Zachary Neeley
CS 482

Project Phase 2 Report

Single Insertion:
To accomplish the Single Insertion the program will prompt the user for the file that contains the needed data for the insertion statement. Inside this file, the data will follow the same format depending on the table being used. Here are examples that will be followed.

Players.txt:

10001, patrick jackson, indianapolis colts, WR, 19, 66, 33568512

Games.txt:

10001, 2005-9-28, everbank, W, 899, 9458

Play.txt:

41761, 2327

Once the data is read into the program singleInsertion() is called. This function will connect to the database and issue the SQL statement once it has been created. This is done by first calling sqlInsertFormat() this function creates the beginning of the insert statement. Using players as an example it will create:

INSERT INTO players (PlayerID, Name, TeamName, Position, TouchDowns, TotalYards, Salary) VALUES (data to be added)

Followed by the values passed by each line in the file. After this, the insertion statement is issued to the database. If an error occurs it will be caught by the program and added to the error Listbox to be shown once the process finishes. This will loop following the same steps until the end of the file is reached.

Multi Insertion:
To accomplish the Multi Insertion first prompt the user for the file to process and insert it into the database. The files will need to be formatted the same way as listed above. Once the file has been given multiRowInsertion() is called. This function will connect to the database and issue the insertion statement when needed. During this process, MultiRowDef() is called this function will create the beginning of the insertion statement. Which would like this, using players as an example:

INSERT INTO players (PlayerID, Name, TeamName, Position, TouchDowns, TotalYards, Salary) VALUES

After the first part of the insertion statement is created the program will call MultiRowText() which will add the data to the insertion statement. This process will create better runs times when compared to Single Insertion because instead of creating a new statement each time the loop repeats Multi insertion will only create one massive insertion statement to issue to the

database. This will cause the run time to decrease when compared to the single insertion method.

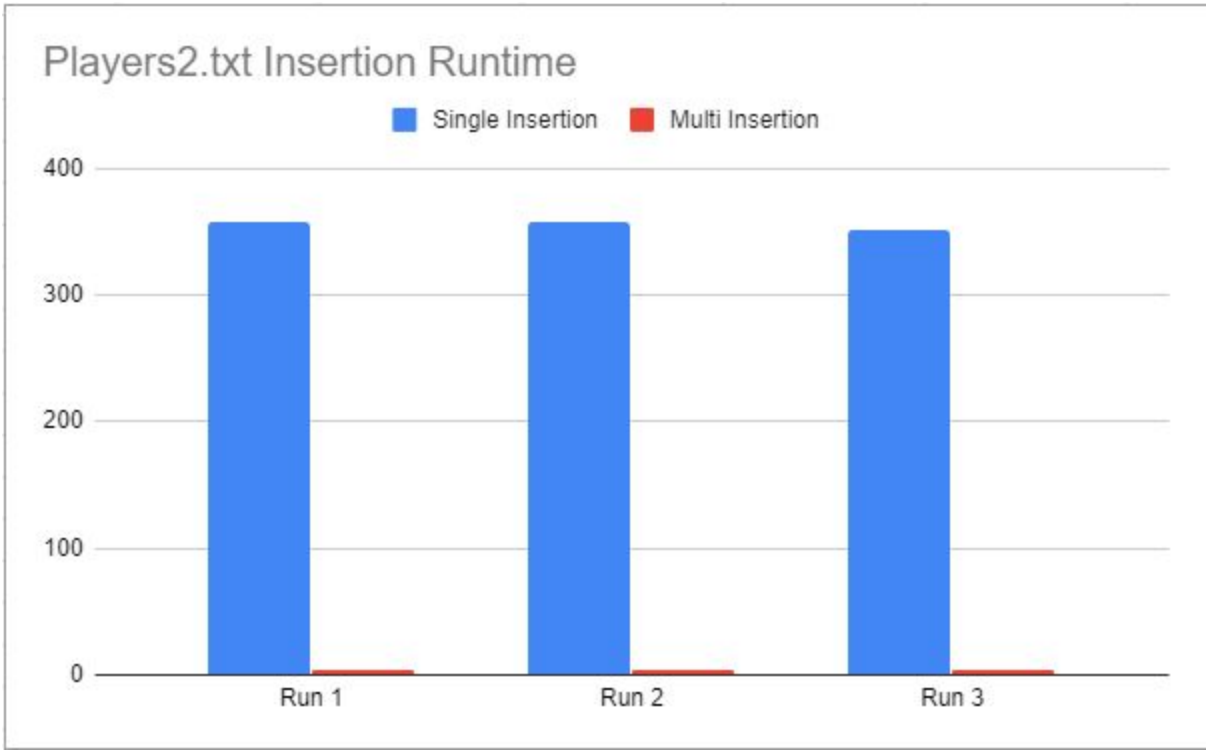
Delete From Table:
To delete the data in the table deleteTable(option) is called. This function will remove the data in the table while keeping the table intact. When called the option variable will store the needed SQL statement. When depending on the option the user picks it will look like this:

DELETE FROM Players

The function will connect to the database and issue the command passed. This will remove all data from the table and if errors occurred then show the errors on the UI.

sqlQuery:
The last main part of the program is the sqlQuery() this will send a query to the database which will return and display the results on the UI. This is done by prompting the user for a query, next the program will connect to the database then query information given.

Run Times:
The graphs below show the runtimes of each type of insertion on a given 3 runtimes with the time in seconds. It is very clear that the run times of single insertion is always 99% longer to complete when compared to Multi insertion.


Players2.txt Insertion Runtime
Single Insertion
Multi Insertion
400
300
200
100
0
Run 1
Run 2
Run 3


Players3.txt Insertion Runtime
Single Insertion
Multi Insertion
500
400
300
200
100
0
Run 1
Run 2
Run 3