Computer Science 457 Project 2

By Justin Carlson

**How your program organizes tuples in a table?**

The structure of my tables are fairly simple. The first line of the table is reserved for the ids and data types that the table holds. Then every line after used for tuple storage. With each value aligning with its data type in the header line. An example can be seen here:

A picture containing text, orange, screenshot

Description automatically generated

When the table is loaded into virtual memory, it is stored in a 2-d array similar to how you would expect a table to actually look.

**How to run Database.py?**

Simply change your current working directory to the one containing both the Database.py and PA2\_test.sql files. Then Simply run python3 Database.py. An example can be seen below:



**Functionality**

From the test script that I was provided we had to implement 4 different SQL commands to get the specified functionality functioning. Those being Insert, Delete, Update, and a more advanced Select.

The Insert command is probably the simplest of all of the ones added in this assignment, it just simply parses the input command for the respective values and appends them to the current table file in the format above.

The Delete command was a little more tricky, for this one I had to create a way to virtualize the table in memory so it could be more easily accessible. To do that, I had made a table parser to load a table from a file into a 2-d array in memory. From there, you just parse the command and get the relevant values and modify the correct rows. All of which is implemented dynamically so it should work with completely different table ids and values. You also had to add operation wise comparisons which I only implemented the ones contained within the assignment.

The Update Command was basically the opposite of the delete command, parse the input for the relevant values, and modify the correct columns and rows after comparing the relevant input. With the virtual table, it is incredibly easy to modify and then write back to the disk after modification is complete.

The Select command from the previous assignment had to be modified to work with more complex queries. My Select commands check for the presence of an asterisk in the input, if one does not exist the Database will do a dynamic parse of the command. It first collects the select ids, then the table name, then the id type that it will be comparing all of the items with. This is all done dynamically and should work with any combination of data types.