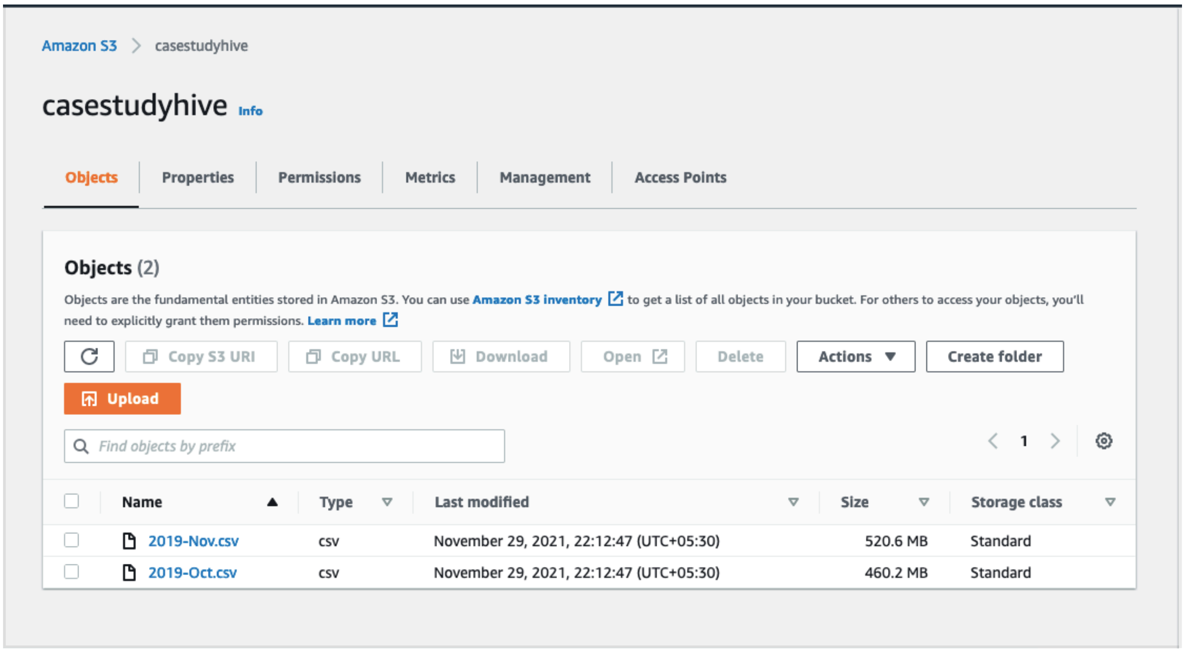
ASSIGNMENT

Copying the data set into the HDFS:

○ Launch an EMR cluster that utilizes the Hive services

First, we upload the files into an s3 bucket.

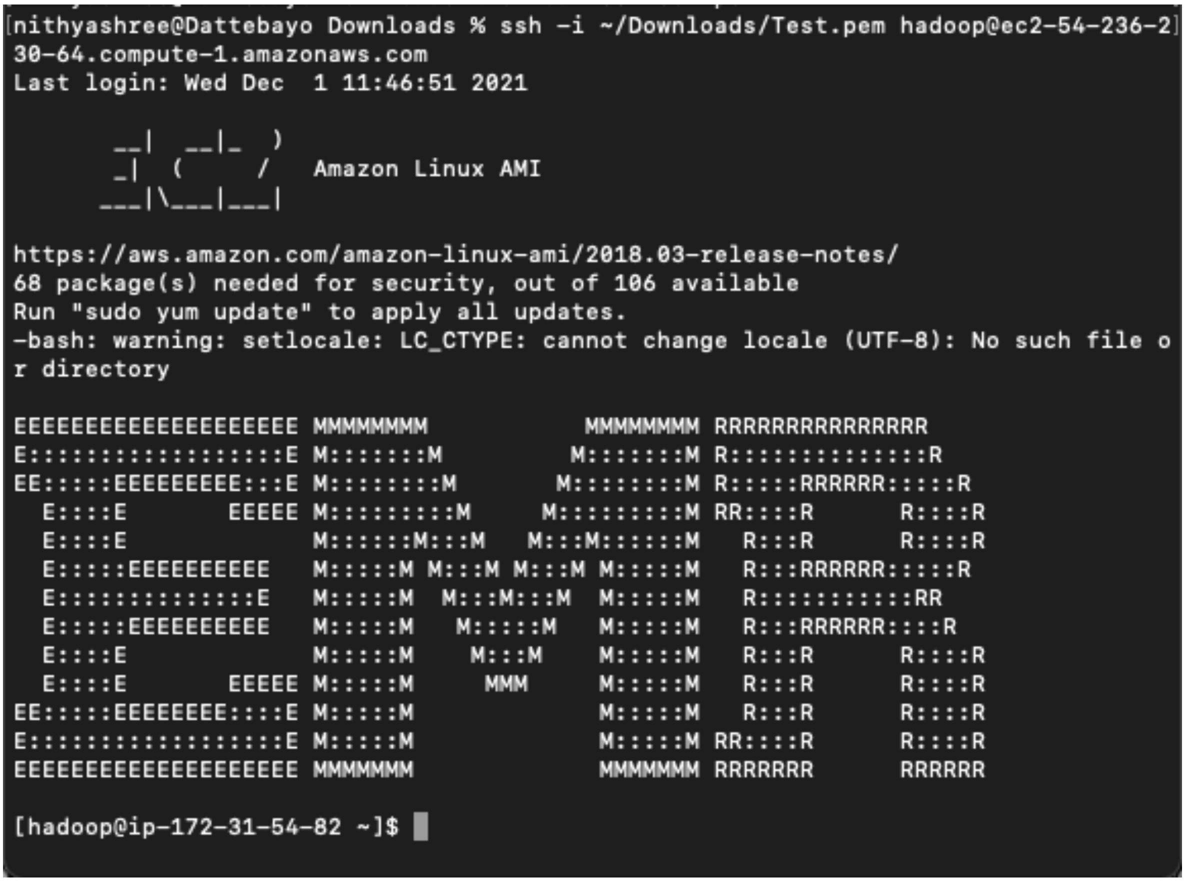


Objects (2)

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	csv	November 29, 2021, 22:12:47 (UTC+05:30)	520.6 MB	Standard
□	2019-Oct.csv	csv	November 29, 2021, 22:12:47 (UTC+05:30)	460.2 MB	Standard

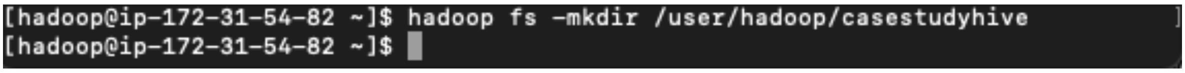
Launch an EMR Cluster and connect to master node through SSH



```
[nithyashree@Dattebayo Downloads % ssh -i ~/Downloads/Test.pem hadoop@ec2-54-236-2]
30-64.compute-1.amazonaws.com
Last login: Wed Dec  1 11:46:51 2021

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
68 package(s) needed for security, out of 106 available
Run "sudo yum update" to apply all updates.
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file o
r directory

EEEEEEEEEEEEEEEEEEEE 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
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-54-82 ~]$ 
```

Create a new directory called casestudyhive to load data

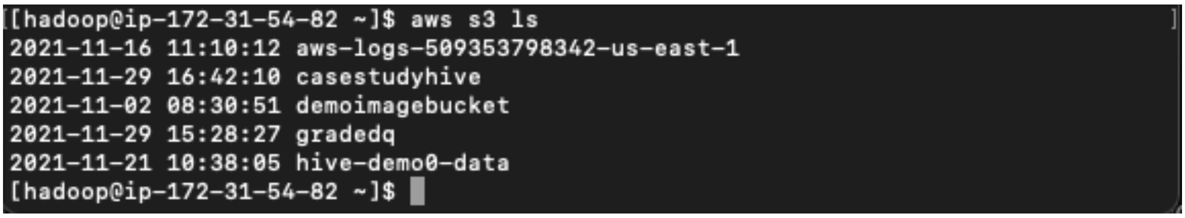
hadoop fs -mkdir /user/hadoop/casestudyhive



```
[hadoop@ip-172-31-54-82 ~]$ hadoop fs -mkdir /user/hadoop/casestudyhive
[hadoop@ip-172-31-54-82 ~]$ 
```

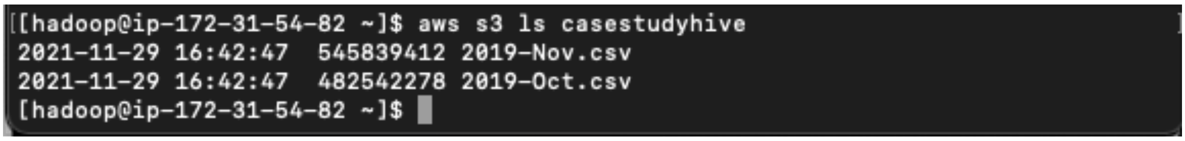
Check s3 list to find Case study and its contents

aws s3 ls



```
[hadoop@ip-172-31-54-82 ~]$ aws s3 ls
2021-11-16 11:10:12 aws-logs-509353798342-us-east-1
2021-11-29 16:42:10 casestudyhive
2021-11-02 08:30:51 demoimagebucket
2021-11-29 15:28:27 gradedq
2021-11-21 10:38:05 hive-demo0-data
[hadoop@ip-172-31-54-82 ~]$ 
```

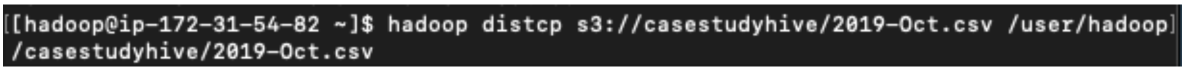
aws s3 ls casestudyhive



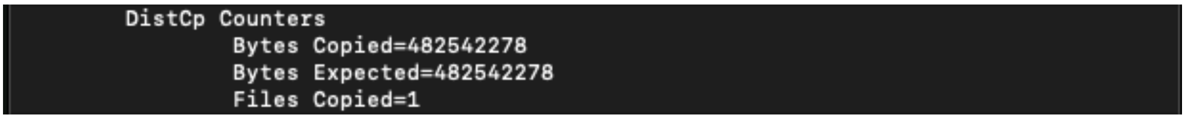
```
[[hadoop@ip-172-31-54-82 ~]$ aws s3 ls casestudyhive
 2021-11-29 16:42:47  545839412 2019-Nov.csv
 2021-11-29 16:42:47  482542278 2019-Oct.csv
 [hadoop@ip-172-31-54-82 ~]$ 
```

- Move the data from the S3 bucket into the HDFS

hadoop distcp s3://casestudyhive/2019-Oct.csv /user/hadoop/casestudyhive/2019-Oct.csv

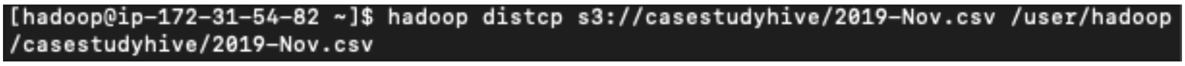


```
[[hadoop@ip-172-31-54-82 ~]$ hadoop distcp s3://casestudyhive/2019-Oct.csv /user/hadoop]
/casestudyhive/2019-Oct.csv
```

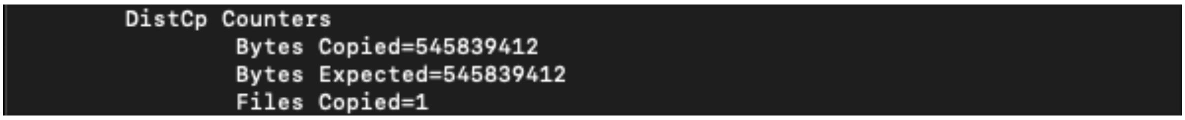


```
        DistCp Counters
                Bytes Copied=482542278
                Bytes Expected=482542278
                Files Copied=1
```

hadoop distcp s3://casestudyhive/2019-Nov.csv /user/hadoop/casestudyhive/2019-Nov.csv



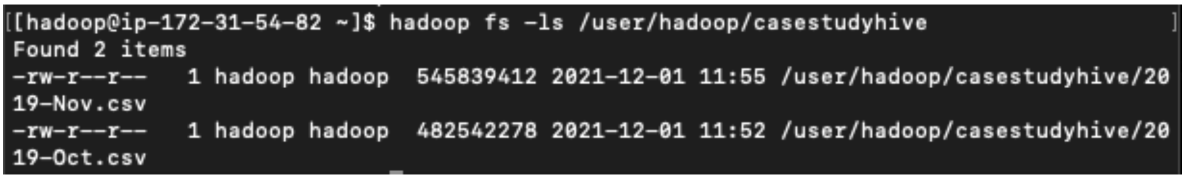
```
[hadoop@ip-172-31-54-82 ~]$ hadoop distcp s3://casestudyhive/2019-Nov.csv /user/hadoop
/casestudyhive/2019-Nov.csv
```



```
        DistCp Counters
                Bytes Copied=545839412
                Bytes Expected=545839412
                Files Copied=1
```

Check directory to make sure the data was loaded

hadoop fs -ls /user/hadoop/casestudyhive

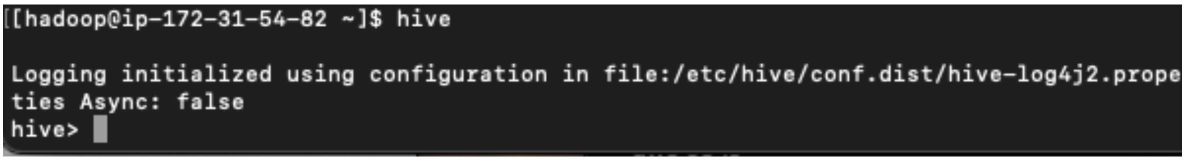


```
[[hadoop@ip-172-31-54-82 ~]$ hadoop fs -ls /user/hadoop/casestudyhive
Found 2 items
-rw-r--r--   1 hadoop hadoop  545839412 2021-12-01 11:55 /user/hadoop/casestudyhive/20
19-Nov.csv
-rw-r--r--   1 hadoop hadoop  482542278 2021-12-01 11:52 /user/hadoop/casestudyhive/20
19-Oct.csv
```

Creating the database and launching Hive queries on your EMR cluster

- Create the structure of your database,

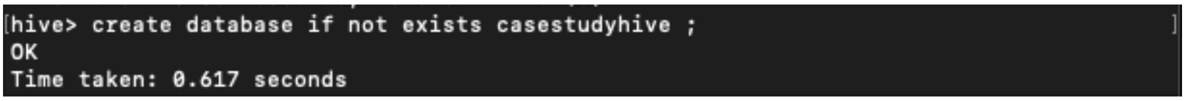
Launch Hive



```
[[hadoop@ip-172-31-54-82 ~]$ hive

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

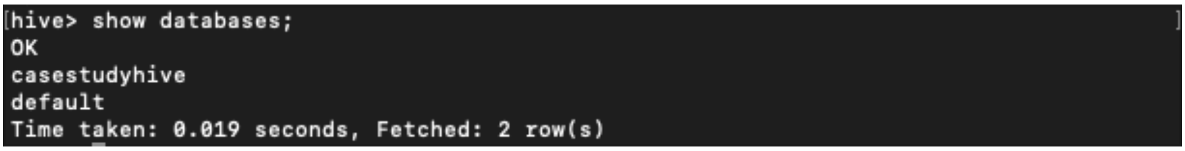
Create database in Hive

create database if not exists casestudyhive;



```
[hive> create database if not exists casestudyhive ;
OK
Time taken: 0.617 seconds
```

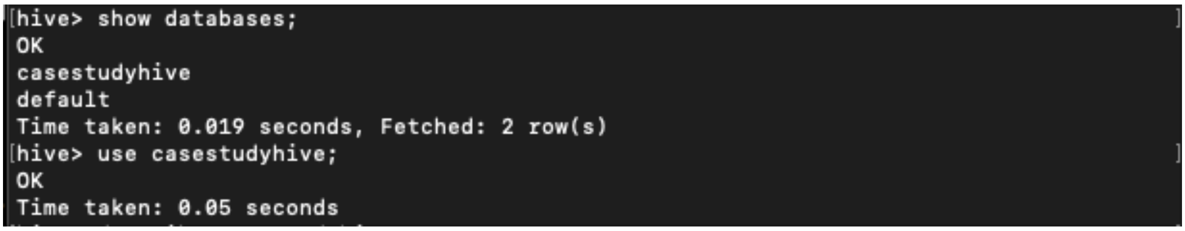
show databases;



```
[hive> show databases;
OK
casestudyhive
default
Time taken: 0.019 seconds, Fetched: 2 row(s)
```

Use the created database for our queries

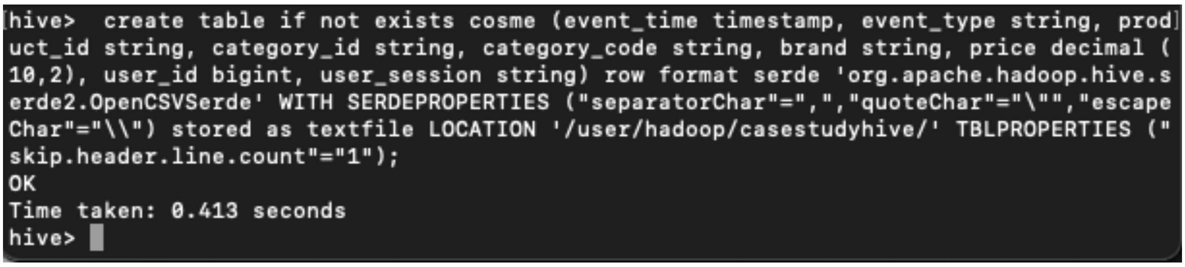
use casestudyhive;



```
[hive> show databases;
OK
casestudyhive
default
Time taken: 0.019 seconds, Fetched: 2 row(s)
[hive> use casestudyhive;
OK
Time taken: 0.05 seconds
```

Create table cosme to load the data

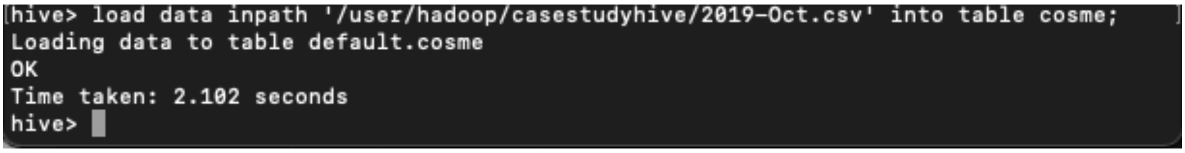
create table if not exists cosme (event_time timestamp, event_type string, product_id string, category_id string, category_code string, brand string, price decimal (10,2), user_id bigint, user_session string) row format serde 'org.apache.hadoop.hive.serde2.OpenCSVSerde' WITH SERDEPROPERTIES ("separatorChar"=",","quoteChar"="\"","escapeChar"="\\") stored as textfile LOCATION '/user/hadoop/casestudyhive/' TBLPROPERTIES ("skip.header.line.count"="1");



```
[hive>  create table if not exists cosme (event_time timestamp, event_type string, prod]
uct_id string, category_id string, category_code string, brand string, price decimal (
10,2), user_id bigint, user_session string) row format serde 'org.apache.hadoop.hive.s
erde2.OpenCSVSerde' WITH SERDEPROPERTIES ("separatorChar"=",","quoteChar"="\"","escape
Char"="\\") stored as textfile LOCATION '/user/hadoop/casestudyhive/' TBLPROPERTIES ("
skip.header.line.count"="1");
OK
Time taken: 0.413 seconds
hive> 
```

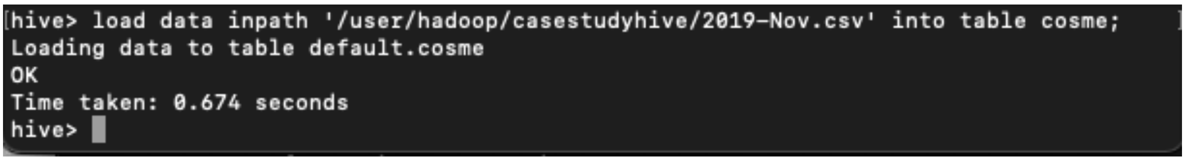
Load data into the table

load data inpath ‘user/hadoop/casestudyhive/2019-Oct.csv’ into table cosme;



```
[hive> load data inpath '/user/hadoop/casestudyhive/2019-Oct.csv' into table cosme;    ]
Loading data to table default.cosme
OK
Time taken: 2.102 seconds
hive> 
```

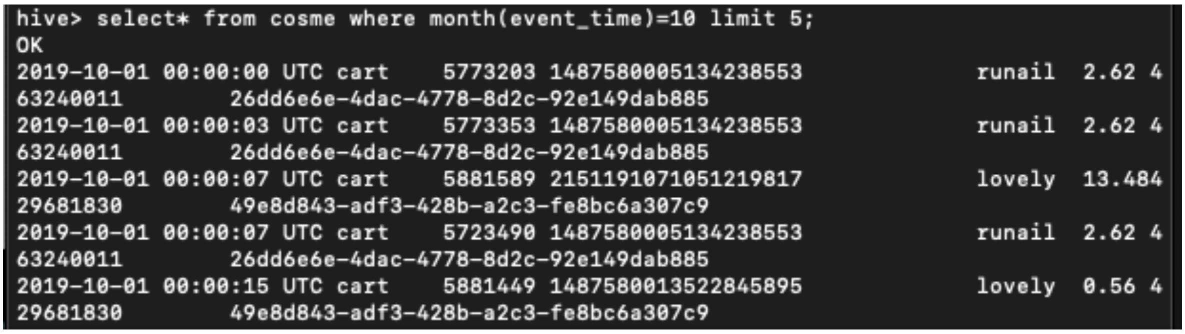
load data inpath ‘user/hadoop/casestudyhive/2019-Nov.csv’ into table cosme;



```
[hive> load data inpath '/user/hadoop/casestudyhive/2019-Nov.csv' into table cosme;    ]
Loading data to table default.cosme
OK
Time taken: 0.674 seconds
hive> 
```

Head of table for October entries

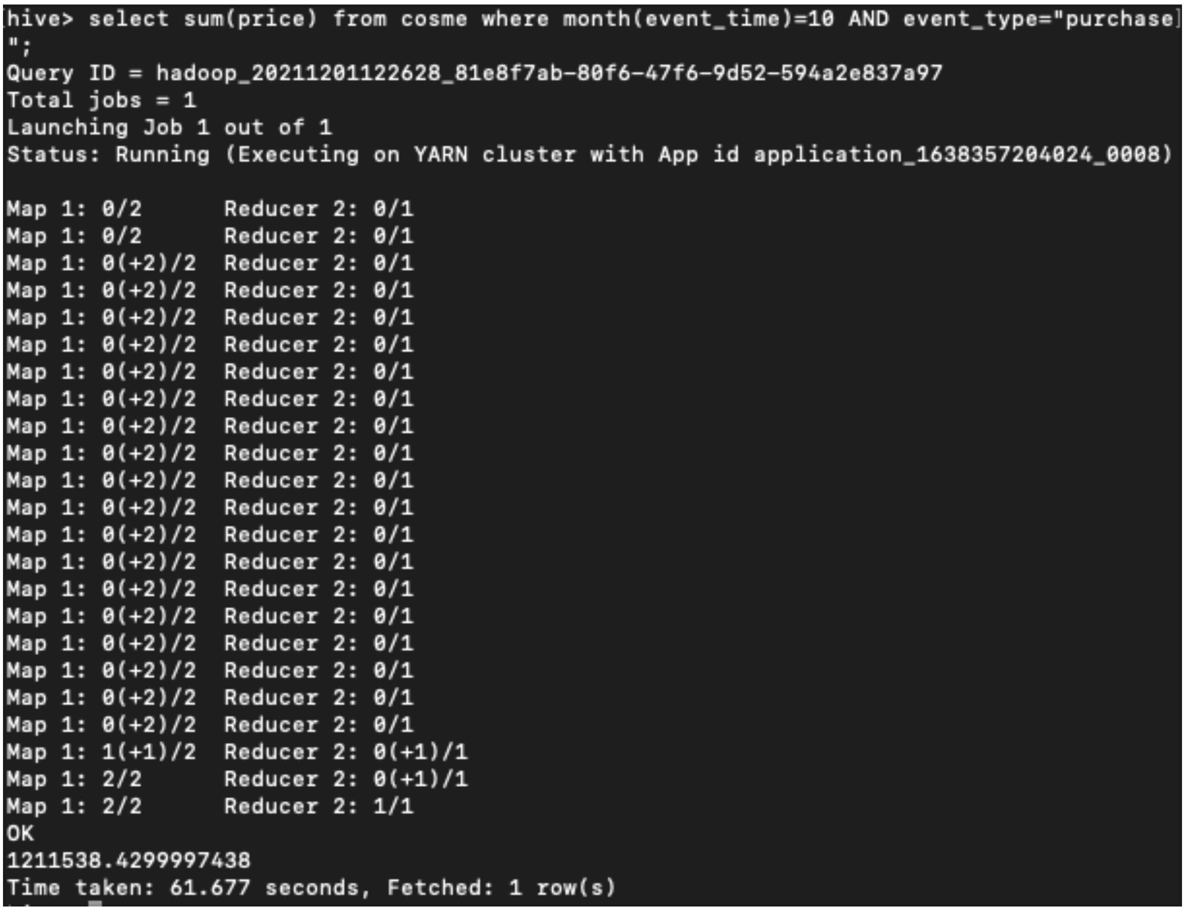
select * from cosme where month(event_time)=10 limit 5;



```
hive> select* from cosme where month(event_time)=10 limit 5;
OK
2019-10-01 00:00:00 UTC cart    5773203 1487580005134238553             runail  2.62 4
63240011        26dd6e6e-4dac-4778-8d2c-92e149dab885
2019-10-01 00:00:03 UTC cart    5773353 1487580005134238553             runail  2.62 4
63240011        26dd6e6e-4dac-4778-8d2c-92e149dab885
2019-10-01 00:00:07 UTC cart    5881589 2151191071051219817             lovely  13.484
29681830        49e8d843-adf3-428b-a2c3-fe8bc6a307c9
2019-10-01 00:00:07 UTC cart    5723490 1487580005134238553             runail  2.62 4
63240011        26dd6e6e-4dac-4778-8d2c-92e149dab885
2019-10-01 00:00:15 UTC cart    5881449 1487580013522845895             lovely  0.56 4
29681830        49e8d843-adf3-428b-a2c3-fe8bc6a307c9
```

Attempt 1st query on the table without partitioning

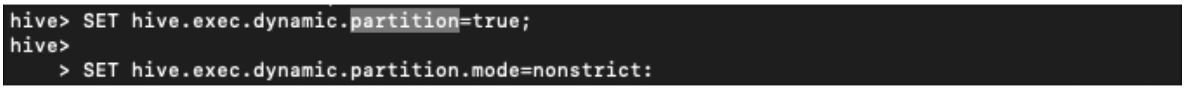
select sum(price_ from cosme where month(event_time)=10 AND event_type=“purchase”;



```
hive> select sum(price) from cosme where month(event_time)=10 AND event_type="purchase
";
Query ID = hadoop_20211201122628_81e8f7ab-80f6-47f6-9d52-594a2e837a97
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1638357204024_0008)

Map 1: 0/2      Reducer 2: 0/1
Map 1: 0/2      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: 2/2      Reducer 2: 0(+1)/1
Map 1: 2/2      Reducer 2: 1/1
OK
1211538.4299997438
Time taken: 61.677 seconds, Fetched: 1 row(s)
```

The above query took 61.677 seconds. Now let us create a table with partitioning to see if this improves the query time. First we set dynamic partition with the following commands.

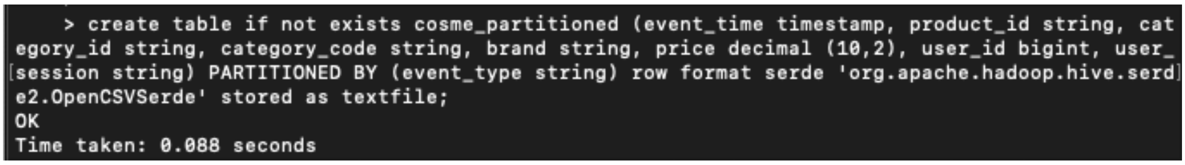
SET hive.exec.dynamic.partition=true;
SET hive.exec.dynamic.partition.mode=nonstrict;



```
hive> SET hive.exec.dynamic.partition=true;
hive>
    > SET hive.exec.dynamic.partition.mode=nonstrict;
```

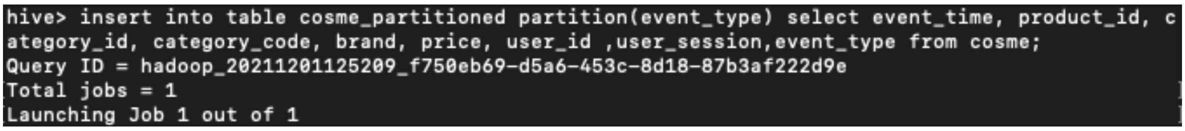
Partitioning

create table if not exists cosme_partitioned (event_time timestamp, product_id string, category_id string, category_code string, brand string, price decimal (10,2), user_id bigint, user_session string) PARTITIONED BY (event_type string) row format serde 'org.apache.hadoop.hive.serde2.OpenCSVSerde' stored as textfile;



```
    > create table if not exists cosme_partitioned (event_time timestamp, product_id string, cat
egory_id string, category_code string, brand string, price decimal (10,2), user_id bigint, user_
session string) PARTITIONED BY (event_type string) row format serde 'org.apache.hadoop.hive.serd
e2.OpenCSVSerde' stored as textfile;
OK
Time taken: 0.088 seconds
```

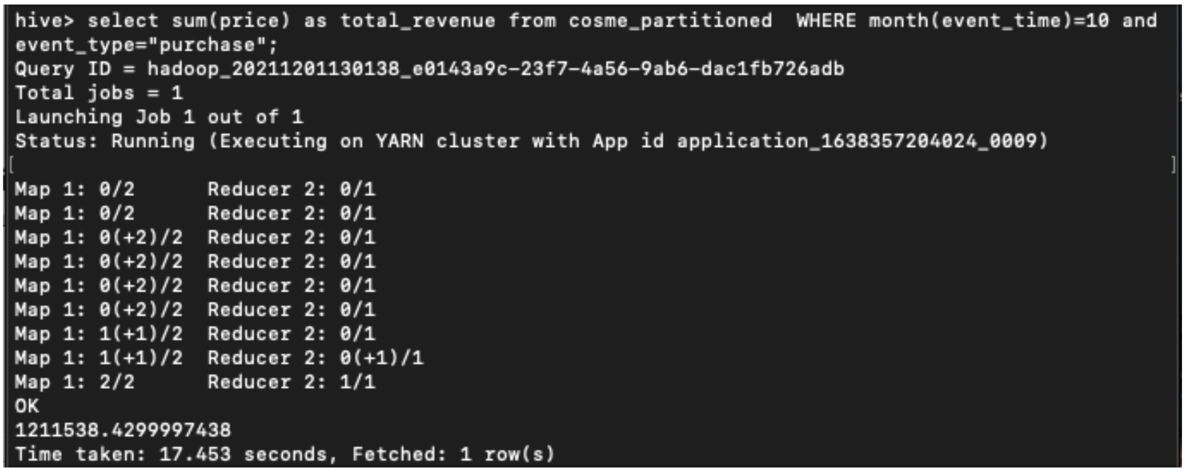
create table if not exists cosme_partitioned (event_time timestamp, product_id string, category_id string, category_code string, brand string, price decimal (10,2), user_id bigint, user_session string) PARTITIONED BY (event_type string) row format serde 'org.apache.hadoop.hive.serde2.OpenCSVSerde' stored as textfile;



```
hive> insert into table cosme_partitioned partition(event_type) select event_time, product_id, c
ategory_id, category_code, brand, price, user_id ,user_session,event_type from cosme;
Query ID = hadoop_20211201125209_f750eb69-d5a6-453c-8d18-87b3af222d9e
Total jobs = 1
Launching Job 1 out of 1
```

insert into table cosme_partitioned partition(event_type) select event_time, product_id, category_id, category_code, brand, price, user_id ,user_session,event_type from cosme;

select sum(price_ from cosme_partitioned where month(event_time)=10 AND event_type="purchase";



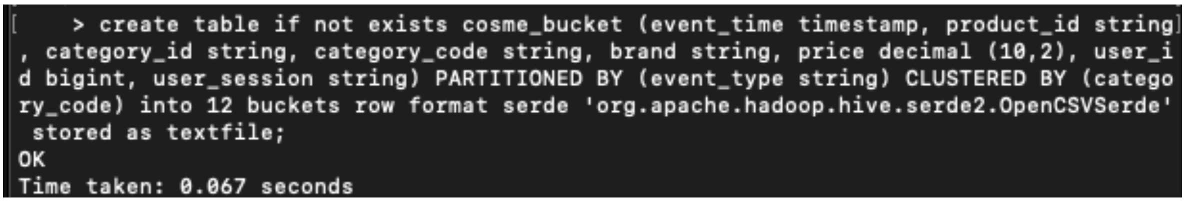
```
hive> select sum(price) as total_revenue from cosme_partitioned  WHERE month(event_time)=10 and
event_type="purchase";
Query ID = hadoop_20211201130138_e0143a9c-23f7-4a56-9ab6-dac1fb726adb
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1638357204024_0009)

Map 1: 0/2      Reducer 2: 0/1
Map 1: 0/2      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: 1/1
OK
1211538.4299997438
Time taken: 17.453 seconds, Fetched: 1 row(s)
```

Running the 1st query with partitioning alone is much faster.

BUCKETING

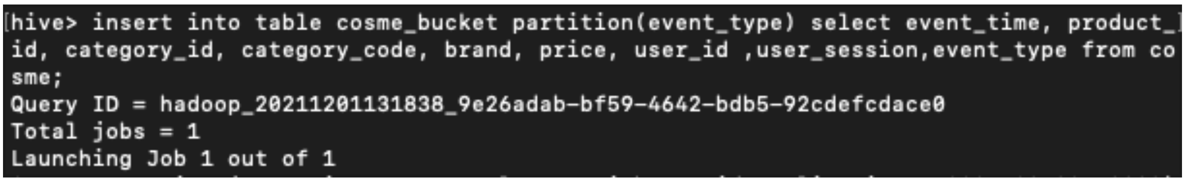
Now we use both partitioning and bucketing to improve query time.

create table if not exists cosme_bucket (event_time timestamp, product_id string, category_id string, category_code string, brand string, price decimal (10,2), user_id bigint, user_session string) PARTITIONED BY (event_type string) CLUSTERED BY (category_code) into 12 buckets row format serde 'org.apache.hadoop.hive.serde2.OpenCSVSerde' stored as textfile;



```
    > create table if not exists cosme_bucket (event_time timestamp, product_id string
, category_id string, category_code string, brand string, price decimal (10,2), user_i
d bigint, user_session string) PARTITIONED BY (event_type string) CLUSTERED BY (catego
ry_code) into 12 buckets row format serde 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
 stored as textfile;
OK
Time taken: 0.067 seconds
```

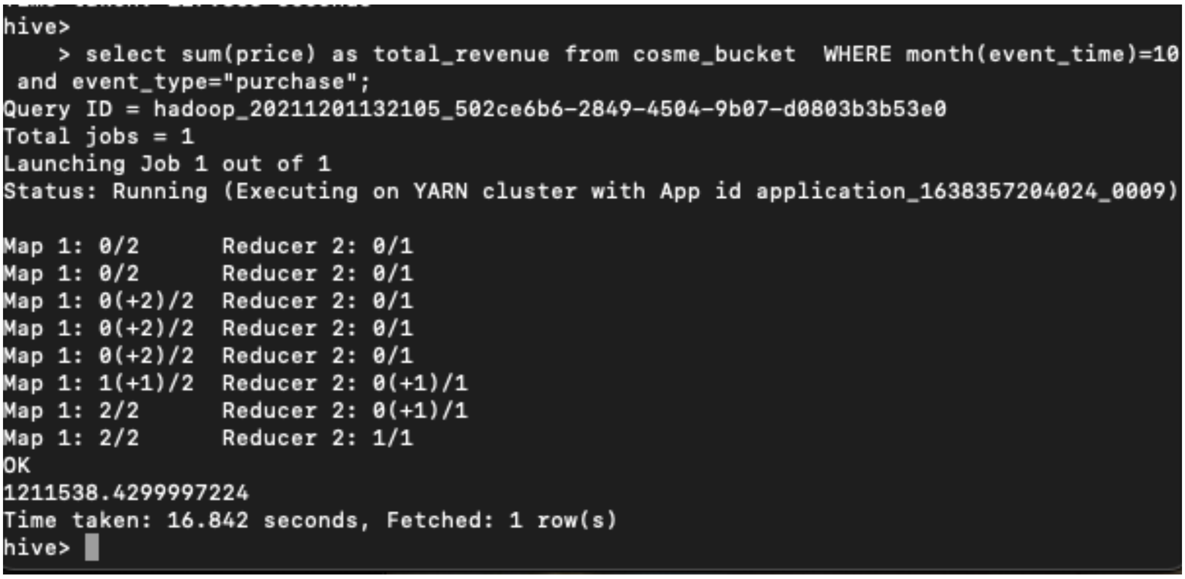
insert into table cosme_bucket partition(event_type) select event_time, product_id, category_id, category_code, brand, price, user_id ,user_session,event_type from cosme;



```
[hive> insert into table cosme_bucket partition(event_type) select event_time, product_
id, category_id, category_code, brand, price, user_id ,user_session,event_type from co
sme;
Query ID = hadoop_20211201131838_9e26adab-bf59-4642-bdb5-92cdefcdace0
Total jobs = 1
Launching Job 1 out of 1
```

Running the 1st query again to compare the query time of all three.

select sum(price) as total_revenue from cosme_bucket WHERE month(event_time)=10 and event_type=“purchase”;



```
hive>
    > select sum(price) as total_revenue from cosme_bucket  WHERE month(event_time)=10
 and event_type="purchase";
Query ID = hadoop_20211201132105_502ce6b6-2849-4504-9b07-d0803b3b53e0
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1638357204024_0009)

Map 1: 0/2      Reducer 2: 0/1
Map 1: 0/2      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
1211538.4299997224
Time taken: 16.842 seconds, Fetched: 1 row(s)
hive>
```

BEFORE PARTITION

select sum(price) as total_revenue from cosme WHERE month(event_time)=10 and event_type=“purchase”;

61.677 seconds

WITH PARTITIONING

select sum(price) as total_revenue from cosme_partitioned WHERE month(event_time)=10 and event_type=“purchase”;

17.453 seconds

select sum(price) as total_revenue from cosme_bucket WHERE month(event_time)=10 and event_type=“purchase”;

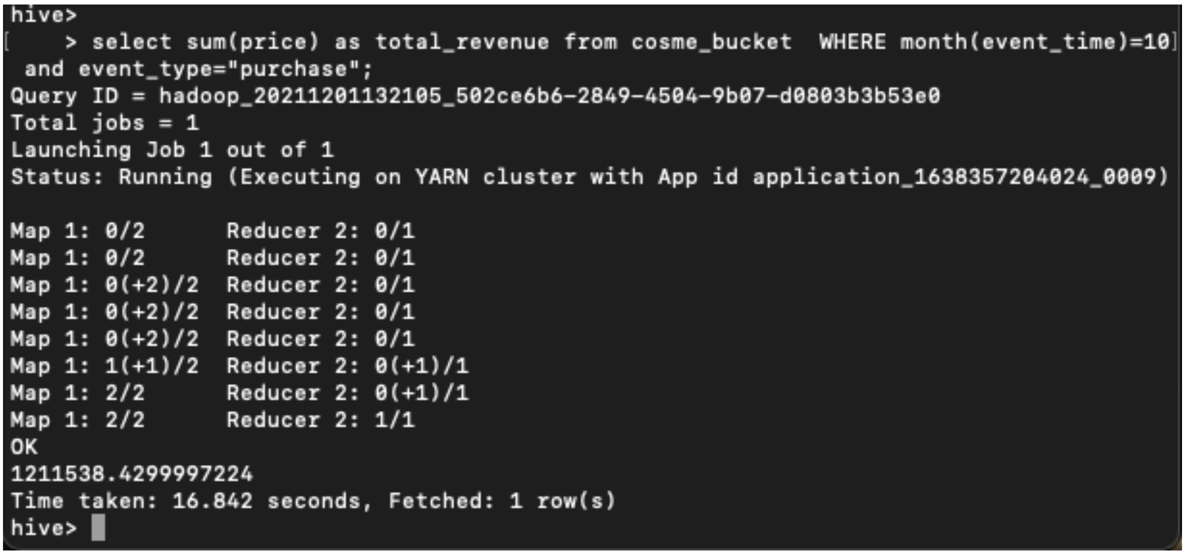
16.842 seconds

Hence, we will be using partitioning and bucketing with cosme_bucket table now on.

QUESTIONS

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

   select sum(price) as total_revenue from cosme_bucket WHERE month(event_time)=10 and event_type="purchase";

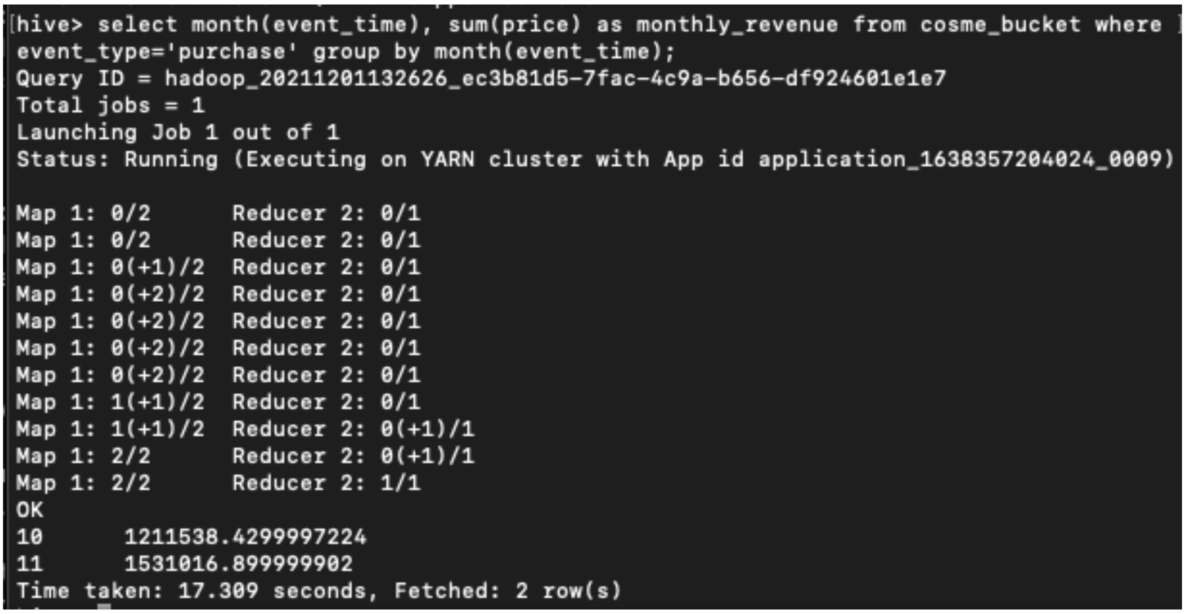


```
hive>
[    > select sum(price) as total_revenue from cosme_bucket  WHERE month(event_time)=10]
 and event_type="purchase";
Query ID = hadoop_20211201132105_502ce6b6-2849-4504-9b07-d0803b3b53e0
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1638357204024_0009)

Map 1: 0/2      Reducer 2: 0/1
Map 1: 0/2      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
1211538.4299997224
Time taken: 16.842 seconds, Fetched: 1 row(s)
hive> █
```

Total revenue from purchases in October is 1211538.4299997224.

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

   select month(event_time), sum(price) as monthly_revenue from cosme_bucket where event_type="purchase" group by month(event_time);



```
[hive> select month(event_time), sum(price) as monthly_revenue from cosme_bucket where ]
 event_type='purchase' group by month(event_time);
 Query ID = hadoop_20211201132626_ec3b81d5-7fac-4c9a-b656-df924601e1e7
 Total jobs = 1
 Launching Job 1 out of 1
 Status: Running (Executing on YARN cluster with App id application_1638357204024_0009)

 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
 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
 10      1211538.4299997224
 11      1531016.899999902
 Time taken: 17.309 seconds, Fetched: 2 row(s)
```

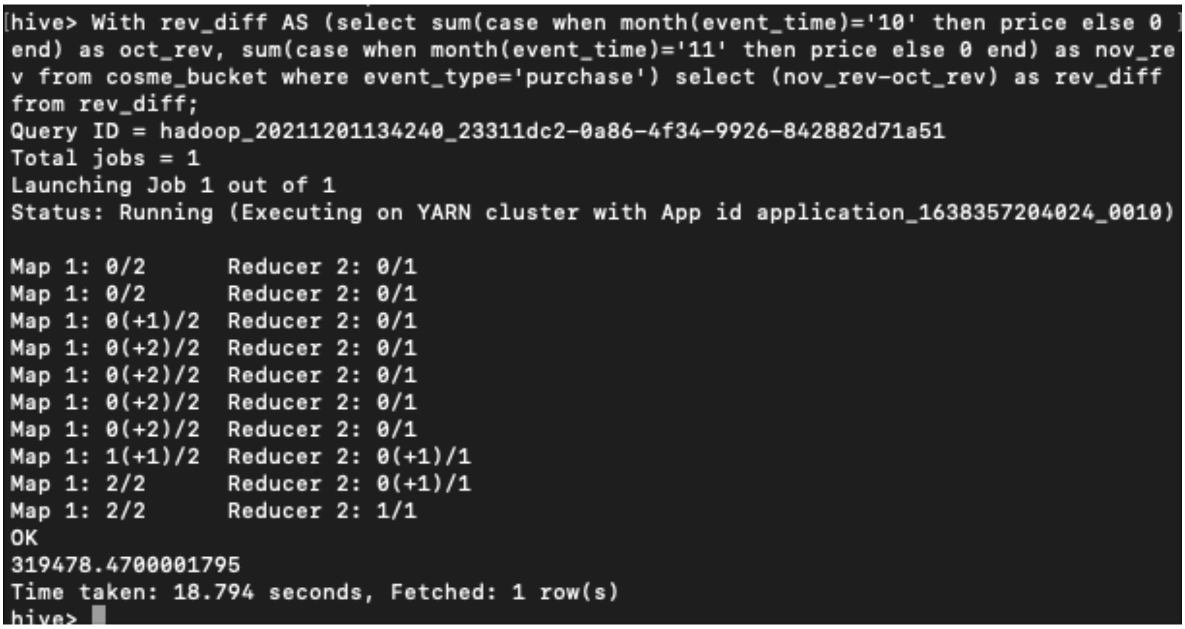
17.309 seconds.

Answer is :

October revenue from purchases: 1211538.4299997224
November revenue from purchases: 1531016.899999902

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

   With rev_diff AS (select sum(case when month(event_time)='10' then price else 0 end) as oct_rev, sum(case when month(event_time)='11' then price else 0 end) as nov_rev from cosme_bucket where event_type='purchase') select (nov_rev-oct_rev) as rev_diff from rev_diff;

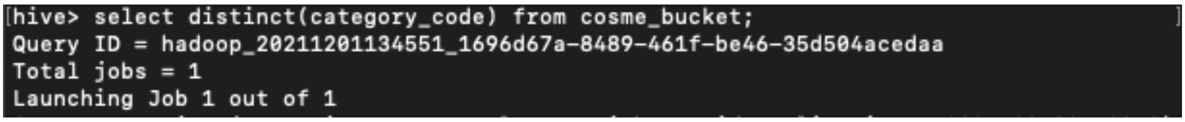


```
[hive> With rev_diff AS (select sum(case when month(event_time)='10' then price else 0 ]
end) as oct_rev, sum(case when month(event_time)='11' then price else 0 end) as nov_re
v from cosme_bucket where event_type='purchase') select (nov_rev-oct_rev) as rev_diff
from rev_diff;
Query ID = hadoop_20211201134240_23311dc2-0a86-4f34-9926-842882d71a51
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1638357204024_0010)

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
319478.4700001795
Time taken: 18.794 seconds, Fetched: 1 row(s)
hive>
```

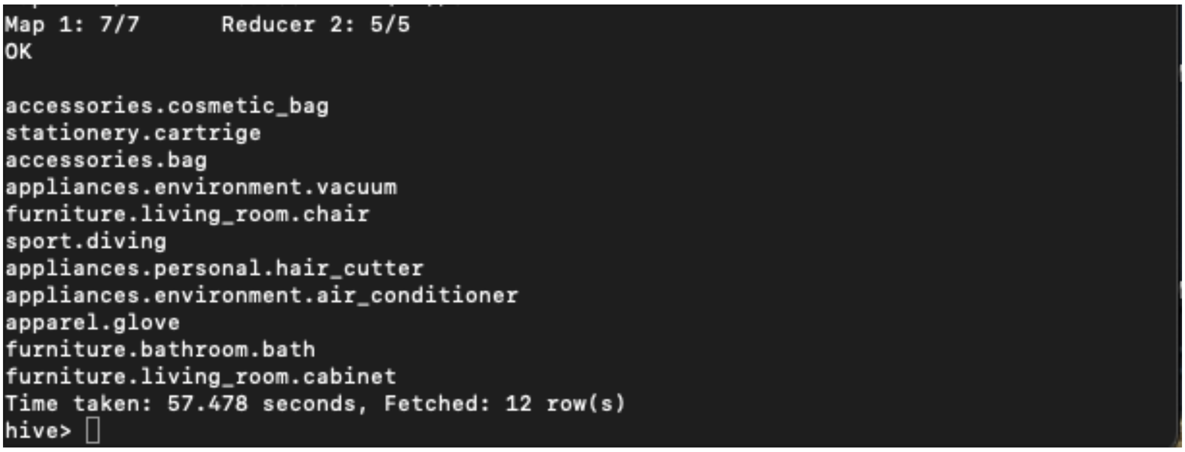
18.794 seconds.

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

   select distinct(category_code) from cosme_bucket;



```
[hive> select distinct(category_code) from cosme_bucket;                              ]
Query ID = hadoop_20211201134551_1696d67a-8489-461f-be46-35d504acedaa
Total jobs = 1
Launching Job 1 out of 1
```



```
Map 1: 7/7      Reducer 2: 5/5
OK

accessories.cosmetic_bag
stationery.cartrige
accessories.bag
appliances.environment.vacuum
furniture.living_room.chair
sport.diving
appliances.personal.hair_cutter
appliances.environment.air_conditioner
apparel.glove
furniture.bathroom.bath
furniture.living_room.cabinet
Time taken: 57.478 seconds, Fetched: 12 row(s)
hive> 
```

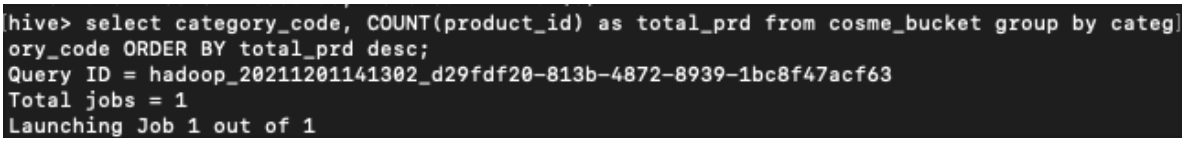
57.478 seconds.

The total distinct categories are 6 :

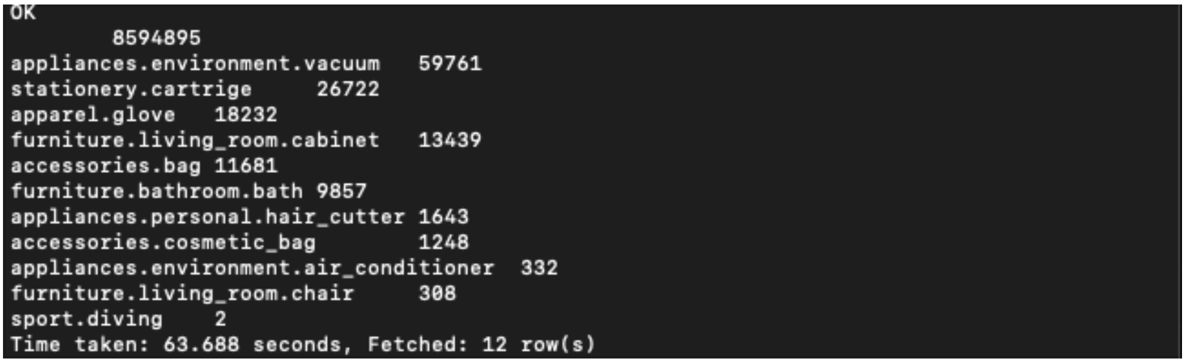
- Accessories
- Stationery
- Accessories
- Appliances
- Furniture
- Sport
- Apparel

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

    select category_code, COUNT(product_id) as total_prd from cosme_bucket group by category_code ORDER BY total_prd desc;



```
hive> select category_code, COUNT(product_id) as total_prd from cosme_bucket group by categ
ory_code ORDER BY total_prd desc;
Query ID = hadoop_20211201141302_d29fdf20-813b-4872-8939-1bc8f47acf63
Total jobs = 1
Launching Job 1 out of 1
```



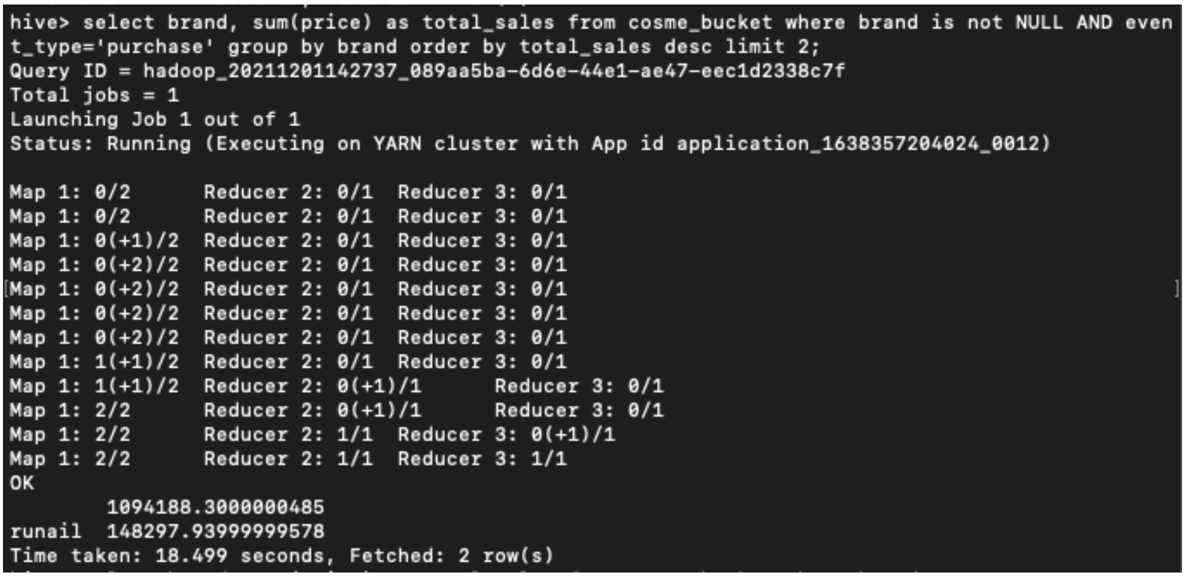
```
OK
        8594895
appliances.environment.vacuum   59761
stationery.cartrige     26722
apparel.glove   18232
furniture.living_room.cabinet   13439
accessories.bag 11681
furniture.bathroom.bath 9857
appliances.personal.hair_cutter 1643
accessories.cosmetic_bag        1248
appliances.environment.air_conditioner  332
furniture.living_room.chair     308
sport.diving    2
Time taken: 63.688 seconds, Fetched: 12 row(s)
```

63.688 seconds.

The total number of products available under each category are as listed above. The highest being Vacuums under the Appliances category and the least being Diving related products under the Sport category.

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

   select brand, sum(price) as total_sales from cosme_bucket where brand<>NULL AND event_type='purchase' group by brand order by total_sales desc limit 2;



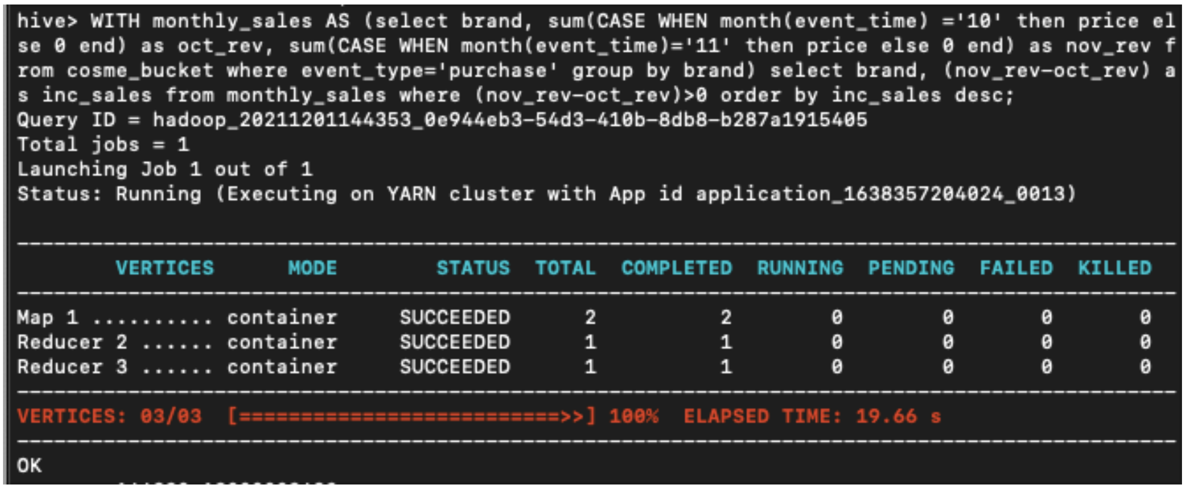
```
hive> select brand, sum(price) as total_sales from cosme_bucket where brand is not NULL AND even
t_type='purchase' group by brand order by total_sales desc limit 2;
Query ID = hadoop_20211201142737_089aa5ba-6d6e-44e1-ae47-eec1d2338c7f
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1638357204024_0012)

Map 1: 0/2      Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0/2      Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0(+1)/2  Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0(+2)/2  Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 1(+1)/2  Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 1(+1)/2  Reducer 2: 0(+1)/1      Reducer 3: 0/1
Map 1: 2/2      Reducer 2: 0(+1)/1      Reducer 3: 0/1
Map 1: 2/2      Reducer 2: 1/1  Reducer 3: 0(+1)/1
Map 1: 2/2      Reducer 2: 1/1  Reducer 3: 1/1
OK
        1094188.3000000485
runail  148297.93999999578
Time taken: 18.499 seconds, Fetched: 2 row(s)
```

18.499 seconds.

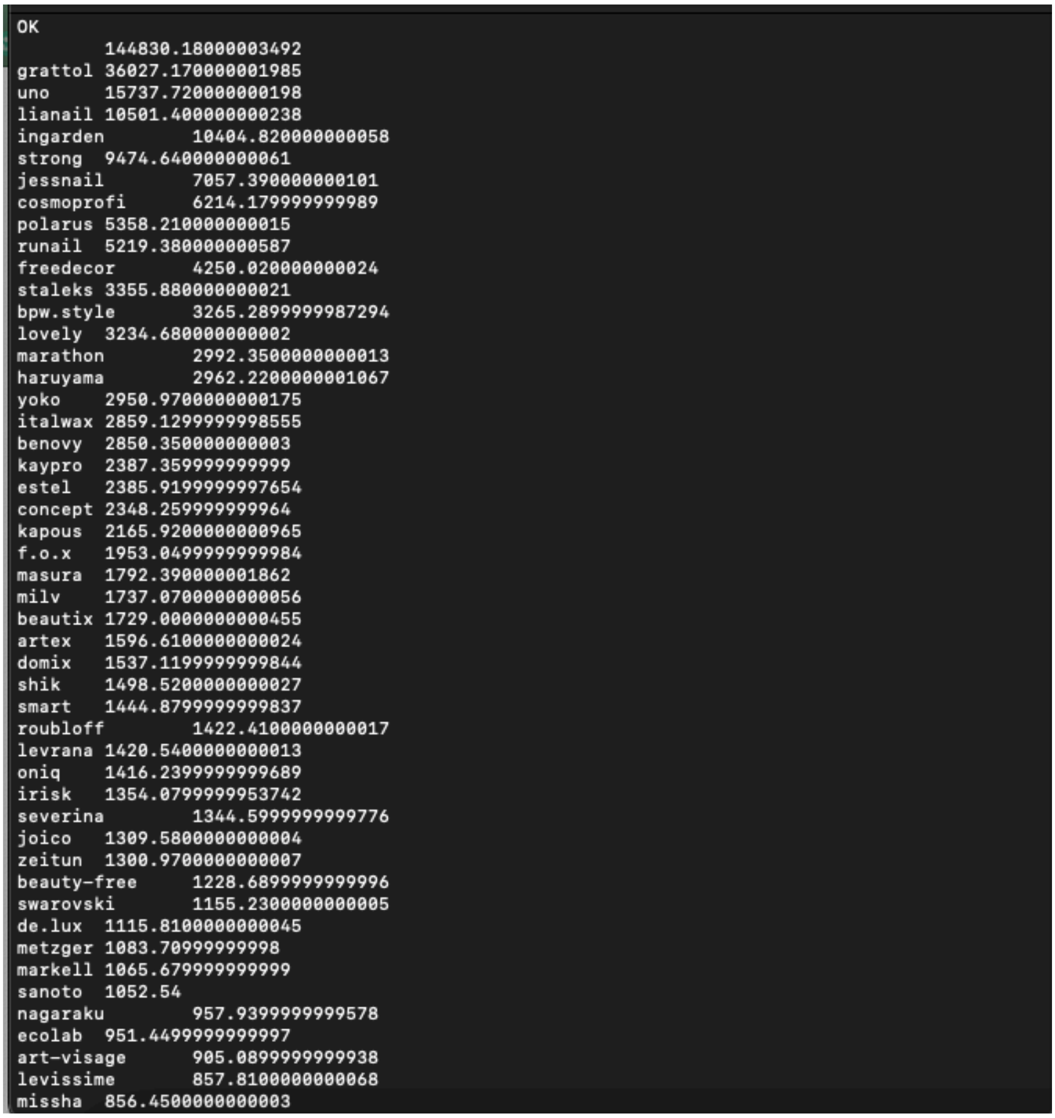
The brand with the maximum sales in October and November combines is Runail with 148297.94 in sales.

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

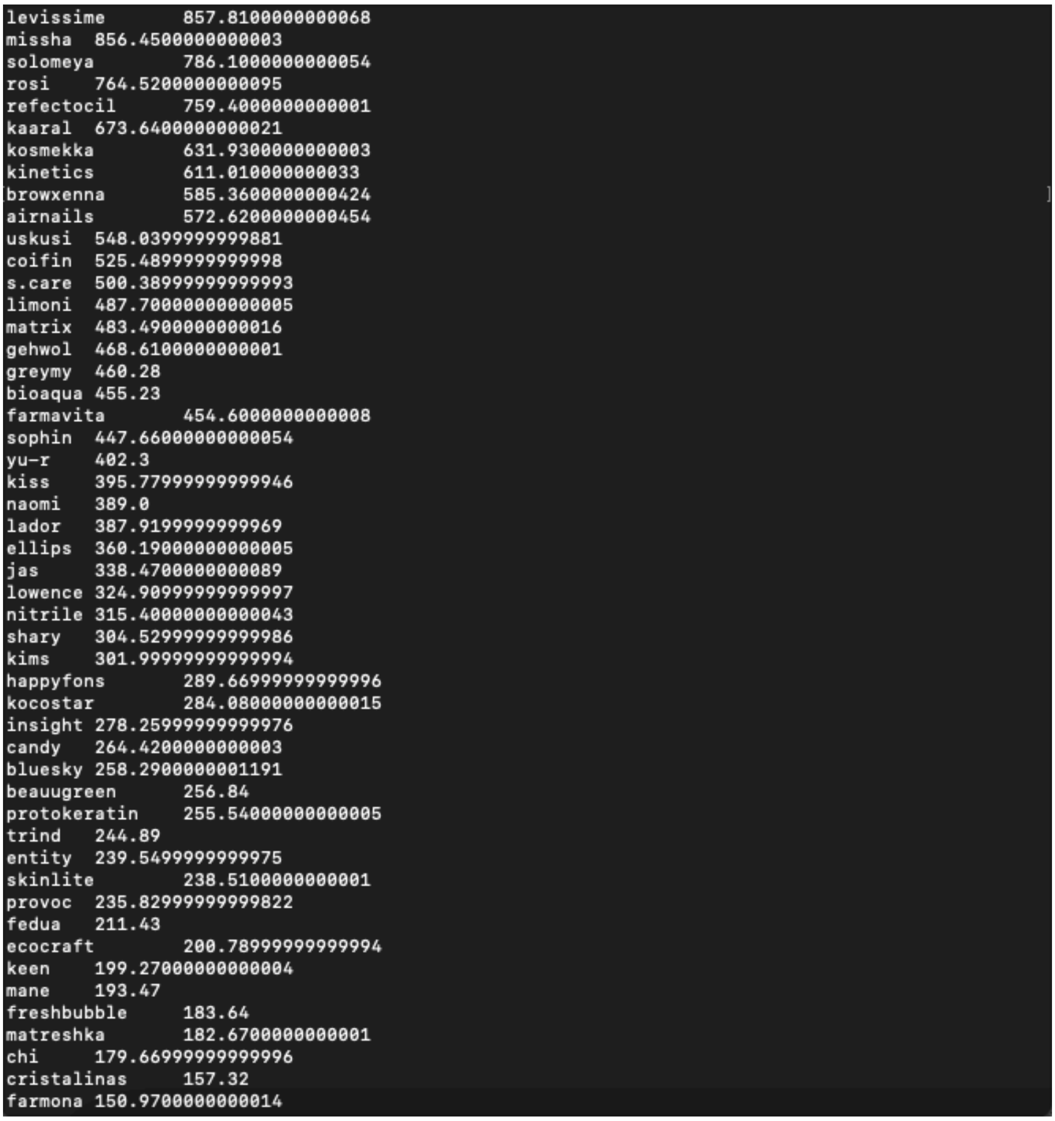
   WITH monthly_sales AS (select brand, sum(CASE WHEN month(event_time) ='10' then price else 0 end) as oct_rev, sum(CASE WHEN month(event_time)='11' then price else 0 end) as nov_rev from cosme_bucket where event_type='purchase' group by brand) select brand,nov_rev, oct_rev, (nov_rev-oct_rev) as inc_sales from monthly_sales where (nov_rev-oct_rev)>0 order by inc_sales desc;



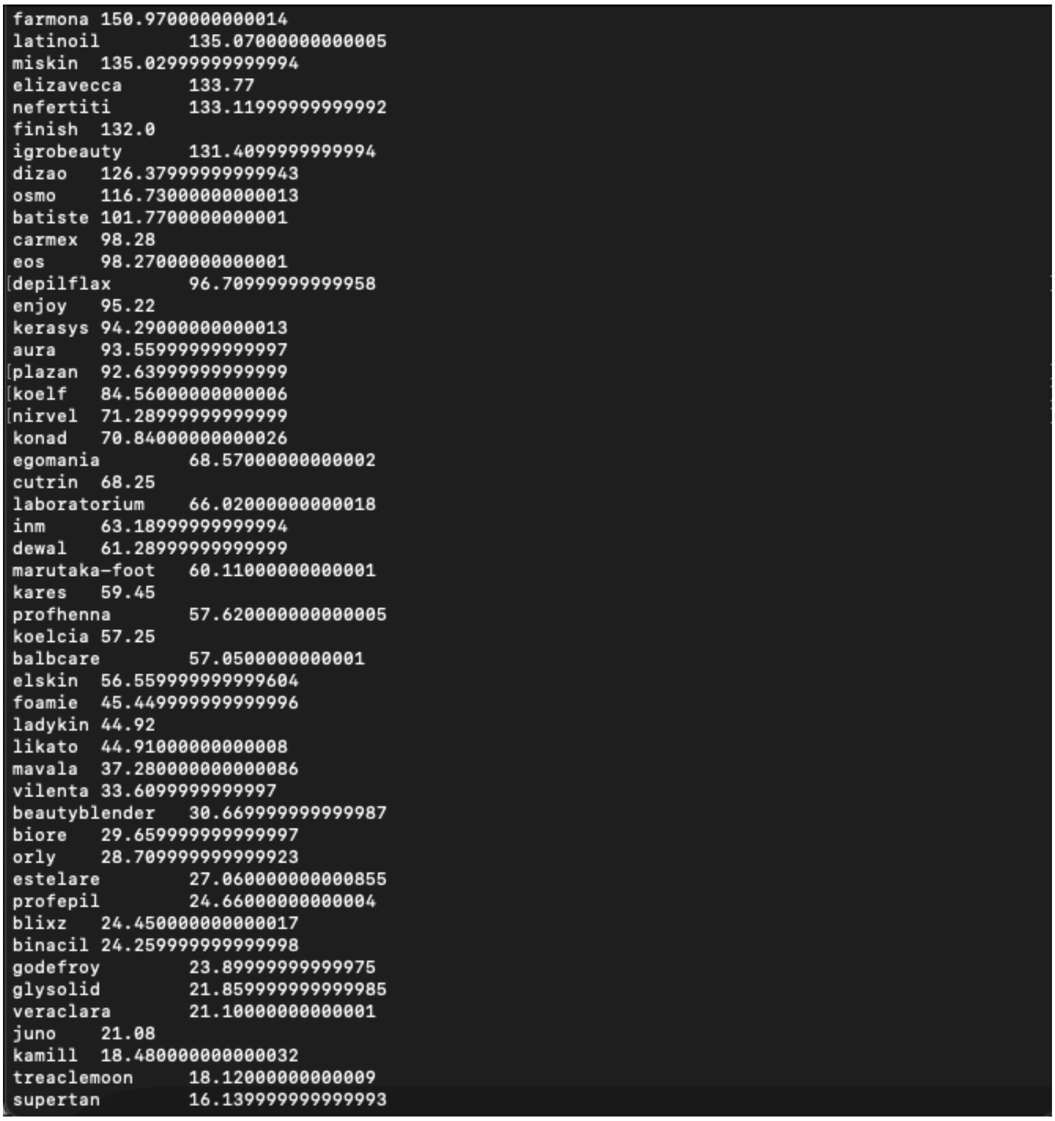
```
hive> WITH monthly_sales AS (select brand, sum(CASE WHEN month(event_time) ='10' then price el
se 0 end) as oct_rev, sum(CASE WHEN month(event_time)='11' then price else 0 end) as nov_rev f
rom cosme_bucket where event_type='purchase' group by brand) select brand, (nov_rev-oct_rev) a
s inc_sales from monthly_sales where (nov_rev-oct_rev)>0 order by inc_sales desc;
Query ID = hadoop_20211201144353_0e944eb3-54d3-410b-8db8-b287a1915405
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1638357204024_0013)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      2          2        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: 19.66 s
----------------------------------------------------------------------------------------------
OK
```



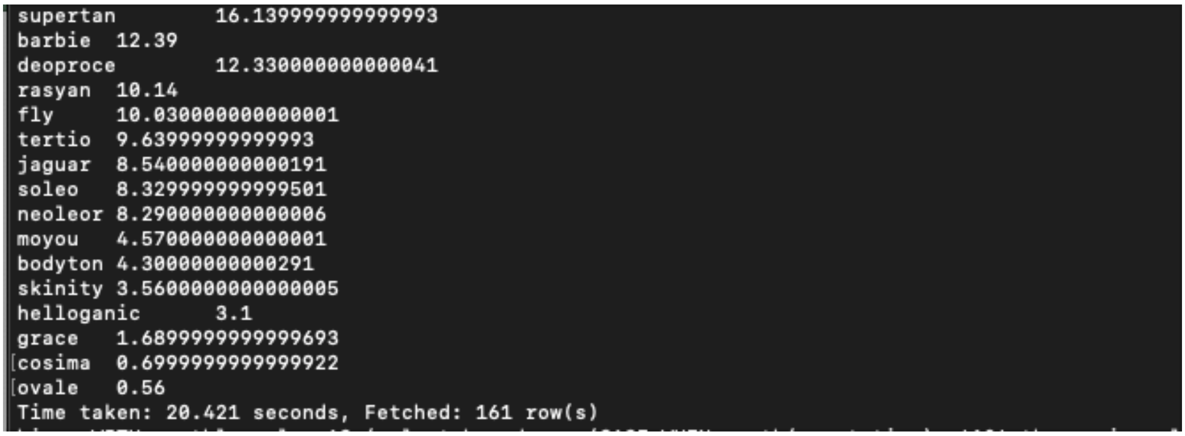
```
OK
        144830.18000003492
grattol 36027.170000001985
uno     15737.720000000198
lianail 10501.400000000238
ingarden        10404.820000000058
strong  9474.640000000061
jessnail        7057.390000000101
cosmoprofi      6214.179999999989
polarus 5358.210000000015
runail  5219.380000000587
freedecor       4250.020000000024
staleks 3355.880000000021
bpw.style       3265.2899999987294
lovely  3234.680000000002
marathon        2992.3500000000013
haruyama        2962.2200000001067
yoko    2950.9700000000175
italwax 2859.1299999998555
benovy  2850.350000000003
kaypro  2387.359999999999
estel   2385.9199999997654
concept 2348.259999999964
kapous  2165.9200000000965
f.o.x   1953.0499999999984
masura  1792.390000001862
milv    1737.0700000000056
beautix 1729.0000000000455
artex   1596.6100000000024
domix   1537.1199999999844
shik    1498.5200000000027
smart   1444.8799999999837
roubloff        1422.4100000000017
levrana 1420.5400000000013
oniq    1416.2399999999689
irisk   1354.0799999953742
severina        1344.5999999999776
joico   1309.5800000000004
zeitun  1300.9700000000007
beauty-free     1228.6899999999996
swarovski       1155.2300000000005
de.lux  1115.8100000000045
metzger 1083.70999999998
markell 1065.679999999999
sanoto  1052.54
nagaraku        957.9399999999578
ecolab  951.449999999997
art-visage      905.0899999999938
levissime       857.8100000000068
missha  856.450000000003
```



```
levissime       857.8100000000068
missha  856.4500000000003
solomeya        786.1000000000054
rosi    764.5200000000095
refectocil      759.4000000000001
kaaral  673.6400000000021
kosmekka        631.9300000000003
kinetics        611.010000000033
browxenna       585.3600000000424
airnails        572.6200000000454
uskusi  548.0399999999881
coifin  525.4899999999998
s.care  500.38999999999993
limoni  487.70000000000005
matrix  483.4900000000016
gehwol  468.6100000000001
greymy  460.28
bioaqua 455.23
farmavita       454.6000000000008
sophin  447.66000000000054
yu-r    402.3
kiss    395.77999999999946
naomi   389.0
lador   387.9199999999969
ellips  360.19000000000005
jas     338.4700000000089
lowence 324.90999999999997
nitrile 315.40000000000043
shary   304.52999999999986
kims    301.99999999999994
happyfons       289.66999999999996
kocostar        284.08000000000015
insight 278.25999999999976
candy   264.4200000000003
bluesky 258.2900000001191
beauugreen      256.84
protokeratin    255.54000000000005
trind   244.89
entity  239.5499999999975
skinlite        238.5100000000001
provoc  235.82999999999822
fedua   211.43
ecocraft        200.78999999999994
keen    199.27000000000004
mane    193.47
freshbubble     183.64
matreshka       182.6700000000001
chi     179.66999999999996
cristalinas     157.32
farmona 150.9700000000014
```



```
farmona 150.9700000000014
latinoil        135.07000000000005
miskin  135.02999999999994
elizavecca      133.77
nefertiti       133.11999999999992
finish  132.0
igrobeauty      131.40999999999994
dizao   126.37999999999943
osmo    116.73000000000013
batiste 101.7700000000001
carmex  98.28
eos     98.27000000000001
[depilflax      96.70999999999958
enjoy   95.22
kerasys 94.29000000000013
aura    93.55999999999997
[plazan 92.63999999999999
[koelf  84.56000000000006
[nirvel 71.28999999999999
konad   70.84000000000026
egomania        68.57000000000002
cutrin  68.25
laboratorium    66.02000000000018
inm     63.18999999999994
dewal   61.28999999999999
marutaka-foot   60.11000000000001
kares   59.45
profhenna       57.620000000000005
koelcia 57.25
balbcare        57.0500000000001
elskin  56.559999999999604
foamie  45.449999999999996
ladykin 44.92
likato  44.91000000000008
mavala  37.280000000000086
vilenta 33.6099999999997
beautyblender   30.669999999999987
biore   29.659999999999997
orly    28.709999999999923
estelare        27.060000000000855
profepil        24.66000000000004
blixz   24.450000000000017
binacil 24.259999999999998
godefroy        23.899999999999975
glysolid        21.859999999999985
veraclara       21.10000000000001
juno    21.08
kamill  18.480000000000032
treaclemoon     18.12000000000009
supertan        16.139999999999993
```



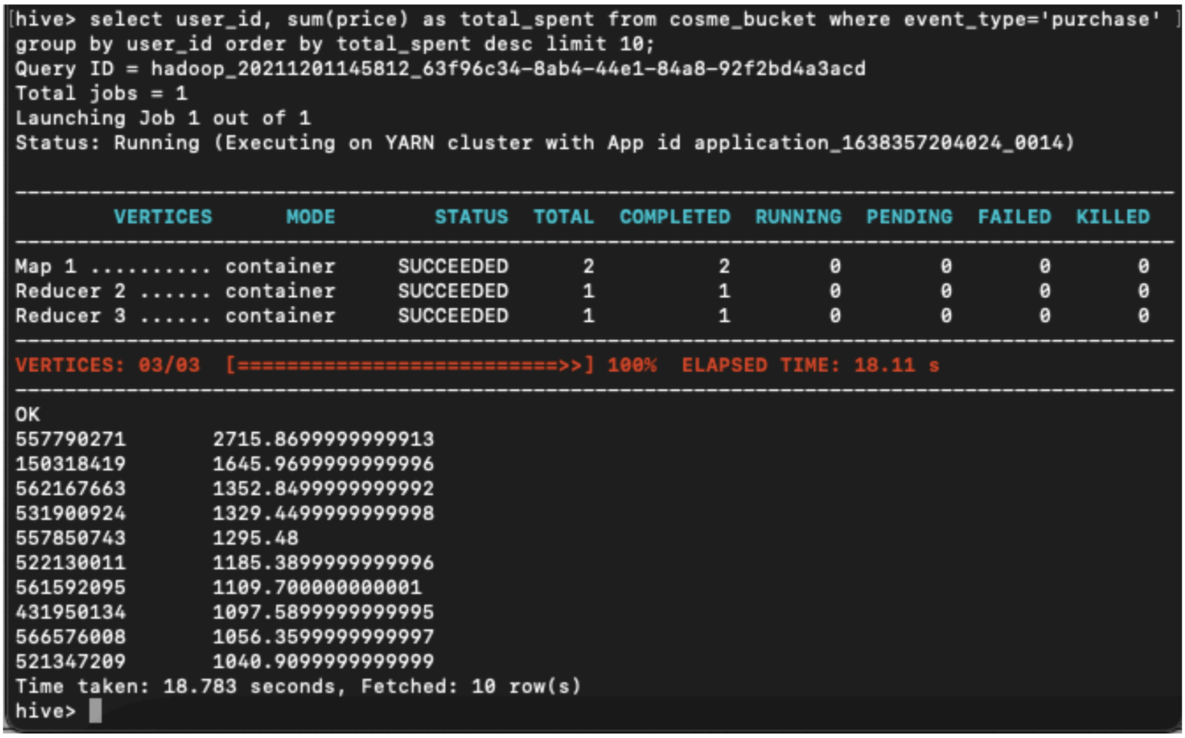
```
supertan        16.139999999999993
barbie  12.39
deoproce        12.330000000000041
rasyan  10.14
fly     10.030000000000001
tertio  9.63999999999993
jaguar  8.540000000000191
soleo   8.329999999999501
neoleor 8.290000000000006
moyou   4.570000000000001
bodyton 4.30000000000291
skinity 3.5600000000000005
helloganic      3.1
grace   1.6899999999999693
[cosima 0.6999999999999922
[ovale  0.56
Time taken: 20.421 seconds, Fetched: 161 row(s)
```

161 brands increased their sales from October to November. Grattol had the highest increase in sales and Ovale had the least increase in sales.

20.421 seconds.

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.

select user_id, sum(price) as total_spent from cosme_bucket where event_type='purchase' group by user_id order by total_spent desc limit 10;



```
[hive> select user_id, sum(price) as total_spent from cosme_bucket where event_type='purchase' ]
group by user_id order by total_spent desc limit 10;
Query ID = hadoop_20211201145812_63f96c34-8ab4-44e1-84a8-92f2bd4a3acd
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1638357204024_0014)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      2          2        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: 18.11 s
----------------------------------------------------------------------------------------------
OK
557790271       2715.8699999999913
150318419       1645.9699999999996
562167663       1352.8499999999992
531900924       1329.4499999999998
557850743       1295.48
522130011       1185.3899999999996
561592095       1109.700000000001
431950134       1097.5899999999995
566576008       1056.3599999999997
521347209       1040.9099999999999
Time taken: 18.783 seconds, Fetched: 10 row(s)
hive> 
```

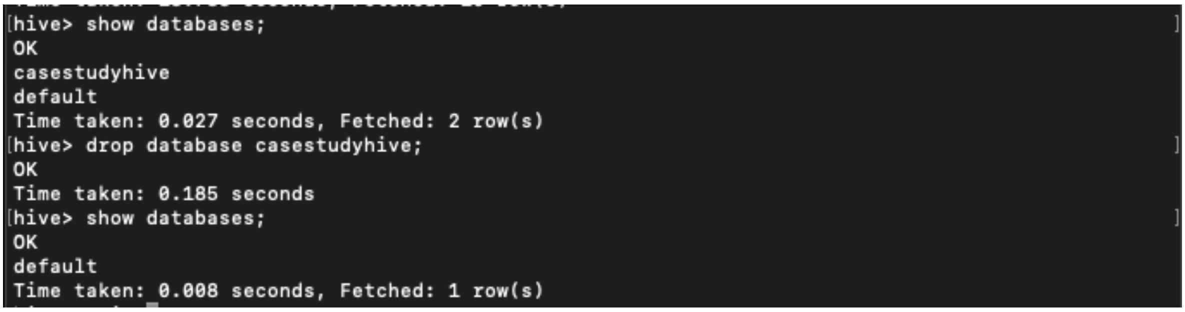
18.783 seconds.

The user ids who have spent the most with the company are as above and can be rewarded as per the Golden Customer plan.

Cleaning up

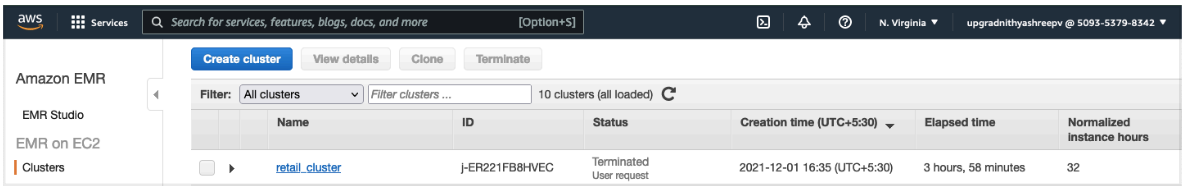
- Drop your database

  drop database casestudyhive;



```
[hive> show databases;                                                                         ]
OK
casestudyhive
default
Time taken: 0.027 seconds, Fetched: 2 row(s)
[hive> drop database casestudyhive;                                                            ]
OK
Time taken: 0.185 seconds
[hive> show databases;                                                                         ]
OK
default
Time taken: 0.008 seconds, Fetched: 1 row(s)
```

- Terminate your cluster



Filter: All clusters | Filter clusters ... | 10 clusters (all loaded)

		Name	ID	Status	Creation time (UTC+5:30)	Elapsed time	Normalized instance hours
□	▸	retail_cluster	j-ER221FB8HVEC	Terminated User request	2021-12-01 16:35 (UTC+5:30)	3 hours, 58 minutes	32