



# E-COMMERCE HIVE CASE STUDY

**Submitted By:**

Abhishek De

Rahul Roy

## PROBLEM STATEMENT

With online sales gaining popularity, tech companies are exploring ways to improve their sales by analysing customer behaviour and gaining insights about product trends.

Furthermore, the websites make it easier for customers to find the products they require without much scavenging. Needless to say, the role of big data analysts is among the most sought-after job profiles of this decade.

Therefore, as a big data analyst, to extract data and gather insights from a real-life data set of an e-commerce company.

The implementation phase can be divided into the following parts:

- Copying the data set into the HDFS:
  - Launch an EMR cluster that utilizes the Hive services, and
  - Move the data from the S3 bucket into the HDFS.
- Creating the database and launching Hive queries on your EMR cluster:
  - Create the structure of your database,
  - Use optimized techniques to run the queries as efficiently as possible.
  - Show the improvement of the performance after using optimization on any single query.
  - Run Hive queries to answer the question.
- Cleaning up
  - Drop the database, and
  - Terminate the cluster.

The following steps are performed in the hive case study:

- 1) Connect the local machine to the master node using SSH:

```
hadoop@ip-172-31-69-190:~$
https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
60 package(s) needed for security, out of 106 available
Run "sudo yum update" to apply all updates.

EEEEEEEEEEEEEEEEEEEE MMMMMMM MMMMMMM RRRRRRRRRRRRRRRR
E::::::::::::::::::::E M::::::::M M::::::::M R:::::::::R
EE::::::::::::::::::::E M::::::::M M::::::::M R:::::::::R
E::::E EEEEE M::::::::M M::::::::M RR::R R:::R
E::::E M::::::::M M::M M::M M::M R::R R:::R
E::::EEEEEEEEEE M::M M::M M::M M::M R::RRRRR:::R
E::::::::::::E M::M M::M M::M M::M R::RRRRR:::R
E::::E M::M M::M M::M M::M R::R R:::R
E::::E EEEEE M::M M M M M M R::R R:::R
EE::::::::::::E M::M M::M M::M R::R R:::R
E::::::::::::E M::M M::M M::M RR::R R:::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM MMMMMMM RRRRRR RRRRRR

[hadoop@ip-172-31-69-190 ~]$ hadoop fs -mkdir /user/hive/cosmetic_sales
[hadoop@ip-172-31-69-190 ~]$ aws s3 ls e-commerce-events-ml
2020-03-17 11:47:09 545839412 2019-Nov.csv
2020-03-17 11:37:31 482542278 2019-Oct.csv
[hadoop@ip-172-31-69-190 ~]$
```

- 2) Create a folder name '**cosmetic\_sales**' in the HDFS using the following command:

***hadoop fs -mkdir /user/hive/cosmetic\_sales***  
***aws s3 ls e-commerce-events-ml***

```
[hadoop@ip-172-31-69-190 ~]$ hadoop fs -mkdir /user/hive/cosmetic_sales
[hadoop@ip-172-31-69-190 ~]$ aws s3 ls e-commerce-events-ml
2020-03-17 11:47:09 545839412 2019-Nov.csv
2020-03-17 11:37:31 482542278 2019-Oct.csv
```

- 3) Import the data to the folder '**cosmetic\_sales**' in the HDFS using the following command:

***hadoop distcp s3://e-commerce-events-ml/ /user/hive/cosmetic\_sales/***

- 4) Now we have imported the data in the HDFS. To see the imported data run the following command :

***hadoop fs -ls /user/hive/cosmetic\_sales***

```
[hadoop@ip-172-31-78-144 ~]$ hadoop fs -ls /user/hive/cosmetic_sales
Found 2 items
-rw-r--r--  1 hadoop hadoop  545839412 2021-05-25 10:33 /user/hive/cosmetic_sales/2019-Nov.csv
-rw-r--r--  1 hadoop hadoop  482542278 2021-05-25 10:33 /user/hive/cosmetic_sales/2019-Oct.csv
```

Here, we can see that both the files are uploaded in the HDFS.

- 5) Launch the Hive Service. For this run the command “**hive**”.
- 6) Creating and using the database named ‘**cosmetic\_sales**’ using the following query :

```
hive> create database cosmetic_sales;
OK
Time taken: 0.861 seconds
hive> use cosmetic_sales;
OK
Time taken: 0.045 seconds
hive> █
```

- 7) Create the external table by using the following query:

***create table if not exists sales\_table(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' stored as textfile location '/user/hive/cosmetic\_sales/' tblproperties("skip.header.line.count"="1");***

```
hive> --Creating table
hive> create table if not exists sales_table(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' stored as textfile location '/user/hive/cosmetic_sales/' tblproperties("skip.header.line.count"="1");
OK
Time taken: 0.356 seconds
hive> █
```

8) Describe the table 'sales\_table' by using the following query :

***describe sales\_table ;***

```
hive> --Describing tables
hive> describe sales_table;
OK
col_name      data_type      comment
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: 0.056 seconds, Fetched: 9 row(s)
hive> █
```

9) To show the headers for all the queries use the following query :

***set hive.cli.print.header=true ;***

10) Create the partitioning and bucketing using the following command :

***create table cosmetic\_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 12 buckets row format serde 'org.apache.hadoop.hive.serde2.OpenCSVSerde' stored as textfile;***

```
hive> --Creating table for partitioning and bucketing
hive> create table cosmetic_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 12 buckets row format serde 'org.apache.hadoop.hive.serde2.OpenCSVSerde' stored as textfile;
OK
Time taken: 0.086 seconds
hive> █
```

[illegible]



## QUERY OPTIMIZATION

We have created the partitioned and bucketed table named '**cosmetic\_bucket**' to optimize the queries. Let's see how we can optimize queries through the example.

### 1) Fetching the first 10 rows of **sales\_table**.

```
hive> select * from sales_table limit 10;
OK
sales_table.event_time sales_table.event_type sales_table.product_id sales_table.category_id sales_table.category_code sales_table.brand sales_table.price
sales_table.user_id sales_table.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-alcc-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-alcc-af0575a34ffb
2019-11-01 00:00:24 UTC remove_from_cart 5826182 1487580007483048900 3.33 553329724 2067216c-31b5-455d-alcc-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
2019-11-01 00:00:37 UTC view 5870803 1487580007675986893 milv 0.79 429913900 2f0bff3c-252f-4fe6-afcd-5d8a6a92839a
Time taken: 0.205 seconds, Fetched: 10 row(s)
hive>
```

Here we can see that the time taken to execute the query in the table '**sales\_table**' is 0.205 seconds.

### 2) Fetching the first 10 rows of **cosmetic\_bucket**.

```
hive> select * from cosmetic_bucket limit 10;
OK
cosmetic_bucket.event_time cosmetic_bucket.product_id cosmetic_bucket.category_id cosmetic_bucket.category_code cosmetic_bucket.brand cosmetic_bucket.
price cosmetic_bucket.user_id cosmetic_bucket.user_session cosmetic_bucket.event_type
2019-10-11 07:53:13 UTC 5813484 1487580005671109489 masura 1.73 559060196 2338c843-45de-43e5-ac06-2804b629ccf9 cart
2019-10-09 11:47:14 UTC 5689725 1487580007852147670 staleks 13.17 404502068 928c919b-42de-4b94-afd4-19423944f5f0 cart
2019-10-08 18:31:54 UTC 5870696 1487580008246412266 4.60 100787781 188a44b5-83f1-4f19-8a93-2fa670f2ec08 cart
2019-10-07 21:38:36 UTC 5797252 1638456119066100510 pole 4.11 533267875 4d44c69e-ea11-4fa6-8f97-39a72e6831cb cart
2019-10-08 18:31:55 UTC 5887003 1487580006317032337 7.94 459127083 76f0c023-c35e-4ca9-8146-34bc5c94382e cart
2019-10-08 18:31:55 UTC 5861279 1487580006317032337 30.95 558176613 6bcac932-1da0-46bb-bea6-6cd19ac6be00 cart
2019-10-09 11:47:14 UTC 5821228 1487580005461394279 bluesky 3.97 320278663 28885b28-a536-40b5-98f3-dbb7faa69e26 cart
2019-10-09 11:47:13 UTC 5777442 1487580009143992338 lakme 11.35 558429809 99d4f1b7-8c09-46ae-9673-60362a44515e cart
2019-10-09 11:47:13 UTC 5847870 1487580006317032337 1.90 558419860 91583ed9-f240-46ea-bcaa-e2ef8bb54003 cart
2019-10-09 11:47:13 UTC 5786837 1783999068909863670 smart 5.56 556485145 4d5939fb-87d2-4c41-b62c-8351fe31cc49 cart
Time taken: 0.361 seconds, Fetched: 10 row(s)
hive>
```

Here we can see that the time taken to execute the query in the table '**cosmetic\_bucket**' is 0.361 sec.

**CONCLUSION:** We can conclude that the partitioned and bucketed table '**cosmetic\_bucket**' takes less time as compared to table '**sales\_table**' to execute the query. So now onwards, we will perform all the queries on the partitioned and bucketed table '**cosmetic\_table**'.

## QUESTIONS AND ANSWERS

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

Answer:

***select sum(price) as revenue from cosmetic\_bucket where month(event\_time)=10 and event\_type='purchase';***

```
hive> select sum(price) as revenue from cosmetic_bucket where month(event_time)=
10 and event_type='purchase';
Query ID = hadoop_20210525145951_f9fec9d9-6688-434f-ab9c-8c9f545d6439
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1621953626007_0003)
```

|           | VERTICES | MODE      | STATUS    | TOTAL | COMPLETED | RUNNING | PENDING | FAILED | KILLED |
|-----------|----------|-----------|-----------|-------|-----------|---------|---------|--------|--------|
| Map 1     | .....    | container | SUCCEEDED | 3     | 3         | 0       | 0       | 0      | 0      |
| Reducer 2 | .....    | container | SUCCEEDED | 1     | 1         | 0       | 0       | 0      | 0      |

```
VERTICES: 02/02 [=====>>] 100% ELAPSED TIME: 23.33 s
```

```
OK
revenue
1211538.4299998006
Time taken: 34.131 seconds, Fetched: 1 row(s)
hive> █
```

The total revenue generated due to the purchases made in October month is  
**1211538.4299998006 .**



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

Answer:

***select month(event\_time), sum(price) from cosmetic\_bucket where year(event\_time)=2019 and event\_type='purchase' group by month(event\_time);***

```
hive> select month(event_time), sum(price) from cosmetic_bucket where year(event_time)=2019 and event_type='purchase' group by month(event_time);
Query ID = hadoop_20210525150300_877ca366-e096-48ab-ba91-0b08ef2fef4f
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1621953626007_0003)

-----
VERTICES      MODE           STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container    SUCCEEDED    3        3          0        0        0        0
Reducer 2 ..... container    SUCCEEDED    1        1          0        0        0        0
-----
VERTICES: 02/02 [=====>>>] 100% ELAPSED TIME: 22.61 s
-----
OK
_c0      _c1
10      1211538.4299998006
11      1531016.9000000061
Time taken: 23.378 seconds, Fetched: 2 row(s)
hive> 
```

The total sum of purchases in the month of October is **1211538.4299998006** and the total sum of purchases in the month of November is **1531016.9000000061**.

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

Answer:

***with diff\_revenue as(select sum(case when month(event\_time) = '10' then price else 0 end) as Oct\_purchase, sum(case when month(event\_time) = '11' then price else 0 end) as Nov\_purchase from cosmetic\_bucket where event\_type= 'purchase') select (Nov\_purchase - Oct\_purchase) as difference\_revenue from diff\_revenue;***

```
hive> with diff_revenue as(select sum(case when month(event_time) = '10' then price else 0 end) as Oct_purchase, sum(case when month(event_time) = '11' then price else 0 end) as Nov_purchase from cosmetic_bucket where event_type= 'purchase') select (Nov_purchase - Oct_purchase) as difference_revenue from diff_revenue;
Query ID = hadoop_20210525151439_c52d7abc-4b43-40e5-ab92-748ca31f931e
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1621953626007_0004)
```

|           | VERTICES | MODE      | STATUS    | TOTAL | COMPLETED | RUNNING | PENDING | FAILED | KILLED |
|-----------|----------|-----------|-----------|-------|-----------|---------|---------|--------|--------|
| Map 1     | .....    | container | SUCCEEDED | 3     | 3         | 0       | 0       | 0      | 0      |
| Reducer 2 | .....    | container | SUCCEEDED | 1     | 1         | 0       | 0       | 0      | 0      |

```
VERTICES: 02/02 [=====>>>] 100% ELAPSED TIME: 28.69 s
OK
difference_revenue
319478.4700002603
Time taken: 37.167 seconds, Fetched: 1 row(s)
hive> █
```

The change in revenue generated due to purchases from October to November is **319478.470000**.

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

Answer:

***select distinct(category\_code) from cosmetic\_bucket;***

```
hive> select distinct(category_code) from cosmetic_bucket;
Query ID = hadoop_20210525151759_6f36390f-b784-4b0e-9e6e-d38da40c4800
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1621953626007_0004)

-----
VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container  SUCCEEDED   10         10         0         0         0         0
Reducer 2 ..... container  SUCCEEDED    5          5         0         0         0         0
-----
VERTICES: 02/02 [=====>>] 100%  ELAPSED TIME: 50.37 s
-----
OK
category_code

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: 51.051 seconds, Fetched: 12 row(s)
hive> █
```

The distinct categories of the products are as follows :

- i) ***Accessories.cosmetic\_bag***
- ii) ***Stationary.cartridge***
- iii) ***Accessories.bag***
- iv) ***Appliances.environment.vacuum***
- v) ***Furniture.living\_room.chair***
- vi) ***Sport.diving***
- vii) ***Appliances.person.hair\_cutter***
- viii) ***Appliances.environment.air\_conditioner***
- ix) ***Apparel.glove***
- x) ***Furniture.bathroom.bath***
- xi) ***Furniture.living\_room.cabinet***

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

Answer:

***select category\_code, count(product\_id) as total\_order from cosmetic\_bucket group by category\_code;***

```
hive> select category_code, count(product_id) as total_order from cosmetic_bucket group by category_code;
Query ID = hadoop_20210525152014_e06d4c64-7dba-4189-ac8d-3b463a5b625d
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1621953626007_0004)
```

| VERTICES        | MODE      | STATUS    | TOTAL | COMPLETED | RUNNING | PENDING | FAILED | KILLED |
|-----------------|-----------|-----------|-------|-----------|---------|---------|--------|--------|
| Map 1 .....     | container | SUCCEEDED | 10    | 10        | 0       | 0       | 0      | 0      |
| Reducer 2 ..... | container | SUCCEEDED | 5     | 5         | 0       | 0       | 0      | 0      |

```
VERTICES: 02/02 [=====>>] 100% ELAPSED TIME: 50.36 s
OK
category_code  total_order
8594895
accessories.cosmetic_bag      1248
stationery.cartridge          26722
accessories.bag 11681
appliances.environment.vacuum  59761
furniture.living_room.chair    308
sport.diving 2
appliances.personal.hair_cutter 1643
appliances.environment.air_conditioner 332
apparel.glove 18232
furniture.bathroom.bath 9857
furniture.living_room.cabinet 13439
Time taken: 51.173 seconds, Fetched: 12 row(s)
hive> █
```

The total number of products available under each category is as follows

- : i) Accessories.cosmetic\_bag - 1248***
- ii) Stationary.cartridge - 26722***
- iii) Accessories.bag – 11681***
- iv) Appliances.environment.vacuum – 59761***
- v) Furniture.living\_room.chair - 308***
- vi) Sport.diving - 2***
- vii) Appliances.person.hair\_cutter -1643***
- viii) Appliances.environment.air\_conditioner - 332***
- ix) Apparel.glove - 18232***
- x) Furniture.bathroom.bath - 9857***
- xi) Furniture.living\_room.cabinet – 13439***

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

Answer:

***select brand, sum(price) as sales from cosmetic\_bucket where brand is not null and event\_type='purchase' group by brand order by sales desc limit 2;***

```
hive> select brand, sum(price) as sales from cosmetic_bucket where brand is not null and event_type='purchase' group by brand order by sales desc limit 2;
Query ID = hadoop_20210525152750_75062a8b-5a8e-4695-8427-7e361a6ec321
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1621953626007_0004)
```

| VERTICES        | MODE      | STATUS    | TOTAL | COMPLETED | RUNNING | PENDING | FAILED | KILLED |
|-----------------|-----------|-----------|-------|-----------|---------|---------|--------|--------|
| Map 1 .....     | container | SUCCEEDED | 3     | 3         | 0       | 0       | 0      | 0      |
| Reducer 2 ..... | container | SUCCEEDED | 1     | 1         | 0       | 0       | 0      | 0      |
| Reducer 3 ..... | container | SUCCEEDED | 1     | 1         | 0       | 0       | 0      | 0      |

```
VERTICES: 03/03 [=====>>>] 100% ELAPSED TIME: 21.83 s
OK
brand    sales
      1094188.3000000082
runail 148297.93999998947
Time taken: 22.562 seconds, Fetched: 2 row(s)
hive> 
```



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

Answer:

***select a.brand from(select brand, sum(price) as sales from cosmetic\_bucket where brand <>' ' and month(event\_time)=10 group by brand) as a inner join (select brand ,sum(price) as sales from cosmetic\_bucket where brand !=' ' and month(event\_time)=11 group by brand) as b on a.brand=b.brand where b.sales-a.sales>0 limit 5 ;***

```
hive> select a.brand from(select brand, sum(price) as sales from cosmetic_bucket where brand <>' ' and month(event_time)=10 group by brand) as a inner join (select brand ,sum(price) as sales from cosmetic_bucket where brand !=' ' and month(event_time)=11 group by brand) as b on a.brand=b.brand where b.sales-a.sales>0 limit 5 ;
Query ID = hadoop_20210525154014_cc1f7b8f-6f7e-4438-a132-ce55004ad119
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1621953626007_0004)
```

| VERTICES        | MODE      | STATUS    | TOTAL | COMPLETED | RUNNING | PENDING | FAILED | KILLED |
|-----------------|-----------|-----------|-------|-----------|---------|---------|--------|--------|
| Map 1 .....     | container | SUCCEEDED | 10    | 10        | 0       | 0       | 0      | 0      |
| Map 4 .....     | container | SUCCEEDED | 10    | 10        | 0       | 0       | 0      | 0      |
| Reducer 2 ..... | container | SUCCEEDED | 3     | 3         | 0       | 0       | 0      | 0      |
| Reducer 3 ..... | container | SUCCEEDED | 3     | 3         | 0       | 0       | 0      | 0      |
| Reducer 5 ..... | container | SUCCEEDED | 3     | 3         | 0       | 0       | 0      | 0      |

```
VERTICES: 05/05 [=====>>>] 100% ELAPSED TIME: 89.64 s
OK
a.brand
airnails
artex
bpw.style
carmex
concept
Time taken: 90.369 seconds, Fetched: 5 row(s)
hive> 
```

The following top 5 brands increased their sales from October to November:

- i) ***airnails***
- ii) ***artex***
- iii) ***bpw.style***
- iv) ***carmex***
- v) ***concept***

H. 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.

Answer:

***select user\_id, sum(price) as spend from cosmetic\_bucket group by user\_id order by spend limit 10;***

```
hive> select user_id, sum(price) as spend from cosmetic_bucket group by user_id order by spend limit 10;
Query ID = hadoop_20210525154417_e5151147-3c04-4544-b707-aff4890d112d
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1621953626007_0004)
```

|                 | VERTICES  | MODE      | STATUS | TOTAL | COMPLETED | RUNNING | PENDING | FAILED | KILLED |
|-----------------|-----------|-----------|--------|-------|-----------|---------|---------|--------|--------|
| Map 1 .....     | container | SUCCEEDED | 10     | 10    | 0         | 0       | 0       | 0      | 0      |
| Reducer 2 ..... | container | SUCCEEDED | 5      | 5     | 0         | 0       | 0       | 0      | 0      |
| Reducer 3 ..... | container | SUCCEEDED | 1      | 1     | 0         | 0       | 0       | 0      | 0      |

```
VERTICES: 03/03 [=====>>] 100% ELAPSED TIME: 62.40 s
OK
user_id spend
291566397    0.0
578464010    0.0
487309736    0.0
426258490    0.0
347894786    0.0
482884551    0.0
436311580    0.0
577823012    0.0
436417977    0.0
479192613    0.0
Time taken: 62.999 seconds, Fetched: 10 row(s)
hive> 
```