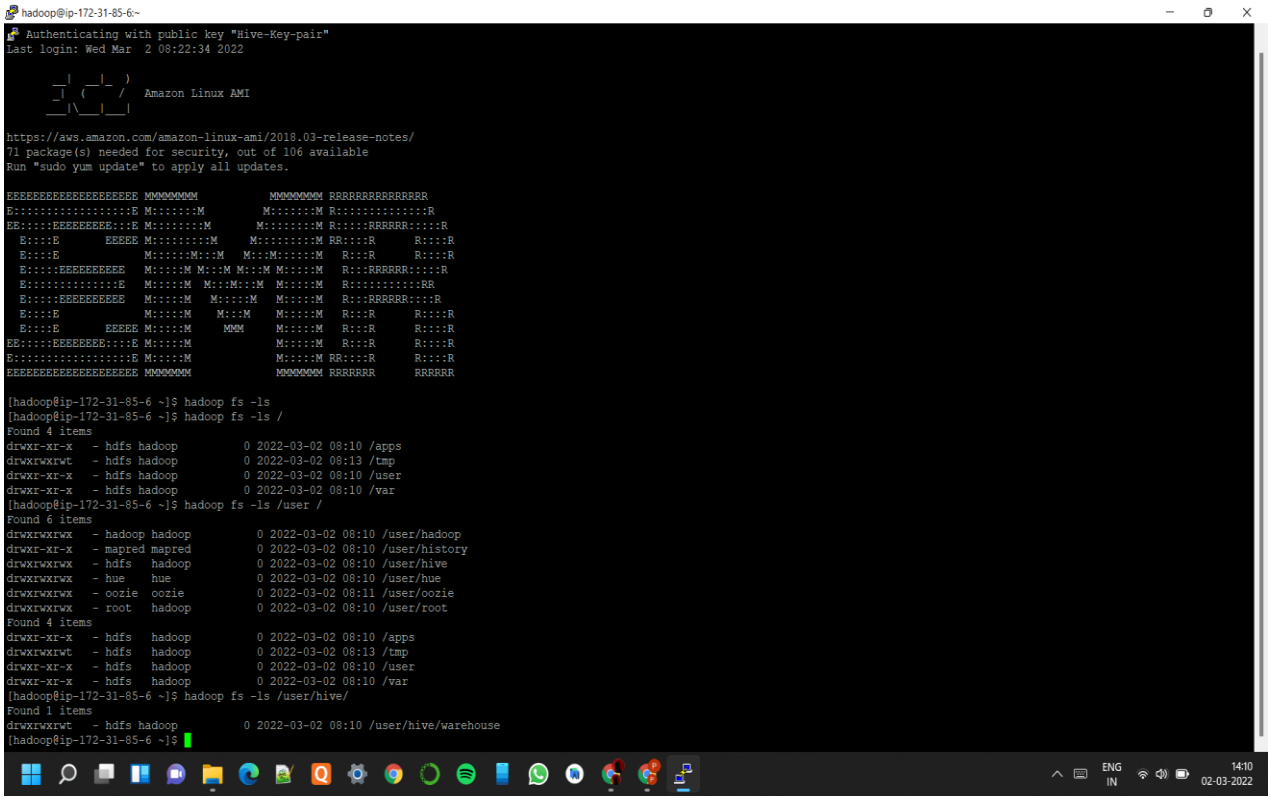
HIVE CASE STUDY

BY PRANAV JOSHI and DARSHAN SAWANT

Problem Statement:

With online sales gaining popularity, tech companies are exploring ways to improve their sales by analysing customer behaviour and gaining insights about product trends. Furthermore, the websites make it easier for customers to find the products they require without much scavenging. Needless to say, the role of big data analysts is among the most sought-after job profiles of this decade. Therefore, as part of this assignment, we will be challenging you, as a big data analyst, to extract data and gather insights from a real-life data set of an e-commerce company.

EMR CLI is launched

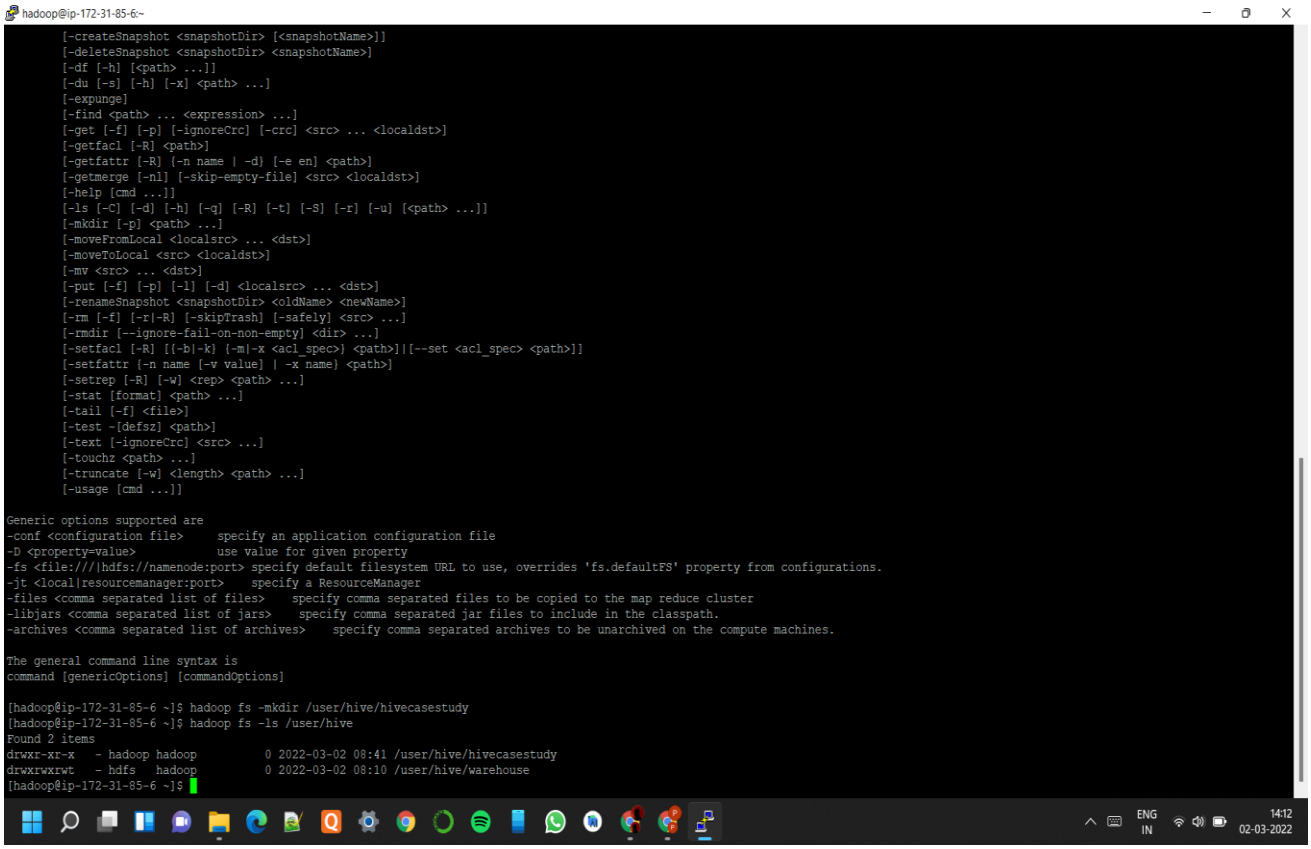


CREATING A NEW DIRECTORY FOR HIVE CASE STUDY:

Commands:

hadoop fs -mkdir /user/hive/hivecasestudy

hadoop fs -ls /user/hive/



New directory is successfully created

LOADING THE DATA FROM S3 BUCKET to HDFS:

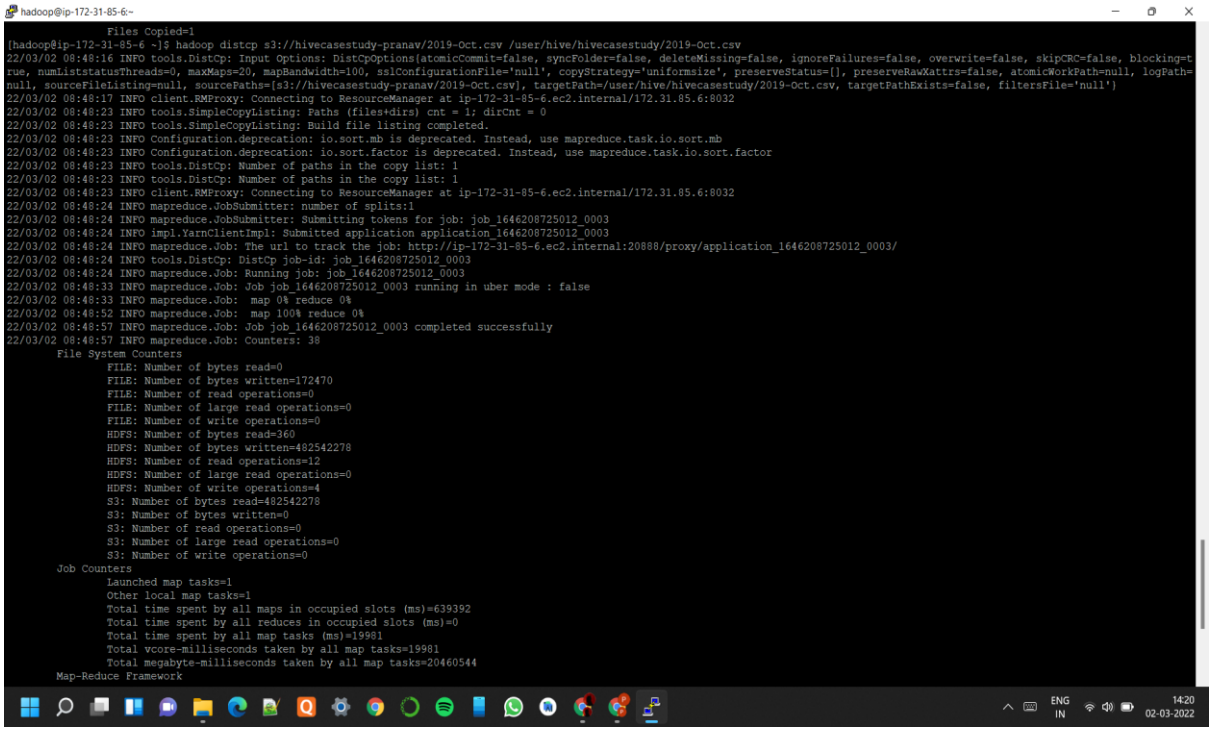
Copy the data from S3 to HDFS

For 2019 October:

hadoop distcp s3://hivecasestudy-pranav/2019-Oct.csv /user/hive/hivecasestudy/2019-Oct.csv

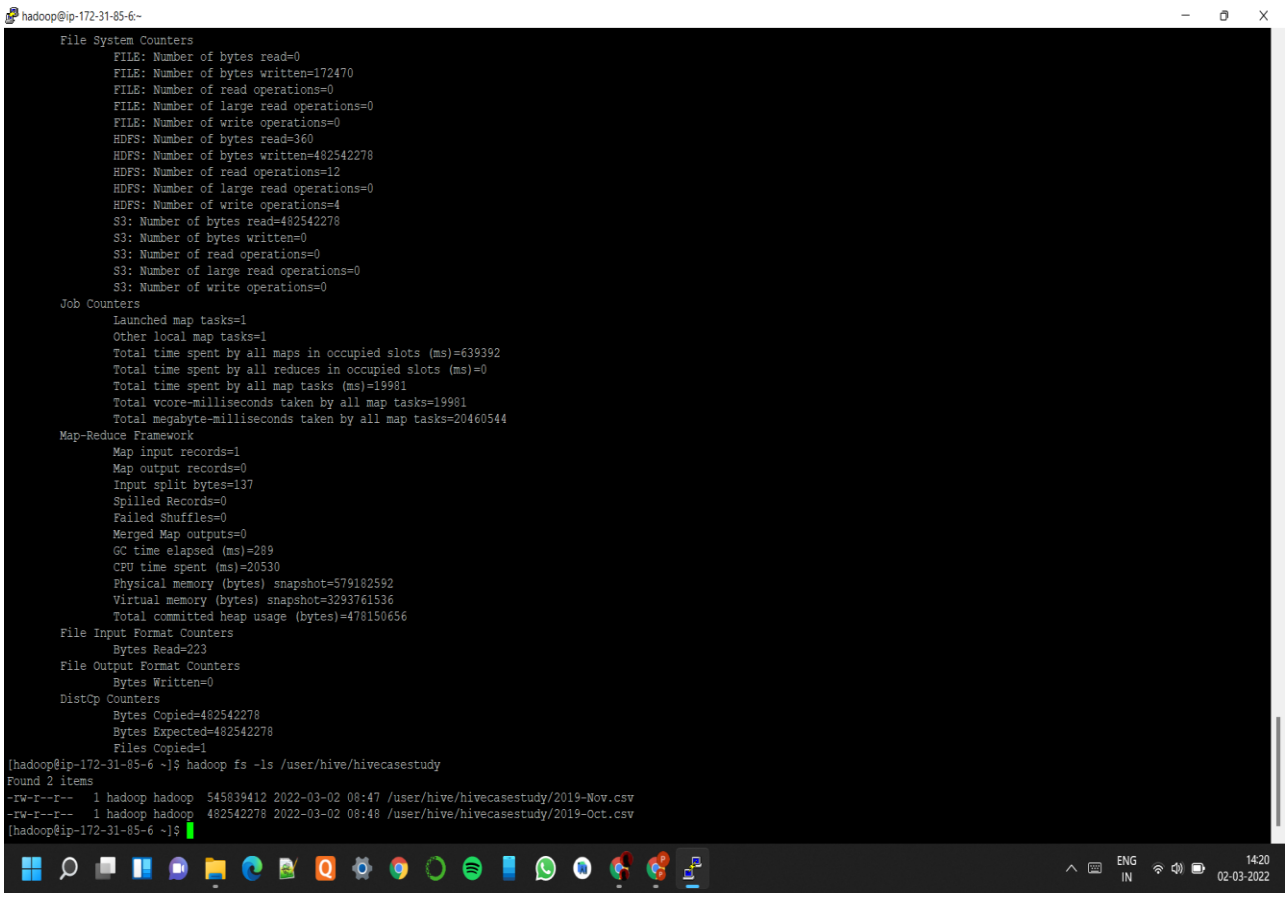
For 2019 November:

hadoop distcp s3://hivecasestudy-pranav/2019-Nov.csv /user/hive/hivecasestudy/2019-Nov.csv

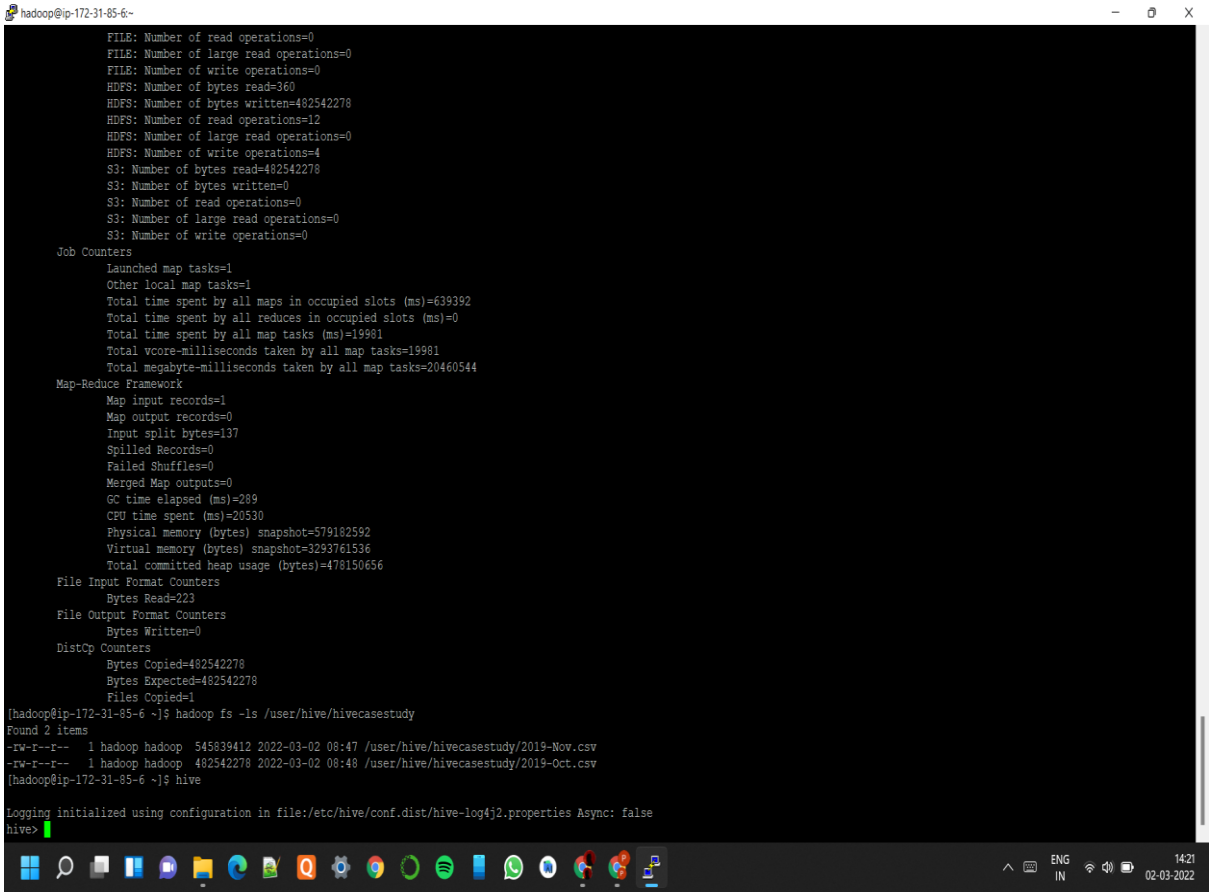


Verifying whether the data is successfully copied into HDFS from S3 buckets

Command: hadoop fs -ls /user/hive/hivecasestudy

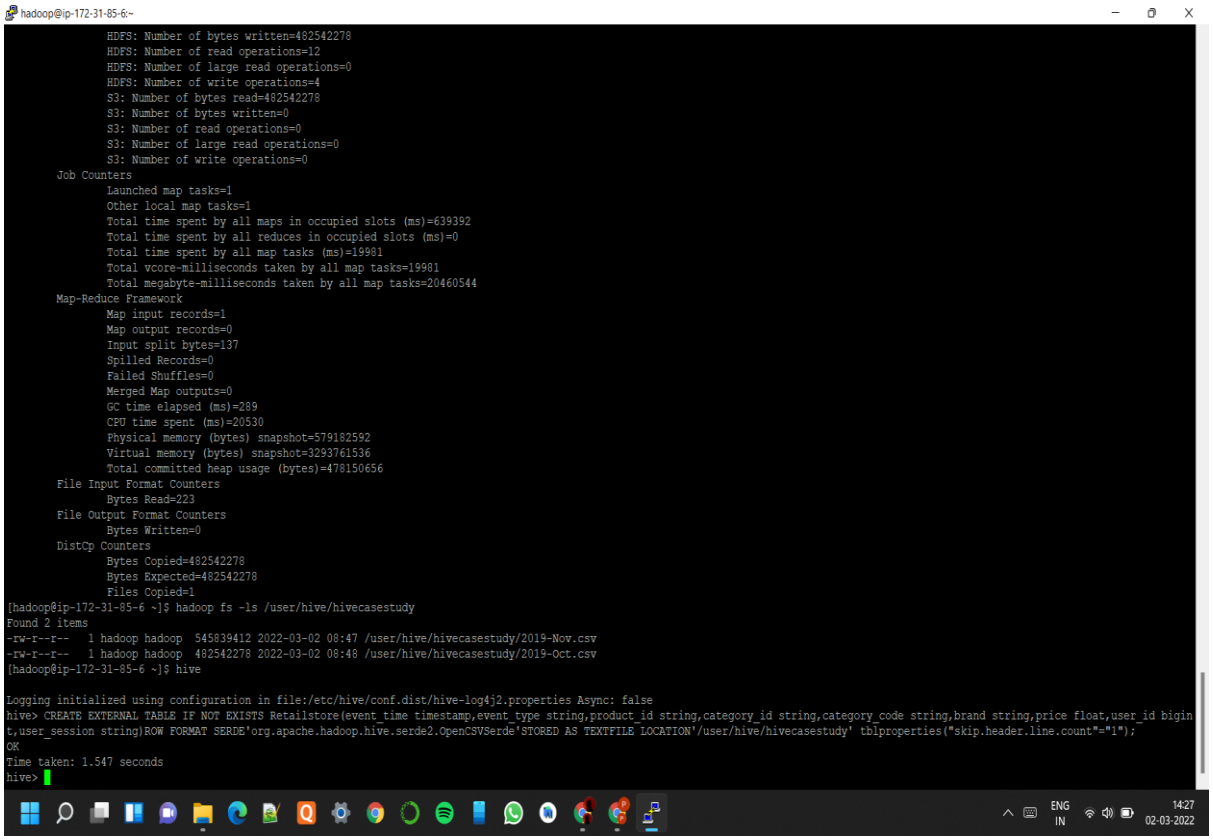


Moving to hive:



CREATING AN EXTERNAL TABLE IN HIVE:

CREATE EXTERNAL TABLE IF NOT EXISTS Retailstore (event_time timestamp, event_type string, product_id string, category_id string, category_code string, brand string, price float, user_id bigint, user_session string) ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' STORED AS TEXTFILE LOCATION '/user/hive/hivecasestudy' tblproperties("skip.header.line.count"="1");



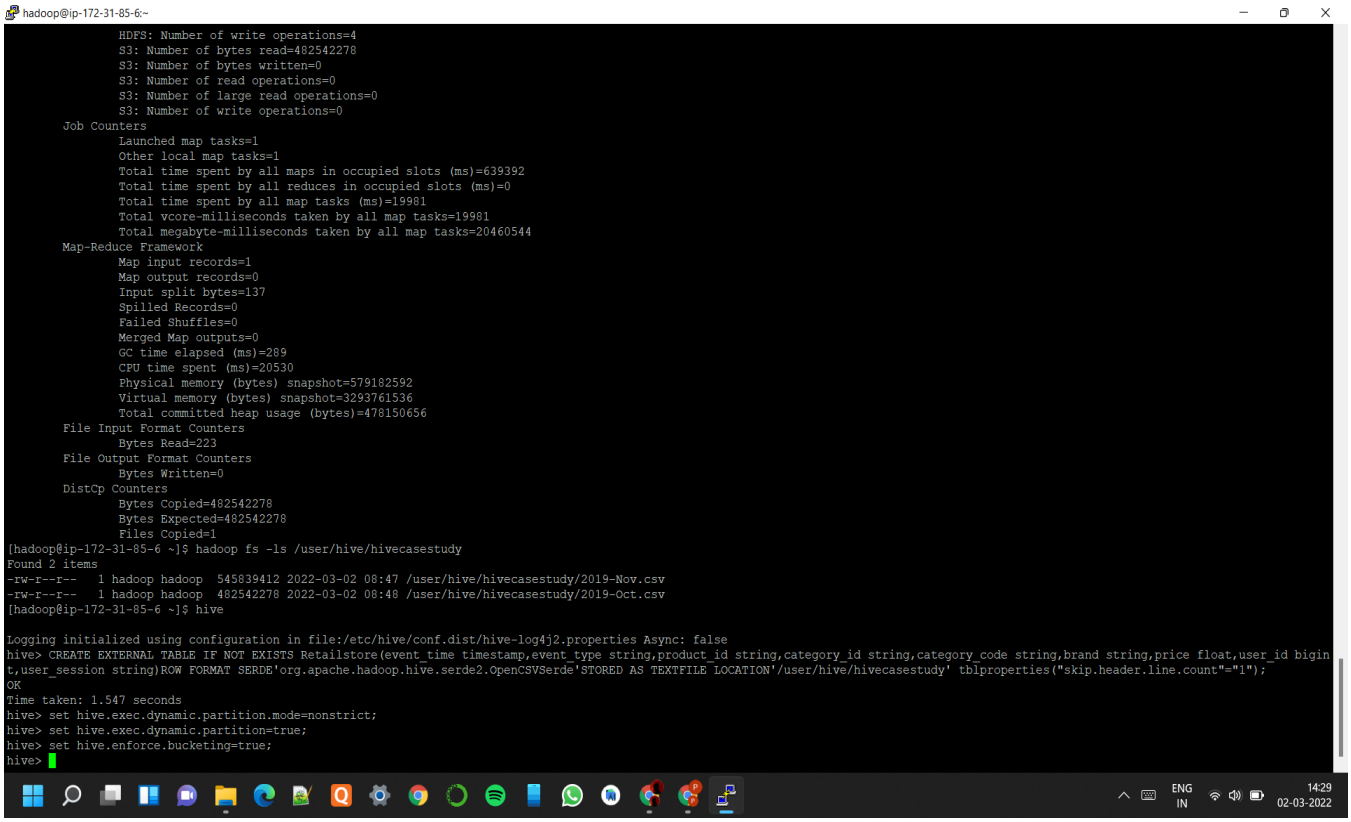
APPLYING OPTIMIZATION TECHNIQUES - PARTITIONING AND BUCKETING:

Below commands are to enable the dynamic partitioning and bucketing

hive> set hive.exec.dynamic.partition.mode = nonstrict;

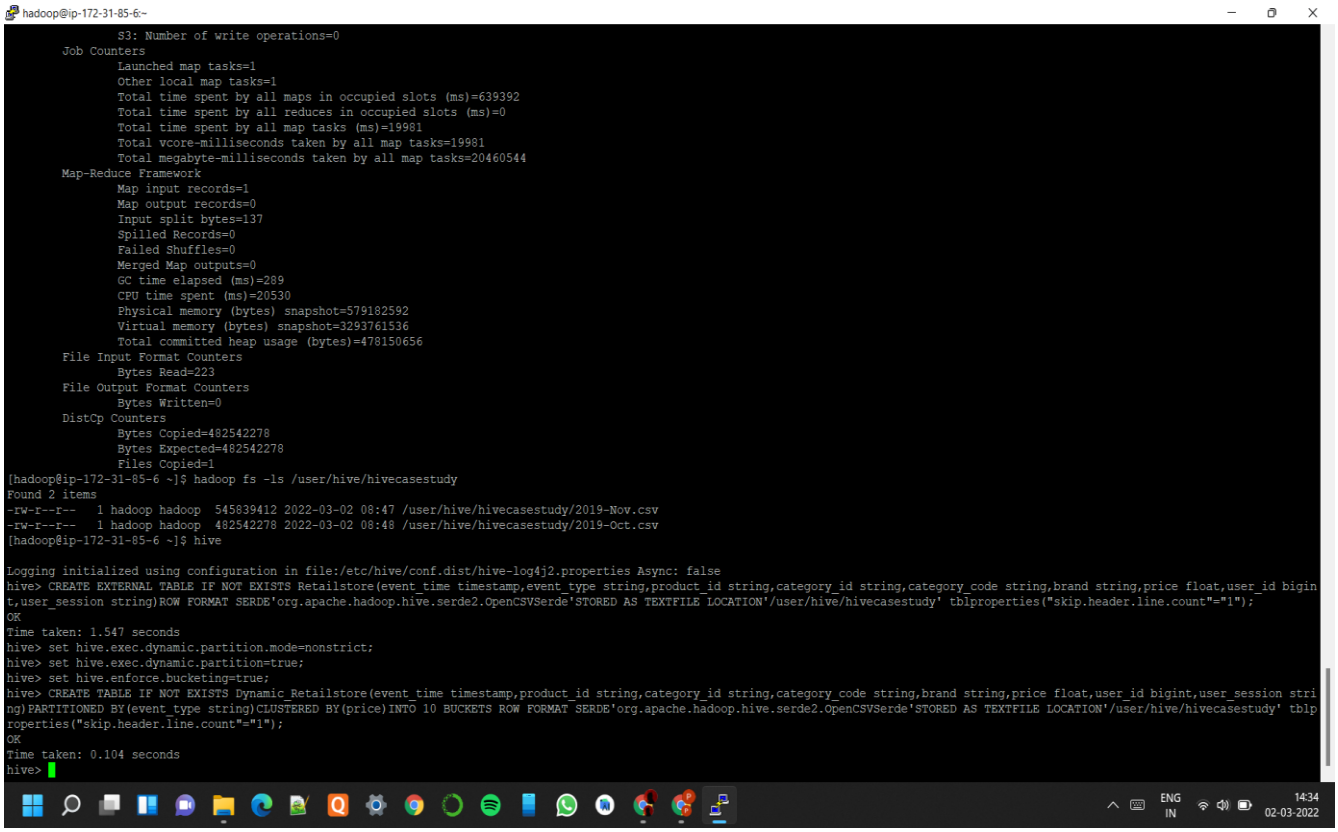
hive> set hive.exec.dynamic.partition = true;

hive> set hive.enforce.bucketing = true;



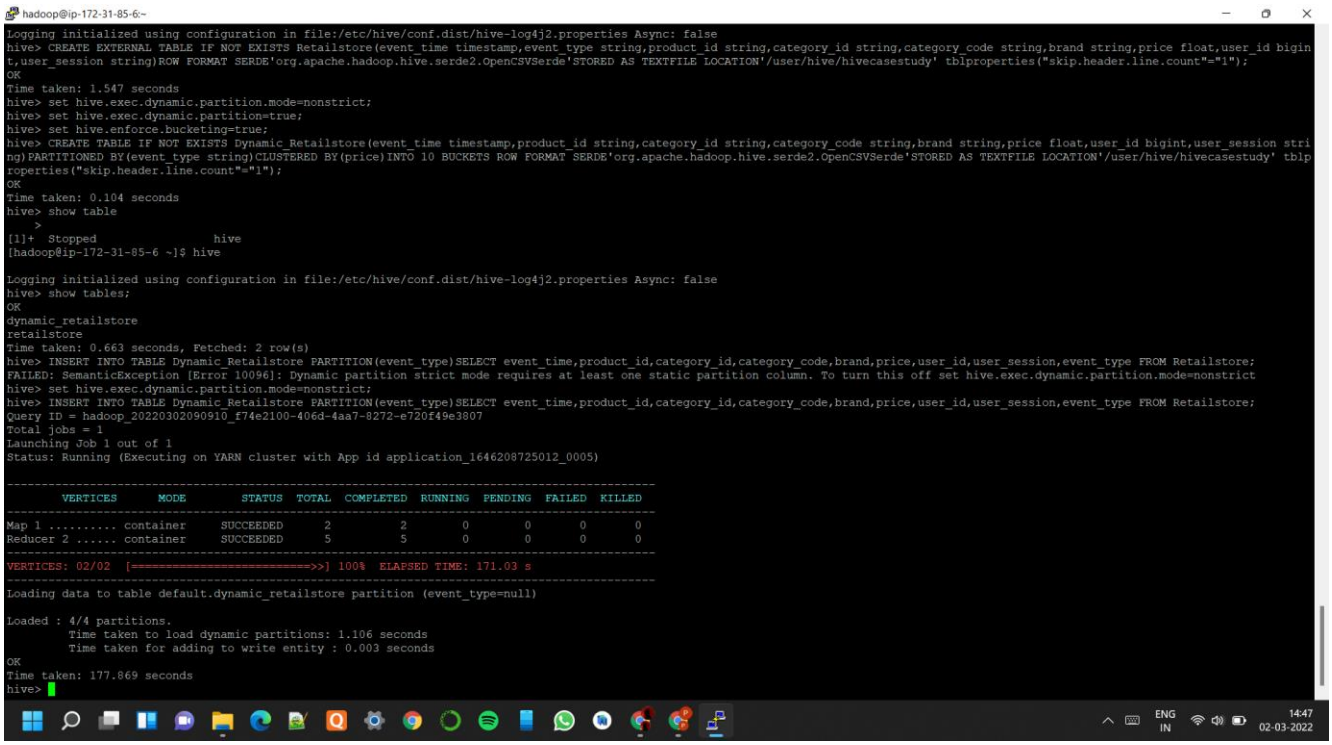
Creating an optimized table by applying partitioning on “event_type” and bucketing on “price”

CREATE TABLE IF NOT EXISTS Dynamic_Retailstore(event_time timestamp, 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 10 BUCKETS ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' STORED AS TEXTFILE LOCATION '/user/hive/hivecasestudy' tblproperties('skip.header.line.count' = '1');



INSERTING THE DATA INTO NEWLY CREATED OPTIMIZED TABLE (Dynamic_Retailstore) FROM EXISTING TABLE(Retailstore):

INSERT INTO TABLE Dynamic_Retailstore PARTITION (event_type) SELECT event_time, product_id, category_id, category_code, brand, price, user_id, user_session, event_type FROM Retailstore;



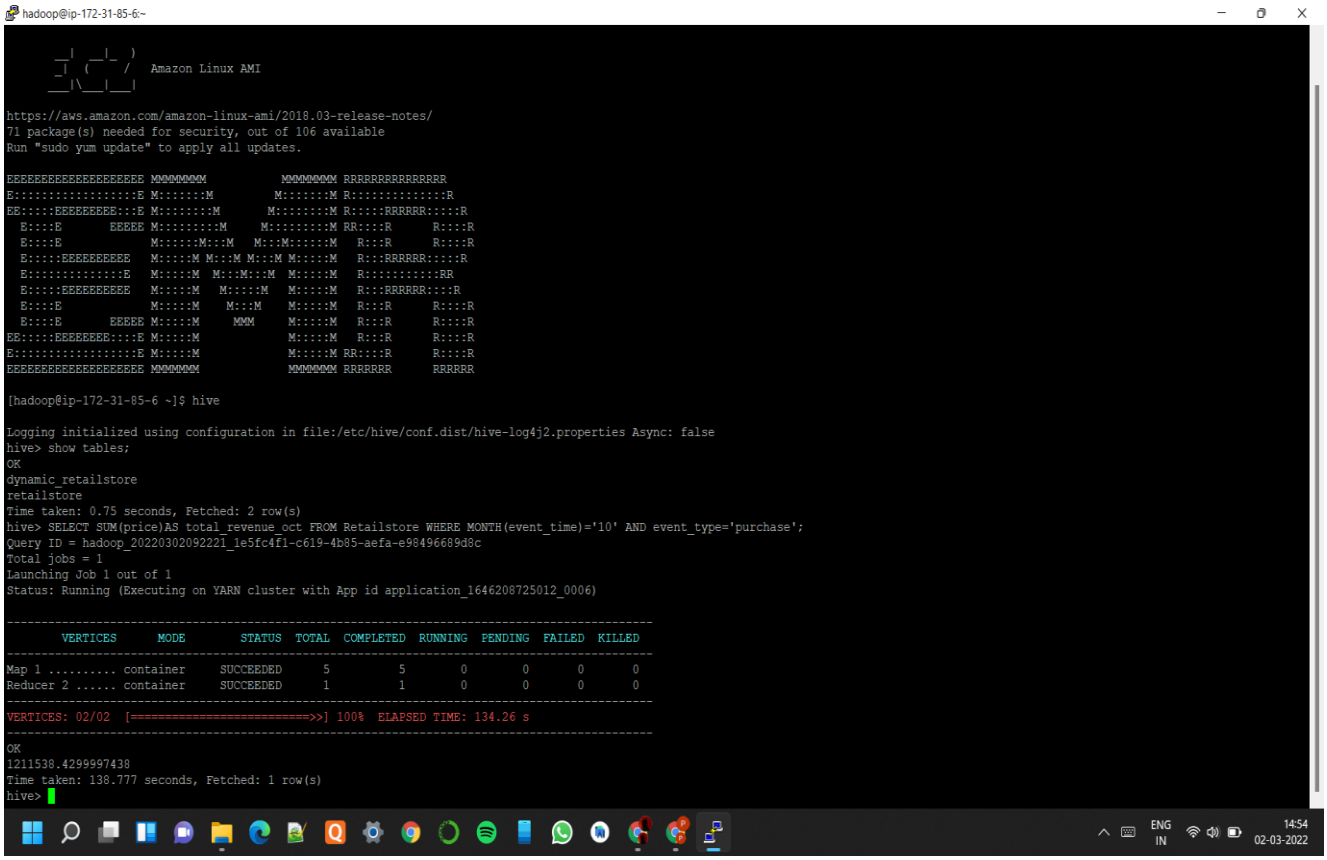
Output: Based on the above results, it partitioned into 4

ANSWERING GIVEN QUESTIONS:

find the total revenue generated due to purchases made in October

Base table:

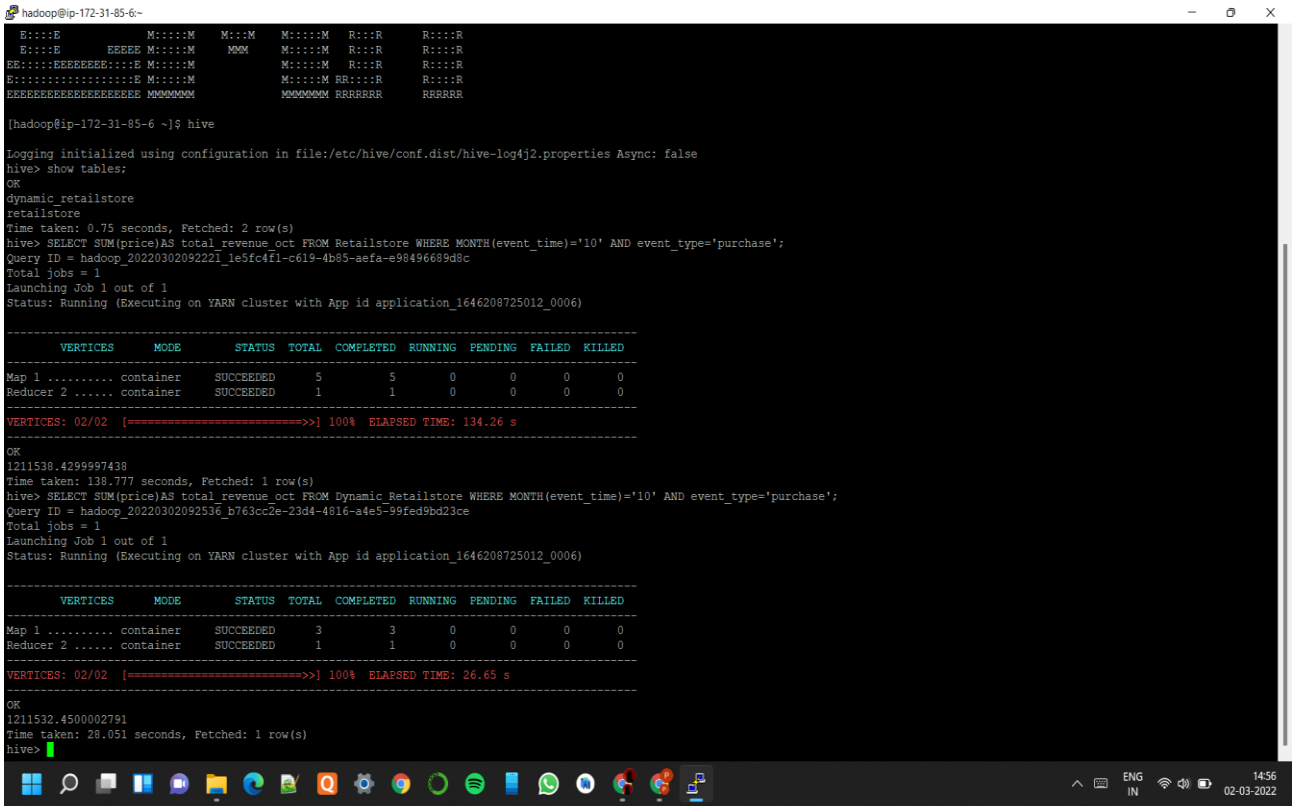
SELECT SUM(price) AS total_revenue_oct FROM Retailstore WHERE MONTH(event_time) = '10' AND event_type = 'purchase';



Time taken is 138.77 seconds

Optimized table:

SELECT SUM(price) AS total_revenue_oct FROM Dynamic_Retailstore WHERE MONTH(event_time) = 10 AND event_type = 'purchase';



Time taken with optimized table is 28.051 seconds

Insights:

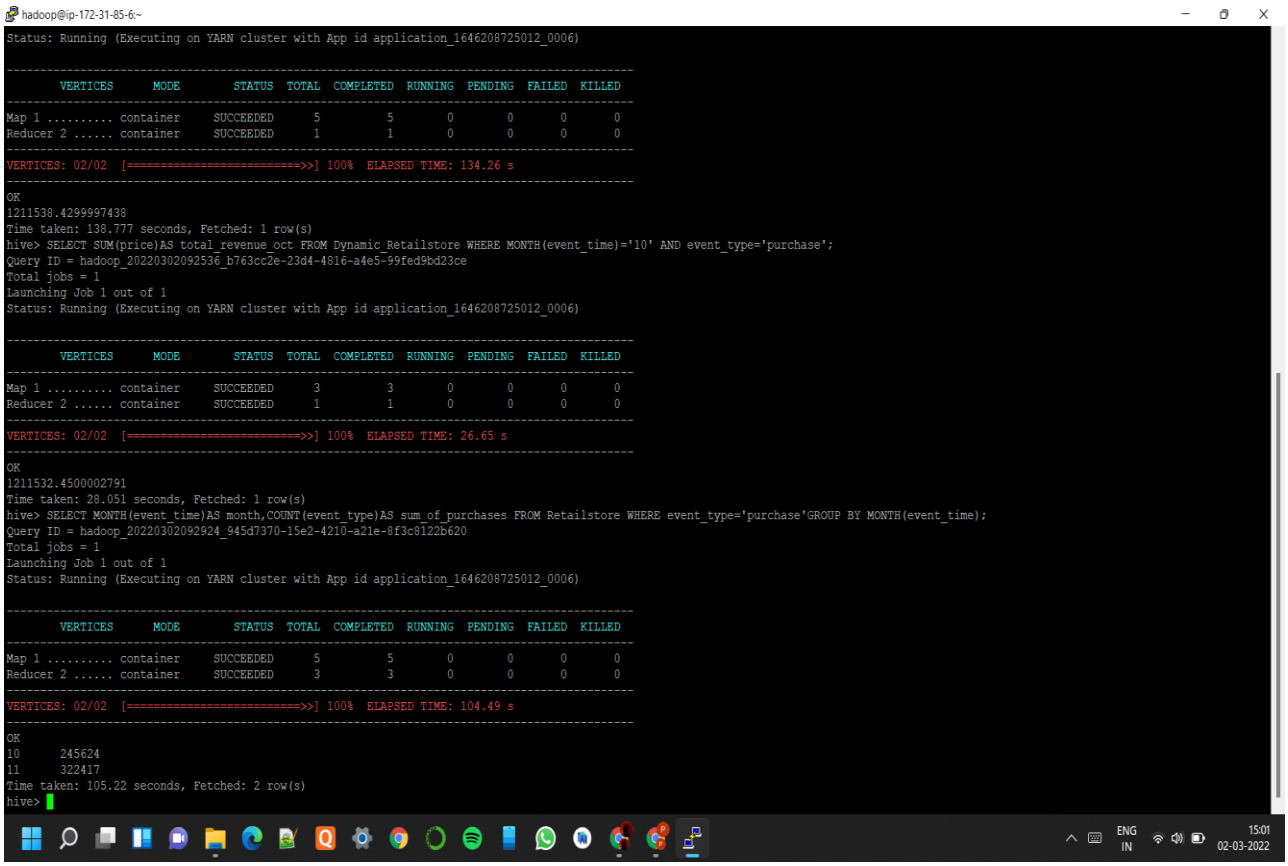
1. The total revenue generated based on Purchase made in the month of October is 1,211,532.45 /-

2. Hence, optimized table gives better performance in execution time.

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

Base Query:

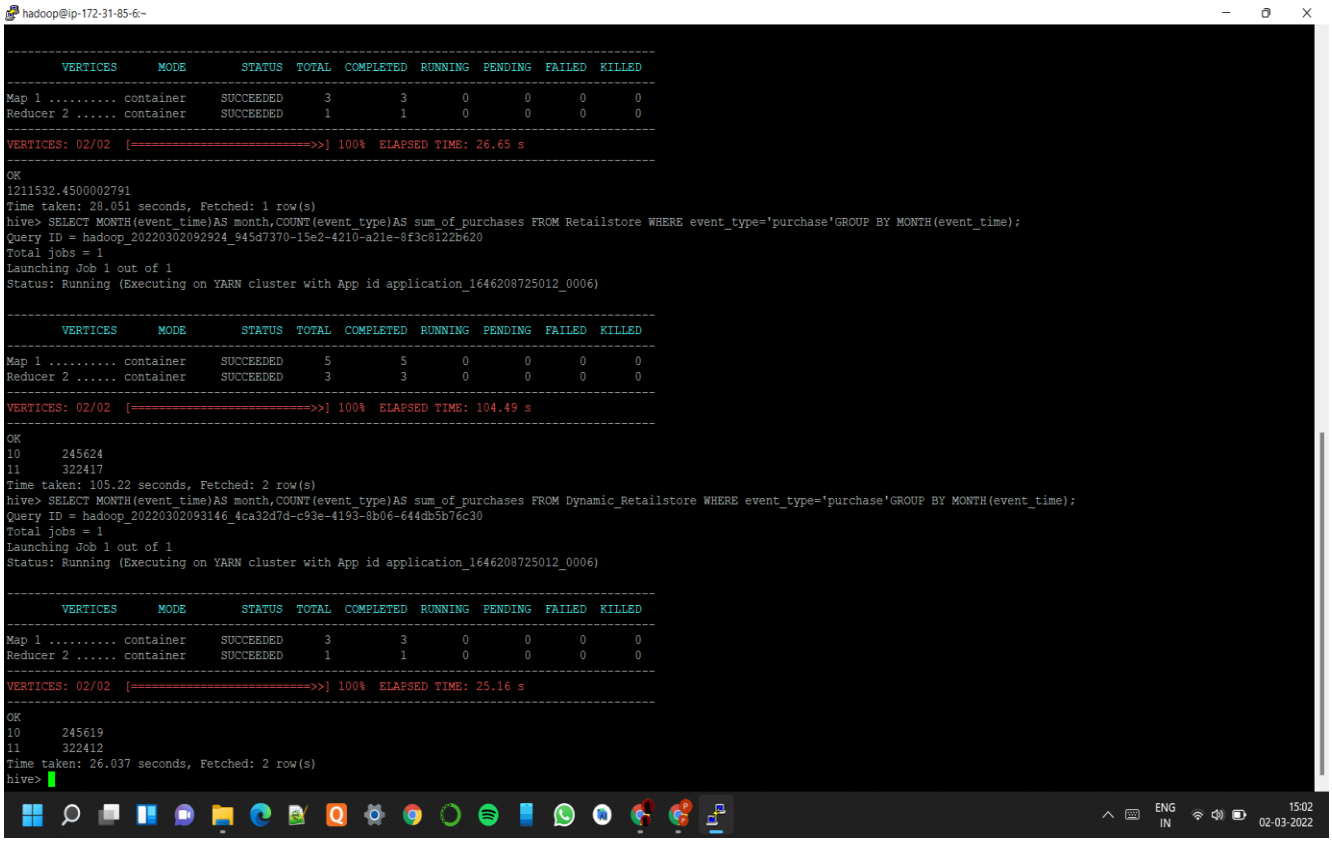
SELECT MONTH(event_time) AS month, COUNT(event_type) AS sum_of_purchases FROM Retailstore WHERE event_type = 'purchase' GROUP BY MONTH(event_time);



Time taken is 105.22 seconds

Optimized table:

SELECT MONTH(event_time) AS month, COUNT(event_type) AS sum_of_purchases FROM Dynamic_Retailstore WHERE event_type = 'purchase' GROUP BY MONTH(event_time);



Time taken is 26.037 seconds

Insights:

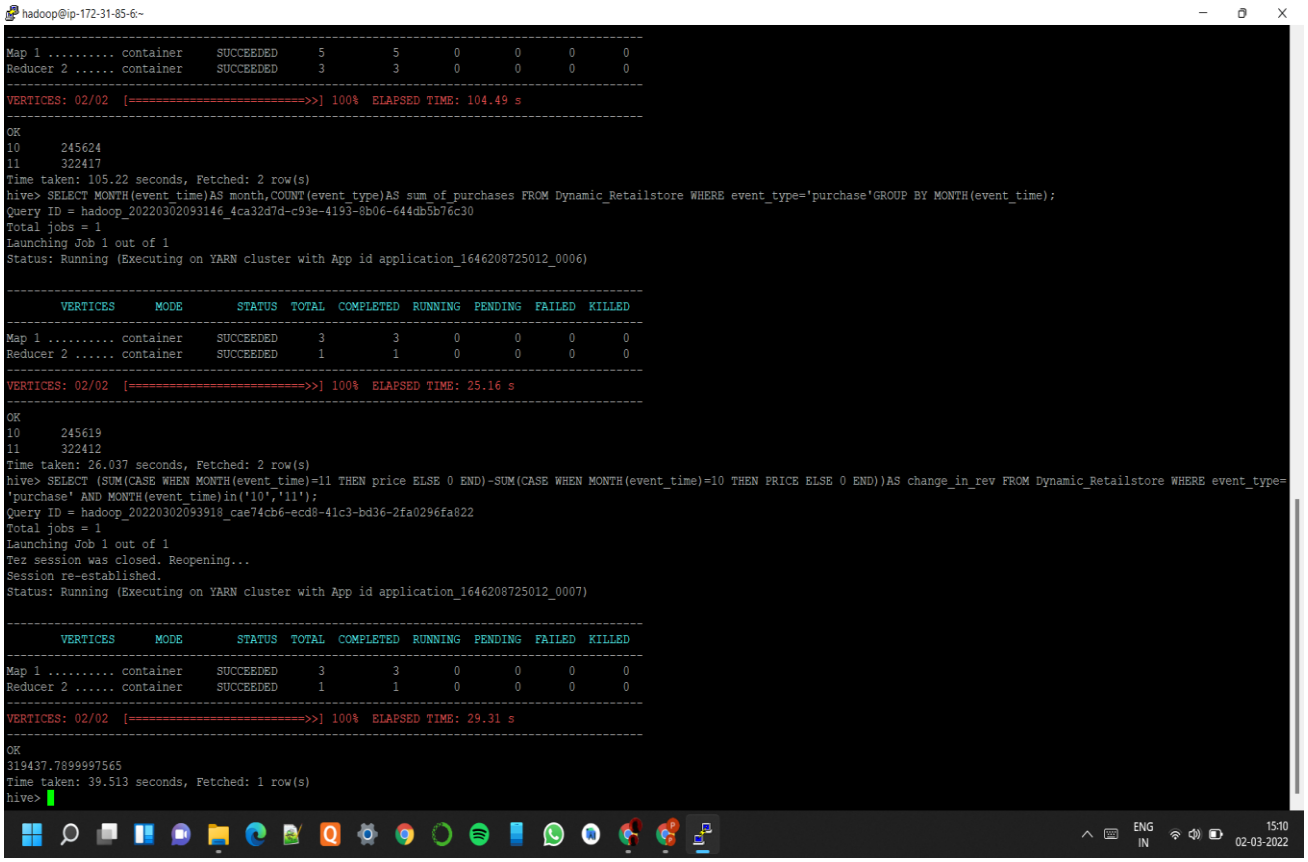
1. Sum of purchases made in the month of October is 245619 and in the month of November 322412, which means number of purchases are increased in November month

2. With proper partitioning and bucketing on table we can reduce execution time.

Using Optimized table from below questions onwards:

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

SELECT (SUM(CASE WHEN MONTH(event_time)=11 THEN price ELSE 0 END) - SUM(CASE WHEN MONTH(event_time)=10 THEN PRICE ELSE 0 END)) AS change_in_rev FROM Dynamic_Retailstore WHERE event_type = 'purchase' AND MONTH(event_time) in ('10','11');



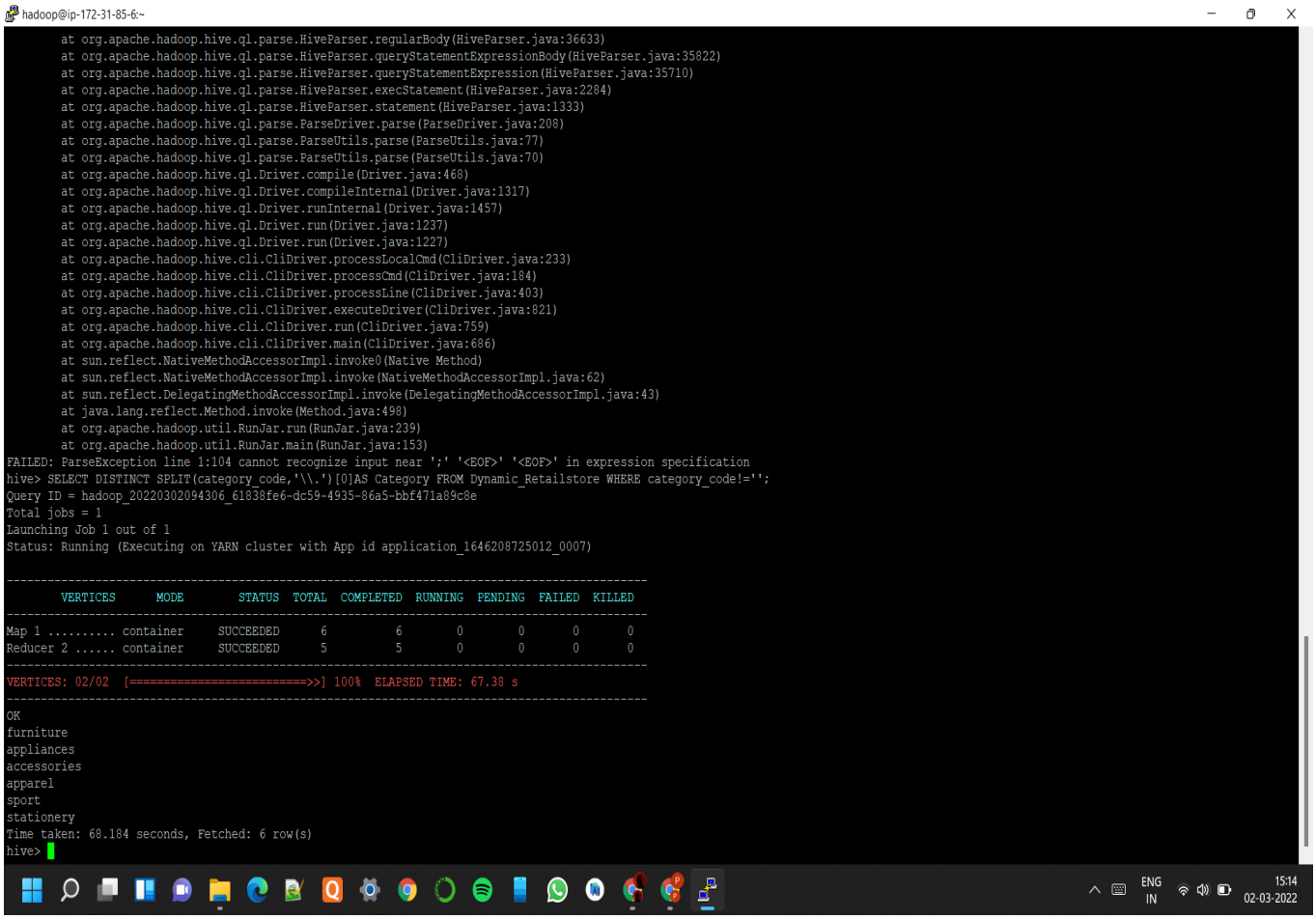
Insights:

1. Time taken to execute the query is 39.513 seconds

2. Revenue increased in November by 319437.789 from October

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

SELECT DISTINCT SPLIT(category_code,'\\.')[0] AS Category FROM dynpart_buck_retailsstore WHERE category_code != '';



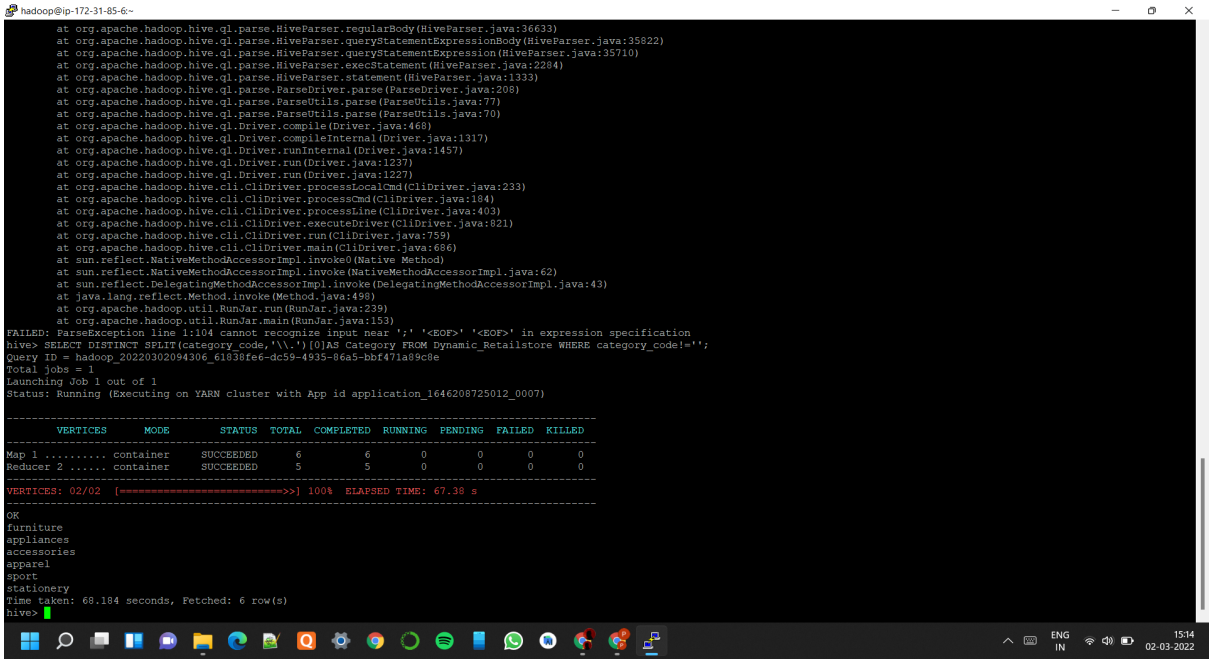
Insights:

1. Time taken to execute the query is 68.184 seconds

2. Total we got 6 distinct categories are – furniture, appliances, accessories, apparel, sport, stationary.

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

SELECT SPLIT(category_code,'\\.')[0] AS Category, COUNT(product_id) AS num_of_prod FROM Dynamic_Retailstore WHERE category_code != '' GROUP BY SPLIT(category_code,'\\.')[0] ORDER BY num_of_prod DESC;



Insights:

1. Time taken to execute the query is 68.111 seconds

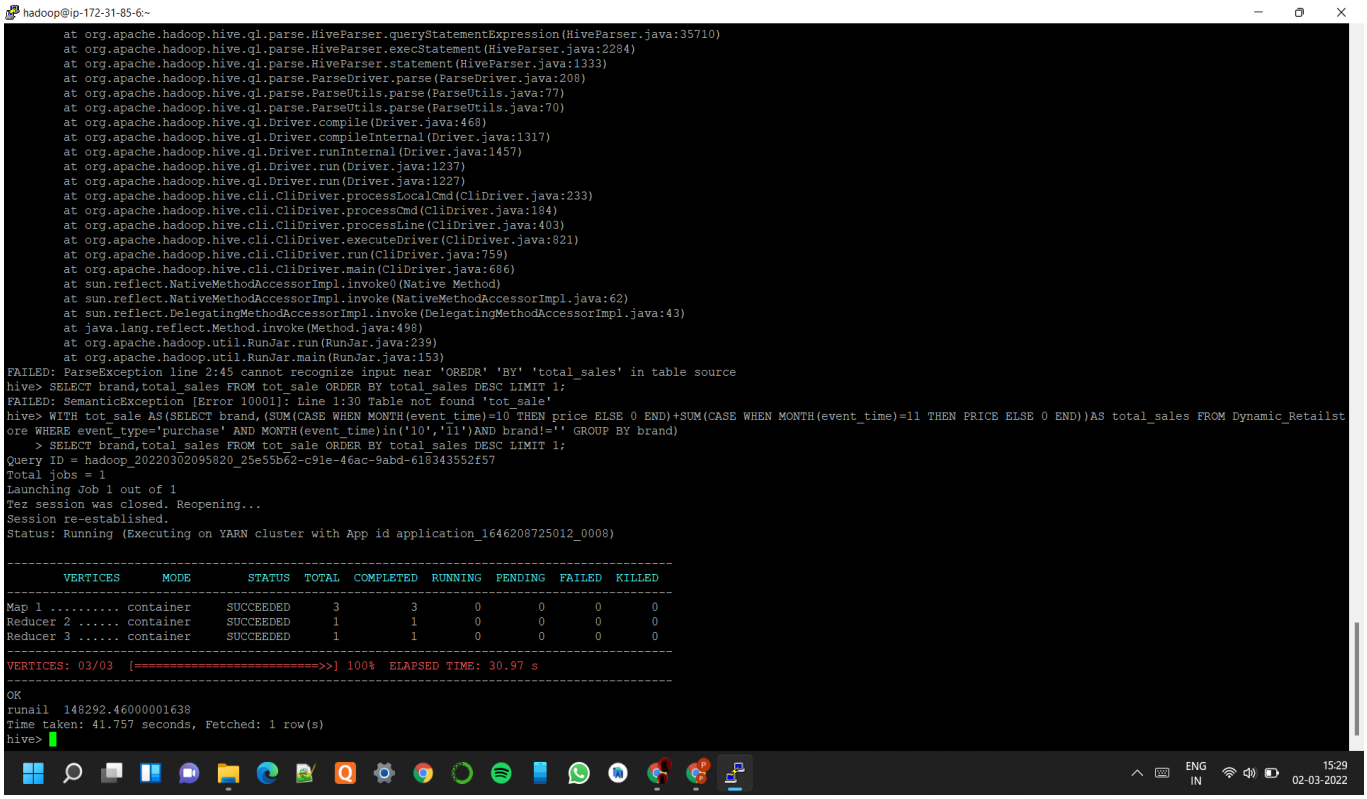
2. Appliances are having highest number of products available with 61736 compared to other categories.

3. Stationary and Furniture categories are almost equally registered with available ranges from 23000 to 27000.

4. Sports category is least available with 2 products

6.Which brand had the maximum sales in October and November combined?

WITH tot_sales AS(SELECT brand, (SUM(CASE WHEN MONTH(event_time)=10 THEN price ELSE 0 END) + SUM(CASE WHEN MONTH(event_time)=11 THEN PRICE ELSE 0 END)) AS total_sales FROM Dynamic_Retailstore WHERE event_type = 'purchase' AND MONTH(event_time) in ('10','11') AND brand != '' GROUP BY brand) SELECT brand, total_sales FROM tot_sales ORDER BY total_sales DESC LIMIT 1;

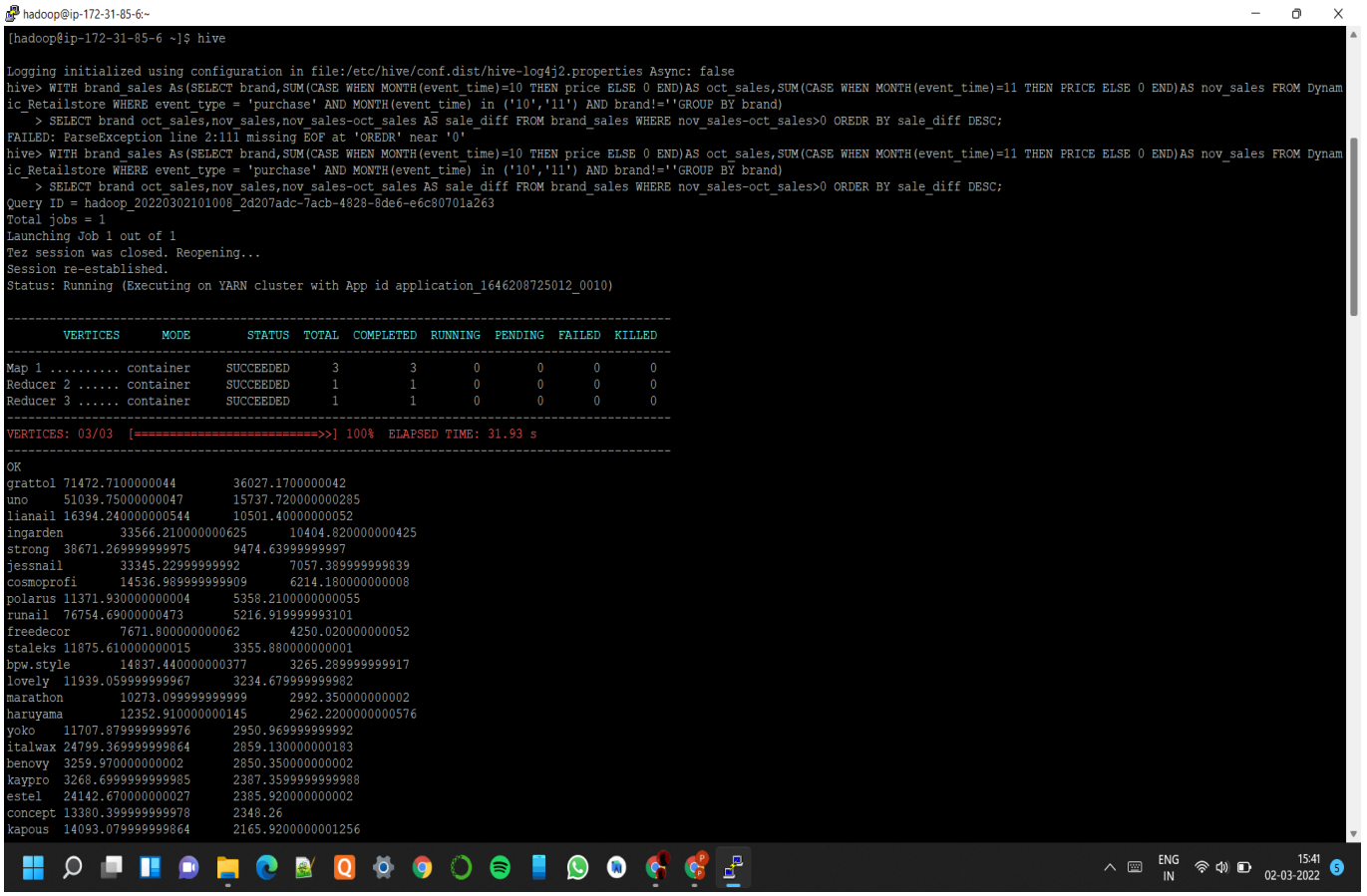


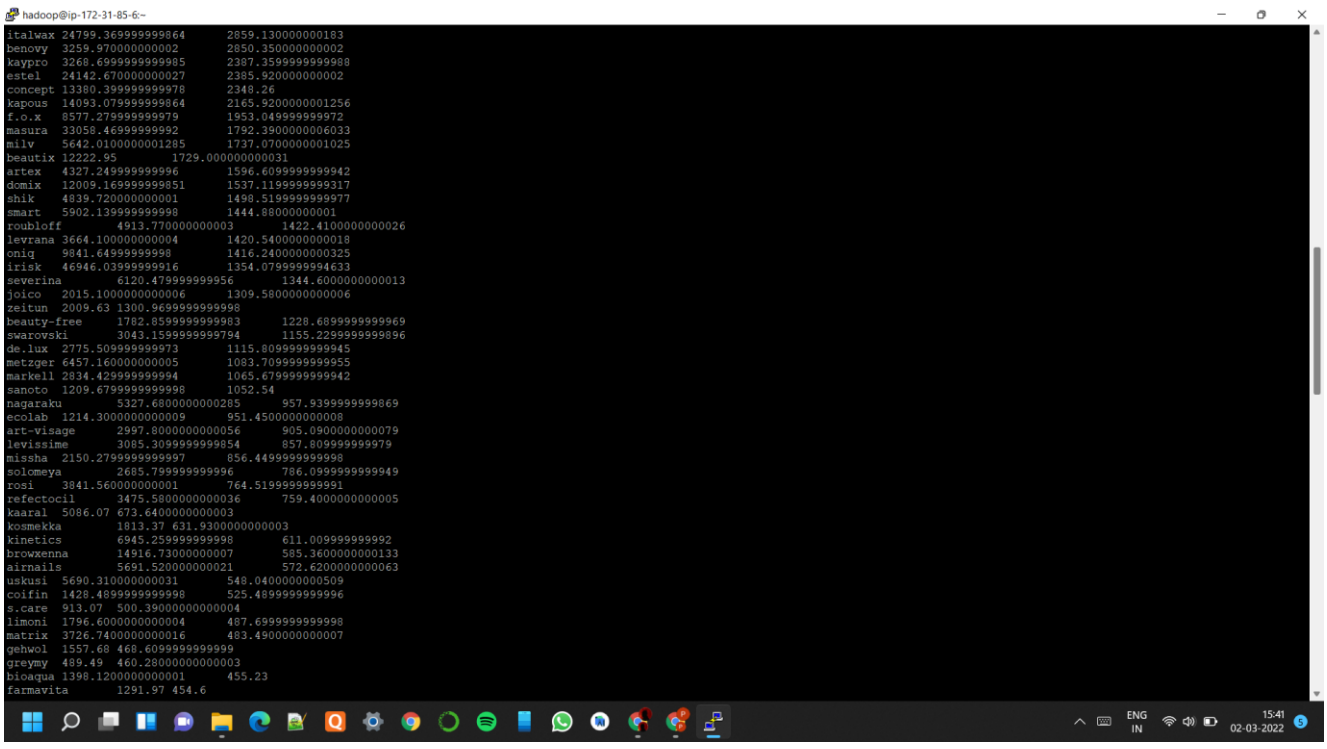
Insights:

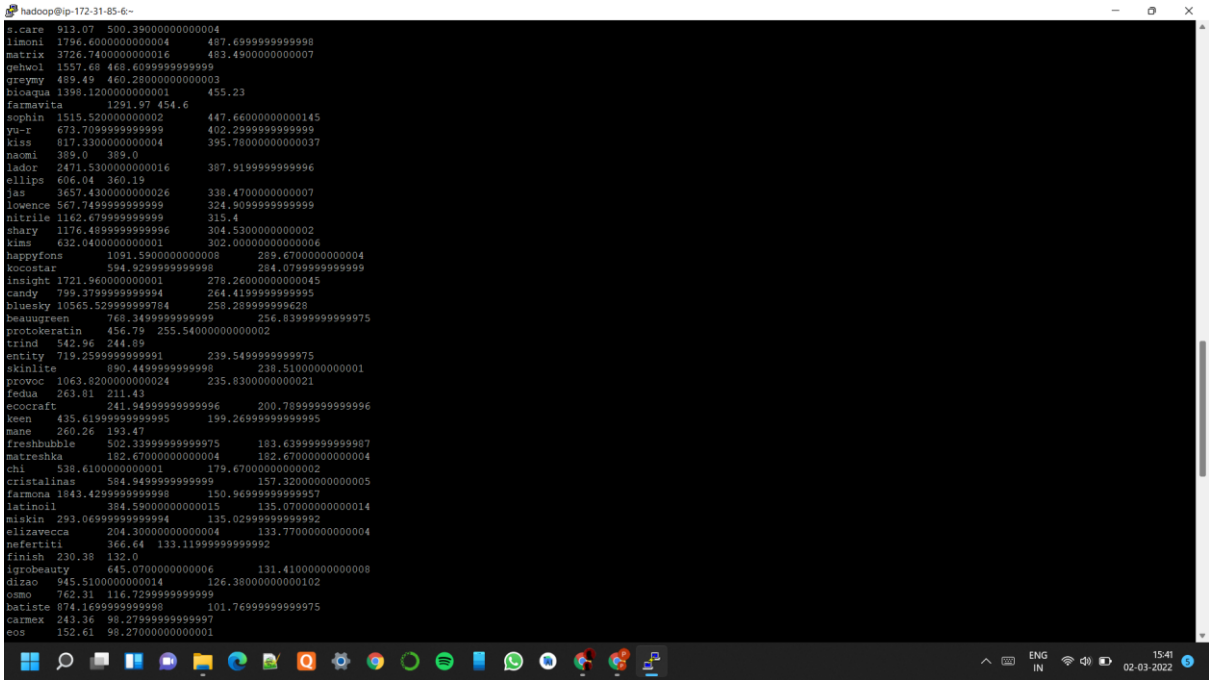
1. Runail is the brand that has the highest sales in total of both the months October and November

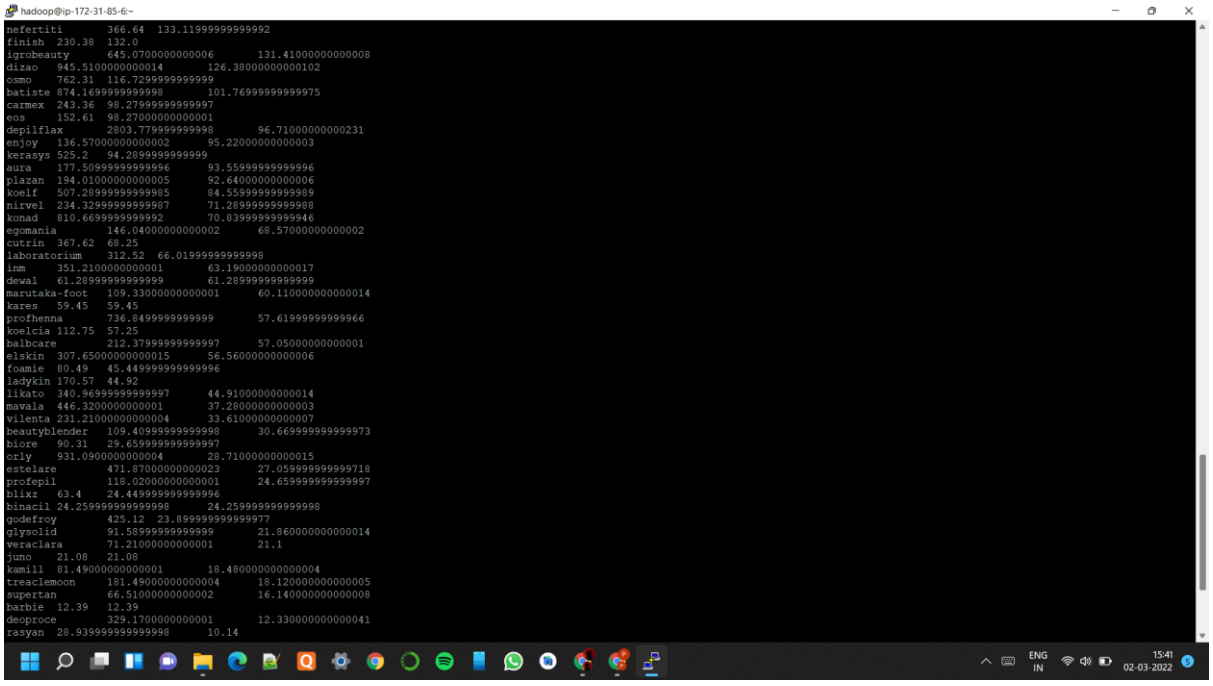
7. Which brands increased their sales from October to November?

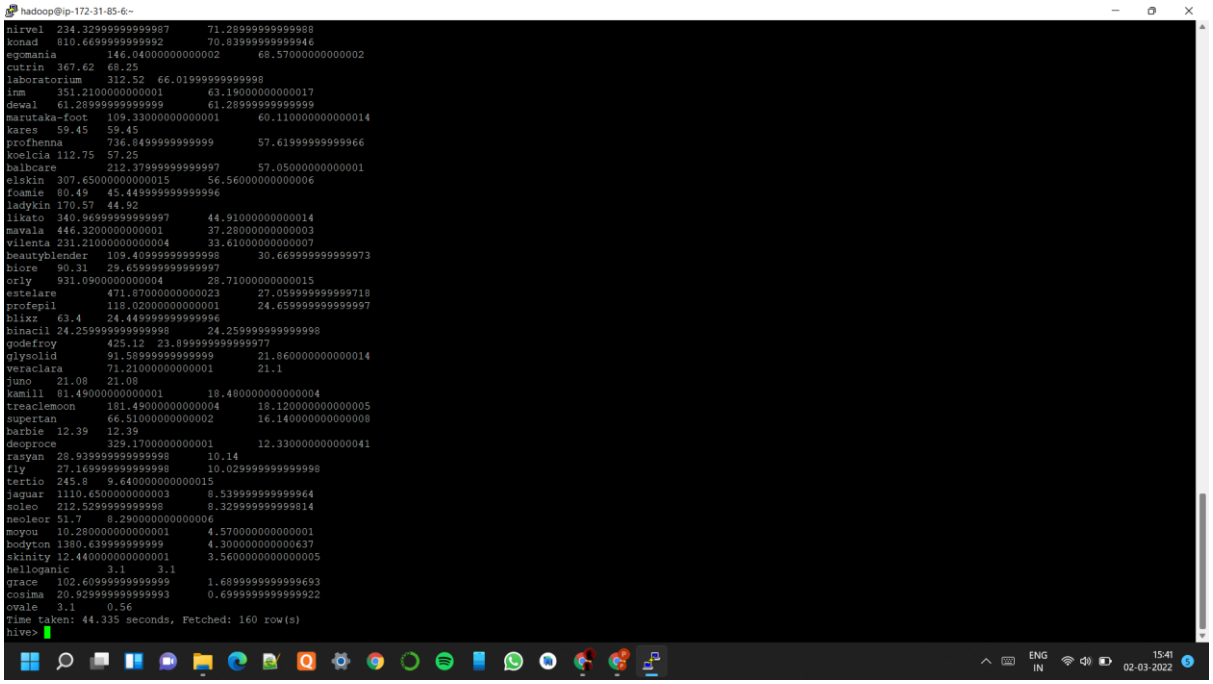
WITH brand_sales AS(SELECT brand, SUM(CASE WHEN MONTH(event_time)=10 THEN price ELSE 0 END) AS oct_sales, SUM(CASE WHEN MONTH(event_time)=11 THEN PRICE ELSE 0 END) AS nov_sales FROM Dynamic_Retailstore WHERE event_type = 'purchase' AND MONTH(event_time) in ('10','11') AND brand != '' GROUP BY brand) SELECT brand, oct_sales, nov_sales, nov_sales-oct_sales AS sale_diff FROM brand_sales WHERE nov_sales-oct_sales > 0 ORDER BY sale_diff DESC;


hadoop@ip-172-31-85-6:~
[hadoop@ip-172-31-85-6 ~]$ hive
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
hive> WITH brand_sales As(SELECT brand,SUM(CASE WHEN MONTH(event_time)=10 THEN price ELSE 0 END)AS oct_sales,SUM(CASE WHEN MONTH(event_time)=11 THEN PRICE ELSE 0 END)AS nov_sales FROM Dynamic_Retailstore WHERE event_type = 'purchase' AND MONTH(event_time) in ('10','11') AND brand!=''GROUP BY brand)
> SELECT brand oct_sales,nov_sales,nov_sales-oct_sales AS sale_diff FROM brand_sales WHERE nov_sales-oct_sales>0 OREDR BY sale_diff DESC;
FAILED: ParseException line 2:111 missing EOF at 'OREDR' near '0'
hive> WITH brand_sales As(SELECT brand,SUM(CASE WHEN MONTH(event_time)=10 THEN price ELSE 0 END)AS oct_sales,SUM(CASE WHEN MONTH(event_time)=11 THEN PRICE ELSE 0 END)AS nov_sales FROM Dynamic_Retailstore WHERE event_type = 'purchase' AND MONTH(event_time) in ('10','11') AND brand!=''GROUP BY brand)
> SELECT brand oct_sales,nov_sales,nov_sales-oct_sales AS sale_diff FROM brand_sales WHERE nov_sales-oct_sales>0 ORDER BY sale_diff DESC;
Query ID = hadoop_20220302101008_2d207adc-7acb-4828-8de6-e6c80701a263
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_1646208725012_0010)
VERTICES MODE STATUS TOTAL COMPLETED RUNNING PENDING FAILED KILLED
Map 1 ..... container SUCCEEDED 3 3 0 0 0 0
Reducer 2 ..... container SUCCEEDED 1 1 0 0 0 0
Reducer 3 ..... container SUCCEEDED 1 1 0 0 0 0
VERTICES: 03/03 [==========>>] 100% ELAPSED TIME: 31.93 s
OK
grattol 71472.7100000044 36027.1700000042
uno 51039.75000000047 15737.720000000285
lianail 16394.240000000544 10501.40000000052
ingarden 33566.210000000625 10404.820000000425
strong 38671.269999999975 9474.63999999997
jessnail 33345.22999999992 7057.389999999839
cosmoprofi 14536.989999999909 6214.180000000008
polarus 11371.930000000004 5358.2100000000055
runail 76754.69000000473 5216.919999993101
freedecor 7671.800000000062 4250.020000000052
staleks 11875.610000000015 3355.880000000001
bpw.style 14837.440000000377 3265.289999999917
lovely 11939.059999999967 3234.679999999982
marathon 10273.099999999999 2992.350000000002
haruyama 12352.910000000145 2962.2200000000576
yoko 11707.879999999976 2950.969999999992
italwax 24799.369999999864 2859.130000000183
benovy 3259.970000000002 2850.350000000002
kaypro 3268.6999999999985 2387.3599999999988
estel 24142.670000000027 2385.920000000002
concept 13380.399999999978 2348.26
kapous 14093.079999999864 2165.9200000001256


hadoop@ip-172-31-85-6:~
italwax 24799.369999999864 2859.130000000183
benovy 3259.970000000002 2850.350000000002
kaypro 3268.6999999999985 2387.3599999999988
estel 24142.670000000027 2385.920000000002
concept 13380.399999999978 2348.26
kapous 14093.079999999864 2165.9200000001256
f.o.x 8577.279999999979 1953.049999999972
masura 33058.46999999992 1792.3900000006033
milv 5642.0100000001285 1737.0700000001025
beautix 12222.95 1729.000000000031
artex 4327.249999999996 1596.6099999999942
domix 12009.169999999851 1537.1199999999317
shik 4839.720000000001 1498.5199999999977
smart 5902.139999999998 1444.88000000001
roubloff 4913.770000000003 1422.4100000000026
levrana 3664.100000000004 1420.5400000000018
oniq 9841.64999999998 1416.2400000000325
irisk 46946.03999999916 1354.0799999994633
severina 6120.479999999956 1344.6000000000013
joico 2015.1000000000006 1309.5800000000006
zeitun 2009.63 1300.9699999999998
beauty-free 1782.8599999999983 1228.6899999999969
swarovski 3043.1599999999794 1155.2299999999896
de.lux 2775.509999999973 1115.8099999999945
metzger 6457.160000000005 1083.7099999999955
markell 2834.429999999994 1065.6799999999942
sanoto 1209.6799999999998 1052.54
nagaraku 5327.6800000000285 957.9399999999869
ecolab 1214.3000000000009 951.4500000000008
art-visage 2997.8000000000056 905.0900000000079
levissime 3085.3099999999854 857.8099999999979
missha 2150.2799999999997 856.4499999999998
solomeya 2685.799999999996 786.0999999999949
rosi 3841.560000000001 764.5199999999991
refectocil 3475.5800000000036 759.4000000000005
kaaral 5086.07 673.6400000000003
kosmekka 1813.37 631.9300000000003
kinetics 6945.259999999998 611.009999999992
browxenna 14916.730000000007 585.3600000000133
airnails 5691.520000000021 572.6200000000063
uskusi 5690.310000000031 548.0400000000509
coifin 1428.4899999999998 525.4899999999996
s.care 913.07 500.39000000000004
limoni 1796.6000000000004 487.6999999999998
matrix 3726.7400000000016 483.4900000000007
gehwol 1557.68 468.6099999999999
greymy 489.49 460.28000000000003
bioaqua 1398.1200000000001 455.23
farmavita 1291.97 454.6


hadoop@ip-172-31-85-6:~
s.care 913.07 500.39000000000004
limoni 1796.6000000000004 487.6999999999998
matrix 3726.7400000000016 483.4900000000007
gehwol 1557.68 468.6099999999999
greymy 489.49 460.28000000000003
bioaqua 1398.1200000000001 455.23
farmavita 1291.97 454.6
sophin 1515.520000000002 447.66000000000145
yu-r 673.7099999999999 402.2999999999999
kiss 817.3300000000004 395.78000000000037
naomi 389.0 389.0
lador 2471.5300000000016 387.9199999999996
ellips 606.04 360.19
jas 3657.4300000000026 338.4700000000007
lowence 567.7499999999999 324.9099999999999
nitrile 1162.679999999999 315.4
shary 1176.4899999999996 304.5300000000002
kims 632.0400000000001 302.0000000000006
happyfons 1091.5900000000008 289.67000000000004
kocostar 594.9299999999998 284.0799999999999
insight 1721.9600000000001 278.26000000000045
candy 799.3799999999994 264.4199999999995
bluesky 10565.529999999784 258.289999999628
beauugreen 768.3499999999999 256.83999999999975
protokeratin 456.79 255.54000000000002
trind 542.96 244.89
entity 719.2599999999991 239.5499999999975
skinlite 890.4499999999998 238.51000000000001
provoc 1063.8200000000024 235.8300000000021
fedua 263.81 211.43
ecocraft 241.94999999999996 200.78999999999996
keen 435.61999999999995 199.26999999999995
mane 260.26 193.47
freshbubble 502.33999999999975 183.63999999999987
matreshka 182.67000000000004 182.67000000000004
chi 538.6100000000001 179.67000000000002
cristalinas 584.9499999999999 157.32000000000005
farmona 1843.4299999999998 150.96999999999957
latinoil 384.59000000000015 135.07000000000014
miskin 293.06999999999994 135.02999999999992
elizavecca 204.30000000000004 133.77000000000004
nefertiti 366.64 133.11999999999992
finish 230.38 132.0
igrobeauty 645.0700000000006 131.41000000000008
dizao 945.5100000000014 126.38000000000102
osmo 762.31 116.72999999999999
batiste 874.1699999999998 101.76999999999975
carmex 243.36 98.27999999999997
eos 152.61 98.27000000000001


hadoop@ip-172-31-85-6:~
nefertiti 366.64 133.11999999999992
finish 230.38 132.0
igrobeauty 645.0700000000006 131.41000000000008
dizao 945.5100000000014 126.38000000000102
osmo 762.31 116.72999999999999
batiste 874.1699999999998 101.76999999999975
carmex 243.36 98.27999999999997
eos 152.61 98.27000000000001
depilflax 2803.779999999998 96.71000000000231
enjoy 136.57000000000002 95.22000000000003
kerasys 525.2 94.2899999999999
aura 177.50999999999996 93.55999999999996
plazan 194.01000000000005 92.64000000000006
koelf 507.28999999999985 84.55999999999989
nirvel 234.32999999999987 71.28999999999988
konad 810.6699999999992 70.83999999999946
egomania 146.04000000000002 68.57000000000002
cutrin 367.62 68.25
laboratorium 312.52 66.01999999999998
inm 351.2100000000001 63.19000000000017
dewal 61.28999999999999 61.28999999999999
marutaka-foot 109.33000000000001 60.110000000000014
kares 59.45 59.45
profhenna 736.8499999999999 57.61999999999966
koelcia 112.75 57.25
balbcare 212.37999999999997 57.05000000000001
elskin 307.65000000000015 56.56000000000006
foamie 80.49 45.44999999999996
ladykin 170.57 44.92
likato 340.96999999999997 44.91000000000014
mavala 446.3200000000001 37.28000000000003
vilenta 231.21000000000004 33.61000000000007
beautyblender 109.40999999999998 30.669999999999973
biore 90.31 29.659999999999997
orly 931.0900000000004 28.71000000000015
estelare 471.87000000000023 27.059999999999718
profepil 118.02000000000001 24.659999999999997
blixz 63.4 24.449999999999996
binacil 24.259999999999998 24.259999999999998
godefroy 425.12 23.899999999999977
glysolid 91.58999999999999 21.860000000000014
veraclara 71.21000000000001 21.1
juno 21.08 21.08
kamill 81.49000000000001 18.480000000000004
treaclemoon 181.49000000000004 18.120000000000005
supertan 66.51000000000002 16.140000000000008
barbie 12.39 12.39
deoproce 329.1700000000001 12.330000000000041
rasyan 28.939999999999998 10.14



Insights:

1. Here are some 160 brands with increment in the selling from October to November.

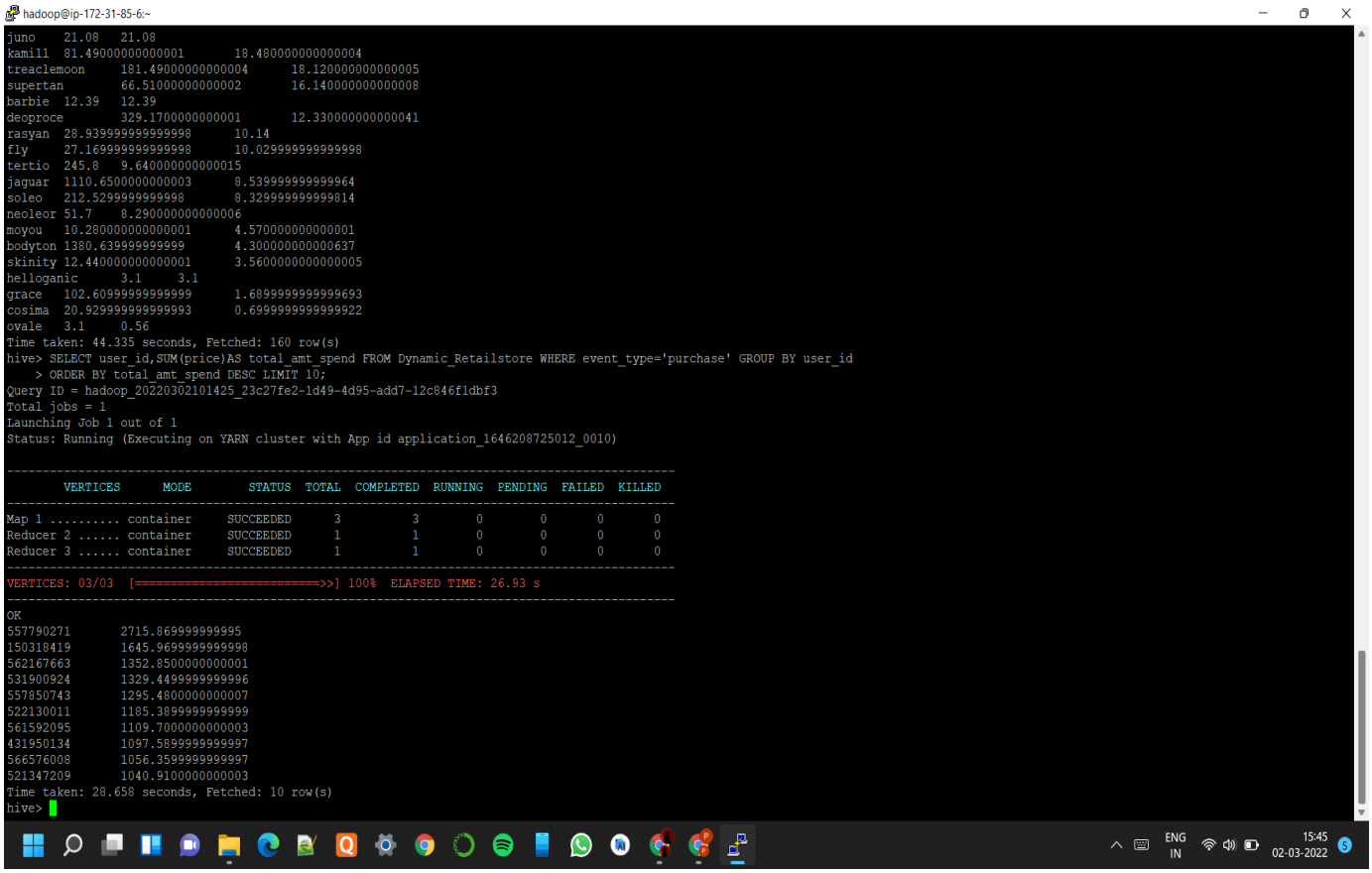
2. ‘Grattol’ brand has the highest total increment i.e., 36,027 /- and ‘Ovale’ seems to have the least increment of 0.56 /- from October to November.

3. Among all these brands lists, ‘Runail’ which was the best brand in terms of selling in October and November combined is also in the top 10 brands with high increment for October (71539.28) to November (76758.61) i.e., increment of total 5219.38.

4. This implies that ‘Runail’ is the best and popular brand among all other brands within people.

8. Your company wants to reward the top 10 users of its websites with a golden customer plan. Write a query to generate a list of top 10 users who spend the most.

SELECT user_id, SUM(price) AS total_amt_spend FROM Dynamic_Retailstore WHERE event_type = 'purchase' GROUP BY user_id ORDER BY total_amt_spend DESC LIMIT 10;



Insights:

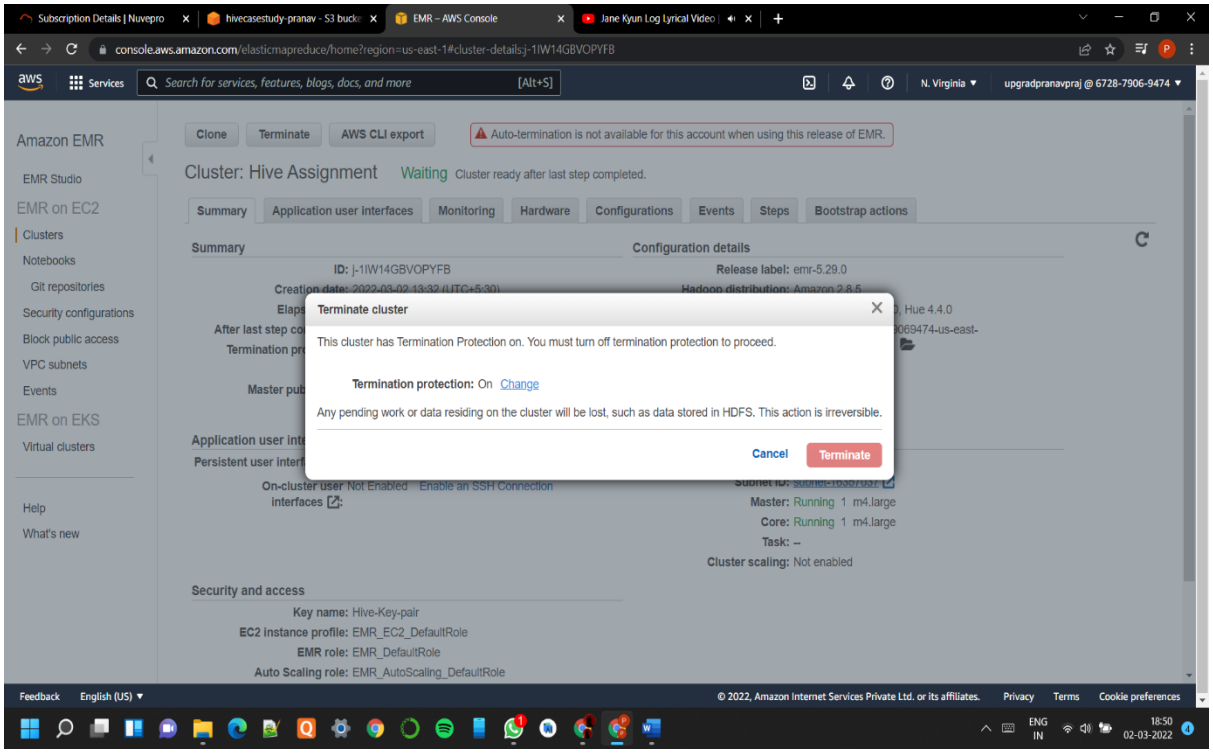
1. Here is the list of the top 10 users or buyers who have spent the most and could be rewarded with a Golden Customer plan to attract more people in the coming future.

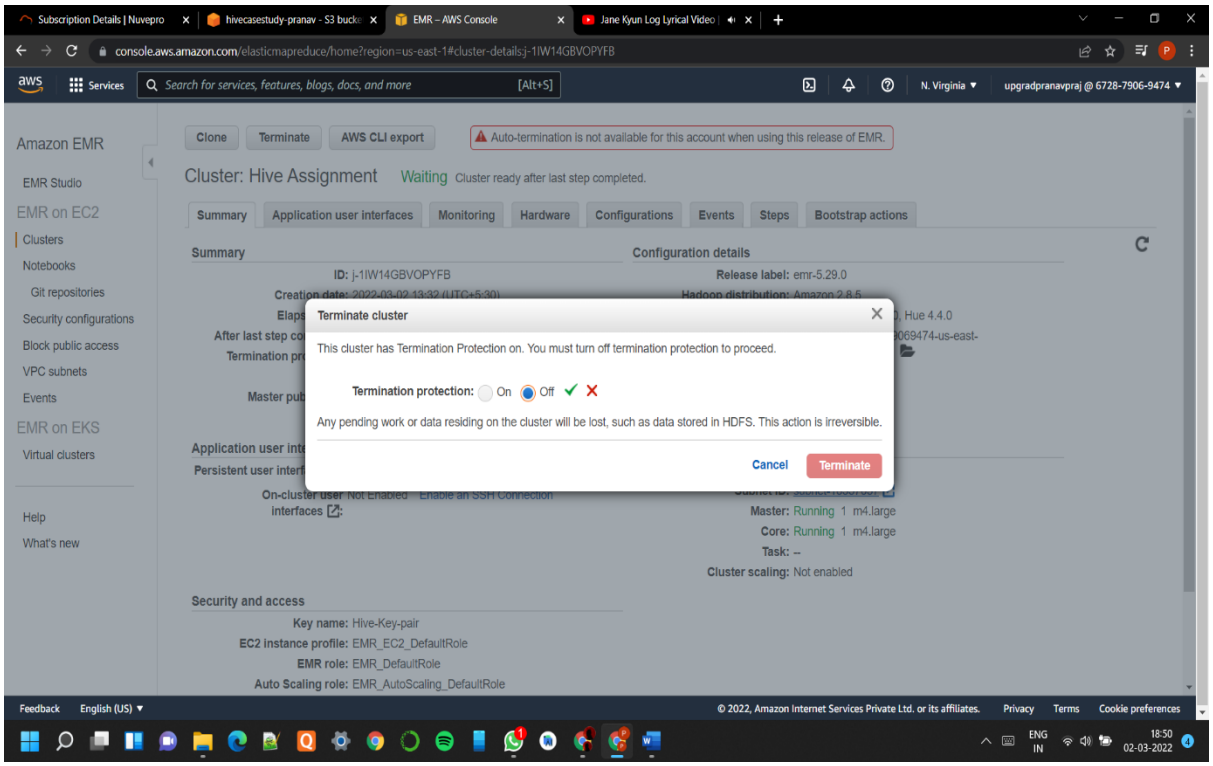
2. With the Optimized table the execution time reduced with proper partitioning and bucketing.

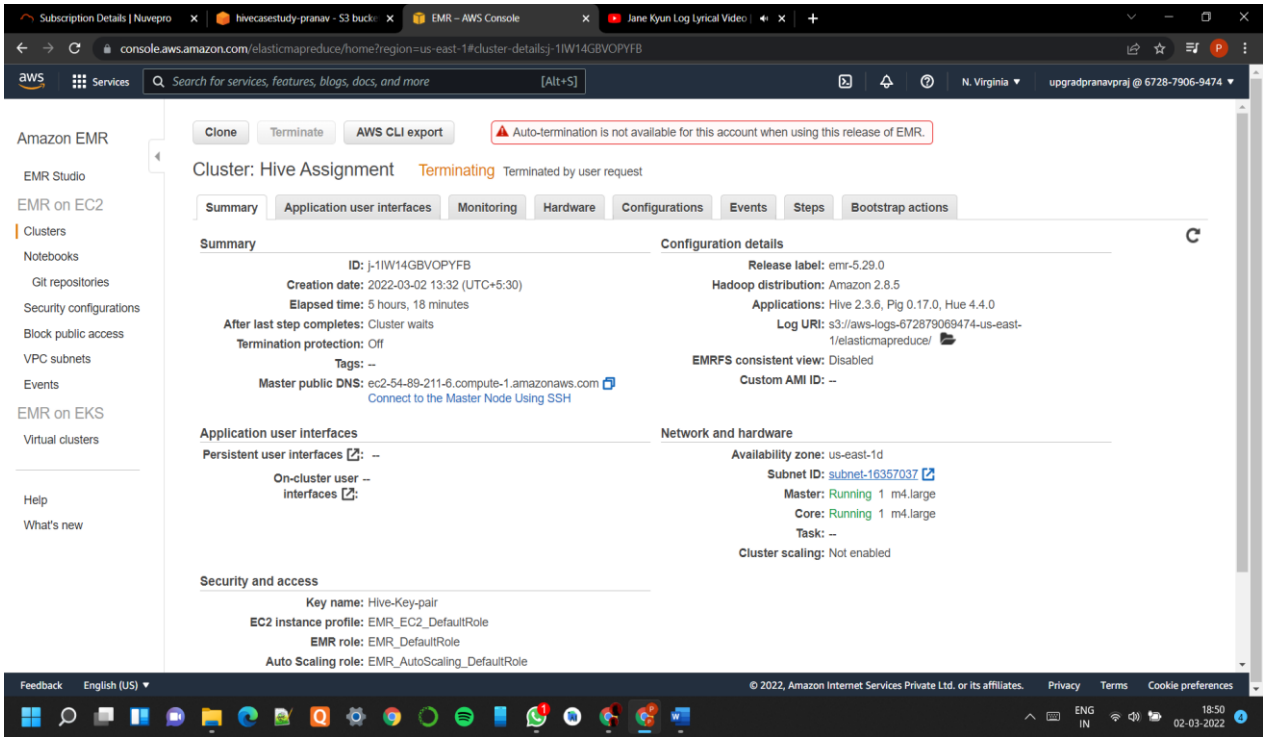
3. Time taken to execute this query on optimized table is 28.658 seconds.

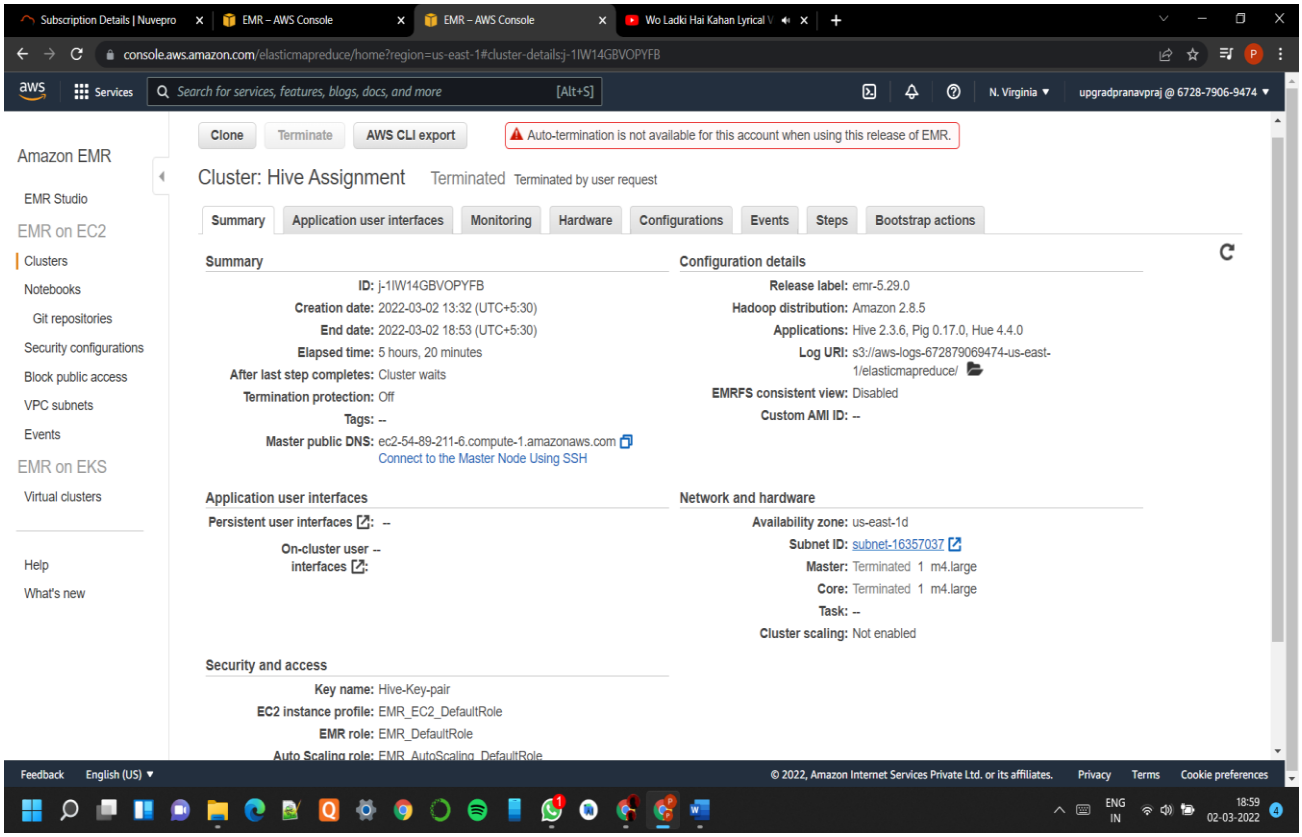
TERMINATION PROCESS:

After completing our analysis, we should terminate the EMR cluster


Subscription Details | Nuvepro
hivecasestudy-pranav - S3 buck
EMR – AWS Console
Jane Kyun Log Lyrical Video
console.aws.amazon.com/elasticmapreduce/home?region=us-east-1#cluster-details:j-1IW14GBVOPYFB
Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia
upgradpranavpraj @ 6728-7906-9474
Amazon EMR
EMR Studio
EMR on EC2
Clusters
Notebooks
Git repositories
Security configurations
Block public access
VPC subnets
Events
EMR on EKS
Virtual clusters
Help
What's new
Clone
Terminate
AWS CLI export
Auto-termination is not available for this account when using this release of EMR.
Cluster: Hive Assignment
Waiting
Cluster ready after last step completed.
Summary
Application user interfaces
Monitoring
Hardware
Configurations
Events
Steps
Bootstrap actions
Summary
ID: j-1IW14GBVOPYFB
Configuration details
Release label: emr-5.29.0
Terminate cluster
This cluster has Termination Protection on. You must turn off termination protection to proceed.
Termination protection: On Change
Any pending work or data residing on the cluster will be lost, such as data stored in HDFS. This action is irreversible.
Cancel
Terminate
Master: Running 1 m4.large
Core: Running 1 m4.large
Task: --
Cluster scaling: Not enabled
Security and access
Key name: Hive-Key-pair
EC2 instance profile: EMR_EC2_DefaultRole
EMR role: EMR_DefaultRole
Auto Scaling role: EMR_AutoScaling_DefaultRole
Feedback
English (US)
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences
ENG
IN
18:50
02-03-2022


Subscription Details | Nuvepro
hivecasestudy-pranav - S3 buck
EMR – AWS Console
Jane Kyun Log Lyrical Video
console.aws.amazon.com/elasticmapreduce/home?region=us-east-1#cluster-details:j-1IW14GBVOPYFB
Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia
upgradpranavpraj @ 6728-7906-9474
Amazon EMR
EMR Studio
EMR on EC2
Clusters
Notebooks
Git repositories
Security configurations
Block public access
VPC subnets
Events
EMR on EKS
Virtual clusters
Help
What's new
Clone
Terminate
AWS CLI export
Auto-termination is not available for this account when using this release of EMR.
Cluster: Hive Assignment
Waiting
Cluster ready after last step completed.
Summary
Application user interfaces
Monitoring
Hardware
Configurations
Events
Steps
Bootstrap actions
Summary
ID: j-1IW14GBVOPYFB
Configuration details
Release label: emr-5.29.0
Terminate cluster
This cluster has Termination Protection on. You must turn off termination protection to proceed.
Termination protection: On Off
Any pending work or data residing on the cluster will be lost, such as data stored in HDFS. This action is irreversible.
Cancel
Terminate
Master: Running 1 m4.large
Core: Running 1 m4.large
Task: --
Cluster scaling: Not enabled
Security and access
Key name: Hive-Key-pair
EC2 instance profile: EMR_EC2_DefaultRole
EMR role: EMR_DefaultRole
Auto Scaling role: EMR_AutoScaling_DefaultRole
Feedback
English (US)
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences
ENG
IN
18:50
02-03-2022





Cluster terminated!!