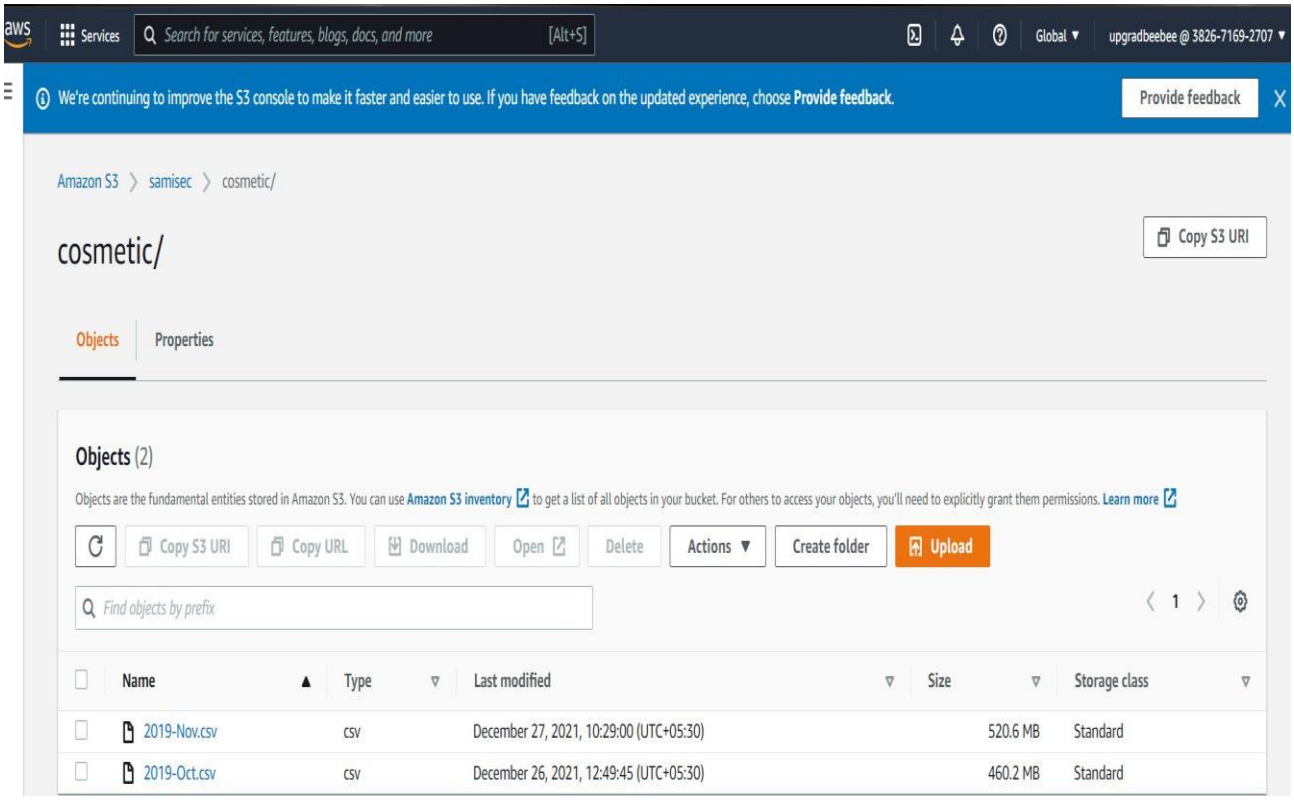
HIVE Case Study

Importing Data into HDFS

First loading data into s3:



Query to laod data into hdfs:

The given two datasets are transferred to HDFS using S3 bucket

creating a dirctory :

hadoop fs -mkdir /hive_assignment

To list the directories:

hadoop fs -ls /

```
[hadoop@ip-172-31-65-31 ~]$ hadoop fs -mkdir /hive_assignment
[hadoop@ip-172-31-65-31 ~]$ hadoop fs -ls /
Found 5 items
drwxr-xr-x   - hdfs   hadoop          0 2022-01-02 12:57 /apps
drwxr-xr-x   - hadoop hadoop          0 2022-01-02 13:09 /hive_assignment
drwxrwxrwt   - hdfs   hadoop          0 2022-01-02 12:59 /tmp
drwxr-xr-x   - hdfs   hadoop          0 2022-01-02 12:57 /user
drwxr-xr-x   - hdfs   hadoop          0 2022-01-02 12:57 /var
```

Loading data to hdfs from s3:

```
[hadoop@ip-172-31-65-31 ~]$ hadoop distcp s3://samisec/cosmetic/2019-Oct.csv /hi
ve_assignment/2019-Oct.csv
22/01/02 13:10:34 INFO tools.DistCp: Input Options: DistCpOptions{atomicCommit=f
alse, syncFolder=false, deleteMissing=false, ignoreFailures=false, overwrite=fal
se, skipCRC=false, blocking=true, numListstatusThreads=0, maxMaps=20, mapBandwid
th=100, sslConfigurationFile='null', copyStrategy='uniformsize', preserveStatus=
[], preserveRawXattrs=false, atomicWorkPath=null, logPath=null, sourceFileListin
g=null, sourcePaths=[s3://samisec/cosmetic/2019-Oct.csv], targetPath=/hive_assig
nment/2019-Oct.csv, targetPathExists=false, filtersFile='null'}
22/01/02 13:10:34 INFO client.RMProxy: Connecting to ResourceManager at ip-172-3
1-65-31.ec2.internal/172.31.65.31:8032
22/01/02 13:10:38 INFO tools.SimpleCopyListing: Paths (files+dirs) cnt = 1; dirC
nt = 0
```

```
    DistCp Counters
            Bytes Copied=482542278
            Bytes Expected=482542278
            Files Copied=1
```

```
[hadoop@ip-172-31-72-60 ~]$ hadoop distcp s3://samisec/cosmetic/2019-Nov.csv /hi
ve_assignment/2019-Nov.csv
22/01/03 06:43:37 INFO tools.DistCp: Input Options: DistCpOptions{atomicCommit=f
alse, syncFolder=false, deleteMissing=false, ignoreFailures=false, overwrite=fal
se, skipCRC=false, blocking=true, numListstatusThreads=0, maxMaps=20, mapBandwid
th=100, sslConfigurationFile='null', copyStrategy='uniformsize', preserveStatus=
[], preserveRawXattrs=false, atomicWorkPath=null, logPath=null, sourceFileListin
g=null, sourcePaths=[s3://samisec/cosmetic/2019-Nov.csv], targetPath=/hive_assig
nment/2019-Nov.csv, targetPathExists=false, filtersFile='null'}
22/01/03 06:43:37 INFO client.RMProxy: Connecting to ResourceManager at ip-172-3
1-72-60.ec2.internal/172.31.72.60:8032
22/01/03 06:43:43 INFO tools.SimpleCopyListing: Paths (files+dirs) cnt = 1; dirC
nt = 0
22/01/03 06:43:43 INFO tools.SimpleCopyListing: Build file listing completed.
22/01/03 06:43:43 INFO Configuration.deprecation: io.sort.mb is deprecated. Inst
ead, use mapreduce.task.io.sort.mb
22/01/03 06:43:43 INFO Configuration.deprecation: io.sort.factor is deprecated.
Instead, use mapreduce.task.io.sort.factor
22/01/03 06:43:43 INFO tools.DistCp: Number of paths in the copy list: 1
22/01/03 06:43:43 INFO tools.DistCp: Number of paths in the copy list: 1
22/01/03 06:43:43 INFO client.RMProxy: Connecting to ResourceManager at ip-172-3
1-72-60.ec2.internal/172.31.72.60:8032
22/01/03 06:43:44 INFO mapreduce.JobSubmitter: number of splits:1
22/01/03 06:43:44 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_16
41191022556_0002
22/01/03 06:43:45 INFO impl.YarnClientImpl: Submitted application application_16
41191022556_0002
```

```
        DistCp Counters
                Bytes Copied=545839412
                Bytes Expected=545839412
                Files Copied=1
```

view the data in hdfs by following commands:

```
[hadoop@ip-172-31-72-60 ~]$ hadoop fs -cat /hive_assignment/2019-Oct.csv |head
event_time,event_type,product_id,category_id,category_code,brand,price,user_id,u
ser_session
2019-10-01 00:00:00 UTC,cart,5773203,1487580005134238553,,runail,2.62,463240011,
26dd6e6e-4dac-4778-8d2c-92e149dab885
2019-10-01 00:00:03 UTC,cart,5773353,1487580005134238553,,runail,2.62,463240011,
26dd6e6e-4dac-4778-8d2c-92e149dab885
2019-10-01 00:00:07 UTC,cart,5881589,2151191071051219817,,lovely,13.48,429681830
,49e8d843-adf3-428b-a2c3-fe8bc6a307c9
2019-10-01 00:00:07 UTC,cart,5723490,1487580005134238553,,runail,2.62,463240011,
26dd6e6e-4dac-4778-8d2c-92e149dab885
2019-10-01 00:00:15 UTC,cart,5881449,1487580013522845895,,lovely,0.56,429681830,
49e8d843-adf3-428b-a2c3-fe8bc6a307c9
2019-10-01 00:00:16 UTC,cart,5857269,1487580005134238553,,runail,2.62,430174032,
73dea1e7-664e-43f4-8b30-d32b9d5af04f
2019-10-01 00:00:19 UTC,cart,5739055,1487580008246412266,,kapous,4.75,377667011,
81326ac6-daa4-4f0a-b488-fd0956a78733
2019-10-01 00:00:24 UTC,cart,5825598,1487580009445982239,,,0.56,467916806,2f5b55
46-b8cb-9ee7-7ecd-84276f8ef486
2019-10-01 00:00:25 UTC,cart,5698989,1487580006317032337,,,1.27,385985999,d30965
e8-1101-44ab-b45d-cc1bb9fae694
```

```
[hadoop@ip-172-31-72-60 ~]$ hadoop fs -cat /hive_assignment/2019-Nov.csv | head
event_time,event_type,product_id,category_id,category_code,brand,price,user_id,u
ser_session
2019-11-01 00:00:02 UTC,view,5802432,1487580009286598681,,,0.32,562076640,09fafd
6c-6c99-46b1-834f-33527f4de241
2019-11-01 00:00:09 UTC,cart,5844397,1487580006317032337,,,2.38,553329724,206721
6c-31b5-455d-a1cc-af0575a34ffb
2019-11-01 00:00:10 UTC,view,5837166,1783999064103190764,,pnb,22.22,556138645,57
ed222e-a54a-4907-9944-5a875c2d7f4f
2019-11-01 00:00:11 UTC,cart,5876812,1487580010100293687,,jessnail,3.16,56450666
6,186c1951-8052-4b37-adce-dd9644b1d5f7
2019-11-01 00:00:24 UTC,remove_from_cart,5826182,1487580007483048900,,,3.33,5533
29724,2067216c-31b5-455d-a1cc-af0575a34ffb
2019-11-01 00:00:24 UTC,remove_from_cart,5826182,1487580007483048900,,,3.33,5533
29724,2067216c-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,,,3.49,514649199,432a4e
95-375c-4b40-bd36-0fc039e77580
2019-11-01 00:00:34 UTC,remove_from_cart,5870838,1487580007675986893,,milv,0.79,
429913900,2f0bff3c-252f-4fe6-afcd-5d8a6a92839a
cat: Unable to write to output stream.
```

Launching Hive:

```
[hadoop@ip-172-31-72-60 ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.
properties Async: false
```

creating databases in hive

```
hive> show databases;
OK
default
Time taken: 1.078 seconds, Fetched: 1 row(s)
hive> create database if not exists case_study ;
OK
Time taken: 0.36 seconds
hive> describe database case_study;
OK
case_study              hdfs://ip-172-31-72-60.ec2.internal:8020/user/hive/wareh
ouse/case_study.db      hadoop  USER
Time taken: 0.06 seconds, Fetched: 1 row(s)
hive> use case_study;
OK
Time taken: 0.03 seconds
```

Creating table from raw data:

```
hive> create external table if not exists oct_nov
    > (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 '/hive_assignment'
    > TBLPROPERTIES("skip.header.line.count"="1");
OK
Time taken: 0.884 seconds
hive> describe oct_nov;
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: 0.159 seconds, Fetched: 9 row(s)
```

Loading the data and checking the data.

```
hive> load data inpath '/hive_assignment/2019-Oct.csv' into table oct_nov;
Loading data to table case_study.oct_nov
OK
Time taken: 2.032 seconds
hive> load data inpath '/hive_assignment/2019-Nov.csv' into table oct_nov;
Loading data to table case_study.oct_nov
OK
Time taken: 0.71 seconds
hive> select * from oct_nov limit 3;
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     2
2.22    556138645       57ed222e-a54a-4907-9944-5a875c2d7f4f
Time taken: 2.268 seconds, Fetched: 3 row(s)
```

Creating a table for analysis:

```
hive> create external table if not exists oct_nov_data
    > (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 delimited fields terminated by ','
    > lines terminated by '\n' stored as textfile;
OK
Time taken: 0.161 seconds
hive> describe oct_nov_data;
OK
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
Time taken: 0.052 seconds, Fetched: 9 row(s)
```

Inserting the data:

```
hive> insert into oct_nov_data
    > select
    > cast(replace(event_time,'UTC','') as timestamp),
    > event_type,
    > product_id,
    > category_id,
    > category_code,
    > brand,
    > cast(price as float),
    > cast(user_id as bigint),
    > user_session
    > from oct_nov;
Query ID = hadoop_20220103065449_954b5a26-9a7f-4e22-bb40-221cadbf5a10
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_1641191022556
_0004)

Map 1: 0/2
Map 1: 0/2
Map 1: 0(+2)/2
Map 1: 0(+2)/2
```

```
Map 1: 0(+2)/2
Map 1: 0(+2)/2
Map 1: 0(+2)/2
Map 1: 0(+2)/2
Map 1: 0(+2)/2
Map 1: 0(+2)/2
Map 1: 1(+1)/2
Map 1: 1(+1)/2
Map 1: 1(+1)/2
Map 1: 1(+1)/2
Map 1: 2/2
Loading data to table case_study.oct_nov_data
OK
Time taken: 135.376 seconds
```

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

Query:

```
select sum(price)
from oct_nov_data where month(event_time)=10 and event_type = 'purchase';
```

```
hive> select sum(price)
    > from oct_nov_data where month(event_time)=10 and event_type = 'purchase';
Query ID = hadoop_20220103065751_357f8efc-7db2-404b-8ae6-5d61e2ddb29a
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1641191022556
_0004)

Map 1: 0/7      Reducer 2: 0/1
Map 1: 0/7      Reducer 2: 0/1
Map 1: 0/7      Reducer 2: 0/1
Map 1: 0(+1)/7  Reducer 2: 0/1
Map 1: 0(+2)/7  Reducer 2: 0/1
Map 1: 0(+3)/7  Reducer 2: 0/1
Map 1: 0(+3)/7  Reducer 2: 0/1
Map 1: 0(+3)/7  Reducer 2: 0/1
Map 1: 0(+3)/7  Reducer 2: 0/1
Map 1: 0(+3)/7  Reducer 2: 0/1
Map 1: 0(+3)/7  Reducer 2: 0/1
```

```
Map 1: 0(+3)/7  Reducer 2: 0/1
Map 1: 0(+3)/7  Reducer 2: 0/1
Map 1: 0(+3)/7  Reducer 2: 0/1
Map 1: 1(+2)/7  Reducer 2: 0/1
Map 1: 1(+3)/7  Reducer 2: 0/1
Map 1: 2(+2)/7  Reducer 2: 0/1
Map 1: 2(+3)/7  Reducer 2: 0/1
Map 1: 3(+3)/7  Reducer 2: 0/1
Map 1: 3(+3)/7  Reducer 2: 0/1
Map 1: 4(+2)/7  Reducer 2: 0/1
Map 1: 4(+3)/7  Reducer 2: 0/1
Map 1: 5(+2)/7  Reducer 2: 0/1
Map 1: 5(+2)/7  Reducer 2: 0(+1)/1
Map 1: 6(+1)/7  Reducer 2: 0(+1)/1
Map 1: 7/7      Reducer 2: 0(+1)/1
Map 1: 7/7      Reducer 2: 1/1
OK
1211538.4295325726
Time taken: 38.302 seconds, Fetched: 1 row(s)
```

Time taken: 38.3 seconds

After the base table is created, we need to optimize the table for quick query result through partioning and bucketing.

our optimised table name is oct_nov_data. now we enable dynamic partioning and create a partioned table with buckets.

```
hive> set hive.exec.dynamic.partition=true ;
hive> set hive.exec.dynamic.partition.mode= nonstrict;
hive> create external table if not exists oct_nov_part
    > (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)
    > partitioned by (year int, month int)
    > clustered by (category_id) into 4 buckets
    > row format delimited fields terminated by ','
    > lines terminated by '\n' stored as textfile;
OK
Time taken: 0.084 seconds
hive> show tables;
OK
oct_nov
oct_nov_data
oct_nov_part
Time taken: 0.075 seconds, Fetched: 3 row(s)
hive> describe oct_nov_part;
OK
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
year                    int
month                   int

# Partition Information
# col_name              data_type               comment

year                    int
month                   int
Time taken: 0.124 seconds, Fetched: 17 row(s)
```

Inserting the data in oct_nov_part table:

```
hive> insert into oct_nov_part partition (year, month)
    > select
    > cast(replace(event_time,'UTC','') as timestamp),
    > event_type,
    > product_id,
    > category_id,
    > category_code,
    > brand,
    > price,
    > user_id,
    > user_session,
    > year(cast(replace(event_time,'UTC','') as timestamp)),
    > month(cast(replace(event_time,'UTC','') as timestamp))
    > from oct_nov;
Query ID = hadoop_20220103070204_4c10cbfc-779d-4fb0-bec4-3f2eb5d230a9
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1641191022556
_0004)

Map 1: 0/2      Reducer 2: 0/5
Map 1: 0/2      Reducer 2: 0/5
Map 1: 0(+2)/2  Reducer 2: 0/5
```

```
Map 1: 2/2      Reducer 2: 1(+3)/5
Map 1: 2/2      Reducer 2: 1(+3)/5
Map 1: 2/2      Reducer 2: 1(+3)/5
Map 1: 2/2      Reducer 2: 1(+3)/5
Map 1: 2/2      Reducer 2: 2(+3)/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: 3(+2)/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: 4(+1)/5
Map 1: 2/2      Reducer 2: 5/5
Loading data to table case_study.oct_nov_part partition (year=null, month=null)


        Time taken to load dynamic partitions: 0.378 seconds
        Time taken for adding to write entity : 0.002 seconds
OK
Time taken: 227.312 seconds
```

We will now run the same query for this optimized table

```
hive> select sum(price) as total_revenue
    > from oct_nov_part where month(event_time)=10 and event_type = 'purchase';
Query ID = hadoop_20220103070617_f721739d-58c4-4362-b88a-2d7cf170ccfd
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1641191022556
_0004)

Map 1: 0/8      Reducer 2: 0/1
Map 1: 0/8      Reducer 2: 0/1
Map 1: 0/8      Reducer 2: 0/1
Map 1: 0(+2)/8  Reducer 2: 0/1
Map 1: 0(+3)/8  Reducer 2: 0/1
Map 1: 0(+3)/8  Reducer 2: 0/1
Map 1: 0(+3)/8  Reducer 2: 0/1
Map 1: 0(+3)/8  Reducer 2: 0/1
Map 1: 0(+3)/8  Reducer 2: 0/1
Map 1: 0(+3)/8  Reducer 2: 0/1
Map 1: 1(+3)/8  Reducer 2: 0/1
Map 1: 2(+3)/8  Reducer 2: 0/1
Map 1: 3(+3)/8  Reducer 2: 0/1
Map 1: 3(+3)/8  Reducer 2: 0/1
Map 1: 4(+2)/8  Reducer 2: 0/1
Map 1: 5(+1)/8  Reducer 2: 0/1
Map 1: 5(+2)/8  Reducer 2: 0/1
Map 1: 5(+3)/8  Reducer 2: 0/1
Map 1: 6(+2)/8  Reducer 2: 0/1
Map 1: 6(+2)/8  Reducer 2: 0(+1)/1
Map 1: 7(+1)/8  Reducer 2: 0(+1)/1
Map 1: 8/8      Reducer 2: 0(+1)/1
Map 1: 8/8      Reducer 2: 1/1
OK
1211538.4295325726
Time taken: 37.468 seconds, Fetched: 1 row(s)
```

Query: select sum(price) as total_revenue from oct_nov_part where month(event_time)=10 and event_type = 'purchase';

Time taken :37.4 seconds

Enabling second approach dynamic partitioning and creating a partitioned tables with buckets.

```
hive> set hive.exec.dynamic.partition=true ;
hive> set hive.exec.dynamic.partition.mode= nonstrict;
hive> create external table if not exists oct_nov_part2
    > (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_id) into 5 buckets
    > ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
    > stored as textfile;
OK
Time taken: 0.085 seconds
hive> describe oct_nov_part2;
OK
event_time              string                  from deserializer
product_id              string                  from deserializer
category_id             string                  from deserializer
category_code           string                  from deserializer
brand                   string                  from deserializer
price                   string                  from deserializer
user_id                 string                  from deserializer
user_session            string                  from deserializer
event_type              string

# Partition Information
# col_name              data_type               comment

event_type              string
Time taken: 0.075 seconds, Fetched: 14 row(s)
```

Inserting the data

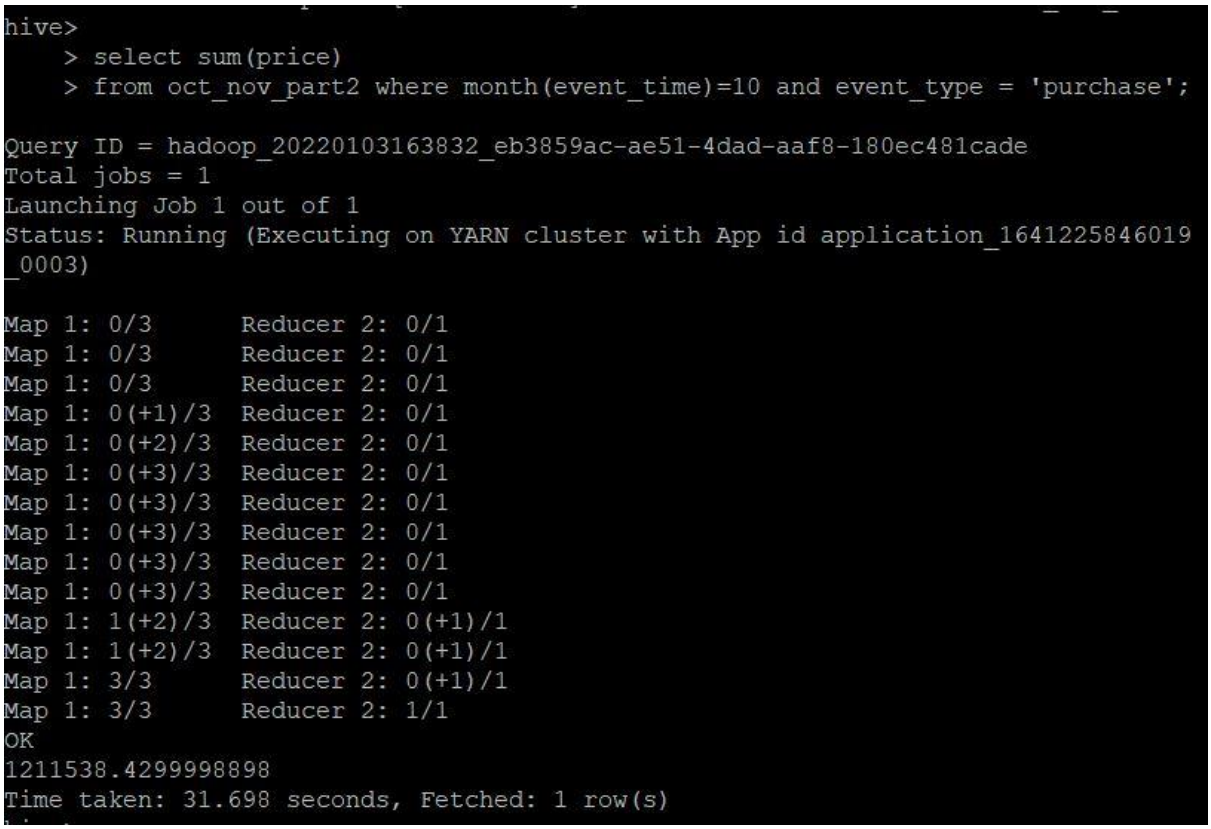
```
hive> insert into oct_nov_part2 partition (event_type)
    > select
    > cast(replace(event_time,'UTC','') as timestamp),
    > product_id,
    > category_id,
    > category_code,
    > brand,
    > cast(price as float),
    > cast(user_id as bigint),
    > user_session,
    > event_type
    > from oct_nov;
Query ID = hadoop_20220103070937_f04506a7-a585-4f54-9bea-4fea55e08210
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1641191022556
_0004)

Map 1: 0/2      Reducer 2: 0/5
Map 1: 0/2      Reducer 2: 0/5
Map 1: 0(+2)/2  Reducer 2: 0/5
Map 1: 0(+2)/2  Reducer 2: 0/5
Map 1: 2/2      Reducer 2: 3(+2)/5
Map 1: 2/2      Reducer 2: 3(+2)/5
Map 1: 2/2      Reducer 2: 3(+2)/5
Map 1: 2/2      Reducer 2: 3(+2)/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 case_study.oct_nov_part2 partition (event_type=null)


         Time taken to load dynamic partitions: 0.666 seconds
         Time taken for adding to write entity : 0.002 seconds
OK
Time taken: 217.709 seconds
```

Query: select sum(price)

from oct_nov_part2 where month(event_time)=10 and event_type = 'purchase';



Time taken is 31 seconds.

So here we find that by Partition by over event_type and clustering by ‘Category_id’ we get the most optimized query.

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

Query:

select sum(price)

from oct_nov_part2 where month(event_time) = 10 and event_type = 'purchase';

```
hive>
    > select sum(price)
    > from oct_nov_part2 where month(event_time)=10 and event_type = 'purchase';

Query ID = hadoop_20220103163832_eb3859ac-ae51-4dad-aaf8-180ec481cade
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1641225846019
_0003)

Map 1: 0/3      Reducer 2: 0/1
Map 1: 0/3      Reducer 2: 0/1
Map 1: 0/3      Reducer 2: 0/1
Map 1: 0(+1)/3  Reducer 2: 0/1
Map 1: 0(+2)/3  Reducer 2: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1
Map 1: 1(+2)/3  Reducer 2: 0(+1)/1
Map 1: 1(+2)/3  Reducer 2: 0(+1)/1
Map 1: 3/3      Reducer 2: 0(+1)/1
Map 1: 3/3      Reducer 2: 1/1
OK
1211538.4299998898
Time taken: 31.698 seconds, Fetched: 1 row(s)
```

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

select month(event_time)as event,

sum(price) from oct_nov_part2

where year(event_time)=2019 and event_type='purchase'

group by month(event_time);

```
hive> select month(event_time)as event,
    > sum(price) from oct_nov_part2
    > where year(event_time)=2019 and event_type='purchase'
    > group by month(event_time);
Query ID = hadoop_20220103071704_61d677bc-3896-4313-9875-ecd58448140e
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1641191022556
_0004)

Map 1: 0/3      Reducer 2: 0/1
Map 1: 0/3      Reducer 2: 0/1
Map 1: 0/3      Reducer 2: 0/1
Map 1: 0(+2)/3  Reducer 2: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1
Map 1: 1(+2)/3  Reducer 2: 0(+1)/1
Map 1: 2(+1)/3  Reducer 2: 0(+1)/1
Map 1: 3/3      Reducer 2: 0(+1)/1
Map 1: 3/3      Reducer 2: 1/1
OK
10      1211538.4299998898
11      1531016.9
Time taken: 27.712 seconds, Fetched: 2 row(s)
```

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

Query:

select sum(case when month(event_time)=10 then price else -1* price end) as change_in_revenue

from oct_nov_part2 where month(event_time) in (10,11) and event_type = 'purchase';

```
hive> select sum(case when month(event_time)=10 then price else -1* price end) a
s change_in_revenue
    > from oct_nov_part2 where month(event_time) in (10,11) and event_type = 'pu
rchase';
Query ID = hadoop_20220103071857_60cf6ff7-299a-4538-a77e-a4c8c4c41c23
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1641191022556
_0004)

Map 1: 0/3      Reducer 2: 0/1
Map 1: 0/3      Reducer 2: 0/1
Map 1: 0/3      Reducer 2: 0/1
Map 1: 0(+2)/3  Reducer 2: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1
Map 1: 1(+2)/3  Reducer 2: 0(+1)/1
Map 1: 2(+1)/3  Reducer 2: 0(+1)/1
Map 1: 3/3      Reducer 2: 0(+1)/1
Map 1: 3/3      Reducer 2: 1/1
OK
-319478.47000012523
Time taken: 28.22 seconds, Fetched: 1 row(s)
```

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

Query:

select distinct split(category_code,'\\.')[0] as cat from oct_nov_part2 where split(category_code,'\\.')[0] <> '';

```
hive> select distinct split(category_code,'\\.')[0] as cat from oct_nov_part2 wh
ere split(category_code,'\\.')[0] <> '';
Query ID = hadoop_20220103072039_2b9d077d-9fa5-4a2b-b529-157460c8128d
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1641191022556
_0004)

Map 1: 0/6      Reducer 2: 0/5
Map 1: 0/6      Reducer 2: 0/5
Map 1: 0/6      Reducer 2: 0/5
Map 1: 0(+2)/6  Reducer 2: 0/5
Map 1: 0(+3)/6  Reducer 2: 0/5
Map 1: 0(+3)/6  Reducer 2: 0/5
Map 1: 0(+3)/6  Reducer 2: 0/5
Map 1: 0(+3)/6  Reducer 2: 0/5
Map 1: 0(+3)/6  Reducer 2: 0/5
Map 1: 0(+3)/6  Reducer 2: 0/5
Map 1: 0(+3)/6  Reducer 2: 0/5
Map 1: 0(+3)/6  Reducer 2: 0/5
Map 1: 0(+3)/6  Reducer 2: 0/5
Map 1: 1(+3)/6  Reducer 2: 0/5
```

```
Map 1: 4(+2)/6  Reducer 2: 0(+1)/5
Map 1: 5(+1)/6  Reducer 2: 0(+1)/5
Map 1: 5(+1)/6  Reducer 2: 0(+2)/5
Map 1: 6/6      Reducer 2: 0(+3)/5
Map 1: 6/6      Reducer 2: 1(+2)/5
Map 1: 6/6      Reducer 2: 2(+3)/5
Map 1: 6/6      Reducer 2: 3(+2)/5
Map 1: 6/6      Reducer 2: 5/5
OK
furniture
appliances
accessories
apparel
sport
stationery
Time taken: 69.656 seconds, Fetched: 6 row(s)
```

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

Query:

```
select  split(category_code,'\\.')[0] as cat, count(product_id) as no_of_products
from oct_nov_part2 where split(category_code,'\\.')[0] <> ''
group by split(category_code,'\\.')[0]
order by no_of_products desc;
```

```
hive> select  split(category_code,'\\.')[0] as cat, count(product_id) as no_of_p
roducts
    > from oct_nov_part2 where split(category_code,'\\.')[0] <> ''
    > group by split(category_code,'\\.')[0]
    > order by no_of_products desc;
Query ID = hadoop_20220103072524_ec1dc62b-bbdf-4c45-b3fc-53b8c6d67ca8
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1641191022556
_0004)

Map 1: 0/6      Reducer 2: 0/5  Reducer 3: 0/1
Map 1: 0/6      Reducer 2: 0/5  Reducer 3: 0/1
Map 1: 0/6      Reducer 2: 0/5  Reducer 3: 0/1
Map 1: 0(+1)/6  Reducer 2: 0/5  Reducer 3: 0/1
Map 1: 0(+2)/6  Reducer 2: 0/5  Reducer 3: 0/1
Map 1: 0(+3)/6  Reducer 2: 0/5  Reducer 3: 0/1
Map 1: 0(+3)/6  Reducer 2: 0/5  Reducer 3: 0/1
Map 1: 0(+3)/6  Reducer 2: 0/5  Reducer 3: 0/1
Map 1: 0(+3)/6  Reducer 2: 0/5  Reducer 3: 0/1
Map 1: 0(+3)/6  Reducer 2: 0/5  Reducer 3: 0/1
```

```
Map 1: 2(+3)/6  Reducer 2: 0/5  Reducer 3: 0/1
Map 1: 3(+3)/6  Reducer 2: 0/5  Reducer 3: 0/1
Map 1: 3(+3)/6  Reducer 2: 0/5  Reducer 3: 0/1
Map 1: 3(+3)/6  Reducer 2: 0/5  Reducer 3: 0/1
Map 1: 3(+3)/6  Reducer 2: 0/5  Reducer 3: 0/1
Map 1: 3(+3)/6  Reducer 2: 0/5  Reducer 3: 0/1
Map 1: 3(+3)/6  Reducer 2: 0/5  Reducer 3: 0/1
Map 1: 4(+2)/6  Reducer 2: 0(+1)/5      Reducer 3: 0/1
Map 1: 5(+1)/6  Reducer 2: 0(+1)/5      Reducer 3: 0/1
Map 1: 5(+1)/6  Reducer 2: 0(+2)/5      Reducer 3: 0/1
Map 1: 6/6      Reducer 2: 0(+3)/5      Reducer 3: 0/1
Map 1: 6/6      Reducer 2: 2(+3)/5      Reducer 3: 0/1
Map 1: 6/6      Reducer 2: 3(+2)/5      Reducer 3: 0(+1)/1
Map 1: 6/6      Reducer 2: 5/5  Reducer 3: 0(+1)/1
Map 1: 6/6      Reducer 2: 5/5  Reducer 3: 1/1
OK
appliances      61736
stationery      26722
furniture       23604
apparel 18232
accessories     12929
sport   2
Time taken: 69.222 seconds, Fetched: 6 row(s)
```

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

Query:

```
select brand, round(sum(price),2) as sales
from oct_nov_part2
where brand <>'' and event_type ='purchase'
group by brand
order by sales desc
limit 1;
```

```
hive>
    > select brand, round(sum(price),2) as sales
    > from oct_nov_part2
    > where brand <>'' and event_type ='purchase'
    > group by brand
    > order by sales desc
    > limit 1;
Query ID = hadoop_20220103072652_fc256c6f-8b16-4e04-a77d-78f46c24b383
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1641191022556
_0004)

Map 1: 0/3      Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0/3      Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0/3      Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0(+2)/3  Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 1(+2)/3  Reducer 2: 0(+1)/1      Reducer 3: 0/1
Map 1: 2(+1)/3  Reducer 2: 0(+1)/1      Reducer 3: 0/1
Map 1: 3/3      Reducer 2: 0(+1)/1      Reducer 3: 0/1
Map 1: 3/3      Reducer 2: 1/1  Reducer 3: 0(+1)/1
Map 1: 3/3      Reducer 2: 1/1  Reducer 3: 1/1
OK
runail  148297.94
Time taken: 24.488 seconds, Fetched: 1 row(s)
```

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

Query:

```
with sale_difference as(select brand,
sum(case when date_format(event_time,'MM')=10 then price else 0 end) oct_month,
sum(case when date_format(event_time,'MM')=11 then price else 0 end) nov_month
from oct_nov_part2
where event_type = 'purchase' and date_format(event_time, 'MM') in ('10','11')
group by brand )
select brand, oct_month, nov_month, (nov_month-oct_month) as sale_diff
from sale_difference
where(nov_month-oct_month)>0
order by sale_diff;
```

```
hive> with sale_difference as(select brand,
    > sum(case when date_format(event_time,'MM')=10 then price else 0 end) oct_m
onth,
    > sum(case when date_format(event_time,'MM')=11 then price else 0 end) nov_m
onth
    > from oct_nov_part2
    > where event_type = 'purchase' and date_format(event_time, 'MM') in ('10','
11')
    > group by brand )
    > select brand, oct_month, nov_month, (nov_month-oct_month) as sale_diff
    > from sale_difference
    > where(nov_month-oct_month)>0
    > order by sale_diff;
Query ID = hadoop_20220103072805_49dd14a4-52d1-4bc1-882f-2f175f400244
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1641191022556
_0004)

Map 1: 0/3      Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0/3      Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0/3      Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0(+1)/3  Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0(+2)/3  Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 1(+2)/3  Reducer 2: 0(+1)/1      Reducer 3: 0/1
Map 1: 2(+1)/3  Reducer 2: 0(+1)/1      Reducer 3: 0/1
Map 1: 3/3      Reducer 2: 0(+1)/1      Reducer 3: 0/1
Map 1: 3/3      Reducer 2: 1/1  Reducer 3: 0(+1)/1
Map 1: 3/3      Reducer 2: 1/1  Reducer 3: 1/1
OK
ovale   2.54    3.1     0.56
cosima  20.23   20.930000000000003      0.7000000000000028
grace   100.92000000000002      102.61000000000004      1.6900000000000261
helloganic      0.0     3.1     3.1
skinity 8.88    12.440000000000001      3.5600000000000005
```

8.Your company wants to reward the top 10 users of its website with a Golden Customer plan. Write a query to generate a list of top 10 users who spend the most.

select user_id, sum(price) as spender

from oct_nov_part2

where event_type ='purchase'

group by user_id

order by spender desc

limit 10;

```
hive> select user_id, sum(price) as spender
    > from oct_nov_part2
    > where event_type ='purchase'
    > group by user_id
    > order by spender desc
    > limit 10;
Query ID = hadoop_20220103072907_5f2893fc-c183-4835-ab95-7ef9360fc0d7
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1641191022556
_0004)

Map 1: 0/3      Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0/3      Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0/3      Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0(+2)/3  Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 0(+3)/3  Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 1(+2)/3  Reducer 2: 0/1  Reducer 3: 0/1
Map 1: 1(+2)/3  Reducer 2: 0(+1)/1      Reducer 3: 0/1
Map 1: 2(+1)/3  Reducer 2: 0(+1)/1      Reducer 3: 0/1
Map 1: 3/3      Reducer 2: 0(+1)/1      Reducer 3: 0/1
Map 1: 3/3      Reducer 2: 1/1  Reducer 3: 0(+1)/1
Map 1: 3/3      Reducer 2: 1/1  Reducer 3: 1/1
OK
557790271       2715.869999999995
150318419       1645.97
562167663       1352.85
531900924       1329.4499999999998
557850743       1295.48
522130011       1185.3899999999999
561592095       1109.7000000000003
431950134       1097.5899999999997
566576008       1056.3600000000006
521347209       1040.91
Time taken: 25.608 seconds, Fetched: 10 row(s)
```

```
neoleor 43.41   51.7    8.290000000000006
soleo   204.20000000000027      212.53000000000014      8.32999999999987
jaguar  1102.1100000000006      1110.6499999999999      8.539999999999281
tertio  236.16000000000008      245.80000000000015      9.640000000000072
fly     17.14   27.17   10.030000000000001
rasyan  18.799999999999997      28.939999999999998      10.14
deoproce        316.84  329.16999999999996      12.329999999999984
barbie  0.0     12.39   12.39
supertan        50.37000000000002       66.51000000000002       16.14
treaclemoon     163.37000000000003      181.49  18.119999999999976
kamill  63.00999999999999       81.49000000000001       18.480000000000018
juno    0.0     21.08   21.08
veraclara       50.11000000000001       71.21000000000001       21.1
glysolid        69.72999999999999       91.58999999999997       21.859999999999985
godefroy        401.2200000000003       425.1200000000003       23.899999999999977
binacil 0.0     24.259999999999998      24.259999999999998
blixz   38.949999999999996      63.4    24.450000000000003
profepil        93.36000000000001       118.02000000000001      24.659999999999997
estelare        444.81  471.8700000000003       27.060000000000286
orly    902.38  931.0899999999999       28.709999999999923
biore   60.650000000000006      90.31   29.659999999999997
beautyblender   78.74000000000001       109.41  30.669999999999987
vilenta 197.59999999999994      231.20999999999992      33.609999999999985
mavala  409.04  446.32000000000005      37.28000000000003
likato  296.06  340.96999999999997      44.90999999999997
ladykin 125.64999999999998      170.57  44.920000000000016
foamie  35.04   80.49   45.449999999999996
elskin  251.09000000000015      307.6500000000005       56.56000000000034
balbcare        155.32999999999996      212.37999999999985      57.049999999999997
koelcia 55.5    112.75  57.25
profhenna       679.2299999999997       736.8499999999998       57.620000000000012
kares   0.0     59.449999999999996      59.449999999999996
marutaka-foot   49.22   109.33  60.11
dewal   0.0     61.29   61.29
inm     288.0199999999998       351.21000000000004      63.190000000000225
laboratorium    246.50000000000003      312.52  66.01999999999995
cutrin  299.37000000000006      367.61999999999995      68.24999999999989
egomania        77.47   146.04000000000002      68.57000000000002
konad   739.8300000000004       810.6700000000008       70.84000000000037
nirvel  163.04000000000002      234.33  71.28999999999999
koelf   422.72999999999985      507.28999999999985      84.56
plazan  101.36999999999999      194.01  92.64
aura    83.95   177.50999999999996      93.55999999999996
kerasys 430.9100000000001       525.2   94.28999999999996
```

```
beautix 10493.949999999986      12222.949999999997      1729.000000000011
milv    3904.939999999983       5642.009999999976       1737.069999999993
masura  31266.07999999823       33058.469999998706      1792.390000000476
f.o.x   6624.23 8577.28000000001        1953.0500000000102
kapous  11927.160000000113      14093.080000000078      2165.9199999999655
concept 11032.139999999974      13380.39999999994       2348.2599999999657
estel   21756.750000000084      24142.67000000007       2385.9199999999873
kaypro  881.34  3268.7000000000003       2387.36
benovy  409.6199999999999       3259.969999999992       2850.349999999992
italwax 21940.23999999973       24799.369999999766      2859.1300000000374
yoko    8756.909999999994       11707.879999999965      2950.9699999999702
haruyama        9390.690000000014       12352.90999999999       2962.2199999999985
marathon        7280.749999999997       10273.099999999986      2992.3499999999885
lovely  8704.37999999999        11939.060000000029      3234.6800000000385
bpw.style       11572.150000001808      14837.440000002425      3265.2900000006175
staleks 8519.730000000023       11875.609999999999      3355.8799999999756
freedecor       3421.7799999999706      7671.800000000216       4250.020000000245
runail  71539.27999999619       76758.65999999736       5219.380000001169
polarus 6013.7200000000075      11371.930000000013      5358.2100000000055
cosmoprofi      8322.809999999996       14536.989999999958      6214.179999999962
jessnail        26287.84000000013       33345.23000000008       7057.389999999952
strong  29196.62999999994       38671.26999999994       9474.64
ingarden        23161.39000000044       33566.2099999995        10404.819999999057
lianail 5892.83999999998        16394.240000000194      10501.400000000214
uno     35302.03000000014       51039.749999998894      15737.719999998757
grattol 35445.54000000078       71472.7099999995        36027.16999999872
        474679.05999999656      619509.2399999899       144830.1799999933
Time taken: 29.027 seconds, Fetched: 161 row(s)
```