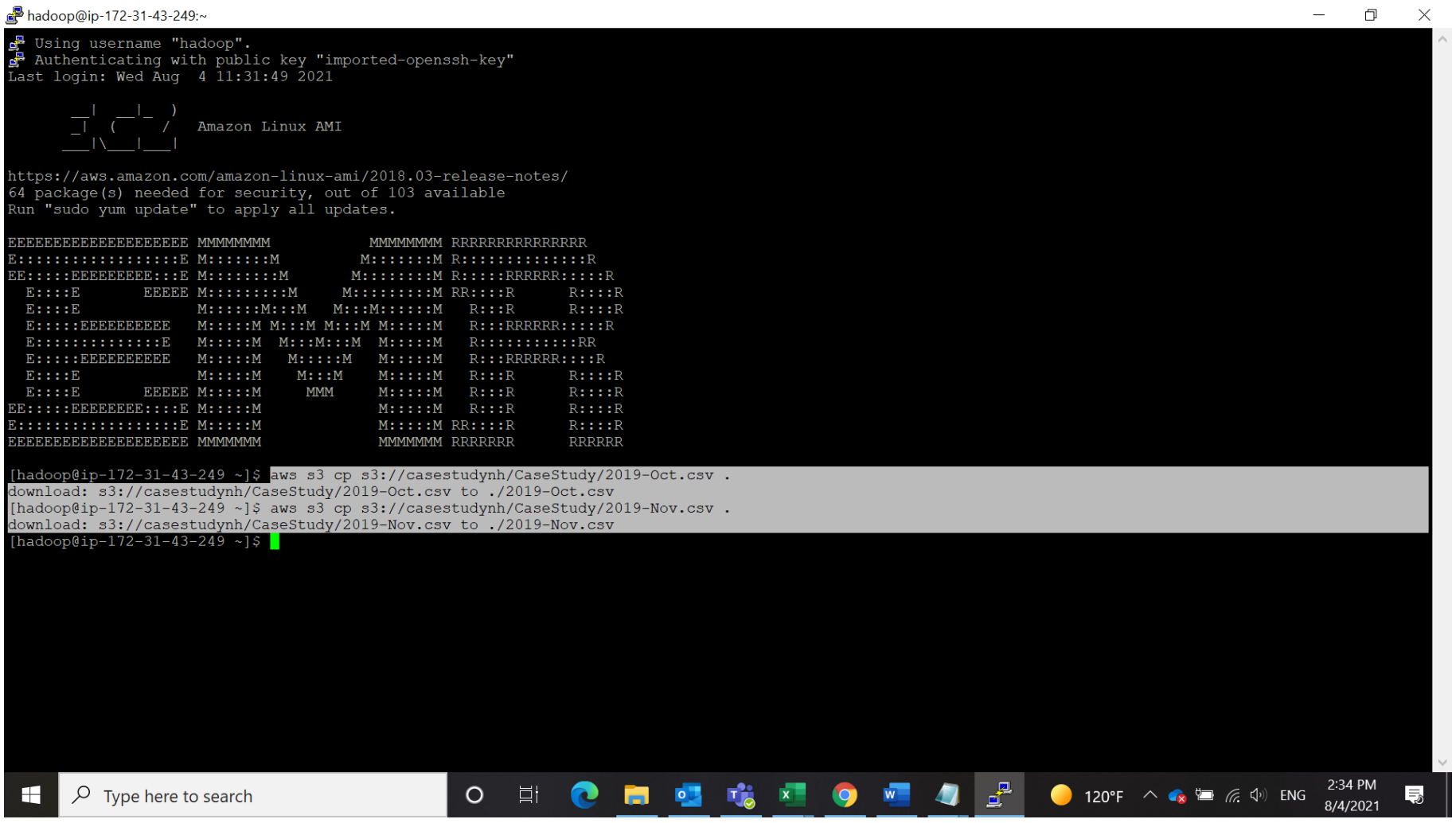
Hive Case Study

Nikshita S. Shetty & Hammad M. Siddiqui

1. We Start by Loading datasets from S3 bucket into Hadoop after connecting successfully.
   aws s3 cp s3://casestudynh/CaseStudy/2019-Oct.csv .
   aws s3 cp s3://casestudynh/CaseStudy/2019-Nov.csv .



2. Logging into Hive

hive

hadoop@ip-172-31-39-181:~

```
EEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE::::::EEEEEEEEE:::E M::::::::M       M::::::::M R::::::RRRRRR:::::R
  E:::::E       EEEEE M:::::::::M     M:::::::::M RR:::::R     R:::::R
  E:::::E             M::::::M:::M   M:::M::::::M   R::::R     R:::::R
  E::::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R::::RRRRRR:::::R
  E::::::::::::::E    M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E::::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R::::RRRRRR::::R
  E:::::E             M:::::M    M:::M    M:::::M   R::::R     R:::::R
  E:::::E       EEEEE M:::::M     MMM     M:::::M   R::::R     R:::::R
EE::::::EEEEEEEE:::E M:::::M             M:::::M   R::::R     R:::::R
E::::::::::::::::::E M:::::M             M:::::M RR:::::R     R:::::R
EEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR     RRRRRRR

[hadoop@ip-172-31-39-181 ~]$ aws s3 cps3://casestudynh/CaseStudy/2019-Nov.csv .
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:

  aws help
  aws <command> help
  aws <command> <subcommand> help
aws: error: argument subcommand: Invalid choice, valid choices are:

ls                                       | website

cp                                       | mv

rm                                       | sync

mb                                       | rb

presign
[hadoop@ip-172-31-39-181 ~]$ aws s3 cp s3://casestudynh/CaseStudy/2019-Nov.csv .

download: s3://casestudynh/CaseStudy/2019-Nov.csv to ./2019-Nov.csv
[hadoop@ip-172-31-39-181 ~]$ aws s3 cp s3://casestudynh/CaseStudy/2019-Oct.csv .
download: s3://casestudynh/CaseStudy/2019-Oct.csv to ./2019-Oct.csv
[hadoop@ip-172-31-39-181 ~]$ show databases;
-bash: show: command not found
[hadoop@ip-172-31-39-181 ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
hive>
```

Type here to search 99°F ENG 9:16 PM 8/3/2021

3. Creating a database

Create database if not exists cosmetics;

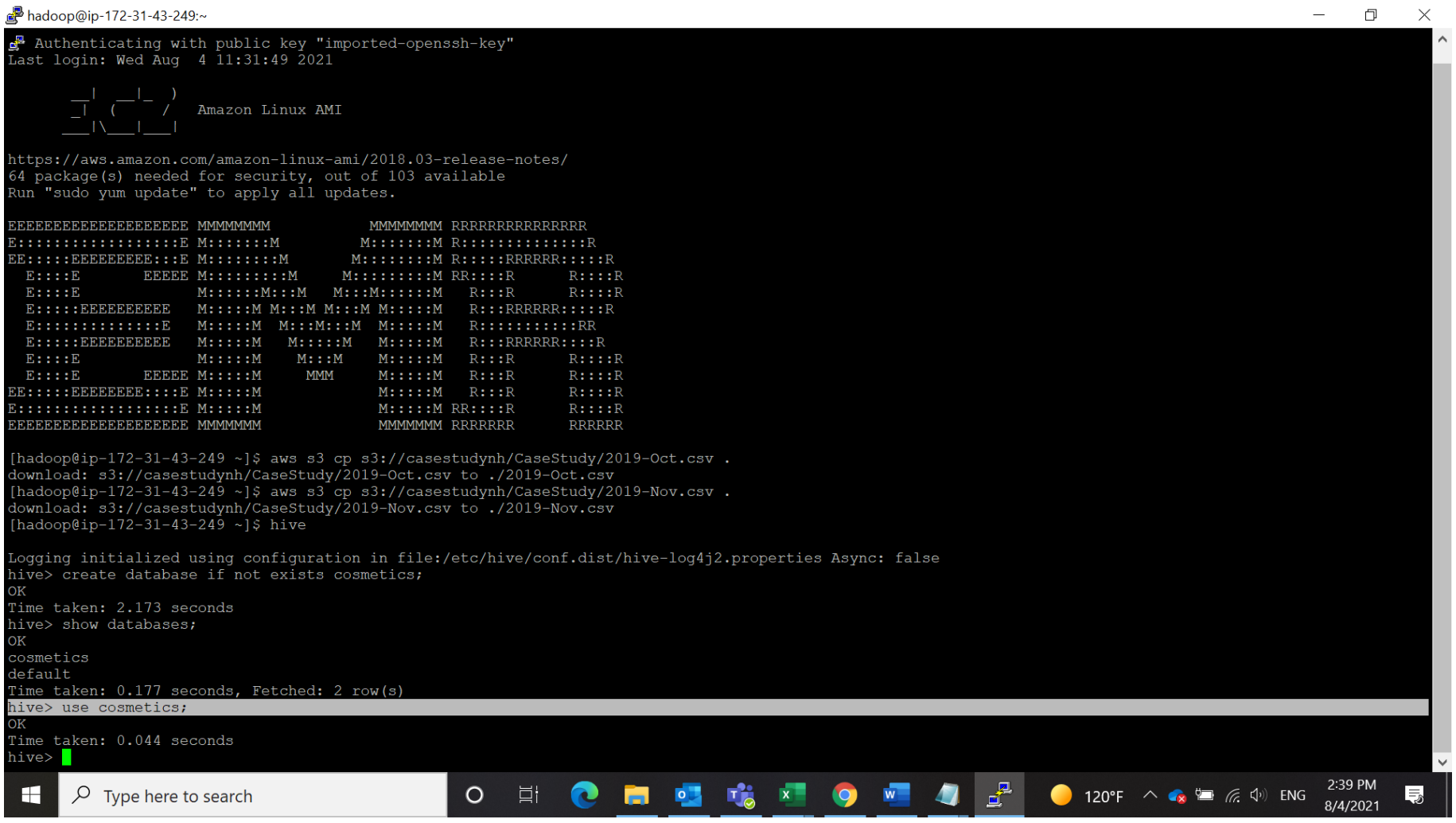
hadoop@ip-172-31-39-181:~

```
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E:::::E            M:::::M    M:::M    M:::::M   R:::R      R:::R
  E:::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R:::R
EE::::::EEEEEEEE:::E M:::::M             M:::::M   R:::R      R:::R
E::::::::::::::::::E M:::::M             M:::::M RR:::R      R:::R
EEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-39-181 ~]$ aws s3 cps3://casestudynh/CaseStudy/2019-Nov.csv .
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:

  aws help
  aws <command> help
  aws <command> <subcommand> help
aws: error: argument subcommand: Invalid choice, valid choices are:

ls                                       | website

cp                                       | mv

rm                                       | sync

mb                                       | rb

presign
[hadoop@ip-172-31-39-181 ~]$ aws s3 cp s3://casestudynh/CaseStudy/2019-Nov.csv .

download: s3://casestudynh/CaseStudy/2019-Nov.csv to ./2019-Nov.csv
[hadoop@ip-172-31-39-181 ~]$ aws s3 cp s3://casestudynh/CaseStudy/2019-Oct.csv .
download: s3://casestudynh/CaseStudy/2019-Oct.csv to ./2019-Oct.csv
[hadoop@ip-172-31-39-181 ~]$ show databases;
-bash: show: command not found
[hadoop@ip-172-31-39-181 ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
hive> show databases;
OK
default
Time taken: 0.649 seconds, Fetched: 1 row(s)
hive> create database if not exists cosmetics;
OK
Time taken: 0.321 seconds
hive>
```

Type here to search 99°F ENG 9:17 PM 8/3/2021

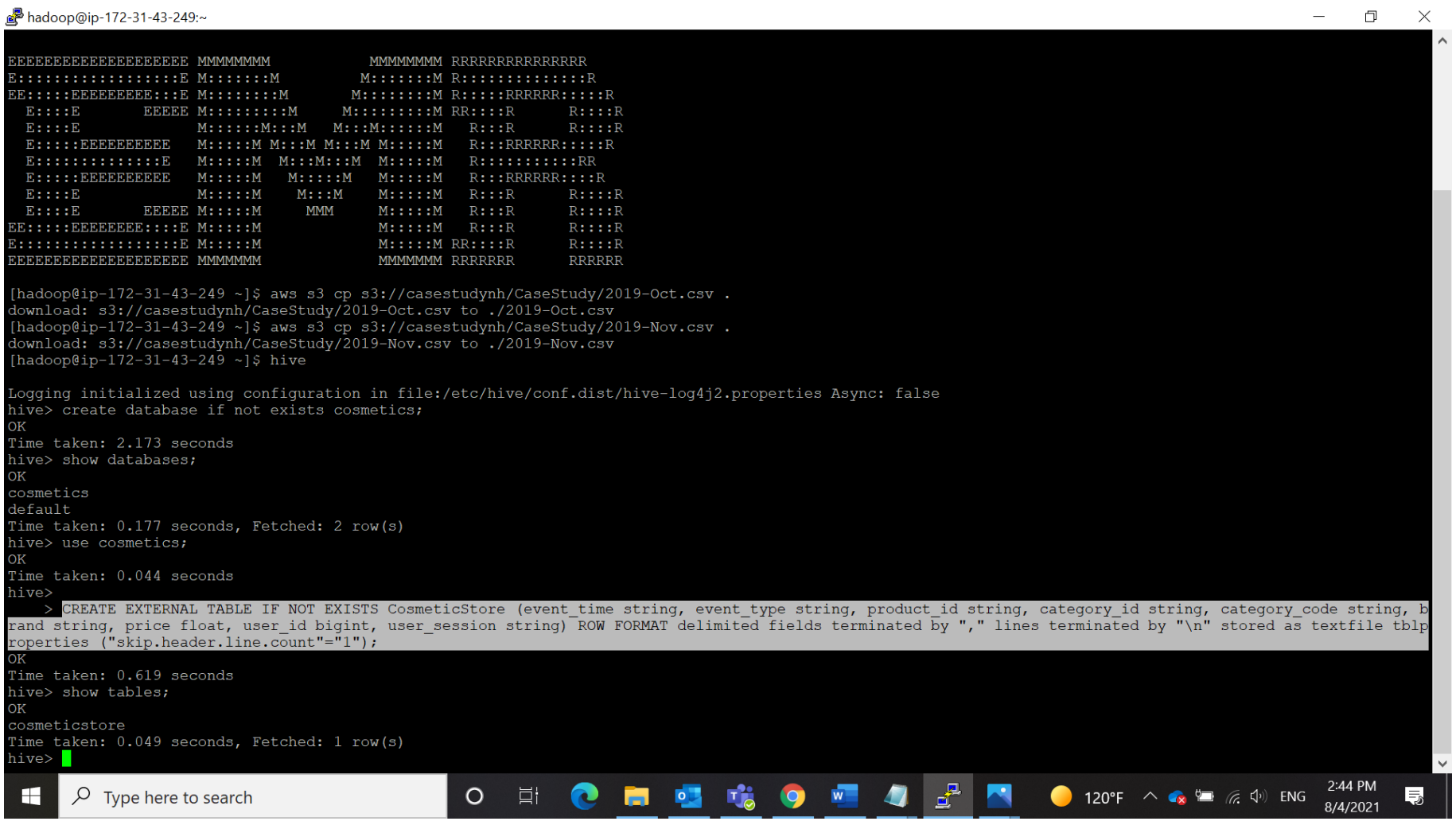
4. Use database

Use cosmetics;



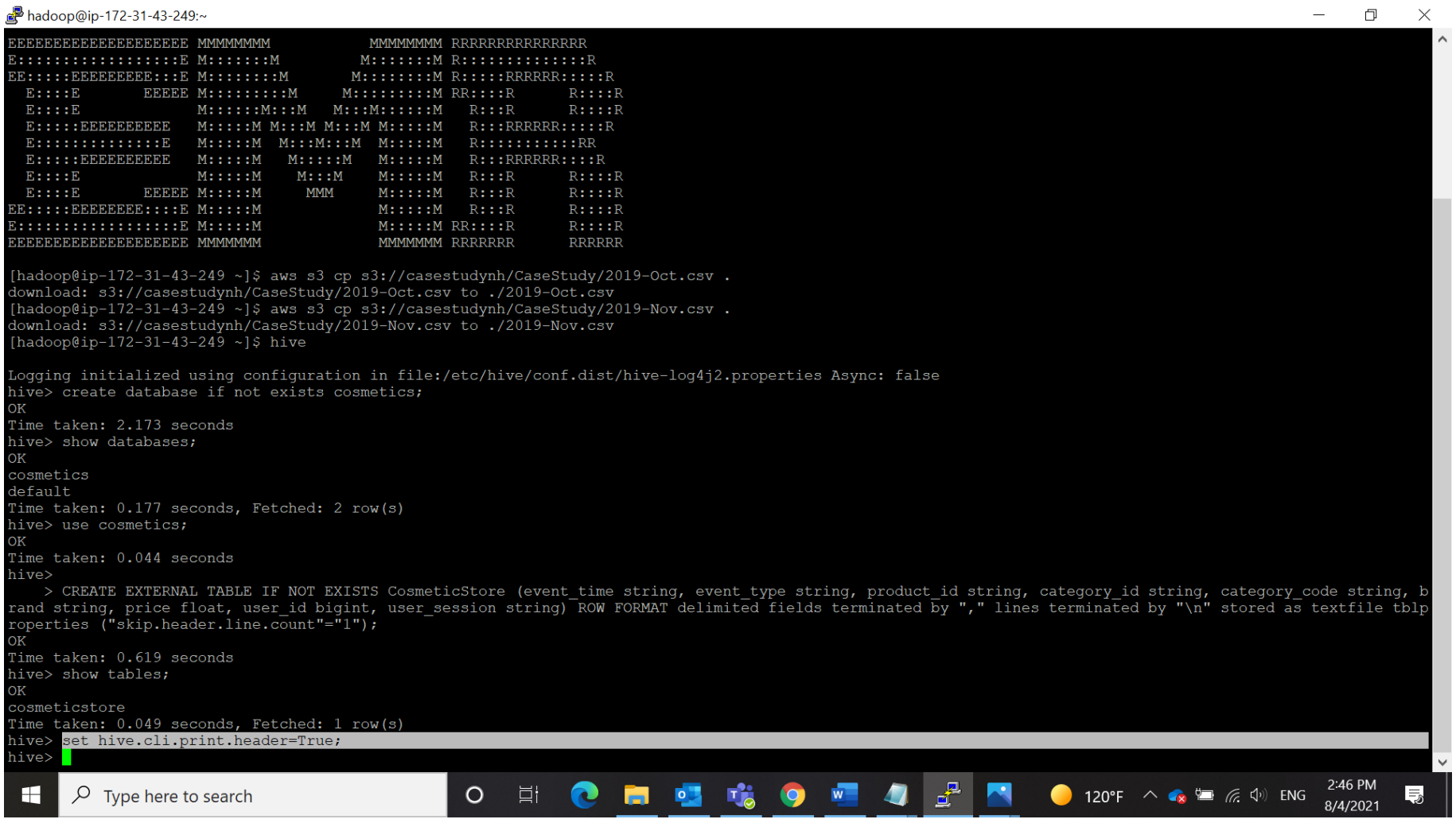
5. Creating An External table named Cosmetic Store

CREATE EXTERNAL TABLE IF NOT EXISTS CosmeticStore (event_time string, event_type string, product_id string, category_id string, category_code string, brand string, price float, user_id bigint, user_session string) ROW FORMAT delimited fields terminated by "," lines terminated by "\n" stored as textfile tblproperties ("skip.header.line.count"="1");



6. Enabling heading in the output

set hive.cli.print.header=True;

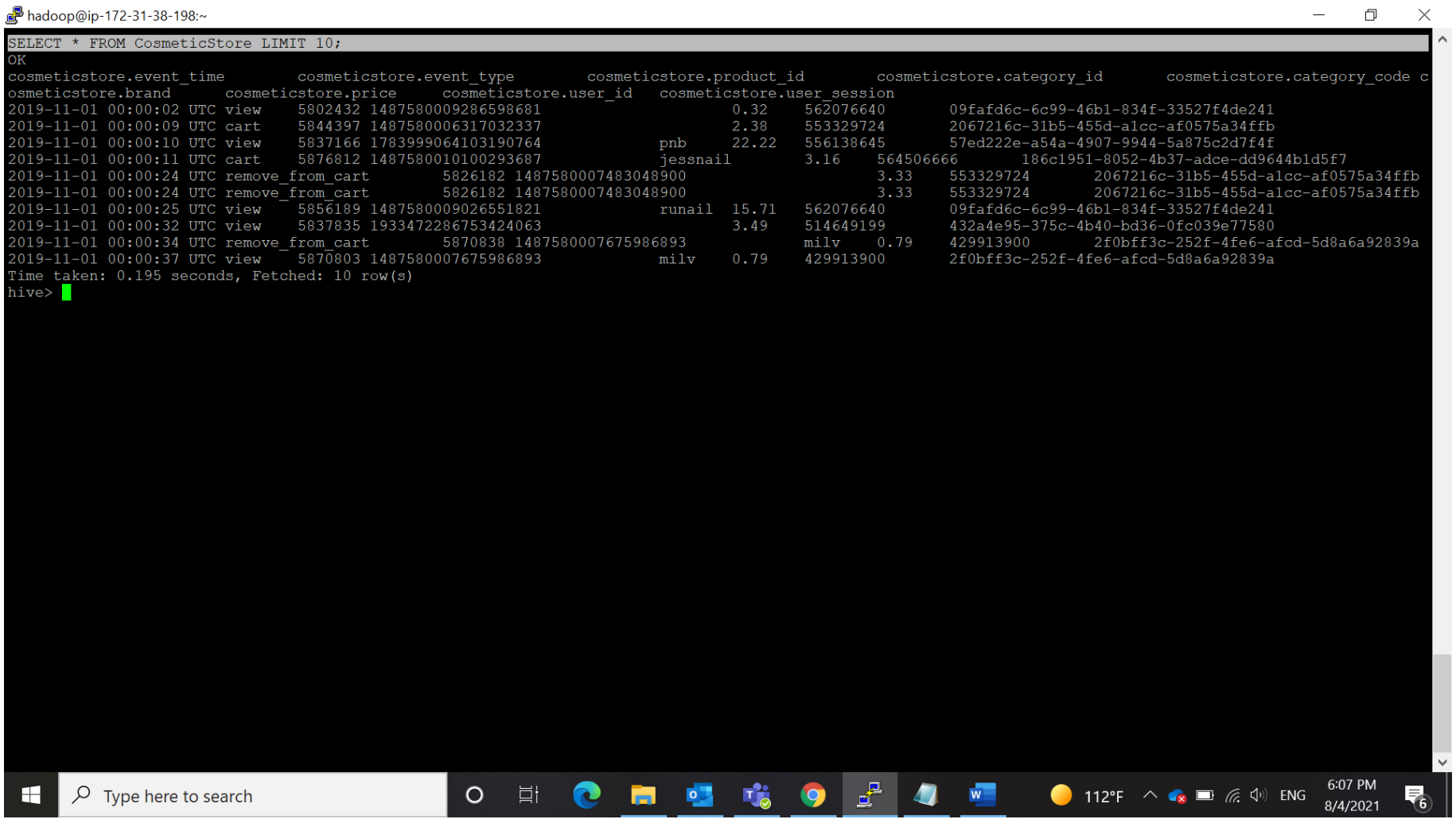


7. Loading Data Into the Table

load data local inpath '/home/hadoop/2019-Oct.csv' into table CosmeticStore;
load data local inpath '/home/hadoop/2019-Nov.csv' into table CosmeticStore;

8. Checking If the Data has Successfully Loaded into the Table

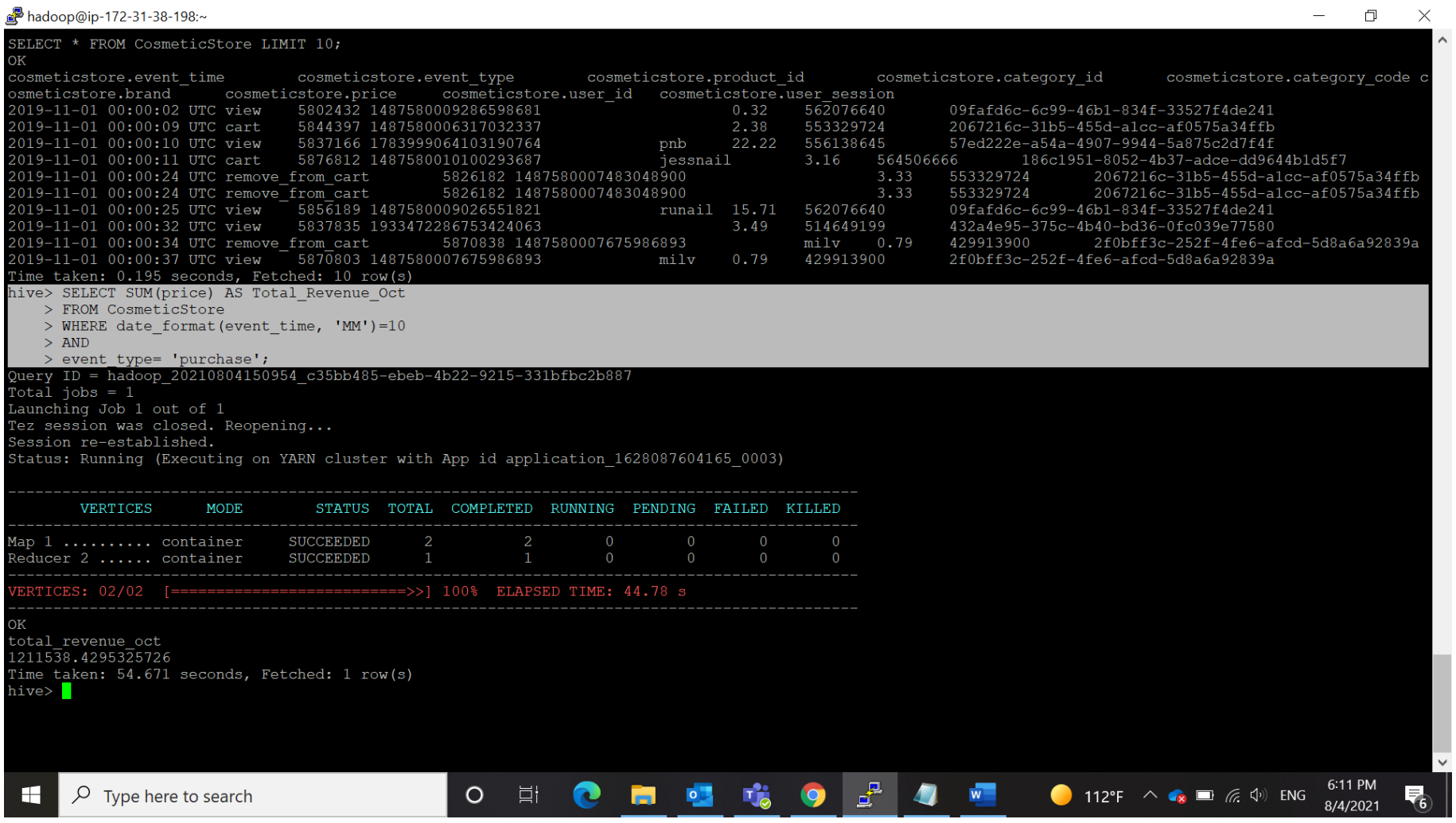
SELECT * FROM CosmeticStore LIMIT 5;



Questions:

1. Find the total revenue generated due to purchases made in October.

```
SELECT SUM(price) AS Total_Revenue_Oct
FROM CosmeticStore
WHERE date_format(event_time, 'MM')=10
AND
event_type= 'purchase';
```

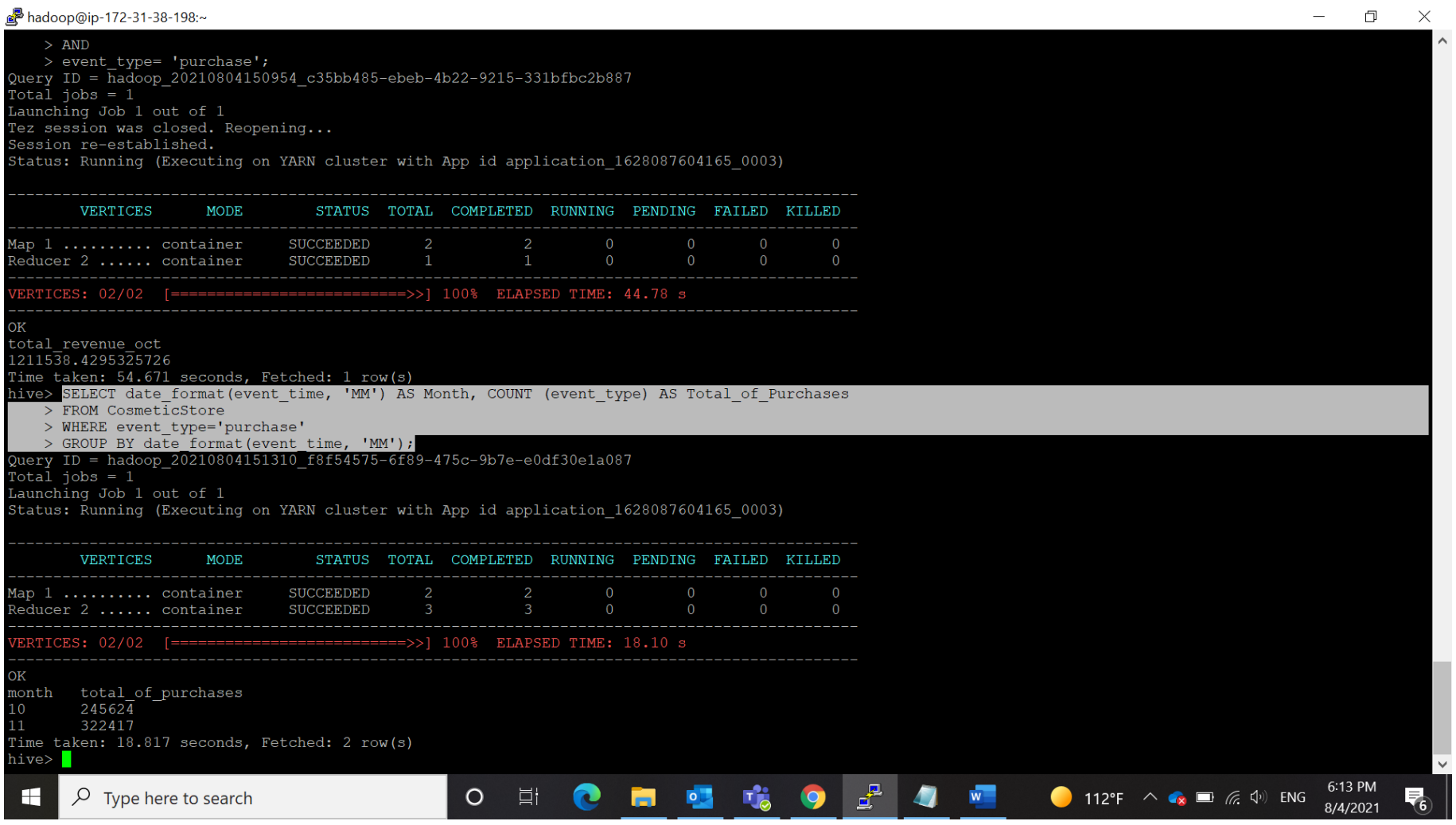


Findings:

- Based on Purchases, the total revenue generated in October 2019 was 12,11,538.43.

2. Write a query to yield the total sum of purchases per month in a single output.

```
SELECT date_format(event_time, 'MM') AS Month, COUNT (event_type) AS Total_of_Purchases
FROM CosmeticStore
WHERE event_type='purchase'
GROUP BY date_format(event_time, 'MM');
```

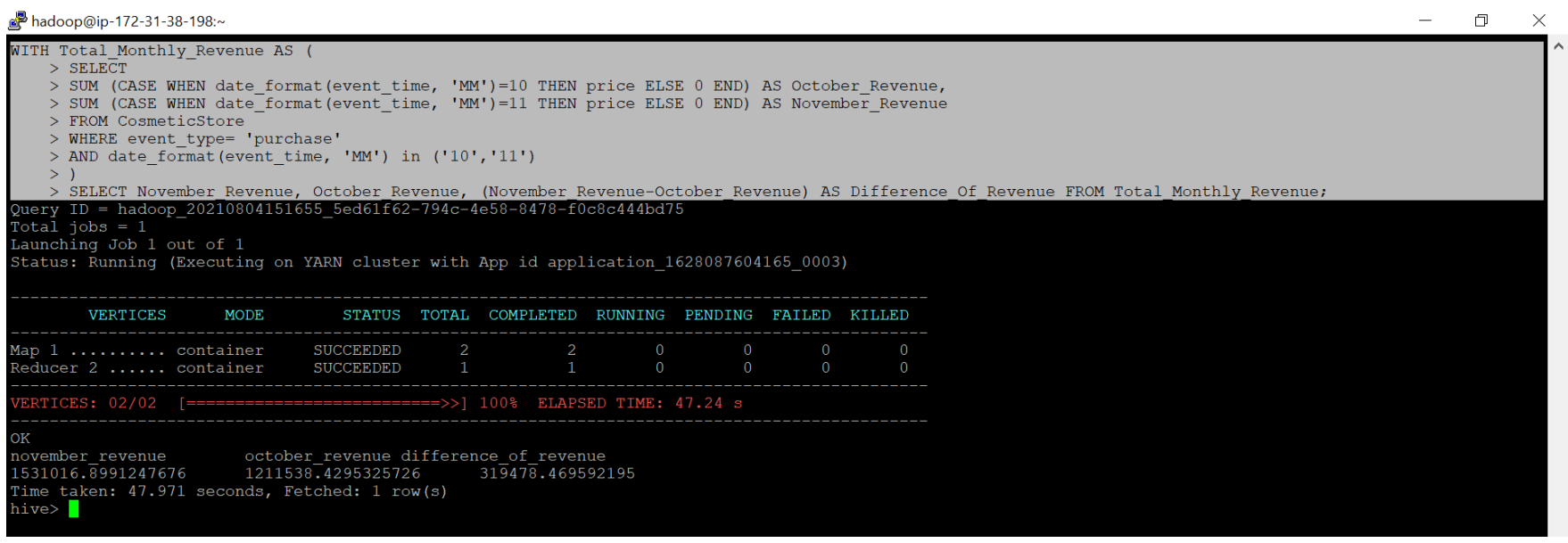


Findings:

- The total purchases made in November is 3,22,417 which exceeds the total purchases made in October i.e., 2,45,624.
- On seeing the above numbers, we can assume that November must have been more profitable than October. But we will proceed further on validating our assumptions.

3. Write a query to find the change in revenue generated due to purchases from October to November.

```
WITH Total_Monthly_Revenue AS (
SELECT
SUM (CASE WHEN date_format(event_time, 'MM')=10 THEN price ELSE 0 END) AS October_Revenue,
SUM (CASE WHEN date_format(event_time, 'MM')=11 THEN price ELSE 0 END) AS November_Revenue
FROM CosmeticStore
WHERE event_type= 'purchase'
AND date_format(event_time, 'MM') in ('10','11')
)
SELECT November_Revenue, October_Revenue, (November_Revenue-October_Revenue) AS
Difference_Of_Revenue FROM Total_Monthly_Revenue;
```



Findings:

- Considering Purchase as an event, we can finally conclude that Revenue in November 2019 exceeds revenue generation of October 2019 which also makes us say, since the organization had higher Sales in November 2019, November 2019 was a more profitable month over October 2019.

4. Find distinct categories of products. Categories with null code can be ignored.

```
SELECT DISTINCT SPLIT(category_code,'\\.')[0] AS Categories
FROM CosmeticStore
WHERE SPLIT (category_code,'\\.')[0]<>'';
```

```
hadoop@ip-172-31-38-198:~
SELECT DISTINCT SPLIT(category_code,'\\.')[0] AS Categories
    > FROM CosmeticStore
    > WHERE SPLIT (category_code,'\\.')[0]<>'';
Query ID = hadoop_20210804152149_286ec3d6-0cc0-4f81-a7a4-5653f97cfe3a
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1628087604165_0003)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      2          2        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      5          5        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 17.94 s
--------------------------------------------------------------------------------
OK
categories
furniture
appliances
accessories
apparel
sport
stationery
Time taken: 18.73 seconds, Fetched: 6 row(s)
hive>
```

Findings:

- The organization has a total of 6 different categories under which they place all their products (Furniture, Appliances, Accessories, Apparel, Sport and Stationery).

5. Find the total number of products available in each category.

```
SELECT SPLIT(category_code,'\\.')[0] AS Categories, COUNT (product_id) AS Count_Of_Products
FROM CosmeticStore
WHERE SPLIT(category_code,'\\.')[0]<>''
GROUP BY SPLIT(category_code,'\\.')[0];
```

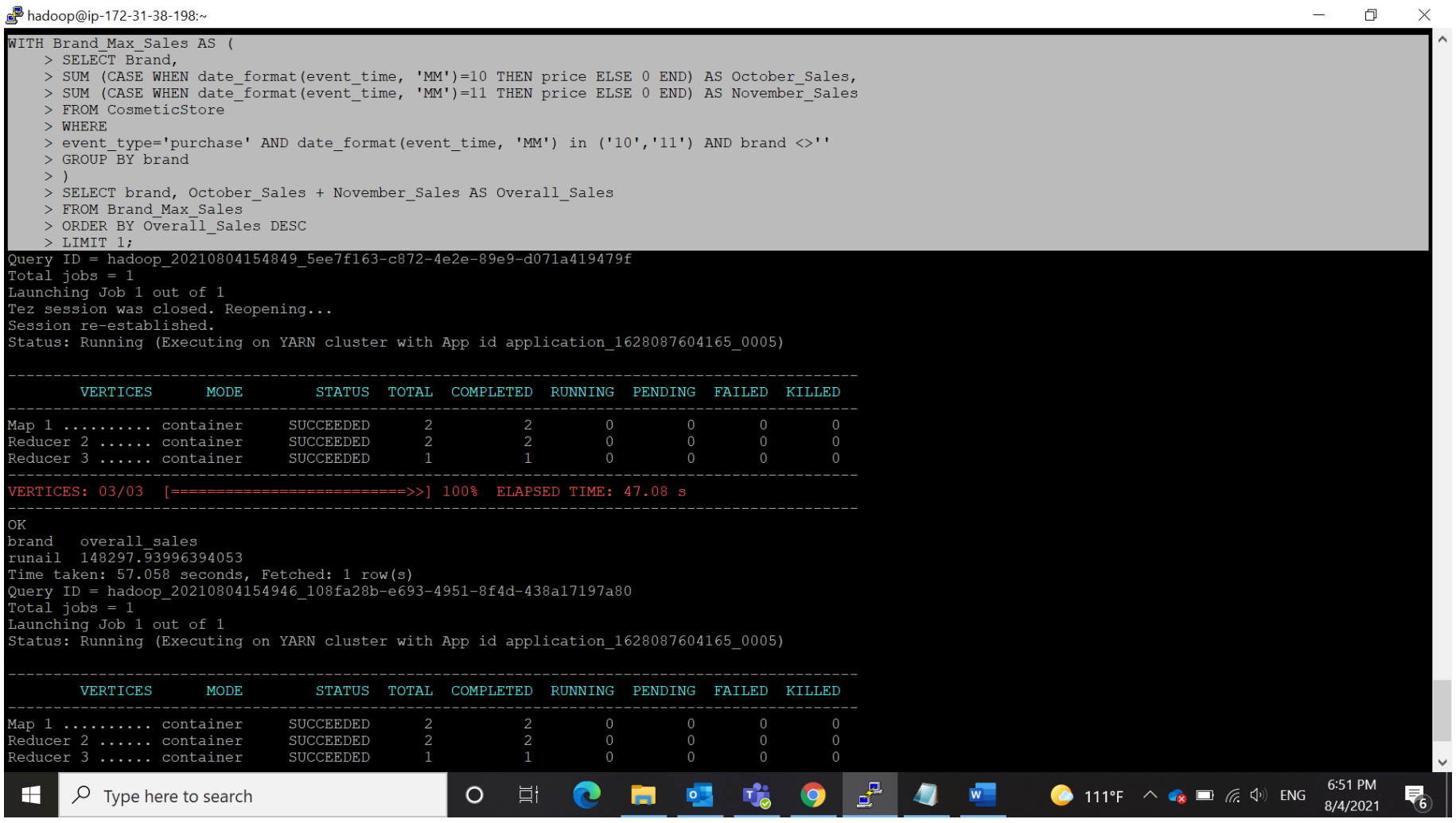
```
hadoop@ip-172-31-38-198:~
SELECT SPLIT(category_code,'\\.')[0] AS Categories, COUNT (product_id) AS Count_Of_Products
    > FROM CosmeticStore
    > WHERE SPLIT(category_code,'\\.')[0]<>''
    > GROUP BY SPLIT(category_code,'\\.')[0];
Query ID = hadoop_20210804153004_d359a46b-d5f8-4e3c-b38d-5568a6e90324
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1628087604165_0004)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      2          2        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      5          5        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 25.30 s
--------------------------------------------------------------------------------
OK
categories      count_of_products
furniture       23604
appliances      61736
accessories     12929
apparel 18232
sport   2
stationery      26722
Time taken: 34.35 seconds, Fetched: 6 row(s)
hive>
```

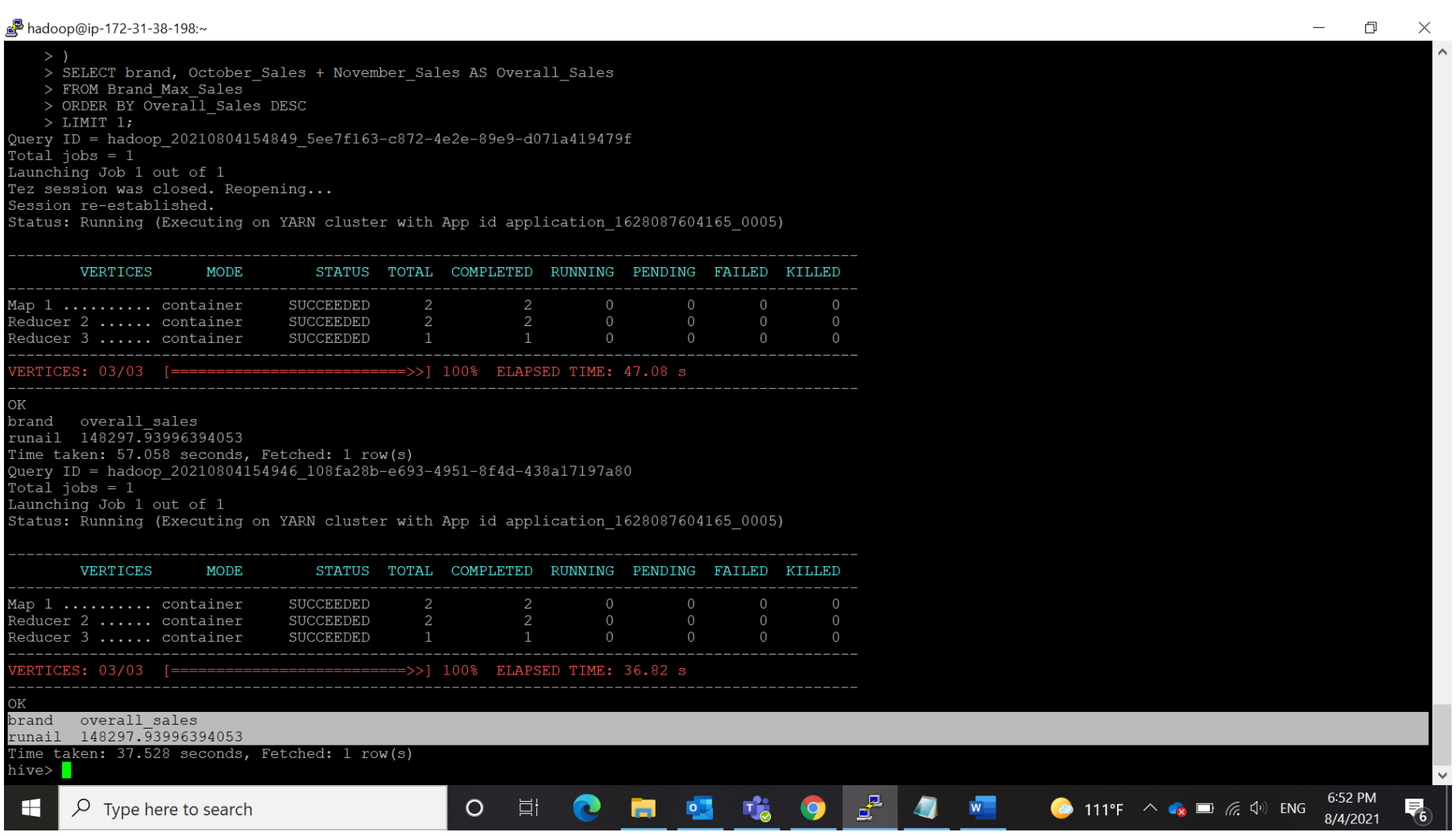
Findings:

- The Appliances category leads by having the most number of products listed under it (61,736 products).
- The second highest is stationery (26,722 products) followed by furniture category (23,604 products) further followed by apparel (18,232 products).
- Accessories ranks fifth with (12,929 products) listed under it and the last category is the sports category with only (2 products) listed under it.

6. Which brand has the maximum sales in October and November Combined?

```
WITH Brand_Max_Sales AS (
SELECT Brand,
SUM (CASE WHEN date_format(event_time, 'MM')=10 THEN price ELSE 0 END) AS October_Sales,
SUM (CASE WHEN date_format(event_time, 'MM')=11 THEN price ELSE 0 END) AS November_Sales
FROM CosmeticStore
WHERE
event_type='purchase' AND date_format(event_time, 'MM') in ('10','11') AND brand <>''
GROUP BY brand
)
SELECT brand, October_Sales + November_Sales AS Overall_Sales
FROM Brand_Max_Sales
ORDER BY Overall_Sales DESC
LIMIT 1;
```



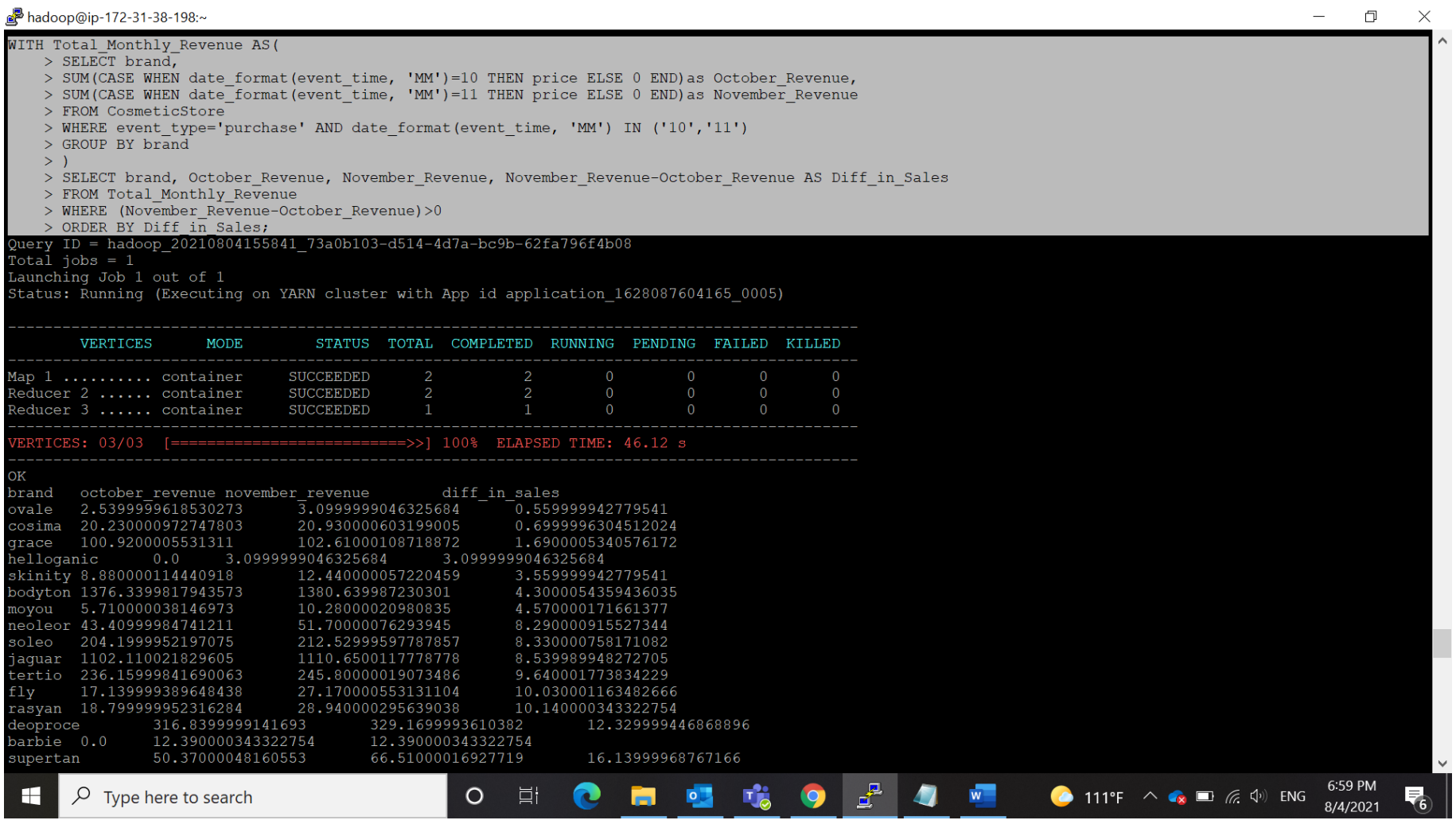


Findings:

- "Runnail" is the brand having the highest overall sales (1,48,297.9 as compared to other brands.
- Since Runnail is popular amongst the cosmetic consumers, introducing more products under this brand will help the organization in increasing their profits.

7. Which brands increased their sales from October to November?

```
WITH Total_Monthly_Revenue AS(
SELECT brand,
SUM(CASE WHEN date_format(event_time, 'MM')=10 THEN price ELSE 0 END)as October_Revenue,
SUM(CASE WHEN date_format(event_time, 'MM')=11 THEN price ELSE 0 END)as November_Revenue
FROM CosmeticStore
WHERE event_type='purchase' AND date_format(event_time, 'MM') IN ('10','11')
GROUP BY brand
)
SELECT brand, October_Revenue, November_Revenue, November_Revenue-October_Revenue AS Diff_in_Sales
FROM Total_Monthly_Revenue
WHERE (November_Revenue-October_Revenue)>0
ORDER BY Diff_in_Sales;
```



hadoop@ip-172-31-38-198:~

```
treaclemoon     163.36999654769897      181.48999691009521      18.12000036239624
kamill  63.010000228881836      81.48999953269958       18.47999930381775
juno    0.0     21.079999923706055      21.079999923706055
veraclara       50.11000084877014       71.21000015735626       21.09999930858612
glysolid        69.73000013828278       91.59000062942505       21.860000491142273
godefroy        401.22000312805176      425.1200022697449       23.899999141693115
binacil 0.0     24.260000228881836      24.260000228881836
blixz   38.94999921321869       63.400001764297485      24.450002551078796
profepil        93.36000156402588       118.01999974250793      24.659998178482056
estelare        444.80999556183815      471.86999905109406      27.060003489255905
orly    902.3799939155579       931.0899903774261       28.709996461868286
biore   60.650001525878906       90.30999946594238       29.659997940063477
beautyblender   78.73999977111816       109.41000175476074      30.670001983642578
vilenta 197.59999787807465      231.20999908447266      33.61000120639801
mavala  409.0400023460388       446.32000255584717      37.28000020980835
likato  296.0599980354309       340.9699954986572       44.90999746322632
ladykin 125.64999961853027      170.56999969482422      44.920000076293945
foamie  35.03999996185303       80.48999977111816       45.44999980926514
elskin  251.0900001525879       307.6499996781349       56.55999952554703
balbcare        155.33000373840332      212.3800015449524       57.04999780654907
koelcia 55.5    112.75  57.25
profhenna       679.2300038337708       736.8500001430511       57.619996309280396
kares   0.0     59.45000076293945       59.45000076293945
marutaka-foot   49.21999979019165       109.33000040054321      60.11000061035156
dewal   0.0     61.28999876976013       61.28999876976013
inm     288.01999855041504      351.2099983692169       63.18999981880188
laboratorium    246.49999952316284      312.5199975967407       66.01999807357788
cutrin  299.3700017929077       367.6199998855591       68.24999809265137
egomania        77.46999835968018       146.04000091552734      68.57000255584717
konad   739.8300001621246       810.6699978709221       70.83999770879745
nirvel  163.04000329971313      234.33000826835632      71.29000496864319
koelf   422.7300081253052       507.29000186920166      84.55999374389648
plazan  101.37000036239624      194.010000705719        92.64000034332275
aura    83.95000076293945       177.5100040435791       93.56000328063965
kerasys 430.9100044965744       525.2000050544739       94.29000055789948
enjoy   41.34999966621399       136.57000184059143      95.22000217437744
depilflax       2707.0699973106384      2803.7799961566925      96.70999884605408
eos     54.34000015258789       152.60999727249146      98.26999711990356
carmex  145.07999897003174      243.3599967956543       98.27999782562256
batiste 772.400013923645        874.1700088977814       101.76999497413635
osmo    645.5800037384033       762.3100028038025       116.72999906539917
dizao   819.1300112009048       945.5100176334381       126.38000643253326
igrobeauty      513.6600003838539       645.0699995160103       131.40999913215637
finish  98.37999773025513       230.37999820709229      132.00000047683716
```


hadoop@ip-172-31-38-198:~

```
metzger 5373.4499744176865      6457.159960865974       1083.709986448288
de.lux  1659.7000161707401      2775.510024756193       1115.810008585453
swarovski       1887.9299856424332      3043.1599831581116      1155.2299975156784
beauty-free     554.1699986457825       1782.8599914312363      1228.6899927854538
zeitun  708.6600031852722       2009.6300013065338      1300.9699981212616
joico   705.5200037956238       2015.1000146865845      1309.5800108909607
severina        4775.8799668848515      6120.479953020811       1344.5999861359596
irisk   45591.96021157503       46946.04018642008       1354.0799748450518
oniq    8425.409879207611       9841.649902820587       1416.240023612976
levrana 2243.5599967837334      3664.0999879837036      1420.5399911999702
roubloff        3491.3600150346756      4913.770027637482       1422.410012602806
smart   4457.259982824326       5902.139976501465       1444.8799936771393
shik    3341.199989080429       4839.720018148422       1498.5200290679932
domix   10472.05003106594       12009.170008182526      1537.1199771165848
artex   2730.6399517059326      4327.249953508377       1596.6100018024445
beautix 10493.949965000153      12222.95004272461       1729.0000777244568
milv    3904.940046072006       5642.01002573967        1737.0699796676636
masura  31266.079910814762      33058.469878435135      1792.3899676203728
f.o.x   6624.229980587959       8577.279987692833       1953.0500071048737
kapous  11927.159952402115      14093.079938054085      2165.91998565197
concept 11032.14000660181       13380.400002479553      2348.2599958777428
estel   21756.749947547913      24142.66994935274       2385.9200018048286
kaypro  881.3400187492371       3268.700007915497       2387.3599891662598
benovy  409.619996547699        3259.969982147217       2850.349985599518
italwax 21940.239994883537      24799.37004429102       2859.130049407482
yoko    8756.910053431988       11707.88005465269       2950.970001220703
haruyama        9390.690077126026       12352.910059452057      2962.2199823260307
marathon        7280.749939441681       10273.099990844727      2992.3500514030457
lovely  8704.380010932684       11939.059989094734      3234.6799781620502
bpw.style       11572.1500659585        14837.440190911293      3265.290124952793
staleks 8519.730030417442       11875.610019385815      3355.8799889683723
freedecor       3421.7800273299217      7671.800070524216       4250.020043194294
runail  71539.28005346656       76758.65991047397       5219.379857007414
polarus 6013.720007181168       11371.930022716522      5358.210015535355
cosmoprofi      8322.80991601944        14536.989881515503      6214.179965496063
jessnail        26287.840348243713      33345.23023867607       7057.389890432358
strong  29196.63009786606       38671.27037525177       9474.640277385712
ingarden        23161.38997283578       33566.209977939725      10404.820005103946
lianail 5892.839952707291       16394.239884018898      10501.399931311607
uno     35302.029363155365      51039.74947929382       15737.720116138458
grattol 35445.53947067261       71472.70888674259       36027.169416069984
        474679.05964545906      619509.2397020273       144830.18005656824
Time taken: 46.784 seconds, Fetched: 161 row(s)
hive>
```


Findings:

- There are around 161 brands which show higher sales in November over October.
- The brand “Grattol” has the highest increment of 36,027 followed by Uno being the second highest with an increment of 15,738.The brand with the least increment is Ovale with an increment of 0.56.
- Earlier we saw Runnail to be the best seller brand in terms of Total Sales for October & November. Runnail is also on this list being the 9th highest brand with an increment of 5219.37.

8. Your company wants to reward the top 10 users of its website with a Golden Customer Plan. Write a query to generate a list of top 10 users who spend the most.

- We will use this question to check on Optimization of Query. First we will run the query on the entire table & check the execution time taken to run the query.

```
SELECT user_id, SUM(price) as Total_Spend
FROM CosmeticStore
WHERE event_type='purchase'
GROUP BY user_id
ORDER BY Total_Spend DESC LIMIT 10;
```

hadoop@ip-172-31-38-198:~

```
SELECT user_id, SUM(price) as Total_Spend
    > FROM CosmeticStore
    > WHERE event_type='purchase'
    > GROUP BY user_id
    > ORDER BY Total_Spend DESC
    > LIMIT 10;
Query ID = hadoop_20210804160529_52d0e4e4-dcf8-4bfa-ab9e-0afa2be5da41
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1628087604165_0006)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      2          2        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      3          3        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 17.68 s
--------------------------------------------------------------------------------
OK
user_id total_spend
557790271       2715.8699957430363
150318419       1645.970008611679
562167663       1352.8499938696623
531900924       1329.4499949514866
557850743       1295.4800310581923
522130011       1185.3899966478348
561592095       1109.700007289648
431950134       1097.5900000333786
566576008       1056.3600097894669
521347209       1040.9099964797497
Time taken: 26.888 seconds, Fetched: 10 row(s)
hive> [hadoop@ip-172-31-38-198 ~]$
```


Time taken to run the above query is 26.888 seconds.

- We will try and optimize the query.

A) Setting the rules to create dynamic partitioning

```
set hive.exec.dynamic.partition.mode=nonstrict;
set hive.exec.dynamic.partition=true;
```

B) Creating table with partition on event_type and clustering on price.

```
CREATE TABLE IF NOT EXISTS Dynamic_Part_Cluster_CosmeticStore(
event_time string, product_id string, category_id string, category_code string, brand string, price float,
user_id bigint, user_session string
)
PARTITIONED BY (event_type string)
CLUSTERED BY (price) INTO 7 BUCKETS
ROW FORMAT delimited fields terminated by "," lines terminated by "\n" STORED AS TEXTFILE;
```

hadoop@ip-172-31-38-198:~

```
CREATE TABLE IF NOT EXISTS Dynamic_Part_Cluster_CosmeticStore(
    > event_time string, product_id string, category_id string, category_code string, brand string, price float, user_id bigint, user_session string
    > )
    > PARTITIONED BY (event_type string)
    > CLUSTERED BY (price) INTO 7 BUCKETS
    > ROW FORMAT delimited fields terminated by "," lines terminated by "\n" STORED AS TEXTFILE;
OK
Time taken: 0.079 seconds
hive> show tables;
OK
cosmeticstore
dynamic_part_cluster_cosmeticstore
Time taken: 0.029 seconds, Fetched: 2 row(s)
hive>
```

C) Adding data from CosmeticStore table into partitioned and clustered table.

```
INSERT INTO TABLE Dynamic_Part_Cluster_CosmeticStore
PARTITION (event_type)
SELECT event_time, product_id, category_id, category_code, brand, price, user_id, user_session, event_type
FROM CosmeticStore;
```

```
hadoop@ip-172-31-38-198:~
CREATE TABLE IF NOT EXISTS Dynamic_Part_Cluster_CosmeticStore(
    > event_time string, product_id string, category_id string, category_code string, brand string, price float, user_id bigint, user_session string
    > )
    > PARTITIONED BY (event_type string)
    > CLUSTERED BY (price) INTO 7 BUCKETS
    > ROW FORMAT delimited fields terminated by "," lines terminated by "\n" STORED AS TEXTFILE;
OK
Time taken: 0.079 seconds
hive> show tables;
OK
cosmeticstore
dynamic_part_cluster_cosmeticstore
Time taken: 0.029 seconds, Fetched: 2 row(s)
hive> INSERT INTO TABLE Dynamic_Part_Cluster_CosmeticStore
    > PARTITION (event_type)
    > SELECT event_time, product_id, category_id, category_code, brand, price, user_id, user_session, event_type
    > FROM CosmeticStore;
Query ID = hadoop_20210804162545_1393d293-ec42-4cef-adf9-714adeff6e49
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1628087604165_0008)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      2          2        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      5          5        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 83.91 s
--------------------------------------------------------------------------------
Loading data to table cosmetics.dynamic_part_cluster_cosmeticstore partition (event_type=null)

Loaded : 4/4 partitions.
        Time taken to load dynamic partitions: 0.672 seconds
        Time taken for adding to write entity : 0.002 seconds
OK
Time taken: 96.992 seconds
hive>
```


D) Checking whether the table has been succesfully created and loaded with the data

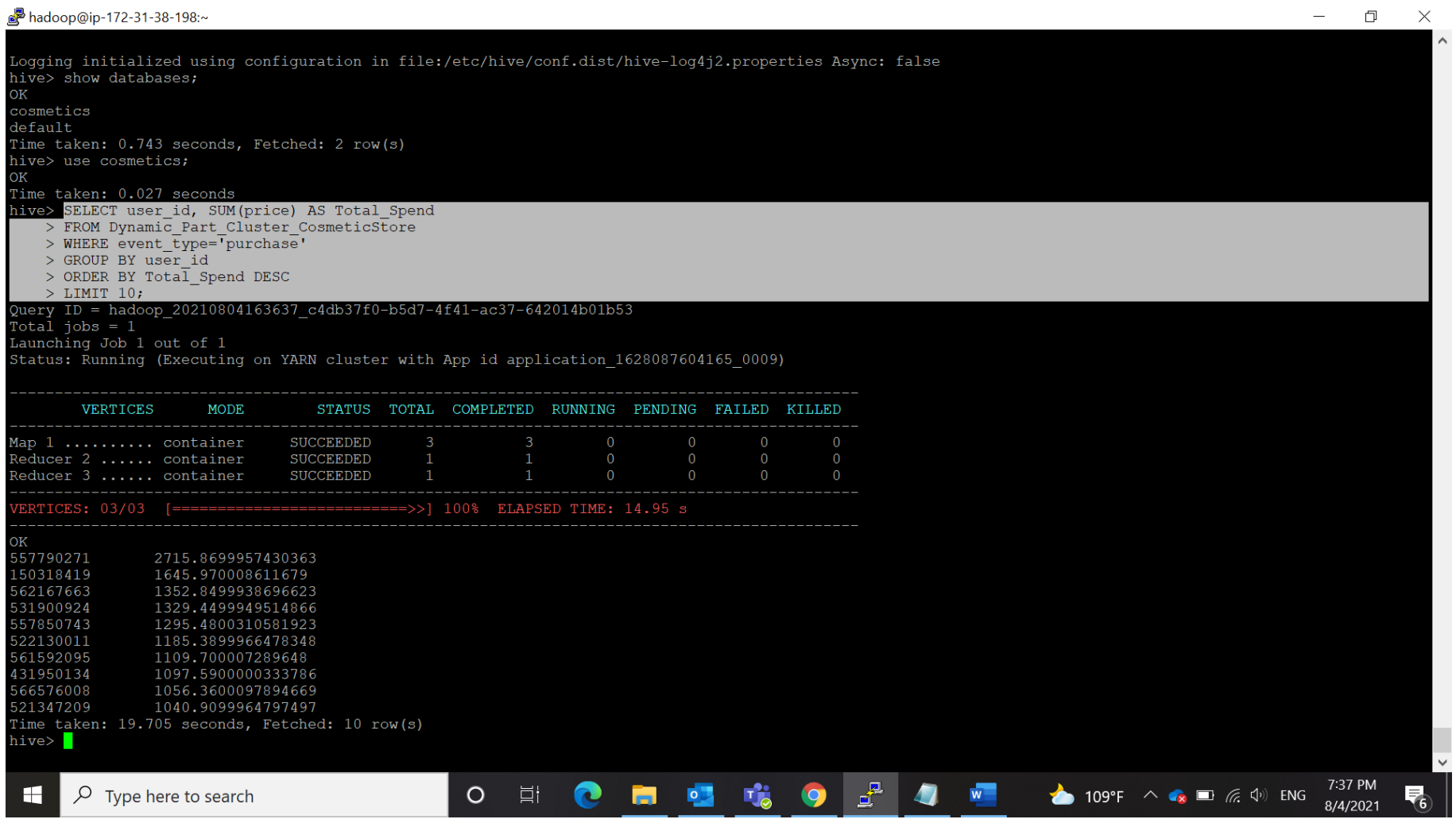
```
hadoop fs -ls /user/hive/warehouse
```

```
hadoop@ip-172-31-38-198:~
hive> show tables;
OK
cosmeticstore
dynamic_part_cluster_cosmeticstore
Time taken: 0.029 seconds, Fetched: 2 row(s)
hive> INSERT INTO TABLE Dynamic_Part_Cluster_CosmeticStore
    > PARTITION (event_type)
    > SELECT event_time, product_id, category_id, category_code, brand, price, user_id, user_session, event_type
    > FROM CosmeticStore;
Query ID = hadoop_20210804162545_1393d293-ec42-4cef-adf9-714adeff6e49
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1628087604165_0008)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      2          2        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      5          5        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 83.91 s
--------------------------------------------------------------------------------
Loading data to table cosmetics.dynamic_part_cluster_cosmeticstore partition (event_type=null)

Loaded : 4/4 partitions.
        Time taken to load dynamic partitions: 0.672 seconds
        Time taken for adding to write entity : 0.002 seconds
OK
Time taken: 96.992 seconds
hive> exit;
[hadoop@ip-172-31-38-198 ~]$ hadoop fs-ls /user/hive/warehouse/Dynamic_Part_Cluster_CosmeticStore
Error: Could not find or load main class fs-ls
[hadoop@ip-172-31-38-198 ~]$ hadoop fs -ls /user/hive/warehouse/Dynamic_Part_Cluster_CosmeticStore
ls: `/user/hive/warehouse/Dynamic_Part_Cluster_CosmeticStore': No such file or directory
[hadoop@ip-172-31-38-198 ~]$ hadoop fs -ls /user/hive
Found 1 items
drwxrwxrwt   - hdfs hadoop          0 2021-08-04 16:12 /user/hive/warehouse
[hadoop@ip-172-31-38-198 ~]$ hadoop fs -ls /user/hive/warehouse
Found 2 items
drwxrwxrwt   - hadoop hadoop          0 2021-08-04 16:13 /user/hive/warehouse/cosmetics.db
drwxrwxrwt   - hadoop hadoop          0 2021-08-04 16:12 /user/hive/warehouse/dynamic_part_cluster_cosmeticstore
[hadoop@ip-172-31-38-198 ~]$
```


E) Reentering Hive and executing the Query

```
SELECT user_id, SUM(price) AS Total_Spend
FROM Dynamic_Part_Cluster_CosmeticStore
WHERE event_type='purchase'
GROUP BY user_id
ORDER BY Total_Spend DESC
```

LIMIT 10;

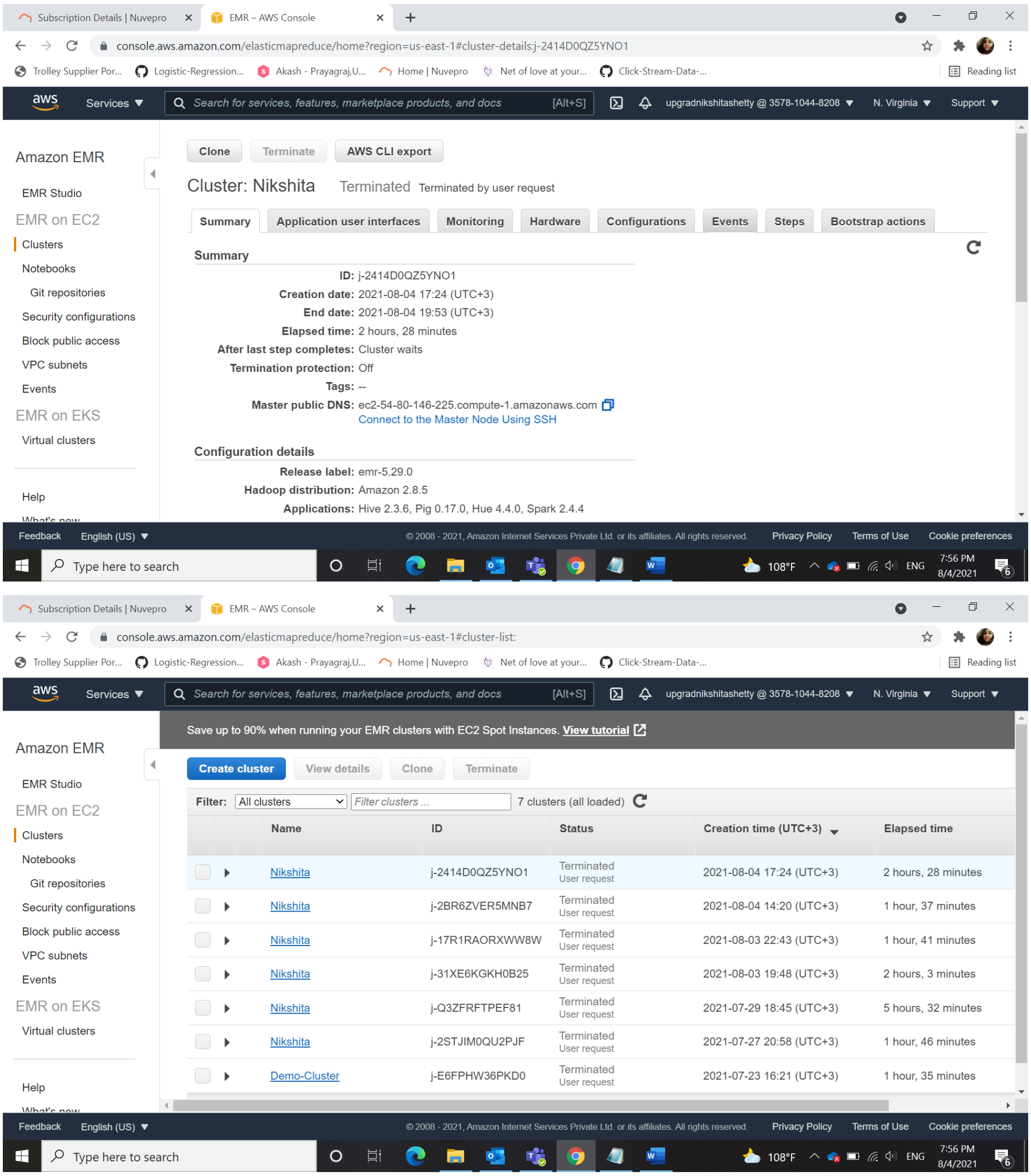


Findings:

- We created an Optimized Table by Partitioning on 'event_type' and Bucketing on 'price' to reduce the execution time for the query which is now 19.705 seconds.
- The above screenshot shows the list of 10 users who can be included in the Gold Plan.

User_ID	Total_Spend
557790271	2715.87
150318419	1645.97
562167663	1352.85
531900924	1329.45
557850743	1295.48
522130011	1185.39
561592095	1109.70
431950134	1097.59
566576008	1056.36
521347209	1040.91

9. Finally Terminating the Cluster



Thank You