Question 4

The PostgreSQL, Neo4j graph, and Mongo DB document databases are vastly different in their approaches and procedures.

To visualize my data loading experience, I would describe RDBMS SQL database data loading as a procedural, almost enumerated approach. Each table was placed in the database, one by one, with minimal consideration as to how they relate to each other. This differed from Neo4j, and MongoDB.

Neo4j data loading was done in a “outer-to-inner” approach in that I created the data tables, and subsequently created the relationship (in the form of a dataset relationship) between these datasets. In parallel, the relationship creation in Neo4j took an extended amount of time, even with the creation of indexes.

To load the MongoDB data, I started with the “relationship dataset”, and worked from an “inner-to-outer” approach. Working from the “inner-to-outer” approach was very different for me in terms of a data loading approach. Loading the data was the easiest of the three databases in my opinion because record nomenclature and data type were not required to be considered.

Querying the datasets were very different as well. In my opinion, I feel that querying a RDBMS database is more intuitive, likely due to my familiarity with this mindset. However, the easiest database type to query, by and far, the Neo4j Graph Database. The queries are much more intuitive to visualize, and require much less code.

The most difficult database type to query for me was the MongoDB document database. The syntax is something I have very little experience with, and thus, made visualizing my queries very difficult. Also, queries took a long time to complete, which (until I figured out using the “limit” command) made troubleshooting my query structure difficult to troubleshoot.

For all three database types, I think that query tuning would be beneficial. I would like to investigate this further going forward.