CS 457 Database Management Systems

Programming Assignment 2

Amir Behmaram and Tanner Jones

3/27/2018

The program stores all of the table's data in JSON format, an example of an empty table can be seen below:

```
{
   "name": {
      "Data": [],
      "Datatype":
   },
   "pid": {
      "Data": [],
      "Datatype":
   },
   "price": {
      "Data": [],
      "Datatype":
   }
}
```

The tuples are stored within the Data as a list, and the columns type is stored within Datatype. Each JSON item represents a column with its name, type, and data.

The insertion function works as follows. First, it cleans the entire query string from any extra tabs, spaces, or newlines. Once that is achieved, the query is checked for correct syntax. After that, we extract the values, remove any spaces or characters that don't belong, convert them to their correct type, and insert them into the table. The implementation for insertion, deletion, and modification functions where designed very similarly. Each function starts off with parsing the command and pulling out key information needed to interact with the current working database which includes table name, column names, operators, and values. Each command then calls its own private

function which uses the information parsed to manipulate and work with each database.

Compile instructions for this project can be found below as well as in the README in the project directory. The project supports commands exactly as they are typed in the supported command list.

This program can be run in one of 2 ways:
1) The program can be passed a SQL file via standard input and it will perform all the operations
in the file.

ex: python main.py < PA2_test.sql will run the PA2_test SQL file.

2) The program can also be run similarly to SQL lite.

ex: python main.py will print a ":>" to the screen. From here you can enter any of the supported
SQL commands with the correct syntax and it will execute them.

Supported Commands:

- CREATE DATABASE DB;
- USE DB;
- CREATE TABLE table (name, type,..., nameX, typeX);
- insert into table values(val,...,valX);
- select * from table;
- update table
  set col = 'val'

```
  where col2= 'val2';
- delete from table
  where col = 'val';
- select col1, col2
  from table
  where col3 = 'val';
```