CS457/657 Database Management Systems

Programming Assignment 3: Table Joins

Overview

In this assignment you will write a program that allows a database user to make queries over multiple tables. That is, you will implement table joins using techniques we discussed in the class. This assignment assumes the metadata and data manipulations on singular tables are already available, which should have been provided (implemented) in the first two programming assignments.

System Design

- You will decide how to implement the join operations
  - In lectures: nested loop, index, hash, sort
  - As always, you are free (in fact, encouraged) to come up with your own design
- In the design document, you should clearly explain how your program joins tuples from two table sources. If in your first assignment you decided to use the mapping of directory-database and file-table, and if in your second assignment you followed the following physical layout of for an example 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

  then the pseudocode for a join could be (omitting the column headers):

  ```
  for each line of table_file1
     for each line of table_file2
        check the condition parsed from the given query
  ```

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:
  - SQL: inner join, left outer join

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
    - Try not to use many exotic libraries…
- A full test script will be provided
    - # ~/cs457/pa3/<your_program> < PA1_test.sql (expect the standard input)
    - # <expected output>
    - You don't need to parse the comment lines (i.e., starting with "- -")
    - 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
    - All group members will receive the same score
- Design document that clarifies the followings: (10 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 (90 points)
    - Coding style, 10 points
        - Appropriate parenthesis locations, indention, etc.
    - 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
    - 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:

    - <your_netid>_pa3
- 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)