DSCI5350 Assignment-2 Satyasriharsha Buddha

Deliverables & requirements:  
You must provide screenshots of your solutions. All the tables that you create should have your  
EUID as suffix. And set the terminal title to your name. The submission should be a word  
document.

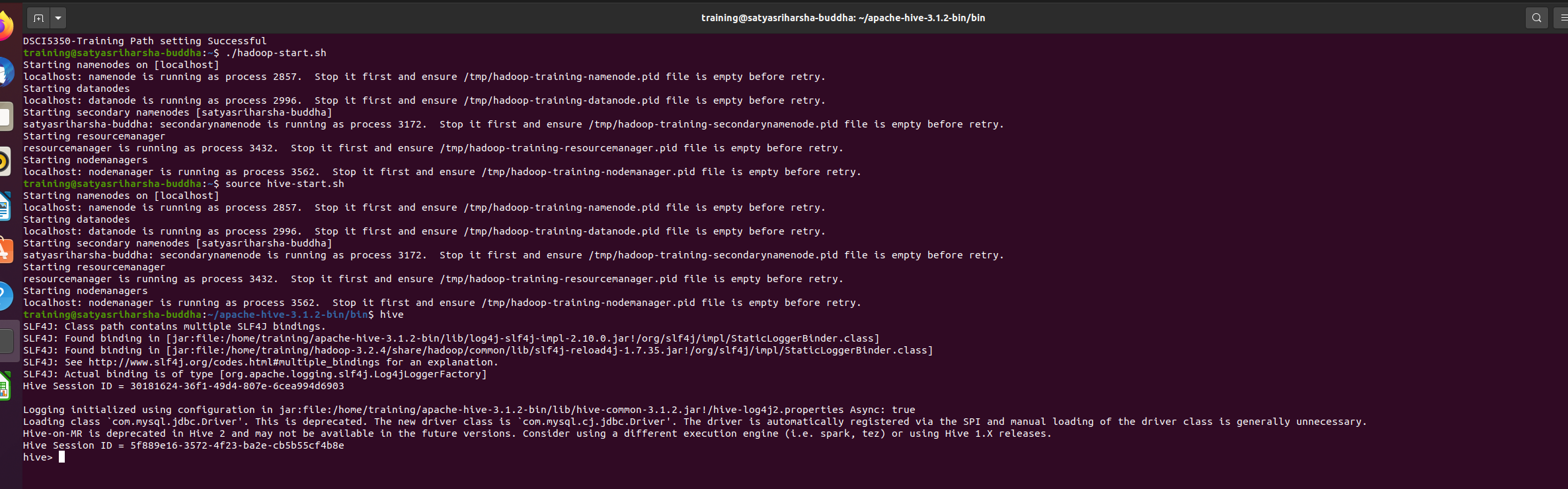
-> Terminal title to my name is set already.

-> The tables I created for assignment are suffixed with my EUID

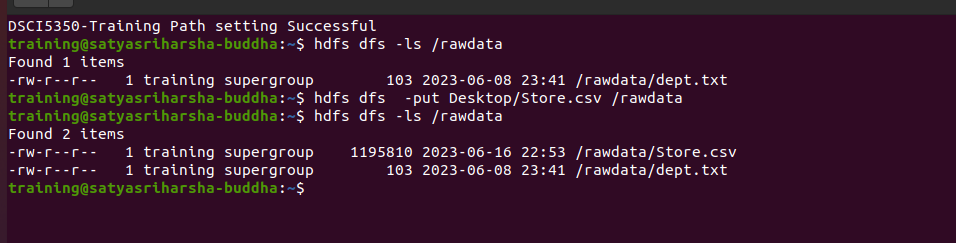
Question-1

Correctly process the given data and store it in the hive display top 5 rows of the data

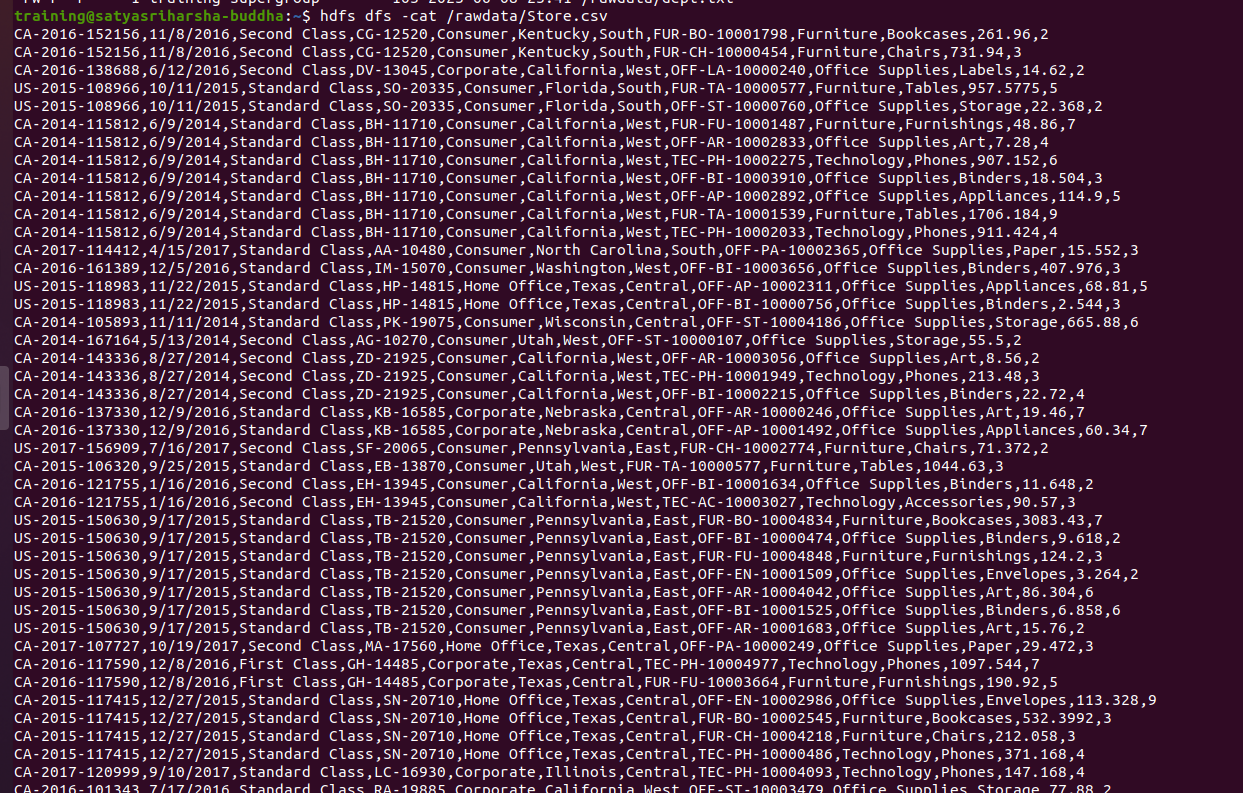
* The Store.csv file was imported to Desktop from Canvas
* The HDFS and hive shells were initiated



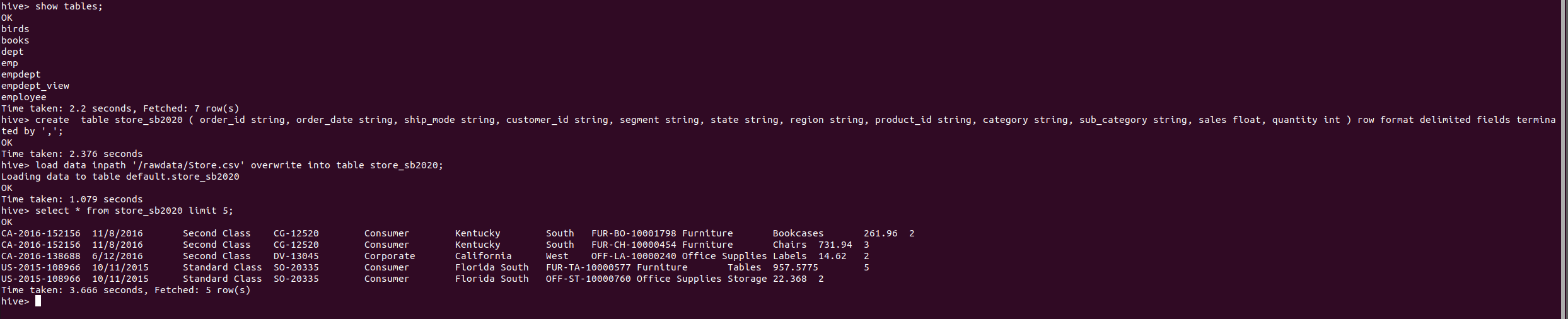
* Loading Store.Csv into hdfs using –put command



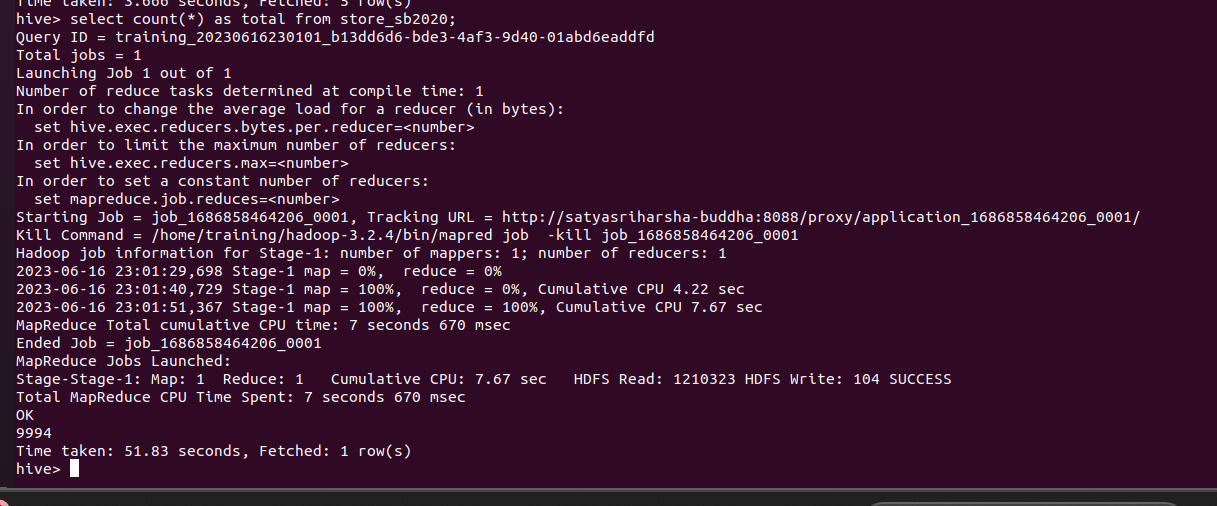
* Viewing the data in the store.csv file using –cat command



* Created a table in hive store\_sb2020 and get top 5 records



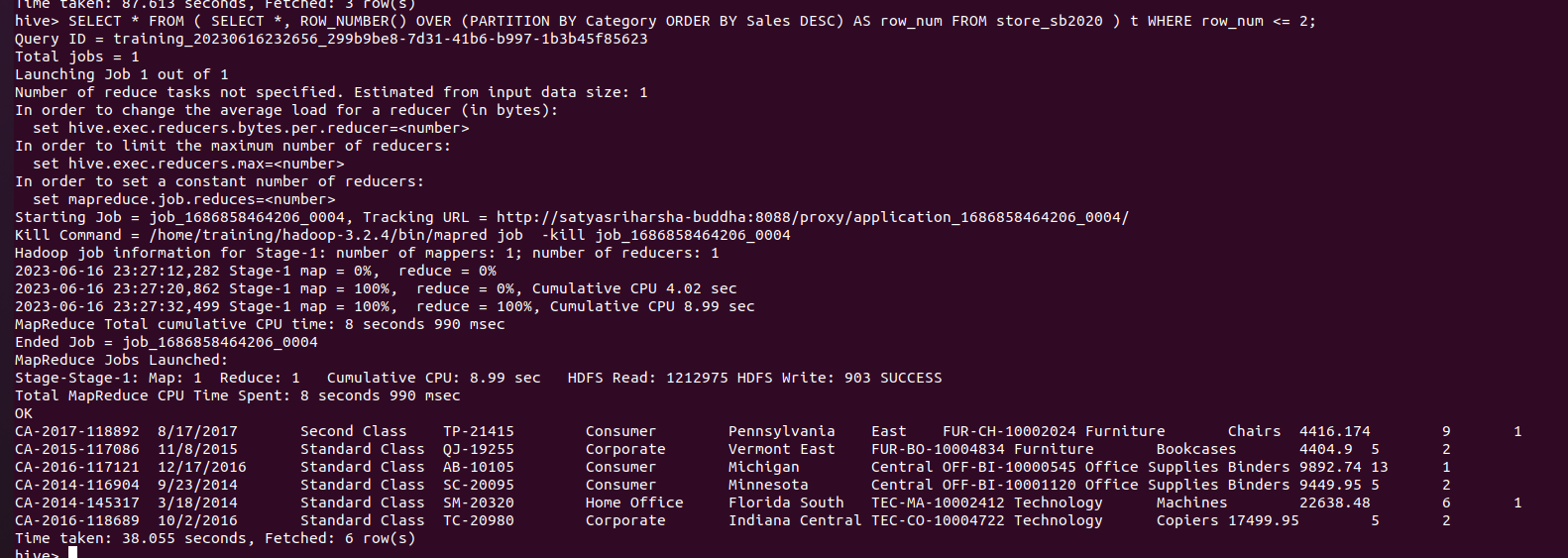
* Using the select count(\*) command to get see if complete data is retrieved



As 9994 rows were retrieved, we can say that complete data was loaded.

Question 2: Display details of highest Sales in each Category type (select top 2 only)

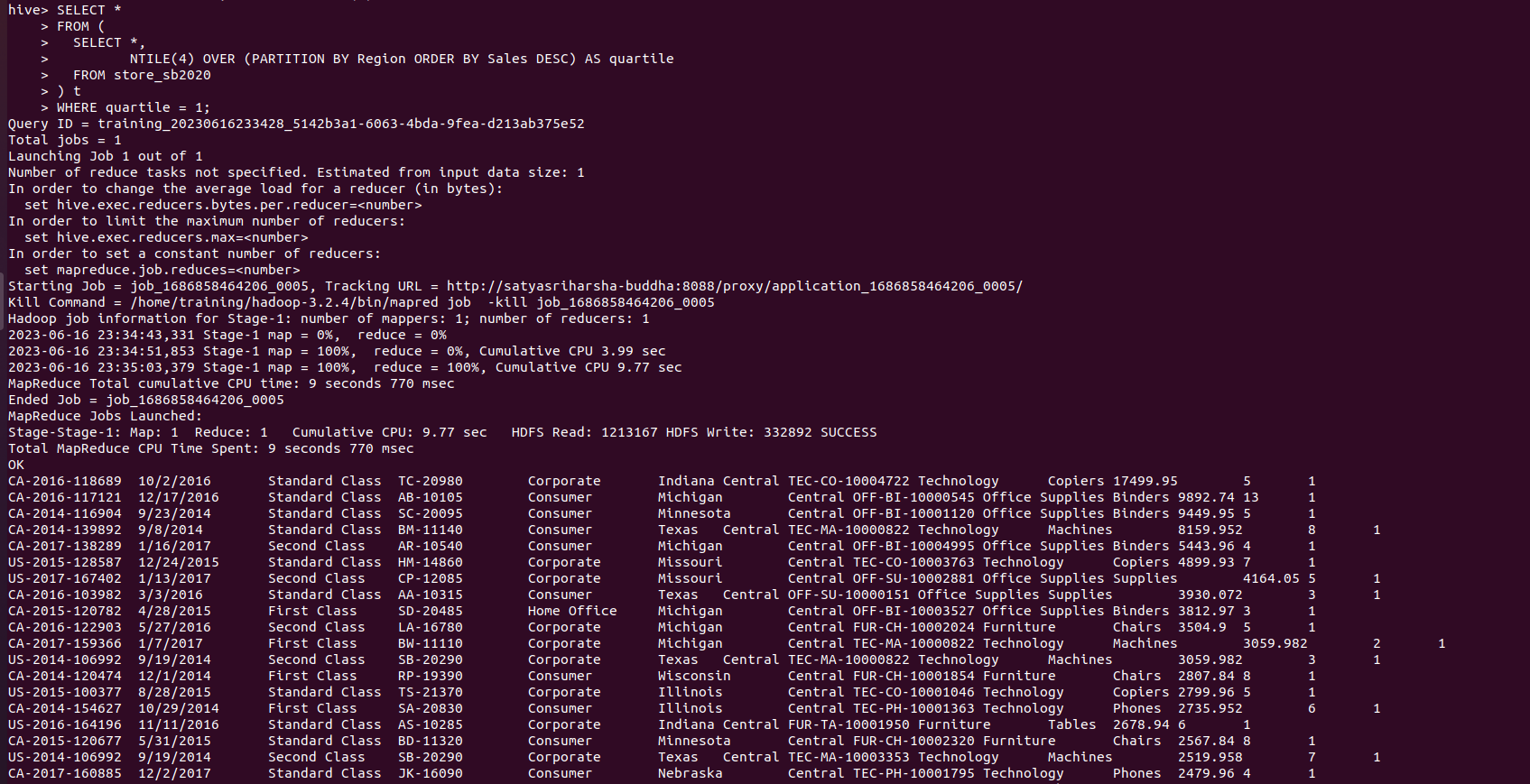
* SELECT \* FROM ( SELECT \*, ROW\_NUMBER() OVER (PARTITION BY Category ORDER BY Sales DESC) AS row\_num FROM store\_sb2020 ) t WHERE row\_num <= 2;



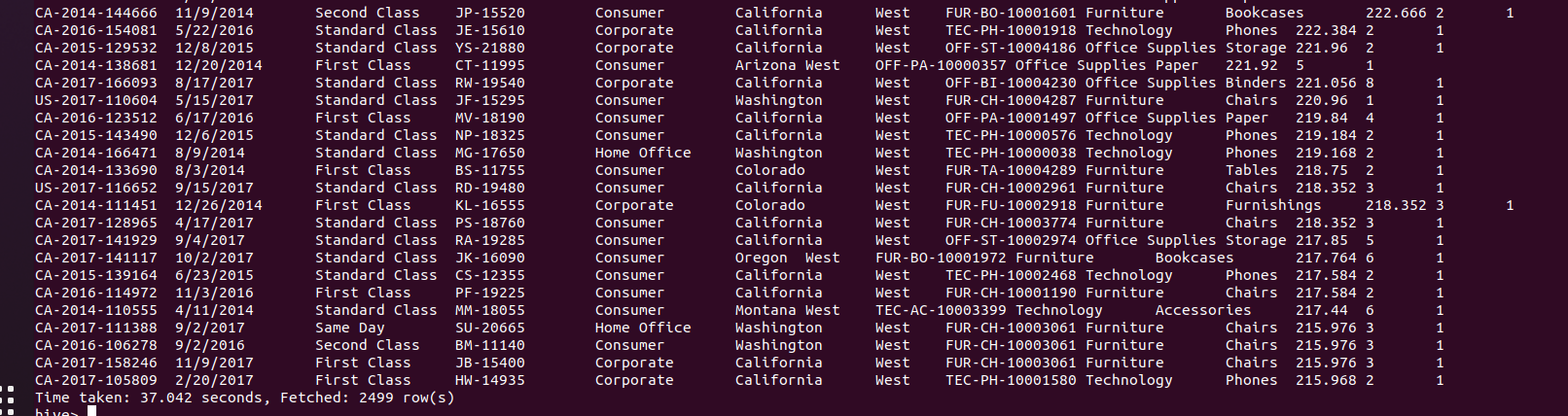
* We order by sales in descending order, and We use the ROW\_NUMBER() function with the PARTITION BY clause to assign a row number to each row within its respective category, and then filters the result to only include rows with a row number less than or equal to 2.

Question 3: Display details of top 25% of sales for each region. (Must use ntile() function)

* select \* from ( select \*, ntile(4) over (partition by region order by sales desc) as quartile from store\_sb2020 ) t where quartile = 1;



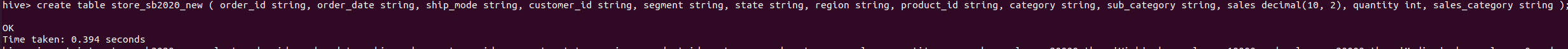
* We use the NTILE(4) function to assign a quartile number to each row within its respective region, with quartile 1 representing the top 25% of sales.



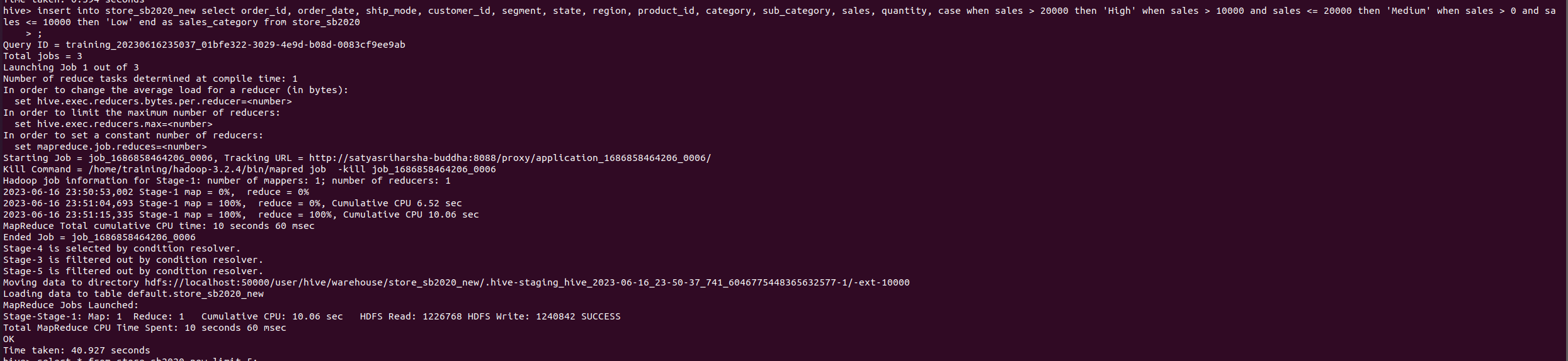
As we can see we retrieved 2499 Rows.

Question 4. Create a new Column Sales\_Category to the table, based on Sales column as shown  
below. Calculate number of unique Products under each Sales\_Category.  
Sales above 20000 High  
Sales above 10000 and below 20000 Medium  
Sales above 0 and below 10000 Low

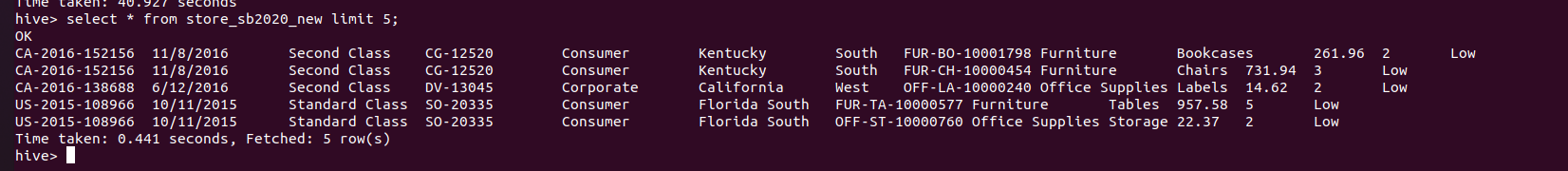
* Created a new table named store\_sb2020\_new with new column Sales\_catetgory



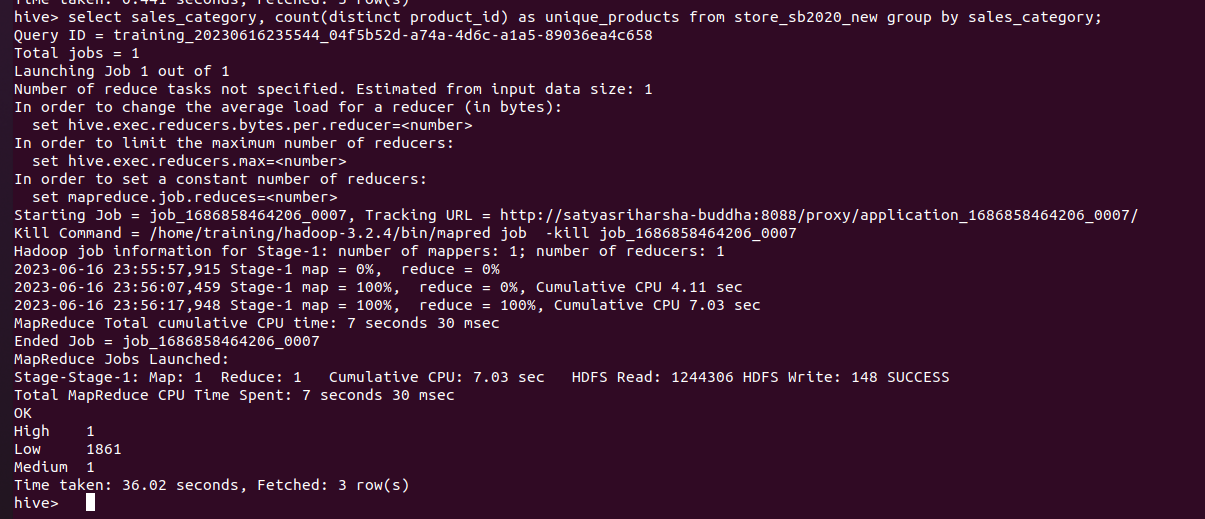
* insert into store\_sb2020\_new select order\_id, order\_date, ship\_mode, customer\_id, segment, state, region, product\_id, category, sub\_category, sales, quantity, case when sales > 20000 then 'High' when sales > 10000 and sales <= 20000 then 'Medium' when sales > 0 and sales <= 10000 then 'Low' end as sales\_category from store\_sb2020



* As you can see we have new column and data
* Select \* from store\_sb2020\_new limit 5;

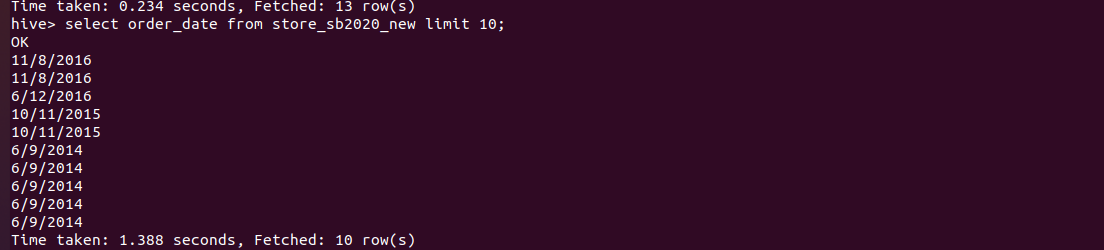


* SELECT Sales\_Category, COUNT(DISTINCT Product\_ID) AS Unique\_Products FROM store\_sb2020\_new GROUP BY Sales\_Category;



Question 5. Display Percent change in average sales across every year.

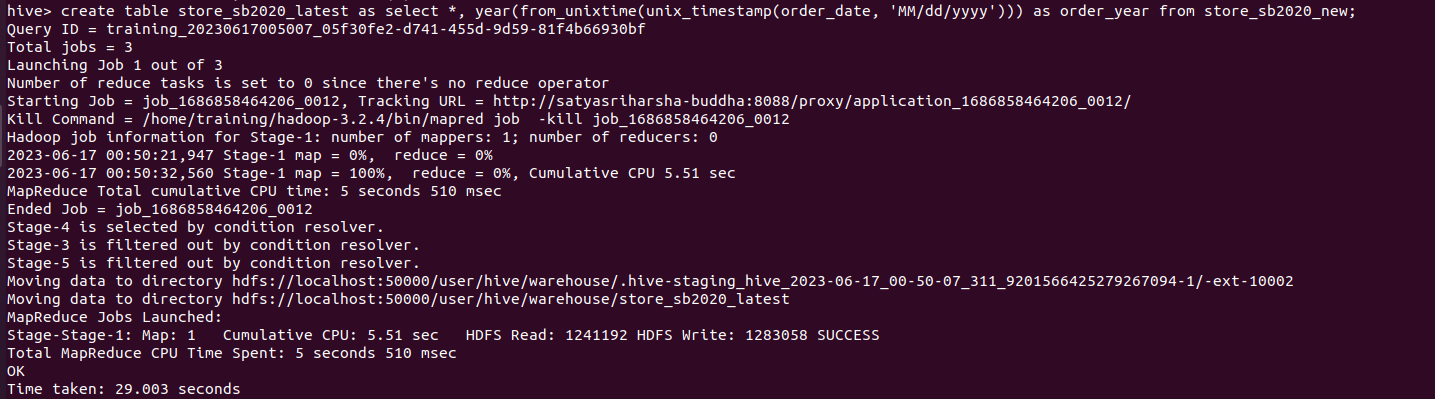
* The Current date format in order\_date column is mm/dd/yyyy

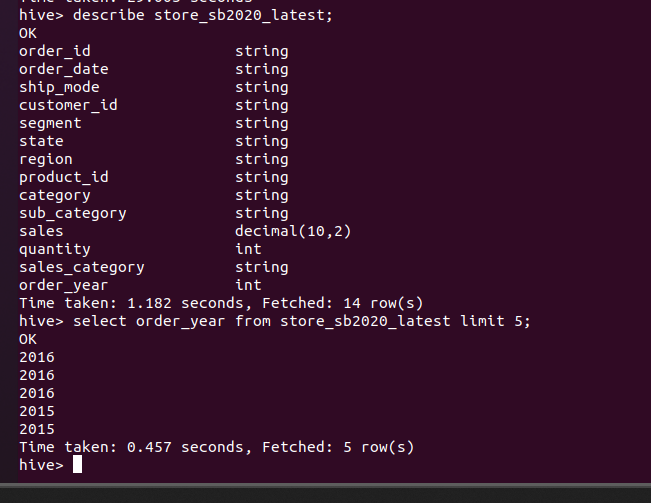


* Creating a new table store\_sb2020\_latest with new column Order\_year for which year is extracted

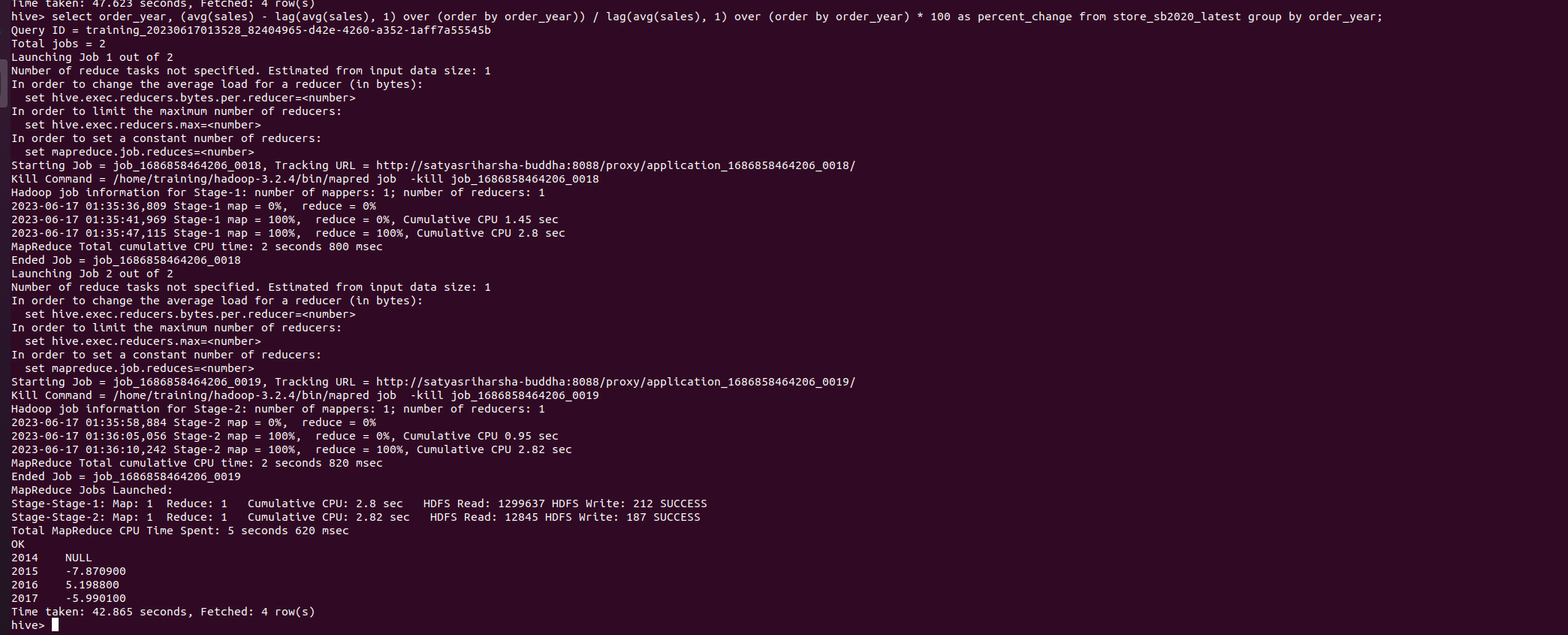
Create table store\_sb2020\_latest as select \*,year(fromunixtime(unix\_timestamp(order\_date,’MM/dd/yyyy’))) as order\_year from store\_sb2020\_new;

* We are extrating year from order\_date into a new column order\_year





* As you can see the order\_year is correctly loaded as per the top rows
* SELECT order\_year, AVG(sales) AS average\_sales, (AVG(sales) - LAG(AVG(sales), 1) OVER (ORDER BY order\_year)) / LAG(AVG(sales), 1) OVER (ORDER BY order\_year) \* 100 AS percent\_change FROM store\_sb2020\_latest GROUP BY order\_year;



* The AVG function calculates the average sales for each year, and the LAG window function retrieves the previous year's average sales.
* The percent change is then calculated by subtracting the previous year's average sales from the current year's average sales, dividing it by the previous year's average sales, and multiplying by 100

References: https://cwiki.apache.org/confluence/display/hive/languagemanual+windowingandanalytics