**Temperature analysis project using Bigdata and AWS EMR**

**PART 1: The first part is to develop a Mapper and Reducer application to retrieve Year and Temperature from original NCDC records (i.e., the dataset we are using for this class) and then write the Year and Temperature data into a text file.**

**Command to compile java files:**

javac -classpath /home/student17/jar4compile/hadoop-common-2.6.1.jar:/home/student17/jar4compile/hadoop-mapreduce-client-core-2.6.1.jar:/home/student17/jar4compile/commons-cli-2.0.jar -d . Temperature.java TemperatureMapper.java TemperatureReducer.java

**Command to Create the jar files:**

jar -cvf temperature.jar Temperature\*.class

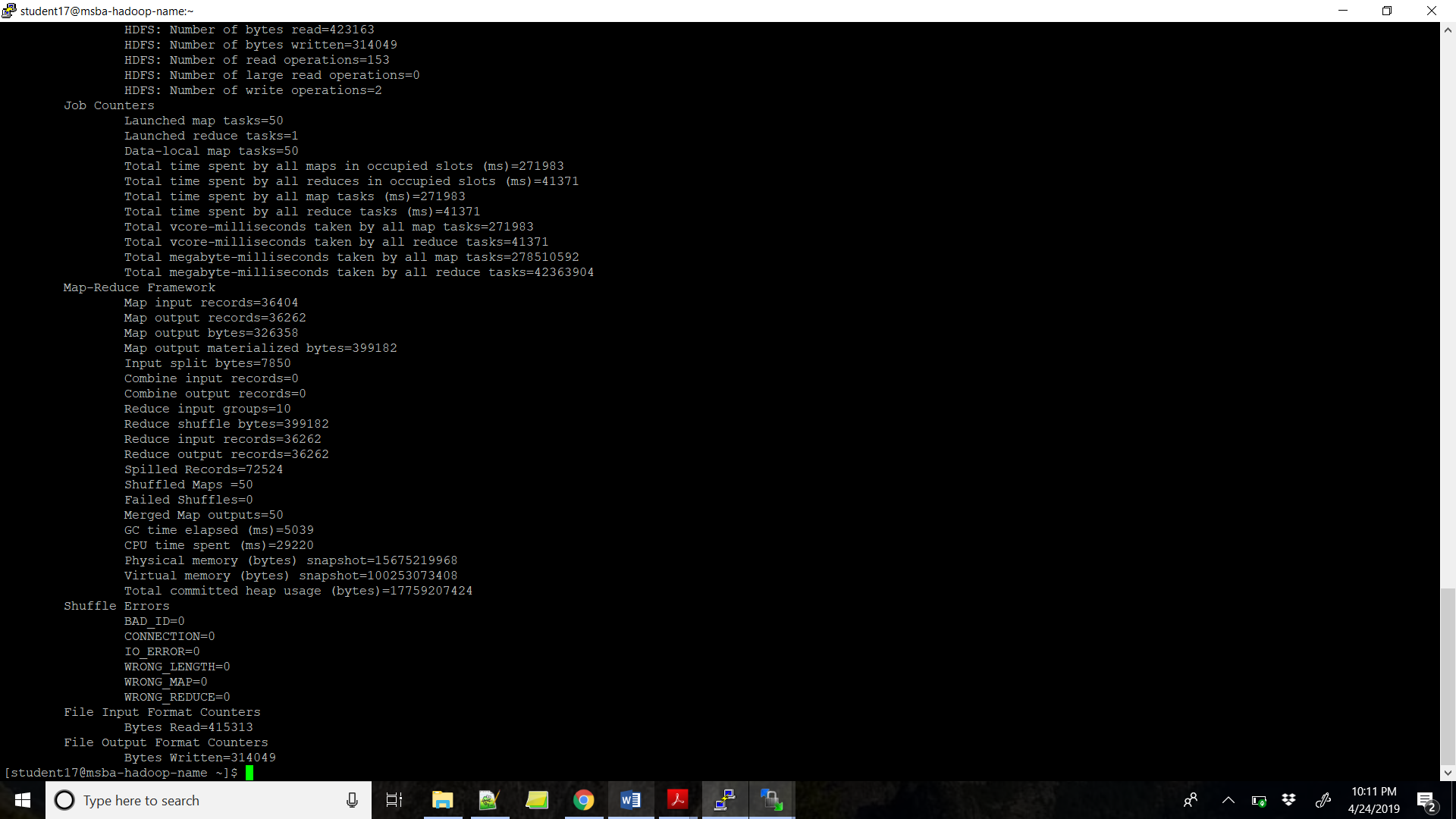
**Command to copy datafile to hdfs:**

hdfs dfs -copyFromLocal CourseProjectData\_BigData /home/17student17/

**Command to run Hadoop jar:**

hadoop jar /home/student17/temperature.jar Temperature /home/17student17/CourseProjectData\_BigData /home/17student17/project\_output3

**Mapper and reducer ran successfully. Output as shown in screenshot below:**

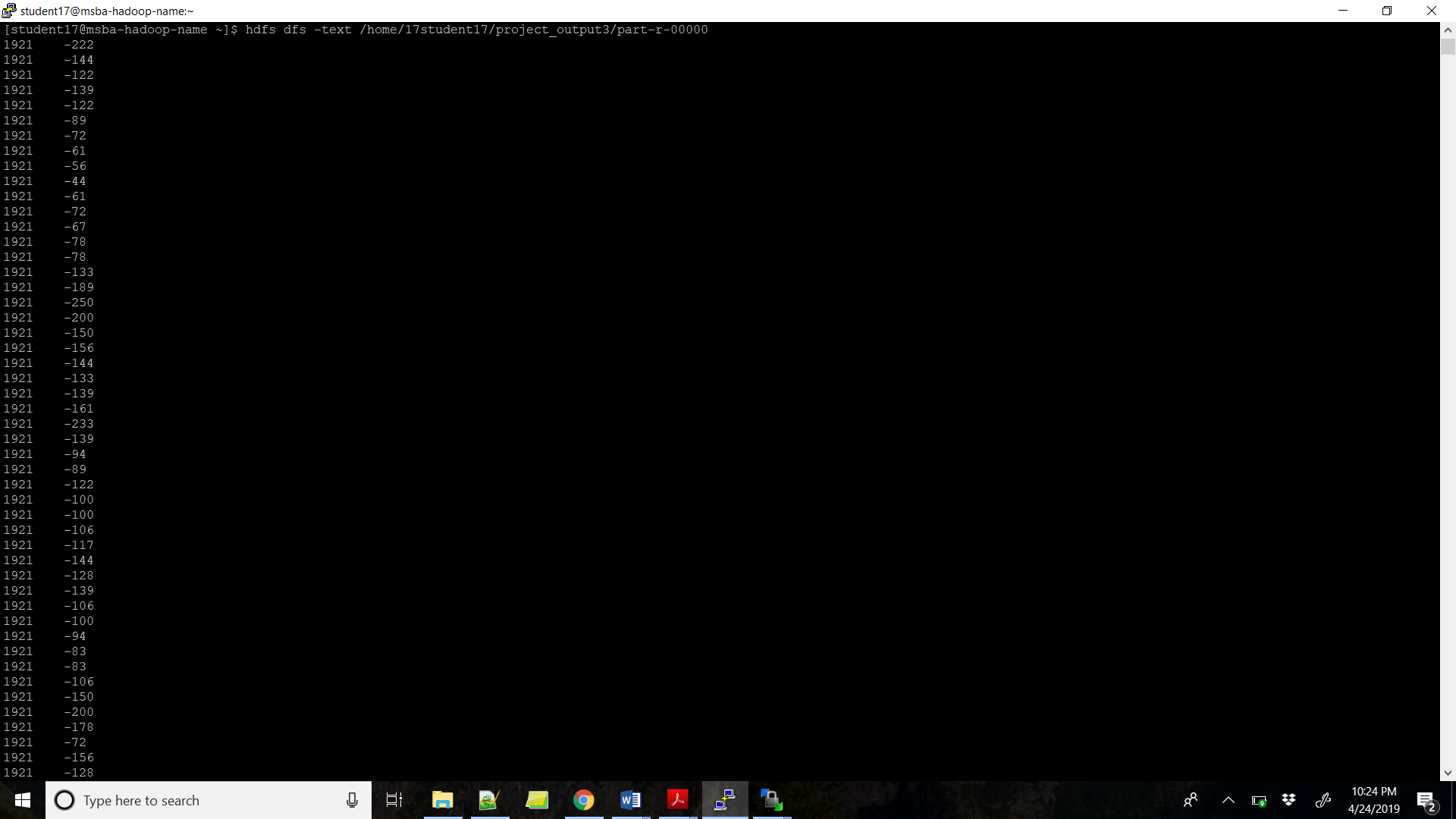


**Command to display text file:**

hdfs dfs -ls /home/17student17/project\_output3/

hdfs dfs -text /home/17student17/project\_output3/part-r-00000

**OUTPUT:**



**Command to copy the folder from hdfs to local drive:**

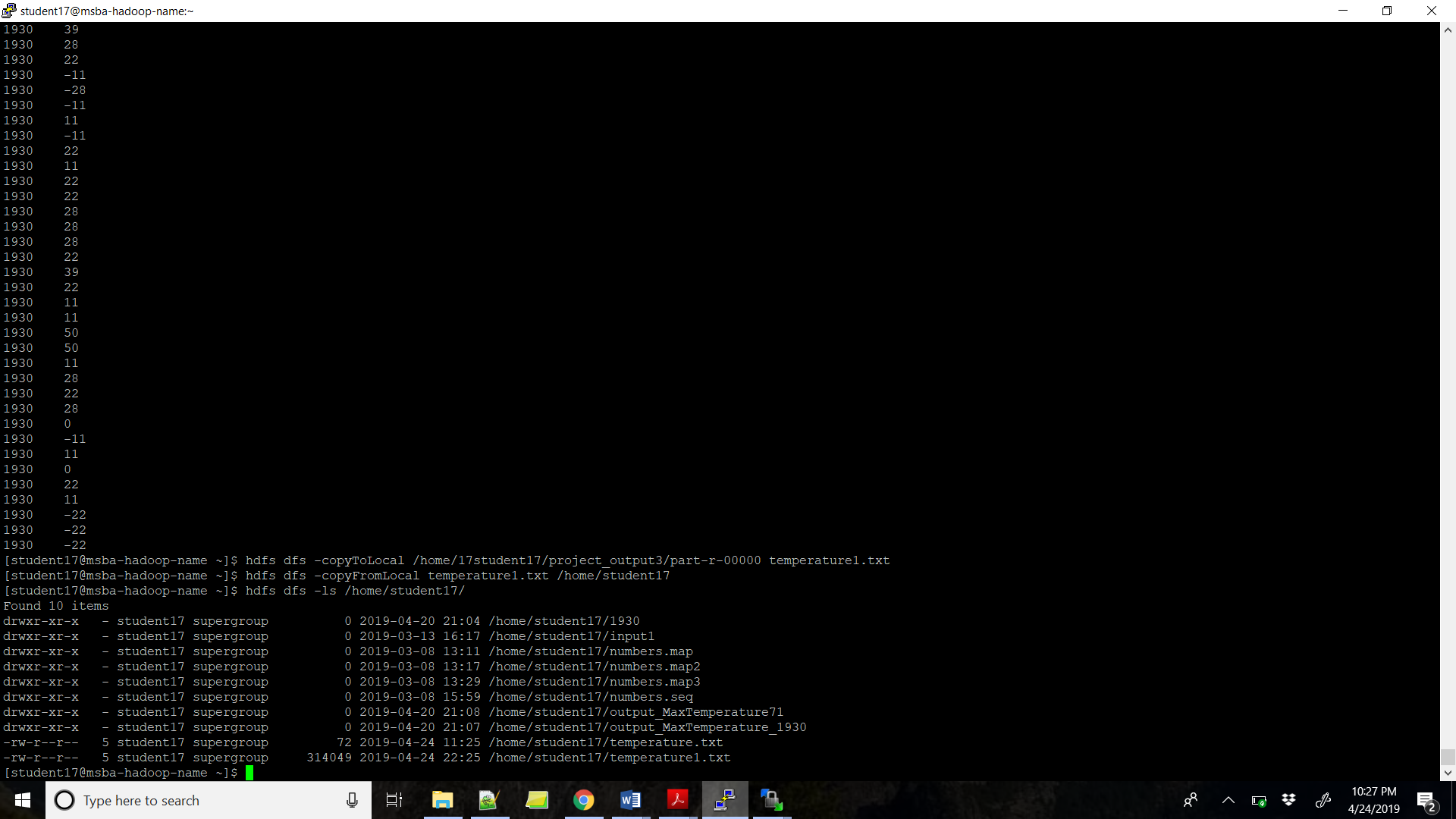
hdfs dfs -copyToLocal /home/17student17/project\_output3/part-r-00000 temperature1.txt

**Command to copy file from local drive to hdfs temperature1.txt:**

hdfs dfs -copyFromLocal temperature1.txt /home/student17

hdfs dfs -ls /home/student17/

**OUTPUT- temperature1.txt file created successfully in student17:**



**PART 2: The second part is to load the text file into Pig and get the highest and lowest temperatures for each year.**

**Command to run the job in pig:**

pig -x local;

**Command to load text file to table in pig:**

temperature\_records = LOAD 'temperature1.txt' AS (year:int, temperature:int);

grouped\_records = GROUP temperature\_records BY year;

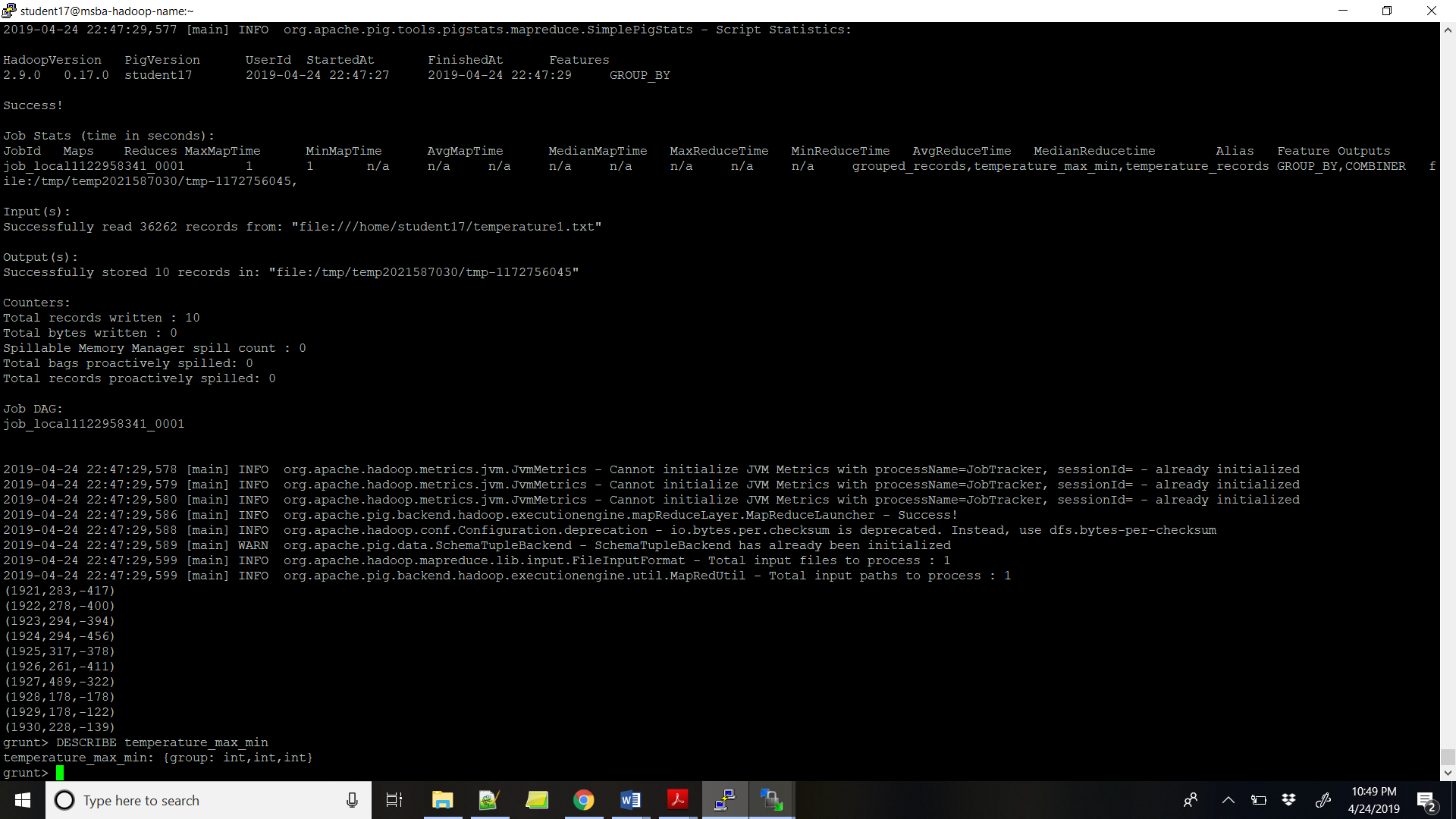
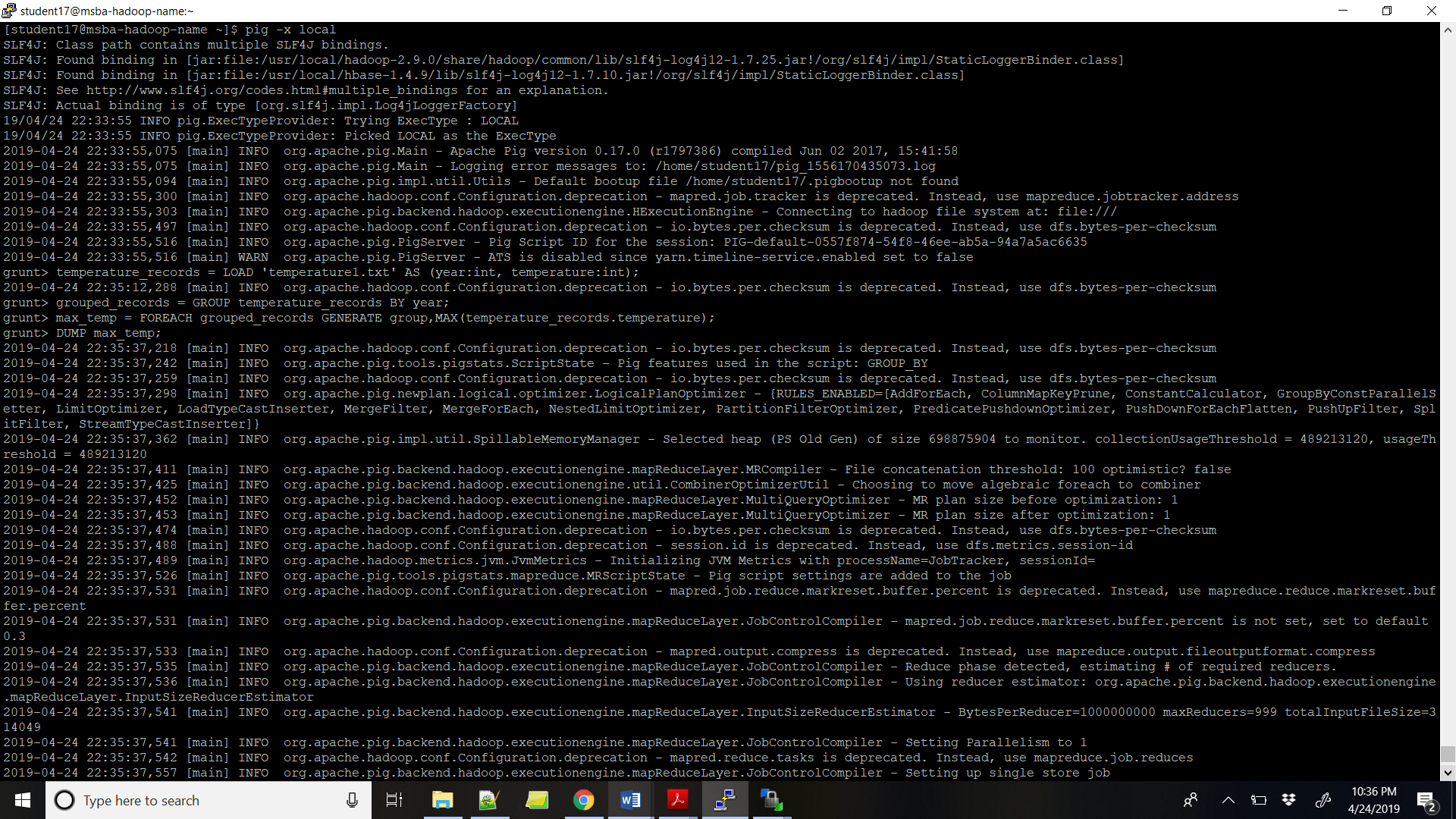
**Command to run highest and lowest temperature for each year:**

temperature\_max\_min = FOREACH grouped\_records GENERATE group, MAX(temperature\_records.temperature),MIN(temperature\_records.temperature);

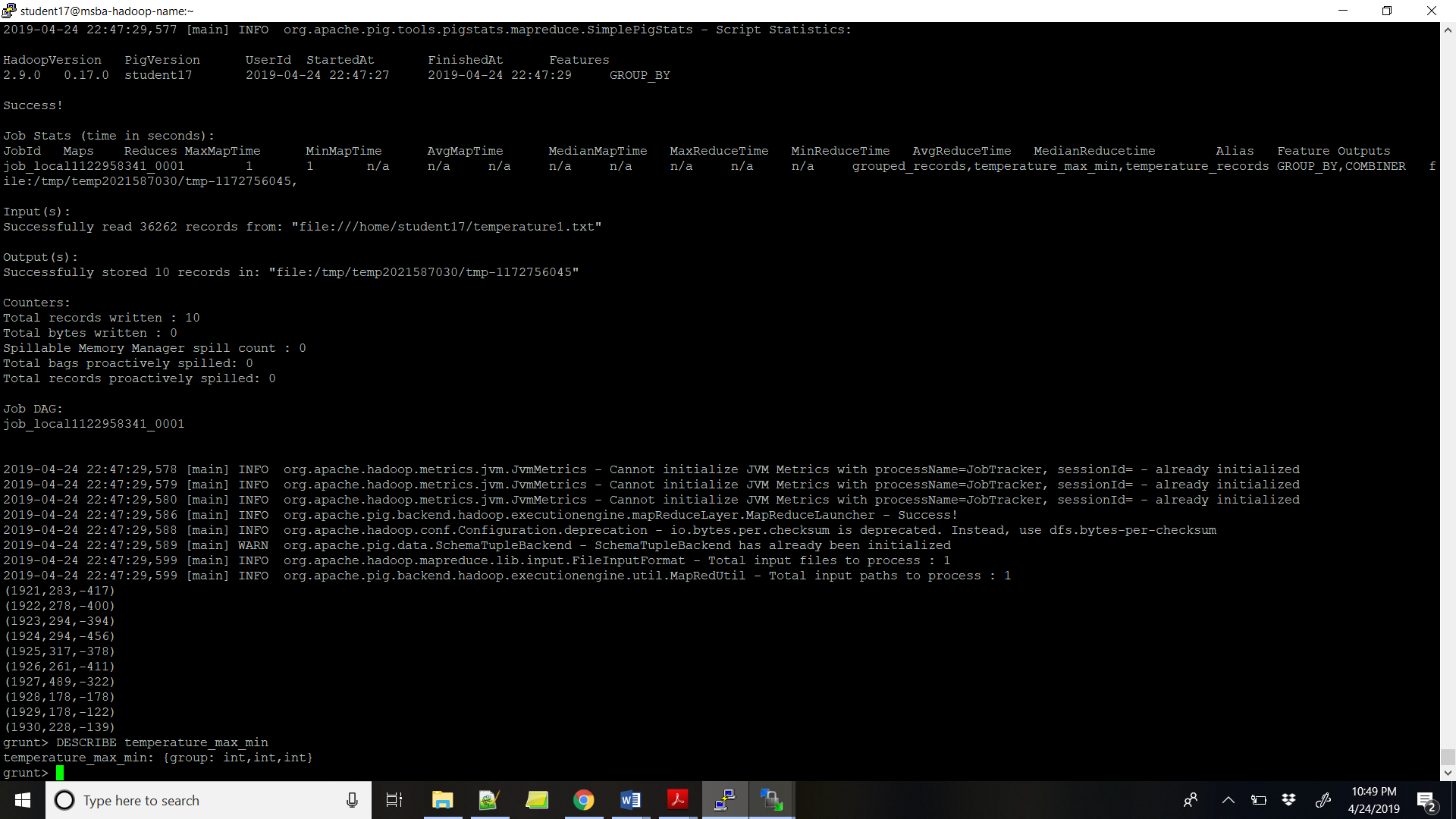
**Command to see final output as (year, Maximum temperature, Minimum temperature):**

DUMP temperature\_max\_min;

**FINAL OUTPUT:**

****

**BELOW IS THE FINAL OUTPUT FROM ABOVE SECREENSHOT (Year, Highest Temperature, Lowest Temperature) (Zoomed):**



**PART 3: The third part is to load the text file into Hive and get the average temperature for each year.**

**Commands to active hive on Hadoop:**

#ls -l | grep meta

#mv metastore\_db metastore\_db.old

#schematool -dbType derby -initSchema

#hive

set hive.metastore.warehouse.dir;

**Command to drop Table if already exists:**

DROP TABLE IF EXISTS temperature\_records;

**Commands to create table in Hive:**

! echo == avg\_temp\_select;

CREATE TABLE temperature\_records1 (year STRING, temperature INT)

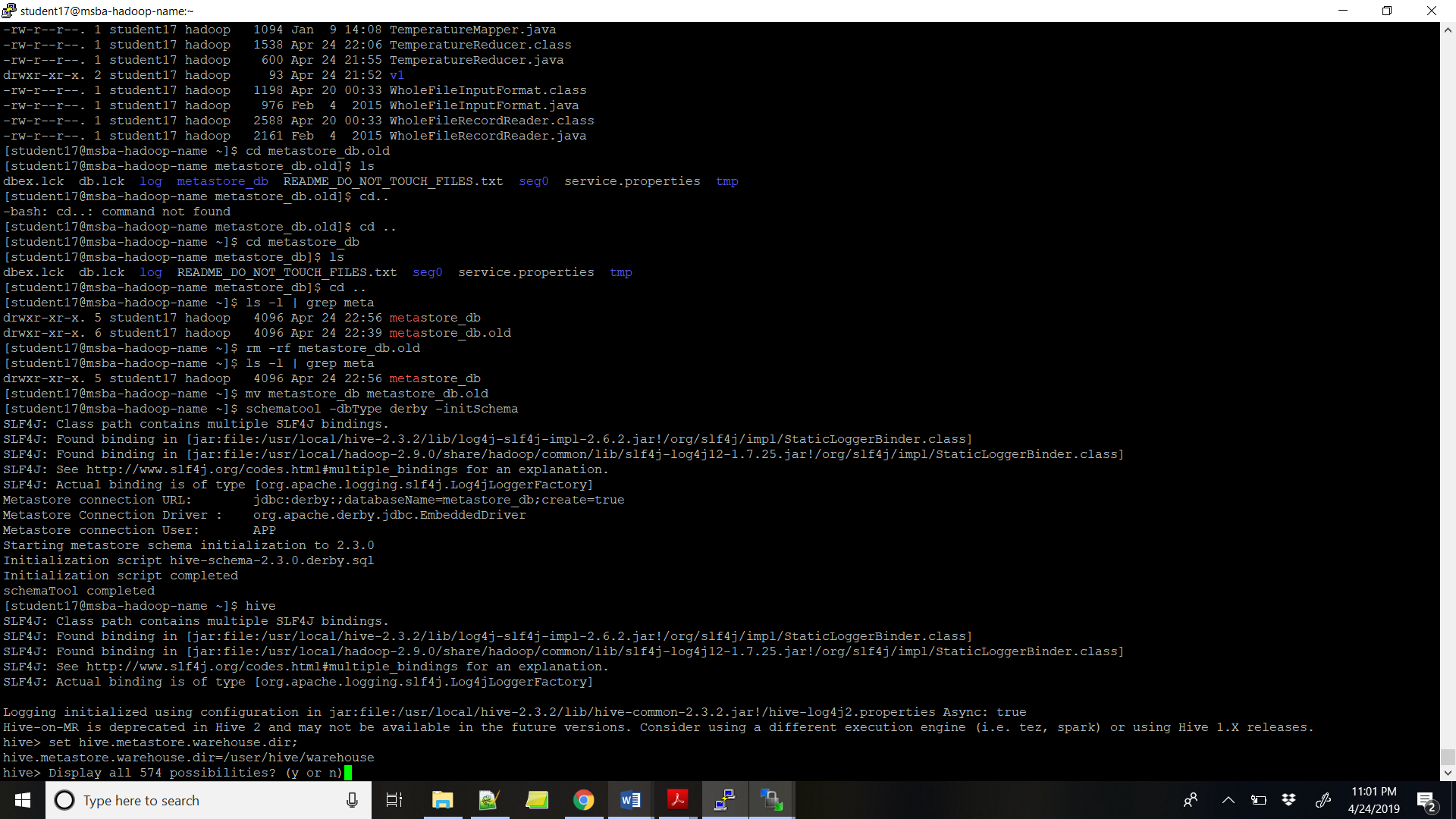
ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t';

**Command to load data from temperature1.txt to table:**

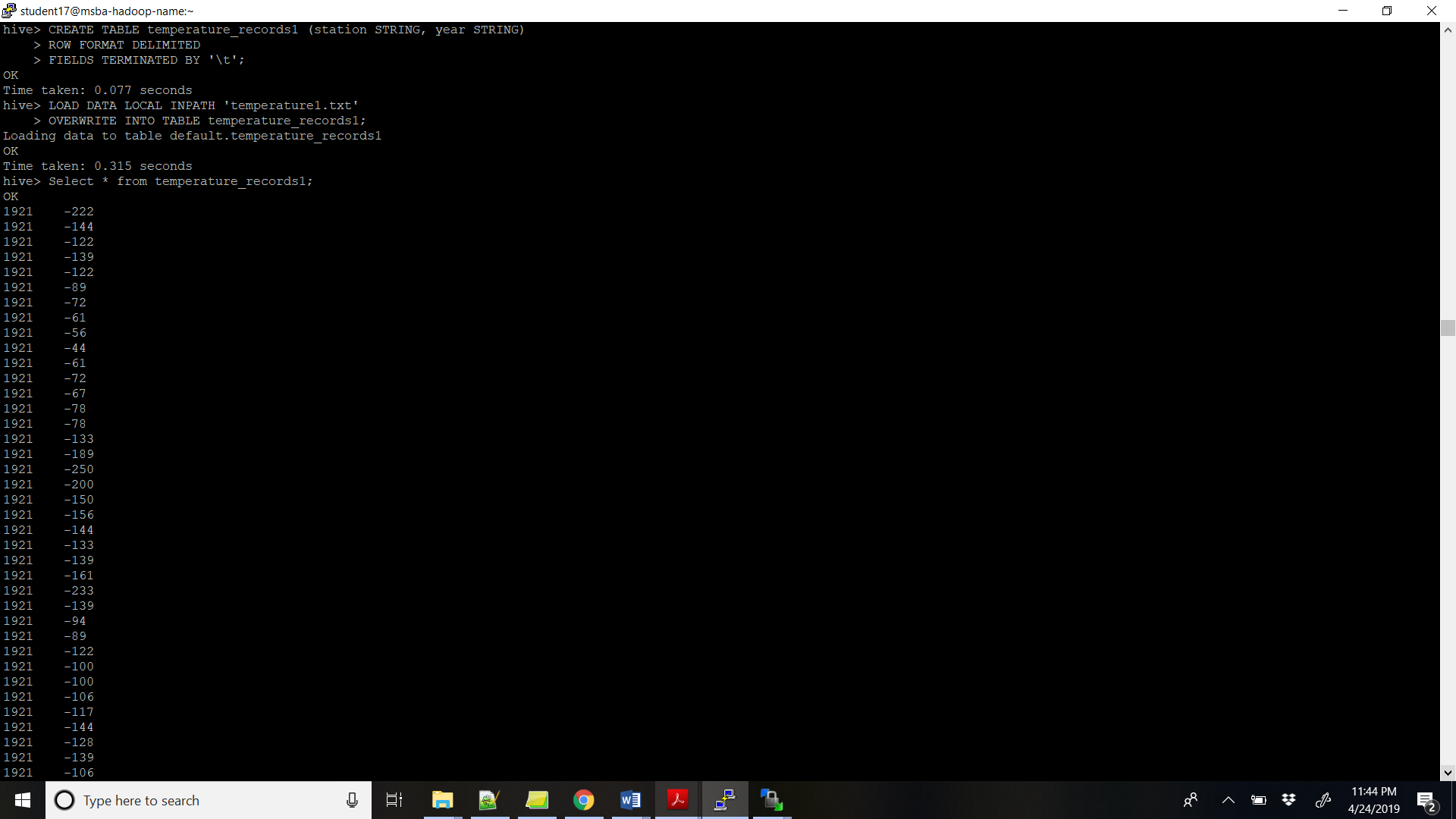
LOAD DATA LOCAL INPATH 'temperature1.txt'

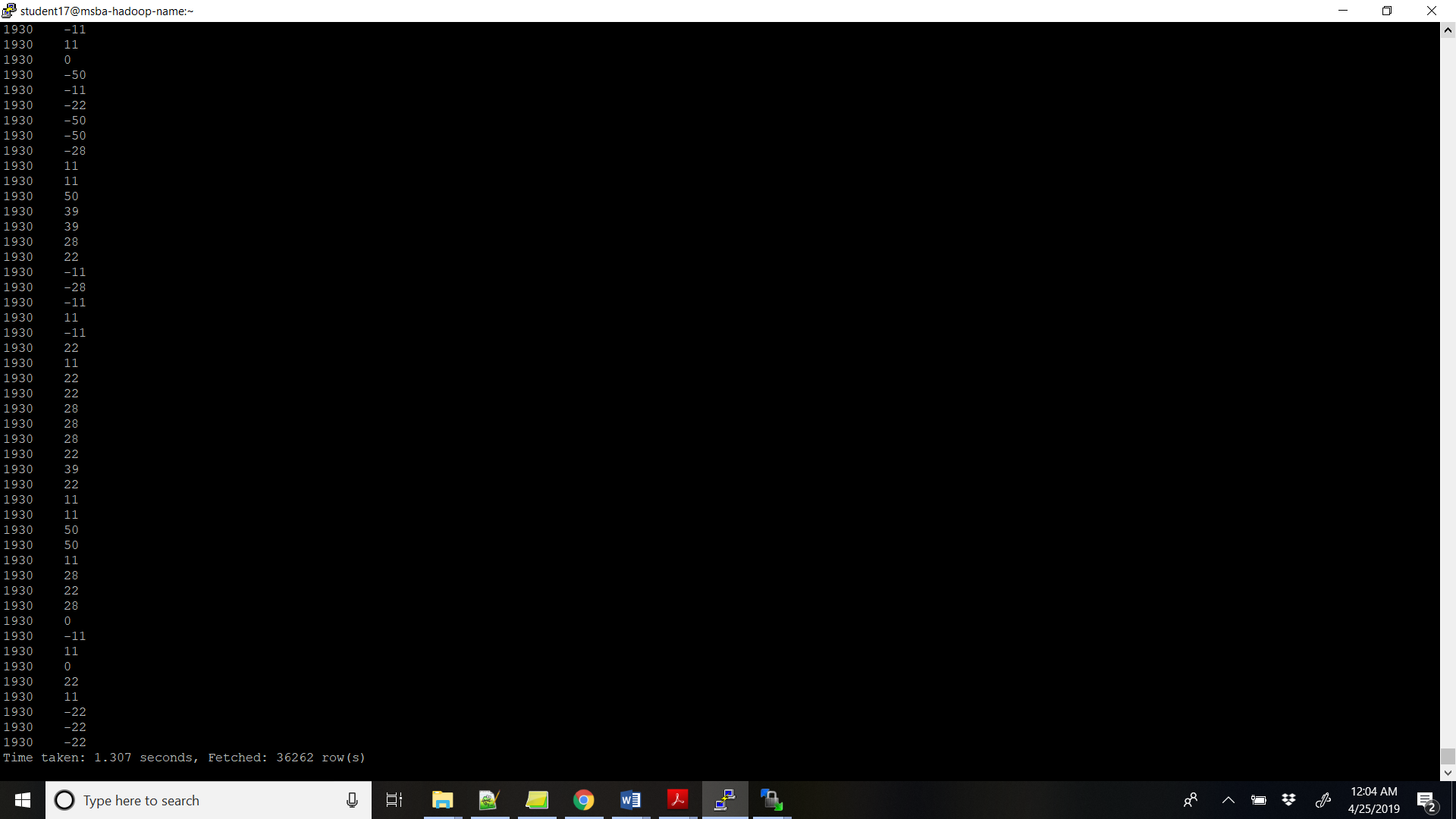
OVERWRITE INTO TABLE temperature\_records1;

**OUTPUT:**

**Comand to show table temperature\_records1:**

Select \* from temperature\_records1;





**Command to show average temperature for each year:**

! echo vv avg\_temp\_select;

SELECT year, AVG(temperature)

FROM temperature\_records1

WHERE temperature != 9999

GROUP BY year;

! echo avg\_temp\_select;

**FINAL OUTPUT (Year, Average of Temperature):**

