CSCE 315 Project 1

## Team Organization

Team five will be composed of Bradley Delaune, Chelsea Hall, and Greg LaFlash. Each of us will be given a different part of the Database to work on at first. Chelsea will be working on adding and deleting tables; inserting, deleting, and modifying data; and will continue on with the application once she is done. Greg will begin to implement the SELECT and FIND functionality of the database and will then add the capability to return the SUM, COUNT, MIN, and MAX of table columns.

### Responsibilities

By dividing the responsibilities this way, we allow for the most efficiency on this project. We will not “step on each other’s toes” by working on the same parts and potentially cause problems with the code. This organization will also help with our schedules and allow for a good staggering of work. This is practical in a sense that we can focus on other school work in between times of lots of coding. One disadvantage to this method will be how we rely on each other for information. Much of what Bradley and Greg do will be reliant on what Chelsea does at first so will not be able to get a whole lot done soon.

### Leader

We have chosen not to have a team leader or coordinator and we hope this will help us keep the work even. With a leader, that person could be more focused on delegating than getting work done which could cause difficult dynamics and more work for those who aren’t leading.

## Code Summary

### Overall Layout

After some discussion of different implementations, we settled on a particular one that we thought would be a good balance between functionality, ease of use, and ease of implementation. Allowing the user to define a database themselves and make commands, changes, and reads from it was our best idea. Users will simply be able to declare a database and then access this database with simple commands defined in the API. The commands will appear as member functions to allow the user to access many different functionalities of the database.

#### The Database Class

The Database Class will include a name, a map of tables, and many member functions to help the user access and manipulate data.

#### The Table Class

The Table Class will hold a name for the table as well as a map of columns in the table.

#### The Column Class

The Column Class will hold the name of the column, the type of data in the column, and a tuple of data and identification number. This will be the very basis of information for the database. The identification number will be defined so as to keep all information across columns linked together with the ID.

#### The Result Class

The Result Class will contain what our functionality returns including an error, if necessary, the name of the result (so the user can access it later), and the table of information. Although it seems overdone, we believe this will be the best way for a user to access the information outputted by the program.

### Implementation Issues

Although this is the best implementation we could develop, there will no doubt be difficulties and issues that we run into. The first that comes to mind is parsing. Parsing the commands will cause many issues that can be overcome with plenty patience and time.