upGrad

#LifeKoKaroLift

Hive Case Study:

Harshita Sharma & Ranjiv Abraham

31/08/2020

Problem Statement

upGrad

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 with in an e-retail company. You are required to provide answers to the questions given below.

1. Find the total revenue generated due to purchases made in October.
2. Write a query to yield the total sum of purchases per month in a single output.
3. Write a query to find the change in revenue generated due to purchases from October to November.
4. Find distinct categories of products.
5. Find the total number of products available under each category.
6. Which brand had the maximum sales in October and November combined?
7. Which brands increased their sales from October to November?
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.

upGrad

Index

1 Importing the data to HDFS

2 Creating Hive tables

3 Analysis using the queries

4 Optimisation techniques

5 Optimized Query Output

6 Cleaning up

7 Glossaries and Notes

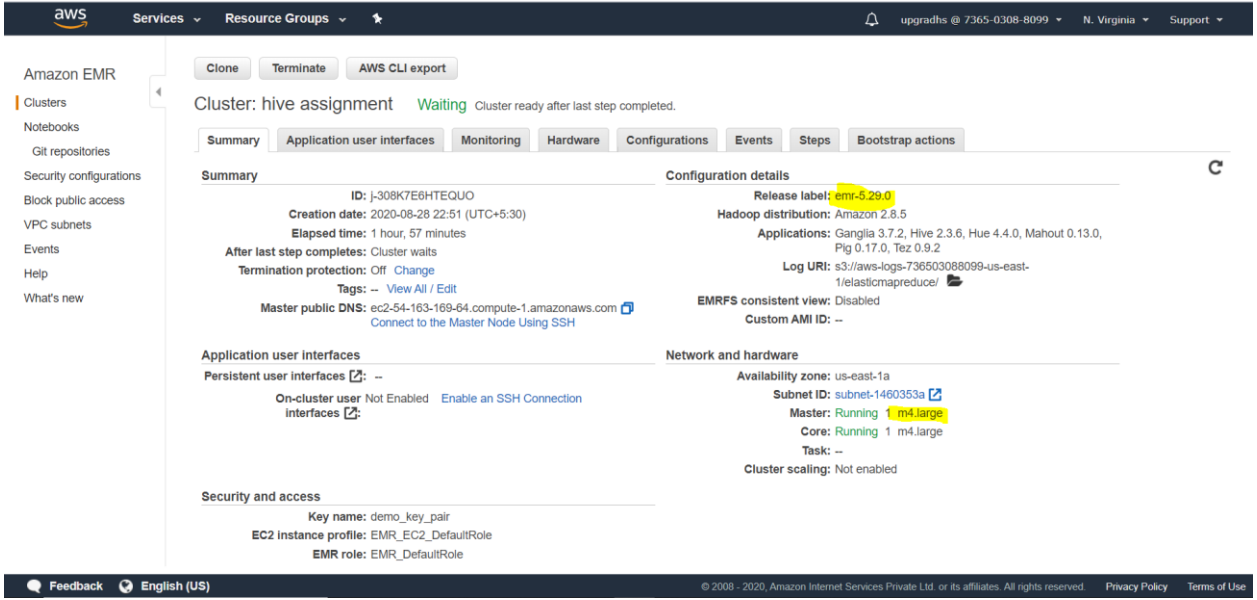
upGrad

Step 1 : Importing the data to HDFS

2.A - Create a Cluster

upGrad

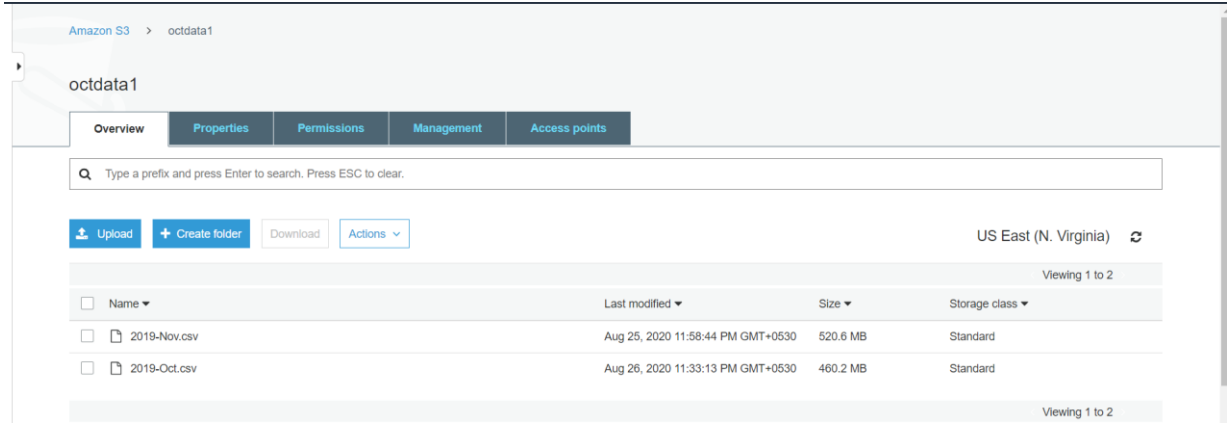
1. Created cluster with emr-5.29.0
2. Master and Core using m4.large instance



2.B – Upload data to S3

upGrad

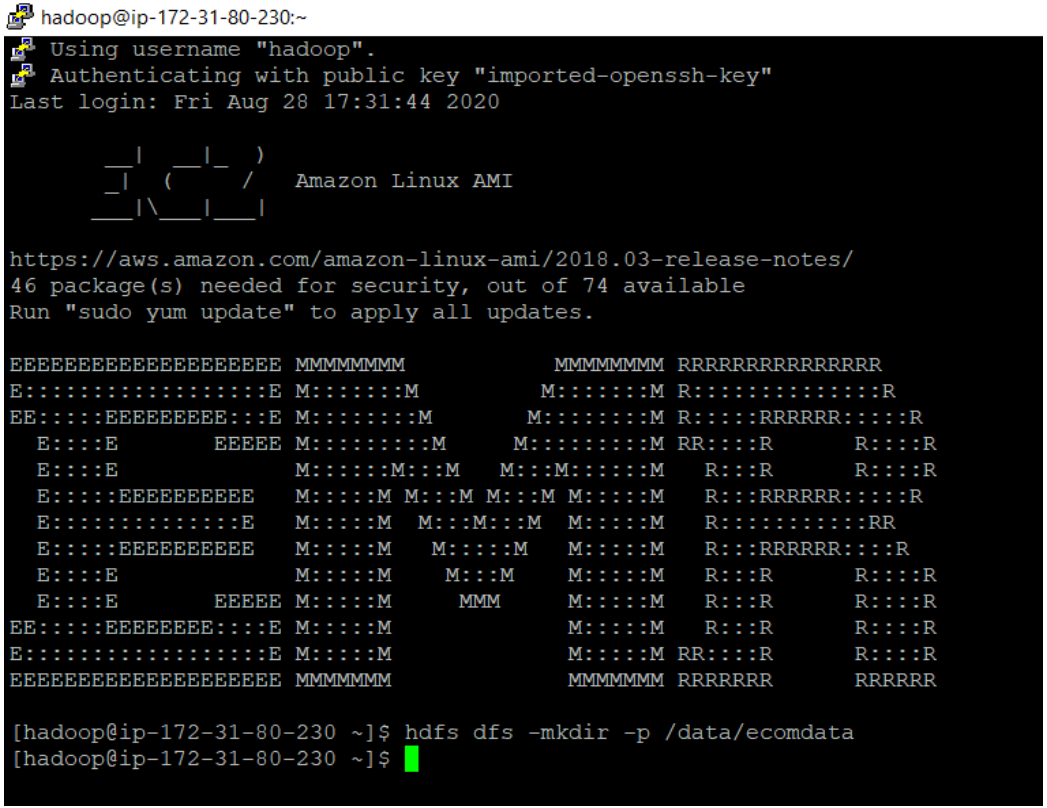
1. Created S3 bucket in AWS.
2. Uploaded CSV files 2019-Oct.csv
3. Uploaded CSV files 2019-Nov.csv



2.C - Transfer data from S3 to HDFS

upGrad

1. Login to EMR through putty and Create directory in HDFS.
2. **hdfs dfs -mkdir -p /data/ecomdata**



2.C - Transfer data from S3 to HDFS (contd.)

upGrad

1. Transfer data from S3 to HDFS.

2. **s3-dist-cp **

**--src='s3://octdata1/' **

--dest=hdfs:///data/ecomdata

```
[hadoop@ip-172-31-86-45 ~]$ s3-dist-cp \
> --src='s3://octdata1/' \
> --dest=hdfs:///data/ecomdata
20/08/26 17:40:49 INFO s3distcp.S3DistCp: Running with args: -libjars /usr/share/aws/emr/s3-dist-cp/lib/byte-buddy-1.9.10.jar,/usr/share/aws/emr/s3-dist-cp/lib/byte-buddy-agent-1.9.10.jar,
usr/share/aws/emr/s3-dist-cp/lib/commons-httpclient-3.1.jar,/usr/share/aws/emr/s3-dist-cp/lib/commons-logging-1.0.4.jar,/usr/share/aws/emr/s3-dist-cp/lib/guava-18.0.jar,/usr/share/aws/emr/
3-dist-cp/lib/mockito-core-2.27.0.jar,/usr/share/aws/emr/s3-dist-cp/lib/objenesis-2.6.jar,/usr/share/aws/emr/s3-dist-cp/lib/s3-dist-cp-2.13.0.jar,/usr/share/aws/emr/s3-dist-cp/lib/s3-dist-
p.jar --src=s3://octdata1/ --dest=hdfs:///data/ecomdata
20/08/26 17:40:50 INFO s3distcp.S3DistCp: S3DistCp args: --src=s3://octdata1/ --dest=hdfs:///data/ecomdata
20/08/26 17:40:50 INFO s3distcp.S3DistCp: Using output path 'hdfs:/tmp/ada9bf13-10af-4be8-a28d-8523a286fbb7/output'
20/08/26 17:40:50 INFO s3distcp.S3DistCp: GET http://169.254.169.254/latest/meta-data/placement/availability-zone result: us-east-1a
20/08/26 17:40:54 WARN cred.CredentialsLegacyConfigLocationProvider: Found the legacy config profiles file at [/home/hadoop/.aws/config]. Please move it to the latest default location [~/.
ws/credentials].
20/08/26 17:40:54 INFO s3distcp.S3DistCp: DefaultAWSCredentialsProviderChain is used to create AmazonS3Client. KeyId: ASIA2W6YBIPR2DYLU366
20/08/26 17:40:55 INFO s3distcp.FileInfoListing: Opening new file: hdfs:/tmp/ada9bf13-10af-4be8-a28d-8523a286fbb7/files/1
20/08/26 17:40:55 INFO s3distcp.S3DistCp: Created 1 files to copy 2 files
20/08/26 17:40:55 INFO s3distcp.S3DistCp: Reducer number: 10
20/08/26 17:40:55 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-86-45.ec2.internal/172.31.86.45:8032
20/08/26 17:40:56 INFO input.FileInputFormat: Total input files to process : 1
20/08/26 17:40:56 INFO mapreduce.JobSubmitter: number of splits:1
20/08/26 17:40:57 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1598462214699_0002
20/08/26 17:40:57 INFO impl.YarnClientImpl: Submitted application application_1598462214699_0002
20/08/26 17:40:57 INFO mapreduce.Job: The url to track the job: http://ip-172-31-86-45.ec2.internal:20888/proxy/application_1598462214699_0002/
20/08/26 17:40:57 INFO mapreduce.Job: Running job: job_1598462214699_0002
20/08/26 17:41:05 INFO mapreduce.Job: Job job_1598462214699_0002 running in uber mode : false
20/08/26 17:41:05 INFO mapreduce.Job:  map 0% reduce 0%
20/08/26 17:41:11 INFO mapreduce.Job:  map 100% reduce 0%
20/08/26 17:41:17 INFO mapreduce.Job:  map 100% reduce 10%
20/08/26 17:41:34 INFO mapreduce.Job:  map 100% reduce 20%
20/08/26 17:41:40 INFO mapreduce.Job:  map 100% reduce 30%
20/08/26 17:41:44 INFO mapreduce.Job:  map 100% reduce 40%
20/08/26 17:41:49 INFO mapreduce.Job:  map 100% reduce 50%
20/08/26 17:41:54 INFO mapreduce.Job:  map 100% reduce 60%
20/08/26 17:42:12 INFO mapreduce.Job:  map 100% reduce 70%
20/08/26 17:42:18 INFO mapreduce.Job:  map 100% reduce 80%
20/08/26 17:42:23 INFO mapreduce.Job:  map 100% reduce 90%
20/08/26 17:42:29 INFO mapreduce.Job:  map 100% reduce 100%
20/08/26 17:42:30 INFO mapreduce.Job: Job job_1598462214699_0002 completed successfully
20/08/26 17:42:30 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=332
                FILE: Number of bytes written=1938094
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=452
                HDFS: Number of bytes written=1028381690
                HDFS: Number of read operations=36
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=22
                S3: Number of bytes read=1028381690
```

2.C - Transfer data from S3 to HDFS (contd.)

upGrad

1. Check data present in HDFS.
2. **hdfs dfs -ls /data/ecomdata**
3. Check data present in csv file along with column name
4. **hdfs dfs -cat /data/ecomdata/2019-Nov.csv | head**
5. **hdfs dfs -cat /data/ecomdata/2019-Oct.csv | head**

```
20/08/28 17:36:54 INFO s3distcp.S3DistCp: Try to recursively delete hdfs:/tmp/98af9467-3620-4561-bdb4-2333d9612bcf
[hadoop@ip-172-31-80-230 ~]$ hdfs dfs -ls /data/ecomdata
Found 2 items
-rw-r--r--   1 hadoop hadoop  545839412 2020-08-28 17:36 /data/ecomdata/2019-Nov.csv
-rw-r--r--   1 hadoop hadoop  482542278 2020-08-28 17:36 /data/ecomdata/2019-Oct.csv
[hadoop@ip-172-31-80-230 ~]$
```

```
[hadoop@ip-172-31-80-230 ~]$ hdfs dfs -cat /data/ecomdata/2019-Nov.csv | head
event_time,event_type,product_id,category_id,category_code,brand,price,user_id,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
2019-11-01 00:00:24 UTC,remove_from_cart,5826182,1487580007483048900,,,3.33,553329724,2067216c-31b5-455d-a1cc-af0575a34ffb
2019-11-01 00:00:24 UTC,remove_from_cart,5826182,1487580007483048900,,,3.33,553329724,2067216c-31b5-455d-a1cc-af0575a34ffb
2019-11-01 00:00:25 UTC,view,5856189,1487580009026551821,,runail,15.71,562076640,09fafd6c-6c99-46b1-834f-33527f4de241
2019-11-01 00:00:32 UTC,view,5837835,1933472286753424063,,,3.49,514649199,432a4e95-375c-4b40-bd36-0fc039e77580
2019-11-01 00:00:34 UTC,remove_from_cart,5870838,1487580007675986893,,milv,0.79,429913900,2f0bff3c-252f-4fe6-afcd-5d8a6a92839a
cat: Unable to write to output stream.
[hadoop@ip-172-31-80-230 ~]$ hdfs dfs -cat /data/ecomdata/2019-Oct.csv | head
event_time,event_type,product_id,category_id,category_code,brand,price,user_id,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
2019-10-01 00:00:15 UTC,cart,5881449,1487580013522845895,,lovely,0.56,429681830,49e8d843-adf3-428b-a2c3-fe8bc6a307c9
2019-10-01 00:00:16 UTC,cart,5857269,1487580005134238553,,runail,2.62,430174032,73dea1e7-664e-43f4-8b30-d32b9d5af04f
2019-10-01 00:00:19 UTC,cart,5739055,1487580008246412266,,kapous,4.75,377667011,81326ac6-daa4-4f0a-b488-fd0956a78733
2019-10-01 00:00:24 UTC,cart,5825598,1487580009445982239,,,0.56,467916806,2f5b5546-b8cb-9ee7-7ecd-84276f8ef486
2019-10-01 00:00:25 UTC,cart,5698989,1487580006317032337,,,1.27,385985999,d30965e8-1101-44ab-b45d-cc1bb9fae694
cat: Unable to write to output stream.
[hadoop@ip-172-31-80-230 ~]$
```

upGrad

Step 2 : Creating Hive tables

2.A – Creating Hive Tables

upGrad

1. Create database and table with column name and data type

2.

```
CREATE DATABASE IF NOT EXISTS stores_analysis;
USE stores_analysis;
CREATE EXTERNAL TABLE
stores_analysis.ecom_dataset1 (
event_time STRING,
event_type STRING,
product_id STRING,
category_id STRING,
category_code STRING,
brand STRING,
price STRING,
user_id STRING,
user_session STRING
)
ROW FORMAT SERDE
'org.apache.hadoop.hive.serde2.OpenCSVSerde'
STORED AS TEXTFILE
LOCATION '/data/ecomdata'
TBLPROPERTIES ("skip.header.line.count"="1");
```

```
hive> use stores_analysis;
OK
Time taken: 0.024 seconds
hive> CREATE EXTERNAL TABLE stores_analysis.ecom_dataset (
    >     event_time STRING,
    >     event_type STRING,
    >     product_id STRING,
    >     category_id STRING,
    >     category_code STRING,
    >     brand STRING,
    >     price STRING,
    >     user_id STRING,
    >     user_session STRING
    > )
    > ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
    > STORED AS TEXTFILE
    > LOCATION '/data/ecomdata1'
    > TBLPROPERTIES ("skip.header.line.count"="1");
OK
Time taken: 0.228 seconds
```

upGrad

Step 3 : Analysis using the queries

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

upGrad

```
SELECT
  sum(cast(price as double)) as
    total_revenue_generated
FROM stores_analysis.ecom_dataset1
WHERE event_type= 'purchase' And
year(from_unixtime(unix_timestamp(ev
    ent_time, 'yyyy-MM-dd HH:mm:ss')))
    = 2019
AND
    month(from_unixtime(unix_timesta
    mp(event_time, 'yyyy-MM-dd
    HH:mm:ss'))) = 10;
```

```
hive> SELECT
    >   sum(cast(price as double)) as total_revenue_generated
    > FROM stores_analysis.ecom_dataset1
    > WHERE event_type= 'purchase' And
    > year(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss'))) = 2019
    > AND month(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss'))) = 10;
Query ID = hadoop_20200828180744_a60dc85b-f25e-4e2c-be3f-365cf0a7b30a
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1598635714320_0009)

----------------------------------------------------------------------------------------------
        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
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 53.59 s
----------------------------------------------------------------------------------------------
OK
1211538.4299997438
Time taken: 59.237 seconds, Fetched: 1 row(s)
hive> SELECT
```

Click icon to view Optimized Query

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

upGrad

```
SELECT
month(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss'))),
 sum(cast(price as double)) as total_purchase
FROM stores_analysis.ecom_dataset1
WHERE event_type= 'purchase' And
year(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss'))) = 2019
AND
    month(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss'))) IN (10,11)
GROUP BY
    month(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss')));
```

```
hive> SELECT
    >   month(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss'))),
    >   sum(cast(price as double)) as total_purchase
    > FROM stores_analysis.ecom_dataset1
    > WHERE event_type= 'purchase' And
    > year(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss'))) = 2019
    > AND month(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss'))) IN (10,11)
    > GROUP BY month(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss')));
Query ID = hadoop_20200828181328_10abd3c9-a4a4-43a1-8cfe-3524c5ce36ce
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1598635714320_0009)

----------------------------------------------------------------------------------------------
        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
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 81.95 s
----------------------------------------------------------------------------------------------
OK
10      1211538.4299997438
11      1531016.900000122
Time taken: 82.828 seconds, Fetched: 2 row(s)
hive>
```

Click icon to view Optimized Query

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

upGrad

```
select round(sum(case
when
    month(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss')))== 10
then price else - 1 * price end),2)
    as change_in_revenue
from
    stores_analysis.ecom_dataset1
where
    month(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss'))) IN (10,11);
```

```
hive> select round(sum(case
    > when month(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss')))== 10
    > then price else - 1 * price end),2) as change_in_revenue
    > from stores_analysis.ecom_dataset1
    > where month(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss'))) IN (10,11);
Query ID = hadoop_20200830130655_ddd069b0-bbe6-45e5-9611-00dde464a769
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1598791103493_0005)

----------------------------------------------------------------------------------------------
        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
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 108.95 s
----------------------------------------------------------------------------------------------
OK
-2633589.59
Time taken: 109.685 seconds, Fetched: 1 row(s)
hive>
```

Click icon to view Optimized Query

4. Find distinct categories of products.

upGrad

SELECT distinct category_code as distinct_category_code FROM stores_analysis.ecom_dataset1 WHERE year(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss'))) = 2019 AND month(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss'))) IN (10,11);

```
hive>  SELECT distinct category_code as distinct_category_code FROM stores_analysis.ecom_dataset1  WHERE year(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss'))) = 2019 AND mo
nth(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss'))) IN (10,11);
Query ID = hadoop_20200828181602_a3e4b29b-bba0-4d53-9b46-f69f9d5b628f
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1598635714320_0009)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      2          2        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 89.84 s
----------------------------------------------------------------------------------------------
OK
accessories.bag
apparel.glove
appliances.environment.vacuum
appliances.personal.hair_cutter
furniture.bathroom.bath
furniture.living_room.cabinet
sport.diving
stationery.cartrige

accessories.cosmetic_bag
appliances.environment.air_conditioner
furniture.living_room.chair
Time taken: 90.533 seconds, Fetched: 12 row(s)
hive>
```

Click icon to view Optimized Query

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

upGrad

```
SELECT  category_code as
    distinct_category_code ,

 COUNT(product_id) as
    total_product_id

FROM
    stores_analysis.ecom_dataset1

WHERE
    year(from_unixtime(unix_timest
    amp(event_time, 'yyyy-MM-dd
    HH:mm:ss'))) = 2019

AND
    month(from_unixtime(unix_tim
    estamp(event_time, 'yyyy-MM-
    dd HH:mm:ss'))) IN (10,11)

GROUP BY category_code;
```

Click icon to view Optimized Query

```
Time taken: 90.533 seconds, Fetched: 12 row(s)
hive> SELECT  category_code as distinct_category_code ,
    >   COUNT(product_id) as total_product_id
    > FROM stores_analysis.ecom_dataset1
    > WHERE year(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss'))) = 2019
    > AND month(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss'))) IN (10,11)
    > GROUP BY category_code;
Query ID = hadoop_20200828181842_97b8f6b7-ba91-4498-9c7e-b60a1f287709
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1598635714320_0009)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      2          2        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 86.14 s
----------------------------------------------------------------------------------------------
OK
accessories.bag 11681
apparel.glove   18232
appliances.environment.vacuum   59761
appliances.personal.hair_cutter 1643
furniture.bathroom.bath 9857
furniture.living_room.cabinet   13439
sport.diving    2
stationery.cartrige     26722
        8594895
accessories.cosmetic_bag        1248
appliances.environment.air_conditioner  332
furniture.living_room.chair     308
Time taken: 86.866 seconds, Fetched: 12 row(s)
hive>
```

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

upGrad

```
select brand , sum(price) as
    maximum_sales

from
    stores_analysis.ecom_dataset1

WHERE event_type= 'purchase'
    And

year(from_unixtime(unix_timestam
    p(event_time, 'yyyy-MM-dd
    HH:mm:ss'))) = 2019

AND
    month(from_unixtime(unix_tim
    estamp(event_time, 'yyyy-MM-
    dd HH:mm:ss'))) IN (10,11)

GROUP BY brand

order by maximum_sales desc limit
    10;
```

```
hive> select brand , sum(price) as maximum_sales
    > from stores_analysis.ecom_dataset1
    > WHERE event_type= 'purchase' And
    > year(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss'))) = 2019
    > AND month(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss'))) IN (10,11)
    > GROUP BY brand
    > order by maximum_sales desc limit 10;
Query ID = hadoop_20200828182312_1084eea5-3257-47c3-ad0f-c7acd80cd8b8
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1598635714320_0009)

----------------------------------------------------------------------------------------------
        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: 79.41 s
----------------------------------------------------------------------------------------------
OK
        1094188.2999999556
runail  148297.9400000003
grattol 106918.25000000178
irisk   92538.00000000806
uno     86341.77999999782
strong  67867.89999999986
masura  64324.54999999529
jessnail        59633.06999999984
cnd     59240.77000000015
ingarden        56727.600000000224
Time taken: 80.114 seconds, Fetched: 10 row(s)
```

Click icon to view Optimized Query

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

upGrad

```
SELECT brand,
  SUM(CASE
    WHEN
      month(from_unixtime(unix_timestamp(eve
      nt_time, 'yyyy-MM-dd HH:mm:ss'))) = 10
    THEN CAST(price AS DOUBLE) ELSE 1 *
      CAST(price AS DOUBLE) END) AS growth
FROM stores_analysis.ecom_dataset1
WHERE
      year(from_unixtime(unix_timestamp(event_
      time, 'yyyy-MM-dd HH:mm:ss'))) = 2019
AND
      month(from_unixtime(unix_timestamp(eve
      nt_time, 'yyyy-MM-dd HH:mm:ss'))) IN (10,
      11)
GROUP BY  brand
order by growth desc limit 10;
```

Click icon to view Optimized Query

```
hive> SELECT brand,
    >   SUM(CASE
    >     WHEN month(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss'))) = 10
    >     THEN CAST(price AS DOUBLE) ELSE 1 * CAST(price AS DOUBLE) END) AS growth
    > FROM stores_analysis.ecom_dataset1
    > WHERE year(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss'))) = 2019
    > AND month(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss'))) IN (10, 11)
    > GROUP BY  brand
    > order by growth desc limit 10;
Query ID = hadoop_20200828182921_76ee83b0-836c-44d7-b80e-5642468bdecf
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1598635714320_0009)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      2          2        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 120.57 s
--------------------------------------------------------------------------------
OK
        2.6194508599959712E7
strong  4927445.599999621
jessnail        3905094.109999678
runail  3838847.3300006418
irisk   2660064.559998853
grattol 1832533.029998915
marathon        1730084.2800002052
cnd     1475543.47000037
masura  1284493.1199984243
kosmekka        1156856.659999969
Time taken: 121.286 seconds, Fetched: 10 row(s)
```

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.

upGrad

```
select user_id as
    golden_customer,sum(price) as
    total_purchase

from stores_analysis.ecom_dataset1

where event_type= 'purchase' And

year(from_unixtime(unix_timestamp(event_ti
    me, 'yyyy-MM-dd HH:mm:ss'))) = 2019

AND
    month(from_unixtime(unix_timestamp(ev
    ent_time, 'yyyy-MM-dd HH:mm:ss'))) IN
    (10,11)

group by user_id

order by total_purchase desc limit 10;
```

```
hive> select user_id as golden_customer,sum(price) as total_purchase
    > from stores_analysis.ecom_dataset1
    > where event_type= 'purchase' And
    > year(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss'))) = 2019
    > AND month(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss'))) IN (10,11)
    > group by user_id
    > order by total_purchase desc limit 10;
Query ID = hadoop_20200828183711_7857a004-c2e6-4726-b9d7-e2a2df9e380d
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1598635714320_0009)

----------------------------------------------------------------------------------------------
        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: 85.89 s
----------------------------------------------------------------------------------------------
OK
557790271       2715.869999999991
150318419       1645.97
562167663       1352.8500000000004
531900924       1329.4500000000003
557850743       1295.4800000000002
522130011       1185.3899999999994
561592095       1109.6999999999996
431950134       1097.5899999999995
566576008       1056.3600000000017
521347209       1040.9099999999999
Time taken: 86.532 seconds, Fetched: 10 row(s)
hive>
```

Click icon to view Optimized Query

upGrad

Step 4 : Optimization Technique

4. Optimization Queries

upGrad

1. Create directory for optimized query

hdfs dfs -mkdir -p /data/ecomdataoptimized

2. Transfer data from s3 to HDFS

**s3-dist-cp **

**--src='s3://octdata1/' **

--dest=hdfs:///data/ecomdataoptimized

3. Check file present in ecomdataoptimized folder

hdfs dfs –ls / data/ecomdataoptimized

4. Check data in csv file and column name

```
[hadoop@ip-172-31-80-230 ~]$ hdfs dfs -ls /data/ecomdataoptimized
Found 2 items
-rw-r--r--   1 hadoop hadoop  545839412 2020-08-28 17:42 /data/ecomdataoptimized/2019-Nov.csv
-rw-r--r--   1 hadoop hadoop  482542278 2020-08-28 17:42 /data/ecomdataoptimized/2019-Oct.csv
[hadoop@ip-172-31-80-230 ~]$
```

```
[hadoop@ip-172-31-80-230 ~]$ hdfs dfs -cat /data/ecomdataoptimized/2019-Nov.csv | head
event_time,event_type,product_id,category_id,category_code,brand,price,user_id,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
2019-11-01 00:00:24 UTC,remove_from_cart,5826182,1487580007483048900,,,3.33,553329724,2067216c-31b5-455d-a1cc-af0575a34ffb
2019-11-01 00:00:24 UTC,remove_from_cart,5826182,1487580007483048900,,,3.33,553329724,2067216c-31b5-455d-a1cc-af0575a34ffb
2019-11-01 00:00:25 UTC,view,5856189,1487580009026551821,,runail,15.71,562076640,09fafd6c-6c99-46b1-834f-33527f4de241
2019-11-01 00:00:32 UTC,view,5837835,1933472286753424063,,,3.49,514649199,432a4e95-375c-4b40-bd36-0fc039e77580
2019-11-01 00:00:34 UTC,remove_from_cart,5870838,1487580007675986893,,milv,0.79,429913900,2f0bff3c-252f-4fe6-afcd-5d8a6a92839a
cat: Unable to write to output stream.
[hadoop@ip-172-31-80-230 ~]$ hdfs dfs -cat /data/ecomdataoptimized/2019-Oct.csv | head
event_time,event_type,product_id,category_id,category_code,brand,price,user_id,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
2019-10-01 00:00:15 UTC,cart,5881449,1487580013522845895,,lovely,0.56,429681830,49e8d843-adf3-428b-a2c3-fe8bc6a307c9
2019-10-01 00:00:16 UTC,cart,5857269,1487580005134238553,,runail,2.62,430174032,73dea1e7-664e-43f4-8b30-d32b9d5af04f
2019-10-01 00:00:19 UTC,cart,5739055,1487580008246412266,,kapous,4.75,377667011,81326ac6-daa4-4f0a-b488-fd0956a78733
2019-10-01 00:00:24 UTC,cart,5825598,1487580009445982239,,,0.56,467916806,2f5b5546-b8cb-9ee7-7ecd-84276f8ef486
2019-10-01 00:00:25 UTC,cart,5698989,1487580006317032337,,,1.27,385985999,d30965e8-1101-44ab-b45d-cc1bb9fae694
cat: Unable to write to output stream.
[hadoop@ip-172-31-80-230 ~]$
```

4. Optimization Queries (contd.)

upGrad

5. Create table, column and data type for optimized query with partition by and clustered condition.

```
CREATE TABLE
stores_analysis.ecom_data_optimized (
    event_time TIMESTAMP,
    event_type STRING,
    product_id STRING,
    category_id STRING,
    category_code STRING,
    brand STRING,
    price DOUBLE,
    user_id INT,
    user_session STRING
)
PARTITIONED BY (year INT, month INT)
CLUSTERED BY (brand) INTO 4 BUCKETS
STORED AS PARQUET
LOCATION '/data/ecomdataoptimized';
```

```
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
hive> CREATE TABLE stores_analysis.ecom_data_optimized (
    >     event_time TIMESTAMP,
    >     event_type STRING,
    >     product_id STRING,
    >     category_id STRING,
    >     category_code STRING,
    >     brand STRING,
    >     price DOUBLE,
    >     user_id INT,
    >     user_session STRING
    > )
    > PARTITIONED BY (year INT, month INT)
    > CLUSTERED BY (brand) INTO 4 BUCKETS
    > STORED AS PARQUET
    > LOCATION '/data/ecomdataoptimized';
OK
Time taken: 0.748 seconds
hive>
```

6. Set partition mode.

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

4. Optimization Queries (contd.)

upGrad

7. Insert data into new optimized table based on condition from ecom dataset

```
INSERT OVERWRITE TABLE
stores_analysis.ecom_data_optimized PARTITION(year,
month)
SELECT
   CAST(event_time AS TIMESTAMP),
   CAST(event_type AS STRING),
   CAST(product_id AS STRING),
   CAST(category_id AS STRING),
   CAST(category_code AS STRING),
   CAST(brand AS STRING),
   CAST(price AS DOUBLE),
   CAST(user_id AS INT),
   CAST(user_session AS STRING),
   year(from_unixtime(unix_timestamp(event_time , 'yyyy-
MM-dd HH:mm:ss'))),
   month(from_unixtime(unix_timestamp(event_time , 'yyyy-
MM-dd HH:mm:ss')))
FROM stores_analysis.ecom_dataset1
WHERE year(from_unixtime(unix_timestamp(event_time,
'yyyy-MM-dd HH:mm:ss'))) = 2019
AND month(from_unixtime(unix_timestamp(event_time,
'yyyy-MM-dd HH:mm:ss'))) IN (10, 11);
```

```
hive> set hive.exec.dynamic.partition.mode=nonstrict;
hive> INSERT OVERWRITE TABLE stores_analysis.ecom_data_optimized PARTITION(year, month)
    > SELECT
    >     CAST(event_time AS TIMESTAMP),
    >     CAST(event_type AS STRING),
    >     CAST(product_id AS STRING),
    >     CAST(category_id AS STRING),
    >     CAST(category_code AS STRING),
    >     CAST(brand AS STRING),
    >     CAST(price AS DOUBLE),
    >     CAST(user_id AS INT),
    >     CAST(user_session AS STRING),
    >     year(from_unixtime(unix_timestamp(event_time , 'yyyy-MM-dd HH:mm:ss'))),
    >     month(from_unixtime(unix_timestamp(event_time , 'yyyy-MM-dd HH:mm:ss')))
    > FROM stores_analysis.ecom_dataset
    > WHERE year(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss'))) = 2019
    > AND month(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss'))) IN (10, 11);
Query ID = hadoop_20200827182349_c93cd491-1f86-4fad-8928-03b22ea8a339
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1598548491850_0007)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      2          2        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 257.47 s
--------------------------------------------------------------------------------
Loading data to table stores_analysis.ecom_data_optimized partition (year=null, month=null)

Loaded : 2/2 partitions.
        Time taken to load dynamic partitions: 0.556 seconds
        Time taken for adding to write entity : 0.002 seconds
OK
Time taken: 262.909 seconds
```

4. Optimization Queries (contd.)

upGrad

8. Check data stored in both directory

dfs -du -s -h /data/ecomdata;
dfs -du -s -h /data/ecomdataoptimized;

```
hive> dfs -du -s -h /data/ecomdata1;
980.7 M  /data/ecomdata1
hive> dfs -du -s -h /data/ecomdataoptimized;
1.3 G  /data/ecomdataoptimized
hive>
```

upGrad

Step 5 : Optimized Query Output

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

upGrad

```
SELECT
  sum(cast(price as double)) as
     total_revenue_generated
FROM
     stores_analysis.ecom_data_optimized
WHERE event_type= 'purchase' And
year = 2019
AND month = 10;
```

```
hive> SELECT sum(cast(price as double)) as total_revenue_generated
    > FROM stores_analysis.ecom_data_optimized
    > WHERE event_type= 'purchase' And
    > year = 2019
    > AND month = 10;
Query ID = hadoop_20200828185233_20600091-9d9f-493c-acc3-4faed0294e84
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1598635714320_0010)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      4          4        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 19.66 s
----------------------------------------------------------------------------------------------
OK
1211538.4299999177
Time taken: 21.079 seconds, Fetched: 1 row(s)
hive>
```

CLICK HERE

Click icon to view Normal Query (It took **59.237** seconds before)

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

upGrad

```
SELECT
month,
  sum(cast(price as double)) as total_purchase
FROM stores_analysis.ecom_data_optimized
WHERE event_type= 'purchase' And
Year = 2019
AND month IN (10,11)
GROUP BY month;
```

```
hive> SELECT
    >   month,
    >   sum(cast(price as double)) as total_purchase
    > FROM stores_analysis.ecom_data_optimized
    > WHERE event_type= 'purchase' And
    > year = 2019
    > AND month IN (10,11)
    > GROUP BY month;
Query ID = hadoop_20200828185511_a2c7465e-a27b-4a1d-b4d6-58eccc56a163
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1598635714320_0010)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..... container     SUCCEEDED      4          4        0        0       0       0
Reducer 2 ..... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 22.21 s
--------------------------------------------------------------------------------
OK
10      1211538.4299999175
11      1531016.8999997238
Time taken: 23.107 seconds, Fetched: 2 row(s)
hive>
```

CLICK HERE

Click icon to view Normal Query (It took **82.828** seconds before)

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

upGrad

```
select round(sum(case
when month = 10
then price else - 1 * price end),2) as
    change_in_revenue
from
    stores_analysis.ecom_data_optimized
where year = 2019
and month IN (10,11);
```

```
hive> select round(sum(case
    > when month = 10
    > then price else - 1 * price end),2) as change_in_revenue
    > from stores_analysis.ecom_data_optimized
    > where year = 2019
    > and month IN (10,11);
Query ID = hadoop_20200830131103_78f0ada2-9003-4bd8-84e3-64274ac62234
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1598791103493_0005)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..... container     SUCCEEDED      6          6        0        0       0       0
Reducer 2 ..... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 22.44 s
--------------------------------------------------------------------------------
OK
-2633589.59
Time taken: 23.53 seconds, Fetched: 1 row(s)
hive>
```

CLICK HERE

Click icon to view Normal Query (It took **109.685** seconds before)

4. Find distinct categories of products.

upGrad

SELECT distinct category_code as distinct_category_code FROM stores_analysis.ecom_data_optimized WHERE year = 2019 AND month IN (10,11);

```
hive> SELECT distinct category_code as distinct_category_code FROM stores_analysis.ecom_data_optimized WHERE year= 2019 AND month IN (10,11);
Query ID = hadoop_20200828185622_3d4af840-e305-4c9b-94c9-95e3835479d7
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1598635714320_0010)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      4          4        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 20.16 s
--------------------------------------------------------------------------------
OK

accessories.bag
accessories.cosmetic_bag
apparel.glove
appliances.environment.air_conditioner
appliances.environment.vacuum
appliances.personal.hair_cutter
furniture.bathroom.bath
furniture.living_room.cabinet
furniture.living_room.chair
sport.diving
stationery.cartrige
Time taken: 20.916 seconds, Fetched: 12 row(s)
hive>
```

CLICK HERE

Click icon to view Normal Query (It took **90.533** seconds before)

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

upGrad

```
SELECT  category_code as
    distinct_category_code ,

 COUNT(product_id) as
    total_product_id

FROM
    stores_analysis.ecom_data_opt
    imized

WHERE year = 2019

AND month IN (10,11)

GROUP BY category_code;
```

```
hive> SELECT  category_code as distinct_category_code ,
    >   COUNT(product_id) as total_product_id
    > FROM stores_analysis.ecom_data_optimized
    > WHERE year = 2019
    > AND month IN (10,11)
    > GROUP BY category_code;
Query ID = hadoop_20200828185806_e375f373-2bed-4bd1-b60e-fd677a9e0d6e
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1598635714320_0010)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..... container     SUCCEEDED      4          4        0        0       0       0
Reducer 2 ..... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 21.93 s
--------------------------------------------------------------------------------
OK
        8594895
accessories.bag 11681
accessories.cosmetic_bag        1248
apparel.glove   18232
appliances.environment.air_conditioner  332
appliances.environment.vacuum   59761
appliances.personal.hair_cutter 1643
furniture.bathroom.bath 9857
furniture.living_room.cabinet   13439
furniture.living_room.chair     308
sport.diving    2
stationery.cartrige     26722
Time taken: 22.691 seconds, Fetched: 12 row(s)
hive>
```

CLICK HERE

Click icon to view Normal Query (It took **86.866** seconds before)

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

upGrad

```
select brand , sum(price) as
    maximum_sales

from
    stores_analysis.ecom_data_opt
    imized

WHERE event_type= 'purchase'
    And

Year = 2019

AND month IN (10,11)

GROUP BY brand

order by maximum_sales desc limit
    10;
```

```
hive> select brand , sum(price) as maximum_sales
    > from stores_analysis.ecom_data_optimized
    > WHERE event_type= 'purchase' And
    > year = 2019
    > AND month IN (10,11)
    > GROUP BY brand
    > order by maximum_sales desc limit 10;
Query ID = hadoop_20200828185921_4e208957-aa8d-455f-a23e-8bf6932e4da9
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1598635714320_0010)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..... container     SUCCEEDED      4          4        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: 22.66 s
--------------------------------------------------------------------------------
OK
        1094188.2999999742
runail  148297.9399999851
grattol 106918.25000000146
irisk   92538.00000000576
uno     86341.7799999995
strong  67867.8999999999
masura  64324.55000002017
jessnail        59633.07000000021
cnd     59240.76999999958
ingarden        56727.59999999931
Time taken: 23.48 seconds, Fetched: 10 row(s)
hive>
```

CLICK HERE

Click icon to view Normal Query (It took **80.114** seconds before)

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

upGrad

```
SELECT brand,
  SUM(CASE
    WHEN month = 10
    THEN CAST(price AS DOUBLE) ELSE 1 *
      CAST(price AS DOUBLE) END) AS growth
FROM stores_analysis.ecom_data_optimized
WHERE year = 2019
AND month IN (10, 11)
GROUP BY  brand
order by growth desc limit 10;
```

```
hive> SELECT brand,
    >   SUM(CASE
    >     WHEN month= 10
    >     THEN CAST(price AS DOUBLE) ELSE 1 * CAST(price AS DOUBLE) END) AS growth
    > FROM stores_analysis.ecom_data_optimized
    > WHERE year= 2019
    > AND month IN (10, 11)
    > GROUP BY  brand
    > order by growth desc limit 10;
Query ID = hadoop_20200828190517_00dc20be-0d20-4db4-8bb0-0cde8df76346
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1598635714320_0011)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      4          4        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: 19.46 s
--------------------------------------------------------------------------------
OK
        2.619450859996384E7
strong  4927445.599999445
jessnail        3905094.1100031626
runail  3838847.329997308
irisk   2660064.5599988815
grattol 1832533.0299993674
marathon        1730084.2800002364
cnd     1475543.4700002677
masura  1284493.1199985454
kosmekka        1156856.6599999631
Time taken: 25.372 seconds, Fetched: 10 row(s)
hive>
```

CLICK HERE

Click icon to view Normal Query (It took **121.286** seconds before)

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.

upGrad

```
select user_id as
    golden_customer,sum(price) as
    total_purchase

from stores_analysis.ecom_data_optimized

where event_type= 'purchase' And

Year = 2019

AND month IN (10,11)

group by user_id

order by total_purchase desc limit 10;
```

```
hive> select user_id as golden_customer,sum(price) as total_purchase
    > from stores_analysis.ecom_data_optimized
    > where event_type= 'purchase' And
    > year= 2019
    > AND month  IN (10,11)
    > group by user_id ,month
    > order by total_purchase desc limit 10;
Query ID = hadoop_20200828191730_6039c06d-96da-4223-bd78-33145cd0b3f0
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1598635714320_0013)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      4          4        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: 20.41 s
--------------------------------------------------------------------------------
OK
557790271       2715.869999999995
557850743       1295.4799999999996
561592095       1109.7
150318419       1104.76
566576008       1056.3600000000006
546827800       1004.4499999999985
546880217       987.52
474623506       914.9100000000002
570967715       904.9199999999998
549368055       903.8800000000008
Time taken: 21.524 seconds, Fetched: 10 row(s)
hive>
```

CLICK HERE

Click icon to view Normal Query (It took **89.071** seconds before)

upGrad

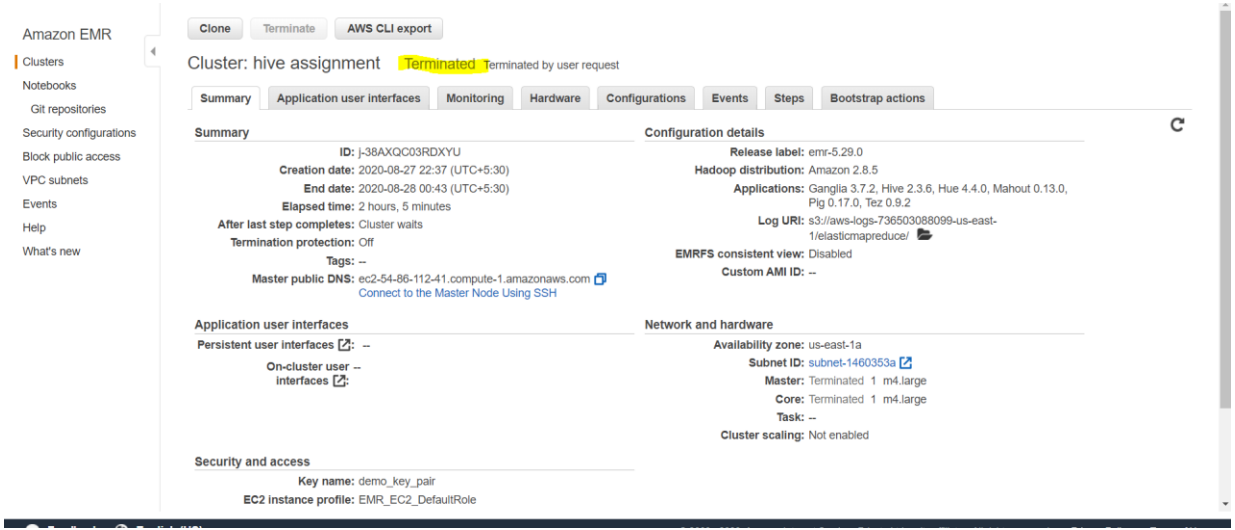
Step 6 : Cleaning Up

upGrad

1. Drop Database

```
hive> drop database stores_analysis;
OK
Time taken: 0.218 seconds
hive>
```

2. Terminate EMR Cluster



upGrad

#LifeKoKaroLift

Thank You!