Project 1-2: Schema Store and Access

General

In this assignment, you will build schema store and accessing function of your simple DBMS, based on your implementation of Project 1-1. After finishing Project 1-2, your simple DBMS has to be able to process 4 DDL statements - CREATE TABLE, DROP TABLE, SHOWTABLES, and DESC. The table schemas should be non-volatile and stored in files. Consider that you will actually insert, delete records in your schema in Project 1-3. Using Berkeley DB is recommended for saving table schema in files,.

1. Requirements

- Implement a simple DBMS which
    - Is based on your implementation of SQL Parser (Project 1-1)
    - Processes DDL statements listed in section 2. If an error occurs, show a suitable error message. Otherwise, in case of success, show a success message.
    - Saves table schema in a non-volatile file (or files) - the table schemas remain even if you exit your DBMS console.
    - Uses Berkeley DB to store table schema (Recommended)

2. SQL

2.1 CREATE TABLE

```
SQL_2008-23593>
CREATE TABLE dbta_test
(
        id INTEGER PRIMARY KEY NOT NULL,
        name CHAR(10)
);
 ‘dbta_test’ table created!!
SQL_2008-23593>
```

- If an error occurs, show a suitable error message. Otherwise, in case of success, store table schema and show a success message – [table_name] table created!!.
- If the table which has the same name already exists, show an error message.
- Only one attribute can be the primary key of the table.
- Every column name in a table has to be different.

2.1 DROP TABLE

```
SQL_2008-23593>
DROP TABLE dbta_test;
 'dbta_test' table dropped!!
SQL_2008-23593>
```

- Drop table, if the table exists.
- If not, show an error message

2.1 SHOWTABLES

```
SQL_2008-23593>
SHOWTABLES;
------------
dbta
test
emp
lw
------------
SQL_2008-23593>
```

- Show a list of all existing tables which have been created
- If there is no created table, show a message - 'No existing tables'

2.1 DESC

```
SQL_2008-23593>
DESC dbta_test;
--------------------------------------------------------------
table_name [dbta_test]
column_name     data_type       null    primary key
id              INTEGER         N       Y
name            CHAR(10)        Y       N
--------------------------------------------------------------
SQL_2008-23593>
```

- Show the schema descriptions of all existing tables
- Each description consists of column name, data type, null option, and primary key option.
- If there is no created table, show a message - 'No existing tables'

3. Development Environment

- 3 Programming languages are allowed (C , C++, Java)
- LINUX or Windows
- API : Berkeley DB, ...
  - You can use any kind of API, but it should be specified in your report

4. Submit

Files to submit

- Source files (must have comments), Binary files, Makefile (or bat),
- A Report (this file contains the following)
  - a) Development environment
  - b) Explanation on the major modules and algorithms
  - c) What you have implemented and what you have not (Specify in detail)
  - d) Brief explanation of your implementation (less than half a page)
  - e) Any assumptions you have made
  - f) How to compile and run
  - g) Talk about your experience of implementing Project 1-2

- ● Please submit the files in .zip format with the filename corresponding to your student id.(e.g**PRJ1-2_StudentID.zip**) via email to the teaching assistant (liza183@europa.snu.ac.kr)
  - ■ Email Title : [introDB Project1-2] Your Student ID, Your Name
- ● Please submit the hard copy of your report to Building# 302 Room# 314-1
- ● Due Date: April 14st,2008 23:59(Mon)

5. Late Assignment Policy

Programming assignments are due at 11:59pm on the date specified. A grading penalty will be applied to late assignments. (10% penalty up to the first 24 hours, 20% for 24 to 48 hours, with no credit received after that)

6. Reference

- ● Bekeley DB
  - ■ http://www.sleepycat.com/docs/ref/toc.html