Hive Case Study
By
Nishant Raj And Guresh Kumar

We have completed the case study in 2 parts.

1. Working with S3 and EMR clusters.
2. Querying with HIVE (HQL)

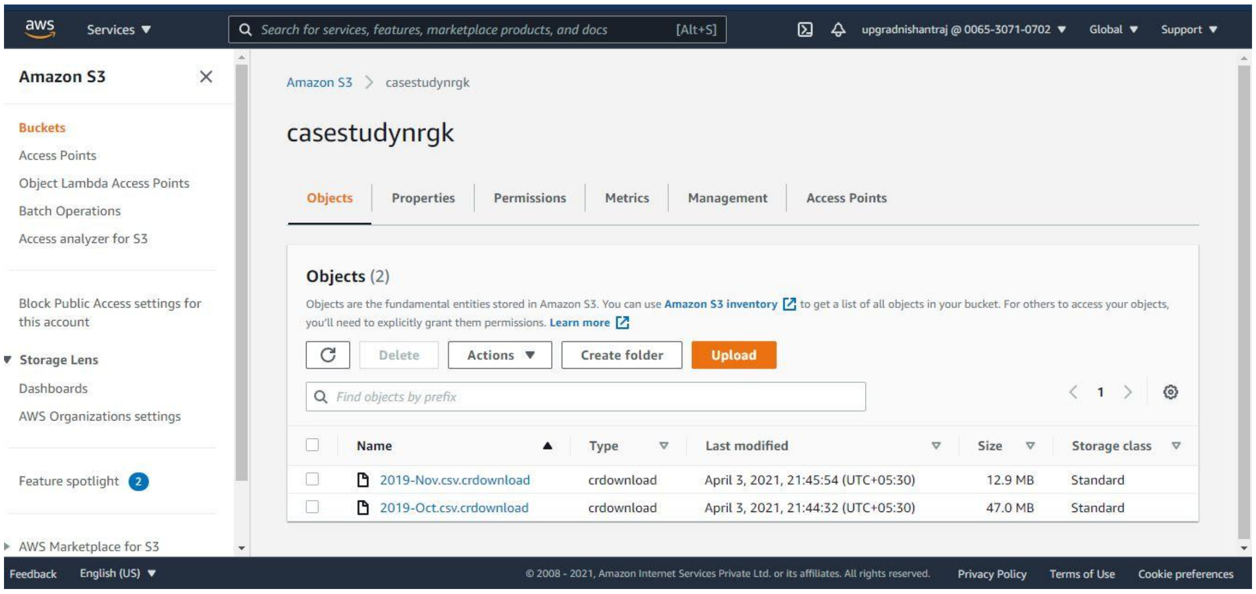
1. Working with S3 and EMR clusters:

(a) Load data into S3 bucket

The first step is to create a S3 bucket and then load Data in it. We are provided with links through which we can get the data.

https://e-commerce-events-ml.s3.amazonaws.com/2019-Oct.csv

https://e-commerce-events-ml.s3.amazonaws.com/2019-Nov.csv



Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
2019-Nov.csv.crdownload	crdownload	April 3, 2021, 21:45:54 (UTC+05:30)	12.9 MB	Standard
2019-Oct.csv.crdownload	crdownload	April 3, 2021, 21:44:32 (UTC+05:30)	47.0 MB	Standard

© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.

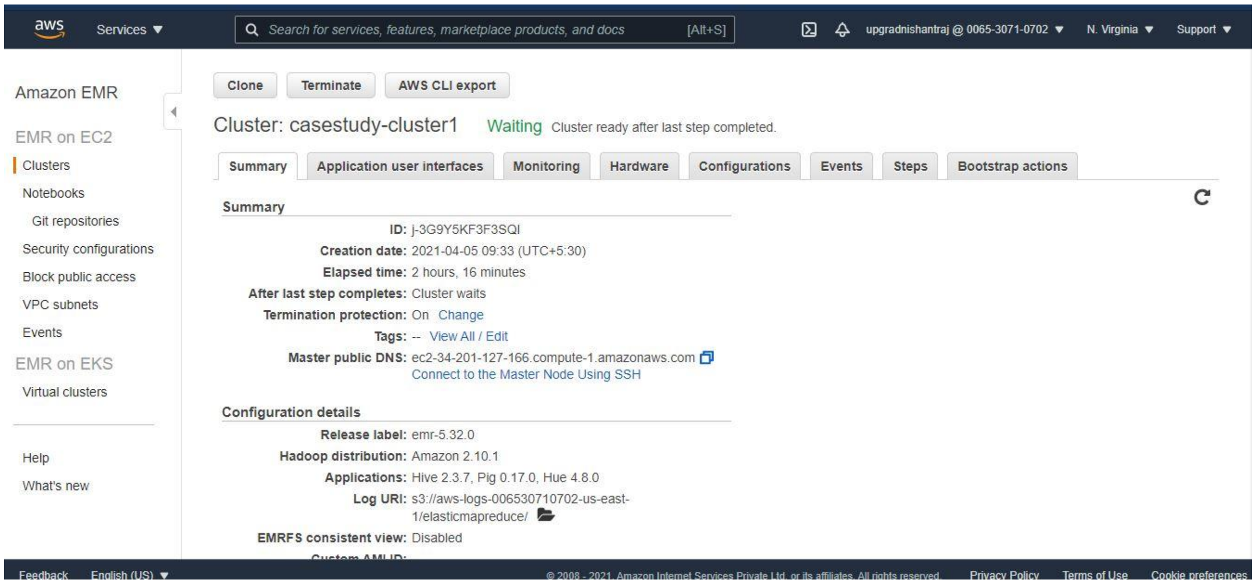
We have loaded the S3 bucket with 2 given data sets that we need to work on,

Data set 1: 2019-Nov.csv.crdownload and
Data set 2: 2019-Oct.csv.crdownload.

We need to extract data and gather insights from these data sets. In the next step will see how the data in S3 bucket is moved into the HDFS.

(b) Launch an EMR cluster

We have created an Elastic MapReduce cluster with the suggested configuration. We have named our cluster as casestudy-cluster1. We have created the cluster with release version 5.32.0 and used two-node cluster [1 Master M4.large and 1 Core M4.large]



Cluster: casestudy-cluster1 Waiting Cluster ready after last step completed.

Summary

ID: j-3G9Y5KF3F3SQI
Creation date: 2021-04-05 09:33 (UTC+5:30)
Elapsed time: 2 hours, 16 minutes
After last step completes: Cluster waits
Termination protection: On Change
Tags: -- View All / Edit
Master public DNS: ec2-34-201-127-166.compute-1.amazonaws.com
Connect to the Master Node Using SSH

Configuration details

Release label: emr-5.32.0
Hadoop distribution: Amazon 2.10.1
Applications: Hive 2.3.7, Pig 0.17.0, Hue 4.8.0
Log URI: s3://aws-logs-006530710702-us-east-1/elasticmapreduce/
EMRFS consistent view: Disabled

We'll utilize Hive services with this EMR cluster and give instructions through commands using PuTTy.

(c) Copy Data from EMR cluster to HDFS

We have used **Hadoop distcp** to copy files in the S3 bucket to the Hadoop file system. This job runs like a MapReduce job in the background to copy the files from EMR cluster to HDFS.

```
load data local inpath "/home/hadoop/2019-Oct.csv.crdownload" into table oct_info;

load data local inpath "/home/hadoop/2019-Nov.csv.crdownload" into table nov_info;
```

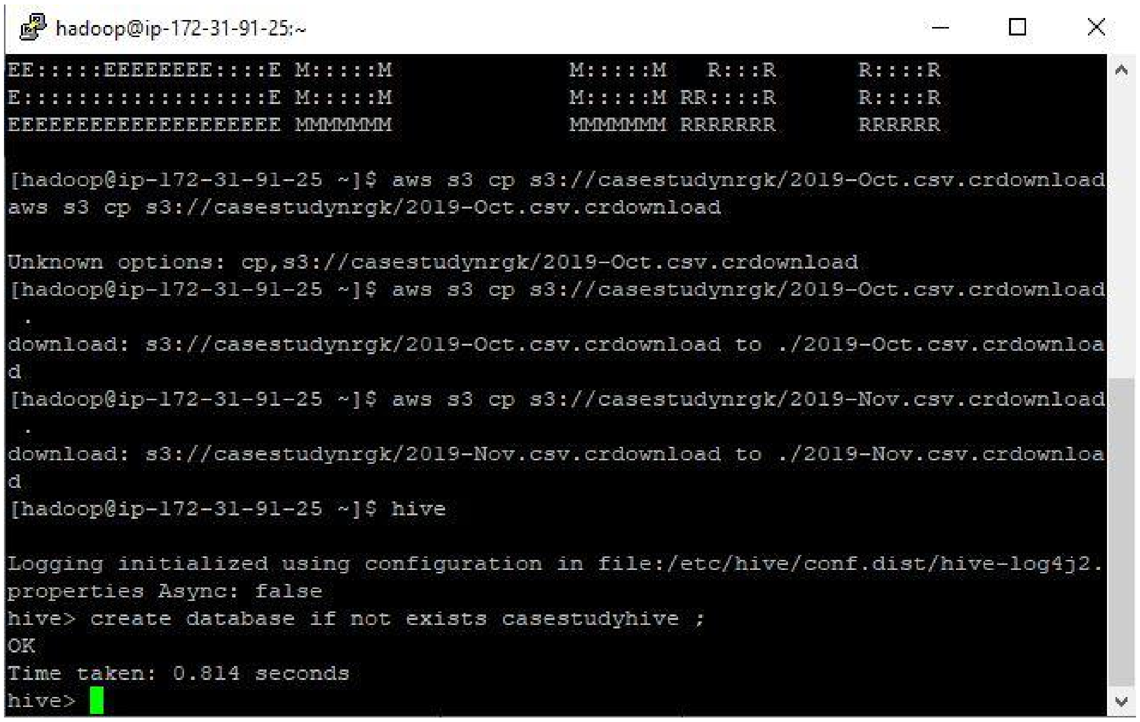
We have used the above commands to achieve the task.

2. Querying with HIVE (HQL):

(a) Structure of your database

Step 1: Create a Database in Hadoop

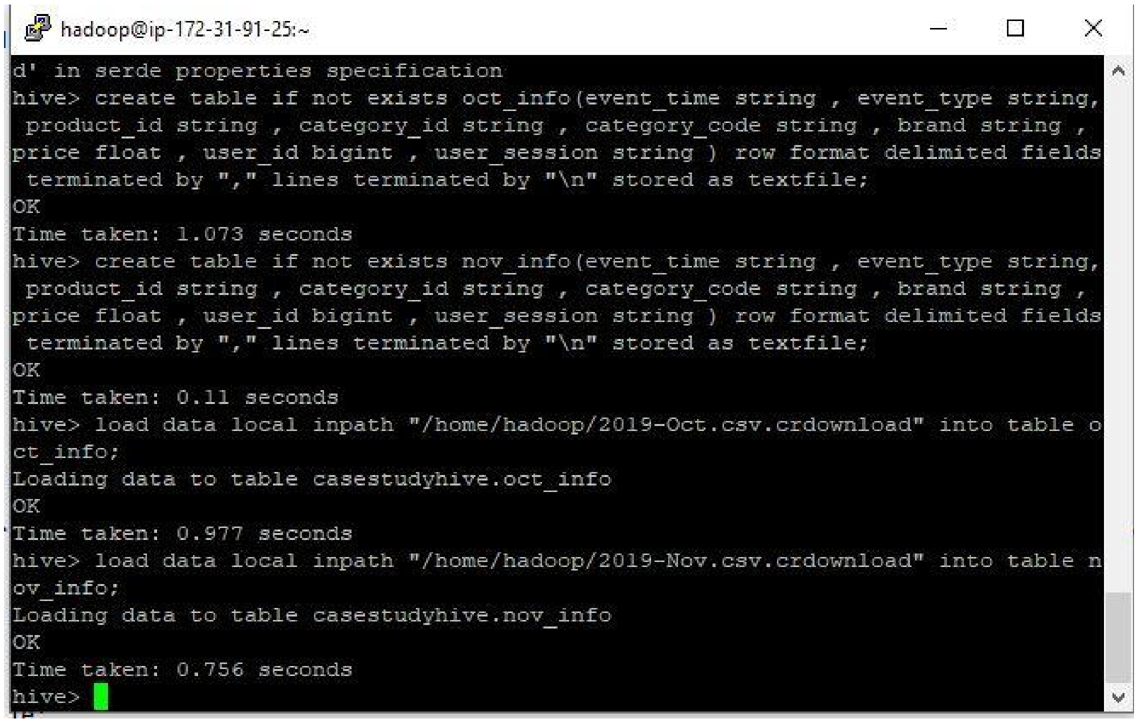
As we can see from the screenshot the data from S3 has been loaded to Hadoop File System and the database with the name of casestudyhive is created.



```
EE:::::EEEEEEEE::::E M:::::M            M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M            M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM            MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-91-25 ~]$ aws s3 cp s3://casestudynrgk/2019-Oct.csv.crdownload
aws s3 cp s3://casestudynrgk/2019-Oct.csv.crdownload

Unknown options: cp,s3://casestudynrgk/2019-Oct.csv.crdownload
[hadoop@ip-172-31-91-25 ~]$ aws s3 cp s3://casestudynrgk/2019-Oct.csv.crdownload
 .
download: s3://casestudynrgk/2019-Oct.csv.crdownload to ./2019-Oct.csv.crdownloa
d
[hadoop@ip-172-31-91-25 ~]$ aws s3 cp s3://casestudynrgk/2019-Nov.csv.crdownload
 .
download: s3://casestudynrgk/2019-Nov.csv.crdownload to ./2019-Nov.csv.crdownloa
d
[hadoop@ip-172-31-91-25 ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.
properties Async: false
hive> create database if not exists casestudyhive ;
OK
Time taken: 0.814 seconds
hive>
```

Step 2: Create required tables in the Database

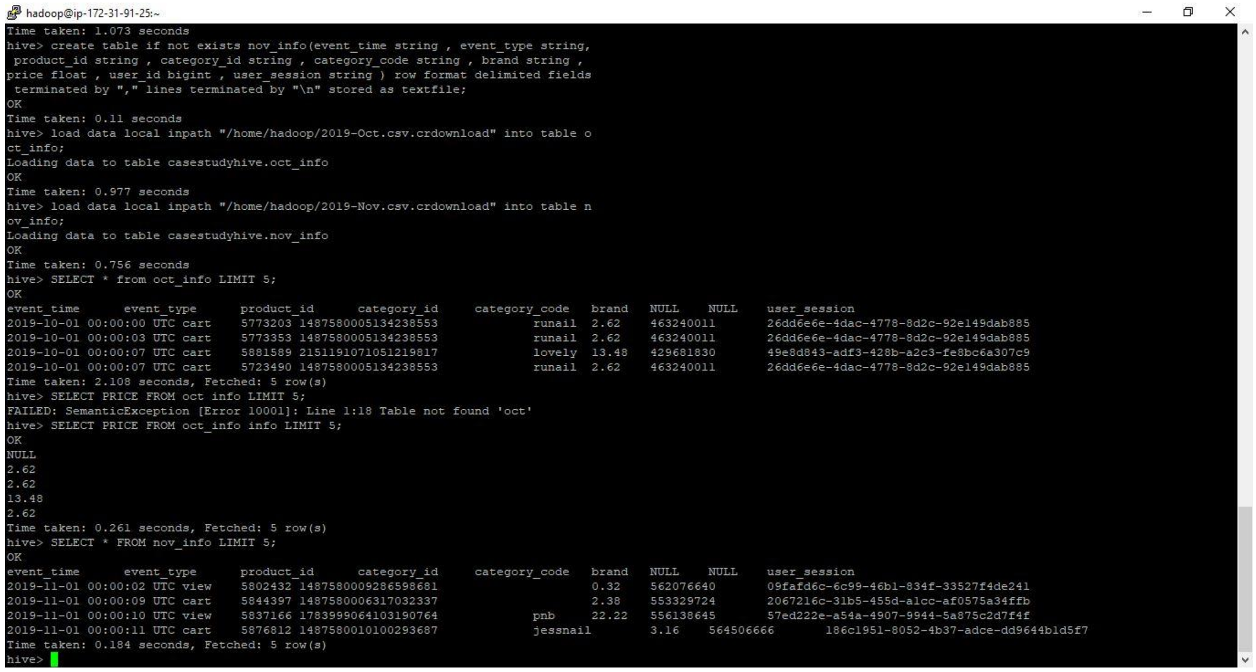
After creating the database we have created the tables so that data could be loaded into the tables. Here we have created two tables namely oct_info and nov_info which will be used in the subsequent screenshots for querying .



```
d' in serde properties specification
hive> create table if not exists oct_info(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;
OK
Time taken: 1.073 seconds
hive> create table if not exists nov_info(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;
OK
Time taken: 0.11 seconds
hive> load data local inpath "/home/hadoop/2019-Oct.csv.crdownload" into table o
ct_info;
Loading data to table casestudyhive.oct_info
OK
Time taken: 0.977 seconds
hive> load data local inpath "/home/hadoop/2019-Nov.csv.crdownload" into table n
ov_info;
Loading data to table casestudyhive.nov_info
OK
Time taken: 0.756 seconds
hive>
```

Step 3: Load Data into tables

After creating the tables, now we have added data into the tables we can see clearly that the data is perfectly loaded into the tables.



```
Time taken: 1.073 seconds
hive> create table if not exists nov_info(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;
OK
Time taken: 0.11 seconds
hive> load data local inpath "/home/hadoop/2019-Oct.csv.crdownload" into table o
ct_info;
Loading data to table casestudyhive.oct_info
OK
Time taken: 0.977 seconds
hive> load data local inpath "/home/hadoop/2019-Nov.csv.crdownload" into table n
ov_info;
Loading data to table casestudyhive.nov_info
OK
Time taken: 0.756 seconds
hive> SELECT * from oct_info LIMIT 5;
OK
event_time      event_type      product_id      category_id     category_code   brand   NULL    NULL    user_session
2019-10-01 00:00:00 UTC cart    5773203 1487580005134238553             runail  2.62    463240011       26dd6e6e-4dac-4778-8d2c-92e149dab885
2019-10-01 00:00:03 UTC cart    5773353 1487580005134238553             runail  2.62    463240011       26dd6e6e-4dac-4778-8d2c-92e149dab885
2019-10-01 00:00:07 UTC cart    5881589 2151191071051219817             lovely  13.48   429681830       49e8d843-adf3-428b-a2c3-fe8bc6a307c9
2019-10-01 00:00:07 UTC cart    5723490 1487580005134238553             runail  2.62    463240011       26dd6e6e-4dac-4778-8d2c-92e149dab885
Time taken: 2.108 seconds, Fetched: 5 row(s)
hive> SELECT PRICE FROM oct info LIMIT 5;
FAILED: SemanticException [Error 10001]: Line 1:18 Table not found 'oct'
hive> SELECT PRICE FROM oct_info info LIMIT 5;
OK
NULL
2.62
2.62
13.48
2.62
Time taken: 0.261 seconds, Fetched: 5 row(s)
hive> SELECT * FROM nov_info LIMIT 5;
OK
event_time      event_type      product_id      category_id     category_code   brand   NULL    NULL    user_session
2019-11-01 00:00:02 UTC view    5802432 1487580009286598681                     0.32    562076640       09fafd6c-6c99-46b1-834f-33527f4de241
2019-11-01 00:00:09 UTC cart    5844397 1487580006317032337                     2.38    553329724       2067216c-31b5-455d-a1cc-af0575a34ffb
2019-11-01 00:00:10 UTC view    5837166 1783999064103190764             pnb     22.22   556138645       57ed222e-a54a-4907-9944-5a875c2d7f4f
2019-11-01 00:00:11 UTC cart    5876812 1487580010100293687             jessnail        3.16    564506666       186c1951-8052-4b37-adce-dd9644b1d5f7
Time taken: 0.184 seconds, Fetched: 5 row(s)
hive>
```

(b) Select an optimized technique to run your queries as efficiently as possible

We have 2 optimization techniques to run a query. One is Partitioning and the other one is Bucketing.
In Partitioning we'll divide the data present in tables into several parts based on columns and the conditions that we apply using the partition keys. In Bucketing, the partitioned data is further subdivided into buckets based on Hash function. Thus, Bucketting helps us with an optimized solution for our queries.

(c) Applying both the techniques for a query to see the performance
**** We haven't captured the screenshot for the query and terminated the cluster. We have mentioned the code that we have used to check the performance.**

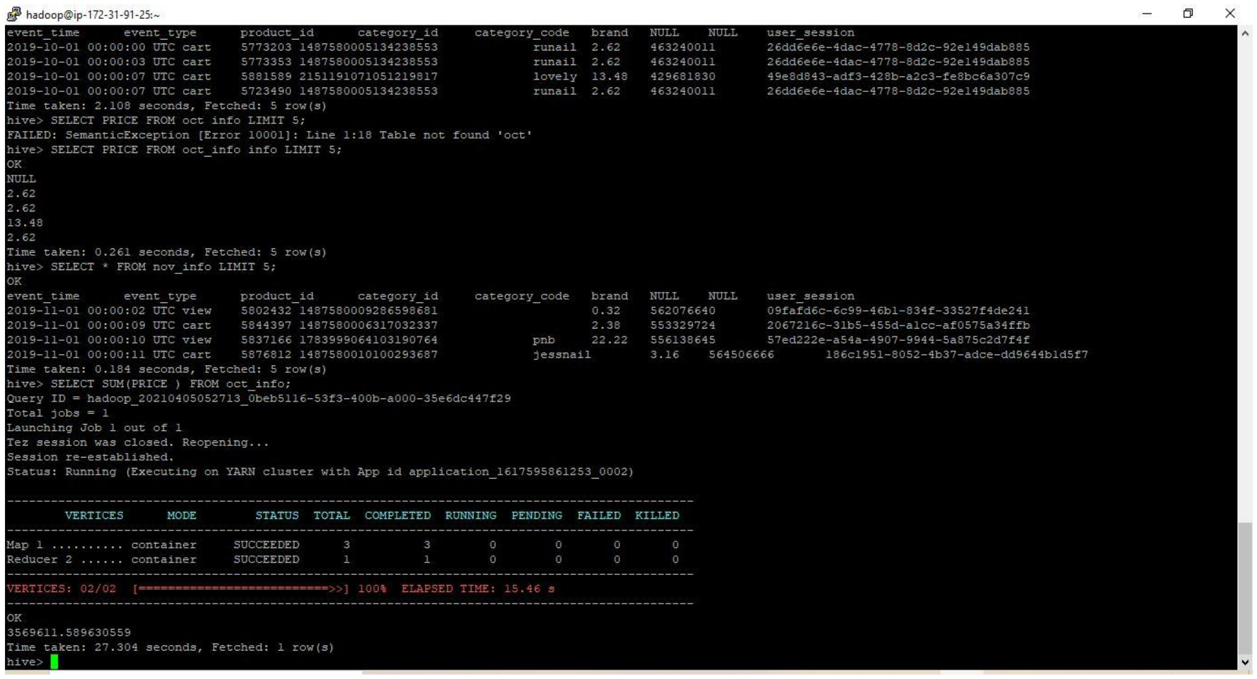
```
create table if not exists part_oct_info( event_time string , event_type
string , product_id string ,category_id string , category_code string ,
price float , user_id bigint , user_session string) partitioned by (brand
string) row farmat delimited by "," lines terminated by "\n";


insert into table part_oct_info partition(category_code) select event_time
string , event_type string , product_id string ,category_id string , brand
string , price float , user_id bigint , user_session string;

exit;

hadoop fs -ls "user/hive/warehouse/part_oct_info"

SELECT user_id , sum(price) from oct_info where brand="xyz" group by
category_code;

SELECT user_id , sum(price) from part_oct_info where brand="xyz" group by
category_code;

=============================================

BUCKETING

create table if not exists buck_oct_info( event_time string , event_type
string , product_id string ,category_id string , category_code string ,
price float , user_id bigint , user_session string) partitioned by (brand
string) clustered by (category_code) into 10 buckets row farmat delimited
by "," lines terminated by "\n";
```

```
insert into table buck_oct_info partition(category_code) select event_time
string , event_type string , product_id string ,category_id string , brand
string , price float , user_id bigint , user_session string;

hadoop fs -ls /user/hive/warehouse/buck_oct_info;

We can clearly see that bucketing will significantly reduce the query
time.
```

(d) Run Hive queries to answer the Case study questions

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

We have used SELECT SUM(PRICE) FROM oct_info; command to solve the query.

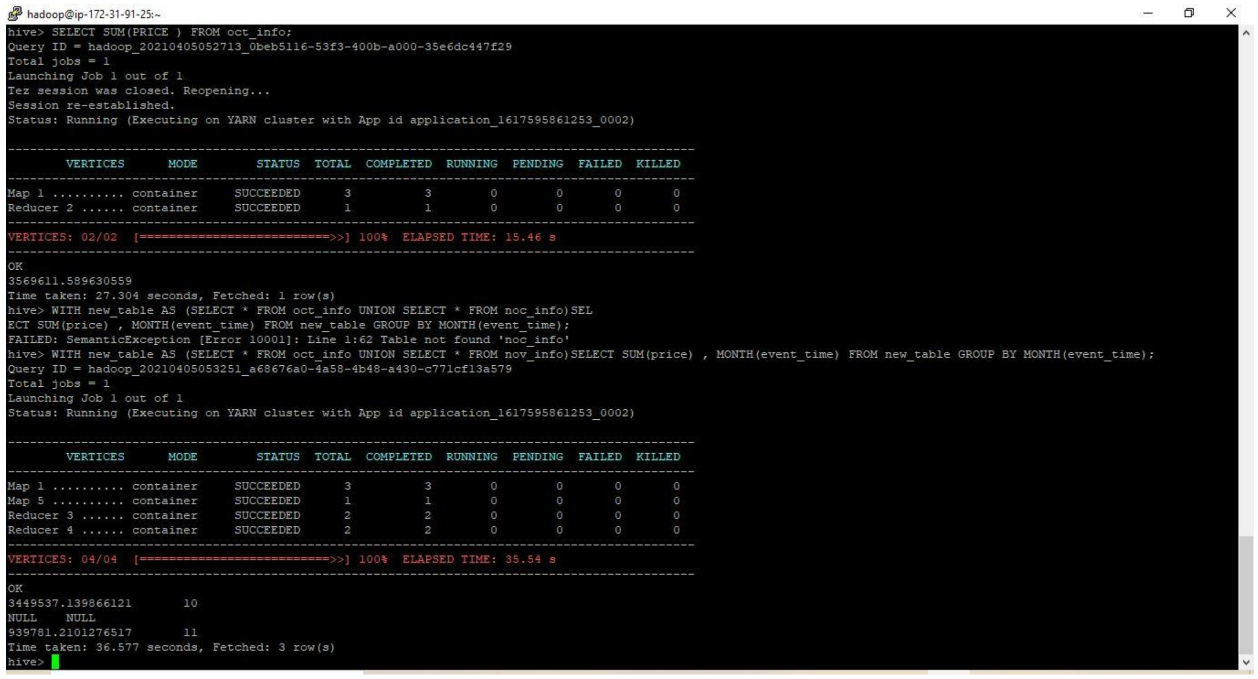


hadoop@ip-172-31-91-25:~

```
event_time      event_type      product_id      category_id     category_code   brand   NULL    NULL    user_session
2019-10-01 00:00:00 UTC cart    5773203 1487580005134238553             runail  2.62    463240011       26dd6e6e-4dac-4778-8d2c-92e149dab885
2019-10-01 00:00:03 UTC cart    5773353 1487580005134238553             runail  2.62    463240011       26dd6e6e-4dac-4778-8d2c-92e149dab885
2019-10-01 00:00:07 UTC cart    5881589 2151191071051219817             lovely  13.48   429681830       49e8d843-adf3-428b-a2c3-fe8bc6a307c9
2019-10-01 00:00:07 UTC cart    5723490 1487580005134238553             runail  2.62    463240011       26dd6e6e-4dac-4778-8d2c-92e149dab885
Time taken: 2.108 seconds, Fetched: 5 row(s)
hive> SELECT PRICE FROM oct info LIMIT 5;
FAILED: SemanticException [Error 10001]: Line 1:18 Table not found 'oct'
hive> SELECT PRICE FROM oct_info info LIMIT 5;
OK
NULL
2.62
2.62
13.48
2.62
Time taken: 0.261 seconds, Fetched: 5 row(s)
hive> SELECT * FROM nov_info LIMIT 5;
OK
event_time      event_type      product_id      category_id     category_code   brand   NULL    NULL    user_session
2019-11-01 00:00:02 UTC view    5802432 1487580009286598681                     0.32    562076640       09fafd6c-6c99-46b1-834f-33527f4de241
2019-11-01 00:00:09 UTC cart    5844397 1487580006317032337                     2.38    553329724       2067216c-31b5-455d-a1cc-af0575a34ffb
2019-11-01 00:00:10 UTC view    5837166 1783999064103190764             pnb     22.22   556138645       57ed222e-a54a-4907-9944-5a875c2d7f4f
2019-11-01 00:00:11 UTC cart    5876812 1487580010100293687             jessnail        3.16    564506666       186c1951-8052-4b37-adce-dd9644b1d5f7
Time taken: 0.184 seconds, Fetched: 5 row(s)
hive> SELECT SUM(PRICE ) FROM oct_info;
Query ID = hadoop_20210405052713_0beb5116-53f3-400b-a000-35e6dc447f29
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_1617595861253_0002)

----------------------------------------------------------------------------------------------
        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
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 15.46 s
----------------------------------------------------------------------------------------------
OK
3569611.589630559
Time taken: 27.304 seconds, Fetched: 1 row(s)
hive>
```

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

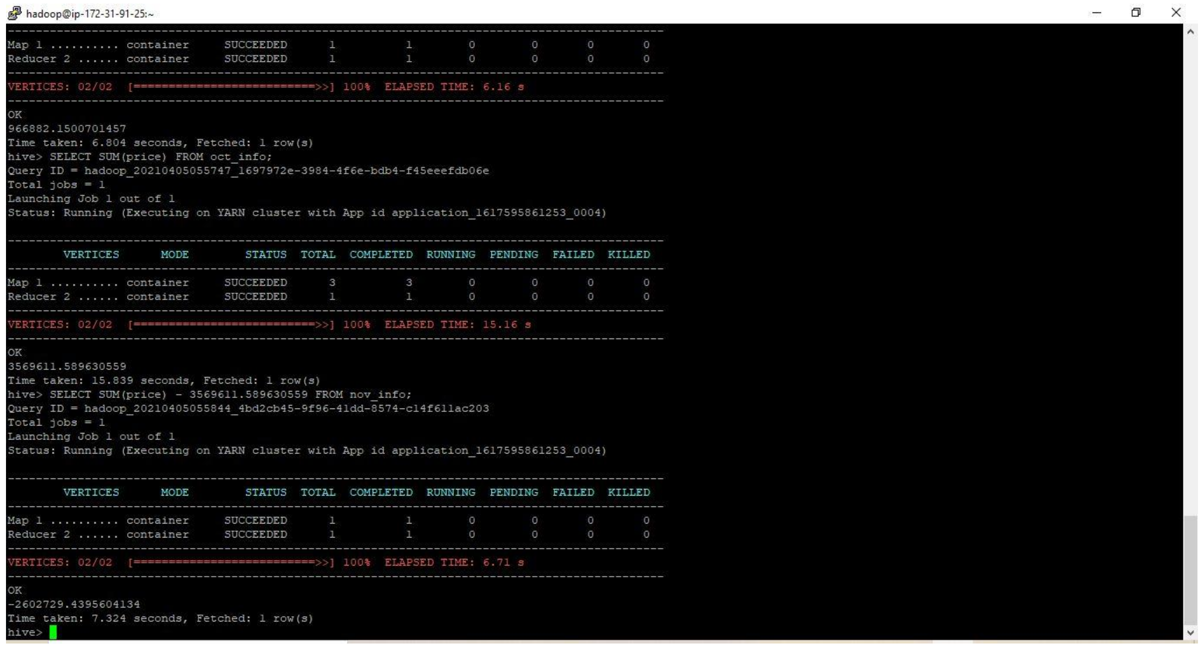
The below query produces the desired output.



```
hive> SELECT SUM(PRICE ) FROM oct_info;
Query ID = hadoop_20210405052713_0beb5116-53f3-400b-a000-35e6dc447f29
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_1617595861253_0002)

----------------------------------------------------------------------------------------------
        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
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 15.46 s
----------------------------------------------------------------------------------------------
OK
3569611.589630559
Time taken: 27.304 seconds, Fetched: 1 row(s)
hive> WITH new_table AS (SELECT * FROM oct_info UNION SELECT * FROM noc_info)SEL
ECT SUM(price) , MONTH(event_time) FROM new_table GROUP BY MONTH(event_time);
FAILED: SemanticException [Error 10001]: Line 1:62 Table not found 'noc_info'
hive> WITH new_table AS (SELECT * FROM oct_info UNION SELECT * FROM nov_info)SELECT SUM(price) , MONTH(event_time) FROM new_table GROUP BY MONTH(event_time);
Query ID = hadoop_20210405053251_a68676a0-4a58-4b48-a430-c771cf13a579
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1617595861253_0002)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      3          3        0        0       0       0
Map 5 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      2          2        0        0       0       0
Reducer 4 ...... container     SUCCEEDED      2          2        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 04/04  [==========================>>] 100%  ELAPSED TIME: 35.54 s
----------------------------------------------------------------------------------------------
OK
3449537.139866121       10
NULL    NULL
939781.2101276517       11
Time taken: 36.577 seconds, Fetched: 3 row(s)
hive>
```

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

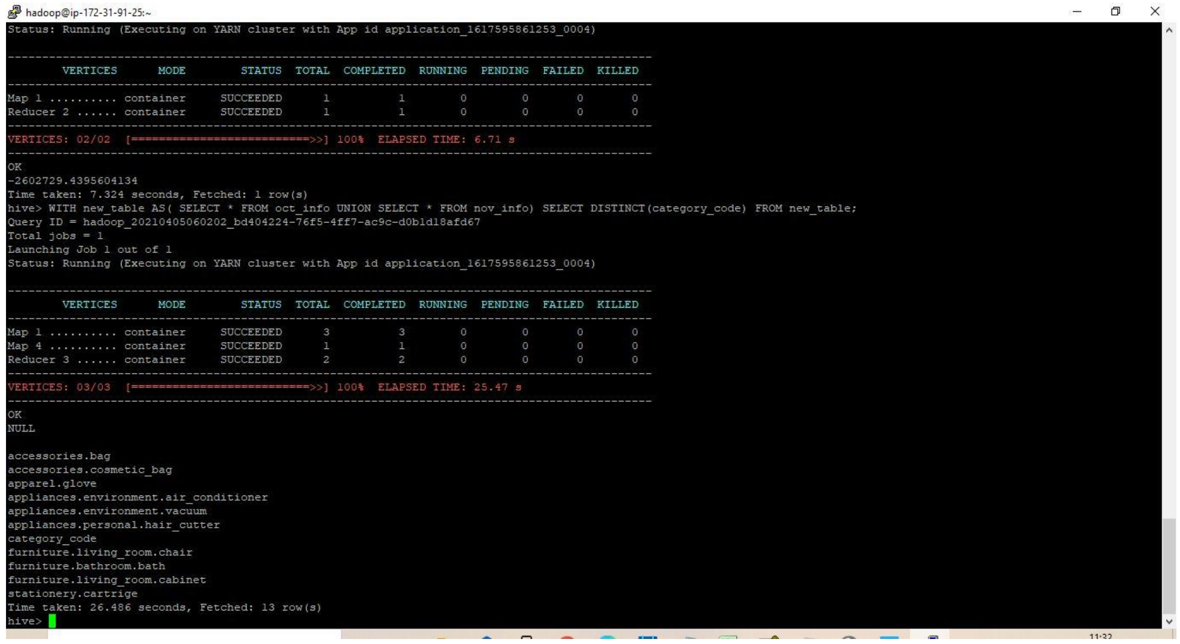
The below query produces the desired output.



```
hadoop@ip-172-31-91-25:~
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 6.16 s
----------------------------------------------------------------------------------------------
OK
966882.1500701457
Time taken: 6.804 seconds, Fetched: 1 row(s)
hive> SELECT SUM(price) FROM oct_info;
Query ID = hadoop_20210405055747_1697972e-3984-4f6e-bdb4-f45eeefdb06e
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1617595861253_0004)

----------------------------------------------------------------------------------------------
        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
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 15.16 s
----------------------------------------------------------------------------------------------
OK
3569611.589630559
Time taken: 15.839 seconds, Fetched: 1 row(s)
hive> SELECT SUM(price) - 3569611.589630559 FROM nov_info;
Query ID = hadoop_20210405055844_4bd2cb45-9f96-41dd-8574-c14f611ac203
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1617595861253_0004)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 6.71 s
----------------------------------------------------------------------------------------------
OK
-2602729.4395604134
Time taken: 7.324 seconds, Fetched: 1 row(s)
hive>
```

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

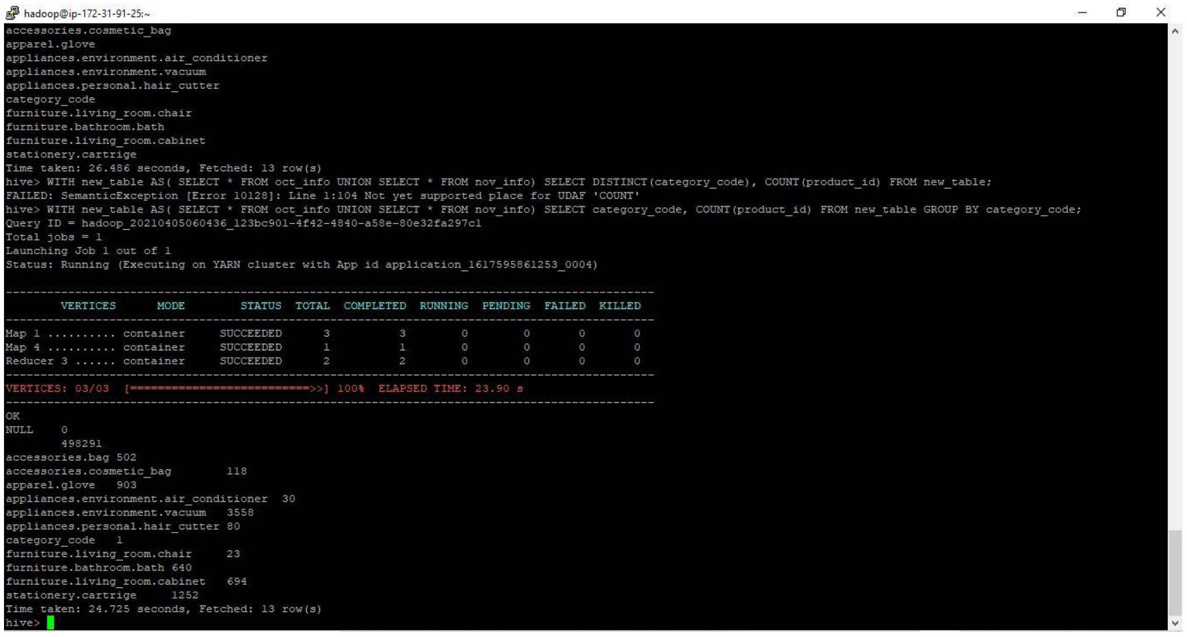
The below query produces the desired output.



```
hadoop@ip-172-31-91-25:~
Status: Running (Executing on YARN cluster with App id application_1617595861253_0004)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 6.71 s
----------------------------------------------------------------------------------------------
OK
-2602729.4395604134
Time taken: 7.324 seconds, Fetched: 1 row(s)
hive> WITH new_table AS( SELECT * FROM oct_info UNION SELECT * FROM nov_info) SELECT DISTINCT(category_code) FROM new_table;
Query ID = hadoop_20210405060202_bd404224-76f5-4ff7-ac9c-d0b1d18afd67
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1617595861253_0004)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      3          3        0        0       0       0
Map 4 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      2          2        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 25.47 s
----------------------------------------------------------------------------------------------
OK
NULL

accessories.bag
accessories.cosmetic_bag
apparel.glove
appliances.environment.air_conditioner
appliances.environment.vacuum
appliances.personal.hair_cutter
category_code
furniture.living_room.chair
furniture.bathroom.bath
furniture.living_room.cabinet
stationery.cartrige
Time taken: 26.486 seconds, Fetched: 13 row(s)
hive>
```

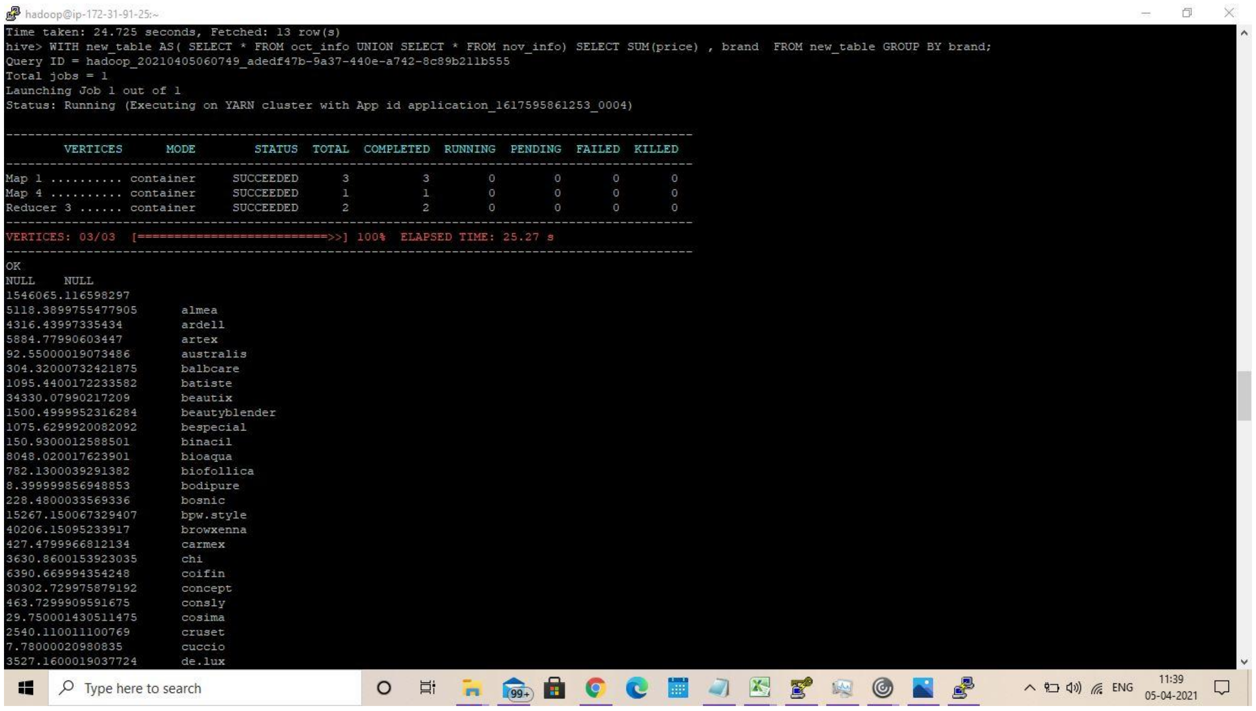
5. Find the total number of products available under each category.

The below query produces the desired output.



```
accessories.cosmetic_bag
apparel.glove
appliances.environment.air_conditioner
appliances.environment.vacuum
appliances.personal.hair_cutter
category_code
furniture.living_room.chair
furniture.bathroom.bath
furniture.living_room.cabinet
stationery.cartrige
Time taken: 26.486 seconds, Fetched: 13 row(s)
hive> WITH new_table AS( SELECT * FROM oct_info UNION SELECT * FROM nov_info) SELECT DISTINCT(category_code), COUNT(product_id) FROM new_table;
FAILED: SemanticException [Error 10128]: Line 1:104 Not yet supported place for UDAF 'COUNT'
hive> WITH new_table AS( SELECT * FROM oct_info UNION SELECT * FROM nov_info) SELECT category_code, COUNT(product_id) FROM new_table GROUP BY category_code;
Query ID = hadoop_20210405060436_123bc901-4f42-4840-a58e-80e32fa297c1
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1617595861253_0004)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      3          3        0        0       0       0
Map 4 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      2          2        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 23.90 s
----------------------------------------------------------------------------------------------
OK
NULL    0
        498291
accessories.bag 502
accessories.cosmetic_bag        118
apparel.glove   903
appliances.environment.air_conditioner  30
appliances.environment.vacuum   3558
appliances.personal.hair_cutter 80
category_code   1
furniture.living_room.chair     23
furniture.bathroom.bath 640
furniture.living_room.cabinet   694
stationery.cartrige     1252
Time taken: 24.725 seconds, Fetched: 13 row(s)
hive>
```

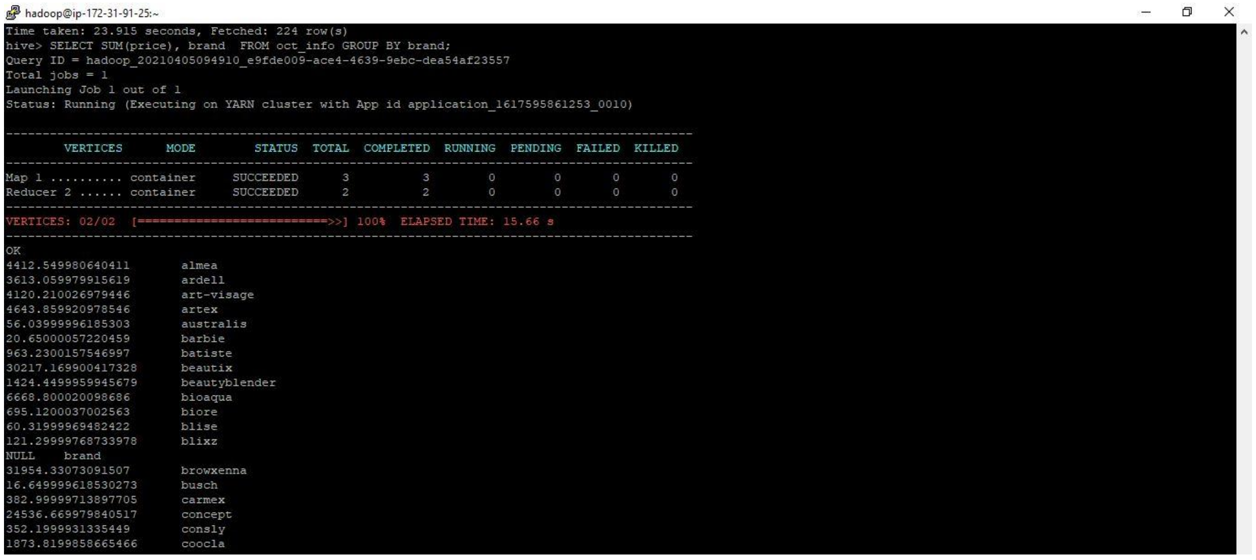
6. Which brand had the maximum sales in October and November combined?



```
Time taken: 24.725 seconds, Fetched: 13 row(s)
hive> WITH new_table AS( SELECT * FROM oct_info UNION SELECT * FROM nov_info) SELECT SUM(price) , brand  FROM new_table GROUP BY brand;
Query ID = hadoop_20210405060749_adedf47b-9a37-440e-a742-8c89b211b555
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1617595861253_0004)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      3          3        0        0       0       0
Map 4 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      2          2        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 25.27 s
----------------------------------------------------------------------------------------------
OK
NULL    NULL
1546065.116598297
5118.3899755477905      almea
4316.43997335434        ardell
5884.77990603447        artex
92.55000019073486       australis
304.32000732421875      balbcare
1095.4400172233582      batiste
34330.07990217209       beautix
1500.4999952316284      beautyblender
1075.6299920082092      bespecial
150.9300012588501       binacil
8048.020017623901       bioaqua
782.1300039291382       biofollica
8.399999856948853       bodipure
228.4800033569336       bosnic
15267.150067329407      bpw.style
40206.15095233917       browxenna
427.4799966812134       carmex
3630.8600153923035      chi
6390.669994354248       coifin
30302.729975879192      concept
463.7299909591675       consly
29.750001430511475      cosima
2540.110011100769       cruset
7.78000020980835        cuccio
3527.1600019037724      de.lux
```

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

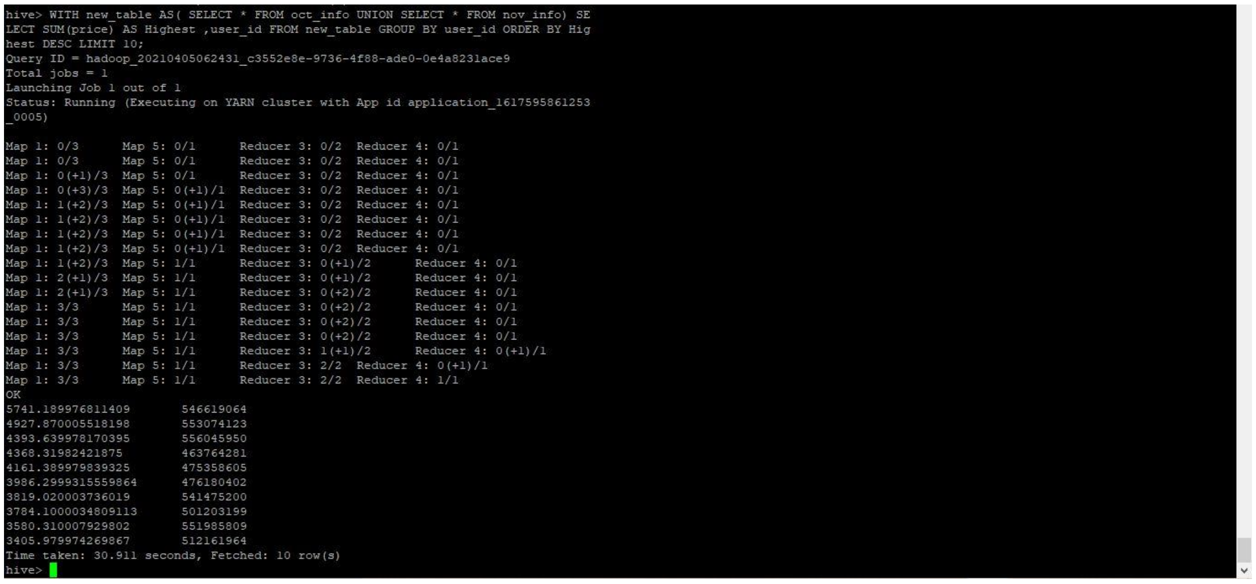
The below query produces the desired output.



```
hadoop@ip-172-31-91-25:~
Time taken: 23.915 seconds, Fetched: 224 row(s)
hive> SELECT SUM(price), brand  FROM oct_info GROUP BY brand;
Query ID = hadoop_20210405094910_e9fde009-ace4-4639-9ebc-dea54af23557
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1617595861253_0010)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      3          3        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 15.66 s
----------------------------------------------------------------------------------------------
OK
4412.549980640411       almea
3613.059979915619       ardell
4120.210026979446       art-visage
4643.859920978546       artex
56.03999996185303       australis
20.65000057220459       barbie
963.2300157546997       batiste
30217.169900417328      beautix
1424.4499959945679      beautyblender
6668.800020098686       bioaqua
695.1200037002563       biore
60.31999969482422       blise
121.29999768733978      blixz
NULL    brand
31954.33073091507       browxenna
16.649999618530273      busch
382.99999713897705      carmex
24536.669979840517      concept
352.1999931335449       consly
1873.8199858665466      coocla
```

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.

The below query produces the desired output.

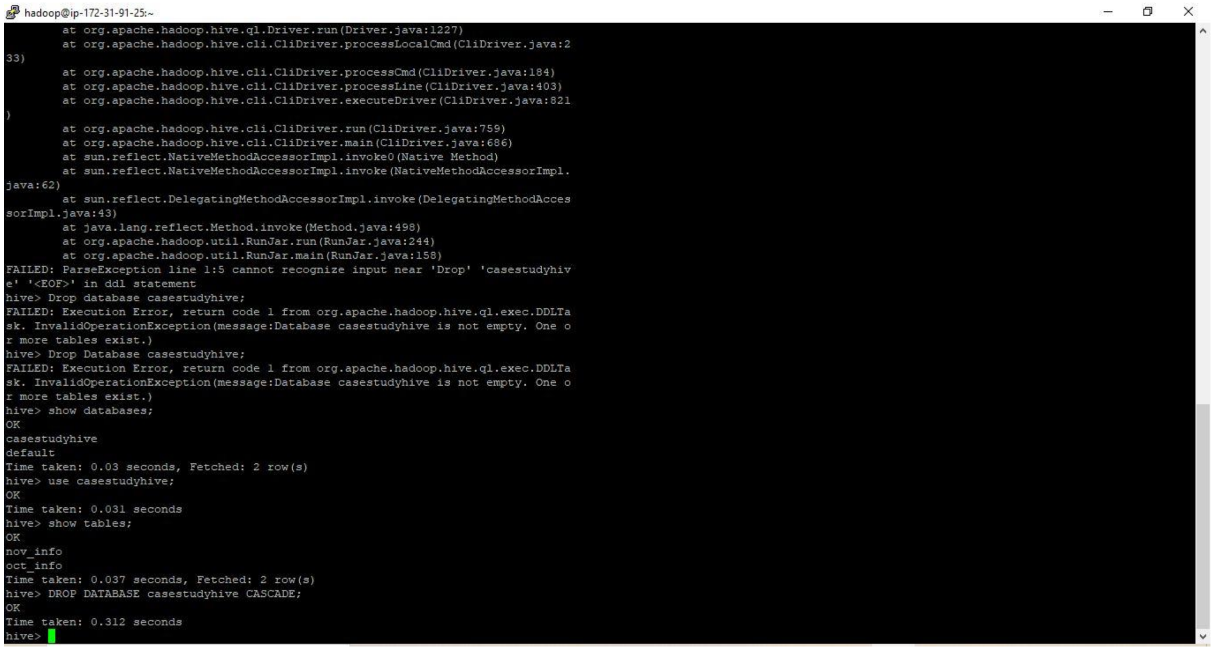


```
hive> WITH new_table AS( SELECT * FROM oct_info UNION SELECT * FROM nov_info) SE
LECT SUM(price) AS Highest ,user_id FROM new_table GROUP BY user_id ORDER BY Hig
hest DESC LIMIT 10;
Query ID = hadoop_20210405062431_c3552e8e-9736-4f88-ade0-0e4a8231ace9
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1617595861253
_0005)

Map 1: 0/3      Map 5: 0/1      Reducer 3: 0/2  Reducer 4: 0/1
Map 1: 0/3      Map 5: 0/1      Reducer 3: 0/2  Reducer 4: 0/1
Map 1: 0(+1)/3  Map 5: 0/1      Reducer 3: 0/2  Reducer 4: 0/1
Map 1: 0(+3)/3  Map 5: 0(+1)/1  Reducer 3: 0/2  Reducer 4: 0/1
Map 1: 1(+2)/3  Map 5: 0(+1)/1  Reducer 3: 0/2  Reducer 4: 0/1
Map 1: 1(+2)/3  Map 5: 0(+1)/1  Reducer 3: 0/2  Reducer 4: 0/1
Map 1: 1(+2)/3  Map 5: 0(+1)/1  Reducer 3: 0/2  Reducer 4: 0/1
Map 1: 1(+2)/3  Map 5: 0(+1)/1  Reducer 3: 0/2  Reducer 4: 0/1
Map 1: 1(+2)/3  Map 5: 1/1      Reducer 3: 0(+1)/2      Reducer 4: 0/1
Map 1: 2(+1)/3  Map 5: 1/1      Reducer 3: 0(+1)/2      Reducer 4: 0/1
Map 1: 2(+1)/3  Map 5: 1/1      Reducer 3: 0(+2)/2      Reducer 4: 0/1
Map 1: 3/3      Map 5: 1/1      Reducer 3: 0(+2)/2      Reducer 4: 0/1
Map 1: 3/3      Map 5: 1/1      Reducer 3: 0(+2)/2      Reducer 4: 0/1
Map 1: 3/3      Map 5: 1/1      Reducer 3: 0(+2)/2      Reducer 4: 0/1
Map 1: 3/3      Map 5: 1/1      Reducer 3: 1(+1)/2      Reducer 4: 0(+1)/1
Map 1: 3/3      Map 5: 1/1      Reducer 3: 2/2  Reducer 4: 0(+1)/1
Map 1: 3/3      Map 5: 1/1      Reducer 3: 2/2  Reducer 4: 1/1
OK
5741.189976811409       546619064
4927.870005518198       553074123
4393.639978170395       556045950
4368.31982421875        463764281
4161.389979839325       475358605
3986.2999315559864      476180402
3819.020003736019       541475200
3784.1000034809113      501203199
3580.310007929802       551985809
3405.979974269867       512161964
Time taken: 30.911 seconds, Fetched: 10 row(s)
hive>
```

*** Cleaning up:**

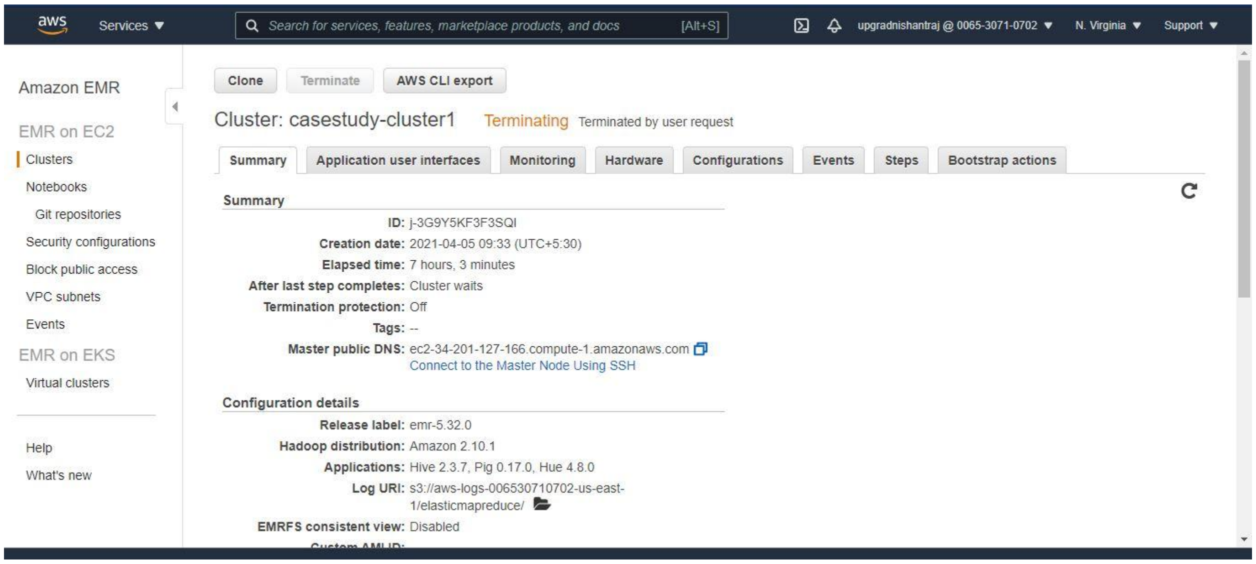
(a) Dropping Database

The below query Drops off the Database.



```
        at org.apache.hadoop.hive.ql.Driver.run(Driver.java:1227)
        at org.apache.hadoop.hive.cli.CliDriver.processLocalCmd(CliDriver.java:2
33)
        at org.apache.hadoop.hive.cli.CliDriver.processCmd(CliDriver.java:184)
        at org.apache.hadoop.hive.cli.CliDriver.processLine(CliDriver.java:403)
        at org.apache.hadoop.hive.cli.CliDriver.executeDriver(CliDriver.java:821
)
        at org.apache.hadoop.hive.cli.CliDriver.run(CliDriver.java:759)
        at org.apache.hadoop.hive.cli.CliDriver.main(CliDriver.java:686)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.
java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAcces
sorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at org.apache.hadoop.util.RunJar.run(RunJar.java:244)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:158)
FAILED: ParseException line 1:5 cannot recognize input near 'Drop' 'casestudyhiv
e' '<EOF>' in ddl statement
hive> Drop database casestudyhive;
FAILED: Execution Error, return code 1 from org.apache.hadoop.hive.ql.exec.DDLTa
sk. InvalidOperationException(message:Database casestudyhive is not empty. One o
r more tables exist.)
hive> Drop Database casestudyhive;
FAILED: Execution Error, return code 1 from org.apache.hadoop.hive.ql.exec.DDLTa
sk. InvalidOperationException(message:Database casestudyhive is not empty. One o
r more tables exist.)
hive> show databases;
OK
casestudyhive
default
Time taken: 0.03 seconds, Fetched: 2 row(s)
hive> use casestudyhive;
OK
Time taken: 0.031 seconds
hive> show tables;
OK
nov_info
oct_info
Time taken: 0.037 seconds, Fetched: 2 row(s)
hive> DROP DATABASE casestudyhive CASCADE;
OK
Time taken: 0.312 seconds
hive>
```

(b) Terminating the Cluster
The below query terminates the Database.



Cluster: casestudy-cluster1 Terminating Terminated by user request

Summary

ID: j-3G9Y5KF3F3SQI
Creation date: 2021-04-05 09:33 (UTC+5:30)
Elapsed time: 7 hours, 3 minutes
After last step completes: Cluster waits
Termination protection: Off
Tags: --
Master public DNS: ec2-34-201-127-166.compute-1.amazonaws.com
Connect to the Master Node Using SSH

Configuration details

Release label: emr-5.32.0
Hadoop distribution: Amazon 2.10.1
Applications: Hive 2.3.7, Pig 0.17.0, Hue 4.8.0
Log URI: s3://aws-logs-006530710702-us-east-1/elasticmapreduce/
EMRFS consistent view: Disabled

Thank you!