Homework 4

Spring 2020: Introduction to Database

Due Friday **May 10, 11:59 PM**

Overview

This assignment is the last assignment for this course. It takes the place of a final exam or project. This homework explores a NoSQL database, neo4j. There will be some written answers and you will have to copy and paste some queries and screenshot into the answer document you submit. We will only accept **typed** answers.

This assignment is broken up into four parts:

1. Downloading the neo4j server
2. Graph Databases 101
3. Using the Movie Database
4. Create your own Database

Your submission should be a single PDF named [uni]_hw4.pdf.

1. Downloading neo4j

Go to https://neo4j.com/ and click 'Download neo4j'

- You should be downloading neo4j for Desktop, which is the default. Please do **not** download the Enterprise or Community Edition as the instructions for setup will differ.
- You will have to provide name, email, and company (use Columbia University) in order to start the download.
- Follow the installation instructions provided, the steps are also listed below:

See the following pages for a Mac and Windows guide

For Mac

1.1. Open the Neo4j Desktop installer

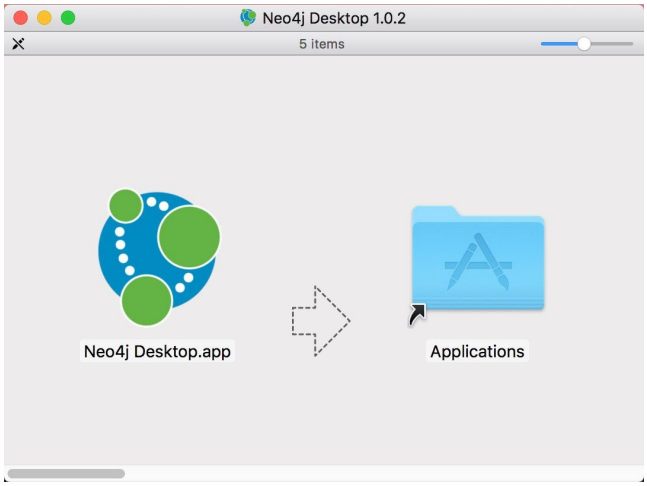
In your Downloads folder, locate and double-click the Desktop .dmg file to begin install.

1.2. Save the App to Applications

Drag and drop the Neo4j Desktop icon over to the Applications folder.

1.3. Open the Neo4j Desktop

Locate the Neo4j icon in Applications and double-click to launch.



For Windows

1.1. Open the Neo4j Desktop installer

Double click the neo4j-desktop-offline-1.2.7-setup.exe application that was just downloaded. Follow the steps on the Neo4j Desktop Setup helper, and install Neo4j Desktop. Click ‘Finish’ when done.

1.2. Open the Neo4j Desktop

Neo4j should automatically run when install is finished. If it doesn't, please search for the desktop app and double click to open it.

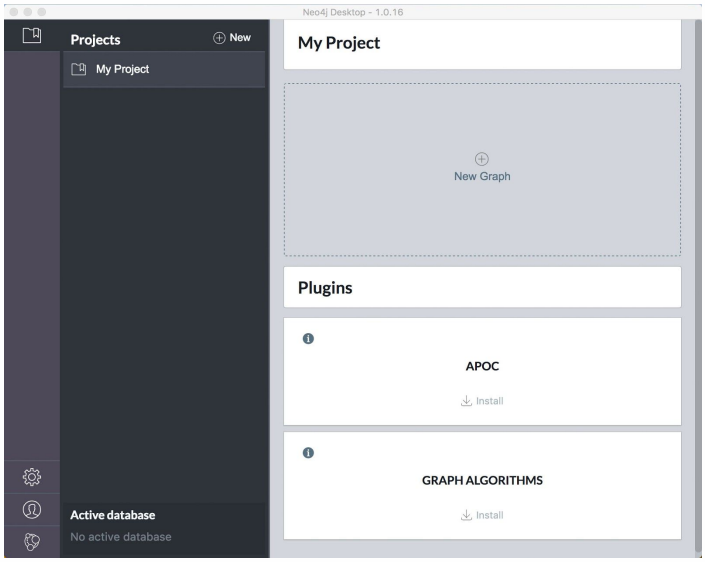
For Mac & Windows

2.1. Authorizing the Desktop App (first launch only)

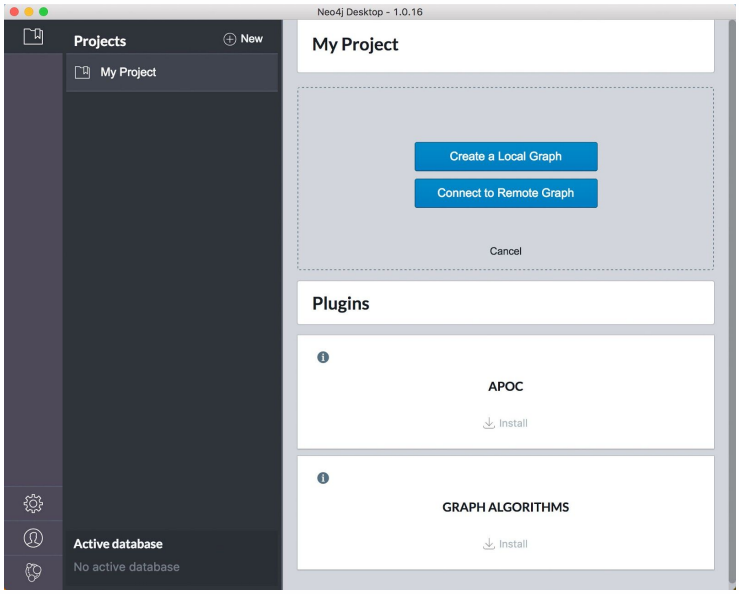
When first launching the app, it will ask for Software Key or a name/email again. You can ignore the Software key part and just put in your name and the email you used to first download it in again.

2.2. Create and start a Database

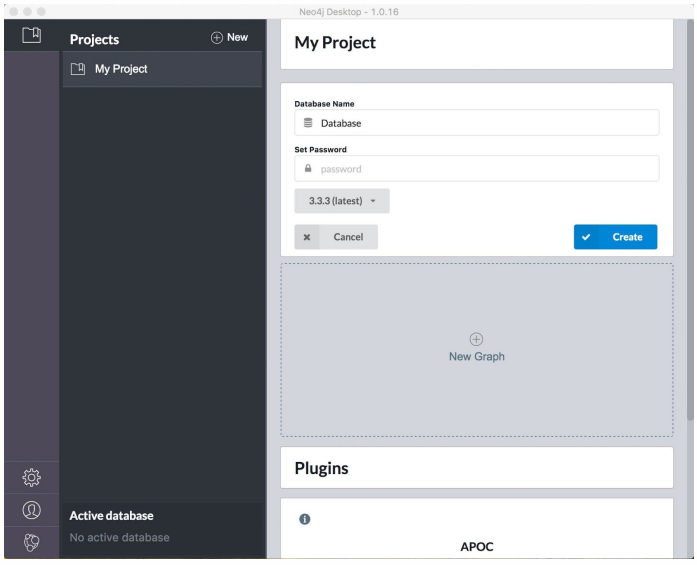
Click the "New Graph" button. May say "Add Database" on Windows.



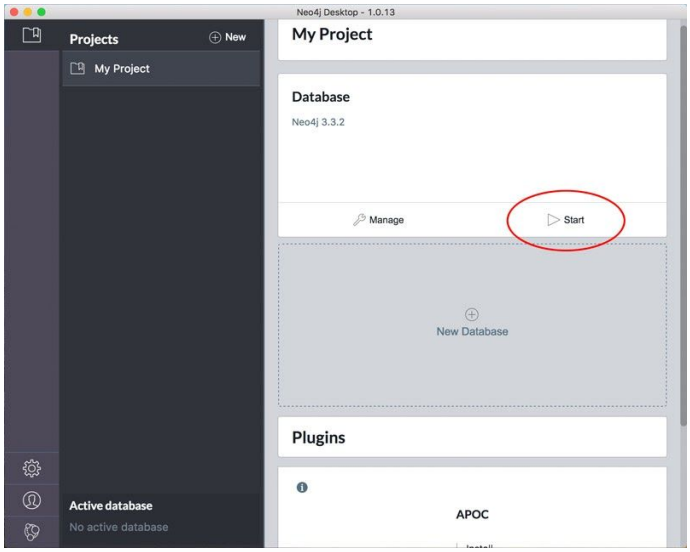
This will turn into two blue buttons. Click the one labeled, "Create a Local Graph."



Enter the password of your choice in the “Set Password” field. Then, click the blue button labeled “Create.”

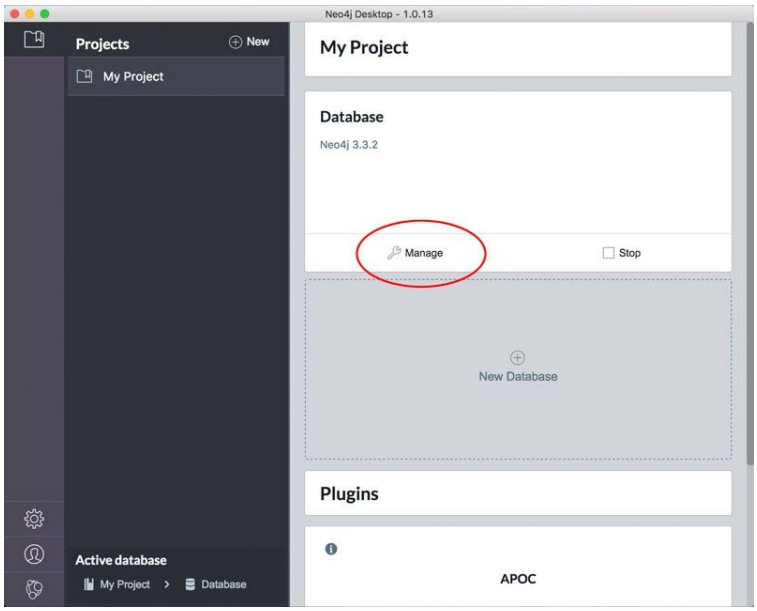


The “Create” button will soon be replaced by a “Start” button. Click it.

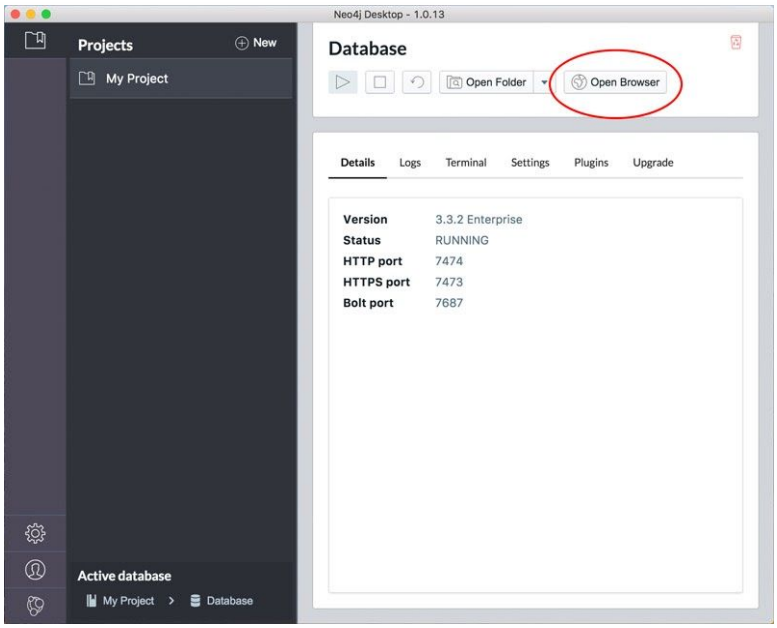


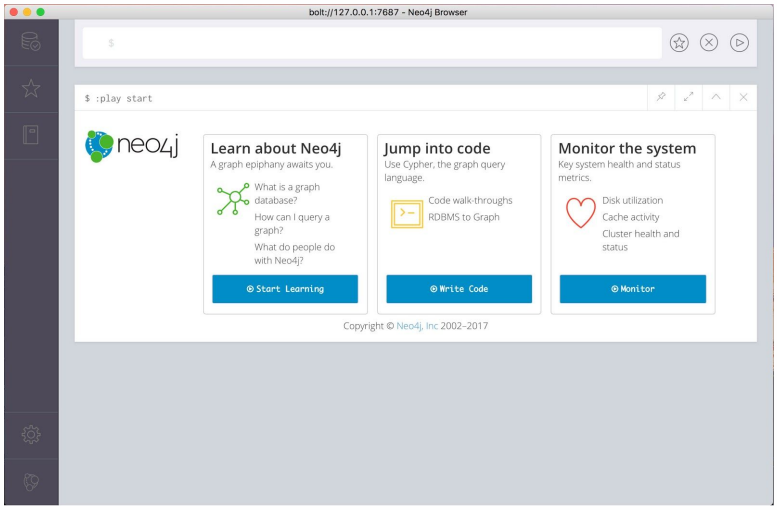
2.3. Open the Neo4j Browser

Once the database has started, click the “Manage” button.



Click on the “Open Browser” button that will appear in the Database Management area. The browser will open in a new window, as shown below.

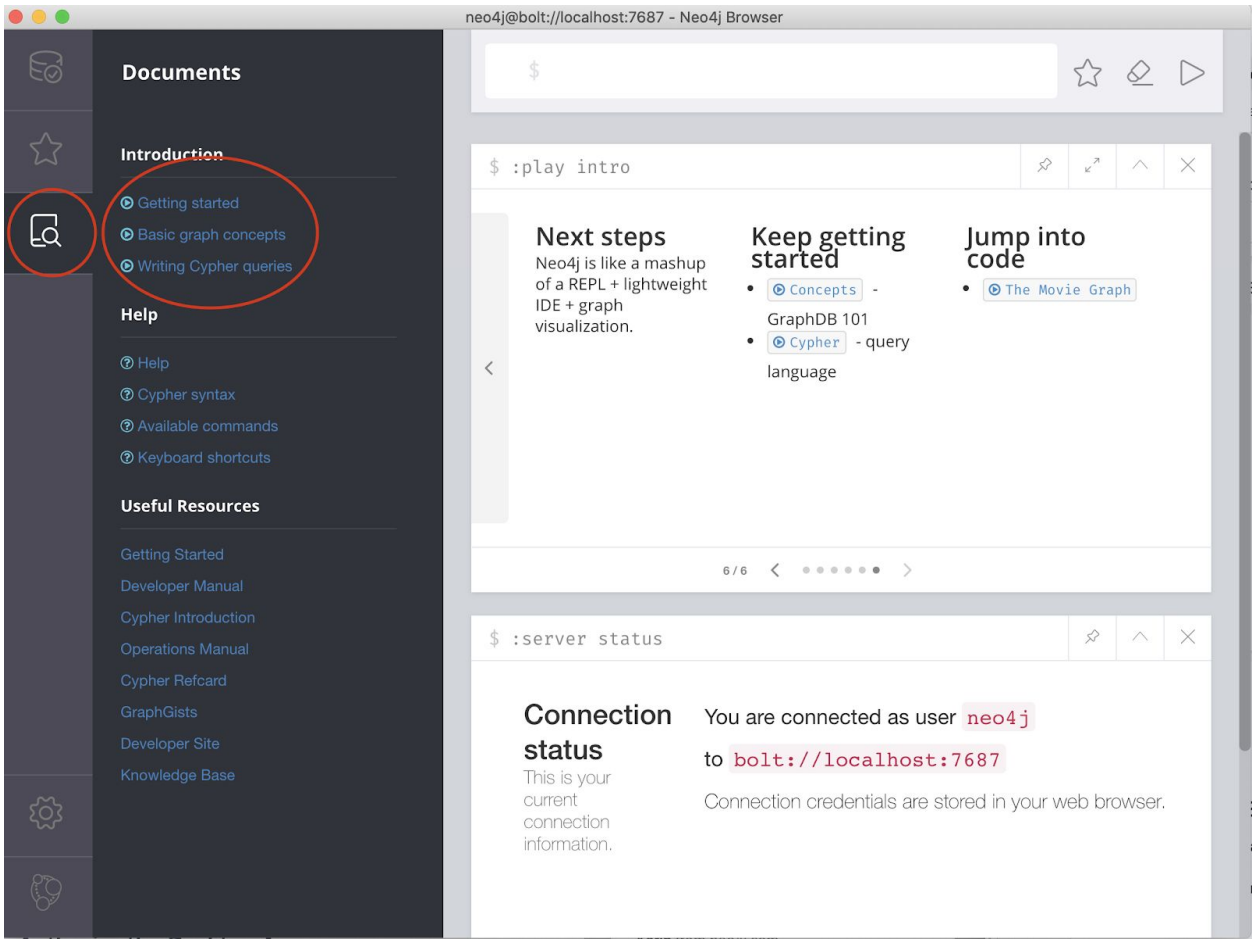




The Neo4j Browser is a tool for developers to explore their Neo4j database, execute Cypher queries and see results in tabular or graph form.

2. What is a Graph Database?

Once you have your instance started, select the notebook icon from the left sidebar and play the three tutorials under introduction



2.1. What are some pros and cons of Graph Databases? Please give an example of when you would want to use a graph database, when you would want to use a traditional relational database.

Answer:

2.2. Please describe the following terms (2-3 sentences/bullets each):

- Nodes
- Labels
- Relationships
- Properties

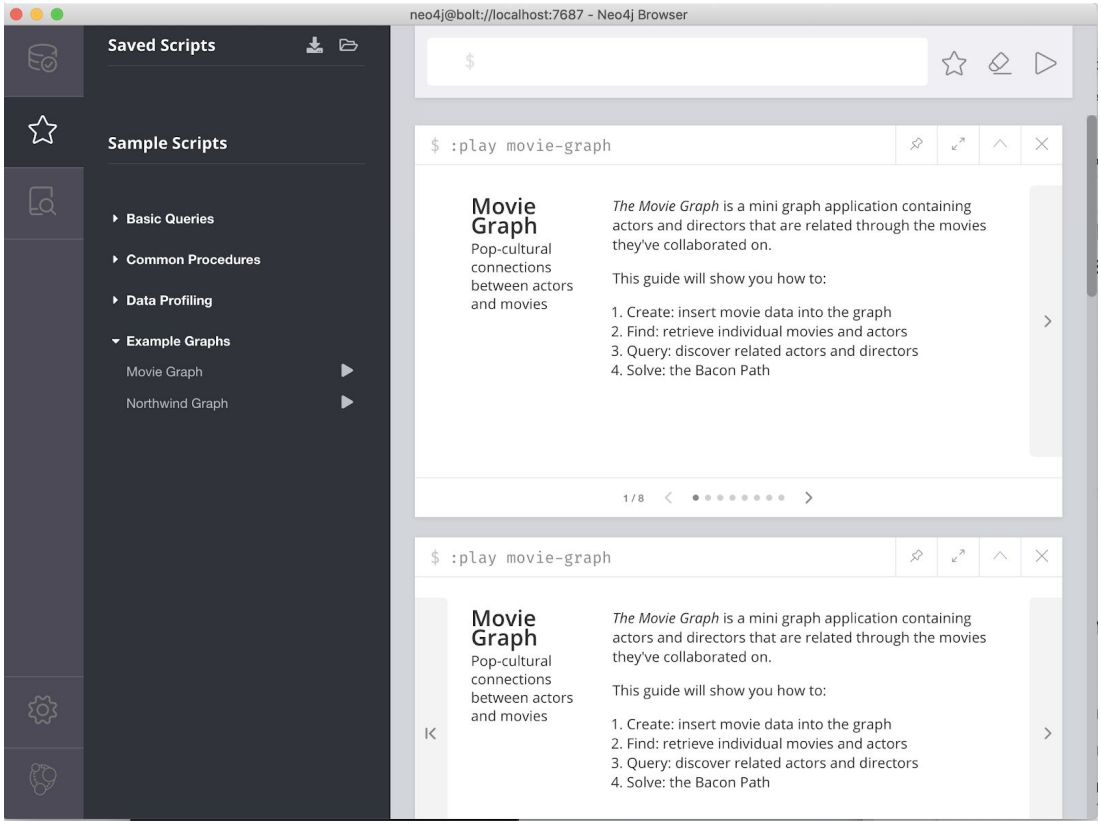
Answer:

2.3. What query language does neo4j use? Please list some similarities and differences between this query language and SQL.

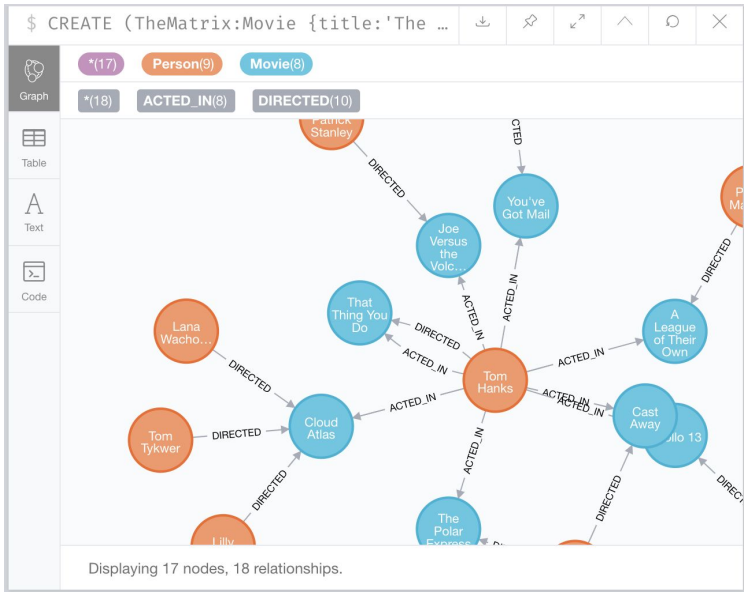
Answers:

3. Using neo4j

We will be using neo4j's example Movie Graph database for the following questions. It can be found under the star icon and then under "Example Graphs." Run through the brief tutorial before beginning the questions. Do **not** delete the movie graph at the end of the tutorial



Include a screenshot that shows that you ran the first query and switched to the graph view. It should look something like this



Here are some other example queries to get you started. Notice that n, m, and r are variable choices. You can make these whatever you want as long as you are consistent. Note also that nodes are called using parentheses (…) while relationships are called using brackets [ ]. All relationships are directed i.e. “->”

Examples

Display Tom Hanks:
MATCH (n:Person) where n.name='Tom Hanks' return n

Find all of the movies that Tom Hanks acted in:
MATCH (n:Person)-[r:ACTED_IN]->(m:Movie) where n.name='Tom Hanks' return n,r,m

Questions

3.1 What does the following query return? Run it on your neo4j instance and include a screenshot below.

```
MATCH (n:Person)-[r:ACTED_IN]->(m:Movie) where n.name='Tom Hanks'
      return n.name, r.roles, m.title, m.released
```

Answer:

3.2 Write a query to return every movie that Kevin Bacon was in between 1990 and 2000. Include your query and screenshot.

Answer:

3.3 Find the director who directed both “The Da Vinci Code” and “Apollo 13”. Include your query and screenshot.

Answer:

3.4 Generate all of the reviews on “Cloud Atlas” include the name of the reviewer and what they said. Include your query and screenshot.

Answer:

3.5 Find all movies in which someone involved was both a director and an actor. List the names of these people and their movies. There may be more than one person per movie or more than one movie per person! Include your query and screenshot.

Answer:

3.6 Robin Williams wants to have a party with all of his co-actors across all of the movies that he has acted in (his A-list). He orders too much food and realizes he has to extend the guest list to make sure it is all eaten. His B-list of party invites now includes the co-actors of his co-actors. What actors are on the A-list and what actors are on the B-list. Be careful, no actor should be on both lists! Include your query and screenshot.

Answer:

4. Create own data

For the final portion of the homework, come up with a dataset that you think would be best represented as a graph database. You must include at least two different types of nodes and four different types of relationships and create a database in neo4j! Each node should have at least three properties.

4.1 Why is it better to write this data in a graph database instead of a SQL database?

Answer:

4.2 Write out the different types of nodes, properties and relationships

Answer:

4.3 Write two example queries that someone looking at this dataset might need. Include the queries and the results that they yield.

Answer: