HIVE CASE STUDY

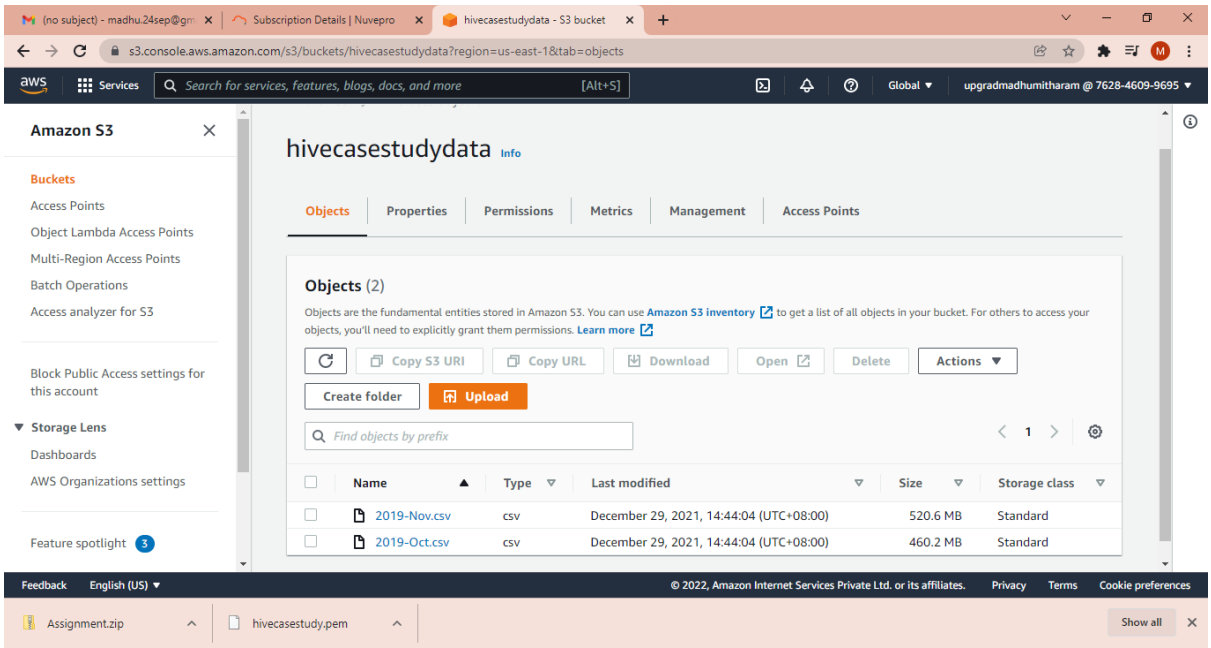
Problem Statement:

For this assignment, you will be working with a public clickstream dataset of a cosmetics store. Using this dataset, your job is to extract valuable insights which generally data engineers come up within an e-retail company.

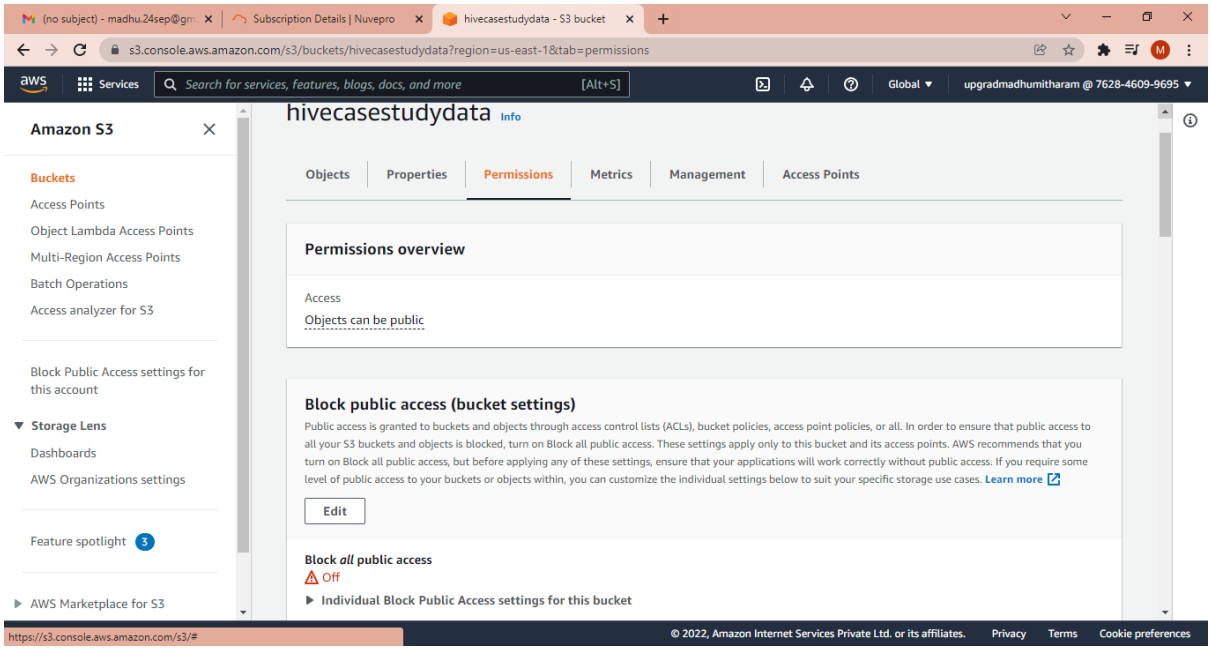
Solution:

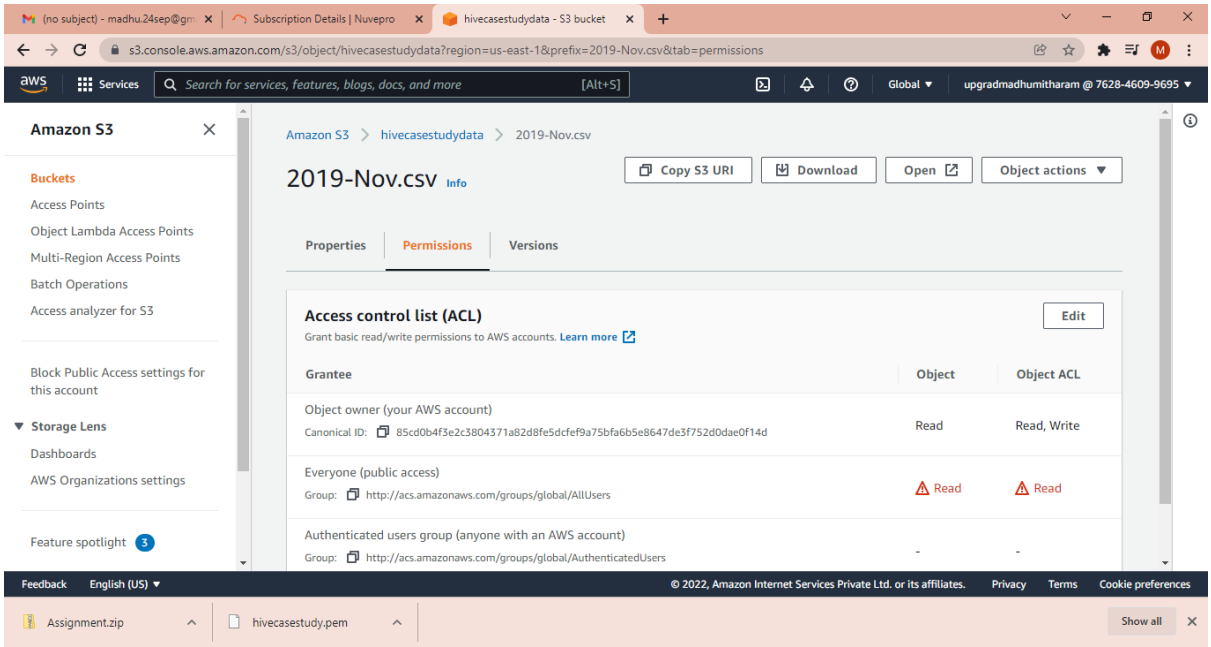
Step 1: Importing the data from S3 to HDFS

In S3 created a bucket and uploaded the 2019-Nov.csv and 2019-Oct.csv data into the bucket.



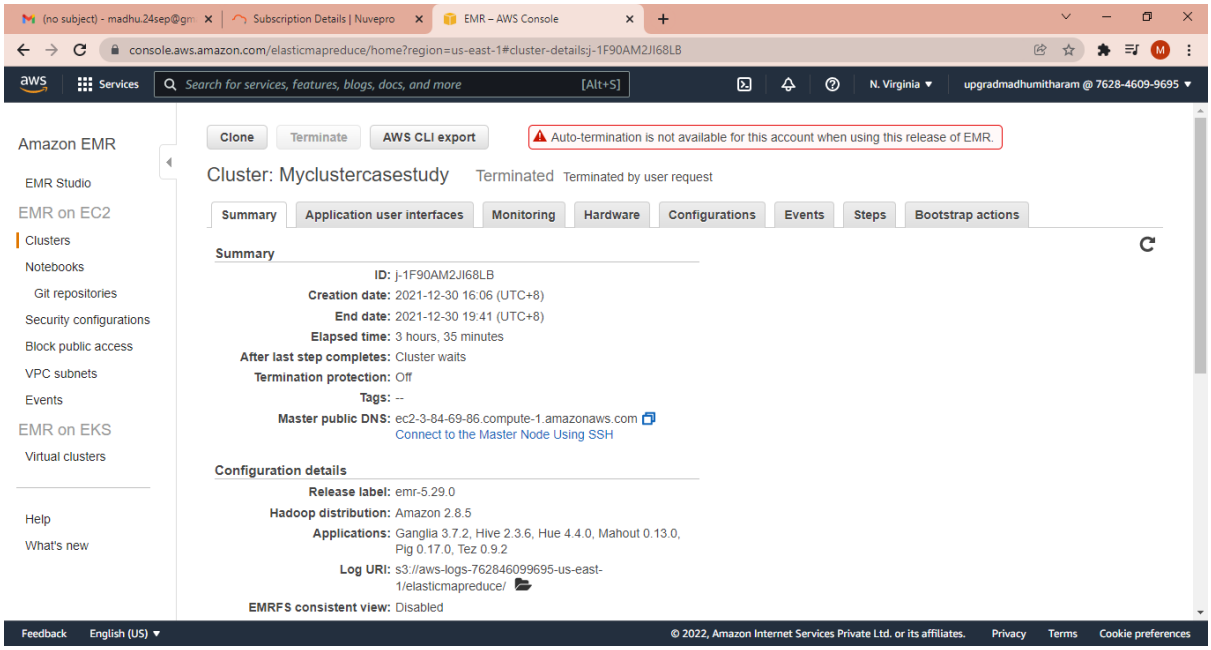
Provided the public access to bucket and individual files.





Create a new cluster (2-node with both the master and core nodes as **M4.large**)on EMR with a new keyvaluepair(hivecasestudy.ppk) created from EC2.

And have used **emr-5.29.0 release for this case study**.



On connecting to the Hadoop using putty and import the data to HDFS from S3 using below commands

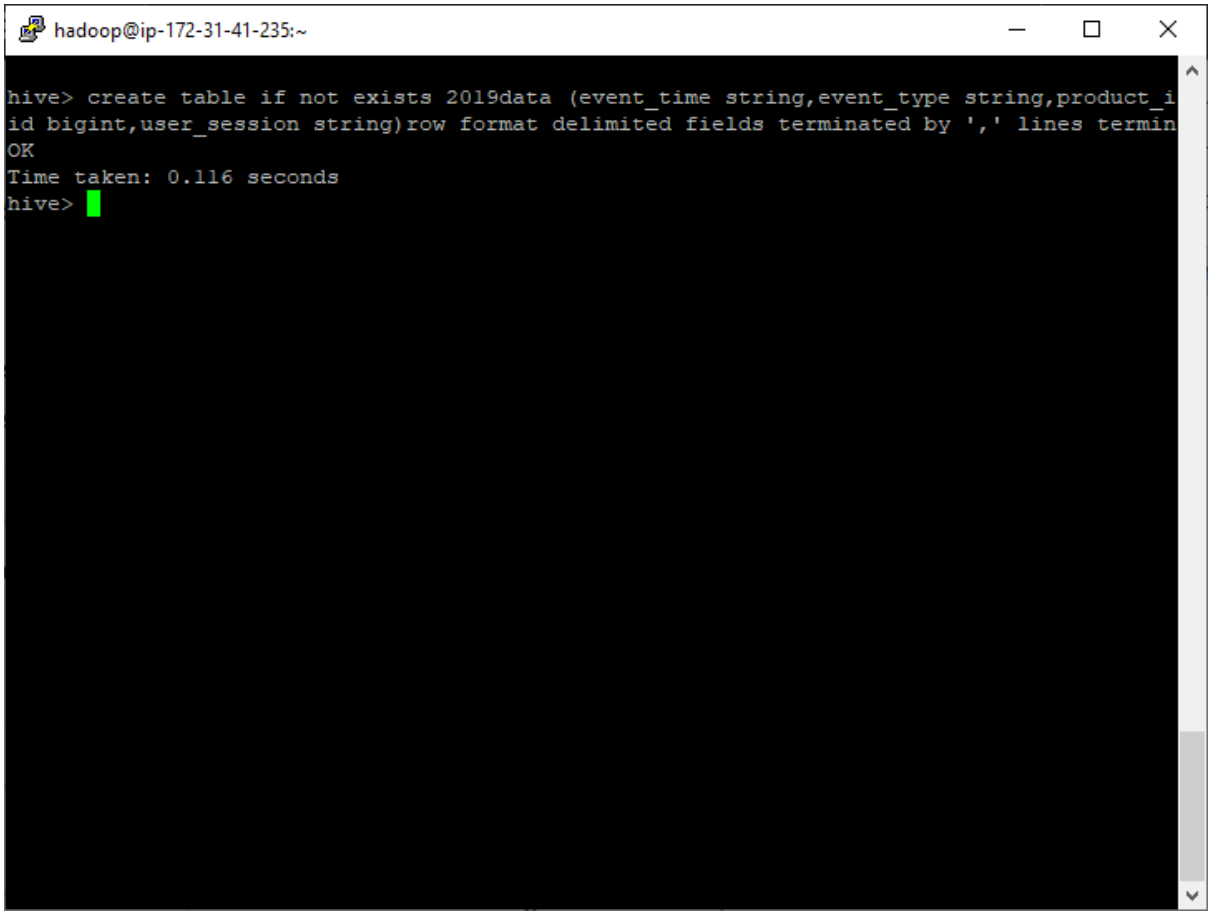
```
[hadoop@ip-172-31-41-235 ~]$ aws s3 cp s3://hivecasestudydata/2019-Nov.csv .
```

```
[hadoop@ip-172-31-41-235 ~]$ aws s3 cp s3://hivecasestudydata/2019-Oct.csv .
```

Step 2:Database and table creation

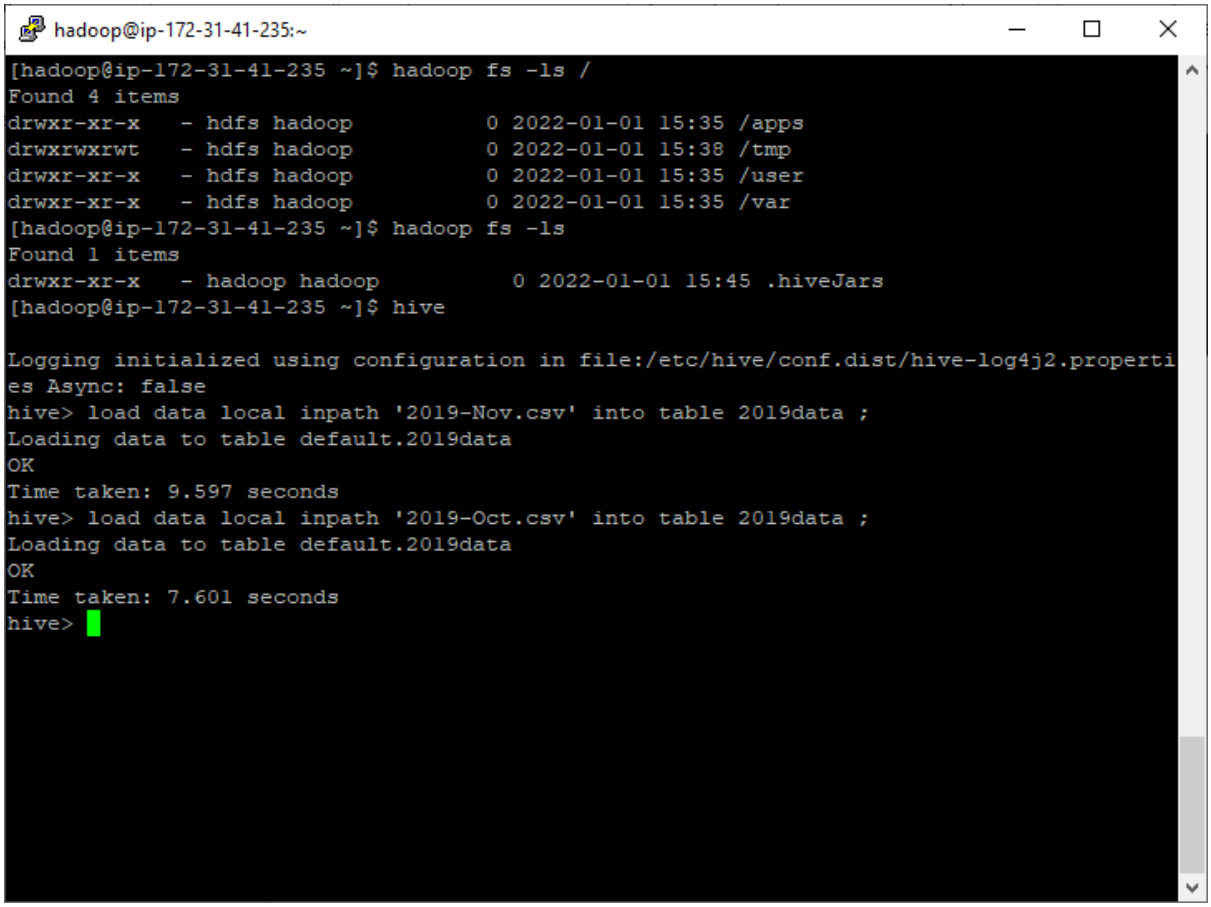
Created the hive schema /tables for the dataset(both Oct and Nov combined) using below commands.

```
hive>create table if not exists 2019data (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");
```



Hive>load data local inpath '2019-Nov.csv' into table 2019data ;

Hive>load data local inpath '2019-Oct.csv' into table 2019data ;



Partitioned table for the data with the event_type as purchase

Created a new partitioned table for the event_type as purchase using below query for optimization.

create table if not exists purchasedata(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");

insert into table purchasedata select event_time , event_type, product_id, category_id,category_code,brand,price,user_id,user_session from 2019data where event_type='purchase';

hadoop@ip-172-31-41-235:~

```
[hadoop@ip-172-31-41-235 ~]$ hadoop fs -ls /
Found 4 items
drwxr-xr-x   - hdfs hadoop          0 2022-01-01 15:35 /apps
drwxrwxrwt   - hdfs hadoop          0 2022-01-01 15:38 /tmp
drwxr-xr-x   - hdfs hadoop          0 2022-01-01 15:35 /user
drwxr-xr-x   - hdfs hadoop          0 2022-01-01 15:35 /var
[hadoop@ip-172-31-41-235 ~]$ hadoop fs -ls
Found 1 items
drwxr-xr-x   - hadoop hadoop          0 2022-01-01 15:45 .hiveJars
[hadoop@ip-172-31-41-235 ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properti
es Async: false
hive> load data local inpath '2019-Nov.csv' into table 2019data ;
Loading data to table default.2019data
OK
Time taken: 9.597 seconds
hive> load data local inpath '2019-Oct.csv' into table 2019data ;
Loading data to table default.2019data
OK
Time taken: 7.601 seconds
hive> create table if not exists purchasedata(event_time string,event_type string,produc
t_id string, category_id string,category_code string,brand string,price float,user_id bi
gint,user_session string)row format delimited fields terminated by ',' lines terminated
by '\n' stored as textfile TBLPROPERTIES ("skip.header.line.count"="1");
OK
Time taken: 0.284 seconds
hive>
```

hadoop@ip-172-31-41-235:~

```
hive> insert into table purchasedata select event_time ,event_type,product_id,category_i
d,category_code,brand,price,user_id,user_session from 2019data where event_type='purchas
e';
Query ID = hadoop_20220101160058_b193bae5-ad6a-4f0e-824c-1065acfd9113
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_1641051379361_0003)

Map 1: 0/2
Map 1: 0/2
Map 1: 0(+1)/2
Map 1: 0(+2)/2
Map 1: 0(+2)/2
Map 1: 0(+2)/2
Map 1: 0(+2)/2
Map 1: 0(+2)/2
Map 1: 0(+2)/2
Map 1: 1(+1)/2
Map 1: 2/2
Loading data to table default.purchasedata
OK
Time taken: 39.532 seconds
hive>
```

Step 3:Query Execution

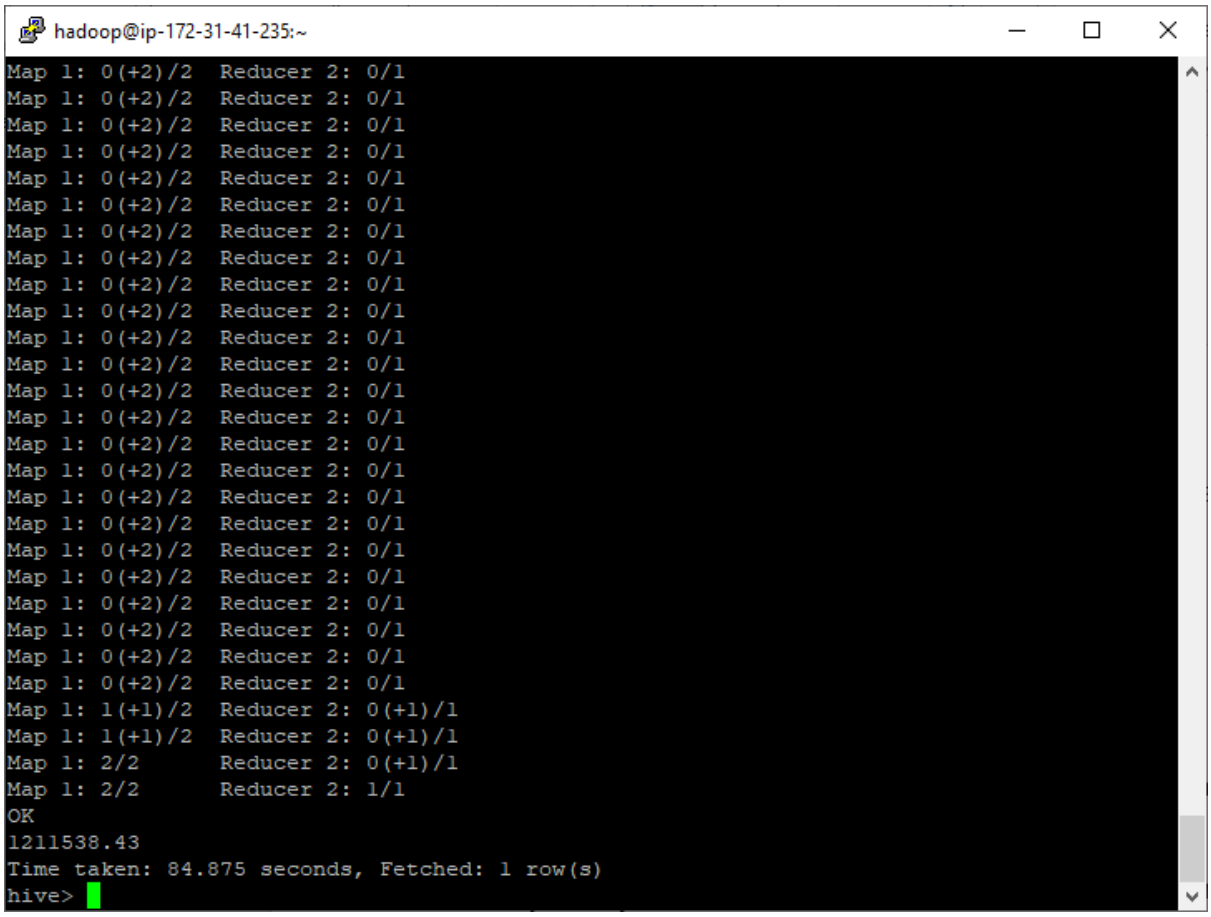
- **Find the total revenue generated due to purchases made in October.**

Ans: 1211538.43

Query:

SELECT round(sum(price),2) AS October_revenue from 2019data where date_format(event_time,'MM')=10 and event_type='purchase';

Output Screenshot:

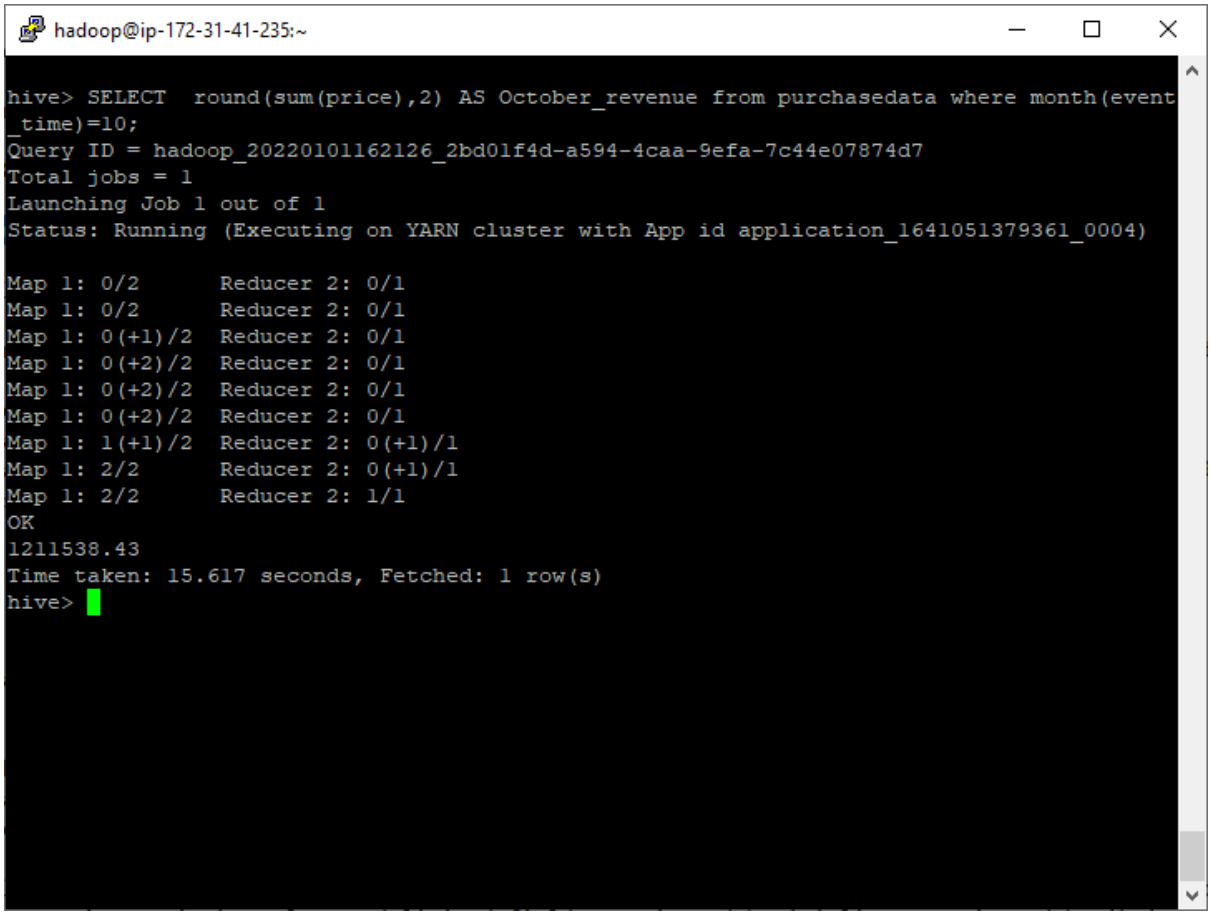


```
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 1(+1)/2  Reducer 2: 0(+1)/1
Map 1: 1(+1)/2  Reducer 2: 0(+1)/1
Map 1: 2/2      Reducer 2: 0(+1)/1
Map 1: 2/2      Reducer 2: 1/1
OK
1211538.43
Time taken: 84.875 seconds, Fetched: 1 row(s)
hive>
```

We can see that running the query on the entire data takes approx. 85 sec to obtained the result.Instead we can use the **partitioned tables on the purchase event_type for optimization** .And when run the same query on the partitioned purchasedata table we obtained the same result in approx. 16 sec.Hence it is advantageous to use the partitioned tables.

Query(optimized):

SELECT round(sum(price),2) AS October_revenue from purchasedata where month(event_time)=10;



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

Ans:

Revenue	month
1211538.43	10
1531016.9	11

Query:

SELECT round(sum(price),2) AS revenue,month(event_time) as month from purchasedata group by month(event_time);

Output Screenshot:

```
hadoop@ip-172-31-41-235:~
Map 1: 1(+1)/2  Reducer 2: 0(+1)/1
Map 1: 2/2      Reducer 2: 0(+1)/1
Map 1: 2/2      Reducer 2: 1/1
OK
1211538.43      10
1531016.9       11
Time taken: 25.421 seconds, Fetched: 2 row(s)
hive>
    > set hive.cli.print.header=true;
hive> SELECT round(sum(price),2) AS revenue,month(event_time) as month from purchasedata
 group by month(event_time);
Query ID = hadoop_20220101162917_c5318979-7032-4211-82a6-170c70ba9a54
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1641051379361_0005)

Map 1: 0/2      Reducer 2: 0/1
Map 1: 0/2      Reducer 2: 0/1
Map 1: 0(+1)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 1(+1)/2  Reducer 2: 0(+1)/1
Map 1: 2/2      Reducer 2: 0(+1)/1
Map 1: 2/2      Reducer 2: 1/1
OK
revenue month
1211538.43      10
1531016.9       11
Time taken: 17.934 seconds, Fetched: 2 row(s)
hive>
```

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

  Ans:319478.469592195

  Query:

with purchasediff as (

SELECT sum(case when date_format(event_time,'MM')=10 then price else 0 end) AS October,

sum(case when date_format(event_time,'MM')=11 then price else 0 end) AS November

FROM purchasedata) select November – October as revenue_change

from purchasediff;

Output Screenshot:

```
hadoop@ip-172-31-41-235:~
    >        sum(case when date_format(event_time,'MM')=11 then price else 0 end) AS Nov
ember
    >  FROM purchasedata) select November - October as revenue_change
    > from purchasediff;
Query ID = hadoop_20220101163550_bd0a239e-0aca-4e64-997d-2219406ea378
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_1641051379361_0006)

Map 1: 0/2      Reducer 2: 0/1
Map 1: 0/2      Reducer 2: 0/1
Map 1: 0(+1)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1
Map 1: 1(+1)/2  Reducer 2: 0/1
Map 1: 1(+1)/2  Reducer 2: 0(+1)/1
Map 1: 2/2      Reducer 2: 0(+1)/1
Map 1: 2/2      Reducer 2: 1/1
OK
revenue_change
319478.469592195
Time taken: 36.664 seconds, Fetched: 1 row(s)
hive>
[1]+  Stopped                 hive
[hadoop@ip-172-31-41-235 ~]$
```

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

Ans:

Category
Furniture
Appliances
Accessories
Apparel
Sport
Stationery

Query: select distinct(split(category_code,'\\.')[0]) as Category from 2019data where category_code<>"" and category_code is not Null;

Output Screenshot:

```
hadoop@ip-172-31-41-235:~
hive> set hive.cli.print.header=true;
hive> select distinct(split(category_code,'\\.')[0]) as Category from 2019data where cat
egory_code<>"" and category_code is not Null;
Query ID = hadoop_20220101164813_0a6e883e-6ec9-41f5-91ae-eb4489505d06
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1641051379361_0008)

Map 1: 0/2      Reducer 2: 0/5
Map 1: 0/2      Reducer 2: 0/5
Map 1: 0(+1)/2  Reducer 2: 0/5
Map 1: 0(+2)/2  Reducer 2: 0/5
Map 1: 0(+2)/2  Reducer 2: 0/5
Map 1: 0(+2)/2  Reducer 2: 0/5
Map 1: 0(+2)/2  Reducer 2: 0/5
Map 1: 0(+2)/2  Reducer 2: 0/5
Map 1: 1(+1)/2  Reducer 2: 0(+1)/5
Map 1: 2/2      Reducer 2: 0(+2)/5
Map 1: 2/2      Reducer 2: 1(+2)/5
Map 1: 2/2      Reducer 2: 3(+0)/5
Map 1: 2/2      Reducer 2: 3(+2)/5
Map 1: 2/2      Reducer 2: 5/5
OK
category
furniture
appliances
accessories
apparel
sport
stationery
Time taken: 22.191 seconds, Fetched: 6 row(s)
hive>
```

- **Find the total number of products available under each category.**

Ans:

category	product_total
appliances	61736
stationery	26722
furniture	23604
apparel	18232
accessories	12929
sport	2

Query: select split(category_code,'\\.')[0] as Category,count(product_id) as product_total from 2019data where category_code<>"" and category_code is not Null group by split(category_code,'\\.')[0] order by product_total desc ;

Output Screenshot:

```
hadoop@ip-172-31-41-235:~
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1641051379361_0008)

Map 1: 0/2      Reducer 2: 0/5  Reducer 3: 0/1
Map 1: 0/2      Reducer 2: 0/5  Reducer 3: 0/1
Map 1: 0(+1)/2  Reducer 2: 0/5  Reducer 3: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/5  Reducer 3: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/5  Reducer 3: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/5  Reducer 3: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/5  Reducer 3: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/5  Reducer 3: 0/1
Map 1: 1(+1)/2  Reducer 2: 0(+1)/5      Reducer 3: 0/1
Map 1: 2/2      Reducer 2: 0(+2)/5      Reducer 3: 0/1
Map 1: 2/2      Reducer 2: 1(+2)/5      Reducer 3: 0/1
Map 1: 2/2      Reducer 2: 2(+2)/5      Reducer 3: 0/1
Map 1: 2/2      Reducer 2: 3(+1)/5      Reducer 3: 0(+1)/1
Map 1: 2/2      Reducer 2: 4(+0)/5      Reducer 3: 0(+1)/1
Map 1: 2/2      Reducer 2: 4(+1)/5      Reducer 3: 0(+1)/1
Map 1: 2/2      Reducer 2: 5/5  Reducer 3: 0(+1)/1
Map 1: 2/2      Reducer 2: 5/5  Reducer 3: 1/1
OK
category        product_total
appliances      61736
stationery      26722
furniture       23604
apparel 18232
accessories     12929
sport   2
Time taken: 25.939 seconds, Fetched: 6 row(s)
hive>
    >
```

- **Which brand had the maximum sales in October and November combined?**

Ans:

brand	sales_amt
runail	148297.94

Query: select brand,round(sum(price),2) as sales_amt from purchasedata where brand<>'' group by brand sort by sales_amt desc limit 1;

Screenshot:

```
hadoop@ip-172-31-41-235:~

hive> select brand,round(sum(price),2)  as sales_amt from purchasedata where brand<>'' g
roup by brand  sort by sales_amt desc limit 1;
Query ID = hadoop_20220101165949_8cd8f904-734c-496c-bfe5-f84431eae66b
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_1641051379361_0009)

Map 1: 0/2      Reducer 2: 0/1  Reducer 3: 0/1  Reducer 4: 0/1
Map 1: 0/2      Reducer 2: 0/1  Reducer 3: 0/1  Reducer 4: 0/1
Map 1: 0(+1)/2  Reducer 2: 0/1  Reducer 3: 0/1  Reducer 4: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1  Reducer 3: 0/1  Reducer 4: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1  Reducer 3: 0/1  Reducer 4: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1  Reducer 3: 0/1  Reducer 4: 0/1
Map 1: 1(+1)/2  Reducer 2: 0(+1)/1      Reducer 3: 0/1  Reducer 4: 0/1
Map 1: 2/2      Reducer 2: 0(+1)/1      Reducer 3: 0/1  Reducer 4: 0/1
Map 1: 2/2      Reducer 2: 1/1  Reducer 3: 0/1  Reducer 4: 0/1
Map 1: 2/2      Reducer 2: 1/1  Reducer 3: 0(+1)/1      Reducer 4: 0/1
Map 1: 2/2      Reducer 2: 1/1  Reducer 3: 1/1  Reducer 4: 0(+1)/1
Map 1: 2/2      Reducer 2: 1/1  Reducer 3: 1/1  Reducer 4: 1/1
OK
brand   sales_amt
runail  148297.94
Time taken: 26.042 seconds, Fetched: 1 row(s)
hive>
```

- **Which brands increased their sales from October to November?**
  Ans:
  brand
  airnails
  art-visage
  artex
  aura
  balbcare
  barbie
  batiste
  beautix
  beauty-free
  beautyblender
  beauugreen
  benovy
  binacil
  bioaqua
  biore
  blixz
  bluesky
  bodyton
  bpw.style
  browxenna
  candy
  carmex

chi
coifin
concept
cosima
cosmoprofi
cristalinas
cutrin
de.lux
deoproce
depilflax
dewal
dizao
domix
ecocraft
ecolab
egomania
elizavecca
ellips
elskin
enjoy
entity
eos
estel
estelare
f.o.x
farmavita
farmona
fedua
finish
fly
foamie
freedecor
freshbubble
gehwol
glysolid
godefroy
grace
grattol
greymy
happyfons
haruyama
helloganic
igrobeauty
ingarden
inm
insight
irisk
italwax
jaguar
jas
jessnail

joico
juno
kaaral
kamill
kapous
kares
kaypro
keen
kerasys
kims
kinetics
kiss
kocostar
koelcia
koelf
konad
kosmekka
laboratorium
lador
ladykin
latinoil
levissime
levrana
lianail
likato
limoni
lovely
lowence
mane
marathon
markell
marutaka-foot
masura
matreshka
matrix
mavala
metzger
milv
miskin
missha
moyou
nagaraku
naomi
nefertiti
neoleor
nirvel
nitrile
oniq
orly
osmo
ovale

plazan
polarus
profepil
profhenna
protokeratin
provoc
rasyan
refectocil
rosi
roubloff
runail
s.care
sanoto
severina
shary
shik
skinity
skinlite
smart
soleo
solomeya
sophin
staleks
strong
supertan
swarovski
tertio
treaclemoon
trind
uno
uskusi
veraclara
vilenta
yoko
yu-r
zeitun
Time taken: 23.332 seconds, Fetched: 160 row(s)

Query:

```
with branddata as(
SELECT brand,sum(case when date_format(event_time,'MM')=10 then price else 0 end) AS October,
      sum(case when date_format(event_time,'MM')=11 then price else 0 end) AS November
 FROM purchasedata group by brand)
select brand from branddata where (November - October)>0 and brand<>'';
```

Output Screenshot:

```
hadoop@ip-172-31-41-235:~
rasyan
refectocil
rosi
roubloff
runail
s.care
sanoto
severina
shary
shik
skinity
skinlite
smart
soleo
solomeya
sophin
staleks
strong
supertan
swarovski
tertio
treaclemoon
trind
uno
uskusi
veraclara
vilenta
yoko
yu-r
zeitun
Time taken: 23.332 seconds, Fetched: 160 row(s)
hive>
```

- **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.**

Ans:

top_10_users
557790271
150318419
562167663
531900924
557850743
522130011
561592095
431950134
566576008
521347209

Query: with user_details as(select user_id,round(sum(price),2) as amount_spent,dense_rank() over(order by sum(price) desc) as rank from purchasedata group by user_id)select user_id as Top_10_users from user_details where rank between 1 and 10 ;

Output Screenshot:

```
hadoop@ip-172-31-41-235:~

hive> set hive.cli.print.header=true;
hive> with user_details as( select user_id,round(sum(price),2) as amount_spent,dense_ran
elect user_id as Top_10_users from user_details where rank between 1 and 10 ;
Query ID = hadoop_20220101172811_f7f94e92-33a4-4bd5-a7da-89d266427b43
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1641051379361_0010)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KI
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      2          2        0        0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 18.39 s
--------------------------------------------------------------------------------
OK
top_10_users
557790271
150318419
562167663
531900924
557850743
522130011
561592095
431950134
566576008
521347209
Time taken: 19.03 seconds, Fetched: 10 row(s)
hive>
```

Step 4 :Terminating the EMR cluster after use