Databases Neo4j Project Summary

**What:**

Our project intends to describe the graphical database Neo4j and compare it against relational databases and (to perhaps a lesser extent) document oriented databases. In our written portion of the project, we plan to begin with a short but broad overview as to what Neo4j is, including that it is a graphical database, runs on Java and Scala, and some facts related to its use in industry today. Followed by this, we will briefly review some of its basic features and queries that are covered in lab 11, and expand on them to include more in-depth features that were not mentioned. Next, we will cover what similarities neo4j shares with relational databases, as well as their most notable differences. Within this section we will compare timings in regard to complicated join queries as well as less interconnected searches and present our findings.

We will then cover the main similarities and differences Neo4j shares with Document Oriented databases. Following the similarities and differences between the different technologies, we will discuss the implications of the differences in the form of advantages and disadvantages of using Neo4j. Depending on how tricky it is to separate this discussion into a different section, we may eventually change our plan to incorporate the advantages and disadvantages into our similarities and differences section.

Our presentation will most likely be focused on a large scale implementation of neo4j, but we will also reference it in our written section and probably include some query commands that we found were most useful at retrieving data or gaining insight about data through relationships. More information may be included in this section once we have chosen and begun working with our implementation. The implementation itself will likely be of a practical or educational focus rather than for entertainment.

Finally, we will discuss some of the libraries and tools that can be used in conjunction with Neo4j, which can be related to additional query functionality or other ways of visualizing the data. Depending on the accessibility of the topic and amount of time we spend on the other topics, we may also address what it is like to access Neo4j through a rest API.

**Who:**

Kyle will research and discuss our information related to our general overview of Neo4j, which we think will serve as a good introduction for our written project (Will introduce concept of graph databases, the inner processes of neo4j, possibly mention the coding environment, and briefly talk about where neo4j fits into industry today).

Colin will be responsible for the Neo4j features aspect of the project which will include a review of the information shown in lab 11, as well as require an exploration and description of numerous additional features that go beyond lab 11.

Kyle will find the most notable similarities and differences that Neo4j has with RDBMS. To illustrate these differences, Kyle will be responsible for timing both technologies in two tasks, one related to joining interrelated data in a query, and another for querying less interrelated data and will present his findings. Kyle will also be in charge of discussing the Big O efficiencies related to these tasks and how they are performed internally.

Colin will be responsible for noting the similarities between neo4j and Document Oriented DBs, such as the usage of KVPairs. Kyle in turn will be responsible for finding the differences between them.

Colin will put together (based on the similarities and differences) the advantages of using Neo4j and its disadvantages when compared to other technologies. These points should be backed up with adequate reasoning and perhaps references, and should emphasize which applications are best suited for Neo4j.

Both of us will contribute to the section that talks about our neo4j implementation and we will probably include information that we each found significant about using Neo4j with the dataset. (More information should be added here later).

Colin will be responsible for exploring some of the notable libraries and tools that neo4j supports and talking about how they are useful and aid the functionality and or efficiency of neo4j in certain contexts. Visualization libraries are expected to be included in this exploration.

If we include a discussion about the REST API, both of us will be responsible for describing how it works and what it is like to use it.

Finally, at this point in time we are agreeing to divide writing the paper between ourselves (probably with each person mostly responsible for their own sections) although it is possible that this may change in the future.

**Point Breakdown:**

Neo4j General Overview (Kyle): 5 points

Neo4j Features (Colin): 10 points

Similarities with RDBMS (Kyle): 10 points

Differences with RDBMS (Kyle): 10 points

Similarities with Document DBs (Colin): 5 points

Differences with Document DBs (Kyle): 5 points

Advantages of Neo4j (Colin): 5 points

Disadvantages of Neo4j (Colin): 5 points

Large scale Model (Colin & Kyle): 10 points

Libraries and Tools (Colin): 5 points

REST API (Colin & Kyle): 10 points

Writing the Paper: (Colin & Kyle): 20 points

If we end up not discussing the REST API section, we propose that its 10 points should be distributed to the advantages and disadvantages, and similarities and differences of RDBMS categories (2.5 points each).