Author = Joshua Insorio

Date = 2/25/21

Project = PA1

Class = CS457

**IMPLEMENTATION:**

Database creation = system call to create folder if the folder does not already exist.

Database deletion = system call to delete folder if it does exist.

Table creation = fstream to create txt file inside of the database denoted by the "USE" command.

Use command = change directory call(chdir) to change current directory to the desired database.

Table deletion = system call to delete txt file inside of the database in use if the file exists.

Table update = fstream append similar to how table creation function is used.

Table query = print out the table if and only if the table exists inside the database in use.

**COMPILE:**

Use the provided make file and type “make”.

To run do ./main < PA1\_test.sql

**DOCUMENTATION:**

As recommended, each database is implemented as a folder to hold the database tables. The program is able to accomplish this by creating a folder via system call. Upon receiving the command “CREATE DATABASE”, using the parser and its helper functions the program is able to extract the name from the command line. The directory is then created via system call “system(“mkdir”)” to create the database. This method of organization allows for a separation between the different databases yet maintain a sense of organization and efficiency. Easily able to call upon the “USE” command to change directories and therefore between the different databases. This is accomplished through the system call “chdir(name.c\_str())”.

The program is able to manage multiple tables through the use of a dedicated txt file for each table. The table would be created, written into, and read from through the use of the fstream library. Specifically, the txt file is created through the command “file.open((name + ".txt").c\_str(), ios::out)” in which a txt file is created if it does not already exist. Then through the use of the “getData()” parser helper function, the data from the command line is then written into that txt file and will be kept separate from the other tables. The tables are only interacted with through the use of commands, but remain entirely in their respected txt files, so there is no need for any data structures. Most of the interaction with the txt files are through the fstream library, aside from deletion which is through another system call “system(("rm " + name + ".txt").c\_str())”. There is also the use of a boolean helper function that determines if the table already exists before file creation or deletion.

In short a majority of the functions needed to operate the database is via stdlib, fstream, string libraries. In theory most of the functions are self-explanatory, however they were unclear blocks of code were labeled in the cpp file. As a given, there are a lot of error checking statements checking for correct directory, existing files, and valid commands. At the current state, the program only takes the data for the tables as strings, however is subject to change should there be a need.