# PILOT STUDENTS – Oct/Nov 2015 (this applies to you)

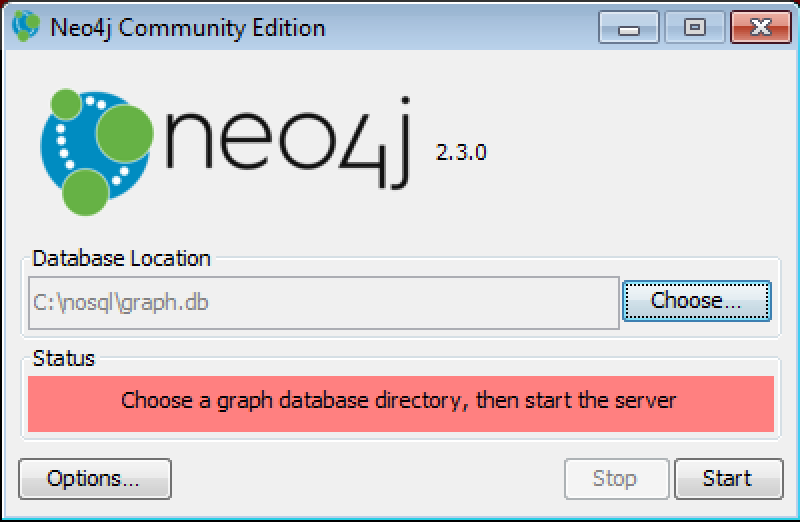
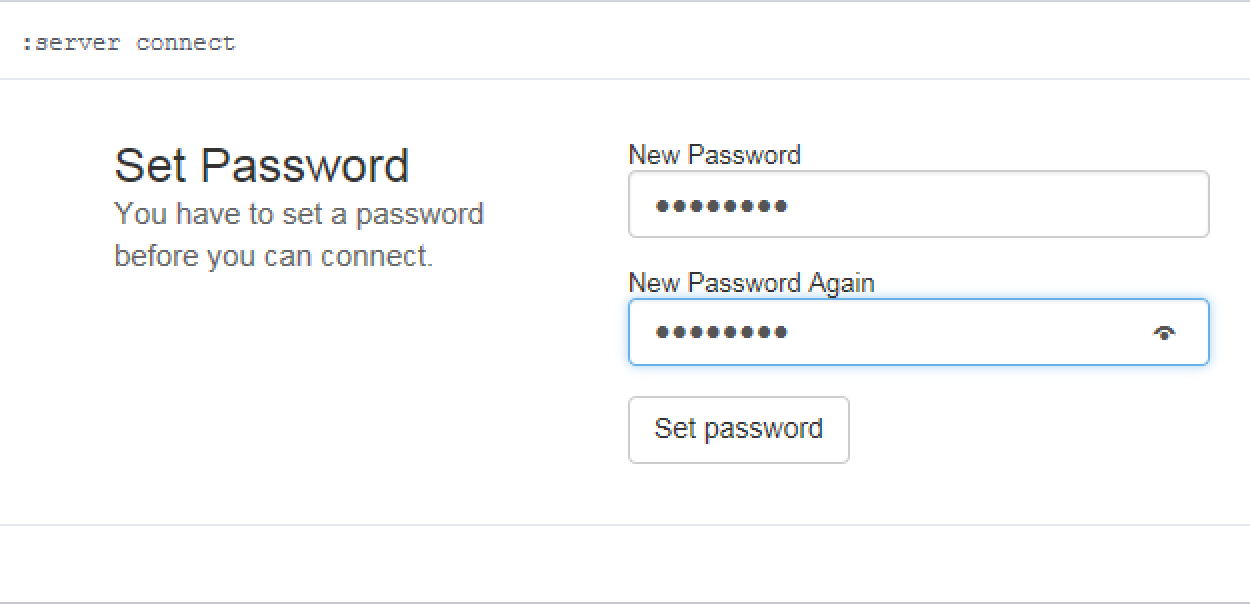
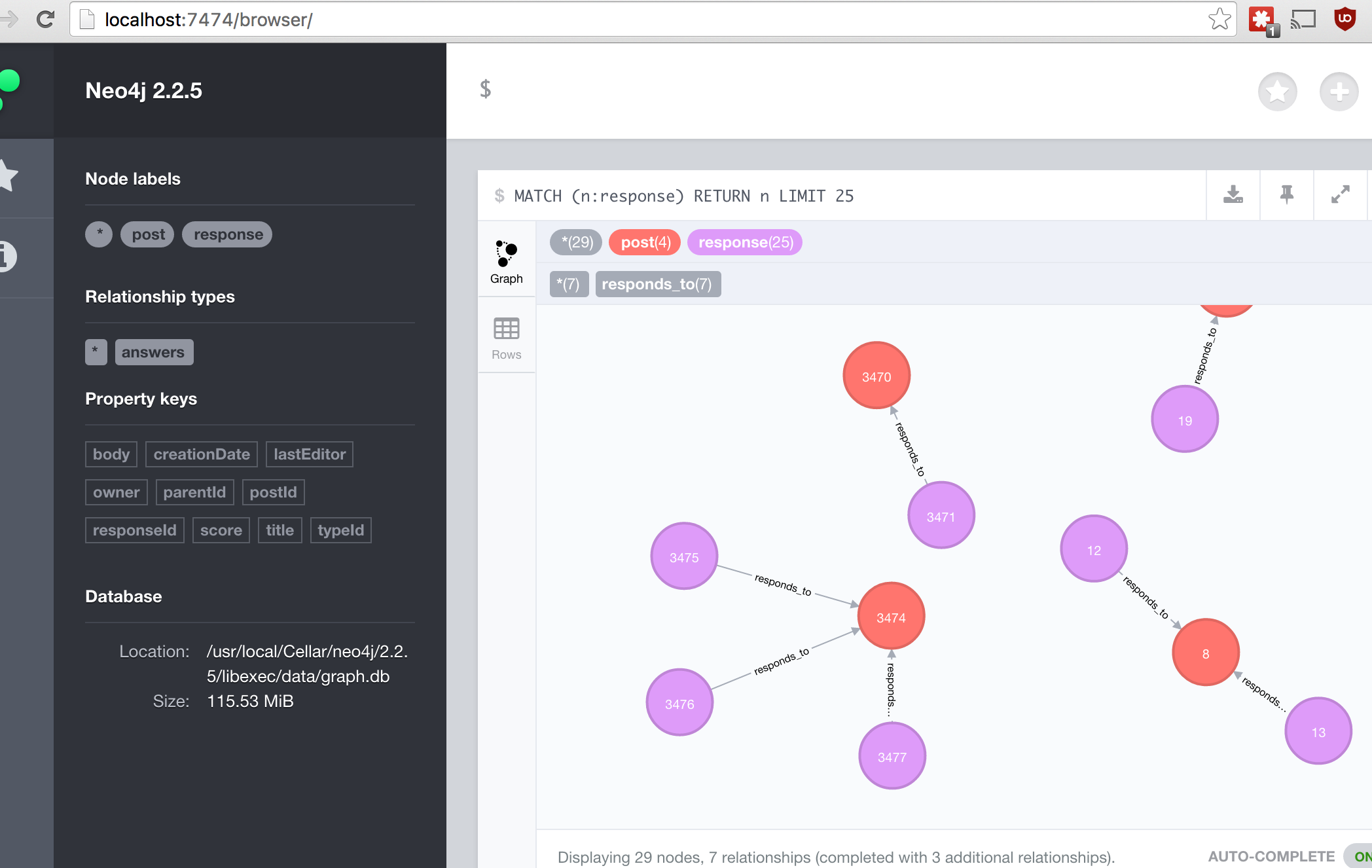
# (1) When you see an instruction to upload something, upload it to your desktop. If faculty needs it, they will ask for it. Contact info is in your FAQ.

# Intro to NoSQL Lab 5: The Graph / Neo4J HandsOn

## What Am I Supposed to Do?

1. Follow the instructions of the Lab.
2. Periodically, you’ll be instruction to **STOP** and do what is requested. Typically, we ask you to copy a screen shot. **YOU MUST DO THIS IF YOU WANT TO PASS THE COURSE!**

## BEGIN LAB:

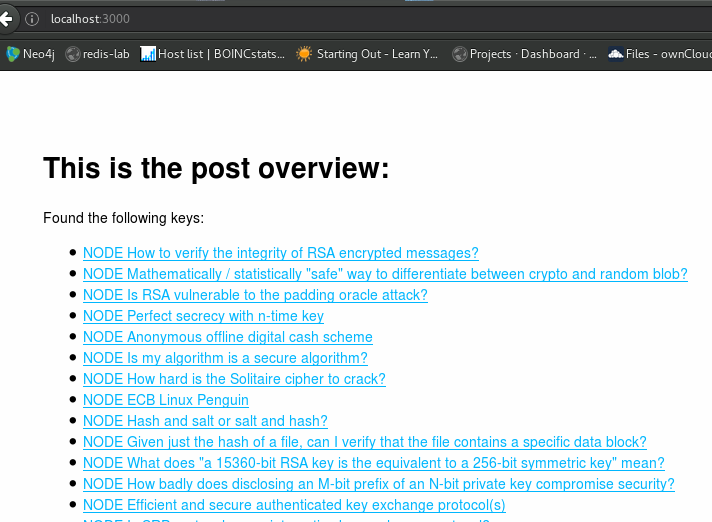
1. Make sure you have your node.js Setup from Lab 1 up and running. You’ll need the following packages unpacked and **ready** (i.E. npm install executed within their directory)
   1. nosql-lab-dataimport: <https://innersource.accenture.com/nosql/nosql-lab-dataimport/archive/master.zip>
   2. nosql-lab-neo4j: <https://innersource.accenture.com/nosql/nosql-lab-neo4j/archive/master.zip>
2. Install Neo4J
   1. You can get it from [neo4j.com](http://neo4j.com/download/). The Community Edition is sufficient
   2. Install & Launch   
      
   3. The first time you open the administration UI of Neo4J on [http://localhost:7474](http://localhost:7474/) you’re required to set a password. Ensure that the combination of username neo4j and password nosqllab is set! Otherwise you have to modify the user/pass in the dataloader as well as the sample-app  
      
   4. Check out the other cool features of Neo4J on the Web-UI
3. Import the data into Neo4J
   1. In the Dataloader directory execute: nodejs neo4j.js  
      (if you get strange errors re-check point 1 of this lab…there could be dependencies missing)
   2. Check in Neo4J wether data is there. Hint: Refresh the UI, you should see 4 labels which you can double click and you’ll get stuff

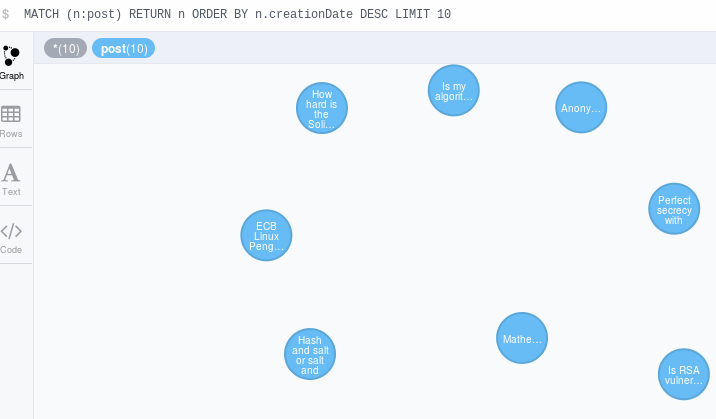
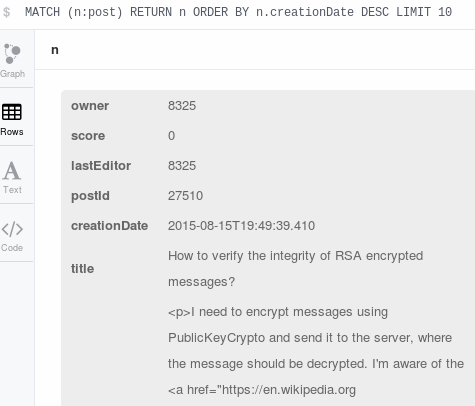
## Perform the Lab

Launch the neo4j sample app

1. In the nosql-lab-neo4j folder execute: nodejs app.js
2. You can now surf to the result of the app at [http://localhost:3000](http://localhost:3000/)
3. Check out what you get, get a feeling for it – especially get comfortable with Cypher to answer the questions
4. **STOP:** What exactly is shown at the main-url? How is this query done?

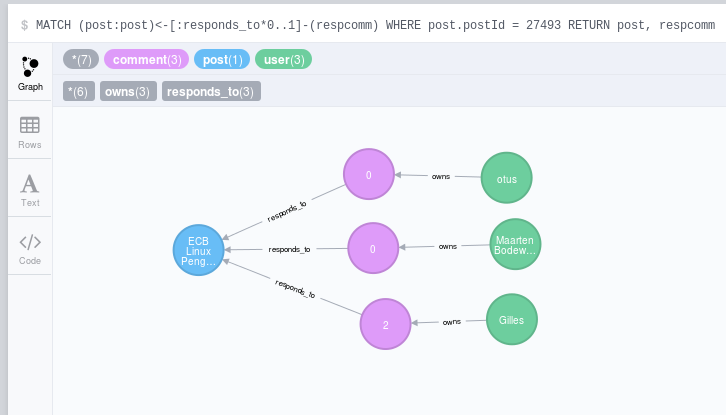
**MATCH (n:post) RETURN n ORDER BY n.creationDate DESC LIMIT 2**





1. **STOP:** What is shown on the detail-page, how does this work?

**MATCH (post:post)<-[:responds\_to\*0..1]-(respcomm) WHERE post.postId = 27493 RETURN post, respcomm**





1. **STOP:** On the Neo4J Console: Find users that commented their own posts and create a screenshot of this with the query and result well visible
2. **STOP:** On the Neo4J Console: Find the post with the highest degree of interaction with multiple other users in the replies/comments, again upload screenshot with result and Cypher query well visible

## Cypher

In RDBMS there is SQL as the *lingua franca* to query data. Within Graph-Databases it is not yet decided and Neo4J supports a wide selection of API’s: REST (with which the node.js application actually works), Native Java-API, the SQL-inspired Cypher query language and more from an historical perspective the Gremlin query language.

Cypher is by far the most approachable technique. Just like in SQL where you SELECT something FROM somewhere in Cypher you follow the Graph, matching on Nodes and Relationships, their types and properties.

Follow the “Introduction to Cypher” on Neo4J’s website to complete the lab questions; with some try and error you’ll quickly get the hang of it: <http://neo4j.com/docs/stable/cypher-getting-started.html>