**Database API**

**Supported Commands**

**Database:**

- Database()
- addTable(Table& t1)
- deleteTable(string& s)
- listTables()
- getTables()
- query(vector<string> attributes, string name, string where)

**Table:**

- Table()
- Table(int n_args, …)
- addAttribute(string s)
- deleteAttribute(string s)
- insertRecord(Record& r)
- getAttributes()
- getSize()
- recordsBegin()
- recordsEnd()
- setKey(const string& s)
- crossJoinTables(Table& t1, Table& t2)
- naturalJoinTables(Table& t1, Table& t2)
- countEntries(const string& s)
- min(const string& s)
- max(const string& s)

**Record:**

- Record(int arg)
- getSize()
- operator[](int input)

**Functionality**

**Database:**

- Database()
  - Creates an empty database with no tables
- Void addTable(Table& t1)
  - Adds a table to the database
- Void deleteTable(string& s)
  - Deletes a table from the database with the argument specifying the name of the table to delete
  - Throw an error if the table with the name passed in as s does not exist in the database
- Vector<string> listTables()
  - Returns a list of all table names in the specified database
- Vector<Table> getTables()
  - Returns a list of all Table objects in the specified database
- Table query(vector<string> attributes, string name, string where)
  - A query takes three arguments
  - The first argument specifies the attributes that need to be looked at
  - The second argument states the name of the table that needs to be looked at
  - The third argument is a condition that the entries within a table's records are compared to. If true the record is added to the table that is going to be returned.

**Table:**

- Table()
  - Creates an empty table with no records or attributes
- Table(int n_args, …)
  - Variable number of arguments
  - Take an arbitrary amount of strings as arguments that represent Table Attributes
  - Need to go through each argument and use it appropriately
- Void addAttribute(string s)
  - Adds a column to the table with the specified attribute name passed in as an argument
  - If the attribute has already been added then throw an error
- Void deleteAttribute(string s)
  - Deletes a column from a table
  - The passed in string is the attribute's name, specifying which column to delete
  - If the attribute is not part of the table throw an error
- Void insertRecord(Record& r)
  - Adds a new Record to a table
  - The passed in record needs to be
- Vector<string> getAttributes()
  - Returns a list of column names (attributes)
- Int getSize()
  - Returns the number of records within the table
- Map<int,string>::iterator recordsBegin()

    - Returns an iterator to the beginning of the collection of records
    - Allows the collection to be scanned through and analyzed and modified if need be
- Map<int,string>::iterator recordsEnd()
    - Returns an iterator to the end of the collection of records in a table
    - This allows for scanning through the collection of records to analyze them
- Void setKey(const string& s)
    - Designates an attribute/column as a key for the table
    - Can have multiple keys
- Static table crossJoinTables(Table& t1, Table& t2)
    - Takes two tables as input
    - Joins the tables by mapping each row from the first table to each row of the second table
    - The resulting size of the new table is the Cartesian Product of the two tables
- Static table naturalJoinTables(Table& t1, Table& t2)
    - Creates a new table based on common columns between two tables
    - With the common columns find matching rows with the same entry in the common columns and then create a new table with these two rows combined
    - If no common columns are found throw an error
- Int countEntries(const string& s)
    - Returns the number of non empty fields within the table specified by the attribute name passed in as s
    - Throw an error if the attribute name does not exist
- String min(const string& s)
    - Returns the minimum valued field specified by the attribute name passed in as s
    - Throw an error if the attribute name does not exist
- String max(const string& s)
    - Returns the maximum valued field specified by the attribute name passed in as s
    - Throw an error if the attribute name does not exist

**Record:**

- Record(int arg)
    - Creates an empty record of size arg which is passed in.
    - This allows a record of arbitrary size to be created
    - Initialize all entries to a null string
- Int getSize()
    - Returns the number of entries in a record
- String operator[](int input)
    - Returns the entry at the index given by input.
    - Throw an error if the value of input is greater than the size of the record