**Porting With Another Team**

(Used Code from Team Including Jacob Zerr and Nathan DeJong)

The main lesson to be learned from this exercise on porting our application with another team’s database is one of specification description. What I mean is that this problem was extremely well documented on the website. Everything was laid out for the teams to follow, and there were some freedoms in design of the higher level interactions, but all in all the databases all do the same thing. The one area where everyone had some freedom was in describing how the application gets results from the database. In the case that the user wants something more descriptive than simply seeing all of the relation, the application needs to have some value which stores strings from the database. This significantly impacts how the application works, and though for our application there were few changes which we had to make, it is very easy to see how one open ended design specification could lead to hours of work in standardizing a final product created by multiple teams. Of course, this could have been resolved by planning beforehand with another team and agreeing on that part of the specification, but I believe this exercise taught a valuable lesson is code planning.

Another way this manifested itself was in the way that the applications call the parser commands. We create an execute function to catch errors, whereas the other team uses a simple query or command call which requires slightly more input on the application designer’s part and has the potential to provide a little less feedback. This also requires that in porting, we go through and every time we call an execute function we need to replace that with their verbage. Again, this is not an incredibly complex problem but due to the lack of specification in this area it is something that in order to port the application we must go back and fix, and in the end it takes time to make these small changes.