Wyatt Payne
Matt Stevens
Neeketh Sheth

Team Assignment 1 – API

DATABASE: member functions

- Database();
  - Constructor. Creates an empty database
  - This is the only constructor available. Not possible to construct database with Tables/Names as arguements
- ~Database();
  - Called by system. Destroys Database object
- void add_table(std::string name, Table table);
  - Add Table to database object
  - Every Table has a name associated with it
- void drop_table(std::string name);
  - Drops table name from database along with associated table
- std::vector<std::string> list_tables();
  - Iterates through the entire database and returns a vector of all names associated with each Table
- std::vector<Table> get_tables();
  - Iterates through entire database and returns a vector of all Tables in database
- Table query(std::string select, std::string from, std::string where);
  - The selector key is a list of attributes to keep in the output table. The attributes should match the order saved in the record. "*" will save all attributes.
  - The from selector identifies which table to run query on from within the database.
  - The where selector is a condition string comparions can be called.
  - Valid comparisons include =, <>, >, =, <=
  - The where clause should contain the attribute name, condition, and value in that order. i.e. "Hours >= 12"

TABLE: member functions

- Table(std::initializer_list<std::string> args);
  - Initializes a list of strings into a vector, to be stored as a record in a vector of records
  - The strings are the attribute names. The first Record in the table will always be a Record containing the order of attributes and their names.
- ~Table();
  - Destroys table object
- void add_attribute(std::string name);
  - Adds attribute to the attributes Record in the vector of recordds
- void delete_attribute(std::string name);
  - deletes attribute that matches with name from record of attribute names

- void insert_recortd(Record r);
    - inserts a record in to the table
- std::vector<std::string> get_attributes();
    - returns a vector of strings containing all the attributes of the table
- int get_size();
    - returns the size of the table, ie number of vector entries
- Record get_record(std::string key);
    - Returns the record associated with each record
- void set_key(std::string attribute);
    - sets a single attribute name to be the key for all records in the table
- Table cross_join(Table table_1, Table table_2);
    - Merges two tables and returns the product of that merge.
- Table natural_join(Table table_1, Table table_2);
    - Joins two tables that must have matching keys and attribute names and joins the two tables.
    - If the keys or names do not match an exception will be thrown
- int get_count(std::string name);
    - gets the count of number of entries based of attribute name
- int get_min(std::string name);
    - gets the minimum entry based off attribute name. min is determined through string comparison
- int get_max(std::string name);
    - gets the maximum entry based off attribute name. max is determined through string comparison

RECORD: member functions

Record();

- Record(int size);
    - Creates a record of certain size
- ~Record();
    - Destroys a record when the system is done with it
- int get_size();
    - returns the size of the record, i.e. the number of possible string inputs
- std::string& operator[](int index);
    - overloads [] operator for assignments and getting the value of certain indices within the record