

|  |  |  |
| --- | --- | --- |
|  | Stop here and wait for the instructor.  Presentation on Ontologies and Reasoning |  |

# Ontology

In this section you will add an ontology to the data and use it for reasoning.

# Review the Ontology

1. Double-click on the file C:\\_gitHub\LinkedDataWorkshop\CSS2018\data\**StudyOntology.TTL** to open into a text editor.
2. Follow along with the instructor's explanation of the file.

# Add the Ontology to the LDWStudy database

1. Select Data | Add from the Stardog menu.
2. Click **Browse** to and and select the file:

C:\\_gitHub\LinkedDataWorkshop\CSS2018\**data\StudyOnotology.TTL**

1. Click **Upload**.
2. You should see the message: **Success! Data added successfully.**

# Explore the data

1. Ensure reasoning is turned **OFF**.
2. Select **Explore | Class Hierarchy**
3. A SPARQL query appears in the window.
4. **Execute** the query and view the results.
5. Note the "types" of things that were added (but not yet applied to your study data), especially these terms:

HumanStudySubject

Investigator

1. Turn on the reasoner by clicking the **Reasoning** button.
2. **Execute** the query again and view the results.
3. Note how **Investigator** and **HumanStudySubject** are both types of **Person**. These are not part of your original study data but can now be used in queries to find these "types of things!"

|  |  |
| --- | --- |
|  | The instructor explains the source of these new triples. |

# Find the names of HumanStudySubjects in your study.

Remember: Your original data contained no definition of a HumanStudySubject.

1. Review the ontology file for how a HumanStudySubject is *inferred* using the ontology.

|  |  |
| --- | --- |
| IdeaIcon_clean_20mm | In SPARQL the predicate rdf:type can be shorted to a and interpreted as "is a".  Example:  You read: eg:Person11 rdf:type eg:HumanStudySubject  as: "Person11 ***is a*** HumanStudySubject." |

Instead of querying on the **participatesIn** predicate, as in:

?person eg:participatesIn ?study ;

You can now query using the *inferred* entity **eg:HumanStudySubject**.

?person a eg:HumanStudySubject;

1. Modify your original query in Section X.X (**230-NameParticipants.rq )** to useeg:HumanStudySubject.
2. Ensure Reasoning is turned on, then execute the query.
3. What happens if you turn OFF the reasoner and execute the query? (Don't forgot to turn the reasoner back ON!)

Solution: **340-NameHumanStudySubject.rq**

Bonus Question: How would you modify the query to select all Investigators in the study?

Hint: How is Investigator defined in the ontology file?

Solution: **341-NameInvestigator.rq**

# Write a reasoner-based query to find all People associated with the study (both human study subjects and investigators)

Your query should have a one-line statement the uses the superclass for both HumanStudySubjects and Investigators using the superclass of these two classes.

CAUTION: The superclass prefix is not "eg:" because that class is borrowed from another ontology!

Solution: **350-NamesAllPeople.rq**

|  |  |  |
| --- | --- | --- |
|  | Stop here and wait for the instructor.  Presentation on Merging Data |  |

# Data from all studies

# Create the Data Pool

1. Click on Admin Console at the top of the Stardog application. 



1. Click on **Drug1Pool** in the Databases panel.
2. In the upper right, click 
3. Open the query **410-PoolAllStudies.rq** into the query window an execute it. A blue progress line will display briefly at the top of the application while the query is executing.

|  |  |
| --- | --- |
| **!** | Having all attendees create the pool may cause bandwidth and connection issues. This happens, follow the alternative instructions provided by the instructor using the query:  **411-AltPoolCreator.rq**. DO NOT run this query unless advised by the instructor! |

1. After the query has completed, execute the query **415-PoolTripleCount.rq**. Note the number of triples and compare them to the number provided by the instructor to ensure you have the correct data.

# List Drug1 studies by phase.

1. Load the query **420-StudiesByPhase.rq** .
2. Execute the query and locate your study in the results.

BONUS Question: How would you alter the query to obtain the number of studies in each phase?

Hint: a) In the **SELECT** statement: (COUNT(**?phase**) **AS** **?count**)

b) Use a **GROUP BY** after the **WHERE**.

SOLUTION: **425-StudyCountByPhase.rq**

# How many HumanStudySubjects participated in all Drug1 studies?

1. Ensure the Reasoner is turned On.
2. Load the query **430-Drug1CountHSS.rq** and examine the query.
3. Execute the query.

BONUS Question : Alter the query to count **all people** involved in Drug1 Studies (not just HumanStudySubjects).

HINT: a) What is the superclass of HumanStudySubject?

b) Ensure you are suing the correct prefix!

Solution: **435-Drug1CountPeople.rq**

# How many women received active treatment (non placebo) across all Drug1 studies combined?

Use the graph of your study in the Graph Editor to determine the paths that need to be followed to obtain the solution.

SOLUTION: **440-Drug1WomenActiveTreat.rq**

# List all Persons in the studies.

List all Person IRIs, their givenName, their assigned Study, and their role in that study (HumanStudySubject or Investigator).

Review the results. Do you find anything surprising in the pooled results?

SOLUTION: **450-Drug1PersonsAllStudies.rq**

# END

Congratulations! You've made to the

---- END OF THE EXERCISES ----

Before you leave, please Sign out from the server using **Windows | Person Icon | Sign out**.



|  |  |  |
| --- | --- | --- |
|  | Stop here and wait for the instructor.  Presentation follows |  |

# Appendix 1: Course Resources

* [Introduction to Semantic Web](http://www.cambridgesemantics.com/semantic-university/introduction-semantic-web)

http://www.cambridgesemantics.com/semantic-university/introduction-semantic-web

* [What is Linked Data?](http://www.cambridgesemantics.com/semantic-university/what-linked-data)

http://www.cambridgesemantics.com/semantic-university/what-linked-data

* [Introduction to Linked data](http://www.cambridgesemantics.com/semantic-university/introduction-linked-data)

http://www.cambridgesemantics.com/semantic-university/introduction-linked-data

* [Learning SPARQL](http://www.learningsparql.com/)

http://www.learningsparql.com/

-------------------------------------------------------------------------------------------------------

[Hands-on Workshop Github Repository](https://github.com/phuse-org/LinkedDataWorkshop/tree/master/Annual2017-EU)

Course content is available at:

https://github.com/phuse-org/LinkedDataWorkshop/tree/master/CSS2018