

William Flores
Kristen Pekala
Stephen Ruske
Group 1

API Specification

Database

constructor

Description - Creates empty database. Default constructor only one provided.

Parameters - None.

Return Type - N/A (constructor).

Errors Thrown - None.

destructor

Description - Deletes all tables pointed to by the database

Parameters - None

Return Type - N/A (destructor).

Errors Thrown - None.

copy constructor

Description - Makes deep copy of database

Parameters - constant database reference

Return Type - N/A (constructor)

Errors Thrown - None.

copy assignment operator

Description - Makes deep copy of database and assign it to existing database

Parameters - constant database reference

Return type - database reference

Errors thrown - None.

addTable

Description - Adds deep copy of given table object to database under provided name.

Parameters - constant table reference, string (name).

Return Type - void

Errors Thrown - Table already exists.

dropTable

Description - Removes table under given name from database.

Parameters - string (name).

Return Type - void

Errors Thrown - Table name not found.

listTables - constant

Description - Returns list of all table names in the database.

Parameters - None.

Return Type - vector of strings.

Errors Thrown - None.

getTables - constant

Description - Returns all tables in database.

Parameters - None.

Return Type - vector of table objects

Errors Thrown - None.

Query - constant

Description - Returns table containing queried information.

Parameters - string (SELECT), string (FROM), string (WHERE).

Return Type - table object.

Errors Thrown - Invalid query.

Table

(empty) constructor - Note: using this constructor is NOT recommended

Description - Creates an empty table with no rows and no columns

Parameters - None.

Return Type - N/A(constructor)

Errors Thrown - None

constructor

Description - Creates an empty table with the given column names

Parameters - vector of strings (column/attribute titles)

Return Type - N/A(constructor)

Errors Thrown - duplicate attribute titles, reserved word for attribute title

addAttribute

Description - adds a column to a table, all entries in this column will begin as NULL

Parameters - string (attribute title)

Return Type - void

Errors Thrown - attribute title already exists in table, attribute title is reserved word

deleteAttribute

Description - deletes a column from a table, also remove value for all records in table

Parameters - string (attribute title)

Return Type - void

Errors Thrown - Attribute does not exist

insert

Description - inserts a row (record) to a table

Parameters - Record

Return Type - void

Errors Thrown - invalid record for given table, duplicate record(if key is set)

getAttributes

Description - returns attributes of table

Parameters - None.

Return Type - vector of strings

Errors Thrown - None.

getSize

Description - returns the number of records in the table

Parameters - N/A

Return type - int

Errors Thrown - None.

makeKey- Note: only one key may be marked for a table, all records with a duplicate value for the attribute marked as the key will be deleted except for the first occurrence

Description - allows a single attribute to be a key for the table

Parameters - string (key)

Return type - void

Errors Thrown - Record does not have this attribute, no key provided, key already set

getKey - constant

Description - retrieves key if it is set

Parameters - None.

Return type - string

Errors thrown - key not set

keyIsSet - constant

Description - tells whether or not a key is set for the table

Parameters - None.

Return type - boolean

Errors thrown - None.

crossJoin

Description - takes two tables as input and produces one table as output

Parameters - constant table reference, constant table reference

Return type - table

Errors Thrown - None.

naturalJoin

Description - combines two tables if the first table has matching attribute(s) name matching the second table's key. Output has row for every entry from the first table combined with the attributes of the second table.

Parameters - constant table reference, constant table reference

Return type - table

Errors Thrown - no key set in joining table, attribute does not exist in initial table for key

inJoiningTable

count

Description - counts all non-null entries of attribute

Parameters - string (attribute name)

Return type - int (number of entries)

Errors Thrown - N/A

min

Description - takes an attribute and returns the entry with minimum of that attribute

Parameters - String (attribute)

Return type - Record

Errors Thrown - Attribute Does Not Exist

max
 Description - takes an attribute and returns the entry with maximum of that table
 Parameters - String(attribute)
 Return Type - Record
 Errors Thrown - Attribute Does Not Exist

begin
 Description - returns iterator to first entry in table
 Parameters - None
 Return type - std::list<record>::iterator
 Errors thrown - None
 Note: constant version provided as well

end
 Description - return end iterator to table
 Parameters - None
 Return Type - std::list<record>::iterator
 Errors thrown - None
 Note: constant version provided as well

Record

Default Constructor

Description - Creates empty record (use of this constructor not recommended)
 Parameters - None
 Return Type - N/A.
 Errors Thrown - None.

Constructor

Description - Creates record with empty values
 Parameters - vector of strings (attribute titles)
 Return Type - N/A.
 Errors Thrown - attribute title defined multiple times, attribute title is reserved word

Constructor

Description - Create record with given attributes and corresponding values
 Parameters - vector of strings (attribute titles), vector of strings (attribute values)
 Return type - N/A
 Errors thrown - parameters do not align, attribute defined multiple times, value shares column name, attribute is reserved word, value is reserved word

getSize - constant

Description - Returns the size of the records(# of entries it holds)
 Parameters - None.
 Return Type - Int
 Errors Thrown - None.

getAttributes - constant

Description - return all attribute titles for record

Parameters - None.

Return Type - vector of strings

Errors Thrown - None.

setData

Description - Adds an entry into the record.

Parameters - attribute (String), data (String).

Return Type - N/A.

Errors Thrown - attribute title not defined, value already name of attribute, value name is

reserved word

addData

Description - adds new field to record

Parameters - attribute (string), value (string)

Return Type - N/A

Errors thrown - attribute already defined, value is name of attribute title, value is reserved

word, attribute is reserved word

getData - constant

Description - Returns entry at given position in record.

Parameters - attribute (String).

Return Type - String.

Errors Thrown - Asking for position that doesn't exist.

removeField

Description - removes given attribute value pair

Parameters - attribute (String)

Return type - void

Errors thrown - attribute not defined

NOTES:

1. Database converts all text to uppercase and is NOT case sensitive
 2. The words "AND", "OR" and "NOT" are reserved and should NOT be used as names for attributes or values
 3. Illegal queries may return an empty table
-