John Widmeyer

Cody Lacy

CS457

February 21, 2018

Programming Assignment 1: Metadata Management

Our “Metadata Management” assignment structures our databases in a ‘class structure’. Each ‘CREATE DATABASE’ input calls the constructor for the ‘database’ object—the database class houses a few critical elements. Firstly, the useFlag Boolean value, this allows checks to verify that only the currently ‘used’ database can have its tables accessed; additionally, this allows separate databases to have tables with the same identifier. Secondly, the string name value—whatever the input value is for the database ‘name’ on create is stored here, this value is referenced when calling the ‘use’ command. Thirdly, an array of all ‘table’ objects stored in this database—this is how we ensure only tables stored in the database can be accessed by that database. Lastly, the numTbles integer value which keeps track of the number of tables stored in each database, an index of sorts.

The handling of the table structure is similar to our databases; our tables exist as class objects, constructed once the ‘CREATE TABLE’ command is called. Inside the Table class we have a string name, and a string element; one is the name passed as a part of the create command, while the other is the data stored within the argument portion of the function call. These table objects are stored within their corresponding databases.

For the implementation of the provided script commands here is a brief breakdown:

CREATE DATABASE – Calls the database constructor storing the provided name in the ‘name’ field and increments the total number of databases in the system.

DROP DATABASE – Removes the named database from the array of all databases, shifting the array down and decrementing the total number of databases in the system.

USE – Sets the named database as the ‘active’, this means all subsequent table commands will query the tables housed within that database and only that database.

CREATE TABLE – Calls the table constructor storing the provided name in the name field, incrementing the total number of tables in the database, and adding the table into the array of tables housed within that database. Additionally, stores the argument string within the object for access later.

DROP TABLE – Removes the named table from the array of tables within the database, decrements the number of tables in the database.

SELECT – We chose to only implement the \* portion of the select command (since it was all that was specified in the test script)—since the only option is to output all of the table elements at once we simply stored the element in a single string, outputting the whole thing together. This method will simply grab the element data from the table object and output it to command line.

ALTER – Since the table object only has one string presently for data management, the ‘alter’ command will simply append the new data to the currently stored data in the table object.