Design Document – Project 3 CS457

Project Overview

Project 3 focuses on different variations of joins in SQL, in addition to the functionality performed in project 1 and project 2. The project implements tuple insertion into a created table. The functionality of creating a table was programmed in project 1. The project will create a database, create a table, insert tuples into two different tables, and then use variations of joins to show desired tuples from each table. The program will implement two syntax variations of the inner join functionality, in addition to a left outer join between the two tables created earlier in the project.

Implementation of Inner Join

The inner join functionality physically resides within the select functionality in my code. Depending on the syntax variation of inner join, the program will look for a given key word to perform the proper joining of the two tables indicated in the query. Each table is created in the beginning of the program with the create table functionality. When performing the inner join, each table that is needed in the query is given is read from its file, and used to compare the parameters required in the query. To start, the headers of each table are printed together on the first line of the output line. This allows the user to see what table each parameter belongs to when comparing and printing the output of the inner join. Next, the inner join functionality is performed, comparing to see where a given parameter matches from one table to the other. Finally, the program will parse each table (each file that was read in earlier in the program), and compare each row of each table to compare the desired query output. The program will then print the tuples from each table together under the headers of the tables, indicating that they have indeed been matched up or equivalent.

In addition, there are two syntax variations for the inner join functionality.

First, the program can:

select \* from Employee E, Sale S where E.id = S.employeeID;

This syntax variation does the same functionality as the syntax variation with inner join in it.

The other inner join syntax looks as follows:

select \* from Employee E inner join Sales S on E.id = S.employeeID;

The output of both syntax variations will be exactly the same when printing the output to the terminal.

Implementation of Left Outer Join

The left outer join functionality physically resides within the select functionality in my code. There is one syntax variation for the left outer join functionality. Each table is created in the beginning of the program with the create table functionality. When queried with a left outer join, the two tables are again compared after the contents of each table have been successfully read into two files for comparing. The query will prompt for a specific parameter to be compared between each table. First, the header from each table is printed together on the first line of the output. Next, the program will check the parameters the query is looking to test, parsing through each file to see where the parameters match up. Finally, the expected output is printed to the terminal. The major difference between the inner join and the left outer join is that while using the left outer join, the left side of the expected output from the table will always be printed out. If the left side of the join does not have a corresponding match in the other table, the output will still print the left side, with a blank space for the right side, indicating that there was no match. If a match has been found, the program will print out each tuple from each table, exactly how it did in the inner join output.

Compile and Execute the Code

1. Download my file: nicholasmason\_pa3
2. Compile my program in c++ using: g++ PA3\_nicholasMason.cpp
3. Then type: ./a.out
4. The program will automatically take in the PA3\_test.txt file in my folder and run the program.
   1. I was given permission to use PA3\_test.txt, rather than PA3\_test.sql because of printing problems.
5. If needed to run the program line by line, compile my program in c++ using: g++ PA3\_lineByLine.cpp
6. Then type: ./a.out
7. The program will prompt the user to input a SQL query line by line, thus displaying the expected output directly after the operation has been executed.
   1. This allows the user to see each operation performed individually.