

Project 1-3: Data Insert, Update, Delete, Select

General

In this assignment, you will implement DML (Data Insert, Update, Delete, and Select) of your simple DBMS based on Project 1-1 & 1-2. After finishing Project 1-3, you should be able to use your DBMS for simple purposes. The instances should be non-volatile and stored in files. Using Berkeley DB is recommended for saving table schema in files.

1. Requirements

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

2. SQL

2.1 INSERT

```
SQL_2005-20000>
INSERT INTO IDS_STUDENT (ID, NAME)
VALUES ( '1' , 'sklee' );
1 rows inserted!!
SQL_2005-20000>
```

- ◆ If an error occurs, show a suitable error message. Otherwise, in case of success, insert values into table and show a success message– [number of rows] rows inserted!!
 - ◆ Insert into non-existing tables
 - ◆ Insert into non-existing fields
 - ◆ Type checking
 - ◆ Insert duplicated values into Primary Key
 - ◆ Not null/null checking
 - ◆ Other Syntactic, Semantic Errors

2.1 UPDATE

```
SQL_2005-20000>
UPDATE IDS
SET name = 'DB TA'
WHERE id = 1
"1 rows updated!!"
SQL_2005-20000>
```

- ◆ If an error occurs, show a suitable error message. Otherwise, in case of success, update values in table and show a success message– [number of rows] rows updated!!.

2.1 DELETE

```
SQL_2005-20000>
DELETE FROM IDS
WHERE ID = 1;
1 rows deleted!!
SQL_2005-20000>
DELETE FROM IDSK
WHERE ID = 1;
"error : no table - 'IDSK' !!"
SQL_2005-20000>
```

- ◆ If an error occurs, show a suitable error message. Otherwise, in case of success, delete rows in table and show a success message– [number of rows] rows deleted!!.

2.1 SELECT

```
SQL_2005-20000>
SELECT id, name
FROM IDS
id          name
-----
1           sglee
2           sklee
3           scpark
4           sskang
SQL_2005-20000>
SELECT IDS.id, CSE.name, CSE.STUDENT_NUMBER
FROM IDS, CSE
WHERE IDS.name = CSE.name
id          name          STUDENT_NUMBER
-----
1           sglee          2003-XXXXX
2           sklee          2004-XXXXX
3           scpark        2004-XXXXX
4           skkang        2004-XXXXX
```

- ◆ If an error occurs, show a suitable error message. Otherwise, in case of success,

show suitable instances from specified table(or tables) – DBMS should be able to join 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 1-3
- Please submit the files in .zip format with the filename corresponding to your student id.(e.g**PRJ1-3_StudentID.zip**) via email to the teaching assistant (liza183@europa.snu.ac.kr)
 - Email Title : [introDB Project1-3] Your Student ID, Your Name
- Please submit the hard copy of your report to Building# 302 Room# 314-1
- Due Date: May 23th,2008 23:59(Fri)

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>