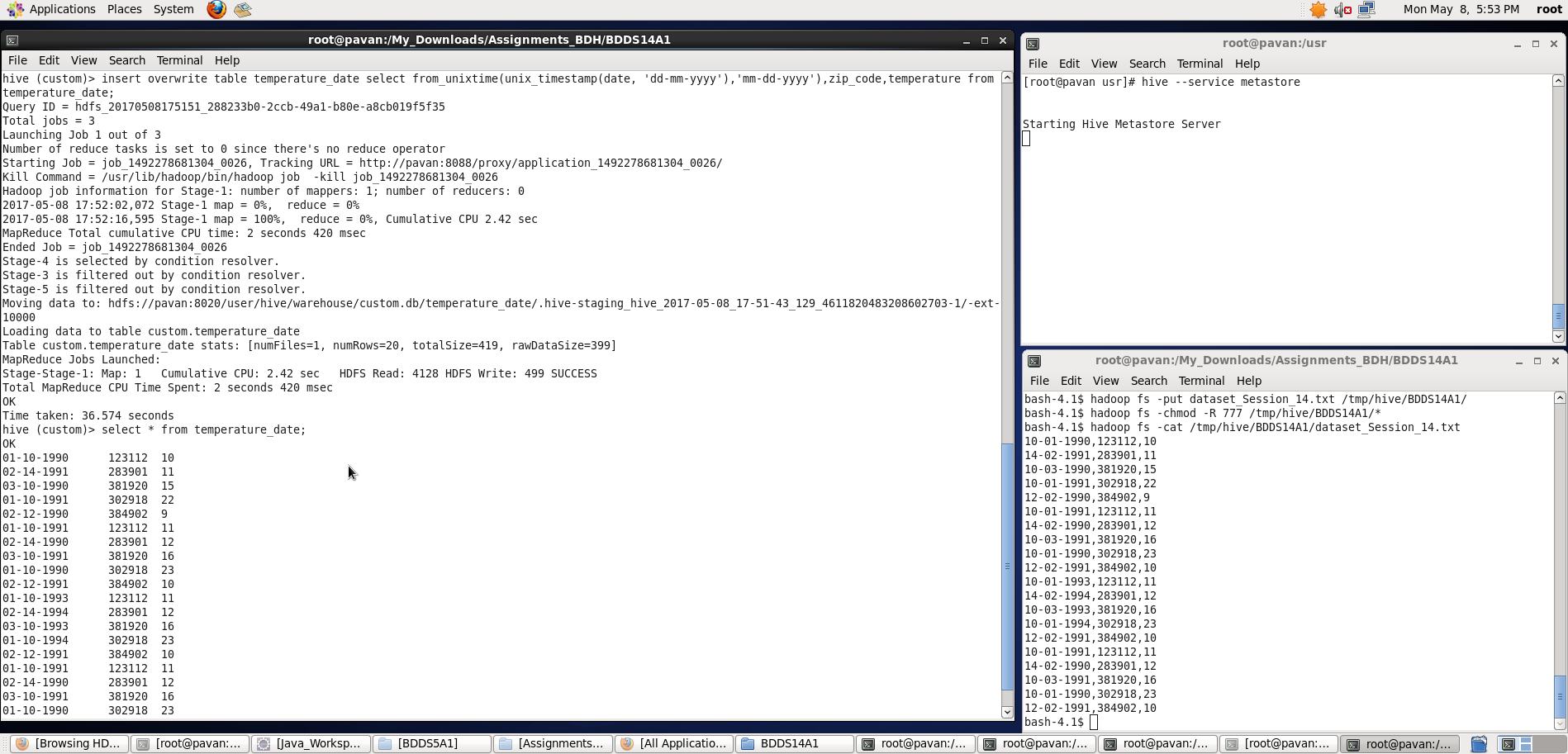
# Big Data Hadoop Training

Session 14 Assignment 2 Solution:

**Dataset**

This was the previous assignment end result – data getting stored in temperature\_date table in custom database as shown below:

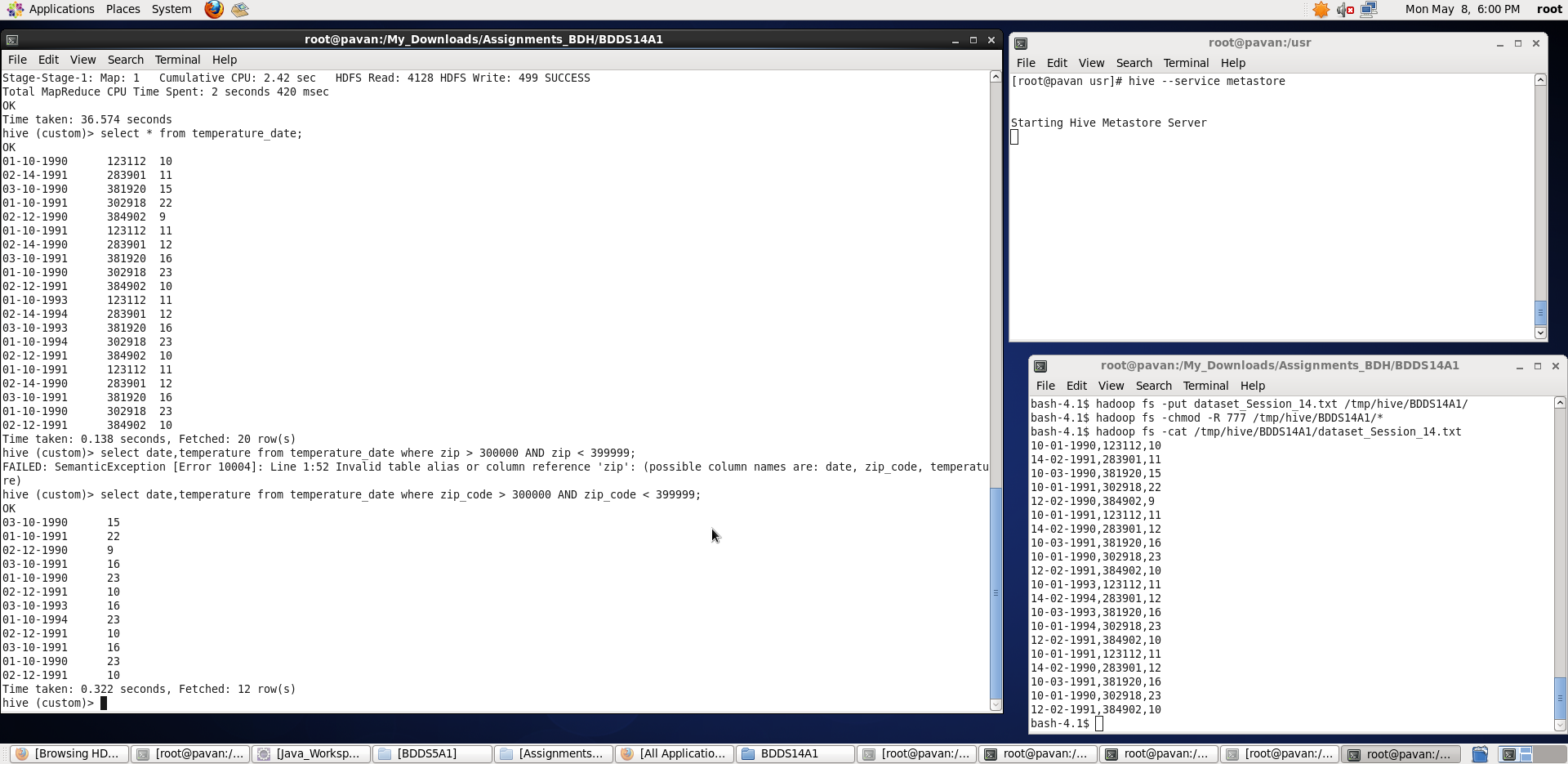


1. Fetch date and temperature from temperature\_data where zip code is greater than 300000 and less than 399999.

A. Now for fetching date and temperature where zip code > 300000 AND zip code < 399999:

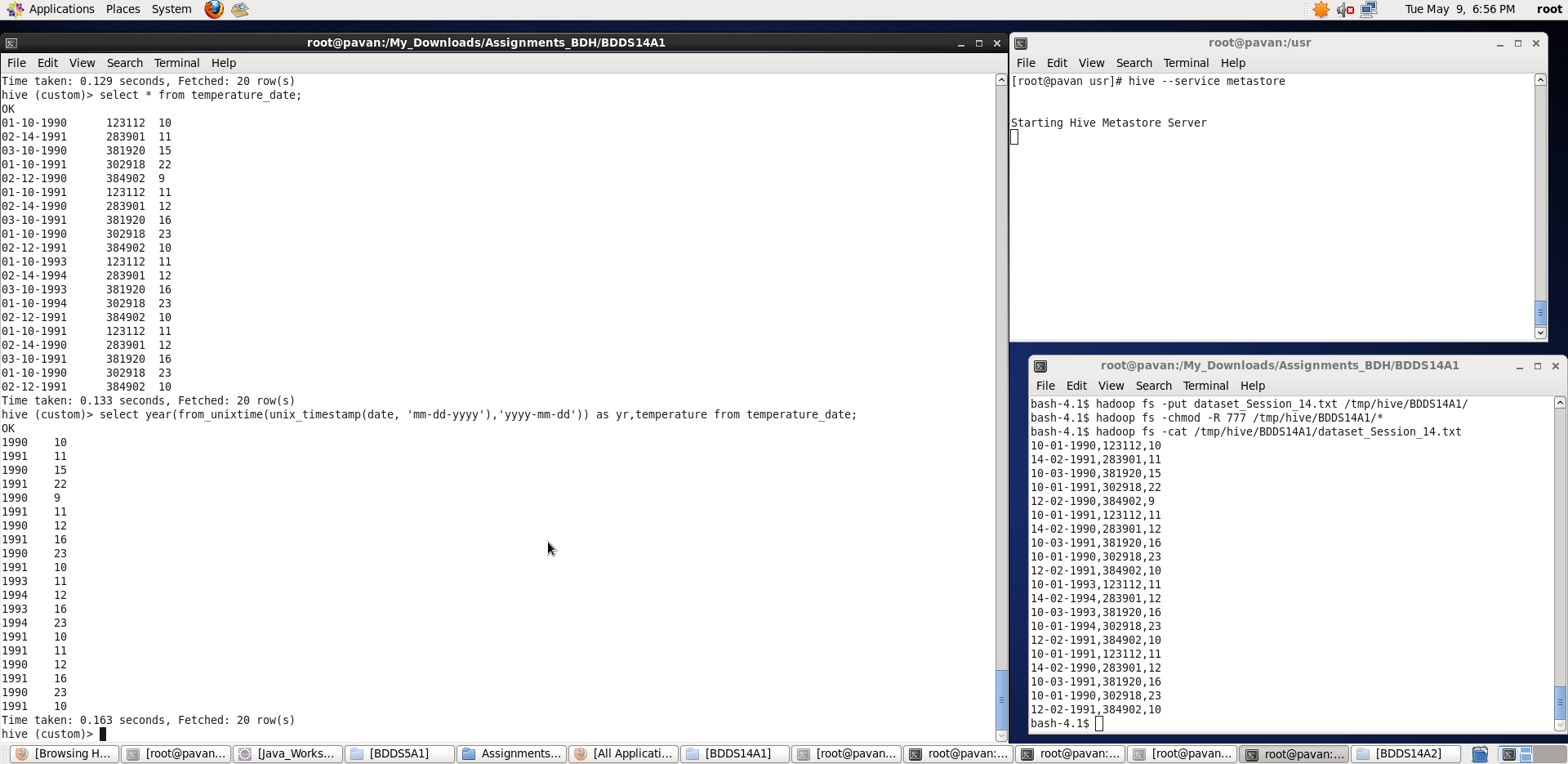
**hive> select date,temperature from temperature\_date where zip\_code > 300000 AND zip\_code < 399999;**

We can see the output in the screenshot:



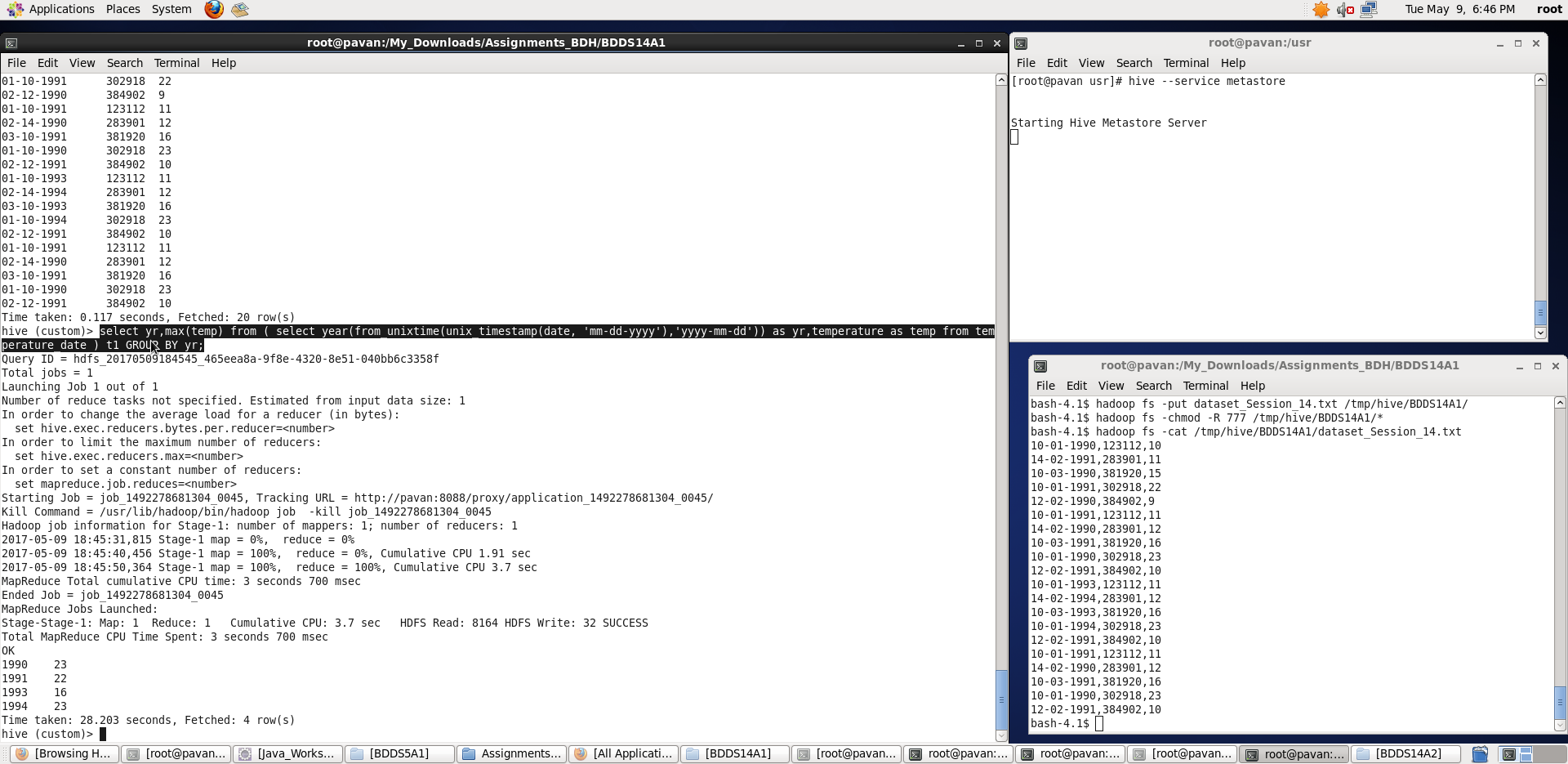
1. Calculate maximum temperature corresponding to every year from temperature\_data table.
2. For this, we need to first extract the year from the “date” field and then group with respect to year and finally, find the max temp corresponding to each year.

**subquery:** **hive>** select year(from\_unixtime(unix\_timestamp(date, 'mm-dd-yyyy'),'yyyy-mm-dd')) as yr,temperature from temperature\_date;



As, we can see, we have extracted the year from date field – now, we will use this as a subquery in our main query to get desired results.

**full cmd**: **hive>** select yr,max(temp) from ( select year(from\_unixtime(unix\_timestamp(date, 'mm-dd-yyyy'),'yyyy-mm-dd')) as yr,temperature as temp from temperature\_date ) t1 GROUP BY yr;

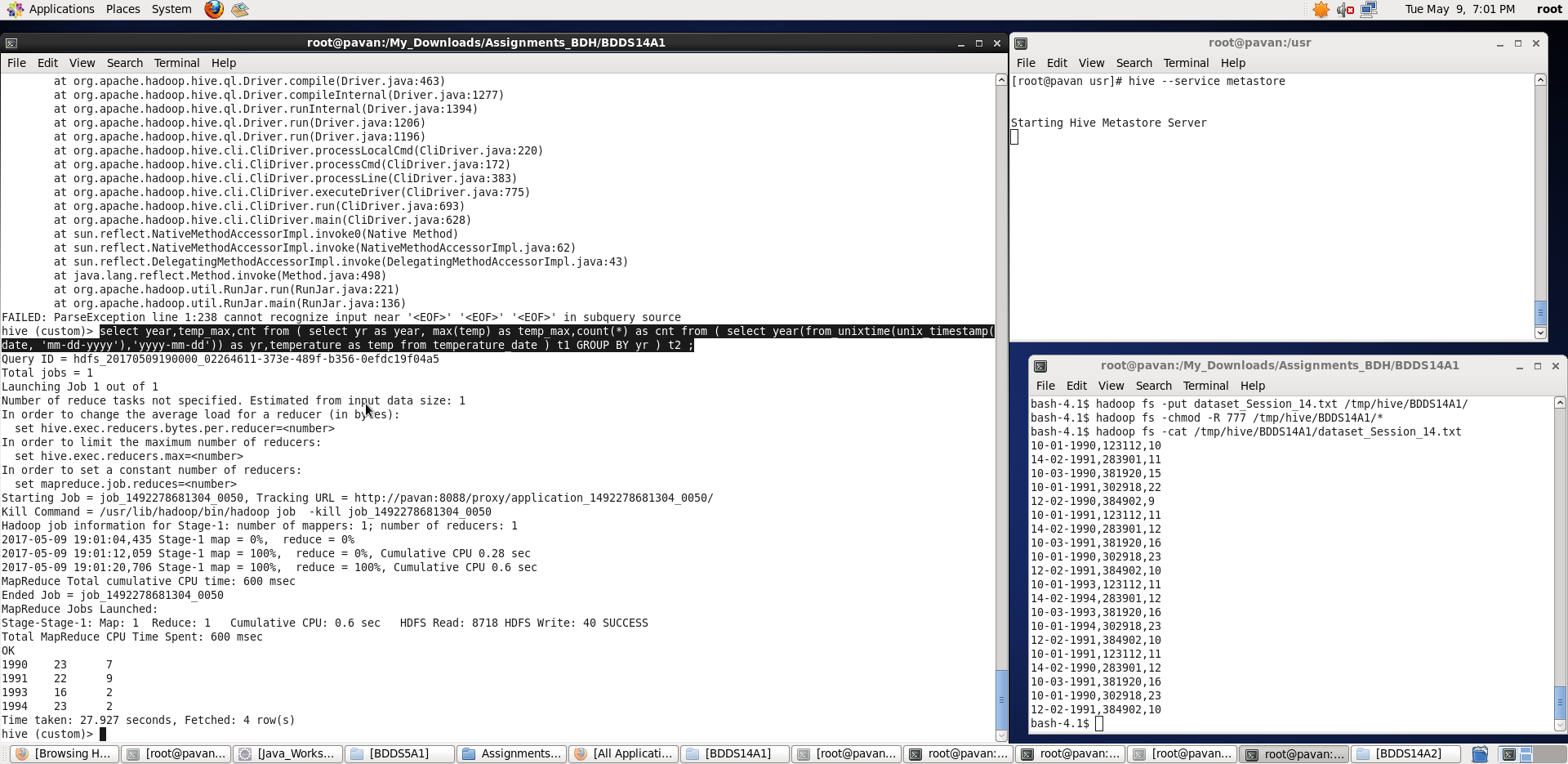


Thus, we found the max temperature corresponding to each year.

1. Calculate maximum temperature from temperature\_data table corresponding to those years which have at least 2 entries in the table.
2. Here, we follow the same steps as in task 2 (above). In addition we will calculate the count (number of times that particular year appeared) and based on that – we will filter out those years whose count is atleast 2.

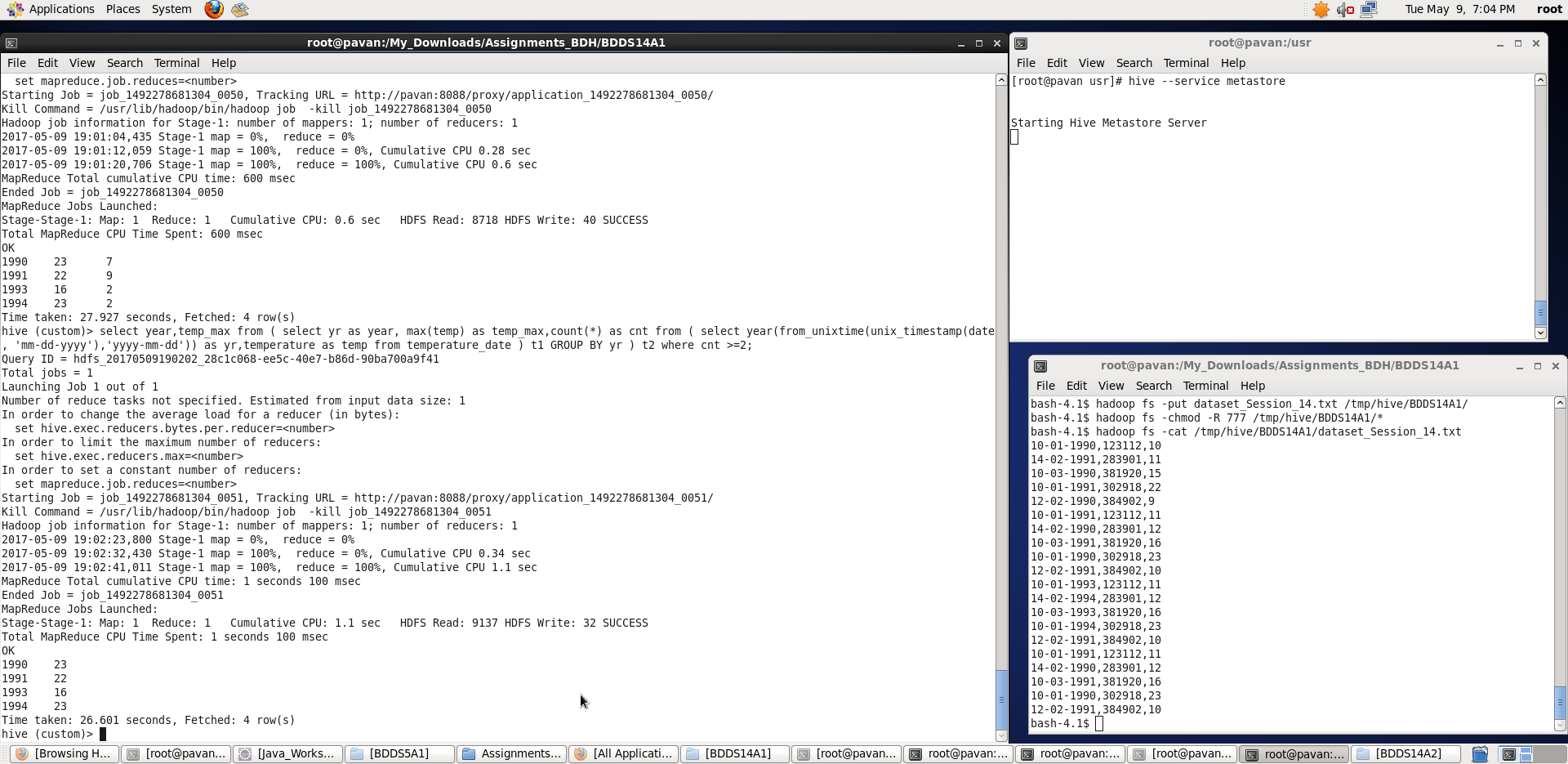
**subquery: hive>** select year(from\_unixtime(unix\_timestamp(date, 'mm-dd-yyyy'),'yyyy-mm-dd')) as yr,temperature from temperature\_date;

**subquery\_2: hive>** select year,temp\_max,cnt from ( select yr as year, max(temp) as temp\_max,count(\*) as cnt from ( select year(from\_unixtime(unix\_timestamp(date, 'mm-dd-yyyy'),'yyyy-mm-dd')) as yr,temperature as temp from temperature\_date ) t1 GROUP BY yr ) t2 ;



Now, we will use these as a subquery in our main query to get desired results

**full cmd**: **hive>** select year,temp\_max from ( select yr as year, max(temp) as temp\_max,count(\*) as cnt from ( select year(from\_unixtime(unix\_timestamp(date, 'mm-dd-yyyy'),'yyyy-mm-dd')) as yr,temperature as temp from temperature\_date ) t1 GROUP BY yr ) t2 where cnt >=2;



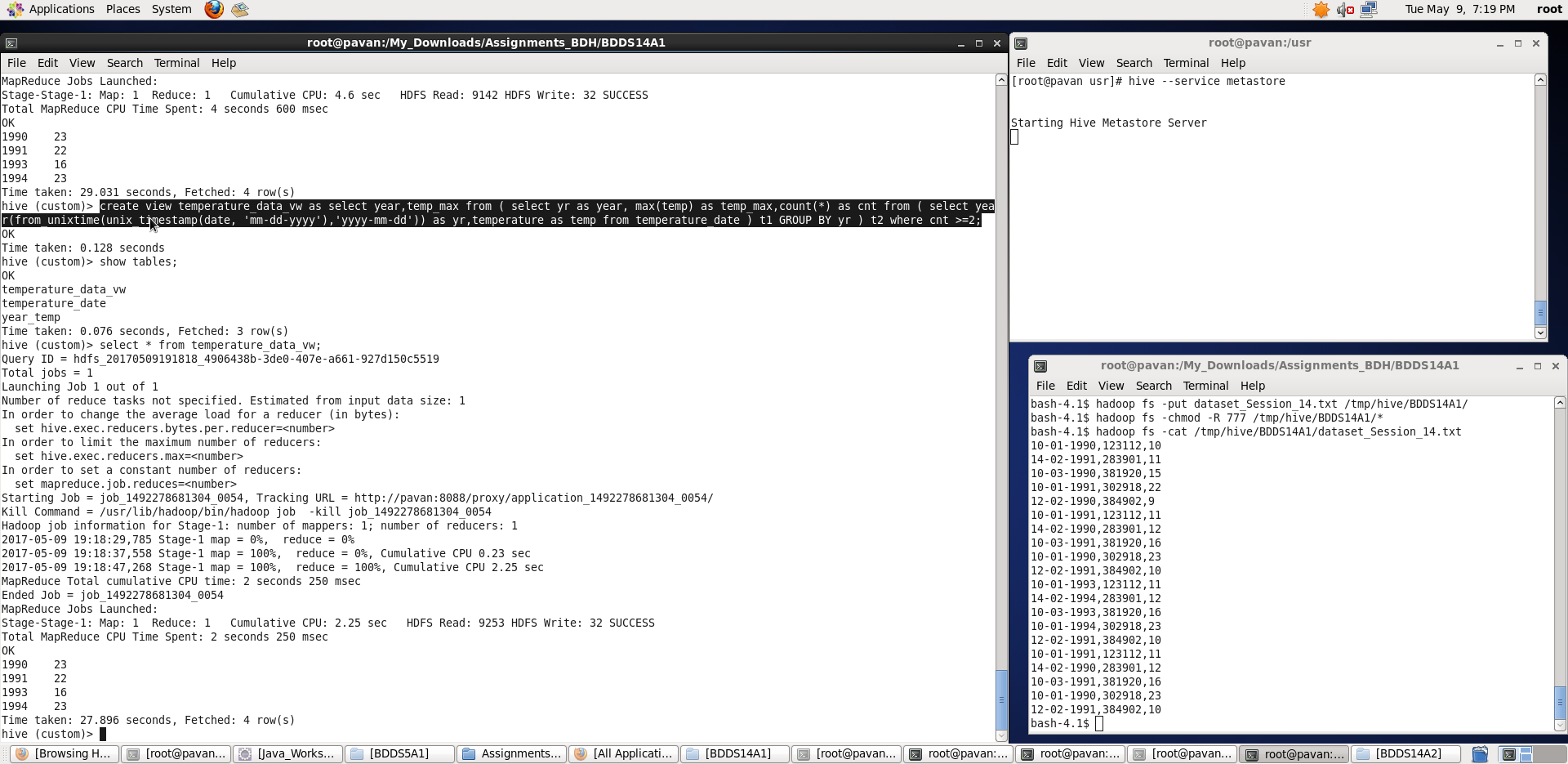
Thus, we found the max temperature corresponding to those years which have atleast two entries.

1. Create a view on the top of last query, name it temperature\_data\_vw.
2. For creating a view, we use ‘create view’ command on top of last query (task 3 full cmd)

**full cmd: hive>** create view temperature\_data\_vw as select year,temp\_max from ( select yr as year, max(temp) as temp\_max,count(\*) as cnt from ( select year(from\_unixtime(unix\_timestamp(date, 'mm-dd-yyyy'),'yyyy-mm-dd')) as yr,temperature as temp from temperature\_date ) t1 GROUP BY yr ) t2 where cnt >=2;

Then show tables;

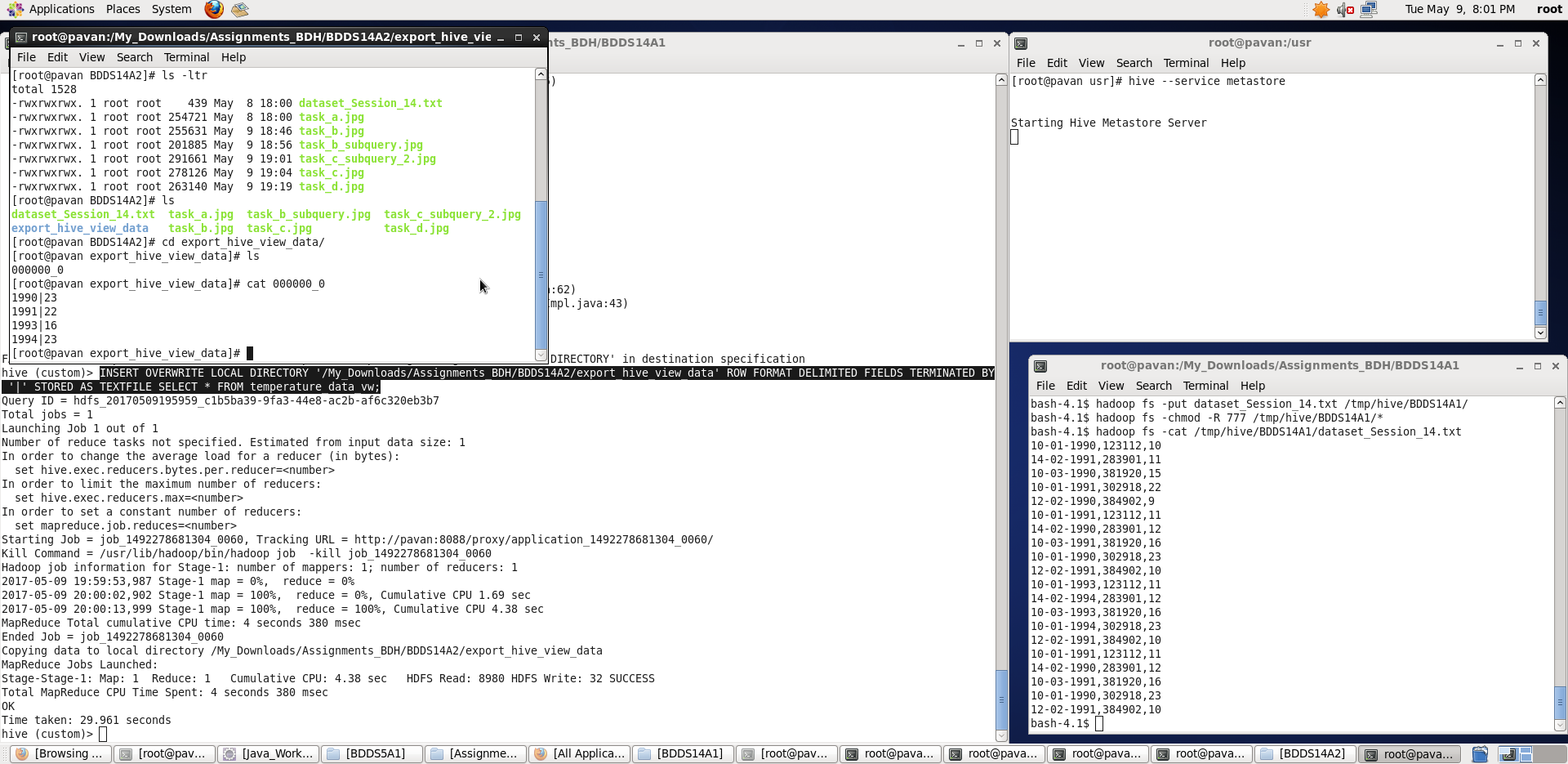
Select \* from temperature\_data\_vw;



In this way, we have created a view on top of last query.

1. Export contents from temperature\_data\_vw to a file in local file system, such that each file is '|' delimited.
2. Exporting contents results in storing of data in a file in a specified location as shown below:

**hive>** INSERT OVERWRITE LOCAL DIRECTORY ‘/tmp/export\_hive\_view\_data' ROW FORMAT DELIMITED FIELDS TERMINATED BY '|' STORED AS TEXTFILE SELECT \* FROM temperature\_data\_vw;



We can see that the contents in the view got stored in a file in local system. Using ‘cat’ command we see the contents – delimited by ‘|’.

Thus, we have performed required tasks on the table temperature\_date which we have created in our previous assignment (session 14 assignment 1).