PhUSE CSS

Linked Data and Graph Database

Hands-on Workshop

EXERCISES

Version 2.0

October 2017

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

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

Contents

[Introduction and Disclaimer 2](#_Toc484442224)

[Server Login and Preparation 2](#_Toc484442225)

[Exercises 3](#_Toc484442226)

[1. Neo4j 3](#_Toc484442227)

[1.1 Sketch the Data Model 3](#_Toc484442228)

[1.2 Transfer Model to Spreadsheet 5](#_Toc484442229)

[1.3 Upload to Neo4j 5](#_Toc484442230)

[1.4 Query and Visualize 5](#_Toc484442231)

[2. Resource Description Framework (RDF) 6](#_Toc484442232)

[2.1 Sketch the Data Model 6](#_Toc484442233)

[2.2 Transfer Model to Spreadsheet 8](#_Toc484442234)

[2.3 Create RDF (TTL) File 9](#_Toc484442235)

[2.4 Visualize the Data 9](#_Toc484442236)

[Demonstrations 11](#_Toc484442237)

[7. SDTM as LPG 11](#_Toc484442238)

[8. SDTM as RDF 11](#_Toc484442239)

# Introduction and Disclaimer

Instructions in this document are specific to the cloud server instance and files used for the PhUSE workshop. The exercises represent one of many possible approaches to the material and make no claim to be best or recommended method.

Your feedback is welcomed and encouraged. Please send your comments to:

**Tim Williams**

**tim.williams@PhUSE.eu**

# Server Login and Preparation

Login to the cloud server provided for the exercises. Instructions assume Windows OS on your local machine.

Connect to the server using Remote Desktop

1. In the search box on the taskbar, type **remote desk,** then select **Remote Desktop Connection** from the Programs section.﻿



1. Click **Show Options** if needed to show the fields for **Computer:** and **User name:**

You will be provided with an IP Address for the Computer: field. Everyone will use the same User name:

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

(Provided by Instructor)

**User name:** phusecss

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

(Provided by Instructor)

1. Click **Connect** after entering the Computer IP address and your username for the session.
2. Enter the password supplied by the instructor and click **OK**.

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

# Exercises

The exercises are presented in two sections.

**Section 1:** Labeled Property Graph Neo4j

**Section 2:** Resource Description Framework (RDF )

# Neo4j

# Sketch the Data Model

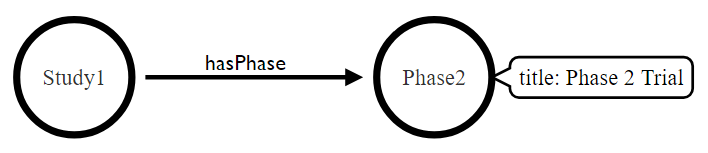
A very basic model of a small number of entities in a clinical trial is provided as starting point, to show how the model is translated into a graph.

1. Examine the initial model in Figure 1 **.** Note how nodes are joined to other nodes using relations <ADD more description here>Neo4j is capable of storing property:value pairs on relations. For simplicity, these exercises only cover property:value pairs on nodes and not for the relations.
2. Add more nodes and relations to the diagram, preferably using a pencil so you can erase and adjust the sketch as needed

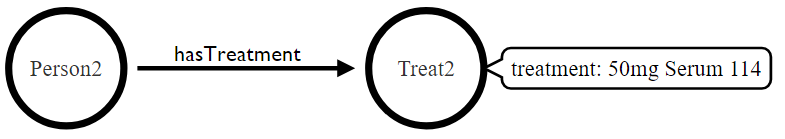
***It is recommended to add no more than six additional nodes to the model and fewer than twelve property:value pairs. This will keep the model manageable for later exercises.***

Consider the following examples if you are having trouble imagining new nodes and relations.

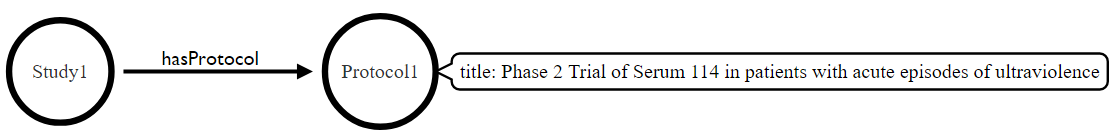
Example 1: Add study phase information to the existing study node.



Example 2: Add a new Person node (Person 2) and assign them to a new treatment node (Treatment2). Consider adding Gender value to the new person.



Example 3: Add a protocol to the study. How would the protocol fit into a description of the clinical trials design process?



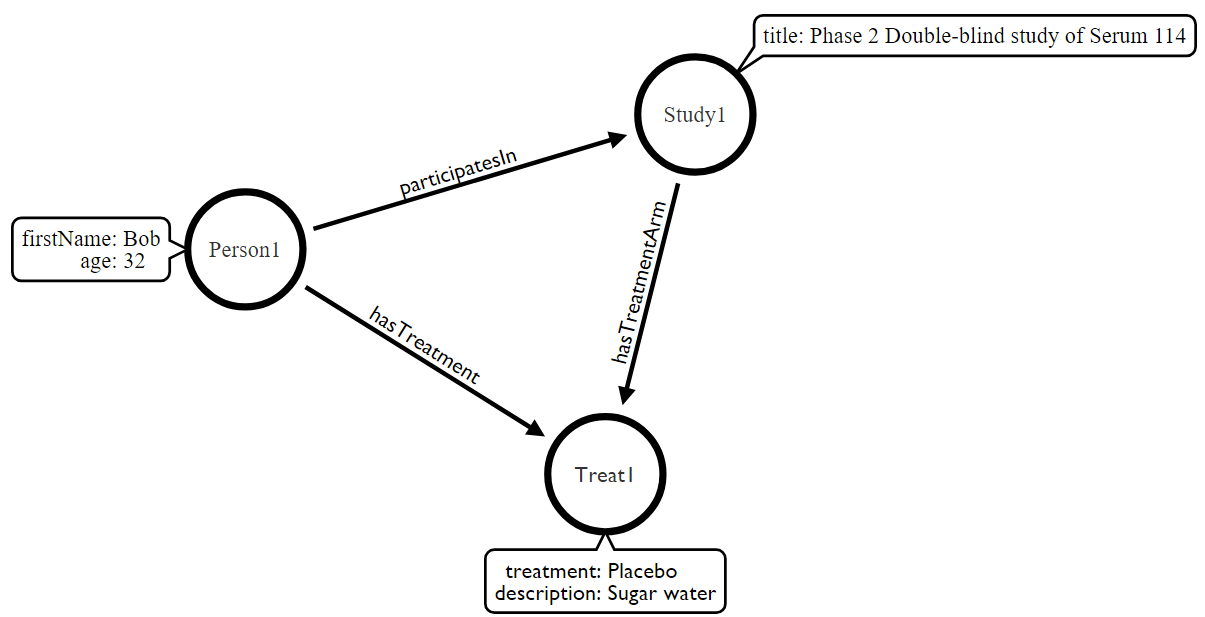


Figure 1 Neo4j Model Sketch

# Transfer Model to Spreadsheet

The data from the basic model will now be recorded in a spreadsheet to document the nodes, property:value pairs, and relations. See Figure 2 for a screen shot of the initial spreadsheet. Your personal copy of the spreadsheet is located at:

/<PATH>/**Neo4jModel.XLSX**

<Add description of how to translate nodes, property:value pairs, and relations into the spreadsheet>

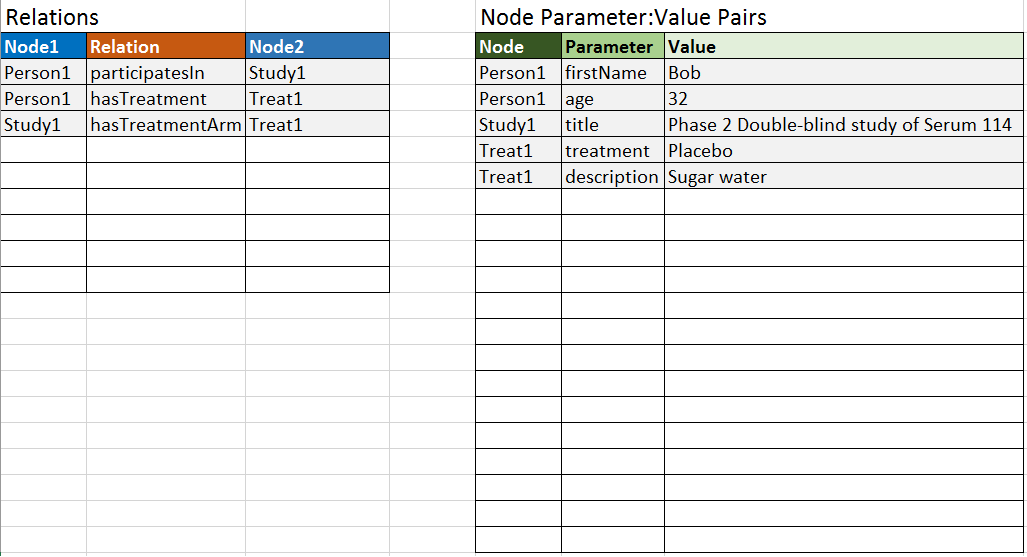


Figure 2 Neo4j Spreadsheet

# Upload to Neo4j

start Neo4j <..yada yada upload instructions, etc.. The easiest solution may be to execute an R script that reads the data from the spreadsheet and uploads it directly into the (running) Neo4 instance using RNeo4j.>

# Query and Visualize

<Show how to execute the pre-prepared queries in the Neo4j "Favorites" sidebar>

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

This is the end of exercises for Neo4j. RDF will be introduced before returning to the exercises in the next section.

# Resource Description Framework (RDF)

The same concepts you modeled using Neo4j will now be represented using RDF.

# Sketch the Data Model

Similar to the Neo4j exercises, an initial model is provided as a starting point for the data.

1. Compare the base model for RDF in Figure 3 with the base model for Neo4j (Figure 1). RDF does not use property:value pairs on nodes and edges. Observe how *Subjects – Predicate --> Object* relations attach values like the age (32) to the Person1 node.
2. Add the same data values to the RDF model (Figure 3 ) that you added to NEO4J by drawing them onto the diagram with pen or pencil. Pencil is preferred so you can make changes.

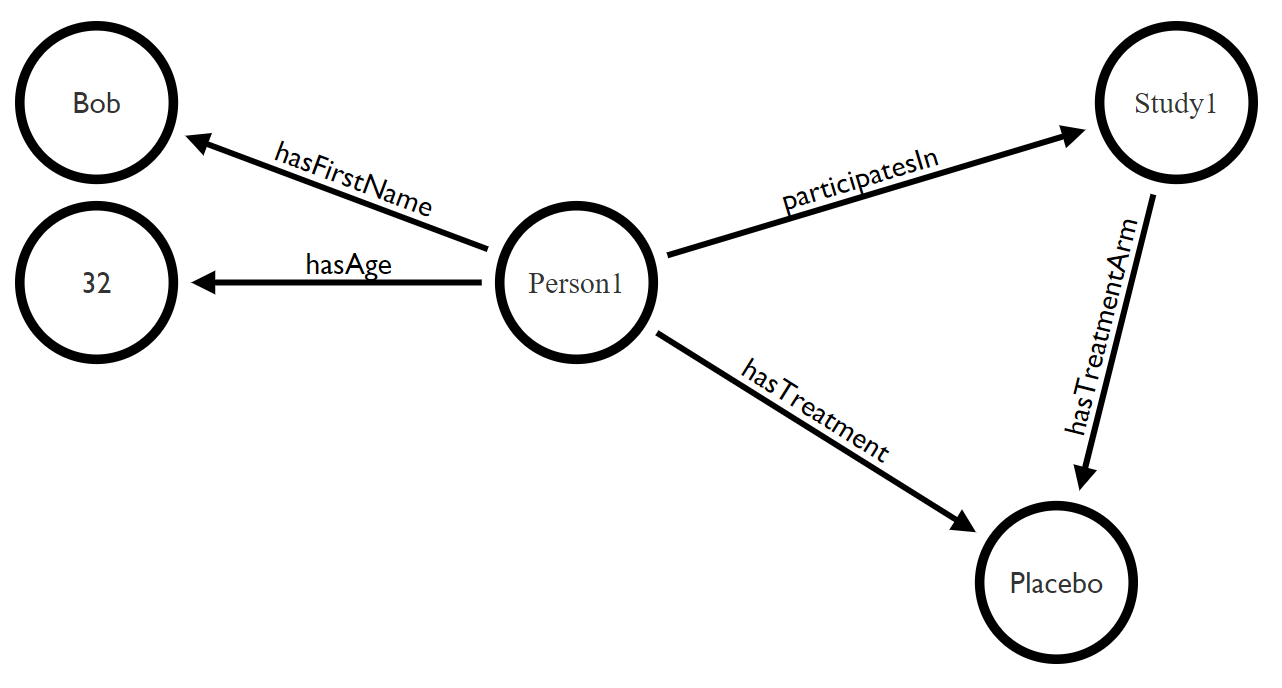


Figure 3 RDF Model Sketch

# Transfer Model to Spreadsheet

You must now convert the model to a machine-readable representation so it can be converted to RDF triples. This is accomplished by transforming your sketch of the model into a spreadsheet, similar to the Neo4j exercise.

1. Double click on the spreadsheet file /<PATH>/**RDFModel.xlsx** to open it**.**
2. The first rows of the spreadsheet contain the data from the base model, before you added your own data. You will add your new data below the rows that are shaded in grey ( Figure 4 ) .
3. You must determine the ObjectType for the Object in each Subject --Predicate --> Object relations. Many different data types are available in RDF. The exercises use only ***string***, ***int***, and ***uri*** types for simplicity.

Recall how it is possible for the Object in one relation to be the Subject in another relation, forming a series of node-to-node relationship paths. These types of nodes are coded as the 'uri' ObjectType because they can form a path of connected links.

|  |  |
| --- | --- |
| **string** | Character values with no outbound link.  Examples: "Bob", "Protocol for Study 123", "Male" |
| **int** | Integer values with no outbound link. Example: 32 |
| **uri** | Object nodes that link to other nodes in your existing model, or represent things that ***could*** participate in another link relationship. |

Ask the instructor for assistance if you are unsure which ObjectType should be assigned to a node in your model.

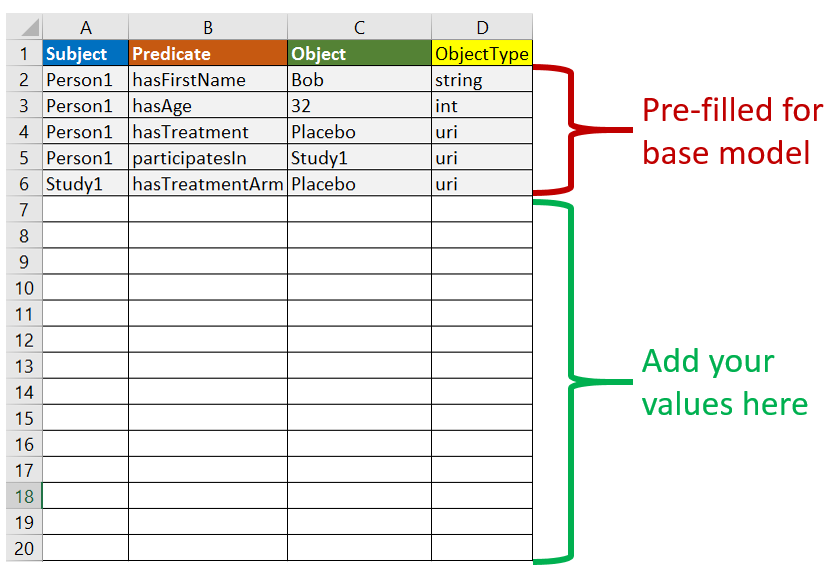


Figure 4 RDF Spreadsheet

1. Confirm all nodes and relations are present in your spreadsheet. You may have several nodes listed as both Subject and Object.
2. Save the file using the **File | Save** menu.

# Create RDF (TTL) File

An R script is used to convert the spreadsheet data into RDF and save the result in a file with a .TTL extension (N3 Turtle serialization).

1. Double click on the R script to open it into R Studio.
2. …<PATH>/r/ **SpreadsheetToRDF.R**
3. Execute the R script by clicking on the Source toolbar button 

|  |
| --- |
| The R script also validates the TTL file. You receive no message if the file is valid RDF. Error messages follow the format that identifies the location and type of error:  ERROR [line: *n*, col: *n*] *type of error*  Contact the instructor or assistant if you encounter an error message. Open the **RDFModel.TTL** file in the /data folder into Notepad++ for troubleshooting. |

1. The script outputs RDF into the file located here:

…<PATH>/data/**RDFModel.TTL**

1. Double click the file to open it into Notepad++. The file will appear similar to Figure 5. Observe how the model in Figure 3 was translated to the data in Figure 4 and lastly into the RDF representation Figure 5 .

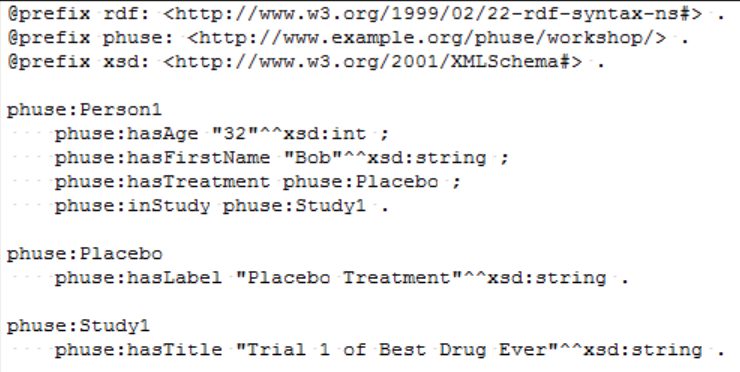


Figure 5 TTL file RDFModel.TTL

# Visualize the Data

A R script is used to load the TTL file and convert it to a force network graph.

1. Double click on the R script to open it into R Studio:

…<PATH>/r/**ViewRDFModel.R**

1. Execute the script to view the force network graph of the data. The graph will look similar to Figure 6

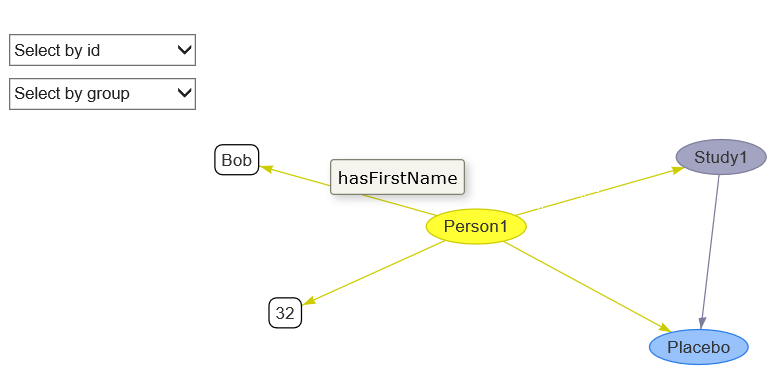


Figure 6 Force network graph of base model

1. Explore the graph by clicking on nodes and using mouse-over to show the relations. You may also use the drop-down selections for highlighting node categories and groups.

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

---- END OF EXERCISES ----

# Demonstrations

In this section the instructors will demonstrate example data from SDTM domains converted to the two different types of graph databases. Files are provided so you may follow along if you wish.

# SDTM as LPG

<content to be added>

# SDTM as RDF

<content to be added>