

# CS457/657 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 ready, 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 work.
  - 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...
- Any programming language is acceptable, e.g., Python, Java, C/C++, Go
  - Just pick one(s) that you are most comfortable/proficient with
  - 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:

- o 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, 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
  - o Try not to use many exotic libraries...
- A full test script will be provided
  - o `# ~/cs457/pa2/<your_program> < PA1_test.sql` (expect the standard input)
  - o `# <expected output>`
  - o You don't need to parse the comment lines (i.e., starting with "`-`"")
  - o We will not to test your programs with any other scripts/commands
    - It's always good to consider more corners cases

## Grading (100 points)

- This assignment can be completed by a group of 1-3 students
  - o All group members will receive the same score
- Design document that clarifies the followings: (10 points)
  - o How your program store tuples in the table
  - o At a very high level, how you implement those required functionalities, i.e., tuple insertion, deletion, modification, and query.
  - o Be very specific on how to compile and execute your code
- Source code (90 points)
  - o Coding style, 10 points
    - Appropriate parenthesis locations, indention, etc.
  - o Coding clarity, 10 points
    - 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
  - o Functionality, 70 points
    - Refer to the test script for detailed breakdowns

## Submission

- WebCampus
- Compress all your source code and documents into one package in this format:
  - o <your\_netid>\_pa2
- Each group member should submit his/her own copy of the package
- Late penalty: 10% per day (if you don't use the 4-day grace period)