~ ~

A Jasim Azizi ~ UTCS 327E - Databases

Professor Shirley Cohen

TA:- Sameer, Karan

Final Project
Milestone One
MySQL

Background Information on the Project

I assessed the write capability for a database management system. The evaluation was around MySQL's compute engine, in which case I used an instance to launch it and make the proper commands to achieve my results. There were a total of 1 million records inserted into a table called Person which is in a database called load_testing.

The results were astounding. The average time and the standard deviation were found as a result. Batch insert techniques were used in order to rapid the efficiency at which the data is being loaded into the load_testing database. System variable turning as well as hardware resources were used in order to fill the Person table.

My goal was to increase the total capability threshold levels for the central processing unit (CPU) and the Random Access Memory (RAM) in order to assess the interval change of time as opposed to the same 1 million data records being loaded. As a result I found that this project helped me learn quite a lot about how to evaluate a database system. The goal and hope is that this can help me expand my database management skills into other systems.

SECTION ONE

Short Description
Time Results
Observations/Challenges

I used Jupyter Notebook along with Python to load the data into my milestone1.ipynb project. A total of 8 million records were loaded into the Person table. The records were inserted separately from each other. The commits were done at 5000 record points. There are more comments and explanations in the .ipynb file on how insertions were nested using for loops. Below is the code which I used to implement this logic.

```
%%timeit
import mysql.connector

# connecting to mysql DB
connection = mysql.connector.connect(
  host="10.128.0.3",
  user="root",
  password="data",
  database="load_testing",
  autocommit=False
)

print("Connection to MySQL \t ", connection)
print("\n")

sqlData = "INSERT INTO Person (first_name, last_name, company_name, address, city, county, state, zip, phone1, phone2, email, web)\
    VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)"

# to keep track of the cursor
counterCursor = 0

# Per question, as commented above
# TRY
try:
    for n, row in df_us_1000000.iterrows():

        record = (row['first_name'], row['last_name'], row['company_name'], row['address'], row['city'],\
               row['county'], row['state'], row['zip'], row['phone1'], row['phone2'], row['email'], row['web'])
        cursor = connection.cursor()
        cursor.execute(sqlData, record)
        counterCursor += cursor.rowcount

        if (n + 1) % 5000 == 0:
            connection.commit()
            print(counterCursor, "records inserted successfully into Person table")
            cursor.close()
# EXCEPT
except mysql.connector.Error as error:
    print("Unsuccessful to insert record(s) into Person table {}".format(error))

# FINALLY
finally:
    if connection.is_connected():
        connection.close()
        print("MySQL connection is closed")
```

```
Connection to MySQL       <mysql.connector.connection_cext.CMySQLConnection object at 0x7f0d22c33d50>
```