Project 2: Database

Feb/22th/2017

-GOAL

Implements SQL like language using JAVA.

WOULD YOU KILL TO SAVE A LIFE? [PERSON OF INTEREST]
MUSIC: HURRICANE - 30 SECONDS TO MARS
EDITED BY: LISA (aka "TheHiddenMemory")
JAN 2015
TheHiddenMemory

-DESIGNING PROCESS

Questions (asked by Angie)

1. What data type to use to store a table (or I mean elements in the table)?
2. What variants should a table instance have (for example: number of rows, number of columns, name of columns, list that stores elements in rows, list that stores the elements in columns.)

3. How to read commands.
4. For load command, is there an instantiated method for us to use to open a file in our computer? What about to read a file?
5. Is database a place that stores all the table we create and implements all the method we need to use to modify a table? Answer: No. database is a place more like a launcher, class table is the place where I should implements all the methods which I need to multiply tables.
6. Since table can also be considered as a list containing lists (like a two-dimensional array), should we build a row class first and then a table class which contains a list of rows?
7. instances of databases are tables, how to create infinite

Other thoughts:

*columns can be maps whose key is a string of column name and whose elements are the content of that column.

-Design

Classes:

class Table:

Instance variables:

```
 /** Stores all of the table data. Everything is stored as strings,
   * even integers and doubles. */
 nestedMap<string, map> contents;

 /** Stores the names of the columns. */
 String[ ] columnNames;

 /** Stores the types of the columns. */
 String[ ] columnTypes;
```

Constructor:

```
  /** Creates a table of data. */
  Table(int numRows, int numCols, String[ ] colNames, String[ ] colTypes, nestedMap<string, map> tableData)

  /** Creates a table of data with the given number of rows and columns, but no data. */
  Table(int numRows, int numCols, String[] colNames, String[] colTypes)

  /** Creates a table of data with zero rows and zero columns. */
  Table()
```

Methods:

```
  /** Returns the join of Table a and Table b. */
  public static Table join(Table a, Table b)

  /** Returns the join of Table a, Table b, and Table c */
  public static Table join(Table a, Table b, Table c)

  /** Returns the join of all the Tables */
  public static Table join(Table[] tables)

  /** Gets all column names that are common to both tables. */
  public String[] getCommonColumnNames(Table a, Table b)

  /** Gets the row indices corresponding to the given condition. */
  public int[] getRows(Condition a)

  /** Returns the combination of the two columns using the column number
      and combinationSymbol in the operation. */
  public static String[] combineColumns(Table a, String columnA, Table b, String columnB, Operation<T> o)

  /** Adds the data in the row to the current table. */
  public void addRow(String[] row)

  /** Adds a new column to the data. */
  public void addColumn(String columnName, String columnType, String[] columnData)

class Columns implements map

class map{

Varients: linkedList Keys

          LinkedList Elements


class dRows implements map {
```