

HIVE CASE STUDY

DA Track

Done by
Mr. PARAMESH E

PROBLEM STATEMENT

With online sales gaining popularity, tech companies are exploring ways to improve their sales by analysing 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. Needless to say, the role of big data analysts is among the most sought-after job profiles of this decade. Therefore, as part of this assignment, we will be challenging you, 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 divided into the following parts:

1. Copying the data set into the HDFS:
2. Launch an EMR cluster that utilizes the Hive services,
3. Move the data from the S3 bucket into the HDFS
4. Creating the database and launching Hive queries on your EMR cluster:
5. Create the structure of your database,
6. Use optimized techniques to run your queries as efficiently as possible
7. Show the improvement of the performance after using optimization on any single query.
8. Running Hive queries to answering the questions given
9. Cleaning up
10. Drop database,
11. Terminate cluster.

Copying the data set into the HDFS:

Uploading data into the S3 buckets called casestudybucket1

The screenshot shows the AWS S3 console interface. At the top, there's a green banner indicating "Upload succeeded". Below this, the "Upload: status" section shows a summary of the upload. The destination is "s3://casestudybucket1". The upload was successful, with 2 files totaling 980.7 MB (100.00%). The "Files and folders" section shows two files: "2019-Nov.csv" (520.6 MB) and "2019-Oct.csv" (460.2 MB), both with a status of "Succeeded".

Name	Folder	Type	Size	Status	Error
2019-Nov.csv	-	text/csv	520.6 MB	Succeeded	-
2019-Oct.csv	-	text/csv	460.2 MB	Succeeded	-

Creating security groups

The screenshot shows the AWS VPC console interface. A green banner at the top indicates "Security group (sg-0f7b87dc45dbe607a | date240722) was created successfully". The "Details" section shows the following information:

Security group name	Security group ID	Description	VPC ID
date240722	sg-0f7b87dc45dbe607a	date240722	vpc-03a96d87df59c502c

Additional details shown include the Owner (520136779139), Inbound rules count (1 Permission entry), and Outbound rules count (1 Permission entry). The "Inbound rules" section shows 1 rule. A "Run Reachability Analyzer" button is visible.

Generating ppk file from putty key generator

The screenshot shows the AWS VPC Management Console with a security group named 'sg-0f7b87dc45d5be607a' created successfully. A PuTTY Key Generator window is open, showing the generated public key for an OpenSSH authorized_keys file. The key is an SSH-RSA key with a fingerprint of 'ssh-rsa 2048 SHA256:rbAKujKJLMH+wErtU+ZpUxNcJkzsNmdWzYIMzWqhM'. The key comment is 'imported-openssh-key'. The key passphrase is 'date240722'. The key is saved as a private key file.

Security group (sg-0f7b87dc45d5be607a | date240722) was created successfully

Details

Security group name: date240722

Owner: 520136779139

Inbound rules (1/1)

You can now check network connectivity with Reachability Analyzer

Run Reachability Analyzer

Generate public/private key pair

Load an existing private key file

Save the generated key

Save public key

Save private key

Parameters

Type of key to generate: ☒ RSA ☐ DSA ☐ ECDSA ☐ EdDSA ☐ SSH-1 (RSA)

Number of bits in a generated key: 2048

Launch an EMR cluster that utilizes the Hive services

The screenshot shows the AWS EMR console with a cluster named 'My HDFS cluster' in a 'Running' state. The cluster is configured with Hadoop distribution Amazon 2.8.5 and applications Hive 2.3.6, Pig 0.17.0, Hue 4.4.0, and Spark 2.4.4. The cluster is located in the us-east-1a availability zone, subnet subnet-0cd41f9828c749623, and has a master node m4.large and a core node m4.large. The cluster scaling is not enabled.

Amazon EMR

EMR Studio

EMR Serverless [New](#)

EMR on EC2

Clusters

Notebooks

Git repositories

Security configurations

Block public access

VPC subnets

Events

EMR on EKS

Virtual clusters

Help

What's new

Cluster: My HDFS cluster **Running** Running step

Summary Application user interfaces Monitoring Hardware Configurations Events Steps Bootstrap actions

Summary

ID: j-3NE7JRHD25LSJ

Creation date: 2022-07-31 09:33 (UTC+5:30)

Elapsed time: 12 minutes

After last step completes: Cluster waits

Termination protection: On [Change](#)

Tags: [View All / Edit](#)

Master public DNS: ec2-3-220-231-211.compute-1.amazonaws.com [Connect to the Master Node Using SSH](#)

Configuration details

Release label: emr-5.29.0

Hadoop distribution: Amazon 2.8.5

Applications: Hive 2.3.6, Pig 0.17.0, Hue 4.4.0, Spark 2.4.4

Log URI: s3://aws-logs-520136779139-us-east-1/elasticmapreduce/

EMRFS consistent view: Disabled

Custom AMI ID: --

Application user interfaces

Persistent user interfaces [Spark history server](#)

On-cluster user Not Enabled [Enable an SSH Connection interfaces](#)

Network and hardware

Availability zone: us-east-1a

Subnet ID: [subnet-0cd41f9828c749623](#)

Master: **Running** 1 m4.large

Core: **Running** 1 m4.large

Task: --

Cluster scaling: Not enabled

Security and access

Key name: date240722

Successfully Created EMR Cluster as well as Connected to Putty CLI Console Hadoop

```
hadoop@ip-10-0-14-76:~  
Using username "hadoop".  
Authenticating with public key "imported-openssh-key"  
Last login: Sun Jul 31 04:14:54 2022  
  
 _ | _ | )  
 _ | ( _ /   Amazon Linux AMI  
 _ | \ _ | _ |  
  
https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/  
64 package(s) needed for security, out of 93 available  
Run "sudo yum update" to apply all updates.  
  
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE MMMMMMMM MMMMMMMM RRRRRRRRRRRRRRRRR  
E:::EEEEEEEEEEEEEEEEEEEEEEEEEEEE M:::M M:::M R:::R  
EE:::EEEEEEEEEEEEEEEEEEEEEEEEEEEE M:::M M:::M R:::RRRRRR:::R  
E:::E EEEEE M:::M M:::M M:::M RR:::R R:::R  
E:::E M:::M M:::M M:::M R:::R R:::R  
E:::EEEEEEEEEEEE M:::M M:::M M:::M R:::RRRRRR:::R  
E:::EEEEEEEEEEEE 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:::EEEEEEEE:::E M:::M M:::M RR:::R R:::R  
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE MMMMMMMM MMMMMMMM RRRRRRR RRRRRR  
  
[hadoop@ip-10-0-14-76 ~]$ pwd  
/home/hadoop  
[hadoop@ip-10-0-14-76 ~]$
```

Making directory in Hadoop

```
[hadoop@ip-10-0-14-76 ~]$ pwd
/home/hadoop
[hadoop@ip-10-0-14-76 ~]$ hadoop fs -ls
[hadoop@ip-10-0-14-76 ~]$ ls
[hadoop@ip-10-0-14-76 ~]$ hadoop fs -ls /
Found 4 items
drwxr-xr-x   - hdfs  hadoop          0 2022-07-31 04:10 /apps
drwxrwxrwt   - hdfs  hadoop          0 2022-07-31 04:13 /tmp
drwxr-xr-x   - hdfs  hadoop          0 2022-07-31 04:10 /user
drwxr-xr-x   - hdfs  hadoop          0 2022-07-31 04:10 /var
[hadoop@ip-10-0-14-76 ~]$ hadoop fs -mkdir /hivecasestudy
[hadoop@ip-10-0-14-76 ~]$ hadoop fs -ls /
Found 5 items
drwxr-xr-x   - hdfs  hadoop          0 2022-07-31 04:10 /apps
drwxr-xr-x   - hadoop hadoop          0 2022-07-31 04:17 /hivecasestudy
drwxrwxrwt   - hdfs  hadoop          0 2022-07-31 04:13 /tmp
drwxr-xr-x   - hdfs  hadoop          0 2022-07-31 04:10 /user
drwxr-xr-x   - hdfs  hadoop          0 2022-07-31 04:10 /var
[hadoop@ip-10-0-14-76 ~]$
```

Moving datasets from the S3 bucket into the HDFS as follows respectively

All the bytes successfully copied to HDFS

```
[hadoop@ip-10-0-14-76 ~]$ hadoop distcp 's3://casestudybucket1/2019-Nov.csv' /hivecasestudy/2019-Nov.csv
```

```
Bytes Copied=545839412
Bytes Expected=545839412
Files Copied=1
[hadoop@ip-10-0-14-76 ~]$ hadoop distcp 's3://casestudybucket1/2019-Oct.csv' /hivecasestudy/2019-Oct.csv
```

```
Bytes Copied=482542278
Bytes Expected=482542278
Files Copied=1
[hadoop@ip-10-0-14-76 ~]$
```

Looking datasets at HDFS hivecasestudy folder

```
hadoop@ip-10-0-14-76:~
```

```
[hadoop@ip-10-0-14-76 ~]$ hadoop fs -ls /
Found 5 items
drwxr-xr-x - hdfs hadoop 0 2022-07-31 04:10 /apps
drwxr-xr-x - hdfs hadoop 0 2022-07-31 04:20 /hivecasestudy
drwxrwxrwt - hdfs hadoop 0 2022-07-31 04:13 /tmp
drwxr-xr-x - hdfs hadoop 0 2022-07-31 04:10 /user
drwxr-xr-x - hdfs hadoop 0 2022-07-31 04:10 /var
[hadoop@ip-10-0-14-76 ~]$ hadoop fs -ls /hivecasestudy/
Found 2 items
-rw-r--r-- 1 hadoop hadoop 545839412 2022-07-31 04:19 /hivecasestudy/2019-Nov.csv
-rw-r--r-- 1 hadoop hadoop 482542278 2022-07-31 04:20 /hivecasestudy/2019-Oct.csv
[hadoop@ip-10-0-14-76 ~]$
```

Launching Hive

```
[hadoop@ip-10-0-14-76 ~]$ hive
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
```

Launching Hive queries on EMR cluster

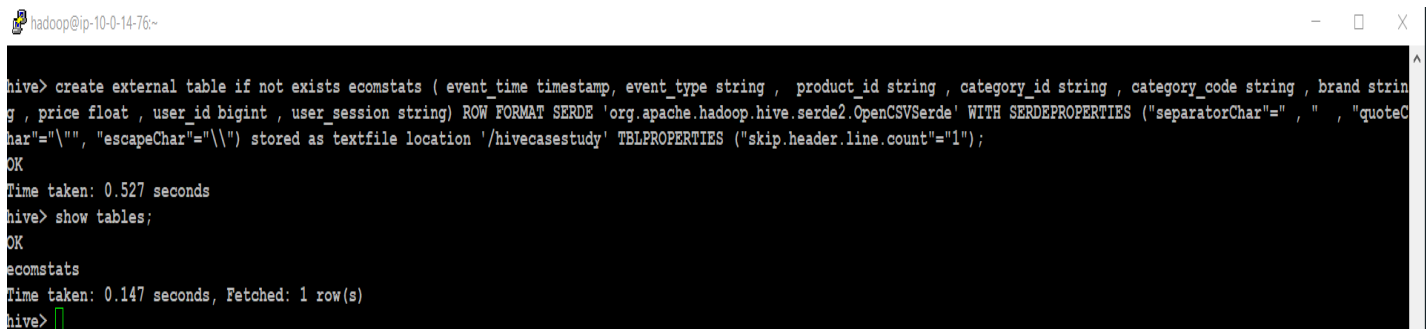
Cross-Checking Databases

```
hive> show databases;  
OK  
default  
Time taken: 1.081 seconds, Fetched: 1 row(s)
```

Creating ecommerce database and starting to use ecommerce db

```
hive> CREATE DATABASE IF NOT EXISTS ecommerce;  
OK  
Time taken: 0.577 seconds  
hive> show databases;  
OK  
default  
ecommerce  
Time taken: 0.032 seconds, Fetched: 2 row(s)  
hive> use ecommerce;  
OK  
Time taken: 0.094 seconds  
hive> █
```

Creating an external table from the raw data



```
hive> create external table if not exists ecomstats ( 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"="," , "quoteChar"="\\"", "escapeChar"="\\"") stored as textfile location '/hivecasestudy' TBLPROPERTIES ("skip.header.line.count"="1");  
OK  
Time taken: 0.527 seconds  
hive> show tables;  
OK  
ecomstats  
Time taken: 0.147 seconds, Fetched: 1 row(s)  
hive> █
```

QUERY:

```
create external table if not exists ecomstats ( 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"="," , "quoteChar"="\\"", "escapeChar"="\\"") stored as textfile location '/hivecasestudy' TBLPROPERTIES ("skip.header.line.count"="1");
```

Describing variables of ecomstats table

```
hive> desc ecomstats;
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: 0.134 seconds, Fetched: 9 row(s)
```

Loading the both datasets into the ecomstats table

```
hive> LOAD DATA INPATH '/hivecasestudy/2019-Nov.csv' into table ecomstats ;
Loading data to table ecommerce.ecomstats
OK
Time taken: 2.22 seconds
hive> LOAD DATA INPATH '/hivecasestudy/2019-Oct.csv' into table ecomstats ;
Loading data to table ecommerce.ecomstats
OK
Time taken: 0.846 seconds
hive> █
```

With the Hive tool enabling header True and

Looking the ecomstats table

```
hive> set hive.cli.print.header=true;
hive> select * from ecomstats limit 5;
OK
ecomstats.event_time  ecomstats.event_type  ecomstats.product_id  ecomstats.category_id  ecomstats.category_code  ecomstats.brand  ecomstats.price  ecomst
ecomstats.user_session
2019-11-01 00:00:02 UTC view      5802432 1487580009286598681          0.32  562076640      09fafd6c-6c99-46b1-834f-33527f4de241
2019-11-01 00:00:09 UTC cart      5844397 1487580006317032337          2.38  553329724      2067216c-31b5-455d-a1cc-af0575a34ffb
2019-11-01 00:00:10 UTC view      5837166 1783999064103190764          pnb   22.22  556138645      57ed222e-a54a-4907-9944-5a875c2d7f4f
2019-11-01 00:00:11 UTC cart      5876812 1487580010100293687          jessnail  3.16  564506666      186c1951-8052-4b37-adce-dd9644b1d5f7
2019-11-01 00:00:24 UTC remove_from_cart  5826182 1487580007483048900          3.33  553329724      2067216c-31b5-455d-a1cc-af0575a34ffb
Time taken: 0.212 seconds, Fetched: 5 row(s)
hive> █
```


From Dynamic Partitioning Tools Creating Optimized Hive table partitioned by Event type and Cluster by User id with 6 buckets

- Enabling Dynamic Partitioning tools

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

- Using optimized techniques Creating Optimized Table called dynamicstats and Confirming

```
hive> Create external table if not exists dynamicstats (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 (user_id) into 6 buckets ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' stored as textfile;
OK
Time taken: 0.118 seconds
hive> show tables;
OK
tab_name
dynamicstats
ecomstats
Time taken: 0.024 seconds, Fetched: 2 row(s)
hive>
```

QUERY:

Create external table if not exists dynamicstats (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 (user_id) into 6 buckets ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' stored as textfile;

- Describe Dynamicstats Variables

```
hive> desc dynamicstats;
OK
col_name      data_type      comment
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_type      string
Time taken: 0.109 seconds, Fetched: 14 row(s)
hive>
```

- Loading Data into the Dynamicstats table from ecomstats table

```
hive> insert into dynamicstats partition (event_type) select event_time , product_id , category_id , category_code , brand , price , user_id , user_session
, event_type from ecomstats ;
Query ID = hadoop_20220731044122_983a92c0-2985-448d-94dc-f9eb06c31303
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1659240713505_0004)
```

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

```
VERTICES: 02/02 [=====>>>] 100% ELAPSED TIME: 168.47 s
Loading data to table ecommerce.dynamicstats partition (event_type=null)
Loaded : 4/4 partitions.
Time taken to load dynamic partitions: 0.439 seconds
Time taken for adding to write entity : 0.002 seconds
OK
event_time      product_id      category_id      category_code      brand      price      user_id      user_session      event_type
Time taken: 170.682 seconds
hive>
```

- Looking at the table dynamicstats

```
hive> select * from dynamicstats limit 5;
OK
dynamicstats.event_time dynamicstats.product_id dynamicstats.category_id dynamicstats.category_code dynamicstats.brand dynamicstats.price d
dynamicstats.user_id dynamicstats.user_session dynamicstats.event_type
2019-10-11 08:11:33 UTC 5810136 1487580009445982239 irisk 1.43 486338323 01764e46-afbe-4de8-9044-77c379c518bf cart
2019-10-07 20:53:42 UTC 5846385 1487580008145748965 lovely 5.70 557752292 bf1270da-9f66-bcee-f336-10f51a280e65 cart
2019-10-07 20:53:45 UTC 5857007 1487580009496313889 runail 3.17 523616115 5b2ba38b-d66c-4127-bc23-275f06541525 cart
2019-10-07 20:54:35 UTC 5817702 1487580009496313889 0.63 523616115 5b2ba38b-d66c-4127-bc23-275f06541525 cart
2019-10-07 20:54:36 UTC 5817702 1487580009496313889 0.63 523616115 5b2ba38b-d66c-4127-bc23-275f06541525 cart
Time taken: 0.342 seconds, Fetched: 5 row(s)
```

- Looking at the partitions Created in dynamicstats table

```
hive> show partitions dynamicstats ;
OK
partition
event_type=cart
event_type=purchase
event_type=remove_from_cart
event_type=view
Time taken: 0.07 seconds, Fetched: 4 row(s)
hive>
```

- Looking at the Partitions Created in HDFS

```
[hadoop@ip-10-0-14-76 ~]$ hadoop fs -ls /user/hive/warehouse/ecommerce.db/dynamicstats ;
Found 4 items
drwxrwxrwt - hadoop hadoop 0 2022-07-31 04:44 /user/hive/warehouse/ecommerce.db/dynamicstats/event_type=cart
drwxrwxrwt - hadoop hadoop 0 2022-07-31 04:44 /user/hive/warehouse/ecommerce.db/dynamicstats/event_type=purchase
drwxrwxrwt - hadoop hadoop 0 2022-07-31 04:44 /user/hive/warehouse/ecommerce.db/dynamicstats/event_type=remove_from_cart
drwxrwxrwt - hadoop hadoop 0 2022-07-31 04:44 /user/hive/warehouse/ecommerce.db/dynamicstats/event_type=view
[hadoop@ip-10-0-14-76 ~]$
```

QUESTION AND ANSWERS

- ✚ We will be Comparing queries efficiency between Hive table ecomstats and Optimized Hive table dynamicstats.

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

Answer: The total revenue generated due to the purchases in the month of October is **1211538.4299997438**

Query:

Select avg(price)*count(event_type) as total_revenue from ecomstats where month(event_time)=10 and event_type = 'purchase';

- ecomstats Time taken = 66.76 s

```
hive> select avg(price)*count(event_type) as total_revenue from ecomstats where month(event_time)=10 and event_type = 'purchase';
Query ID = hadoop_20220731043633_afe48fed-b699-43c5-8068-af2d7adb2f1e
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_1659240713505_0004)

-----
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: 66.76 s
-----
OK
total_revenue
1211538.4299997438
Time taken: 79.952 seconds, Fetched: 1 row(s)
hive>
```

- dynamicstats Time taken = 25.53 s

```
hive> select avg(price)*count(event_type) as total_revenue from dynamicstats where month(event_time)=10 and event_type = 'purchase';
Query ID = hadoop_20220731044845_32150b54-4ae0-457c-b209-3f2fb6aba048
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1659240713505_0005)

-----
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: 25.53 s
-----
OK
total_revenue
1211538.429999948
Time taken: 29.239 seconds, Fetched: 1 row(s)
hive>
```

Looking at the results, the improvement of the performance after using optimization is much better so we will be using dynamicstats for answering the next questions.

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

QUERY:

```
select month(event_time) as month_purchases, count(product_id) as total_purchases
from ecomstats where event_type = 'purchase' group by month(event_time);
```

Answer: in November purchases is gone high as compared to October

Month_purchases	total_purchases
-----------------	-----------------

10	245624
----	--------

11	322417
----	--------

```
hive> select month(event_time) as month_purchases, count(product_id) as total_purchases
> from dynamicstats where event_type = 'purchase' group by month(event_time);
```

Query ID = hadoop_20220731045305_20e57aa8-a490-449b-b2ec-58c3d170a2f8

Total jobs = 1

Launching Job 1 out of 1

Status: Running (Executing on YARN cluster with App id application_1659240713505_0005)

	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: 26.11 s

OK

month_purchases	total_purchases
-----------------	-----------------

10	245624
----	--------

11	322417
----	--------

Time taken: 27.444 seconds, Fetched: 2 row(s)

hive> █

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

QUERY:

WITH monthly_sales as (select round (sum (case when date_format (event_time, 'MM')=11 then price else 0 end),2) as sale_nov, round (sum (case when date_format (event_time, 'MM') =10 then price else 0 end),2) as sale_oct from dynamicstats where event_type = 'purchase' and date_format (event_time, 'MM') in ('10', '11')) select sale_nov, sale_oct, (sale_nov - sale_oct) as change_in_revenue from monthly_sales;

ANWER: November month Generated Revenue, There is increase in revenue from October to November is 319478.47,

Sale_nov	sale_oct	change_in_revenue
1531016.9	1211538.43	319478.47

```
hadoop@ip-10-0-14-76:~$ hiveshell
hive> WITH monthly_sales as (select round (sum (case when date_format (event_time, 'MM')=10 then price else 0 end),2) as sale_oct, round (sum (case when date
format (event_time, 'MM') =11 then price else 0 end),2) as sale_nov from dynamicstats where event_type = 'purchase' and date_format (event_time, 'MM') in ('
10', '11')) select sale_nov, sale_oct, (sale_nov - sale_oct) as change_in_revenue from monthly_sales;
Query ID = hadoop_20220731045755_bac7c148-a8a0-47da-83e4-5472db4f0e41
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1659240713505_0005)

-----
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: 40.46 s
-----
OK
sale nov      sale oct      change in_revenue
1531016.9     1211538.43    319478.47
Time taken: 41.197 seconds, Fetched: 1 row(s)
hive>
```

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

QUERY:

```
select distinct split(category_code,'\\\.')[0] as product_category from dynamicstats where split(category_code,'\\\.')[0] IS NOT NULL;
```

ANSWER: distinct categories of products

Furniture

Appliances

Accessories

Apparel

Sport

Stationary

```
hive> select distinct split(category_code,'\\\.')[0] as product_category from dynamicstats where split(category_code,'\\\.')[0] IS NOT NULL;
Query ID = hadoop_20220731050158_29781378-05a5-4dc1-9aa1-4467e583ba48
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1659240713505_0005)

-----
      VERTICES      MODE      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container  SUCCEEDED    6         6         0         0         0         0
Reducer 2 ..... container  SUCCEEDED    5         5         0         0         0         0
-----
VERTICES: 02/02 [=====>>] 100% ELAPSED TIME: 75.86 s
-----
OK
product_category

furniture
appliances
accessories
apparel
sport
stationery
Time taken: 76.488 seconds, Fetched: 7 row(s)
hive> █
```

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

QUERY:

```
select split(category_code,'\\\.')[0] as product_category, count(product_id) as
total_products from dynamicstats where split(category_code,'\\\.')[0] IS NOT NULL group
by split(category_code,'\\\.')[0];
```

ANSWER: Appliances category having maximum number of products followed by stationary category

product_category	total_products
Furniture	23604
Appliances	61736
Accessories	12929
Apparel	18232
Sport	2
Stationary	26722

hadoop@ip-10-0-14-76:~

```
hive> select split(category_code,'\\\.')[0] as product_category, count(product_id) as total_products from dynamicstats
> where split(category_code,'\\\.')[0] IS NOT NULL group by split(category_code,'\\\.')[0];
Query ID = hadoop_20220731050445_76e553d9-370b-4efa-9b6c-5ae5c2c9a111
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1659240713505_0005)

-----
VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container  SUCCEEDED    6         6         0         0         0         0
Reducer 2 ..... container  SUCCEEDED    5         5         0         0         0         0
-----
VERTICES: 02/02 [=====>>>] 100%  ELAPSED TIME: 77.68 s
-----
OK
product_category    total_products
8594895
furniture          23604
appliances          61736
accessories         12929
apparel 18232
sport 2
stationery          26722
Time taken: 78.392 seconds, Fetched: 7 row(s)
hive>
```

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

QUERY:

select brand, round(sum(price),2) as Sales from dynamicstats where brand <>' and event_type = 'purchase' group by brand order by Sales desc limit 1;

ANSWER: Runail brand had the maximum sales 148297.94 in October and November combined

BRAND	SALES
Runail	148297.94

hadoop@ip-10-0-14-76:~

```
hive> select brand, round(sum(price),2) as Sales from dynamicstats
> where brand <>' and event_type = 'purchase' group by brand order by Sales desc limit 1;
Query ID = hadoop_20220731050741_a538c2ad-2d04-402c-90e8-8d718b81d843
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1659240713505_0005)
```

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

VERTICES: 03/03 [======>>>] 100% ELAPSED TIME: 23.00 s

```
OK
brand  sales
runail 148297.94
Time taken: 23.732 seconds, Fetched: 1 row(s)
hive> █
```


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

QUERY:

WITH monthly_sales as (select brand, round (sum (case when date_format (event_time, 'MM') = 10 then price else 0 end),2) as sale_oct, round (sum (case when date_format (event_time, 'MM') =11 then price else 0 end),2) as sale_nov from dynamicstats where event_type = 'purchase' and date_format (event_time, 'MM') in ('10', '11') group by brand) select brand, sale_oct, sale_nov, (sale_nov - sale_oct) as difference_in_sales from monthly_sales where (sale_nov - sale_oct) > 0 order by difference_in_sales desc;

ANSWER:

```
hadoop@ip-10-0-14-76:~$ hive> WITH monthly_sales as (select brand, round ( sum (case when date_format (event_time, 'MM') = 10 then price else 0 end),2) as sale_oct,
> round (sum (case when date_format (event_time, 'MM') =11 then price else 0 end),2) as sale_nov from dynamicstats
> where event_type = 'purchase' and date_format (event_time, 'MM') in ('10', '11') group by brand)
> select brand, sale_oct, sale_nov, (sale_nov - sale_oct) as difference_in_sales from monthly_sales
> where (sale_nov - sale_oct) > 0 order by difference_in_sales desc;
Query ID = hadoop_20220731051713_e5e250a5-722d-4381-baa8-bc5dd14c79fa
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1659240713505_0006)

-----
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: 43.03 s
-----
OK
brand  sale_oct  sale_nov  difference_in_sales
474679.06 619509.24 144830.18
grattol 35445.54 71472.71 36027.170000000006
uno 35302.03 51039.75 15737.720000000001
lianail 5892.84 16394.24 10501.400000000001
lingarden 23161.39 33566.21 10404.82
strong 29196.63 38671.27 9474.639999999996
jessnail 26287.84 33345.23 7057.390000000003
cosmoprofi 8322.81 14536.99 6214.18
polarus 6013.72 11371.93 5358.21
runail 71539.28 76758.66 5219.380000000005
freedecor 3421.78 7671.8 4250.02
staleks 8519.73 11875.61 3355.880000000001
bpw_style 11572.15 14837.44 3265.290000000001
lovely 8704.38 11939.06 3234.680000000003
marathon 7280.75 10273.1 2992.350000000004
haruyama 9390.69 12352.91 2962.2199999999993
yoko 8756.91 11707.88 2950.9699999999993
italwax 21940.24 24799.37 2859.1299999999974
benovy 409.62 3259.97 2850.35
kaypro 881.34 3268.7 2387.3599999999997
estel 21756.75 24142.67 2385.9199999999983
concept 11032.14 13380.4 2348.26
kapous 11927.16 14093.08 2165.92
f.o.x 6624.23 8577.28 1953.050000000001

f.o.x 6624.23 8577.28 1953.050000000001
masura 31266.08 33058.47 1792.3899999999994
milv 3904.94 5642.01 1737.0700000000002
beautix 10493.95 12222.95 1729.0
artex 2730.64 4327.25 1596.6100000000001
domix 10472.05 12009.17 1537.1200000000008
shik 3341.2 4839.72 1498.5200000000004
smart 4457.26 5902.14 1444.88
roubloff 3491.36 4913.77 1422.4100000000003
levrana 2243.56 3664.1 1420.54
oniq 8425.41 9841.65 1416.2399999999998
izink 45591.96 46946.04 1354.0800000000017
severina 4775.88 6120.48 1344.5999999999995
joico 705.52 2015.1 1309.58
zeitun 708.66 2009.63 1300.9700000000003
beauty-free 554.17 1782.86 1228.69
swarovski 1887.93 3043.16 1155.2299999999998
de.lux 1659.7 2775.51 1115.8100000000002
metzger 5373.45 6457.16 1083.71
markell 1768.75 2834.43 1065.6799999999998
sanoto 157.14 1209.68 1052.54
nagaraku 4369.74 5327.68 957.9400000000005
ecolab 262.85 1214.3 951.4499999999999
art-visage 2092.71 2997.8 905.0900000000001
levissime 2227.5 3085.31 857.81
missha 1293.83 2150.28 856.4500000000003
solomeya 1899.7 2685.8 786.1000000000001
rosi 3077.04 3841.56 764.52
refectocil 2716.18 3475.58 759.4000000000001
kaaral 4412.43 5086.07 673.6399999999994
kosmekka 1181.44 1813.37 631.9299999999998
kinetics 6334.25 6945.26 611.0100000000002
browxenna 14331.37 14916.73 585.3599999999998
airnails 5118.9 5691.52 572.6200000000008
uskusi 5142.27 5680.31 548.04
coifin 903.0 1428.49 525.49
s.care 412.68 913.07 500.39000000000004
limoni 1308.9 1796.6 487.6999999999998
matrix 3243.25 3726.74 483.4899999999998
gehwol 1089.07 1557.68 468.6100000000001
greyymy 29.21 489.49 460.28000000000003
bioaqua 942.89 1398.12 455.2299999999999
farmavita 837.37 1291.97 454.6
sophin 1067.86 1515.52 447.6600000000001
yu-z 271.41 673.71 402.3
```

```
hadoop@ip-10-0-14-76:~$
yu-r 271.41 673.71 402.3
kiss 421.55 817.33 395.780000000000003
naomi 0.0 389.0 389.0
lador 2083.61 2471.53 387.920000000000001
ellips 245.85 606.04 360.189999999999994
jas 3318.96 3657.43 338.46999999999998
lowence 242.84 567.75 324.90999999999997
nitriile 847.28 1162.68 315.400000000000001
shary 871.96 1176.49 304.53
kims 330.04 632.04 301.999999999999994
happyfons 801.92 1091.59 289.669999999999996
kocostar 310.85 594.93 284.07999999999999
insight 1443.7 1721.96 278.26
candy 534.96 799.38 264.419999999999996
bluesky 10307.24 10565.53 258.290000000000009
beauugreen 511.51 768.35 256.840000000000003
protokeratin 201.25 456.79 255.540000000000002
trind 298.07 542.96 244.890000000000004
entity 479.71 719.26 239.55
skinlite 651.94 890.45 238.51
provoc 827.99 1063.82 235.829999999999993
fedua 52.38 263.81 211.43
ecocraft 41.16 241.95 200.79
keen 236.35 435.62 199.27
mane 66.79 260.26 193.469999999999997
freshbubble 318.7 502.34 183.64
matreshka 0.0 182.67 182.67
chi 358.94 538.61 179.670000000000002
cristalinas 427.63 584.95 157.320000000000005
farmona 1692.46 1843.43 150.970000000000003
latinoil 249.52 384.59 135.069999999999996
miskin 158.04 293.07 135.03
elizavecca 70.53 204.3 133.77
nefertiti 233.52 366.64 133.119999999999998
finnish 98.38 230.38 132.0
igrobeauty 513.66 645.07 131.410000000000008
dizao 819.13 945.51 126.38
osmo 645.58 762.31 116.729999999999999
batiste 772.4 874.17 101.769999999999998
carmex 145.08 243.36 98.28
eos 54.34 152.61 98.270000000000001
depilflax 2707.07 2803.78 96.710000000000004
enjoy 41.35 136.57 95.22
kerasys 430.91 525.2 94.290000000000002
aura 83.95 177.51 93.559999999999999
```

```
hadoop@ip-10-0-14-76:~$
laboratorium 246.5 312.52 66.019999999999998
inm 288.02 351.21 63.19
dewal 0.0 61.29 61.29
marutaka-foot 49.22 109.33 60.11
kares 0.0 59.45 59.45
profhenna 679.23 736.85 57.620000000000005
koelcia 55.5 112.75 57.25
balbcare 155.33 212.38 57.049999999999998
elskin 251.09 307.65 56.5599999999999974
foamie 35.04 80.49 45.449999999999996
ladykin 125.65 170.57 44.919999999999999
likato 296.06 340.97 44.9100000000000025
mavala 409.04 446.32 37.279999999999997
vilenta 197.6 231.21 33.6100000000000014
beautyblender 78.74 109.41 30.67
biore 60.65 90.31 29.660000000000004
orly 902.38 931.09 28.7100000000000036
estelare 444.81 471.87 27.060000000000002
profepil 93.36 118.02 24.659999999999997
blixz 38.95 63.4 24.449999999999996
binacil 0.0 24.26 24.26
godefroy 401.22 425.12 23.899999999999977
glysolid 69.73 91.59 21.86
veraclara 50.11 71.21 21.099999999999994
juno 0.0 21.08 21.08
kamill 63.01 81.49 18.479999999999997
treaclemoon 163.37 181.49 18.120000000000005
supertan 50.37 66.51 16.140000000000008
barbie 0.0 12.39 12.39
deoproce 316.84 329.17 12.330000000000041
rasyan 18.8 28.94 10.14
fly 17.14 27.17 10.030000000000001
tertio 236.16 245.8 9.6400000000000015
jaguar 1102.11 1110.65 8.5400000000000191
soleo 204.2 212.53 8.330000000000013
neoleor 43.41 51.7 8.290000000000006
moyou 5.71 10.28 4.569999999999999
bodyton 1376.34 1380.64 4.3000000000000182
skinity 8.88 12.44 3.5599999999999987
helloganic 0.0 3.1 3.1
grace 100.92 102.61 1.6899999999999977
cosima 20.23 20.93 0.6999999999999993
ovale 2.54 3.1 0.56
Time taken: 43.67 seconds, Fetched: 161 row(s)
hive>
```

```
hadoop@ip-10-0-14-76:~$
kerasys 430.91 525.2 94.290000000000002
aura 83.95 177.51 93.559999999999999
Olazan 101.37 194.01 92.639999999999999
koelif 422.73 507.29 84.56
nirvel 163.04 234.33 71.290000000000002
konad 739.83 810.67 70.839999999999992
egomania 77.47 146.04 68.57
cutrin 299.37 367.62 68.25
laboratorium 246.5 312.52 66.019999999999998
inm 288.02 351.21 63.19
dewal 0.0 61.29 61.29
marutaka-foot 49.22 109.33 60.11
kares 0.0 59.45 59.45
profhenna 679.23 736.85 57.620000000000005
koelcia 55.5 112.75 57.25
balbcare 155.33 212.38 57.049999999999998
elskin 251.09 307.65 56.5599999999999974
foamie 35.04 80.49 45.449999999999996
ladykin 125.65 170.57 44.919999999999999
likato 296.06 340.97 44.9100000000000025
mavala 409.04 446.32 37.279999999999997
vilenta 197.6 231.21 33.6100000000000014
beautyblender 78.74 109.41 30.67
biore 60.65 90.31 29.660000000000004
orly 902.38 931.09 28.7100000000000036
estelare 444.81 471.87 27.060000000000002
profepil 93.36 118.02 24.659999999999997
blixz 38.95 63.4 24.449999999999996
binacil 0.0 24.26 24.26
godefroy 401.22 425.12 23.899999999999977
glysolid 69.73 91.59 21.86
veraclara 50.11 71.21 21.099999999999994
juno 0.0 21.08 21.08
kamill 63.01 81.49 18.479999999999997
treaclemoon 163.37 181.49 18.120000000000005
supertan 50.37 66.51 16.140000000000008
barbie 0.0 12.39 12.39
deoproce 316.84 329.17 12.330000000000041
rasyan 18.8 28.94 10.14
fly 17.14 27.17 10.030000000000001
tertio 236.16 245.8 9.6400000000000015
jaguar 1102.11 1110.65 8.5400000000000191
soleo 204.2 212.53 8.330000000000013
neoleor 43.41 51.7 8.290000000000006
moyou 5.71 10.28 4.569999999999999
```

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?

QUERY:

```
select user_id, round(sum(price),2) as total_purchase from dynamicstats where  
event_type = 'purchase' group by user_id order by total_purchase desc limit 10;
```

ANSWER:

```
hive> select user_id, round(sum(price),2) as total_purchase from dynamicstats  
> where event_type = 'purchase' group by user_id order by total_purchase desc limit 10;  
Query ID = hadoop_20220731051549_1c96b3ac-4d5f-4ee8-8c2a-77bbaaf705f0
```

Total jobs = 1

Launching Job 1 out of 1

Status: Running (Executing on YARN cluster with App id application_1659240713505_0006)

	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: 28.31 s

OK

```
user_id total_purchase
```

```
557790271 2715.87
```

```
150318419 1645.97
```

```
562167663 1352.85
```

```
531900924 1329.45
```

```
557850743 1295.48
```

```
522130011 1185.39
```

```
561592095 1109.7
```

```
431950134 1097.59
```

```
566576008 1056.36
```

```
521347209 1040.91
```

Time taken: 29.081 seconds, Fetched: 10 row(s)

hive> █

DROPPING TABLES:

- Dropping dynamicstats table

```
hive> drop table dynamicstats;  
OK  
Time taken: 0.17 seconds
```

- Dropping ecomstats table

```
hive> drop table ecomstats;  
OK  
Time taken: 0.109 seconds
```

DROPPING DATABASE

```
hive> drop database ecommerce;  
OK  
Time taken: 0.069 seconds  
hive> █
```

TERMINATING EMR CLUSTER

The screenshot shows the AWS Management Console for an EMR cluster. The cluster is in a 'Terminated' state. The left sidebar shows the navigation menu with options like EMR Studio, EMR Serverless, EMR on EC2, Clusters, Notebooks, Security configurations, Block public access, VPC subnets, Events, EMR on EKS, and Virtual clusters. The main content area displays the cluster details for 'My HDFS cluster'. A red box highlights a message: 'Auto-termination is not available for this account when using this release of EMR.' The cluster details are organized into sections: Summary, Configuration details, Application user interfaces, Network and hardware, and Security and access. The Summary section shows the cluster ID, creation and end dates, and the release label. The Configuration details section shows the Hadoop distribution, applications, and log URI. The Application user interfaces section shows the persistent user interfaces. The Network and hardware section shows the availability zone, subnet ID, and instance types. The Security and access section shows the key name.

Cluster: My HDFS cluster Terminated Terminated by user request

Summary

ID: j-3NE7JRH25LSJ
Creation date: 2022-07-31 09:33 (UTC+5:30)
End date: 2022-07-31 10:59 (UTC+5:30)
Elapsed time: 1 hour, 26 minutes
After last step completes: Cluster waits
Termination protection: Off
Tags: --
Master public DNS: ec2-3-220-231-211.compute-1.amazonaws.com
Connect to the Master Node Using SSH

Configuration details

Release label: emr-5.29.0
Hadoop distribution: Amazon 2.8.5
Applications: Hive 2.3.6, Pig 0.17.0, Hue 4.4.0, Spark 2.4.4
Log URI: s3://aws-logs-520136779139-us-east-1/elasticmapreduce/
EMRFS consistent view: Disabled
Custom AMI ID: --

Application user interfaces

Persistent user interfaces: Spark history server
On-cluster user -- interfaces:

Network and hardware

Availability zone: us-east-1a
Subnet ID: subnet-0cd41f9828c749623
Master: Terminated 1 m4.large
Core: Terminated 1 m4.large
Task: --
Cluster scaling: Not enabled

Security and access

Key name: date240722

CONFIRMING:

In Instances running showing ZERO

The screenshot shows the AWS Management Console for the EC2 Global view. The left sidebar shows the navigation menu with options like New EC2 Experience, EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images, AMIs, AMI Catalog, and Elastic Block Store. The main content area displays the EC2 Global view. A message states: 'You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:'. The resources are listed in a table: Instances (running) 0, Dedicated Hosts 0, Elastic IPs 0, Instances 2, Key pairs 1, Load balancers 1 (API Error), Placement groups 0, Security groups 5, Snapshots 0, and Volumes 2. A message states: 'Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more'. The Launch instance section shows a 'Launch instance' button and a 'Migrate a server' button. The Service health section shows the status of the EC2 service in the US East (N. Virginia) region, which is 'operating normally'.

Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Filter resources by tag(s)

Instances (running)	0	Dedicated Hosts	0	Elastic IPs	0
Instances	2	Key pairs	1	Load balancers	1 API Error
Placement groups	0	Security groups	5	Snapshots	0
Volumes	2				

Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Launch instance

Migrate a server

Service health

AWS Health Dashboard

Region

US East (N. Virginia)

Status

This service is operating normally

Account attributes

Supported platforms

- VPC

Default VPC

vpc-0d29461f1b48c9a57

Settings

EBS encryption

Zones

EC2 Serial Console

Default credit specification

Console experiments

Explore AWS

Save up to 90% on EC2 with Spot Instances

Optimize price-performance by combining EC2 purchase options in a single EC2 ASG. Learn more

Get Up to 40% Better Price Performance

T4g instances deliver the best price performance for burstable general purpose workloads in Amazon EC2. Learn more

CONCLUSION:

- ❖ 1211538.4299997438 revenue generated due to purchases made in October
- ❖ the total sum of purchases per month in OCTOBER is 245624 and november 322417
- ❖ 319478.47 revenue generated due to purchases from October to November.
- ❖ Appliances category having 61736 highest products and followed by stationary category with 26722 number of products available under category.
- ❖ Runail is the brand had the maximum sales in October and November combined

THANK YOU

CASE STUDY DONE BY
PARAMESH E