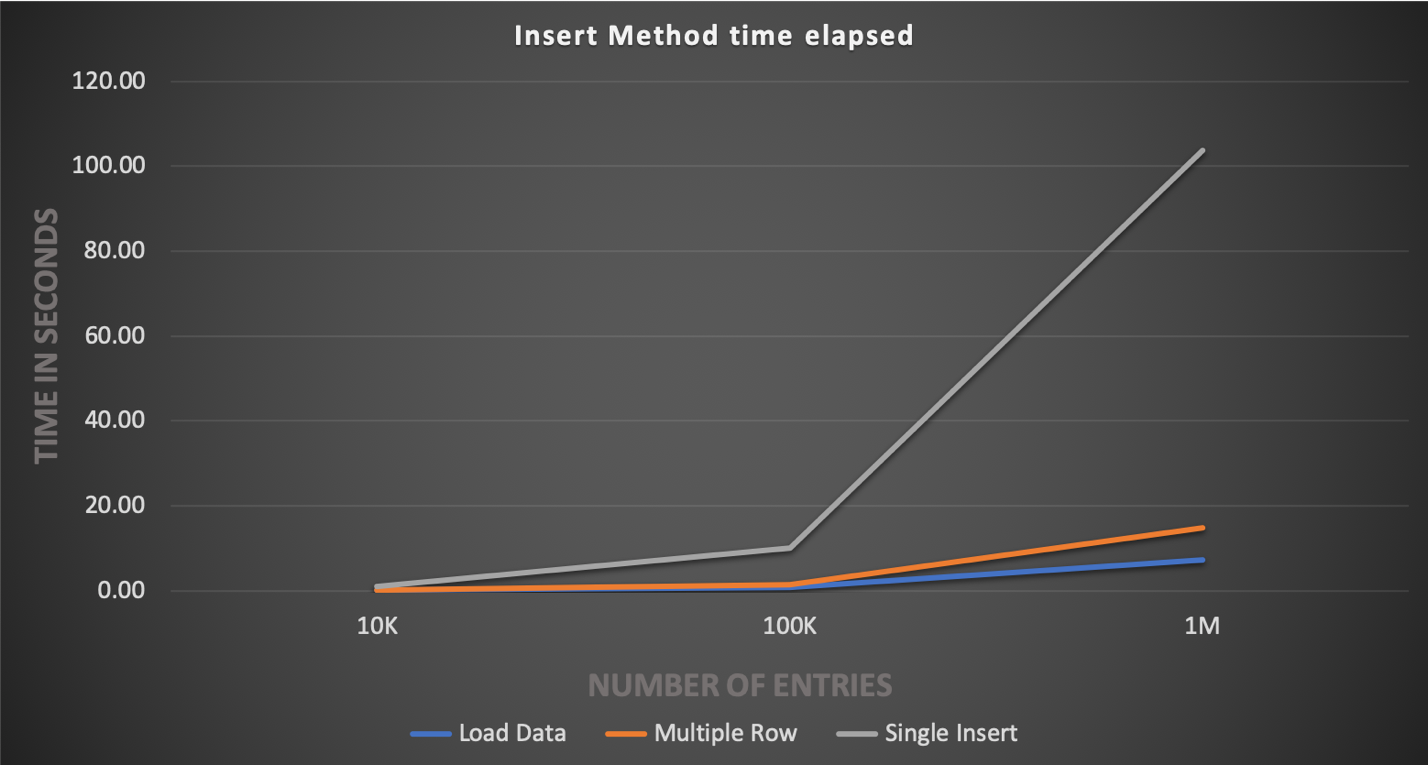
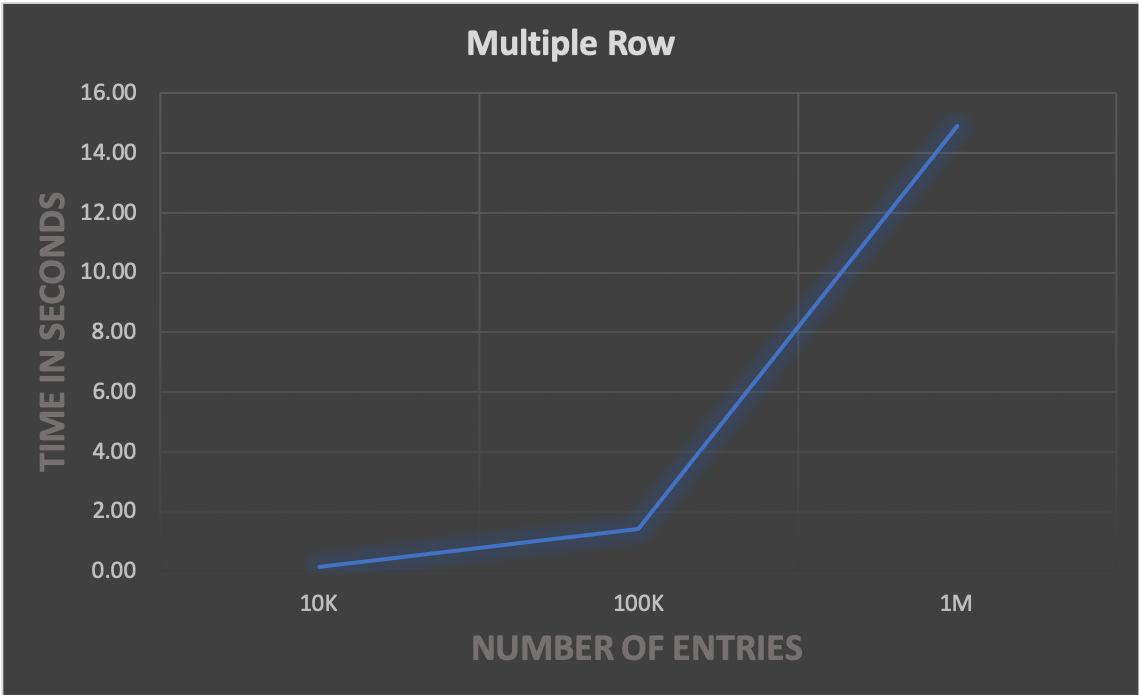
Project: Phase 2

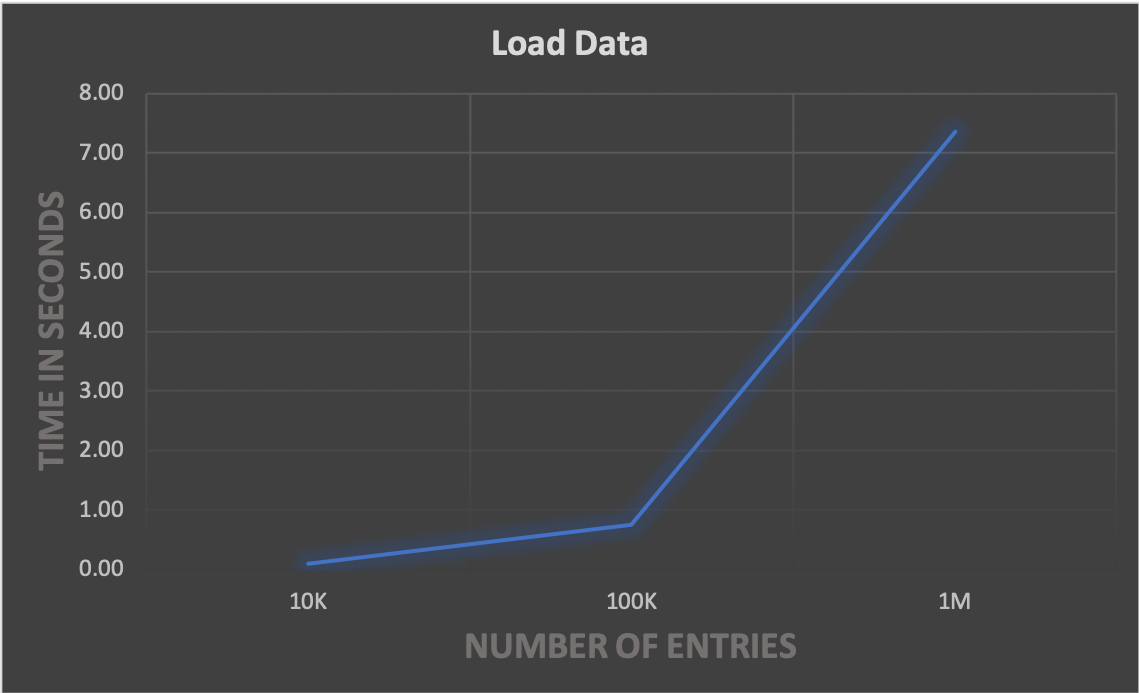
Task 1: Performance Report



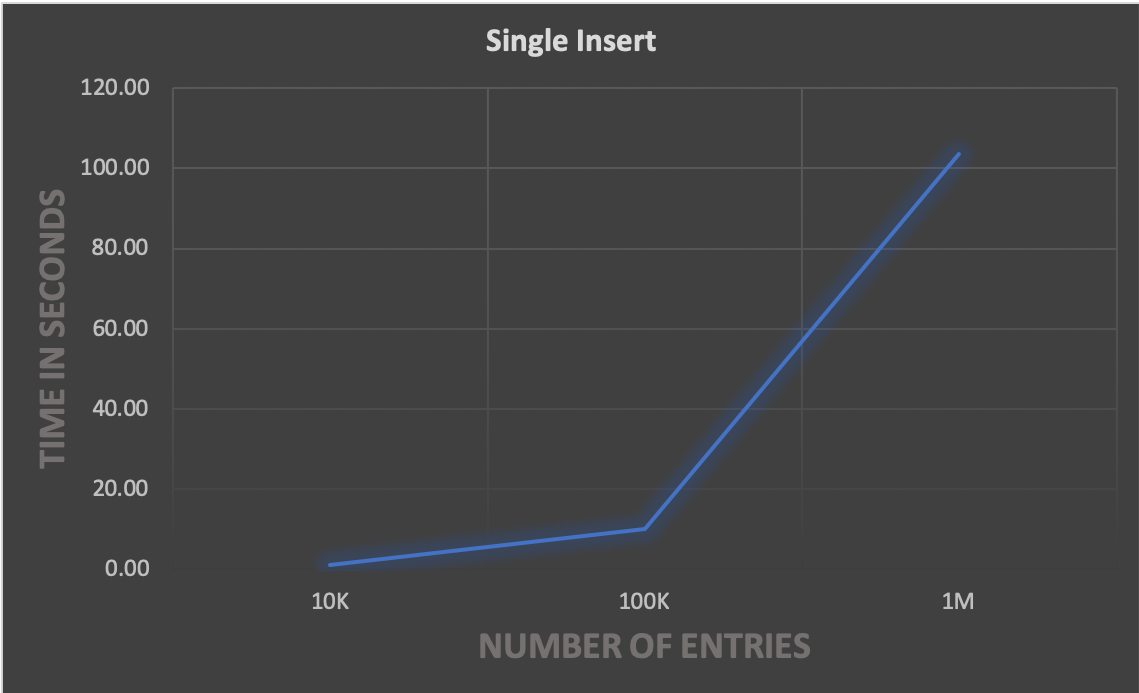
Overall performance of the program has the load data insert method as the fastest and the single insert method as the slowest by a significant amount with all methods growing in insertion time exponentially as the number of elements that are being inserted increases. The speed of the load data insertion method is due to the significant amount of overhead that is reduced over the single insertion methods. Since with every operation the database needs to connect, receive the query, parse the query, insert the row, insert the index, and then close the connection. With the load data method only the insertion of the rows and indexes are a significant portion of the overall runtime. Using single insertion, this entire process is repeated beginning to end for every entry that is made into the database so performance suffers. Performance for all three methods at the lowest tested insert size was fairly similar with there being only a small negligible difference in their insertion speed. This leads us to believe that method selection is not as important when considering performance with a lower number of insertions.



The performance of the multi row insertion method was in between the other two methods with a similar rate of growth as the other two methods. It was only marginally slower than the load data method but also incredibly faster than the single insertion method.



The load data method of insertion was the fastest of the three methods, but it also grew in an exponential form as the number of elements being inserted increased.



Single insertion was the slowest performer of the three with insertion of one million elements taking over 100 seconds. It’s performance when inserting ten thousand elements was comparable to the performance of the load data method with one million elements making it overall ten times slower than load data.