CS 457/657 Database Management Systems

Programming Assignment 2: Basic Data Manipulation

Overview

In this assignment you will write a program that allows a database user to insert, delete, modify, and query their data on basic tables (i.e., no table joins). It is assumed that the database and table structures (i.e., their metadata) are available, which should have been completed in the first assignment.

System Design

- You will decide how to physically store the data
  - In lectures: row based, column based, etc.
  - As always, you are free (in fact, encouraged) to come up with your own design
- If in your first assignment you decided to use the mapping of directory-database and file-table, then the following is one possible physical layout of the file for table ***Product (pid, name, price)***:

  pid int | name varchar(20) | price float
  1 | laptop | 1999.99
  2 | mobile phone | 899.99
  3 | monitor | 1399.99
- Once you decide the physical layout, you will then need to think about how those four operations (insert, delete, modify, and query) will be implemented.
  - Again, if you follow the above file-table design:
    - Insert: append to the last line?
    - Modify and query: should be easy
    - Delete: move the tuples around?

Implementation

- The program should not use external database library/application…
  - E.g., don't write a wrapper over Sqlite3
- Any popular language is fine, e.g., Python, Java, C/C++, Go
  - Just pick one(s) that you are most comfortable/proficient with
    - Or, contact the TA about your programming language before you start
  - But keep in mind we will test your code in Linux with OS-level utilities (e.g., files)
    - So, probably not: C#, Object-C, JavaScript, Prolog…
- Functionalities:
  - SQL: insert tuples, delete tuples, modify tuples, basic select on tuples

Interface

- A similar but simpler interface than Sqlite3
- Same as homework #1: standard input or filename argument, standard output.

Testing

- We will test your program on Ubuntu (version 14 or above)
- If your program cannot compile on our testbed, we may ask you to demo your program
    - Try not to use many exotic libraries…
- A full test script will be provided, for example (assuming the standard input):
    - # ~/cs457/pa2/<your_program>
        - Then copy & paste the test script
    - # <expected output>
    - You don't need to parse the comment lines (i.e., starting with "- -")
    - We will not test your programs with other scripts/commands
        - It's always good to consider more edge cases

Grading (20 points)

- This is an individual assignment
- Design document that clarifies the followings: (5 points)
    - How your program store tuples in the table
    - At a very high level, how you implement those required functionalities, i.e., tuple insertion, deletion, modification, and query.
    - Be very specific on how to compile and execute your code
- Source code (15 points)
    - Coding style and clarity, 5 points
        - Appropriate parenthesis locations, indention, etc.
        - Always write comments at the beginning of any files
            - Author, date, history, etc.
        - Always write comments at the beginning of any non-trivial class/function
            - What this class/function does, high-level algorithm if needed
        - Write in-line comments for non-trivial blocks of code
    - Functionality, 10 points

Submission

- WebCampus
- Compress all your source code and documents into one package in this format:
    - <your_netid>_pa2
- Late penalty: 10% per day