

# Hive clickstream case study

## Problem statement:

With online sales gaining popularity, tech companies are exploring ways to improve their sales by analyzing customer behavior and gaining insights about product trends. Furthermore, the websites make it easier for customers to find the products they require without much scavenging.

This case study is done with public clickstream dataset of a cosmetics store. This data is collected by tracking clicks. Extract valuable insights from the data.

## Data:

Data set links:

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

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

## Instructions followed:

- 2-node cluster was created and used for the whole case study
- M4.large machines were used
- Putty was used to connect through SSH
- Emr-5.29.0 software configuration was used.
- CSVSerde used for loading the dataset into hive tables

## Implementation steps:

- Importing dataset to Hadoop from aws.
- Creating and connecting EMR cluster
- Creating database and launching hive queries on the cluster
- Cleaning up by dropping the database and terminating the cluster.

## Creating EMR:

The screenshot displays the Amazon EMR console for a cluster with ID j-2BEL19HRIJ49. The left sidebar shows navigation options like EMR Studio, EMR on EC2, Clusters, Notebooks, and various security configurations. The main content area is divided into several tabs: Summary, Application user interfaces, Monitoring, Hardware, Configurations, Events, Steps, and Bootstrap actions. The 'Summary' tab is active, showing the cluster's creation date (2021-10-26 10:23 UTC+5:30), elapsed time (15 minutes), and termination protection (On). It also lists the master public DNS and provides a link to connect to the master node using SSH. The 'Configuration details' section shows the release label (emr-5.29.0), Hadoop distribution (Amazon 2.8.5), and applications (Hive 2.3.6, Pig 0.17.0, Hue 4.4.0). The 'Network and hardware' section indicates the availability zone (us-east-1d), subnet ID (subnet-015cca971393ba550), and instance types (m4.large). The 'Security and access' section lists the key name (hiveupg), EC2 instance profile (EMR\_EC2\_DefaultRole), EMR role (EMR\_DefaultRole), and security groups for the master and core nodes.

Puttygen was used to convert a .pem file to a .ppk file.

Putty was used to connect from local machine to master node through SSH with the key value pair.

## Connecting EMR Master node through SSH:

```
Using username "hadoop".
Authenticating with public key "imported-openssh-key"
Last login: Tue Oct 26 08:54:08 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.

EEEEEEEEEEEEEEEEEEEE MMMMMMMM MMMMMMMM RRRRRRRRRRRRRRRR
E::::::::::::::::::::E M::::::::M M::::::::M R::::::::::::R
EE::::::::EEEEEEEEEE::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 M:::M::M R:::R R:::R
E:::EEEEEEEEEE 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 MMMMMMMM MMMMMMMM RRRRRRR RRRRRR

[hadoop@ip-172-31-15-71 ~]$ hadoop fs -ls /user/hive/ecom_cstudy
```

Create directory in hdfs to collect data:

```
hadoop fs -mkdir /user/hive/ecom_cstudy
```

```
[hadoop@ip-172-31-15-71 ~]$ hadoop fs -mkdir /user/hive/ecom_cstudy
[hadoop@ip-172-31-15-71 ~]$ aws s3 ls hivecstudy
2021-10-25 06:34:02 545839412 2019-Nov.csv
2021-10-25 06:34:02 482542278 2019-Oct.csv
```

To Import from aws s3:

Listing the s3 storage

```
aws s3 ls hivecstudy
```

```
[hadoop@ip-172-31-15-71 ~]$ aws s3 ls hivecstudy
2021-10-25 06:34:02 545839412 2019-Nov.csv
2021-10-25 06:34:02 482542278 2019-Oct.csv
```

listing the hadoop files:

```
hadoop fs -ls /user/hive/ecom_cstudy/
```

```
[hadoop@ip-172-31-12-218 ~]$ hadoop fs -ls /
Found 4 items
drwxr-xr-x - hdfs hadoop      0 2021-10-26 04:59 /apps
drwxrwxrwt - hdfs hadoop      0 2021-10-26 05:01 /tmp
drwxr-xr-x - hdfs hadoop      0 2021-10-26 04:59 /user
drwxr-xr-x - hdfs hadoop      0 2021-10-26 04:59 /var
[hadoop@ip-172-31-12-218 ~]$ hadoop fs -ls /user/hive/
Found 2 items
drwxr-xr-x - hadoop hadoop      0 2021-10-26 05:19 /user/hive/ecom_cstudy
drwxrwxrwt - hdfs hadoop      0 2021-10-26 04:59 /user/hive/warehouse
[hadoop@ip-172-31-12-218 ~]$
```

Copying data from aws s3 to Hadoop:

```
hadoop distcp 's3://hivecstudy/*' 'user/hive/ecom_cstudy/'
```

```

hadoop@ip-172-31-15-71:~$
21/10/26 09:04:49 INFO client.HMFSProxy: Connecting to ResourceManager at ip-172-31-15-71.ec2.internal/172.31.15.71:8032
[hadoop@ip-172-31-15-71:~]$ hadoop distcp *s3://hivecstudy/* "/user/hive/ecom_cstudy/"
21/10/26 09:04:49 INFO tools.DistCp: Input Options: DistCpOptions{atomicCommit=false, syncFolder=false, deleteMissing=false, ignoreFailure=false, overwrite=false, skipCRC=false, blocking=true, numListStatusThreads=0, maxMap=10, mapBandwidth=100, minConfigurationFile=null, copyStrategy=uniformsize, preserveHadoop=false, atomicOverwrite=null, logPath=null, sourceFileListing=null, sourcePath=s3://hivecstudy/, targetPath=/user/hive/ecom_cstudy, targetPathExists=true, filtersFile=null}
21/10/26 09:04:49 INFO tools.SimpleCopyListing: Paths (fileset) ms = 0; dirCnt = 0
21/10/26 09:04:49 INFO tools.SimpleCopyListing: Build file listing completed.
21/10/26 09:04:49 INFO Configuration.deprecation: io.sort.mb is deprecated. Instead, use mapreduce.task.io.sort.mb
21/10/26 09:04:49 INFO tools.DistCp: Number of paths in the copy list: 2
21/10/26 09:04:49 INFO tools.DistCp: Number of paths in the copy list: 2
21/10/26 09:04:49 INFO client.HMFSProxy: Connecting to ResourceManager at ip-172-31-15-71.ec2.internal/172.31.15.71:8032
21/10/26 09:04:50 INFO mapreduce.JobSubmitter: number of splits:2
21/10/26 09:04:50 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1635238076037_0001
21/10/26 09:04:51 INFO mapreduce.Job: The url to track the job: http://ip-172-31-15-71.ec2.internal:20888/proxy/application_1635238076037_0001/
21/10/26 09:04:51 INFO tools.DistCp: Job id: job_1635238076037_0001
21/10/26 09:04:51 INFO mapreduce.Job: Running job: job_1635238076037_0001
21/10/26 09:05:01 INFO mapreduce.Job: Job job_1635238076037_0001 running in uber mode : false
21/10/26 09:05:01 INFO mapreduce.Job: map 0% reduce 0%
21/10/26 09:05:23 INFO mapreduce.Job: map 100% reduce 0%
21/10/26 09:05:40 INFO mapreduce.Job: Job job_1635238076037_0001 completed successfully
21/10/26 09:05:40 INFO mapreduce.Job: Counter: 38
File System Counters
  FILE: Number of bytes read=0
  FILE: Number of bytes written=344914
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=868
  HDFS: Number of bytes written=1028381690
  HDFS: Number of read operations=36
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=6
  S3: Number of bytes read=1028381690
  S3: Number of bytes written=0
  S3: Number of read operations=0
  S3: Number of large read operations=0
  S3: Number of write operations=0
Job Counters
  Launched map tasks=2
  Other local map tasks=2
  Total time spent by all maps in occupied slots (ms)=2211820
  Total time spent by all reducers in occupied slots (ms)=0
  Total time spent by all map tasks (ms)=66435
  Total vcore-milliseconds taken by all map tasks=66435
  Total megabyte-milliseconds taken by all map tasks=71101440
Map-Reduce Framework
  Map input records=2
  Map output records=0
  Input split bytes=272
  Spilled records=0
  Failed Shuffles=0
  Merged Map outputs=0
  GC time elapsed (ms)=1248
  CPU time spent (ms)=45830
  Physical memory (bytes) snapshot=1068787376
  Virtual memory (bytes) snapshot=656949632
  Total committed heap usage (bytes)=901775360
File Input Format Counters
  Bytes Read=868
File Output Format Counters
  Bytes Written=0
DistCp Counters
  Bytes Copied=1028381690

```

Create a database:  
create database clickstream\_info;

```

hive> create database clickstream_info;
OK
Time taken: 0.389 seconds
hive> use clickstream_info;
OK
Time taken: 0.049 seconds
hive> create external table if not exists clickstream(event_time timestamp, event_type string, product_id string,
> category_id string, category_code string, brand string, price float, user_id bigint, user_session string)
> ROW FORMAT SERDE
> 'org.apache.hadoop.hive.serde2.OpenCSVSerde' WITH SERDEPROPERTIES ('separatorChar'=',', 'escapeChar'='\')
> stored as textfile
> location '/user/hive/ecom_cstudy/'
> tblproperties("skip.header.line.count"="1");
OK
Time taken: 0.568 seconds
hive> select * from clickstream limit 5 ;
OK
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   856138645   57ed222e-a54a-4907-9944-5a875c2d7f4f

```

Creating external base table:

create external table if not exists clickstream(event\_time timestamp, event\_type string,  
product\_id string,

category\_id string, category\_code string, brand string, price float, user\_id bigint, user\_session  
string)

ROW FORMAT SERDE

'org.apache.hadoop.hive.serde2.OpenCSVSerde' WITH SERDEPROPERTIES ('separatorChar'=',',  
'escapeChar'='\\')

stored as textfile

location '/user/hive/ecom\_cstudy/'

tblproperties("skip.header.line.count"="1");

```
hive> create external table if not exists clickstream(event_time timestamp, event_type string, product_id string,  
> category_id string, category_code string, brand string, price float, user_id bigint, user_session string)  
> ROW FORMAT SERDE  
> 'org.apache.hadoop.hive.serde2.OpenCSVSerde' WITH SERDEPROPERTIES ('separatorChar'=',', 'escapeChar'='\\')  
> stored as textfile  
> location '/user/hive/ecom_cstudy/'  
> tblproperties("skip.header.line.count"="1");  
OK  
Time taken: 0.568 seconds  
hive> select * from clickstream limit 5 ;  
OK  
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-alcc-af0575a34ffb  
2019-11-01 00:00:10 UTC view 5837166 1783999064103190764 pnb 22.22 556138645 57ed222e-a54a-4907-9944-5a875c2d7ff4f  
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-alcc-af0575a34ffb  
Time taken: 2.225 seconds, Fetched: 5 row(s)  
hive> set hive.exec.dynamic.partition.mode = nonstrict;  
hive> set hive.exec.dynamic.partition = true;  
hive> set hive.enforce.bucketing = true;
```

To enable partitioning and bucketing:

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

set hive.exec.dynamic.partition = true;

set hive.enforce.bucketing = true;

Creating table for bucketing an partition:

create table if not exists sales\_bucket(event\_time timestamp, product\_id string, category\_id string,

category\_code string, brand string, price float, user\_id bigint, user\_session string)

partitioned by (event\_type string)

clustered by (category\_code) into 10 buckets

ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'

stored as textfile;

```
hive> create table if not exists sales_bucket(event_time timestamp, product_id string, category_id string,  
> category_code string, brand string, price float, user_id bigint, user_session string)  
> partitioned by (event_type string)  
> clustered by (category_code) into 14 buckets  
> ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'  
> STORED AS textfile;  
OK  
Time taken: 0.124 seconds  
hive> insert into table sales_bucket partition(event_type) select event_time, product_id, category_id,
```

Inserting data into table:

insert into table sales\_bucket partition(event\_type) select event\_time, event\_type, product\_id,  
category\_id,

category\_code, brand, price, user\_id, user\_session from clickstream;

```
hive> insert into table sales_bucket partition(event_type) select event_time, product_id, category_id,  
> category_code, brand, price, user_id, user_session,event_type from clickstream;  
Query ID = hadoop_20211026091027_59439857-8964-4fbl-9a32-2dde644942dd  
Total jobs = 1  
Launching Job 1 out of 1  
Status: Running (Executing on YARN cluster with App id application_1635238076037_0002)  
  
-----  
VERTICES      MODE           STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED  
-----  
Map 1 ..... container    SUCCEEDED    2        2          0        0        0        0  
Reducer 2 ..... container    SUCCEEDED    5        5          0        0        0        0  
-----  
VERTICES: 02/02  [=====>>] 100%  ELAPSED TIME: 163.87 s  
-----  
Loading data to table clickstream_info.sales_bucket partition (event_type=null)  
  
Loaded : 4/4 partitions.  
Time taken to load dynamic partitions: 1.058 seconds  
Time taken for adding to write entity : 0.004 seconds  
OK  
Time taken: 168.838 seconds  
hive> desc sales_bucket;
```

desc clickstream;

```
event_type      string
Time taken: 0.34 seconds, Fetched: 14 row(s)
hive> desc clickstream;
OK
event_time      string          from deserializer
event_type      string          from deserializer
product_id      string          from deserializer
category_id     string          from deserializer
category_code   string          from deserializer
brand           string          from deserializer
price           string          from deserializer
user_id         string          from deserializer
user_session    string          from deserializer
Time taken: 0.053 seconds, Fetched: 9 row(s)
hive>
```

desc sales\_bucket;

```
hive> desc sales_bucket;
OK
event_time      string          from deserializer
product_id      string          from deserializer
category_id     string          from deserializer
category_code   string          from deserializer
brand           string          from deserializer
price           string          from deserializer
user_id         string          from deserializer
user_session    string          from deserializer
event_type      string

# Partition Information
# col_name      data_type      comment

event_time      string
Time taken: 0.34 seconds, Fetched: 14 row(s)
hive>
```

Select \* from sales\_bucket limit 5;

```
hive> set hive.cli.print.header=true;
hive> select * from sales_bucket limit 5 ;
OK
sales_bucket.event_time sales_bucket.product_id sales_bucket.category_id sales_bucket.category_code sales_bucket.brand sales_bucket.price sales_bucket.user_id sales_bucket.user_session sales_bucket.event_type
2019-10-07 20:53:08 UTC 5635080 1487580005754995573 4.44 527827629 b5f0f964-9457-40fd-bade-239a50de5c5d cart
2019-10-01 00:00:03 UTC 5773353 1487580005134238553 runall 2.62 463240011 26dd6e6e-4dac-f778-8d2c-92e149dab885 cart
2019-10-01 00:00:07 UTC 5881589 2151151071051215917 lovely 13.48 429681830 49e8d843-adf3-428b-a2c3-feb0c6a307c9 cart
2019-10-01 00:00:07 UTC 5723490 1487580005134238553 runall 2.62 463240011 26dd6e6e-4dac-f778-8d2c-92e149dab885 cart
2019-10-01 00:00:15 UTC 5891449 1487580013522845395 lovely 0.56 429681830 49e8d843-adf3-428b-a2c3-feb0c6a307c9 cart
Time taken: 0.148 seconds, Fetched: 5 row(s)
hive>
```

Show partitions;

```
hive> show partitions sales_bucket;
OK
partition
event_type=cart
event_type=purchase
event_type=remove_from_cart
event_type=view
Time taken: 0.101 seconds, Fetched: 4 row(s)
hive>
```

## Query Solutions to Questions:

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

select sum(price) from sales\_bucket where event\_type= 'purchase' and month(event\_time) = 10;

```
hive> select sum(price) from sales_bucket where event_type= 'purchase' and month(event_time) = 10;
Query ID = hadoop_20211026091517_a3661057-26dd-44e4-b02a-110e2221da7a
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1635238076037_0002)

-----
      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: 22.91 s
-----
OK
1211538.4299998297
Time taken: 24.825 seconds, Fetched: 1 row(s)
hive>
```

The total revenue generated by purchases made in October is **1211538.4299**.



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

`select sum(price) from sales_bucket where event_type= 'purchase' group by month(event_time);`

```
hive> select sum(price) from sales_bucket where event_type= 'purchase' group by month(event_time);
Query ID = hadoop_20211026091754_021a2082-1bdc-4197-a88f-370f8f8747e4
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1635238076037_0002)

-----
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: 18.48 s
-----
OK
1211538.4299998297
1531016.9000001058
Time taken: 19.256 seconds, Fetched: 2 row(s)
hive>
```

The total sum of purchases made in October is **1211538.4299**, in November is **1531016.900**

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

`select (sum(case when month(event_time)=11 then price else 0 end) - sum(case when month(event_time)=10 then price else 0 end)) as revenue_generated from sales_bucket where event_type='purchase' ;`

```
hive> select (sum(case when month(event_time)=11 then price else 0 end) - sum(case when
> month(event_time)=10 then price else 0 end)) as revenue_generated from sales_bucket where event_type='purchase' ;
Query ID = hadoop_20211026091938_a2117788-7d1f-4baf-adb5-74ab2d0d14c4
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1635238076037_0002)

-----
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: 19.16 s
-----
OK
319478.4700002761
Time taken: 19.855 seconds, Fetched: 1 row(s)
hive>
```

Change in revenue due to purchases from October to November is **319478.4700**.

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

select distinct category\_code from sales\_bucket;

```
hive> select distinct category_code from sales_bucket;
Query ID = hadoop_20211026092121_79f5d6f9-ald5-4333-bc9d-77b4306642b4
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1635238076037_0002)

-----
VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container  SUCCEEDED   7         7         0         0         0         0
Reducer 2 ..... container  SUCCEEDED   5         5         0         0         0         0
-----
VERTICES: 02/02  [=====>>>] 100%  ELAPSED TIME: 66.32 s
-----
OK

accessories.cosmetic_bag
stationery.cartridge
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: 67.028 seconds, Fetched: 12 row(s)
hive>
```

The result shows **11 categories** of product.

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

select category\_id, count(product\_id) from sales\_bucket group by category\_id;

```
hive> select category_id, count(product_id) from sales_bucket group by category_id;
Query ID = hadoop_20211026092401_28cdc3e8-6839-44aa-a693-f63a88102d30
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1635238076037_0002)

-----
VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container  SUCCEEDED   7         7         0         0         0         0
Reducer 2 ..... container  SUCCEEDED   5         5         0         0         0         0
-----
VERTICES: 02/02  [=====>>>] 100%  ELAPSED TIME: 69.58 s
-----
OK
1487580004966466385      16
1487580005050352463     83278
1487580005176181595      127
1487580005369119587        3
1487580005570446188       24
1487580005671109489     300570
1487580005687886706       14
1487580005922767741      640
1487580006056985476     1794
1487580006073762693     7856
1487580006174425994      466
1487580006216369036        3
1487580006585467807       17
1487580007281722301     34854
1487580007432717250     64400
1487580007508214725       36
1487580007592100809     12450
1487580007852147670     42694
1487580007894090712      4957
1487580007910867929     51488
1487580007952810971     24742
1487580008053474272     2629
1487580008070231489     1643
1487580008087028706      778
1487580008221246441       46
1487580008472904691       65
14875800085323236341      11
1487580009227878422     12922
1487580009336930331     5784
1487580009605365797     39093
1487580010955931741      835
1487580011098538083     3368
1487580011224367209        1
1487580011283087468     28420
1487580011408916594     20134
1487580011476025461     1871
1487580011677352062     17310
1487580011702517887     30290
1487580011970953351     4128
1487580012121948301     4731
```

hadoop@ip-172-31-15-71:~

```
lianail  
likato  
limoni  
lovely  
lowence  
mane  
marathon  
markell  
marutaka-foot  
masura  
matreshka  
matrix  
mavala  
metzger  
milv  
misikin  
missha  
moyou  
nagaraku  
naomi  
nefertiti  
neoleor  
nirvel  
nitrile  
oniq  
orly  
osmo  
ovale  
plazan  
polarus  
profepil  
profhenna  
protokeratin  
provoc  
rasyan  
refectocil  
rosi  
roubloff  
runail  
s.care  
sanoto  
severina  
shary  
shik  
skinity  
skinlite  
smart  
soleo  
solomeya  
sophin  
staleks  
strong  
supertan  
swarowski  
tertio  
treaclemoon  
trind  
uno  
uskusi  
veraclara  
vilenta  
yoko  
yu-r  
zeitun  
Time taken: 19.016 seconds, Fetched: 161 row(s)  
hive>
```

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

select brand, sum(price) as sales from sales\_bucket where event\_type = 'purchase'

group by brand

order by sales

desc limit 2;

```

Time taken: 17.307 seconds, Fetched: 1 row(s)
hive> select brand, sum(price) as sales from sales_bucket where event_type = 'purchase' group by brand order by sales
> desc limit 2;
Query ID = hadoop_20211026094742_3ad36080-6343-4fd7-8a3c-c61da9a6d48e
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1635238076037_0002)

-----
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: 16.68 s
-----
OK
      1094188.2999999976
runail 148297.93999999133
Time taken: 17.307 seconds, Fetched: 2 row(s)
hive>

```

There is lot of data that have blank spaces in the field of brand. **Runail** has the maximum sallies in October and November combined.

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

with brand\_sales as ( select brand, sum(case when month(event\_time)=10 then price else 0 end) as oct\_sales,  
oct\_sales,

sum(case when month(event\_time)=11 then price else 0 end) as nov\_sales

from sales\_bucket where event\_type='purchase' group by

brand ) select brand from brand\_sales where (nov\_sales-oct\_sales)>0 ;

```

hive> with brand_sales as ( select brand, sum(case when month(event_time)=10 then price else 0 end) as oct_sales,
> sum(case when month(event_time)=11 then price else 0 end) as nov_sales
> from sales_bucket where event_type='purchase' group by
> brand ) select brand from brand_sales where (nov_sales-oct_sales)>0 ;
Query ID = hadoop_20211026094017_02e3f9a6-d8f0-4a2c-af0f-e27004d22a26
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1635238076037_0002)

-----
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: 18.48 s
-----
OK

airnails
art-visage
artex
aura
ballbcare
barbie
batiste
beautix
beauty-free
beautyblender
beauugreen
benovy
binacil
bioaqua
biore
blixz
bluesky
bodyton
bpw.style
browxenna
candy
carmex
chi
coffin
concept
cosima
cosmoprofi
cristalinas
cutrin
de.lux

```

These 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.

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

```
hive> select user_id, sum(price) as sales from sales_bucket where event_type = 'purchase'
> group by user_id order by sales desc limit 10;
Query ID = hadoop_20211026094257_6eb19588-c251-4fcd-825b-ca6c525e95b3
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1635238076037_0002)
```

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: 18.83 s
OK
557790271      2715.8699999999996
150318419      1645.97
562167663      1352.85
531900924      1329.4500000000003
557850743      1295.4799999999998
522130011      1185.3899999999996
561592095      1109.7
431950134      1097.5899999999997
566576008      1056.3599999999997
521347209      1040.9099999999999
Time taken: 19.441 seconds, Fetched: 10 row(s)
hive>
```

The above result shows the top 10 customers to be rewarded golden customer plan.

## Cleaning:

Drop database clickstream\_info;

--- End ---