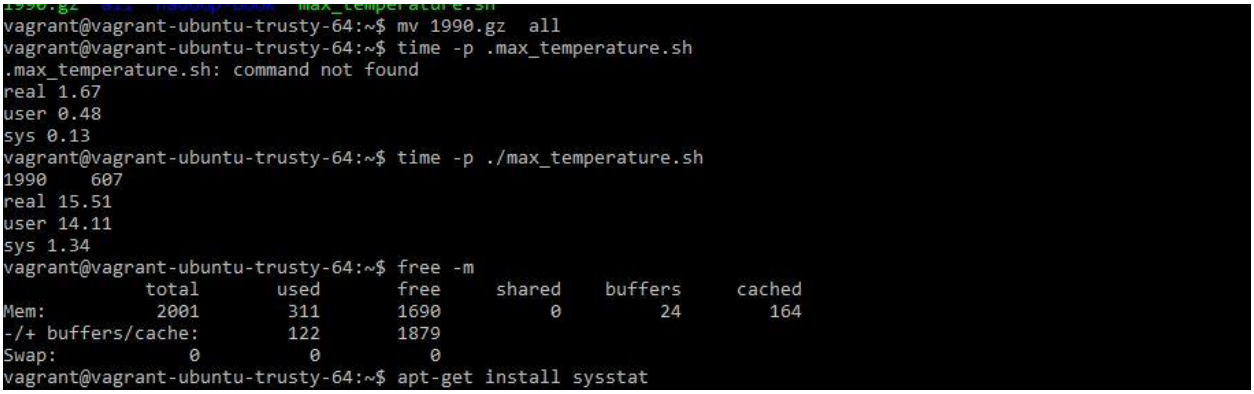
Part 0:

Link for introduction: https://github.com/illinoistech-itm/agupta110

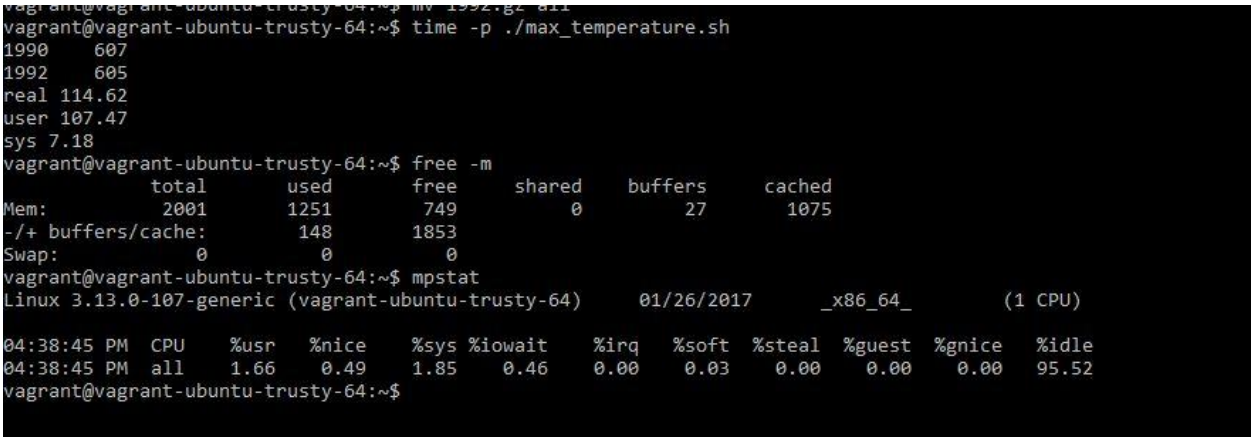
Link for image: https://github.com/illinoistech-itm/agupta110/issues

Part 1:

Dataset 1: Year 1990



Dataset 2: Year 1990 and 1992

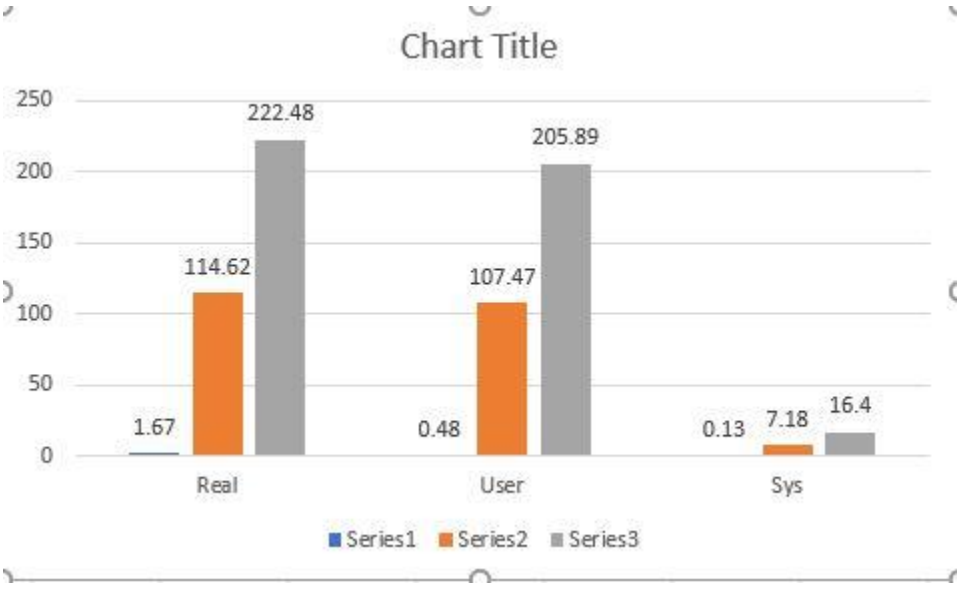


Dataset 3: All years

```
vagrant@vagrant-ubuntu-trusty-64:~$ mv 1993.gz all
vagrant@vagrant-ubuntu-trusty-64:~$ time -p ./max_temperature.sh
1990    607
1991    607
1992    605
1993    567
real 222.48
user 205.89
sys 16.40
vagrant@vagrant-ubuntu-trusty-64:~$ free -m
             total       used       free     shared    buffers     cached
Mem:          2001       1934         67          0          1       1784
-/+ buffers/cache:        148       1853
Swap:            0          0          0
vagrant@vagrant-ubuntu-trusty-64:~$ mpstat
Linux 3.13.0-107-generic (vagrant-ubuntu-trusty-64)     01/26/2017      _x86_64_        (1 CPU)

04:45:52 PM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
04:45:52 PM  all    3.50    0.47    1.98    0.45    0.00    0.03    0.00    0.00    0.00   93.56
vagrant@vagrant-ubuntu-trusty-64:~$
```

Comparison:



Series 1: Dataset 1

Series 2: Dataset 2

Series 3: Dataset 3

Since Dataset 1 has only 1 year's records, its execution time is the least. Whereas since Dataset 2 has 2 year's records, its execution time is more than that of dataset 2. Dataset 3 has records of 4 years so its dataset is the biggest hence its execution time is the highest.

Part 2.

My code is not running.

I broke down the components into various sub parts. I used the internet to see the code.

These are as follows:

1. I first made a connection to the database. I used the internet to see the code.
   Reference: https://www.tutorialspoint.com/jdbc/jdbc-sample-code.htm

2. After that used the concept of FileInput Stream and GZIP input stream to read the file.
   Reference: http://www.java2s.com/Code/Java/File-Input-Output/Readsomedatafromagzipfile.htm

3. Then I used substring concept using the explanation of the data from the book and found the indices of the substring as taught in the class. I used the below explanation to decide the indices.

Example 2-1. Format of a National Climatic Data Center record

```
0057
332130   # USAF weather station identifier
99999    # WBAN weather station identifier
19500101 # observation date
0300     # observation time
4
+51317   # latitude (degrees x 1000)
+028783  # longitude (degrees x 1000)
FM-12
+0171    # elevation (meters)
99999
V020
320      # wind direction (degrees)
1        # quality code
N
0072
1
00450    # sky ceiling height (meters)
1        # quality code
C
N
010000   # visibility distance (meters)
1        # quality code
N
9
-0128    # air temperature (degrees Celsius x 10)
1        # quality code
-0139    # dew point temperature (degrees Celsius x 10)
1        # quality code
10268    # atmospheric pressure (hectopascals x 10)
1        # quality code
```

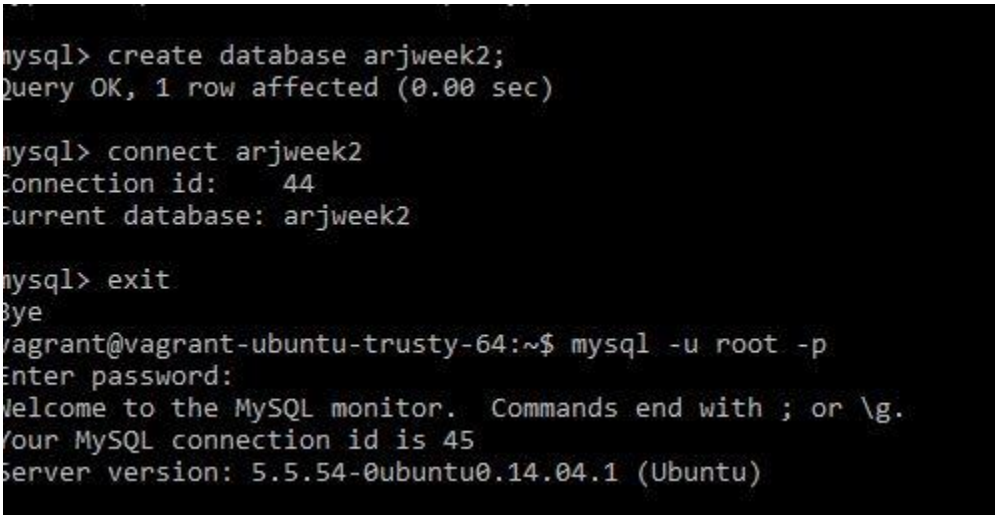
Book: Hadoop: The Definitive Guide, 4th Edition Storage and Analysis at Internet Scale, By Tom White

4. I finally read the internet and used preparedstmt.setstring() to insert and finalise the statements.

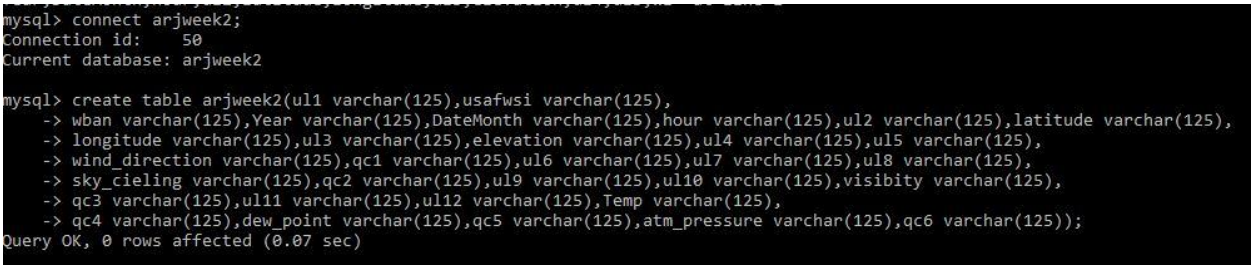
   Reference:
   http://www.java2s.com/Code/JavaAPI/java.sql/PreparedStatementsetStringintparameterIndexStringx.htm

Some screenshots:

1) Database creation



2) Schema creation



3) Error message