**Kennedy Anukam – CS 457**

**Project One**

**Overview:**

The purpose of this project was the introductory steps to creating our own relational database management systems with the Sqlite3 statements. The statements that this project can handle so far is database creation/deletion and table creation/deletion/update/query. I organized my source code into two different modules as one took in the statements and the other was responsible for parsing the statements and determining what functionalities to do and error handling for incorrect input statements.

**Design:**

I designed this project by basing it off the key sqlite3 special words and implementing functions to handle the statements. Error handling was done in most functions to see if the statements were actually possible in a database management system. I organized multiple databases by creating a new directory for the database. If the database was already created, the code gives an error message saying it already exists. This was done through the os module mkdir() method which throws an exception if the folder already exists. The helper module in my code has a global variable which keeps track of the original starting directory the databases are being created from. For example, in my use database it goes to a new directory, but when the user wants to create another database it backtracks to the starting directory with the global variable and creates it there. Removing databases was done in a similar fashion as in the try except blocks, an error is thrown if a folder that does not exist is deleted. Creating tables was done by simply creating a text file within the database directory and inputting the datatypes and variable names into the text file. There is error checks within this method as when a duplicate table is created, the user is prompted the table already exists. If the user is not in a database, they are also prompted that they need to use a database first. This was done by checking if the global starting directory is == to the current directory. Dropping the table also first checked the directory and checks if the file exists. Helper methods were also used within the helper module to replace repeated code and keep the code cleansy. The main function uses the helper module to process all of the inputted lines and passing them into the statements function in the module. In this method, the string literals are first pre-validated by checking if the final character is a ‘;’. If it is not, the user is told why the statement is not valid. It then has if-elif-else statements to see if the sqlite3 commands are in the string literal. These are checked in a lowercase fashion as in sqlite3 it is case insensitive.