API Specification

Our API consists of a namespace called Database and has primarily been divided into three different classes: Database, Table, and Record. In addition, we implemented a member class function inside of Databased known as Query and a struct for iterators. Query should be further developed to implement the properties of SELECT, FROM, and WHERE in order to return the appropriate table from these requests. The struct for iterators is defined in order to be further implemented to provide the functionality of iterators based on the purpose and use needed. The following documentation provides a description of each of the main classes and their functions that have been included for further implementation.

Database Class: Allows the user to create a single database object which will include the commands listed below.

- Function Name: Database
    - Purpose: Constructing an empty database.
    - Request Parameters: No arguments
    - Response Format: Returns the database object that has been created.
- Function Name: addTable
    - Purpose: Adds a table to the database by passing in a single table object and the name of this table
    - Request Parameters: Provide a single table object of type Table and a name for this table of type String.
    - Response Format: Return a Boolean in order to know if the table was indeed added, with true meaning that it was added and false meaning that it was unable to be added to the database.
- Function Name: dropTable
    - Purpose: Takes the name of a table, locates it within the database, and deletes the table from the database
    - Request Parameters: Provide a name for the table being deleted of type String.
    - Response Format: Return a Boolean in order to know if the table was deleted, with true meaning that it was deleted and false meaning that it was not deleted from the database.
- Function Name: listTables
    - Purpose: Provides a list of all the table names that are contained in the database
    - Request Parameters: No arguments
    - Response Format: Returns a Vector of type String in order to provide the list of table names to the user.
- Function Name: getTables
    - Purpose: Provides all the tables to the database to the user in map format in order to know the name of the table and have the object of the table.
    - Request Parameters: No arguments
    - Response Format: Returns a Map of type String and Table in order to have the name of the table along with the actual table object
- Function Name: query
    - Purpose: Function that will call upon the member class Query, in which the properties of SELECT, FROM, and WHERE should be defined.

    - Request Parameters: Three arguments that are all of type String format, one for SELECT, FROM, and WHERE.
    - Response Format: Return a Table with the information requested from the three arguments.

Table Class: Allows the database to create the tables with the commands and functionality listed below.

- Function Name: Table
    - Purpose: Constructing an empty table with no columns or rows
    - Request Parameters: No arguments
    - Response Format: Return the table object that has been created.
- Function Name: Table
    - Purpose: Constructing a table with one or more attribute names for the table
    - Request Parameters: Provide an attribute name or names of type String in a Vector structure
    - Response Format: Return the table object that has been created with the attributes provided
- Function Name: addAttribute
    - Purpose: Adding additional attribute names to the Table created by adding a column to the end of the table. All entries within that column should be implemented to be NULL.
    - Request Parameters: Provide an attribute name of type String
    - Response Format: Return a Boolean in order to know if the attribute was added to the table, with true meaning it was successfully added and false meaning otherwise.
- Function Name: deleteAttribute
    - Purpose: Delete a specific attribute from the Table including all the entire and information it contained within that attribute.
    - Request Parameters: Provide an attribute name of type String
    - Response Format: Return a Boolean in order to know if the attribute was deleted from the table with true meaning it was successfully deleted and false meaning otherwise.
- Function Name: insert
    - Purpose: Insert a specific record to the table
    - Request Parameters: Provide a record of type Record, described in class Record
    - Response Format: Return a Boolean in order to know if the record was indeed added to the table, with true meaning it was successfully added and false meaning it did not get added
- Function Name: getAttributes
    - Purpose: Returns all the attributes that pertain to the table in a Vector container
    - Request Parameters: No arguments
    - Response Format: Return a list in form of a Vector with type String that contain all the attribute names of the table
- Function Name: getSize
    - Purpose: Provides the size of the table based on the number of records.
    - Request Parameters: No arguments

    - Response Format: Return an unsigned int of the number of records considering it has to be greater than zero.
- Function Name: begin
    - Purpose: Provides an iterator type to be pointing to the beginning of the structure, container, or list upon which it is called on.
    - Request Parameters: No arguments
    - Response Format: Return a type tblIterator in order to represent the pointer at the beginning of a list or container.
- Function Name: end
    - Purpose: Provides an iterator type to be pointing to the end of the structure, container, or list upon which it is called on.
    - Request Parameters: No arguments
    - Response Format: Return a type tblIterator in order to represent the pointer at the end of a list or container.
- Function Name: defineKey
    - Purpose: Allows to assign an attribute name as the key of the table
    - Request Parameters: Provide an attribute name of type String in order to define the key
    - Response Format: Return a boolean in order to know if the key was indeed implemented with true meaning that a key was assigned to the table and false otherwise.
- Function Name: crossJoin
    - Purpose: Allows the user to combine two different tables into one single table
    - Request Parameters: Provide two arguments, both of type Table and each containing a different table object
    - Response Format: Return a single table of type Table that contains both original tables combined into one.
- Function Name: naturalJoin
    - Purpose: Allows the user to combine two different tables into one single table, where the first table should have an attribute name which is the key of the second table. If the first table does not have an attribute name with such a key or the second table has not been assigned a key, an exception error should be thrown.
    - Request Parameters: Provide two arguments, both of type Table and each containing a different table object
    - Response Format: Return a single table of type Table that contains the conditions listed.
- Function Name: routines
    - Purpose: Finds the count, minimum, and maximum of an attribute name that is included in the table
    - Request Parameters: Provide an attribute name of type String.
    - Response Format: Return the results of the count, minimum, and maximum for the specific attribute name.

Record Class: Allows the database to store individual records.

- Function Name: Record
    - Purpose: Allows the user to create a record of an arbitrary size and all entries should be initialized to a null string.
    - Request Parameters: Provide the size of the record of type Unsigned Int.
    - Response Format: Return the record object that has been created
- Function Name: getSize
    - Purpose: Return the size of the record
    - Request Parameters: No arguments
    - Response Format: Return the number of entries in the record of type Unsigned Int.
- Function Name: set
    - Purpose: Allow the user to write to a record with a specific characteristic about it at the specific index based on the size of the record.
    - Request Parameters: Provide two arguments, one indicating the index of type Unsigned Int and the other indicating the characteristic that needs to be added to the record of type String.
    - Response Format: Return a Boolean if such characteristic was successfully added to the specific record with true meaning it was added and false otherwise.
- Function Name: get
    - Purpose: Allow the user access to specific entry within the record.
    - Request Parameters: Provide the specific index of type Unsigned Int
    - Response Format: Return the characteristic of this specific index of type String.