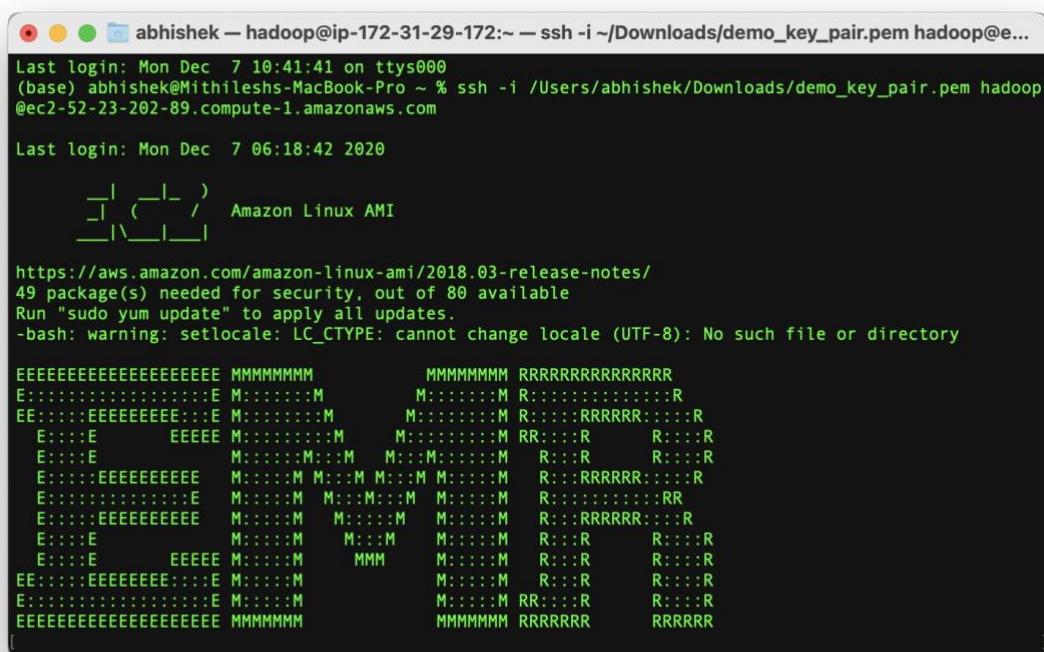


Hive Case Study

The following steps are performed in the hive case study :

- 1) Connect the local machine to the master node using SSH by running the following command :

```
ssh -i /Users/abhishek/Downloads/demo_key_pair.pem hadoop@ec2-52-23-202-89.compute-1.amazonaws.com
```



The image shows a terminal window titled 'abhishek — hadoop@ip-172-31-29-172:~ — ssh -i ~/Downloads/demo_key_pair.pem hadoop@e...'. The terminal output shows the user 'abhishek' logging in from a MacBook-Pro to an Amazon Linux AMI instance. It displays the last login time, the Amazon Linux logo, and system updates. At the bottom, there is a large ASCII art graphic composed of 'E', 'M', and 'R' characters.

```
abhishek — hadoop@ip-172-31-29-172:~ — ssh -i ~/Downloads/demo_key_pair.pem hadoop@e...
Last login: Mon Dec 7 10:41:41 on ttys000
(base) abhishek@Mithileshs-MacBook-Pro ~ % ssh -i /Users/abhishek/Downloads/demo_key_pair.pem hadoop
@ec2-52-23-202-89.compute-1.amazonaws.com

Last login: Mon Dec 7 06:18:42 2020

 _ _ | _ _ |
 _ _ | _ _ | /   Amazon Linux AMI

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
49 package(s) needed for security, out of 80 available
Run "sudo yum update" to apply all updates.
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file or directory

EEEEEEEEEEEEEEEEEEEE MMMMMMMM MMMMMMMM RRRRRRRRRRRRRRRR
E:EEEEEEEEEEEEEEEEEEEE M:EEEEEEEE M:EEEEEEEE M:EEEEEEEE R:EEEEEEEE R:EEEEEEEE
EE:EEEEEEEEEEEEEEEEEEEE M:EEEEEEEE M:EEEEEEEE M:EEEEEEEE R:EEEEEEEE R:EEEEEEEE
E:EEEEEEEEEEEEEEEEEEEE M:EEEEEEEE M:EEEEEEEE M:EEEEEEEE R:EEEEEEEE R:EEEEEEEE
E:EEEEEEEEEEEEEEEEEEEE M:EEEEEEEE M:EEEEEEEE M:EEEEEEEE R:EEEEEEEE R:EEEEEEEE
E:EEEEEEEEEEEEEEEEEEEE M:EEEEEEEE M:EEEEEEEE M:EEEEEEEE R:EEEEEEEE R:EEEEEEEE
E:EEEEEEEEEEEEEEEEEEEE M:EEEEEEEE M:EEEEEEEE M:EEEEEEEE R:EEEEEEEE R:EEEEEEEE
E:EEEEEEEEEEEEEEEEEEEE M:EEEEEEEE M:EEEEEEEE M:EEEEEEEE R:EEEEEEEE R:EEEEEEEE
E:EEEEEEEEEEEEEEEEEEEE M:EEEEEEEE M:EEEEEEEE M:EEEEEEEE R:EEEEEEEE R:EEEEEEEE
E:EEEEEEEEEEEEEEEEEEEE M:EEEEEEEE M:EEEEEEEE M:EEEEEEEE R:EEEEEEEE R:EEEEEEEE
EEEEEEEEEEEEEEEEEEEEEEEE MMMMMMMM MMMMMMMM RRRRRRRR RRRRRR
```

- 2) Create a folder named 'abhishek' in the HDFS using the following command :

```
hadoop fs -mkdir /abhishek
```

- 3) To check the created folder in the HDFS use the following command :

```
hadoop fs -ls /
```

```
abhishek — hadoop@ip-172-31-29-172:~ — ssh -i ~/Downloads/...  
[hadoop@ip-172-31-29-172 ~]$ hadoop fs -mkdir /abhishek  
[hadoop@ip-172-31-29-172 ~]$ hadoop fs -ls /  
Found 5 items  
drwxr-xr-x   - hadoop hadoop          0 2020-12-07 07:00 /abhishek  
drwxr-xr-x   - hdfs  hadoop          0 2020-12-07 05:09 /apps  
drwxrwxrwt   - hdfs  hadoop          0 2020-12-07 05:11 /tmp  
drwxr-xr-x   - hdfs  hadoop          0 2020-12-07 05:09 /user  
drwxr-xr-x   - hdfs  hadoop          0 2020-12-07 05:09 /var
```

We can see in the screenshot that the folder named 'abhishek' is created in the HDFS.

- 4) Import the data to the folder 'abhishek' in the HDFS using the following two commands :

i) *`hadoop distcp s3://mithilesh-hive-case-study/Mithilesh/2019-Oct.csv /abhishek`*

ii) *`hadoop distcp s3://mithilesh-hive-case-study/Mithilesh/2019-Nov.csv /abhishek`*

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

`hadoop fs -ls /abhishek/`

```
abhishek — hadoop@ip-172-31-29-172:~ — ssh -i ~/Downloads/demo_key...  
[hadoop@ip-172-31-29-172 ~]$ hadoop fs -ls /abhishek  
Found 2 items  
-rw-r--r--   1 hadoop hadoop  545839412 2020-12-07 07:12 /abhishek/2019-Nov.csv  
-rw-r--r--   1 hadoop hadoop  482542278 2020-12-07 07:10 /abhishek/2019-Oct.csv
```

We can see both the files uploaded in the HDFS.

- 6) Launch the hive service. For this run the command 'hive'.
7) Create the database named 'casestudy' using the following query :

`create database if not exists casestudy ;`

- 8) To see the created database run the following query :

`show databases ;`

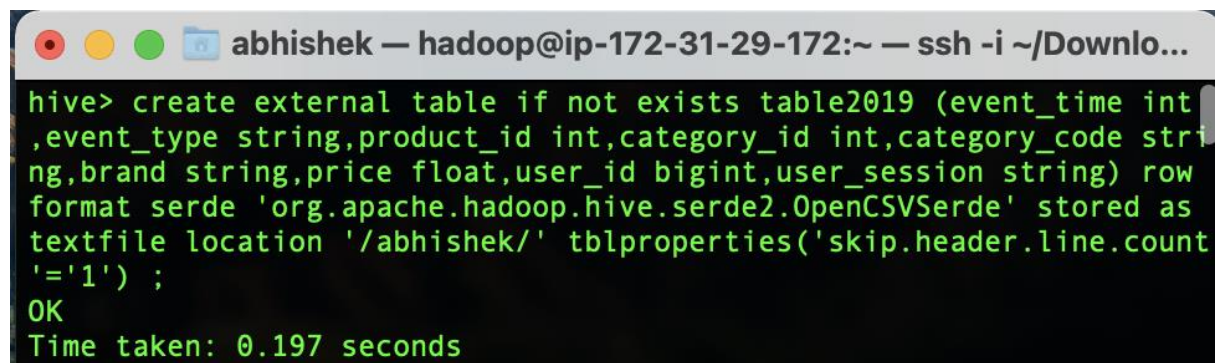
```
[hive> show databases ;
OK
casestudy
default
Time taken: 0.023 seconds, Fetched: 2 row(s)
```

We can see in the screenshot above that the database named 'casestudy' has created.

9) To use the database 'casestudy' run the command 'use casestudy;'.

10) Create the external table by using the following query :

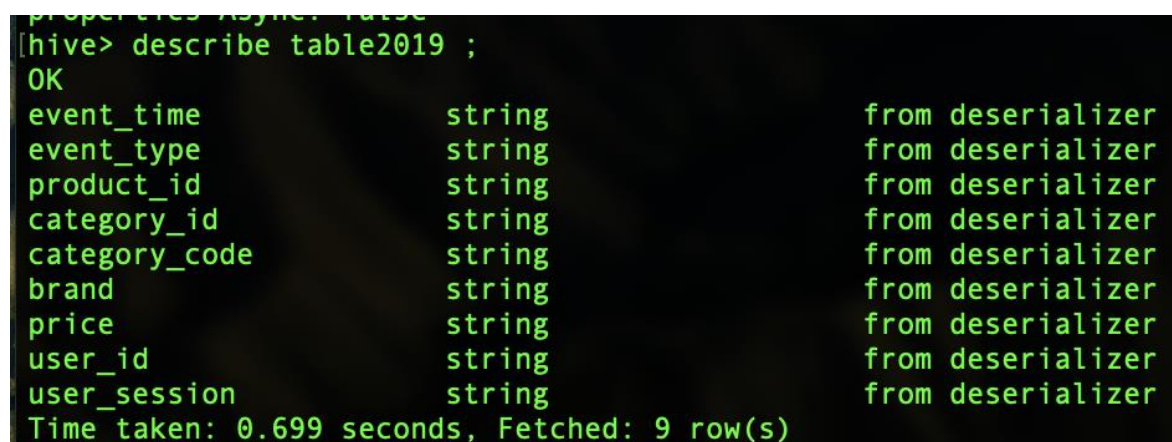
```
create table if not exists table2019 ( event_time int , event_type string ,
product_id int , category_id int , category_code string , brand string , price float
, user_id int , user_session string ) row format serde
'org.apache.hadoop.hive.serde2.OpenCSVSerde' stored as textfile location
'/abhishek/' tblproperties ("skip.header.line.count"="1");
```



```
abhishek — hadoop@ip-172-31-29-172:~ — ssh -i ~/Downlo...
hive> create external table if not exists table2019 (event_time int
,event_type string,product_id int,category_id int,category_code stri
ng,brand string,price float,user_id bigint,user_session string) row
format serde 'org.apache.hadoop.hive.serde2.OpenCSVSerde' stored as
textfile location '/abhishek/' tblproperties('skip.header.line.count
'='1') ;
OK
Time taken: 0.197 seconds
```

11) Describe the table 'table2019' by using the following query :

```
describe table2019 ;
```



```
properties_async: false
[hive> describe table2019 ;
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: 0.699 seconds, Fetched: 9 row(s)
```

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

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

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

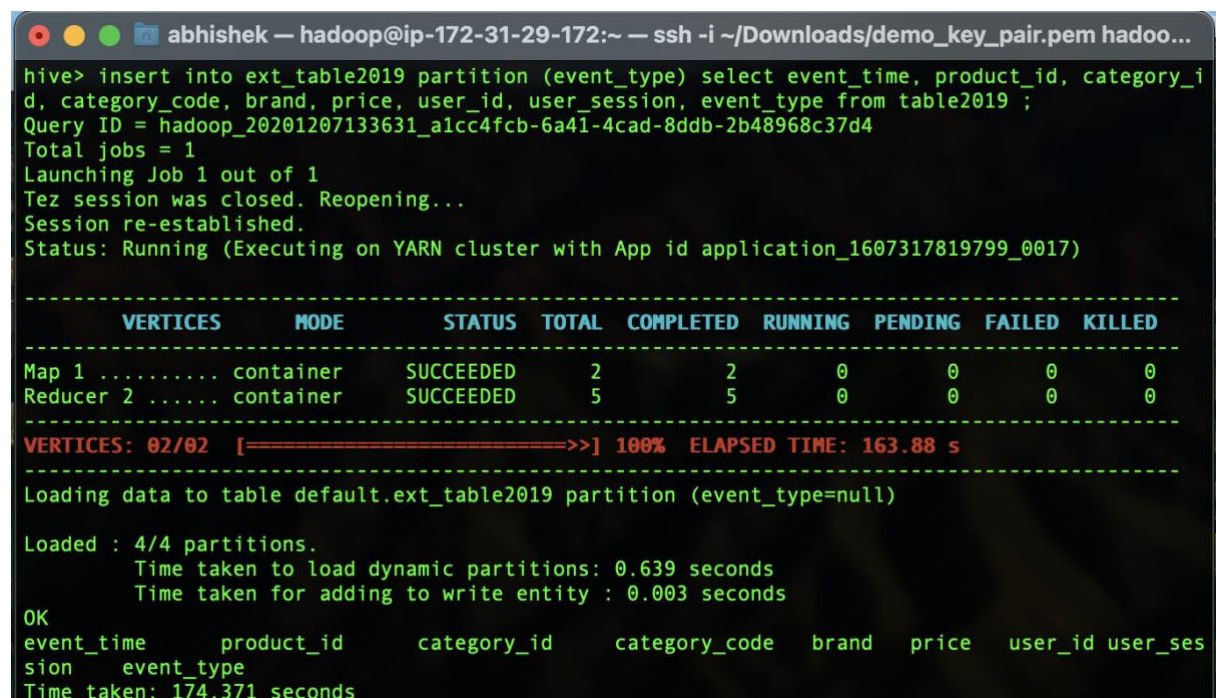
```
create external table if not exists ext_table2019 (event_time
timestamp,product_id string,category_id string,category_code string,brand
string,price float,user_id bigint,user_session string) partitioned by (event_type
string) clustered by (category_code) into 12 buckets row format serde
'org.apache.hadoop.hive.serde2.OpenCSVSerde' stored as textfile;
```

14) Set dynamic partitioning mode to nonstrict using the following command :

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

15) Load the data in the partitioned and bucketed table named 'ext_table2019' using the following command :

```
insert into ext_table2019 partition (event_type) select event_time, product_id,
category_id, category_code, brand, price, user_id, user_session, event_type
from table2019 ;
```



```
abhishek — hadoop@ip-172-31-29-172:~ — ssh -i ~/Downloads/demo_key_pair.pem hadoo...
hive> insert into ext_table2019 partition (event_type) select event_time, product_id, category_id,
category_code, brand, price, user_id, user_session, event_type from table2019 ;
Query ID = hadoop_20201207133631_a1cc4fcb-6a41-4cad-8ddb-2b48968c37d4
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_1607317819799_0017)
```

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: 163.88 s
Loading data to table default.ext_table2019 partition (event_type=null)
Loaded : 4/4 partitions.
Time taken to load dynamic partitions: 0.639 seconds
Time taken for adding to write entity : 0.003 seconds
OK
event_time      product_id      category_id      category_code    brand    price    user_id user_ses
sion      event_type
Time taken: 174.371 seconds
```


Query Optimization

We have created the partitioned and bucketed table named 'ext_table2019' to optimize the queries. Let's see how we can optimize queries through the example.

Q. Fetch the first 10 rows.

First we will fetch the first 10 rows of the table 'table2019'.

```
hive> select * from table2019 limit 10 ;
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 556138645 57ed222e-a54a-4907-9944-5a875c2d774f
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
2019-11-01 00:00:37 UTC view 5870803 1487580007675986893 milv 0.79 429913900 2f0bff3c-252f-4fe6-afcd-5d8a6a92839a
Time taken: 2.574 seconds, Fetched: 10 row(s)
```

Here we can see that the time taken to execute the query in the table 'table2019' is 2.574 seconds.

Now, we will fetch the first 10 rows of the partitioned and bucketed table 'ext_table2019'.

```
hive> select * from ext_table2019 limit 10 ;
OK
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 17839990689090863670 smart 5.56 556485145 4d5939fb-87d2-4c41-b62c-8351fe31cc49 cart
Time taken: 0.286 seconds, Fetched: 10 row(s)
```

Here we can see that the time taken to execute the query in the partitioned and bucketed table 'ext_table2019' is 0.286 seconds.

From this example, we can conclude that the partitioned and bucketed table 'ext_table2019' takes less time as compared to table 'table2019' to execute the query. So now onwards, we will perform all the queries on the partitioned and bucketed table 'ext_table2019'.

Questions and Answers

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

Answer : *select sum(price) as revenue from ext_table2019 where month(event_time)=10 and event_type = 'purchase';*

```
hive> select sum(price) as revenue from ext_table2019 where month(event_time)=10 and event_type = 'purchase';
Query ID = hadoop_20201207135854_c67e84e2-1970-410b-8605-584e1c39745c
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_1607317819799_0019)

-----
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: 17.52 s
-----
OK
1211538.4299997962
Time taken: 27.964 seconds, Fetched: 1 row(s)
```

The total revenue generated due to the purchases made in October month is 1211538.4299997962 .

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

Answer : *select month(event_time), sum(price) from ext_table2019 where year(event_time)=2019 and event_type='purchase' group by month(event_time);*

```
hive> select month(event_time), sum(price) from ext_table2019 where year(event_time)=2019 and event_type='purchase' group by month(event_time);
Query ID = hadoop_20201207140338_6de328d5-34fb-4cf1-b7b5-5138d4899f0a
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1607317819799_0019)

-----
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: 18.12 s
-----
OK
10      1211538.4299997962
11      1531016.9000000611
Time taken: 18.973 seconds, Fetched: 2 row(s)
```

The total sum of purchases in the month of October is 1211538.4299997962 and the total sum of purchases in the month of November is 1531016.9000000611 .

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

Answer : *select sum(case when month(from_unixtime(unix_timestamp(event_time,'yyyy-MM-dd HH:mm:ss'))) = 10 then cast (price as float) else - 1 * cast(price as float)end) as change_in_revenue from ext_table2019 where month(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss'))) IN (10, 11) and event_type = 'purchase';*

```
hive> select sum(case when month(from_unixtime(unix_timestamp(event_time,'yyyy-MM-dd HH:mm:ss'))) = 10 then cast (price as float) else - 1 * cast(price as float)end) as change_in_revenue from ext_table2019 where month(from_unixtime(unix_timestamp(event_time, 'yyyy-MM-dd HH:mm:ss'))) IN (10, 11) and event_type = 'purchase';
Query ID = hadoop_20201207140827_a312b891-d520-4ef9-ba05-fdf564839a78
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1607317819799_0019)

-----
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: 18.29 s
-----
OK
-319478.469592195
Time taken: 19.0 seconds, Fetched: 1 row(s)
```

The change in revenue generated due to purchases from October to November is 319478.469592195 . The negative sign indicates that the revenue of November is less than the revenue of October .

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

Answer : `select distinct(category_code) from ext_table2019 ;`

```
abhishek — hadoop@ip-172-31-29-172:~ — ssh -i ~/Downloads/demo_key_pair.pem hadoop@e
[hive> select distinct(category_code) from ext_table2019 ;
Query ID = hadoop_20201207142644_e25680d7-4799-4a0d-a304-a0a2cab20f71
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1607317819799_0020)

-----
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: 64.34 s
-----
OK

accessories.cosmetic_bag
stationery.cartrige
accessories.bag
appliances.environment.vacuum
furniture.living_room.chair
sport.diving
appliances.personal.hair_cutter
appliances.environment.air_conditioner
apparel.glove
furniture.bathroom.bath
furniture.living_room.cabinet
Time taken: 65.101 seconds, Fetched: 12 row(s)
hive> █
```

The distinct categories of the products are as follows :

- i) Accessories.cosmetic_bag
- ii) Stationary.cartrige
- 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

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

Answer : *select category_code ,count(product_id) as total_order from ext_table2019 group by category_code;*

```
abhishek — hadoop@ip-172-31-29-172:~ — ssh -i ~/Downloads/demo_key_pair.pem hadoop@ec2-52-2
[hive> select category_code ,count(product_id) as total_order from ext_table2019 group by category_code;
Query ID = hadoop_20201207142943_a2c23be4-d131-422a-9944-3f6ddff7695f
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1607317819799_0020)

-----
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: 64.37 s
-----
OK
      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: 65.087 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

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

Answer : *select brand, sum(price)as sales from ext_table2019 where brand is not null and event_type='purchase' group by brand order by sales desc limit 1 ;*

```
abhishek — hadoop@ip-172-31-29-172:~ — ssh -i ~/Downloads/demo_key_pair.pem hadoop@ec...
hive> select brand, sum(price)as sales from ext_table2019 where event_type='purchase' group by brand
order by sales desc limit 1 ;
Query ID = hadoop_20201207143849_48d684ed-db74-4093-9ca0-6ce85f350a78
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_1607317819799_0021)
```

	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: 17.24 s
OK
1094188.3000000063
Time taken: 26.035 seconds, Fetched: 1 row(s)
```

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

Answer : *select a.brand from(select brand, sum(price) as sales from ext_table2019 where brand <>' and month(event_time)=10 group by brand) as a inner join (select brand,sum(price) as sales from ext_table2019 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 ;*

```
abhishek — hadoop@ip-172-31-29-172:~ — ssh -i ~/Downloads/demo_key_pair.pem hadoo...
hive> select a.brand from(select brand, sum(price) as sales from ext_table2019 where brand <>'
and month(event_time)=10 group by brand) as a inner join (select brand,sum(price) as sales from
ext_table2019 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_20201207153141_43871cf0-f77b-4ed6-80e1-c8cdb91a526a
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1607317819799_0025)

-----
VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container  SUCCEEDED   7         7         0         0         0         0
Map 4 ..... container  SUCCEEDED   7         7         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: 136.18 s
-----
OK
airnails
artex
bpw.style
carmex
concept
Time taken: 137.781 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

- 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.

Answer : *select user_id, sum(price) as spend from ext_table2019 group by user_id order by spend limit 10 ;*

```
abhishek — hadoop@ip-172-31-29-172:~ — ssh -i ~/Downloads/demo_key_pair.pem hadoop@ec2-...
[hive> select user_id, sum(price) as spend from ext_table2019 group by user_id order by spend limit 10 ;]
Query ID = hadoop_20201207153727_bf5b0ad1-7915-4a36-8566-45e9fe4e6b8f
Total jobs = 1
Launching Job 1 out of 1
[Status: Running (Executing on YARN cluster with App id application_1607317819799_0025)]

-----
      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
Reducer 3 ..... container      SUCCEEDED      1           1           0           0           0           0
-----
VERTICES: 03/03 [=====>>] 100% ELAPSED TIME: 83.03 s
-----
OK
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: 83.742 seconds, Fetched: 10 row(s)
hive>
```