COL362: Database Management Systems
Assignment 1

By-
Nilaksh Agarwal (2015PH10813)
Sagar Goyal (2015CS10253)
Mehak Preet Dhaliwal (2015CS10238)

Timings Observed:
Line by Line Insert: 221 s
Bulk Load: 19s
JDBC Line by Line Insert: 193s
JDBC Batch Insert: 39.4s

Bulk Load:

- The **query doesn't need to be reparsed**, i.e., the control doesn't need to retrieve the query line by line after every execution. This takes time as the file is in the secondary memory and accessing this takes up time.
- The values are transmitted in **one round-trip to the server**, which results in saving the time taken to transfer the values line by line.
- The commands are inside a single transaction which ensures **only the initial and final states are logged**, as compared to every statement in case of single statements. The changes to the database are logged only after the entire operation as compared to after every query.

JDBC:

- We see that the batch insert is slower than the bulk load, since the **JDBC bridge needs to be established**, which takes time. Moreover, having an additional layer to parse the input over takes more time.
- In statement by statement however, we have the **logging aspect** coming into the picture again, since we don't log our JDBC outputs, there is massive time saving there. Moreover, since the bridge needs to be established only once, there is no further time losses due to that.

Line by Line:

- This is the slowest process, since the control has to **output the result to the console** after every query execution. This results in a time delay in line by line.
- The query is **fetched from secondary memory** individually, resulting in large time spent in memory retrieval.