Trace Levinson

Peyton Norder

Project 2 Write Up

PostgreSQL:

The data used for this project was heavily connected creating the issue of trying to decide how we were going to deal with the relations. The best option we decided was to make a separate table named ailed\_by that allowed for storing of the patient and doctor relationship along with the treatment the patient would receive for each illness. This eliminated the need for foreign keys and allowed for easy lookup of information. The big drawback again to PostgreSQL was the need to know exactly how we were going to use the information before using it. Several times we had to restart the process of creating tables because we thought of an idea and then later decided to renege on that idea and start with a new one. Creating mock data for PostgreSQL is very easy and importing a CSV into the database is also easy. A python script was used to insert the data and we were able to follow the same basic format as project 1 when it came to data creation and insertion. A big issue for us was figuring out how to have a doctor who is also a patient in PostgreSQL. The issue with this was keeping the ids in line along with all the table constraints. The ailed\_by table ended up helping us with this because we did not have to deal with foreign key constraints.

Neo4j:

Neo4j was a really cool database to use for this project. Being able to see the database in a graph in real time was a big help to making the schema. The schema did not take a long time to create and the free flowing way of neo4j was a big help. Creating the data for neo4j was more challenging compared to postgresql, this is because we did not have a lot of experience with neo4j data creation and the format we chose to make our data was not in csv format. We chose to do create statements in a cql file or match statements in the cql files. This took a lot of time to figure out and was a major drawback to neo4j. The ability to just chose what would be a relation with what was really nice and helped because the patient can have separate relationships with the doctors and the illness, along these lines the illness can have several treatments. Using Cypher helped a lot and the web interface makes it easy to see if the data actually got put into the database and if it did if the data was correctly inserted. Switching between the command line and the bolt interface did get a little annoying and while it was not necessary being able to see the data in bolt was a major part to us due to the idea that neo4j is very visual and we came into this project with that expectation.

Overall, we thought PostgreSQL was the better database for this project. This could be for several reasons, first relational databases may be better suited to deal with a problem like this, the strict schema made it so we had to keep our data in order and stay on top of how we were using it. We also have used sql databases significantly more than nosql ones. Specifically we have now used PostgreSQL for two big projects making us very comfortable in it.