CSE102 – Computer Programming Homework #11

Due Date: 12/06/2022

You can email iaytutuldu@gtu.edu.tr if you have any questions.

In this homework, you will be implementing a simple database application using dynamic data structures. Your database can have any number of tables. Each table can have any number of columns and rows. All the operations such as creating database, tables, etc. reading (querying) or writing (inserting) from/to a table will be in memory. However, we would like to able to run the database with previous one. Therefore, only one time you must save your information on the file. In other words, when your program ends, for testing, we will be able to create a new database, related tables and row data again. Also, we may run with the previous one. Be careful that this is not an assignment that requires a development of a database depend on the multiple files as tables.

You are free to design and implement these operations as mentioned above in order to test your structure. Since this is the final homework assignment for this semester, you are free to use your programming knowledge and skills that has been improving throughout the semester.

1. Read from a file
2. Write to a file

You may use the following structure, also any reasonable changes or insertions (if you commented in detail) will be accepted.

typedef struct database {
 	
    tables * tList; /* to be implemented as a linked list */
 	
int n; /* num of entries */
 	
char * name; /* table name */
 	
} database;
	typedef struct tables {
	
	tables *next;
	
	table * t;
	
	} tables;
		typedef struct table {
		
		    char **field;
		
		char **type;
		
		bool *isNull;
		
		bool *isKey;
		
		} table;

```
mysql> DESCRIBE pet;
+---------+-------------+------+-----+
| Field   | Type        | Null | Key |
+---------+-------------+------+-----+
| name    | varchar(20) | YES  |     |
| owner   | varchar(20) | YES  |     |
| species | varchar(20) | YES  |     |
| sex     | char(1)     | YES  |     |
| birth   | date        | YES  |     |
| death   | date        | YES  |     |
+---------+-------------+------+-----+
```

For each table;

- 1st row of the table is the name of the fields(columns) with their type. (e.g. name | char (20))
- Other rows can be the different types such as char, int, float, double, char [64].

The database application will have a similar functionality as MySQL database; therefore, you would like to see the web page: MySQL

Functionality	MySQL command examples	C program function name
Create a new database if not exists	mysql> CREATE DATABASE <name>;	**void create_database(char *name);**
Show tables on a database by names	mysql> SHOW TABLES;	**void show_table(database *d);**

Describe tables	mysql> DESCRIBE <name>;	void desc_table(database *d, table *t);
Create a new table with columns on a database	mysql> CREATE TABLE pet (name VARCHAR (20), owner VARCHAR (20), species VARCHAR (20));	**void insert_table(database *d, table *t);**
Remove a table from a database	mysql> DROP TABLE <name>;	**void remove_table(database *d, char *name);**
Insert a key on a table	mysql> ALTER TABLE contacts ADD CONSTRAINT contacts_pk PRIMARY KEY (contact_id);	**Void insert_key(database *d, table *t, key_value)**

Once your database, tables and all operations ready, write a test driver and test your program by creating databases, tables, keys, and all the functions.

Submit all of your implementations including driver program and your output screens, files, reports if any, and any documentation that you write or used.

NOTE: You are free to change the design or implementation details (you must obey the linkedlist structure) that are reasonable for you if you comment on them.