



## Neo4j Assignment

19.10.2021

### 1 Introduction

In this assignment, you are asked to build a graph representing units offered in a university and their relationships. You also need to design a few queries to show that the graph can be used to check degree and unit rules. There is no given data set; you should come up with your own data to demonstrate your model and query. You are encouraged to use degrees and units offered in this university as sample data but you can use any unit or degree.

### 2 Problem Domain

You are asked to model some hypothetical **degrees** and **units** offered in a single School. Each **degree** has a code, a name, and completion requirements. For simplicity, we assume a degree's completion requirements only include rules for total credit points and credit points at each year level. Following the naming convention in our university, the first year units are referred to as 1000 level units, second year units as 2000 level unit, and third year units as 3000 level units. Below are sample degree requirements for a three year degree. You can design your own degree requirements that are similar to it.

- Complete at least 144cp comprising:
  - at least 48cp at 1000 level;
  - at least 36cp at 2000 level;
  - at least 36cp at 3000 level; and
  - the rest at any level.

Each **unit** has a unique code and a descriptive title. It is designed for a particular year level and has a specific number of credit points. Units may have different credit points. A typical unit in our school has 6 credit points. We also have some units that have 12 credit points or 2 credit points.

Some units have a list of prohibited units. Students are not allowed to enrol in a unit if they have completed any of the prohibited units in the corresponding list.

Many units have prerequisite units. Students need to satisfy the prerequisite rule to be enrolled in a unit. The prerequisite rule is usually given as a Boolean expression. Most units require a single prerequisite unit and have a list of similar units as prerequisites.

Below are details of a sample unit COMP3027:

```
code: "COMP3027",
title: "Algorithm Design",
cp: 6,
level: 3000,
prerequisites: "COMP2123" OR "INFO1105"
prohibited: "COMP2007" OR "COMP2907" OR COMP3927"
```

The unit COMP3027 requires one unit as prerequisite, which could be either “COMP2123” or “INFO1105”. The unit also has a prohibition list containing three units. A student cannot enrol in COMP3027 if they have completed any of "COMP2007", "COMP2907", or COMP3927" before.

Some units require more than one unit as prerequisites. This rule is expressed using AND operators. For instance, a unit with the prerequisites expression ("COMP2017" OR "COMP2129") AND ("COMP2123" OR COMP2823) requires a student to complete one unit from the list ["COMP2017", "COMP2129"] and another unit from the list ["COMP2123", "COMP2823"] before enrolling in this unit. There could be units with more complex prerequisite Boolean expressions but your graph only needs to model the following two basic forms.

- $u1 \text{ OR } u2 \text{ OR } u3 \dots$
- $(u1 \text{ OR } u2 \text{ OR } u3 \dots) \text{ AND } (u4 \text{ OR } u5) \text{ AND } \dots$

### 3 Graph Requirements

Your graph should model

- at least one degree (three or four year) with a set of completion requirements;
- enough units at 1000, 2000, and 3000 or above level to satisfy that degree’s completion requirements (there should be at least three or four units per level);
- some units with no prerequisites;
- some units with a list of prohibition units;
- some units with a list of prerequisite units linked by OR operators;

- some units with two or more prerequisite units linked by OR and AND operators; and
- a chain of units linked by two or three prerequisite relationships.

An example chain with three prerequisite relationships is as follows: COMP3520 has COMP21017 as a prerequisite, which in turn has INFO1103 as a prerequisite, which in turn has INFO1110 as a prerequisite.

## 4 Neo4j Browser Guide

You are asked to prepare a **custom browser guide** to set up the graph and to create use cases for rule checking. A browser guide consists of a number of slides. Most slides should contain one or more executable Cypher queries and a textual description. It is written as a partial HTML document using the basic structure described in the **Creating remote browser guides in HTML** document. A browser guide can be hosted remotely or locally. A locally hosted browser guide should be put under the respective project folder and has the file extension “neo4j-browser-guide”.

Neo4j’s built-in guide on Movie graph is an example browser guide with many slides, most with clickable Cypher queries. You can find the source file “about-movies.neo4j-browser-guide” in the “File” section of Neo4j Desktop’s project page (see Figure 1).

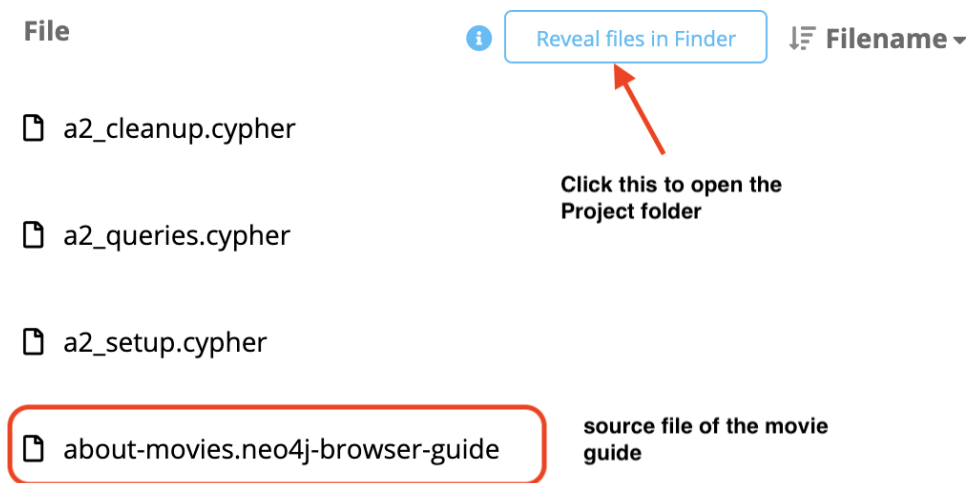


Figure 1: File Section of Neo4j Desktop

To access the actual file, click “Reveal files in Finder” if you are using macOS, or “Reveal files in File Explorer” if you are using Windows, to open the project folder in a finder or file explorer window. This folder is the place you should put your own browser guide in; doing so allows Neo4j Desktop to load the guide automatically.

You can use any text editor to open “about-movies.neo4j-browser-guide” and inspect its contents.

Your guide should have a similar structure. It must contain the following slides:

- **Slide one:** Graph Building. In this slide, you should provide one or more queries to build a graph containing units, degrees, and their relationships. You should prepare a data file containing units information and load that file to build a graph units information. You may write a separate query to add degree information. In the textual description section of this slide, briefly describe the nodes and relationships in the graph.
- **Slide two:** Graph Inspection. In this slide, include some queries to show that your graph contains units with the following features:
  - has no prerequisite
  - has a prohibition list
  - has a list of prerequisites units linked by OR operators
  - has a list of prerequisites units linked by AND and OR operators
- **Slide three:** Prohibition Rule. In this slide, write a query to insert a student’s information in the graph. The student should have one basic property such as a name and should have completed some units. Write a query to check if the student is allowed to enrol in a unit based on its prohibition rule. You should prepare two unit codes: one that satisfies the prohibition rule and one that does not satisfy the rule. The unit codes to be used should be included in the textual description part of this slide.
- **Slide four:** Prerequisite Rule. This slide also needs a student’s information. You may reuse the one added in slide three if appropriate. Otherwise, write a query to insert another student’s information in the graph. The student should have completed some units. The units completed by the student should
  - satisfy the prerequisites of a unit (*A*) with only OR linked prerequisites;
  - satisfy the prerequisites of another unit (*B*) with both AND and OR linked prerequisites; and
  - does not satisfy the prerequisites of a third unit (*C*) with both AND and OR linked prerequisites.

You should prepare three unit codes that correspond to the above three units *A*, *B* and *C*, and include them in the textual description part of this slide. [Write ONE query to check the prerequisite rule. The query should have a default value to check the prerequisite of unit \*A\*. It should be able to check unit \*B\* and \*C\* by replacing the unit code accordingly.](#)[\[updated: 28/10/2021\]](#)

- **Slide five:** Graduation Checking. Insert another student’s information in the graph. This student should have completed the degree’s total credit points requirement but does not meet some level’s credit points requirements. Write a query to run the check and print out a list of Boolean values, each representing the status of a rule.
- **Slide six:** Indirect Prerequisite. In this slide, write a query to find all 1000 level indirect prerequisite units of a given unit and print out the prerequisite chain from the unit to the 1000 level indirect prerequisite unit. An indirect prerequisite unit could be one or more units away from the given unit in a prerequisite chain. You should prepare two unit codes: one that has a maximum chain length of 2, the other has a maximum chain length of 3. The unit codes should be included in the textual description of this slide.
- **Slide seven:** Foundational Unit. In this slide write a query to find the unit(s) appearing in most other unit’s direct or indirect prerequisite list. In the textual description part, briefly describe the expected result of this query.
- **Slide eight:** Clear the Graph. In this slide write a query to delete all nodes and their relationships in the graph.

In any slide that requires multiple sets of input, you may use form field(s) to automatically populate the input in the text on the slide and in the query. You can find examples of using form fields from many slides in the movie graph guide.

A custom browser guide using week 8 json loading exercise is provided as a sample. The file (wk8-json-guide.neo4j-browser-guide) should be copied to the project folder where “about-movies.neo4j-browser-guide” is located (Figure 2).

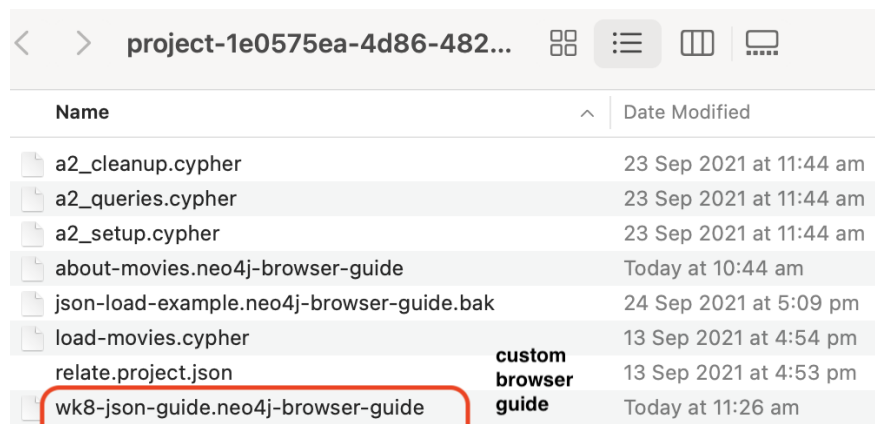


Figure 2: Project folder with custom browser guide

Once copied, you can see the sample guide in the file section (Figure 3). Click the “Open” button next to the file name to load the guide in the Neo4j browser, which should look similar to Figure 4. The guide has seven slides; most slides contain queries you have

used in the week 8 lab. The last slide introduces a new query to check if two persons are siblings. It also demonstrates how to use form fields to dynamically supply values in a Cypher query.

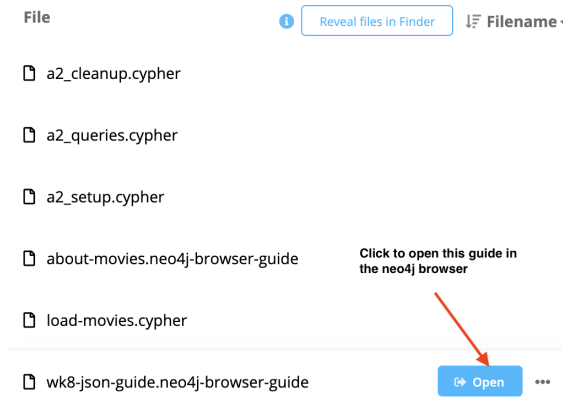


Figure 3: Custom guide in File section



Figure 4: Custom guide opened in browser

## 5 Deliverable and Submission Guidelines

The final deliverable of this assignment consists of the following:

- A zip file containing the browser guide file and the data file due on Monday 23:59, 8<sup>th</sup> of November, 2021.
- A pdf document of up to 2 pages due on Monday 23:59, 8<sup>th</sup> of November, 2021. The document should contain a screenshot of the graph (either a complete one or a partial one) and a brief description of the nodes and their relationships.
- A **10 minutes** zoom demo with your tutor in week 13. In the demo, the **tutor** will load your submitted browser guide and run queries on all slides. The demo will run in week 13 and during the study break.

## 6 Mark Distribution

- Graph model and the brief description: 5 points
- Browser Guide: 15 points.
  - 2 points each for slide one to seven
  - 1 point for overall quality