

Hive Query Case Study

1. Create a folder in HDFS to store the data files

```
hadoop fs -mkdir /tmp/test-folder
```

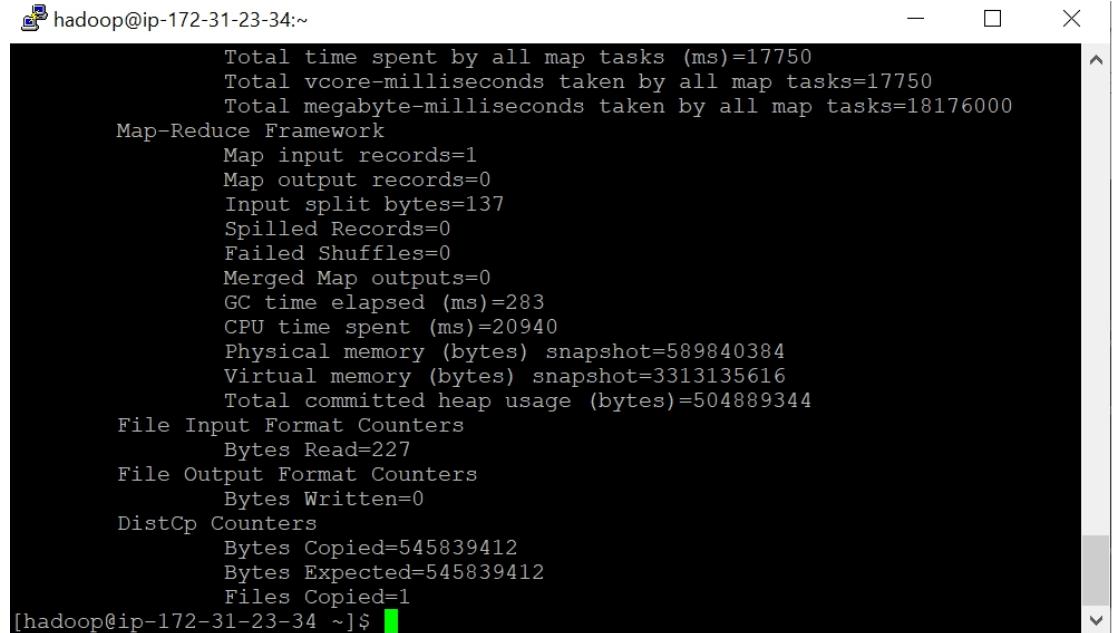
```
[hadoop@ip-172-31-23-34 ~]$ hadoop fs -mkdir /tmp/test-folder
[hadoop@ip-172-31-23-34 ~]$ hadoop fs -ls /tmp/
Found 3 items
drwxrwxrwx  - mapred mapred          0 2021-05-03 13:55 /tmp/hadoop-yarn
drwx-wx-wx  - hive    hadoop          0 2021-05-03 13:57 /tmp/hive
drwxr-xr-x  - hadoop  hadoop          0 2021-05-03 14:17 /tmp/test-folder
[hadoop@ip-172-31-23-34 ~]$ █
```

2. Copy the data files provided to “/tmp/test-folder” folder created on HDFS

```
hadoop distcp s3://bucketvijay/2019-Oct.csv /tmp/test-folder/
```

```
hadoop@ip-172-31-23-34:~$ █
Total time spent by all map tasks (ms)=17114
Total vcore-milliseconds taken by all map tasks=17114
Total megabyte-milliseconds taken by all map tasks=17524736
Map-Reduce Framework
  Map input records=1
  Map output records=0
  Input split bytes=136
  Spilled Records=0
  Failed Shuffles=0
  Merged Map outputs=0
  GC time elapsed (ms)=302
  CPU time spent (ms)=19780
  Physical memory (bytes) snapshot=607199232
  Virtual memory (bytes) snapshot=3305443328
  Total committed heap usage (bytes)=513277952
File Input Format Counters
  Bytes Read=227
File Output Format Counters
  Bytes Written=0
DistCp Counters
  Bytes Copied=482542278
  Bytes Expected=482542278
  Files Copied=1
[hadoop@ip-172-31-23-34 ~]$ █
```

```
hadoop distcp s3://bucketvijay/2019-Nov.csv /tmp/test-folder/
```



A screenshot of a terminal window titled "hadoop@ip-172-31-23-34:~". The window displays various performance metrics from a Map-Reduce job. Key statistics include:

- Total time spent by all map tasks (ms) = 17750
- Total vcore-milliseconds taken by all map tasks = 17750
- Total megabyte-milliseconds taken by all map tasks = 18176000
- Map-Reduce Framework metrics: Map input records=1, Map output records=0, Input split bytes=137, Spilled Records=0, Failed Shuffles=0, Merged Map outputs=0, GC time elapsed (ms)=283, CPU time spent (ms)=20940, Physical memory (bytes) snapshot=589840384, Virtual memory (bytes) snapshot=3313135616, Total committed heap usage (bytes)=504889344.
- File Input Format Counters: Bytes Read=227.
- File Output Format Counters: Bytes Written=0.
- DistCp Counters: Bytes Copied=545839412, Bytes Expected=545839412, Files Copied=1.

The command prompt at the bottom is [hadoop@ip-172-31-23-34 ~]\$.

3. Create a database called "casestudy"

```
create database if not exists casestudy;
```

```
hive>
>
>
> create database if not exists casestudy;
OK
Time taken: 0.815 seconds
hive> show databases;
OK
casestudy
default
Time taken: 0.141 seconds, Fetched: 2 row(s)
hive>
```

4. Create a table, load the CSV data files and skip the header row.

```
create table if not exists ecom (
event_time timestamp,
event_type string,
product_id string,
category_id string,
category_code string,
brand string,
price decimal(10,3),
user_id bigint,
user_session string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES (
"separatorChar" = ",",
"quoteChar" = "\"",
"escapeChar" = "\\")
```

```
STORED AS TEXTFILE
LOCATION '/tmp/test-folder/'
TBLPROPERTIES ("skip.header.line.count" = "1");
```

```
hive> create table if not exists ecom (
    > event_time timestamp,
    > event_type string,
    > product_id string,
    > category_id string,
    > category_code string,
    > brand string,
    > price decimal(10, 3),
    > user_id bigint,
    > user_session string
    > )
    > ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
    > WITH SERDEPROPERTIES (
    >     "separatorChar" = ",",
    >     "quoteChar" = "\"",
    >     "escapeChar" = "\\")
    > STORED AS TEXTFILE
    > LOCATION '/tmp/test-folder/'
    > TBLPROPERTIES ("skip.header.line.count" = "1");
OK
Time taken: 0.723 seconds
hive> show tables;
OK
ecom
Time taken: 0.049 seconds, Fetched: 1 row(s)
hive> █
```

5. Show 5 rows from “ecom” table;

Select * from ecom limit 5;

```
hive> Select * from ecom limit 5;
OK
ecom.event_time ecom.event_type ecom.product_id ecom.category_id      ecom.cat
egory_code      ecom.brand      ecom.price       ecom.user_id      ecom.user_sessio
n
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      2
2.22      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
Time taken: 0.149 seconds, Fetched: 5 row(s)
hive> █
```

6. Describe “ecom” table

describe ecom;

```

hive> describe ecom;
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: 0.737 seconds, Fetched: 9 row(s)
hive> █

```

7. Count the number of rows inserted into the table.

```
Select count(*) from ecom;
```

```

hive> select count(*) from ecom;
Query ID = hadoop_20210503142858_6a8518de-2731-42ad-8230-9e46584e551c
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1620050213727_0003)

-----  

      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: 51.17 s
-----  

OK  

8738120  

Time taken: 56.563 seconds, Fetched: 1 row(s)
hive> █

```

8. Set the parameters to create the dynamic partition.

```
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;
hive> █

```

9. Create a partition table on “event month” and buckets based on “event type”.

```

create table if not exists dyn_part_ecom
(
  event_type string,
  product_id string,
  category_id string,
  category_code string,
  brand string,
  price decimal(10,3),
  user_id bigint,
  user_session string
)
partitioned by (month int) clustered by (event_type) into 4 buckets;
```

```

hive> create table if not exists dyn_part_ecom
  > (
  >   event_type string,
  >   product_id string,
  >   category_id string,
  >   category_code string,
  >   brand string,
  >   price decimal(10,3),
  >   user_id bigint,
  >   user_session string
  > )
  > partitioned by (month int) clustered by (event_type) into 4 buckets;
OK
Time taken: 0.09 seconds
hive> █

```

10. Show 5 rows from “dyn_part_ecom” table;

```
Select * from dyn_part_ecom limit 5;
```

```

hadoop@ip-172-31-23-34:~ 2.22      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
Time taken: 0.149 seconds, Fetched: 5 row(s)
hive> Select * from dyn_part_ecom limit 5;
OK
dyn_part_ecom.event_type      dyn_part_ecom.product_id      dyn_part_ecom.ca
tегory_id      dyn_part_ecom.category_code      dyn_part_ecom.brand      dyn_part
_ecom.price      dyn_part_ecom.user_id      dyn_part_ecom.user_session      dyn_part
_ecom.month
cart      5854812 1602943681873052386      grattol 5.240      558342455      b
a64d316-a136-4554-85b9-e2ad07a9d742      10
cart      5635080 1487580005754995573      4.440      527827629      b
5f0f964-9457-4dfd-bade-239a9cde9c5d      10
cart      5686080 1487580008145748965      0.510      550107440      3
827208f-7baf-4ed3-826b-141276bd13a7      10
cart      5773353 1487580005134238553      runail 2.620      463240011      2
6dd6e6e-4dac-4778-8d2c-92e149dab885      10
cart      5802162 1487580013069861041      italwax 23.810      484697155      7
1efeb63-8da3-4a03-8de1-2c9bacc64f2c      10
Time taken: 0.232 seconds, Fetched: 5 row(s)
hive> █

```

11. Describe dyn_part_ecom table

```
describe dyn_part_ecom;
```

```

hive> describe dyn_part_ecom;
OK
event_type          string
product_id         string
category_id        string
category_code      string
brand              string
price              decimal(10,3)
user_id             bigint
user_session        string
month              int
# Partition Information
# col_name           data_type           comment
month              int
Time taken: 0.354 seconds, Fetched: 14 row(s)
hive> █

```

12. Populate the partitioned table based on initial data load completed in “ecom” table.

```
insert into table dyn_part_ecom partition (month)
select
event_type,
product_id,
category_id,
category_code,
brand,
price,
user_id,
user_session,
month(event_time) as month
from ecom;

Map 1: 2/2      Reducer 2: 1(+2)/5
Map 1: 2/2      Reducer 2: 1(+3)/5
Map 1: 2/2      Reducer 2: 2(+2)/5
Map 1: 2/2      Reducer 2: 2(+3)/5
Map 1: 2/2      Reducer 2: 3(+2)/5
Map 1: 2/2      Reducer 2: 4(+1)/5
Map 1: 2/2      Reducer 2: 5/5
Loading data to table default.dyn_part_ecom partition (month=null)

Time taken to load dynamic partitions: 0.378 seconds
Time taken for adding to write entity : 0.002 seconds
OK
Time taken: 168.734 seconds
hive> █
```

13. Confirm number of rows in original table (“ecom”) matches with number of rows inserted into partitioned table (“dyn_part_ecom”)

```
Select count(*) from dyn_part_ecom;
```

```
hive> Select count(*) from dyn_part_ecom;
OK
8738120
Time taken: 0.289 seconds, Fetched: 1 row(s)
hive> █
```

14. Set the parameter to allow hive to perform Cartesian product

```
set hive.strict.checks.cartesian.product=false;
set hive.mapred.mode=nonstrict;
```

```
hive> set hive.strict.checks.cartesian.product=false;
hive> set hive.mapred.mode=nonstrict;
hive> █
```

Assessment Questions:

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

```
select sum(price) as OCT_Revenue
from dyn_part_ecom
where month= 10 and event_type = 'purchase';

Map 1: 0/5      Reducer 2: 0/1
Map 1: 0/5      Reducer 2: 0/1
Map 1: 0/5      Reducer 2: 0/1
Map 1: 0(+1)/5  Reducer 2: 0/1
Map 1: 0(+3)/5  Reducer 2: 0/1
Map 1: 1(+3)/5  Reducer 2: 0/1
Map 1: 2(+2)/5  Reducer 2: 0/1
Map 1: 3(+2)/5  Reducer 2: 0/1
Map 1: 3(+2)/5  Reducer 2: 0(+1)/1
Map 1: 4(+1)/5  Reducer 2: 0(+1)/1
Map 1: 5/5      Reducer 2: 0(+1)/1
Map 1: 5/5      Reducer 2: 1/1
OK
oct_revenue
1211538.430
Time taken: 21.872 seconds, Fetched: 1 row(s)
hive> █
```

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

```
select month, sum(price) Revenue
from dyn_part_ecom
where event_type = 'purchase'
group by month;

Map 1: 0/7      Reducer 2: 0/2
Map 1: 0/7      Reducer 2: 0/2
Map 1: 0(+2)/7  Reducer 2: 0/2
Map 1: 0(+3)/7  Reducer 2: 0/2
Map 1: 1(+2)/7  Reducer 2: 0/2
Map 1: 1(+3)/7  Reducer 2: 0/2
Map 1: 2(+3)/7  Reducer 2: 0/2
Map 1: 3(+2)/7  Reducer 2: 0/2
Map 1: 3(+3)/7  Reducer 2: 0/2
Map 1: 4(+3)/7  Reducer 2: 0/2
Map 1: 5(+2)/7  Reducer 2: 0/2
Map 1: 5(+2)/7  Reducer 2: 0(+1)/2
Map 1: 6(+1)/7  Reducer 2: 0(+2)/2
Map 1: 7/7      Reducer 2: 0(+2)/2
Map 1: 7/7      Reducer 2: 2/2
OK
month    revenue
10      1211538.430
11      1531016.900
Time taken: 26.745 seconds, Fetched: 2 row(s)
hive> █
```

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

Before using mapjoin (query time taken = 33 secs):

```

SELECT OCT_Revenue, NOV_Revenue, (NOV_Revenue - OCT_Revenue) as Revenue_Growth
From
(select sum(price) as OCT_Revenue
from dyn_part_ecom
where month= 10 and event_type = 'purchase') t1
FULL OUTER JOIN
(select sum(price) as NOV_Revenue
from dyn_part_ecom
where month= 11 and event_type = 'purchase') t2 ;

```

```

Map 1: 4(+1)/5  Map 4: 5/5      Reducer 2: 0/1  Reducer 3: 0/1  Reducer 5: 0(+1)
/1
Map 1: 4(+1)/5  Map 4: 5/5      Reducer 2: 0(+1)/1      Reducer 3: 0/1  Reducer
5: 0(+1)/1
Map 1: 4(+1)/5  Map 4: 5/5      Reducer 2: 0(+1)/1      Reducer 3: 0/1  Reducer
5: 1/1
Map 1: 5/5       Map 4: 5/5      Reducer 2: 0(+1)/1      Reducer 3: 0/1  Reducer
5: 1/1
Map 1: 5/5       Map 4: 5/5      Reducer 2: 1/1  Reducer 3: 0(+1)/1      Reducer
5: 1/1
Map 1: 5/5       Map 4: 5/5      Reducer 2: 1/1  Reducer 3: 1/1  Reducer 5: 1/1
OK
oct_revenue      nov_revenue     revenue_growth
1211538.430     1531016.900    319478.470
Time taken: 33.542 seconds, Fetched: 1 row(s)
hive> █

```

After using mapjoin (query time taken = 29 secs):

```

SELECT /*+ MAPJOIN(OCT_Revenue) */ OCT_Revenue, NOV_Revenue, (NOV_Revenue -
OCT_Revenue) as Revenue_Growth
From
(select sum(price) as OCT_Revenue
from dyn_part_ecom
where month= 10 and event_type = 'purchase') t1
FULL OUTER JOIN
(select sum(price) as NOV_Revenue
from dyn_part_ecom
where month= 11 and event_type = 'purchase') t2 ;

```

```

Map 1: 4(+1)/5  Map 4: 5/5      Reducer 2: 0(+1)/1      Reducer 3: 0/1  Reducer
5: 0(+1)/1
Map 1: 4(+1)/5  Map 4: 5/5      Reducer 2: 0(+1)/1      Reducer 3: 0/1  Reducer
5: 1/1
Map 1: 5/5       Map 4: 5/5      Reducer 2: 1/1  Reducer 3: 0/1  Reducer 5: 1/1
Map 1: 5/5       Map 4: 5/5      Reducer 2: 1/1  Reducer 3: 0(+1)/1      Reducer
5: 1/1
Map 1: 5/5       Map 4: 5/5      Reducer 2: 1/1  Reducer 3: 1/1  Reducer 5: 1/1
OK
oct_revenue      nov_revenue     revenue_growth
1211538.430     1531016.900    319478.470
Time taken: 29.737 seconds, Fetched: 1 row(s)
hive> █

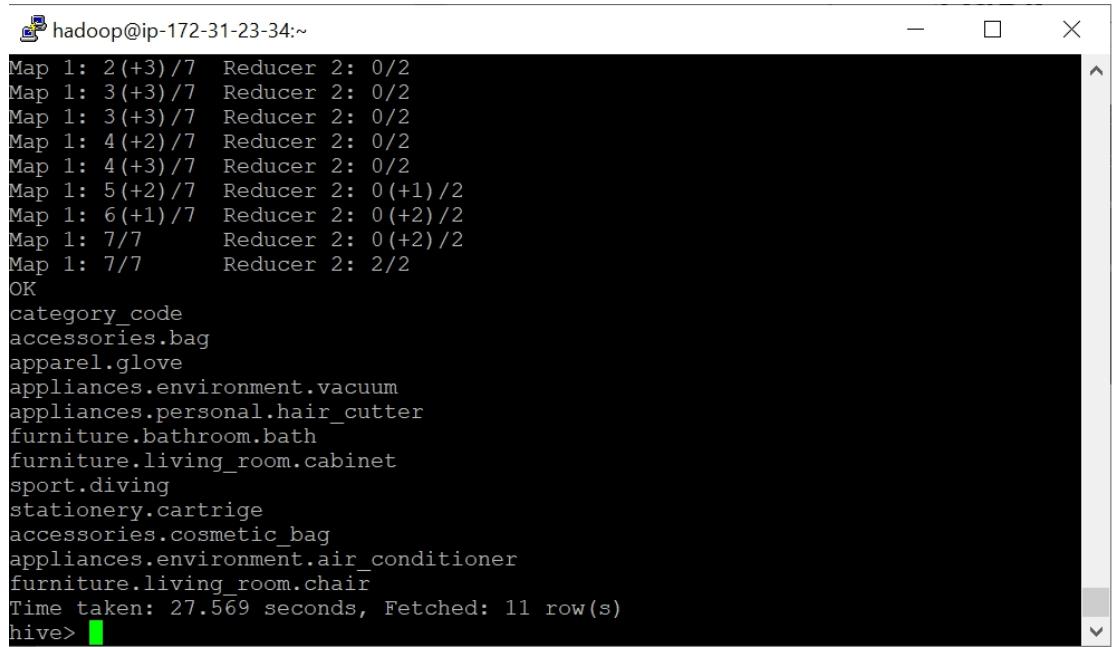
```

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

```

select distinct category_code
from dyn_part_ecom
where LENGTH(category_code) > 0 ;

```



A screenshot of a terminal window titled "hadoop@ip-172-31-23-34:~". The window displays the output of a Hive query. The output shows 11 rows of data, each consisting of a category code and its count. The categories listed are: 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, and appliances.environment.air_conditioner. The final line of output indicates that the query took 27.569 seconds and fetched 11 rows.

```
Map 1: 2(+3)/7 Reducer 2: 0/2
Map 1: 3(+3)/7 Reducer 2: 0/2
Map 1: 3(+3)/7 Reducer 2: 0/2
Map 1: 4(+2)/7 Reducer 2: 0/2
Map 1: 4(+3)/7 Reducer 2: 0/2
Map 1: 5(+2)/7 Reducer 2: 0(+1)/2
Map 1: 6(+1)/7 Reducer 2: 0(+2)/2
Map 1: 7/7 Reducer 2: 0(+2)/2
Map 1: 7/7 Reducer 2: 2/2
OK
category_code
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: 27.569 seconds, Fetched: 11 row(s)
hive> █
```

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

```
select category_code, count(product_id) as product_count
from dyn_part_ecom
where LENGTH(category_code) > 0
group by category_code;
```

```

hadoop@ip-172-31-23-34:~ 
Map 1: 2(+3)/7 Reducer 2: 0/2
Map 1: 3(+3)/7 Reducer 2: 0/2
Map 1: 3(+3)/7 Reducer 2: 0/2
Map 1: 4(+2)/7 Reducer 2: 0/2
Map 1: 5(+2)/7 Reducer 2: 0/2
Map 1: 5(+2)/7 Reducer 2: 0(+1)/2
Map 1: 6(+1)/7 Reducer 2: 0(+2)/2
Map 1: 7/7 Reducer 2: 0(+2)/2
Map 1: 7/7 Reducer 2: 2/2
OK
category_code product_count
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.cartridge 26722
accessories.cosmetic_bag 1248
appliances.environment.air_conditioner 332
furniture.living_room.chair 308
Time taken: 28.592 seconds, Fetched: 11 row(s)
hive> 

```

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

```

select brand, sum(price) as sales
from dyn_part_ecom
where LENGTH(brand) > 0
and event_type = 'purchase'
group by brand
order by sales desc limit 1;

```

```

hadoop@ip-172-31-23-34:~ 
Map 1: 0/7 Reducer 2: 0/1 Reducer 3: 0/1
Map 1: 0/7 Reducer 2: 0/1 Reducer 3: 0/1
Map 1: 0(+1)/7 Reducer 2: 0/1 Reducer 3: 0/1
Map 1: 0(+3)/7 Reducer 2: 0/1 Reducer 3: 0/1
Map 1: 0(+3)/7 Reducer 2: 0/1 Reducer 3: 0/1
Map 1: 0(+3)/7 Reducer 2: 0/1 Reducer 3: 0/1
Map 1: 0(+3)/7 Reducer 2: 0/1 Reducer 3: 0/1
Map 1: 0(+3)/7 Reducer 2: 0/1 Reducer 3: 0/1
Map 1: 0(+3)/7 Reducer 2: 0/1 Reducer 3: 0/1
Map 1: 1(+3)/7 Reducer 2: 0/1 Reducer 3: 0/1
Map 1: 2(+3)/7 Reducer 2: 0/1 Reducer 3: 0/1
Map 1: 3(+3)/7 Reducer 2: 0/1 Reducer 3: 0/1
Map 1: 3(+3)/7 Reducer 2: 0/1 Reducer 3: 0/1
Map 1: 4(+3)/7 Reducer 2: 0/1 Reducer 3: 0/1
Map 1: 6(+1)/7 Reducer 2: 0/1 Reducer 3: 0/1
Map 1: 6(+1)/7 Reducer 2: 0(+1)/1 Reducer 3: 0/1
Map 1: 7/7 Reducer 2: 0(+1)/1 Reducer 3: 0/1
Map 1: 7/7 Reducer 2: 1/1 Reducer 3: 0/1
Map 1: 7/7 Reducer 2: 1/1 Reducer 3: 0(+1)/1
Map 1: 7/7 Reducer 2: 1/1 Reducer 3: 1/1
OK
brand sales
runail 148297.940
Time taken: 28.904 seconds, Fetched: 1 row(s)
hive> 

```

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

```

SELECT t1.brand, OCT_Revenue, NOV_Revenue, (NOV_Revenue - OCT_Revenue) as
Revenue_Growth
From
(select brand, sum(price) as OCT_Revenue

```

```

from dyn_part_ecom
where month= 10 and event_type = 'purchase' and LENGTH(brand) > 0
group by brand) t1
LEFT JOIN
(select brand, sum(price) as NOV_Revenue
from dyn_part_ecom
where month= 11 and event_type = 'purchase' and LENGTH(brand) > 0 group by brand) t2
WHERE t1.brand = t2.brand
AND (NOV_Revenue - OCT_Revenue) > 0;

```

tl.brand	oct_revenue	nov_revenue	revenue_growth
airmalls	5118.900	5691.520	572.620
art-vtsage	2092.710	2997.800	905.090
artex	2730.640	4327.250	1596.610
aura	83.950	177.510	93.560
balbcare	155.330	212.380	57.050
batiste	772.400	874.170	101.770
beautix	10493.950	12222.950	1729.000
beauty-free	554.170	1782.860	1228.690
beautyblender	78.740	109.410	30.670
beautygroup	510.110	668.390	256.840
benny	409.620	3259.970	2850.350
bioaqua	942.890	1398.120	455.230
bioré	60.650	90.310	29.460
blixt	38.950	63.400	24.450
bluesky	10307.240	10565.530	258.290
bodyton	1376.340	1380.640	4.300
bpw.style	11572.150	14837.440	3265.290
brownenna	14331.370	14916.730	585.360
candy	534.960	799.380	264.420
carmex	140.100	243.360	98.280
chanel	558.500	588.610	179.670
coifin	903.000	1428.490	525.490
concept	11032.140	13380.400	2348.260
cosima	20.230	20.930	0.700
cosmoprofi	8322.810	14536.990	6214.180
cristalinas	427.630	584.950	157.320
cutrin	299.370	367.620	68.250
de.lux	1659.700	2775.510	1115.810
deoproce	316.840	329.170	12.330
depiliflax	2707.070	2803.780	96.710
dizainix	819.130	1045.510	126.300
ecocraft	10472.050	12009.170	1537.120
ecolab	41.160	241.950	200.790
egomania	1214.300	951.450	
elizavecca	70.530	204.300	133.770
ellipsa	245.850	606.040	360.190
elskin	251.090	307.650	56.560
enjoy	41.350	136.570	95.220
entity	479.710	719.260	239.550
eos	54.340	152.610	98.270
estel	21756.750	24142.670	2385.920
estelare	444.810	471.870	27.060

tl.brand	oct_revenue	nov_revenue	revenue_growth
estelare	21756.750	24142.670	2385.920
estelare	444.810	471.870	27.060
estelare	6624.230	7077.280	1953.050
firmavita	837.370	1291.570	454.600
farmona	1692.460	1843.430	150.970
fedua	52.380	263.810	211.430
finish	98.380	230.380	132.000
fly	17.140	27.170	10.030
foamie	35.040	80.490	45.450
freedecor	3421.780	7671.800	4250.020
freshbubble	318.700	502.340	183.640
gehwol	1089.070	1500.680	468.610
glysolid	69.730	114.590	21.860
godefroy	401.220	425.120	23.900
gracis	100.920	102.610	1.690
grattol	35445.540	71472.710	36027.170
greymy	29.210	489.490	460.280
happyfons	801.920	1091.590	289.670
haruyama	9390.690	12352.910	2962.220
igrobeauty	513.660	645.070	131.410
ingarden	23161.390	33566.210	10404.820
imx	288.020	351.210	63.160
insight	143.700	1721.960	278.260
italk	15521.560	40495.040	1354.080
italmax	21940.240	24799.370	2859.130
jaguar	1102.110	1110.650	8.540
jas	3318.960	3657.430	338.470
jessmail	26207.840	33345.230	7057.390
joico	705.520	2015.100	1309.580
kaaral	4412.430	5086.070	673.640
kamill	63.010	81.490	18.480
kapous	11927.160	14093.080	2165.920
keyn	236.380	431.620	199.270
kimogys	330.910	525.200	94.290
kimse	330.040	632.040	302.000
kinetica	6334.250	6945.260	611.010
kiss	421.550	817.330	395.780
kocostar	310.850	594.930	284.080
koelcia	55.500	112.750	57.250
koelf	422.730	507.290	84.560
konad	739.830	810.670	70.840
kosmekka	1181.440	1813.370	631.930
laboratorium	246.500	312.520	66.020

hadoop@ip-172-31-23-34:~				
konad	739.830	810.670	70.640	
kosmekka	1181.440		1813.370	631.930
laboratorium	246.500	312.520	66.020	
lador	2083.610		2471.530	387.920
ladykin	125.650	170.570	44.920	
latinoil		249.520	384.590	135.070
levissime	2227.500		3085.310	857.810
levrana	2243.560		3664.100	1420.540
lanail	5892.840		15394.240	10501.400
limon	1105.060	340.970	44.910	
limoni	1308.900		1796.600	487.700
lovely	8704.380		11939.060	3234.680
lowence	242.840	567.750	324.910	
mane	66.790	260.260	193.470	
marathon		7280.750		10273.100
markell	1768.750		2834.430	2992.350
marutaka-foot	49.220	109.330	60.110	
masura	31266.080		33058.470	1792.390
matrix	340.250		570.200	463.490
maulala	109.040	446.320	37.280	
mettler	5373.450		6457.160	1083.710
mily	3904.940		5642.010	1737.070
misikin	158.040	293.070	135.030	
missha	1293.830		2150.280	856.450
moyou	5.710	10.280	4.570	
nagaraku		4369.740		5327.680
nefertiti	233.520	366.640	133.120	
neoleer	43.410	51.700	8.290	
nirvel	160.040	234.330	71.290	
nirville	147.830	232.660		315.400
nitris	8425.110		9841.650	1416.240
orly	902.380	931.090	28.710	
osmo	645.580	762.310	116.730	
ovale	2.540	3.100	0.560	
plazan	101.370	194.010	92.640	
polarus	6013.720		11371.930	5358.210
profepil		93.360	118.020	24.660
profthenna	679.230	736.850	57.620	
protokeratin	201.250	456.790	255.540	
provoc	82.990	1063.820		235.830
rayyan	13.800	13.940	10.140	
refectocil		2716.180	3475.580	759.400
rosi	3077.040		3841.560	764.520
roublöff	3491.360		4913.770	1422.410

roublöff	3491.360	4913.770	1422.410
runail	71539.280	76758.660	5219.380
s.care	412.680	913.070	500.390
sanoto	157.140	1209.680	1052.540
severina		4775.880	6120.480
shary	871.960	1176.490	304.530
shik	3341.200		4839.720
skinity	8.880	12.440	3.560
skinlite		651.940	890.450
smart	4457.260		5902.140
soleo	204.200	212.530	8.330
solomeya		1899.700	2685.800
sophin	1067.860		1515.520
staleks	8519.730		11875.610
strong	29196.630		38671.270
supertan		50.370	66.510
swarovski		1887.930	3043.160
tertio	236.160	245.800	9.640
treaclemoon		163.370	181.490
trind	298.070	542.960	244.890
uno	35302.030		51039.750
uskusi	5142.270		5690.310
veraclara		50.110	71.210
vilenta	197.600	231.210	33.610
yoko	8756.910		11707.880
yu-r	271.410	673.710	402.300
zeitun	708.660	2009.630	1300.970
Time taken:	31.957	seconds,	Fetched: 152 row(s)
hive>	██████████		

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) total_spend
from dyn_part_ecom
where event_type = 'purchase'
group by user_id
order by total_spend desc
limit 10;
```

```
-----  
 VERTICES    MODE      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED  
-----  
Map 1 ..... container  SUCCEEDED   7       7       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: 29.28 s  
-----  
OK  
user_id total_spend  
557790271     2715.870  
150318419     1645.970  
562167663     1352.850  
531900924     1329.450  
557850743     1295.480  
522130011     1185.390  
561592095     1109.700  
431950134     1097.590  
566576008     1056.360  
521347209     1040.910  
Time taken: 29.872 seconds, Fetched: 10 row(s)  
hive> █
```