Project 2

1. *Work the five examples on pages 384 and 385, that is, from the title “Schemas” to the subtitle “Validation and nulls.”* ***P****rint out a screenshot or a couple of screenshots for the results of these five examples in CSUEB Hadoop.*

Commands for running in Pig:

records = LOAD '/home/student8/sample.txt' AS (year:int, temperature:int, quality:int);

DESCRIBE records;

records = LOAD '/home/student8/sample.txt' AS (year, temperature, quality);

DESCRIBE records;

records = LOAD '/home/student8/sample.txt'AS (year, temperature:int, quality:int);

DESCRIBE records;

records = LOAD '/home/student8/sample.txt';

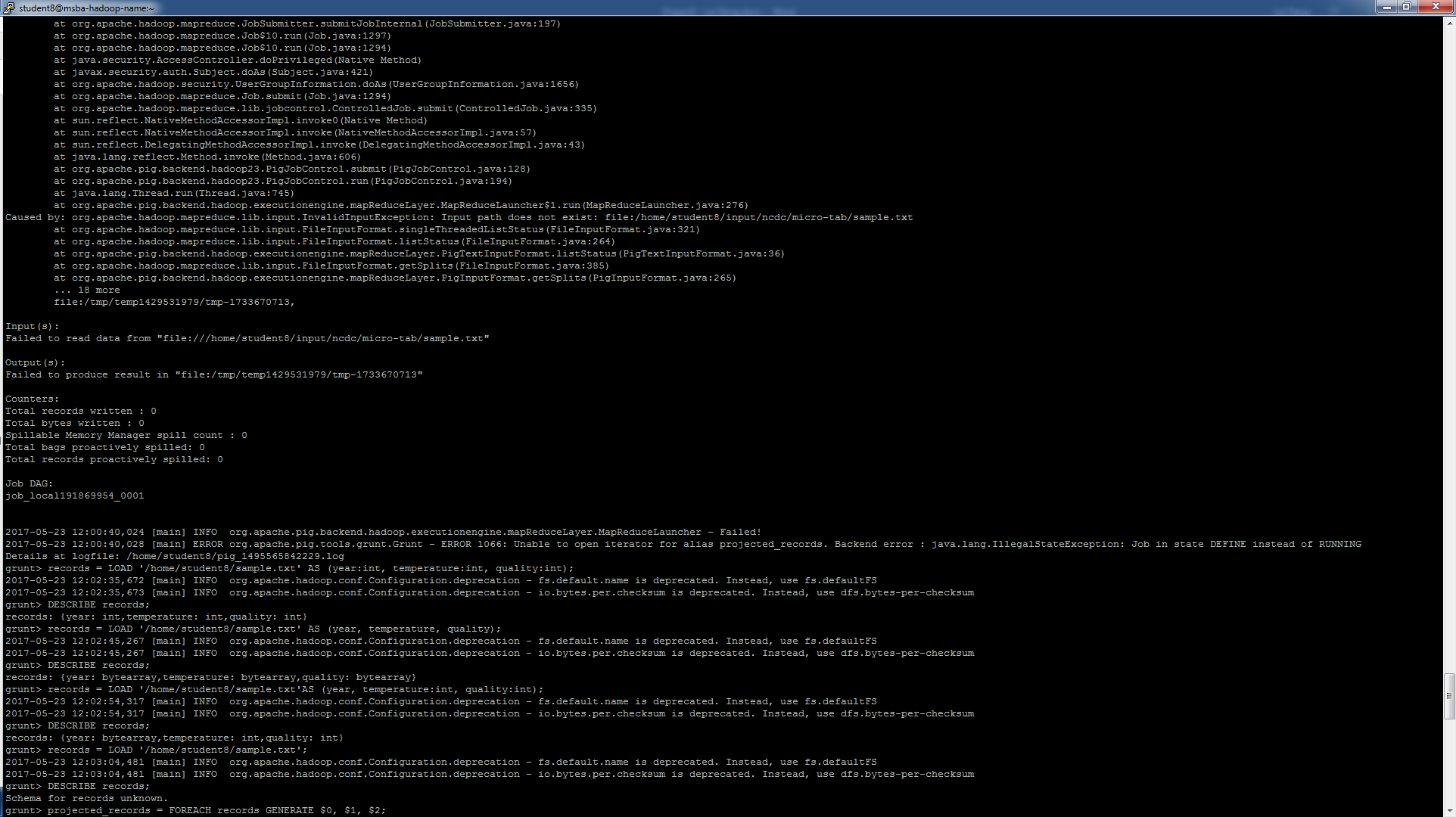
DESCRIBE records;

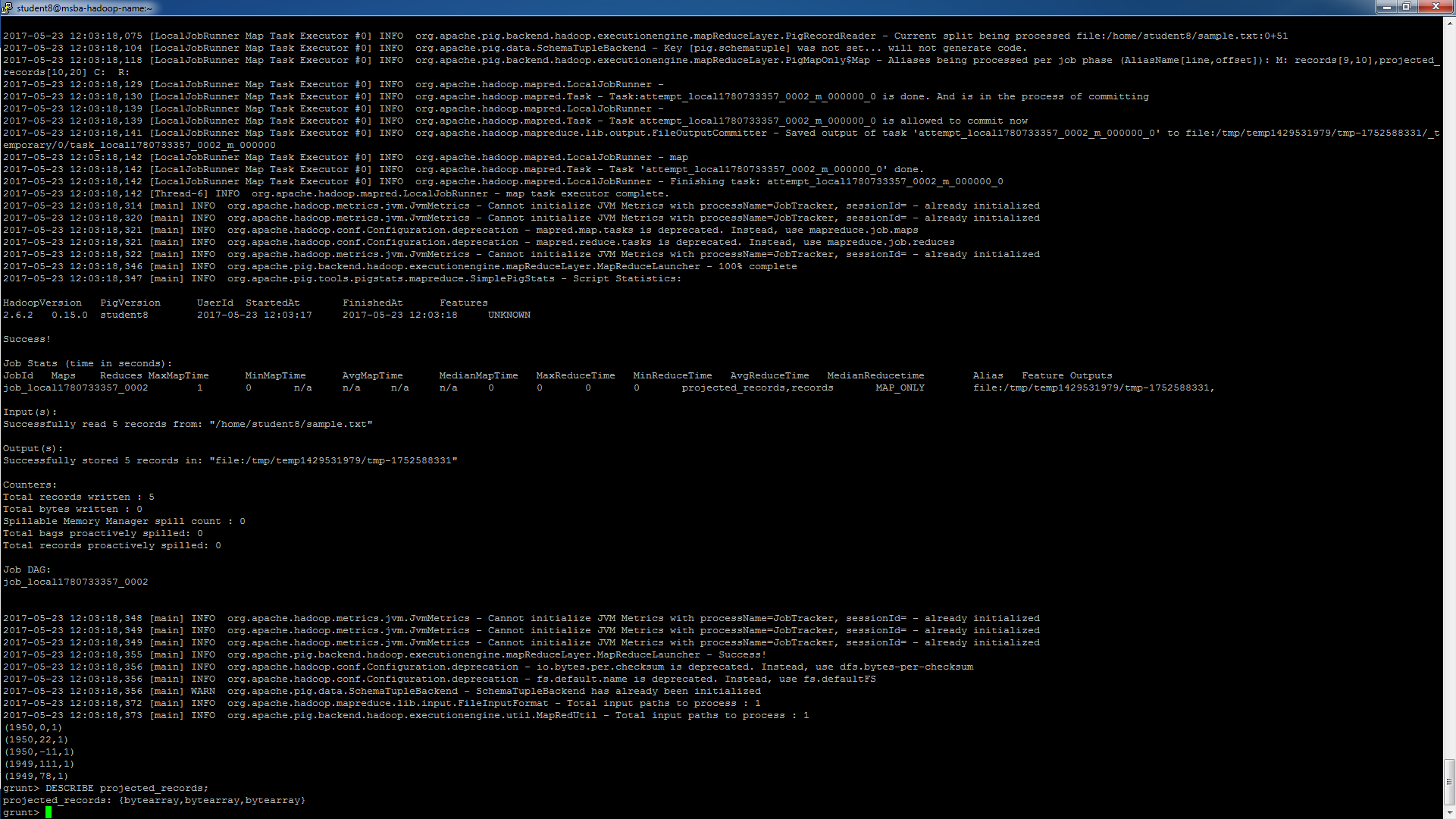
projected\_records = FOREACH records GENERATE $0, $1, $2;

DUMP projected\_records;

DESCRIBE projected\_records;

Screenshot of final results in CSUEB:





1. *Work “A Load UDF” example on pages 396 and 397. Type out all the commands in each step of the process and print out a screenshot of the final results in CSUEB Hadoop.*

*Note. 1) This example also has Range.java, which is located in the same folder as CutLoadFunc.java. 2) Need pig-0.11.0.jar and commons-logging-1.2.jar for compilation. The two jars are available in ITM6273-JarForCompile on Blackboard.*

Commands for running in Hadoop:

javac -classpath /home/student8/hadoop-common-2.6.1.jar:/home/student8/hadoop-mapreduce-client-core-2.6.1.jar:/home/student8/commons-cli-2.0.jar:/home/student8/pig-0.11.0.jar:/home/student8/commons-logging-1.2.jar -d . CutLoadFunc.java Range.java

jar -cvf cut-load-func.jar com/hadoopbook/pig/CutLoadFunc.class

jar -cvf range.jar com/hadoopbook/pig/Range.class

Commands for running in Pig:

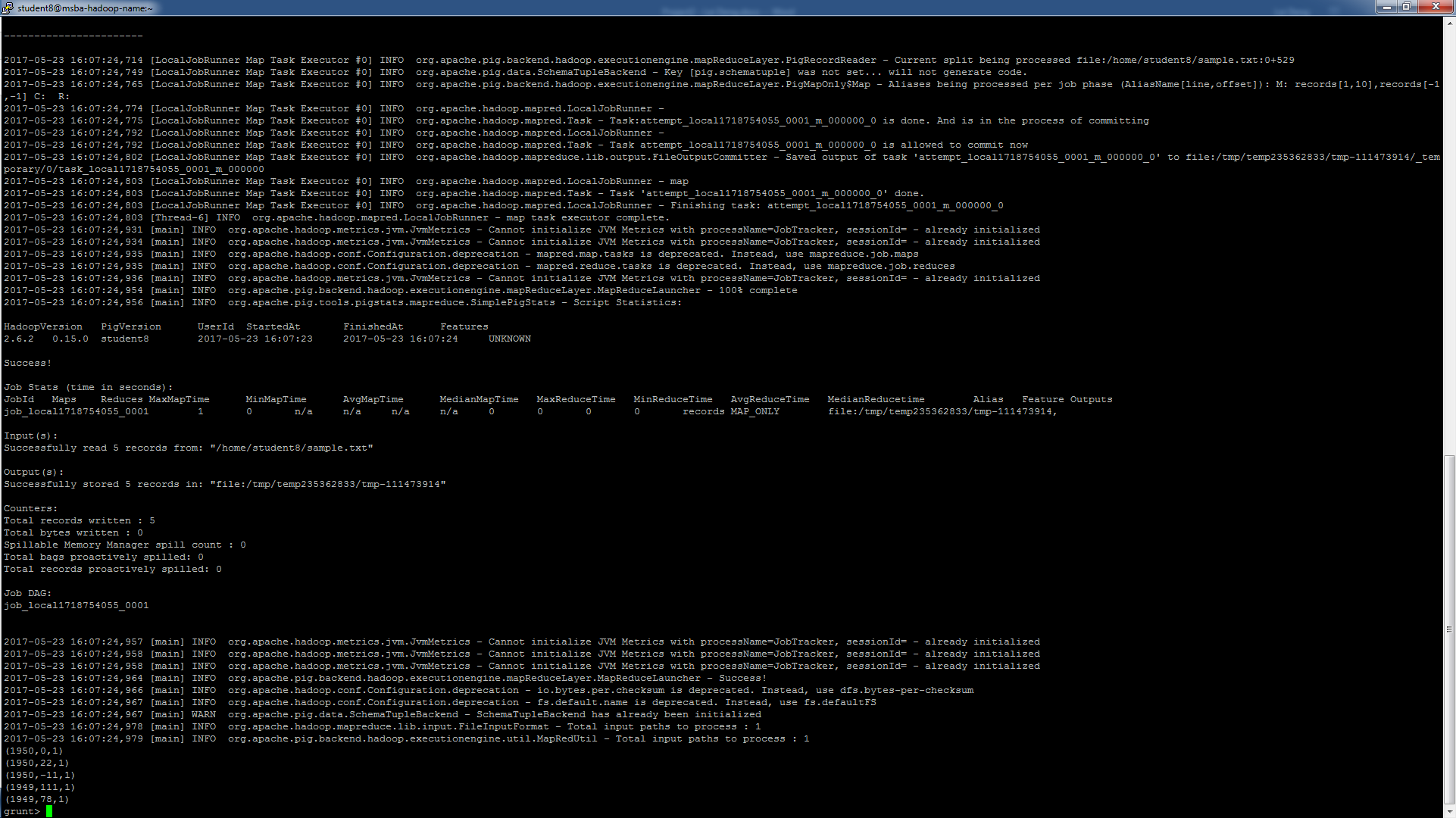
REGISTER cut-load-func.jar;

REGISTER range.jar;

records = LOAD '/home/student8/sample.txt' USING com.hadoopbook.pig.CutLoadFunc('16-19,88-92,93-93') AS (year:int, temperature:int, quality:int);

DUMP records;

Screenshot of final results in CSUEB:



1. *Work the four examples on pages 416 and 417, that is, from the title “An Example” to the title “Running Hive.” Type out all the commands in each step of the process and print out a screenshot of the final results in CSUEB Hadoop.*

Commands for running in Hive:

CREATE TABLE records (year STRING, temperature INT, quality INT)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t';

LOAD DATA LOCAL INPATH '/home/student8/sample.txt'

OVERWRITE INTO TABLE records;

hdfs dfs -ls /user/hive/warehouse/records

SELECT year, MAX(temperature)

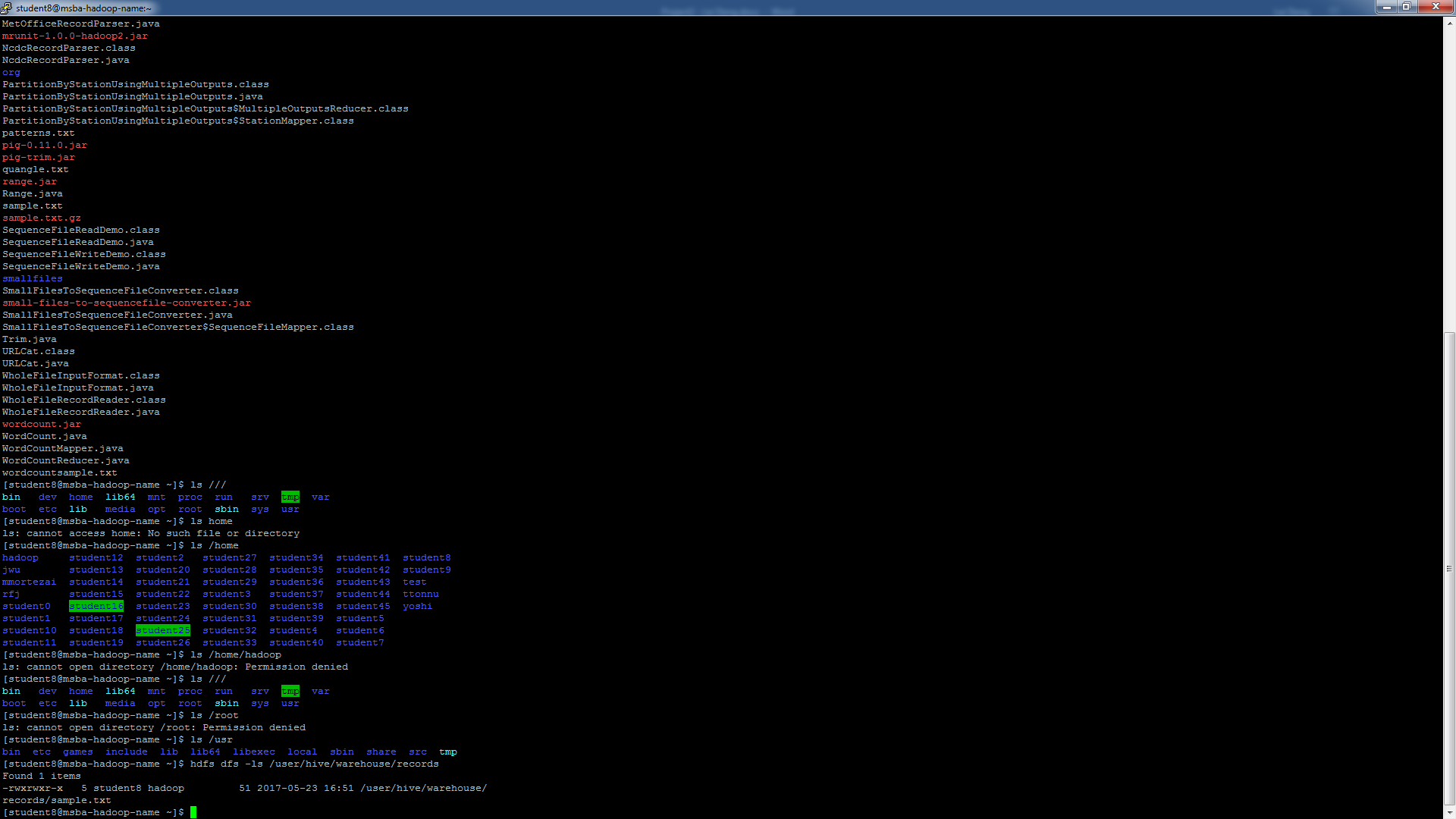
FROM records

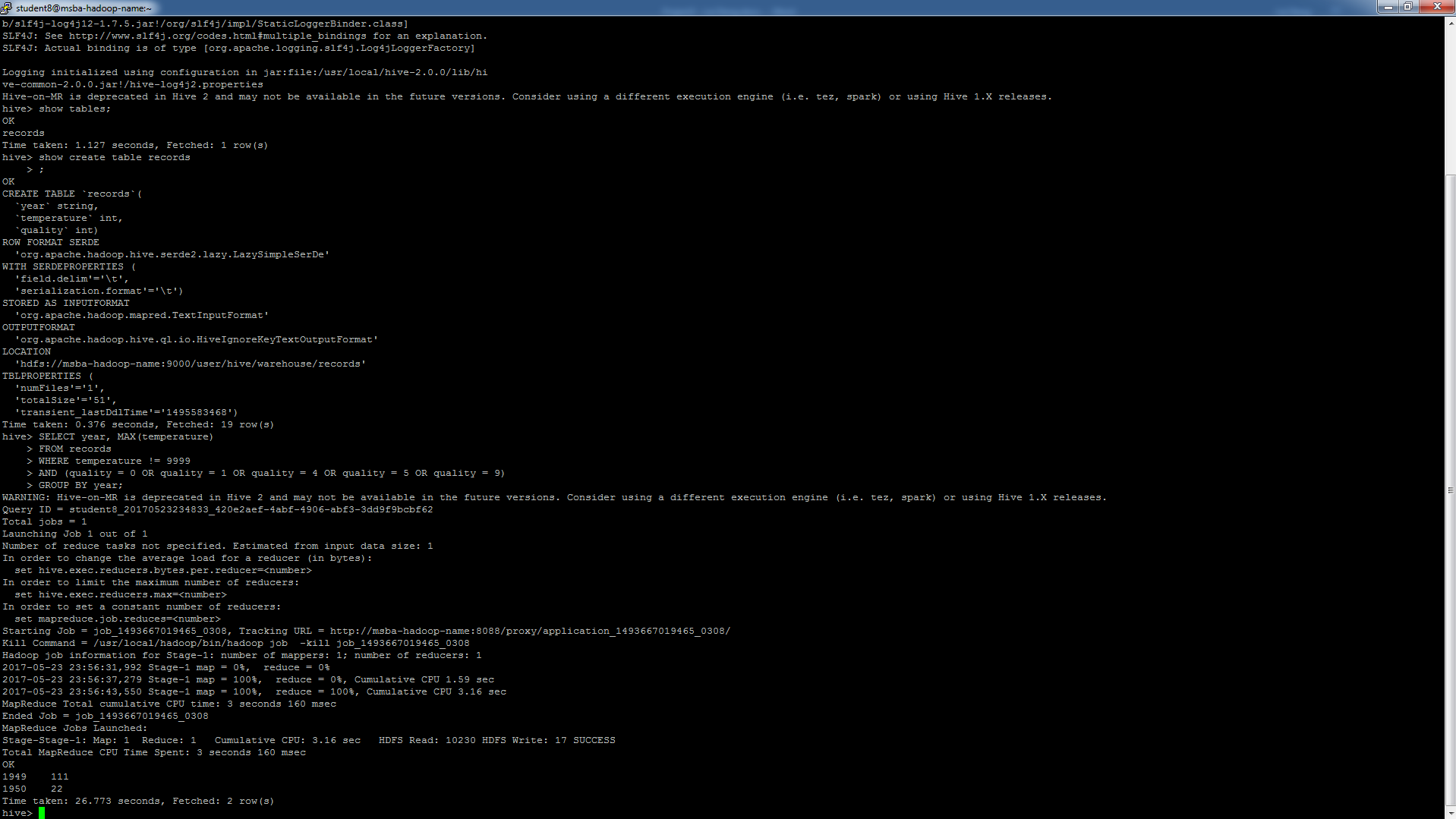
WHERE temperature != 9999

AND (quality = 0 OR quality = 1 OR quality = 4 OR quality = 5 OR quality = 9)

GROUP BY year;

Screenshot of final results in CSUEB:





1. Work the only example on page 444. The codes to create table records2 are as follows:

DROP TABLE IF exists records2;

CREATE TABLE records2 (station STRING, year STRING, temperature INT, quality INT)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t';

LOAD DATA LOCAL INPATH '/your/path/to /sample2.txt'

OVERWRITE INTO TABLE records2;

The sample2.txt can be found in Hadoop-Book-Master/input/ncdc/micro-tab/sample2.txt. Type out all the commands in each step of the process and print out a screenshot of the final results in CSUEB Hadoop.

Commands for running in Hive:

DROP TABLE IF exists records2;

CREATE TABLE records2 (station STRING, year STRING, temperature INT, quality INT)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t';

LOAD DATA LOCAL INPATH '/home/student8/sample2.txt'

OVERWRITE INTO TABLE records2;

FROM records2

SELECT year, temperature

DISTRIBUTE BY year

SORT BY year ASC, temperature DESC;

Screenshot of final results in CSUEB:

