PROJECT REPORT

CLASS PROJECT PART – 1

INTRODUCTION AND PROBLEM DESCRIPTION

This project focuses on using the analytical skills by using big data technologies like AWS S3, AWS EMR – Hive and HDFS and AWS Athena. We will be using the Amazon reviews dataset available in S3. Our dataset will be in parquet format to improve our processing and is partitioned by product category. We begin our analysis from 2005. We choose a few categories from the list available and exclude multiple reviews by customers and only choose the most recent reviews. Based on the results, we will attempt to answer simple data exploratory questions along with trend analysis of our metrics. We will also attempt to correlate multiple product categories and answer questions based on the results.

We aim to perform analysis on the amazon reviews dataset. We will attempt to analyse trends on these metrics and try to correlate them and use different window functions and analytical aggregate functions to demonstrate the concepts of percentiles and moving average. We will also visualize some of our findings to provide clarity on the results obtained.

We begin by provisioning the EMR cluster and then copying the amazon reviews to EMR's HDFS. We then create an external table in Hive by pointing it to the HDFS folder. We then perform query operation on this table.

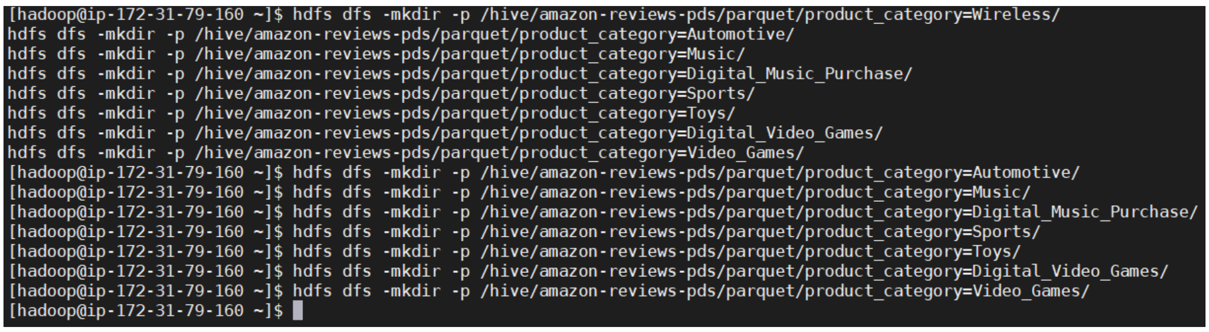
TECHNICAL SCRIPTS, SQL QUERIES AND EXPLANATION WITH VISUALIZATIONS

The steps we will perform to run hive queries on the external table are as follows:

1. Create directories in HDFS for our product categories

   Query:

   ```
   `hdfs dfs -mkdir -p /hive/amazon-reviews-pds/parquet/product_category=Wireless/`
   `hdfs dfs -mkdir -p /hive/amazon-reviews-pds/parquet/product_category=Automotive/`
   `hdfs dfs -mkdir -p /hive/amazon-reviews-pds/parquet/product_category=Music/`
   `hdfs dfs -mkdir -p /hive/amazon-reviews-pds/parquet/product_category=Digital_Music_Purchase/`
   `hdfs dfs -mkdir -p /hive/amazon-reviews-pds/parquet/product_category=Sports/`
   `hdfs dfs -mkdir -p /hive/amazon-reviews-pds/parquet/product_category=Toys/`
   `hdfs dfs -mkdir -p /hive/amazon-reviews-pds/parquet/product_category=Digital_Video_Games/`
   `hdfs dfs -mkdir -p /hive/amazon-reviews-pds/parquet/product_category=Video_Games/`
   ```

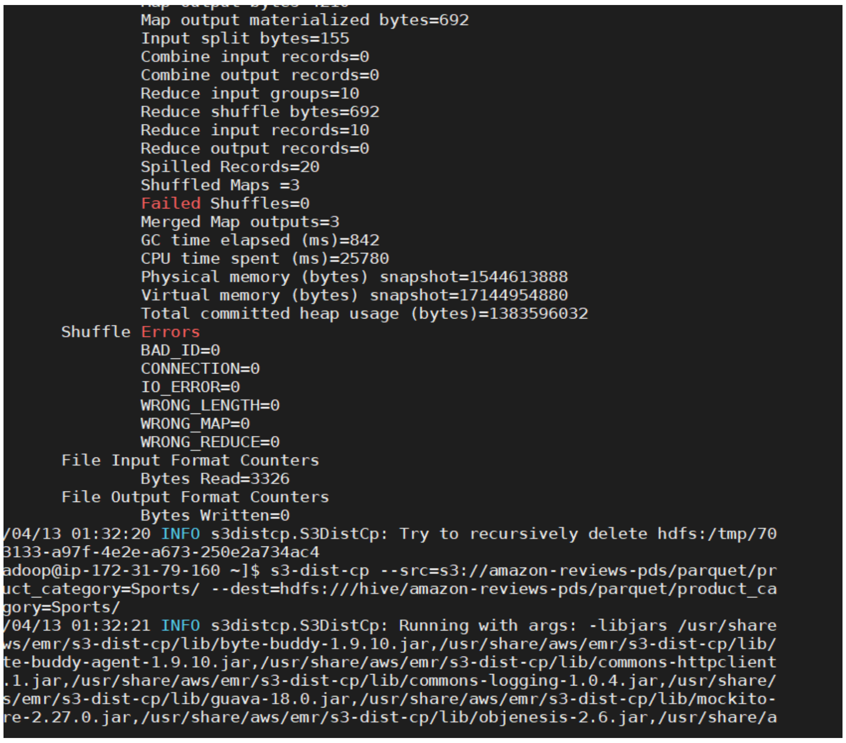


```
[hadoop@ip-172-31-79-160 ~]$ hdfs dfs -mkdir -p /hive/amazon-reviews-pds/parquet/product_category=Wireless/
hdfs dfs -mkdir -p /hive/amazon-reviews-pds/parquet/product_category=Automotive/
hdfs dfs -mkdir -p /hive/amazon-reviews-pds/parquet/product_category=Music/
hdfs dfs -mkdir -p /hive/amazon-reviews-pds/parquet/product_category=Digital_Music_Purchase/
hdfs dfs -mkdir -p /hive/amazon-reviews-pds/parquet/product_category=Sports/
hdfs dfs -mkdir -p /hive/amazon-reviews-pds/parquet/product_category=Toys/
hdfs dfs -mkdir -p /hive/amazon-reviews-pds/parquet/product_category=Digital_Video_Games/
hdfs dfs -mkdir -p /hive/amazon-reviews-pds/parquet/product_category=Video_Games/
[hadoop@ip-172-31-79-160 ~]$ hdfs dfs -mkdir -p /hive/amazon-reviews-pds/parquet/product_category=Automotive/
[hadoop@ip-172-31-79-160 ~]$ hdfs dfs -mkdir -p /hive/amazon-reviews-pds/parquet/product_category=Music/
[hadoop@ip-172-31-79-160 ~]$ hdfs dfs -mkdir -p /hive/amazon-reviews-pds/parquet/product_category=Digital_Music_Purchase/
[hadoop@ip-172-31-79-160 ~]$ hdfs dfs -mkdir -p /hive/amazon-reviews-pds/parquet/product_category=Sports/
[hadoop@ip-172-31-79-160 ~]$ hdfs dfs -mkdir -p /hive/amazon-reviews-pds/parquet/product_category=Toys/
[hadoop@ip-172-31-79-160 ~]$ hdfs dfs -mkdir -p /hive/amazon-reviews-pds/parquet/product_category=Digital_Video_Games/
[hadoop@ip-172-31-79-160 ~]$ hdfs dfs -mkdir -p /hive/amazon-reviews-pds/parquet/product_category=Video_Games/
[hadoop@ip-172-31-79-160 ~]$ 
```

2. Copy data from S3 to these directories

   Query:

   `s3-dist-cp --src=s3://amazon-reviews-pds/parquet/product_category=Wireless/ --dest=hdfs:///hive/amazon-reviews-pds/parquet/product_category=Wireless/`
   `s3-dist-cp --src=s3://amazon-reviews-pds/parquet/product_category=Automotive/ --dest=hdfs:///hive/amazon-reviews-pds/parquet/product_category=Automotive/`
   `s3-dist-cp --src=s3://amazon-reviews-pds/parquet/product_category=Music/ --dest=hdfs:///hive/amazon-reviews-pds/parquet/product_category=Music/`
   `s3-dist-cp --src=s3://amazon-reviews-pds/parquet/product_category=Digital_Music_Purchase/ --dest=hdfs:///hive/amazon-reviews-pds/parquet/product_category=Digital_Music_Purchase/`
   `s3-dist-cp --src=s3://amazon-reviews-pds/parquet/product_category=Sports/ --dest=hdfs:///hive/amazon-reviews-pds/parquet/product_category=Sports/`
   `s3-dist-cp --src=s3://amazon-reviews-pds/parquet/product_category=Toys/ --dest=hdfs:///hive/amazon-reviews-pds/parquet/product_category=Toys/`
   `s3-dist-cp --src=s3://amazon-reviews-pds/parquet/product_category=Digital_Video_Games/ --dest=hdfs:///hive/amazon-reviews-pds/parquet/product_category=Digital_Video_Games/`
   `s3-dist-cp --src=s3://amazon-reviews-pds/parquet/product_category=Video_Games/ --dest=hdfs:///hive/amazon-reviews-pds/parquet/product_category=Video_Games/`

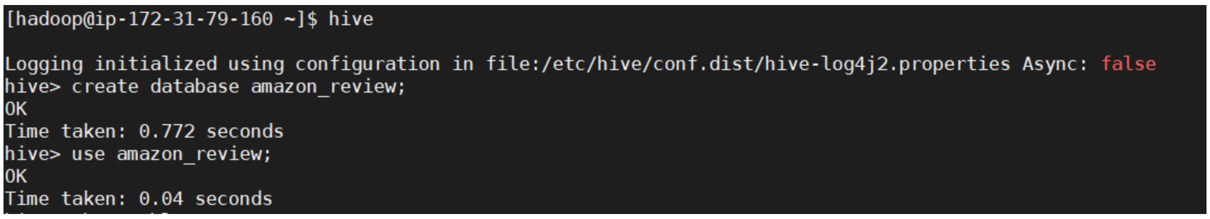


```
                Map output materialized bytes=692
                Input split bytes=155
                Combine input records=0
                Combine output records=0
                Reduce input groups=10
                Reduce shuffle bytes=692
                Reduce input records=10
                Reduce output records=0
                Spilled Records=20
                Shuffled Maps =3
                Failed Shuffles=0
                Merged Map outputs=3
                GC time elapsed (ms)=842
                CPU time spent (ms)=25780
                Physical memory (bytes) snapshot=1544613888
                Virtual memory (bytes) snapshot=17144954880
                Total committed heap usage (bytes)=1383596032
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=3326
        File Output Format Counters
                Bytes Written=0
/04/13 01:32:20 INFO s3distcp.S3DistCp: Try to recursively delete hdfs:/tmp/70
3133-a97f-4e2e-a673-250e2a734ac4
adoop@ip-172-31-79-160 ~]$ s3-dist-cp --src=s3://amazon-reviews-pds/parquet/pr
uct_category=Sports/ --dest=hdfs:///hive/amazon-reviews-pds/parquet/product_ca
gory=Sports/
/04/13 01:32:21 INFO s3distcp.S3DistCp: Running with args: -libjars /usr/share
ws/emr/s3-dist-cp/lib/byte-buddy-1.9.10.jar,/usr/share/aws/emr/s3-dist-cp/lib/
te-buddy-agent-1.9.10.jar,/usr/share/aws/emr/s3-dist-cp/lib/commons-httpclient
.1.jar,/usr/share/aws/emr/s3-dist-cp/lib/commons-logging-1.0.4.jar,/usr/share/
s/emr/s3-dist-cp/lib/guava-18.0.jar,/usr/share/aws/emr/s3-dist-cp/lib/mockito-
re-2.27.0.jar,/usr/share/aws/emr/s3-dist-cp/lib/objenesis-2.6.jar,/usr/share/a
```

3. Create database

   Query:

   `create database amazon_review;`



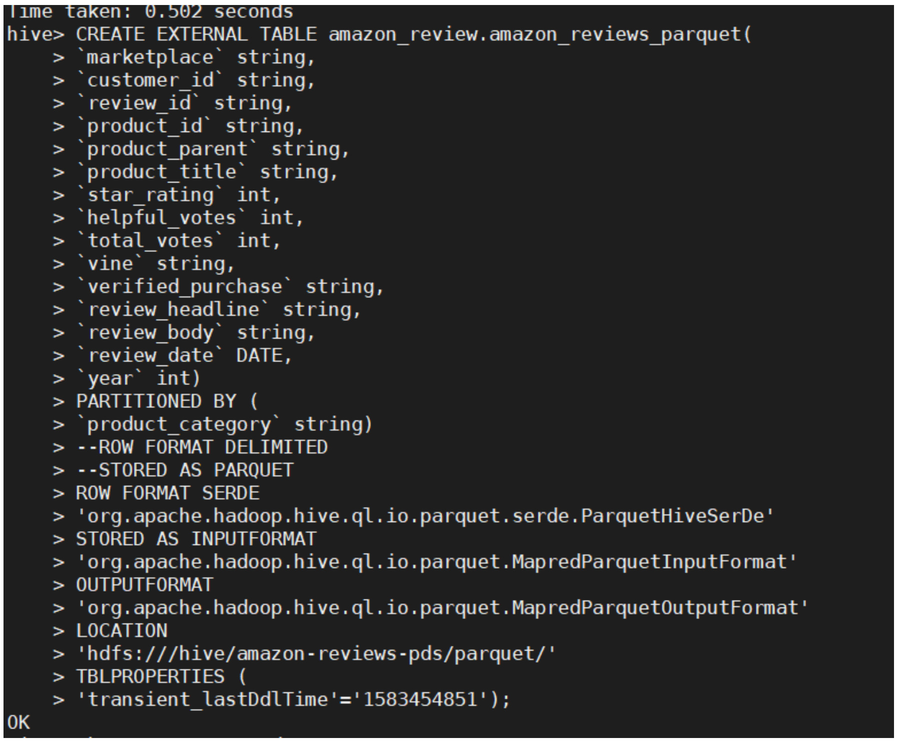
```
[hadoop@ip-172-31-79-160 ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
hive> create database amazon_review;
OK
Time taken: 0.772 seconds
hive> use amazon_review;
OK
Time taken: 0.04 seconds
```

4. Create an external table pointing to HDFS

   Query:

```
`CREATE EXTERNAL TABLE amazon_review.amazon_reviews_parquet(
`marketplace` string,
`customer_id` string,
`review_id` string,
`product_id` string,
`product_parent` string,
`product_title` string,
`star_rating` int,
`helpful_votes` int,
`total_votes` int,
`vine` string,
`verified_purchase` string,
`review_headline` string,
`review_body` string,
`review_date` DATE,
`year` int)
PARTITIONED BY (
`product_category` string)
--ROW FORMAT DELIMITED
--STORED AS PARQUET
ROW FORMAT SERDE
'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
STORED AS INPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
LOCATION
'hdfs:///hive/amazon-reviews-pds/parquet/'
TBLPROPERTIES (
'transient_lastDdlTime'='1583454851');`
```

   Msck repair table amazon_review.amazon_reviews_parquet;



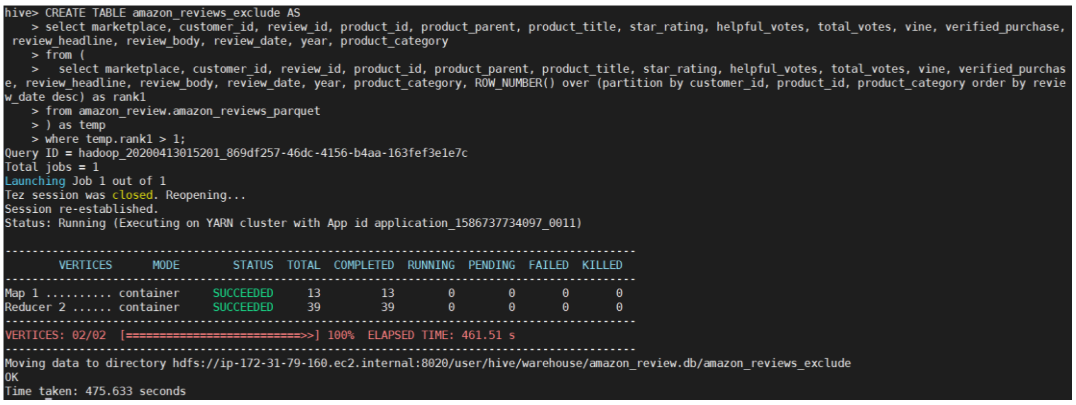
```
Time taken: 0.502 seconds
hive> CREATE EXTERNAL TABLE amazon_review.amazon_reviews_parquet(
    > `marketplace` string,
    > `customer_id` string,
    > `review_id` string,
    > `product_id` string,
    > `product_parent` string,
    > `product_title` string,
    > `star_rating` int,
    > `helpful_votes` int,
    > `total_votes` int,
    > `vine` string,
    > `verified_purchase` string,
    > `review_headline` string,
    > `review_body` string,
    > `review_date` DATE,
    > `year` int)
    > PARTITIONED BY (
    > `product_category` string)
    > --ROW FORMAT DELIMITED
    > --STORED AS PARQUET
    > ROW FORMAT SERDE
    > 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
    > STORED AS INPUTFORMAT
    > 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetInputFormat'
    > OUTPUTFORMAT
    > 'org.apache.hadoop.hive.ql.io.parquet.MapredParquetOutputFormat'
    > LOCATION
    > 'hdfs:///hive/amazon-reviews-pds/parquet/'
    > TBLPROPERTIES (
    > 'transient_lastDdlTime'='1583454851');
OK
```

5. Create an exclude table to exclude multiple reviews by a customer

   We create a table that will exclude all the review ids for customers that have multiple reviews.

   Query:

   `CREATE TABLE amazon_reviews_exclude AS`
   `select marketplace, customer_id, review_id, product_id, product_parent, product_title, star_rating, helpful_votes, total_votes, vine, verified_purchase, review_headline, review_body, review_date, year, product_category`
   `from (`
   `select marketplace, customer_id, review_id, product_id, product_parent, product_title, star_rating, helpful_votes, total_votes, vine, verified_purchase, review_headline, review_body, review_date, year, product_category, ROW_NUMBER() over (partition by customer_id, product_id, product_category order by review_date desc) as rank1`
   `from amazon_review.amazon_reviews_parquet`
   `) as temp`
   `where temp.rank1 > 1;`
   `select count(*) from amazon_reviews_exclude; --7159367`



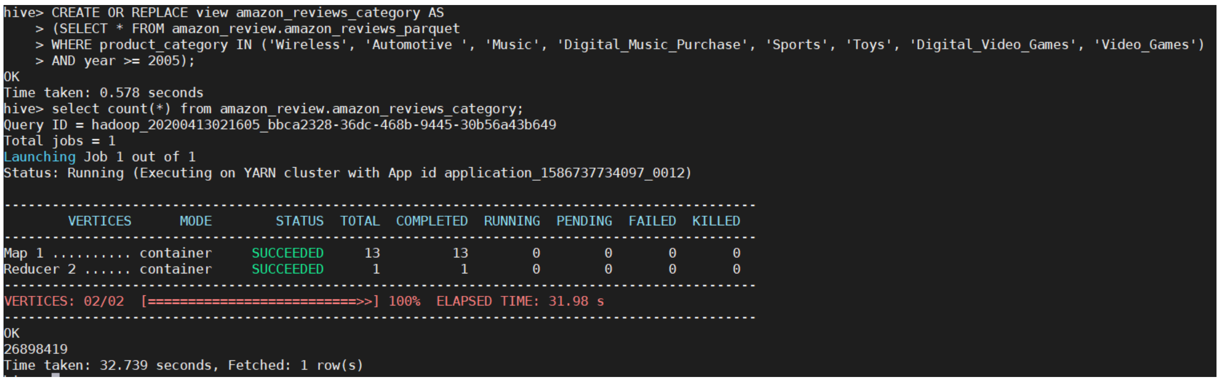
```
hive> CREATE TABLE amazon_reviews_exclude AS
    > select marketplace, customer_id, review_id, product_id, product_parent, product_title, star_rating, helpful_votes, total_votes, vine, verified_purchase,
 review_headline, review_body, review_date, year, product_category
    > from (
    >   select marketplace, customer_id, review_id, product_id, product_parent, product_title, star_rating, helpful_votes, total_votes, vine, verified_purchas
e, review_headline, review_body, review_date, year, product_category, ROW_NUMBER() over (partition by customer_id, product_id, product_category order by revie
w_date desc) as rank1
    > from amazon_review.amazon_reviews_parquet
    > ) as temp
    > where temp.rank1 > 1;
Query ID = hadoop_20200413015201_869df257-46dc-4156-b4aa-163fef3e1e7c
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_1586737734097_0011)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     13         13        0        0       0       0
Reducer 2 ...... container     SUCCEEDED     39         39        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 461.51 s
----------------------------------------------------------------------------------------------
Moving data to directory hdfs://ip-172-31-79-160.ec2.internal:8020/user/hive/warehouse/amazon_review.db/amazon_reviews_exclude
OK
Time taken: 475.633 seconds
```

6. Create a category view which will apply filters for our reviews

   We created a view that will only consider those records with year >= 2005 and should be in one of the categories we need.

   Query:

   `CREATE OR REPLACE VIEW amazon_reviews_category AS`
   `SELECT * FROM amazon_review.amazon_reviews_parquet`
   `WHERE ("product_category" IN ('Wireless', 'Automotive ', 'Music', 'Digital_Music_Purchase', 'Sports', 'Toys', 'Digital_Video_Games', 'Video_Games')`
   `AND "year" >= 2005);`
   `select count(*) from amazon_reviews_category;` -- 26898419

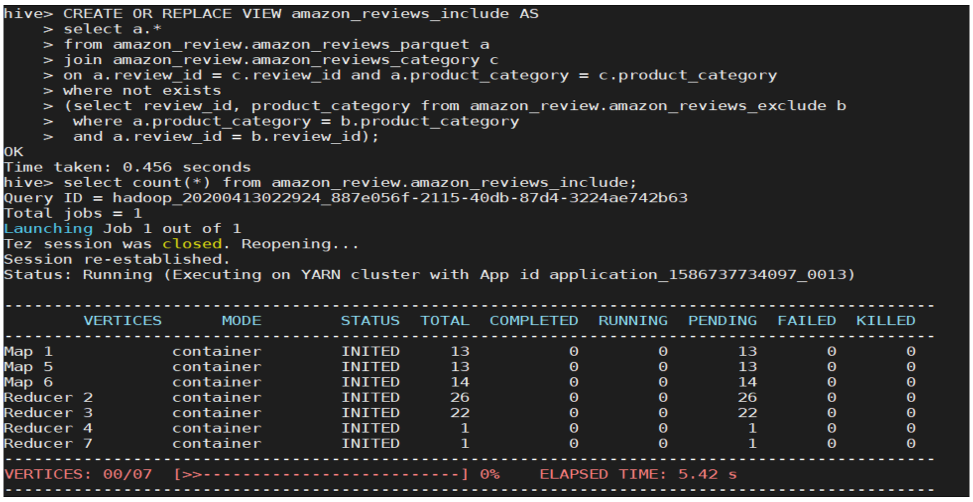


```
hive> CREATE OR REPLACE view amazon_reviews_category AS
    > (SELECT * FROM amazon_review.amazon_reviews_parquet
    > WHERE product_category IN ('Wireless', 'Automotive ', 'Music', 'Digital_Music_Purchase', 'Sports', 'Toys', 'Digital_Video_Games', 'Video_Games')
    > AND year >= 2005);
OK
Time taken: 0.578 seconds
hive> select count(*) from amazon_review.amazon_reviews_category;
Query ID = hadoop_20200413021605_bbca2328-36dc-468b-9445-30b56a43b649
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1586737734097_0012)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     13         13        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 31.98 s
----------------------------------------------------------------------------------------------
OK
26898419
Time taken: 32.739 seconds, Fetched: 1 row(s)
```

7. Create an include table that will be a join between the exclude table and the category view and will be our final table to query on.

   Query:

   `CREATE OR REPLACE VIEW amazon_reviews_include AS`
   `select a.*`
   `from amazon_review.amazon_reviews_parquet a`
   `join amazon_review.amazon_reviews_category c`
   `on a.review_id = c.review_id and a.product_category = c.product_category`
   `where not exists`
   `(select review_id, product_category from amazon_review.amazon_reviews_exclude b`
    `where a.product_category = b.product_category`
    `and a.review_id = b.review_id);`



```
hive> CREATE OR REPLACE VIEW amazon_reviews_include AS
    > select a.*
    > from amazon_review.amazon_reviews_parquet a
    > join amazon_review.amazon_reviews_category c
    > on a.review_id = c.review_id and a.product_category = c.product_category
    > where not exists
    > (select review_id, product_category from amazon_review.amazon_reviews_exclude b
    >  where a.product_category = b.product_category
    >  and a.review_id = b.review_id);
OK
Time taken: 0.456 seconds
hive> select count(*) from amazon_review.amazon_reviews_include;
Query ID = hadoop_20200413022924_887e056f-2115-40db-87d4-3224ae742b63
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_1586737734097_0013)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1            container       INITED     13          0        0       13       0       0
Map 5            container       INITED     13          0        0       13       0       0
Map 6            container       INITED     14          0        0       14       0       0
Reducer 2        container       INITED     26          0        0       26       0       0
Reducer 3        container       INITED     22          0        0       22       0       0
Reducer 4        container       INITED      1          0        0        1       0       0
Reducer 7        container       INITED      1          0        0        1       0       0
----------------------------------------------------------------------------------------------
VERTICES: 00/07  [>>--------------------------] 0%    ELAPSED TIME: 5.42 s
----------------------------------------------------------------------------------------------
```

`select count(*) from amazon_review.amazon_reviews_include;` -- 25335861

Using AWS EMR Hive and AWS Athena, answer the following questions:

1. Explore the dataset and provide basic exploratory analysis:

1. Number of reviews

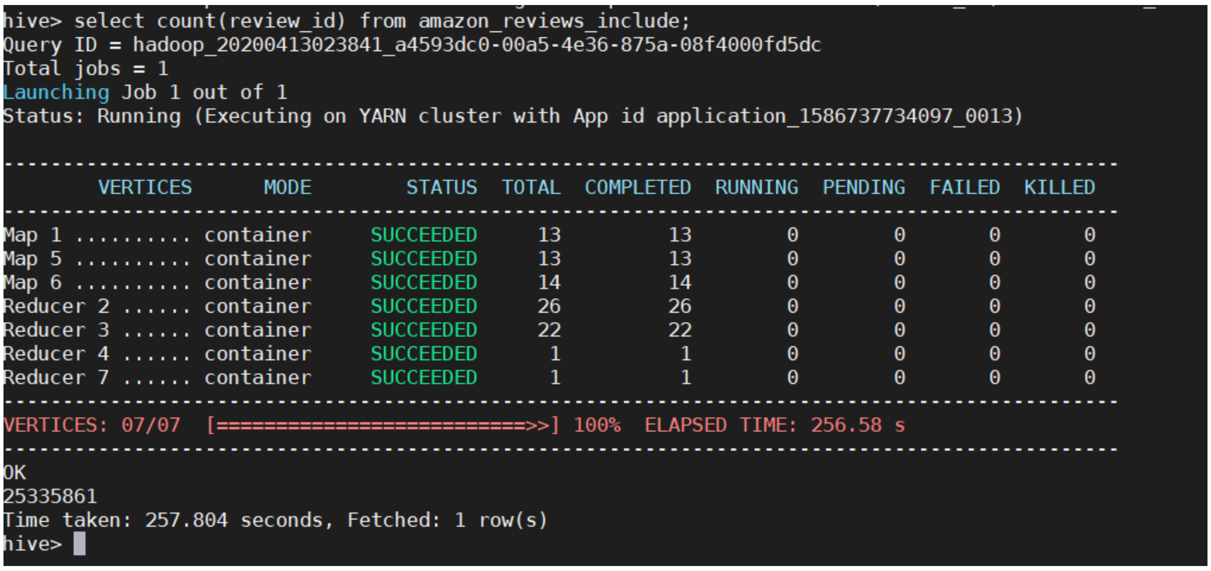
To calculate the number of reviews in our final table – amazon_reviews_include

Query:

`select count(review_id) from amazon_reviews_include;`

Output:

25335861



```
hive> select count(review_id) from amazon_reviews_include;
Query ID = hadoop_20200413023841_a4593dc0-00a5-4e36-875a-08f4000fd5dc
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1586737734097_0013)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     13         13        0        0       0       0
Map 5 .......... container     SUCCEEDED     13         13        0        0       0       0
Map 6 .......... container     SUCCEEDED     14         14        0        0       0       0
Reducer 2 ...... container     SUCCEEDED     26         26        0        0       0       0
Reducer 3 ...... container     SUCCEEDED     22         22        0        0       0       0
Reducer 4 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 7 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 07/07  [==========================>>] 100%  ELAPSED TIME: 256.58 s
----------------------------------------------------------------------------------------------
OK
25335861
Time taken: 257.804 seconds, Fetched: 1 row(s)
hive>
```

2. Number of users

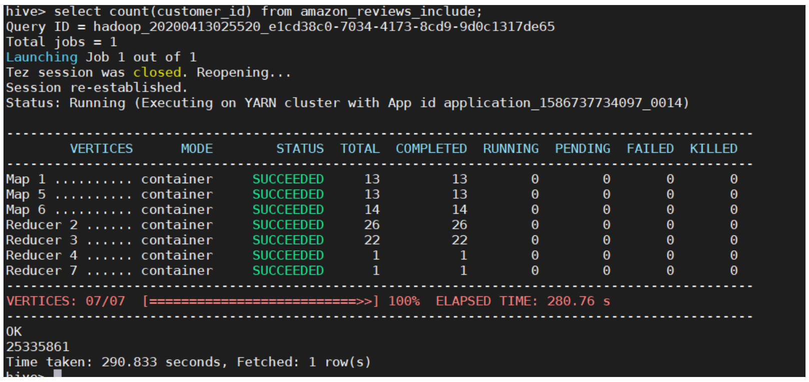
To calculate the number of users a.k.a number of customer ids in our final table.

Query:

`select count(customer_id) from amazon_reviews_include;`

Output:

25335861



```
hive> select count(customer_id) from amazon_reviews_include;
Query ID = hadoop_20200413025520_e1cd38c0-7034-4173-8cd9-9d0c1317de65
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_1586737734097_0014)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     13         13        0        0       0       0
Map 5 .......... container     SUCCEEDED     13         13        0        0       0       0
Map 6 .......... container     SUCCEEDED     14         14        0        0       0       0
Reducer 2 ...... container     SUCCEEDED     26         26        0        0       0       0
Reducer 3 ...... container     SUCCEEDED     22         22        0        0       0       0
Reducer 4 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 7 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 07/07  [==========================>>] 100%  ELAPSED TIME: 280.76 s
----------------------------------------------------------------------------------------------
OK
25335861
Time taken: 290.833 seconds, Fetched: 1 row(s)
```

3. Average review stars

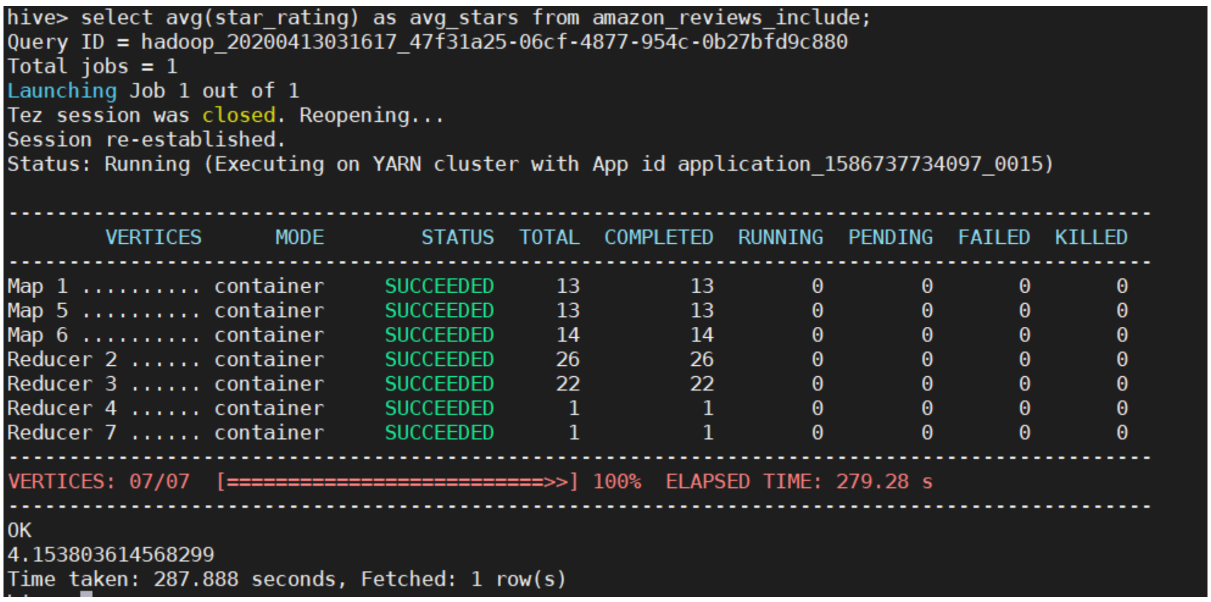
To calculate the average review stars for all years from 2005 for all product categories.

Query:

` select avg(star_rating) as avg_stars from amazon_reviews_include;`

Output:

4.153803614568299



```
hive> select avg(star_rating) as avg_stars from amazon_reviews_include;
Query ID = hadoop_20200413031617_47f31a25-06cf-4877-954c-0b27bfd9c880
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_1586737734097_0015)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     13         13        0        0       0       0
Map 5 .......... container     SUCCEEDED     13         13        0        0       0       0
Map 6 .......... container     SUCCEEDED     14         14        0        0       0       0
Reducer 2 ...... container     SUCCEEDED     26         26        0        0       0       0
Reducer 3 ...... container     SUCCEEDED     22         22        0        0       0       0
Reducer 4 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 7 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 07/07  [==========================>>] 100%  ELAPSED TIME: 279.28 s
----------------------------------------------------------------------------------------------
OK
4.153803614568299
Time taken: 287.888 seconds, Fetched: 1 row(s)
```

4. Average length of the review

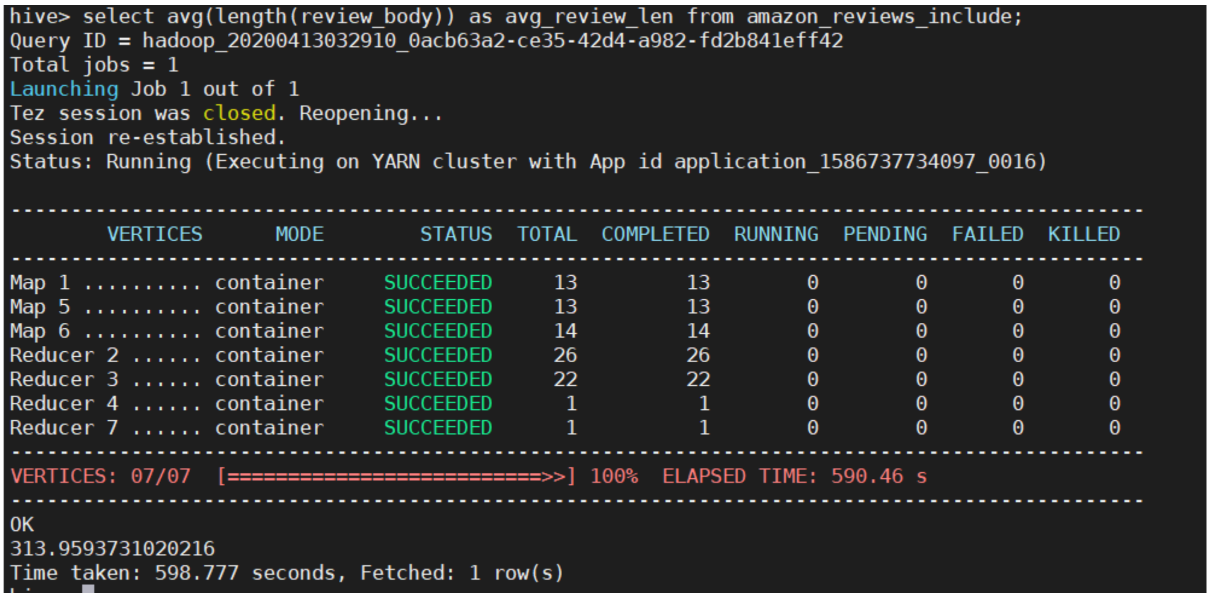
To calculate the average length of reviews by customers.

Query:

`select avg(length(review_body)) as avg_review_len from amazon_reviews_include;`

Output:

313.9593731020216



```
hive> select avg(length(review_body)) as avg_review_len from amazon_reviews_include;
Query ID = hadoop_20200413032910_0acb63a2-ce35-42d4-a982-fd2b841eff42
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_1586737734097_0016)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     13         13        0        0       0       0
Map 5 .......... container     SUCCEEDED     13         13        0        0       0       0
Map 6 .......... container     SUCCEEDED     14         14        0        0       0       0
Reducer 2 ...... container     SUCCEEDED     26         26        0        0       0       0
Reducer 3 ...... container     SUCCEEDED     22         22        0        0       0       0
Reducer 4 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 7 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 07/07  [==========================>>] 100%  ELAPSED TIME: 590.46 s
----------------------------------------------------------------------------------------------
OK
313.9593731020216
Time taken: 598.777 seconds, Fetched: 1 row(s)
```

5. Number of verified versus unverified reviews

To calculate the number of verified and unverified purchases of products on amazon by the customers.

Query:

` select 'verified' as review_type, count() from amazon_reviews_include where verified_purchase = 'Y'

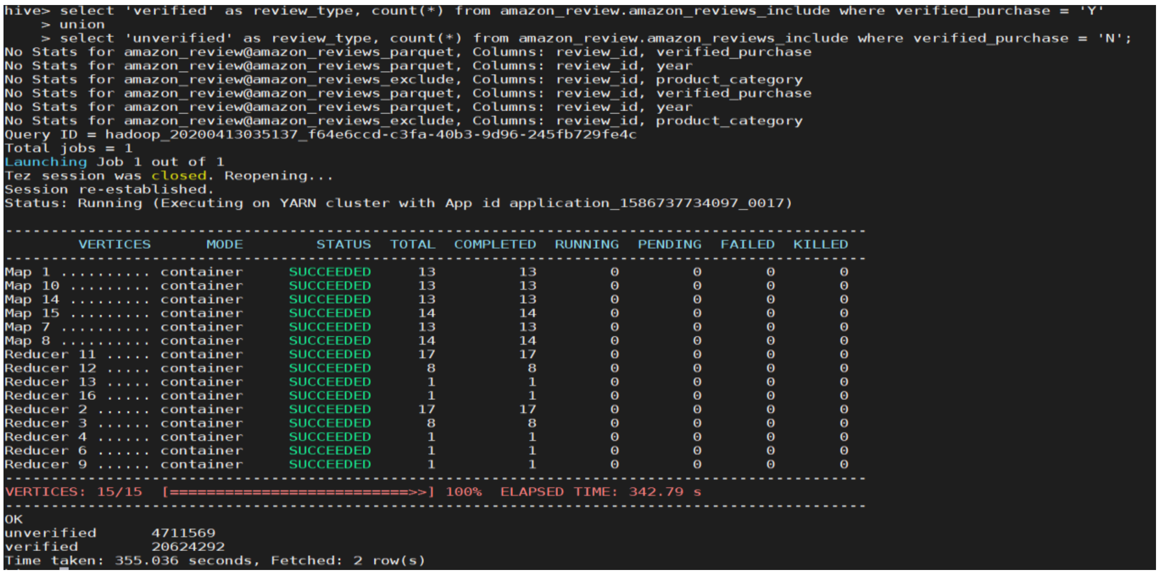
union

select 'unverified' as review_type, count() from amazon_reviews_include where verified_purchase = 'N'`

Output:

Verified Purchase – 20624292

Unverified Purchase - 4711569



```
hive> select 'verified' as review_type, count(*) from amazon_review.amazon_reviews_include where verified_purchase = 'Y'
    > union
    > select 'unverified' as review_type, count(*) from amazon_review.amazon_reviews_include where verified_purchase = 'N';
No Stats for amazon_review@amazon_reviews_parquet, Columns: review_id, verified_purchase
No Stats for amazon_review@amazon_reviews_parquet, Columns: review_id, year
No Stats for amazon_review@amazon_reviews_exclude, Columns: review_id, product_category
No Stats for amazon_review@amazon_reviews_parquet, Columns: review_id, verified_purchase
No Stats for amazon_review@amazon_reviews_parquet, Columns: review_id, year
No Stats for amazon_review@amazon_reviews_exclude, Columns: review_id, product_category
Query ID = hadoop_20200413035137_f64e6ccd-c3fa-40b3-9d96-245fb729fe4c
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_1586737734097_0017)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     13         13        0        0       0       0
Map 10 ......... container     SUCCEEDED     13         13        0        0       0       0
Map 14 ......... container     SUCCEEDED     13         13        0        0       0       0
Map 15 ......... container     SUCCEEDED     14         14        0        0       0       0
Map 7 .......... container     SUCCEEDED     13         13        0        0       0       0
Map 8 .......... container     SUCCEEDED     14         14        0        0       0       0
Reducer 11 ..... container     SUCCEEDED     17         17        0        0       0       0
Reducer 12 ..... container     SUCCEEDED      8          8        0        0       0       0
Reducer 13 ..... container     SUCCEEDED      1          1        0        0       0       0
Reducer 16 ..... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED     17         17        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      8          8        0        0       0       0
Reducer 4 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 6 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 9 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 15/15  [==========================>>] 100%  ELAPSED TIME: 342.79 s
----------------------------------------------------------------------------------------------
OK
unverified      4711569
verified        20624292
Time taken: 355.036 seconds, Fetched: 2 row(s)
```

6. Top 10 Customer IDs with most helpful votes

To display the top 10 customers with the most number of helpful votes.

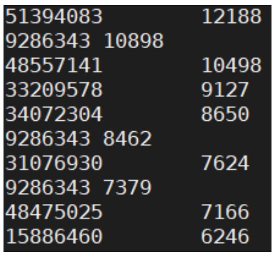
Query:

`select customer_id, helpful_votes from amazon_reviews_include`

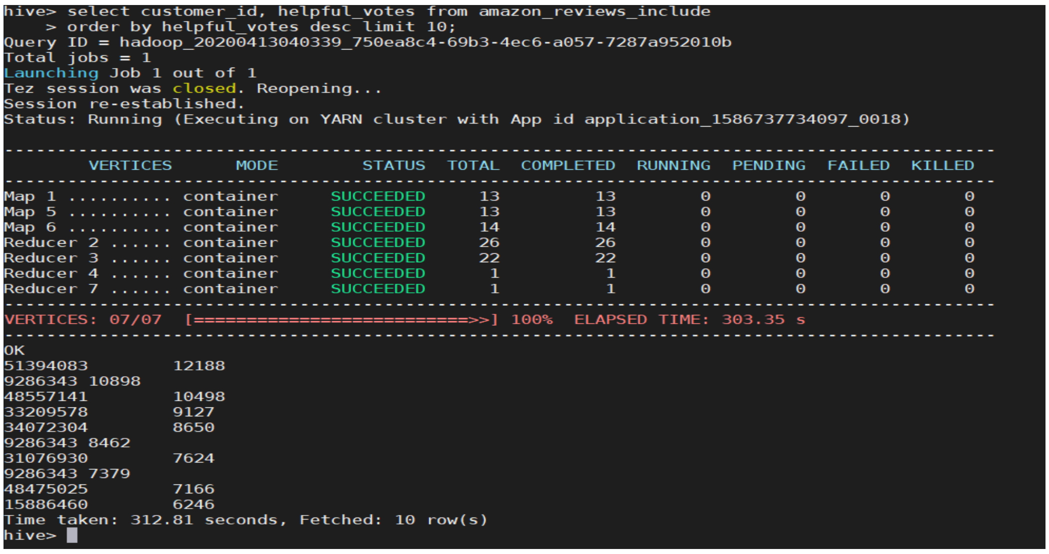
`order by helpful_votes desc limit 10;`

Output:

Customer_id   helpful_votes



```
51394083        12188
9286343 10898
48557141        10498
33209578        9127
34072304        8650
9286343 8462
31076930        7624
9286343 7379
48475025        7166
15886460        6246
```



```
hive> select customer_id, helpful_votes from amazon_reviews_include
    > order by helpful_votes desc limit 10;
Query ID = hadoop_20200413040339_750ea8c4-69b3-4ec6-a057-7287a952010b
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_1586737734097_0018)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     13         13        0        0       0       0
Map 5 .......... container     SUCCEEDED     13         13        0        0       0       0
Map 6 .......... container     SUCCEEDED     14         14        0        0       0       0
Reducer 2 ...... container     SUCCEEDED     26         26        0        0       0       0
Reducer 3 ...... container     SUCCEEDED     22         22        0        0       0       0
Reducer 4 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 7 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 07/07  [==========================>>] 100%  ELAPSED TIME: 303.35 s
----------------------------------------------------------------------------------------------
OK
51394083        12188
9286343 10898
48557141        10498
33209578        9127
34072304        8650
9286343 8462
31076930        7624
9286343 7379
48475025        7166
15886460        6246
Time taken: 312.81 seconds, Fetched: 10 row(s)
hive>
```

7. Most number of product categories reviewed in a given year

To calculate the total number of product categories reviewed in a given year.

Query:

`select year, count(product_category) as NoOfCategoriesReviewed`

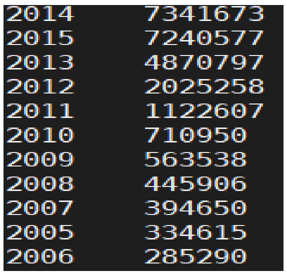
`from amazon_reviews_include`

`group by year`

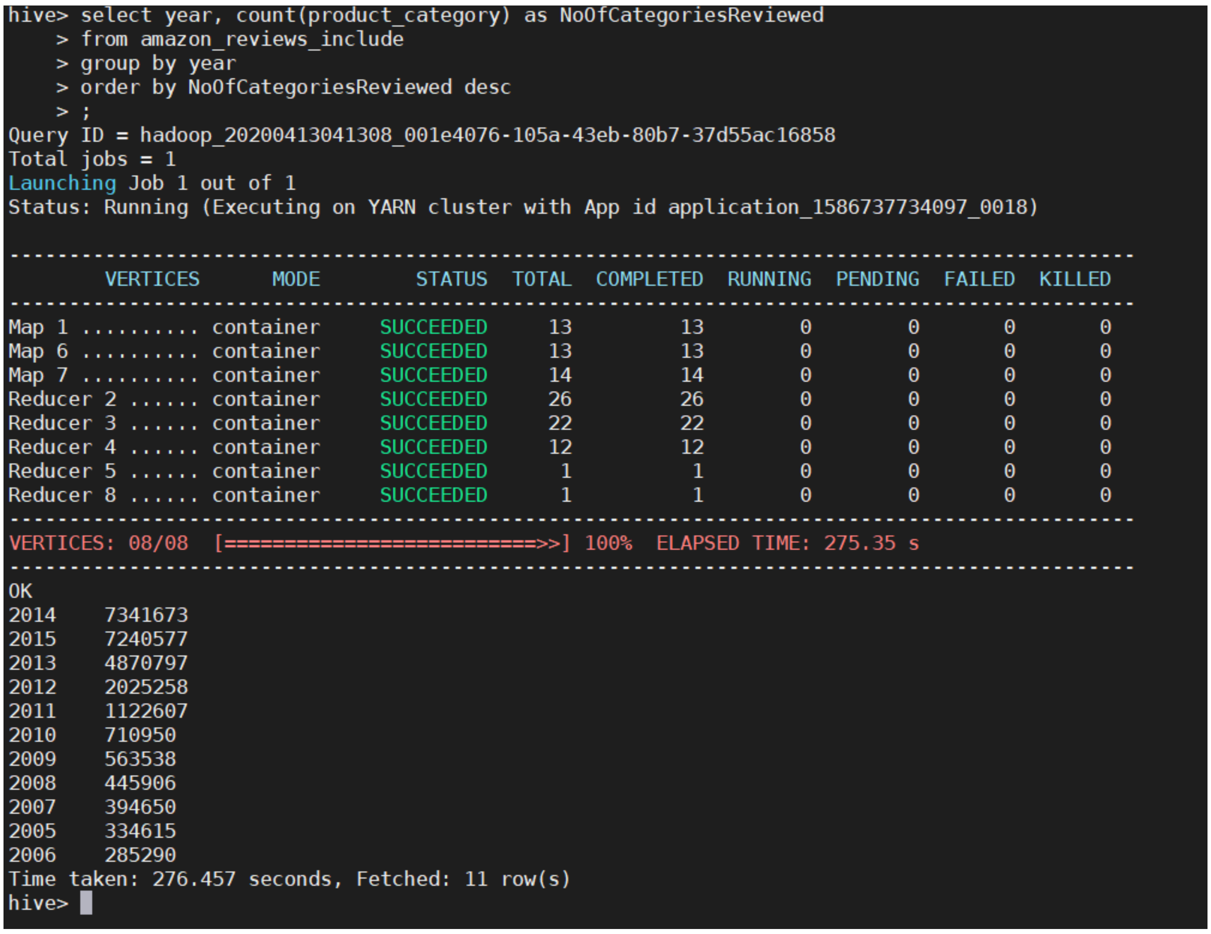
`order by NoOfCategoriesReviewed desc`

Output:

Year NoOfCatgoriesReviewed



```
2014      7341673
2015      7240577
2013      4870797
2012      2025258
2011      1122607
2010      710950
2009      563538
2008      445906
2007      394650
2005      334615
2006      285290
```



```
hive> select year, count(product_category) as NoOfCategoriesReviewed
    > from amazon_reviews_include
    > group by year
    > order by NoOfCategoriesReviewed desc
    > ;
Query ID = hadoop_20200413041308_001e4076-105a-43eb-80b7-37d55ac16858
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1586737734097_0018)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     13         13        0        0       0       0
Map 6 .......... container     SUCCEEDED     13         13        0        0       0       0
Map 7 .......... container     SUCCEEDED     14         14        0        0       0       0
Reducer 2 ...... container     SUCCEEDED     26         26        0        0       0       0
Reducer 3 ...... container     SUCCEEDED     22         22        0        0       0       0
Reducer 4 ...... container     SUCCEEDED     12         12        0        0       0       0
Reducer 5 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 8 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 08/08  [==========================>>] 100%  ELAPSED TIME: 275.35 s
----------------------------------------------------------------------------------------------
OK
2014    7341673
2015    7240577
2013    4870797
2012    2025258
2011    1122607
2010    710950
2009    563538
2008    445906
2007    394650
2005    334615
2006    285290
Time taken: 276.457 seconds, Fetched: 11 row(s)
hive>
```

8. Provide trending (over time) analysis of each of the metrics above

Queries:

-- marketplace

To calculate the number of reviews in a marketplace in a given year.

Query:

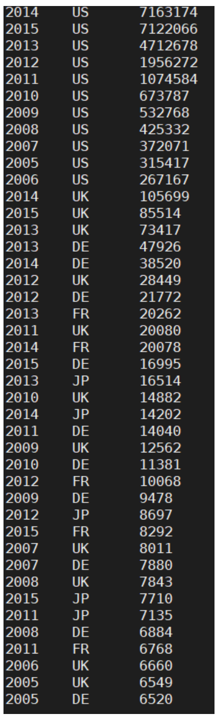
select year, marketplace, count(*) as CountMarketplace

from amazon_reviews_include

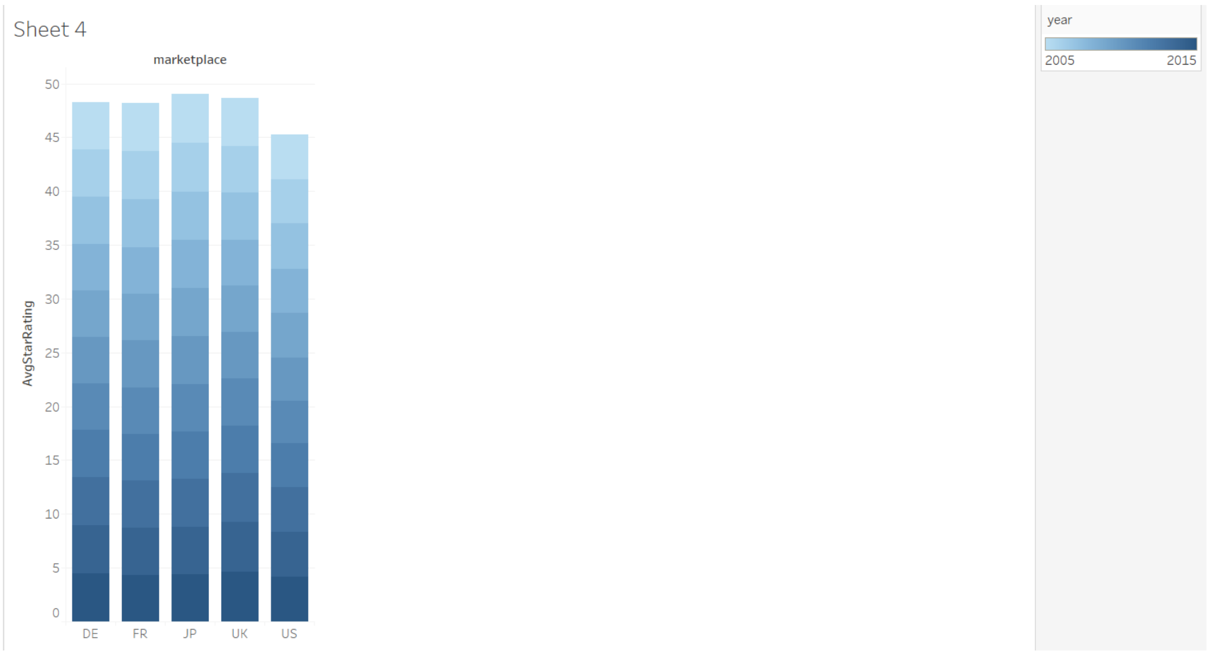
group by year, marketplace

order by CountMarketplace desc;

Output:



```
2014    US      7163174
2015    US      7122066
2013    US      4712678
2012    US      1956272
2011    US      1074584
2010    US      673787
2009    US      532768
2008    US      425332
2007    US      372071
2005    US      315417
2006    US      267167
2014    UK      105699
2015    UK      85514
2013    UK      73417
2013    DE      47926
2014    DE      38520
2012    UK      28449
2012    DE      21772
2013    FR      20262
2011    UK      20080
2014    FR      20078
2015    DE      16995
2013    JP      16514
2010    UK      14882
2014    JP      14202
2011    DE      14040
2009    UK      12562
2010    DE      11381
2012    FR      10068
2009    DE      9478
2012    JP      8697
2015    FR      8292
2007    UK      8011
2007    DE      7880
2008    UK      7843
2015    JP      7710
2011    JP      7135
2008    DE      6884
2011    FR      6768
2006    UK      6660
2005    UK      6549
2005    DE      6520
```



-- customer_id

To count the number of users reviewing products in a given year

Query:

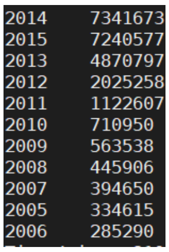
select year, count(customer_id) as CountCustomerID

from amazon_reviews_include

group by year

order by CountCustomerID desc;

Output:



```
2014    7341673
2015    7240577
2013    4870797
2012    2025258
2011    1122607
2010    710950
2009    563538
2008    445906
2007    394650
2005    334615
2006    285290
```

-- review_id

To count the number of reviews reviewing products in a given year

Query:

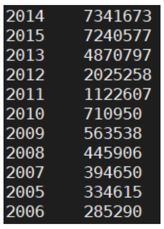
select year, count(review_id) as CountReviewID

from amazon_reviews_include

group by year

order by CountReviewID desc;

Output:


```
2014    7341673
2015    7240577
2013    4870797
2012    2025258
2011    1122607
2010    710950
2009    563538
2008    445906
2007    394650
2005    334615
2006    285290
```

-- product_id

To count the number of products in a given year.

Query:

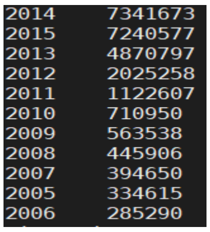
select year, count(product_id) as CountProductID

from amazon_reviews_include

group by year

order by CountProductID desc;

Output:


```
2014    7341673
2015    7240577
2013    4870797
2012    2025258
2011    1122607
2010    710950
2009    563538
2008    445906
2007    394650
2005    334615
2006    285290
```

-- star_rating

To calculate the average star rating for all the products in a given year ordered in decreasing order.

Query:

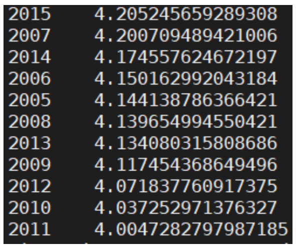
select year, avg(star_rating) as AvgStarRating

from amazon_reviews_include

group by year

order by AvgStarRating desc;

Output:



```
2015    4.205245659289308
2007    4.200709489421006
2014    4.174557624672197
2006    4.150162992043184
2005    4.144138786366421
2008    4.139654994550421
2013    4.134080315808686
2009    4.117454368649496
2012    4.071837760917375
2010    4.037252971376327
2011    4.0047282797987185
```

-- helpful_votes

To calculate the average of helpful votes in a given year ordered by the average votes in descending order.

Query:

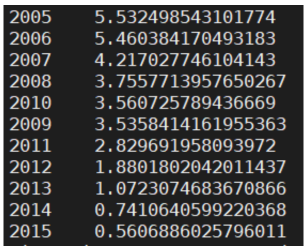
select year, avg(helpful_votes) as AvgHelpfulVotes

from amazon_reviews_include

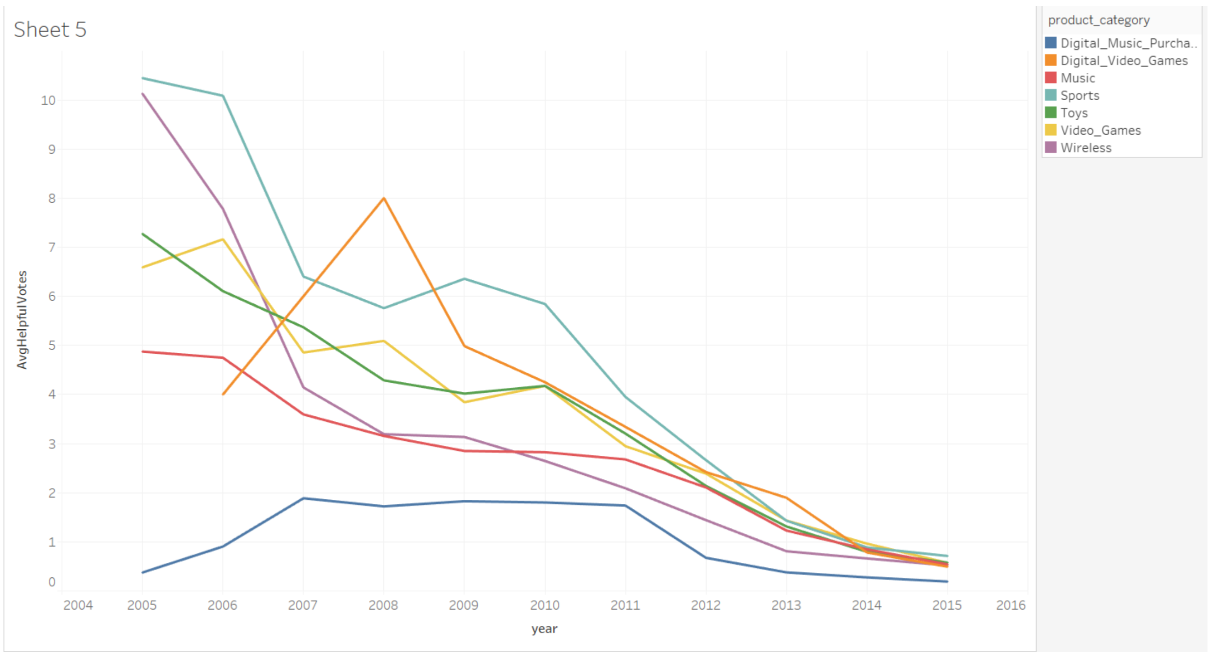
group by year

order by AvgHelpfulVotes desc;

Output:



```
2005    5.532498543101774
2006    5.460384170493183
2007    4.217027746104143
2008    3.7557713957650267
2010    3.560725789436669
2009    3.5358414161955363
2011    2.829691958093972
2012    1.8801802042011437
2013    1.0723074683670866
2014    0.7410640599220368
2015    0.5606886025796011
```



-- total_votes

To calculate average of total votes by customers in a given year and ordered by the average of total votes in descending votes.

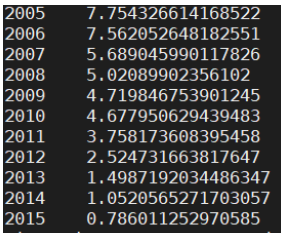
select year, avg(total_votes) as AvgTotalVotes

from amazon_reviews_include

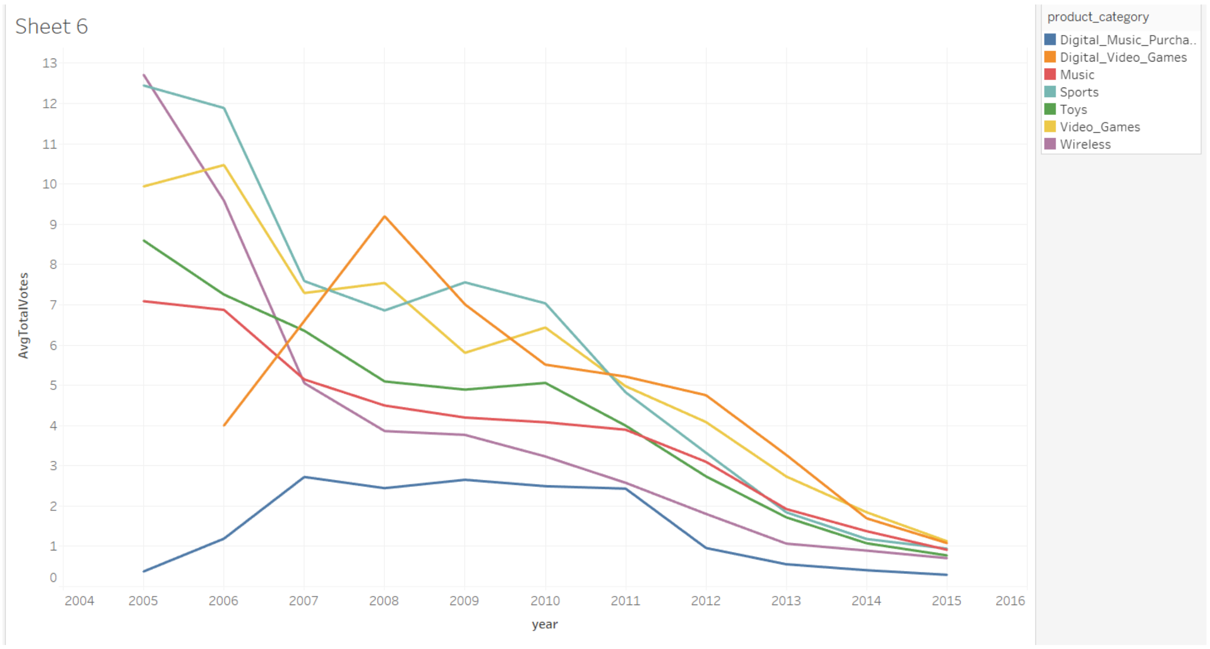
group by year

order by AvgTotalVotes desc;

Output:



```
2005    7.754326614168522
2006    7.562052648182551
2007    5.689045990117826
2008    5.02089902356102
2009    4.719846753901245
2010    4.677950629439483
2011    3.758173608395458
2012    2.524731663817647
2013    1.4987192034486347
2014    1.0520565271703057
2015    0.786011252970585
```



-- verified_purchase

To calculate the number of verified and unverified purchases by all the customers staring from year 2005.

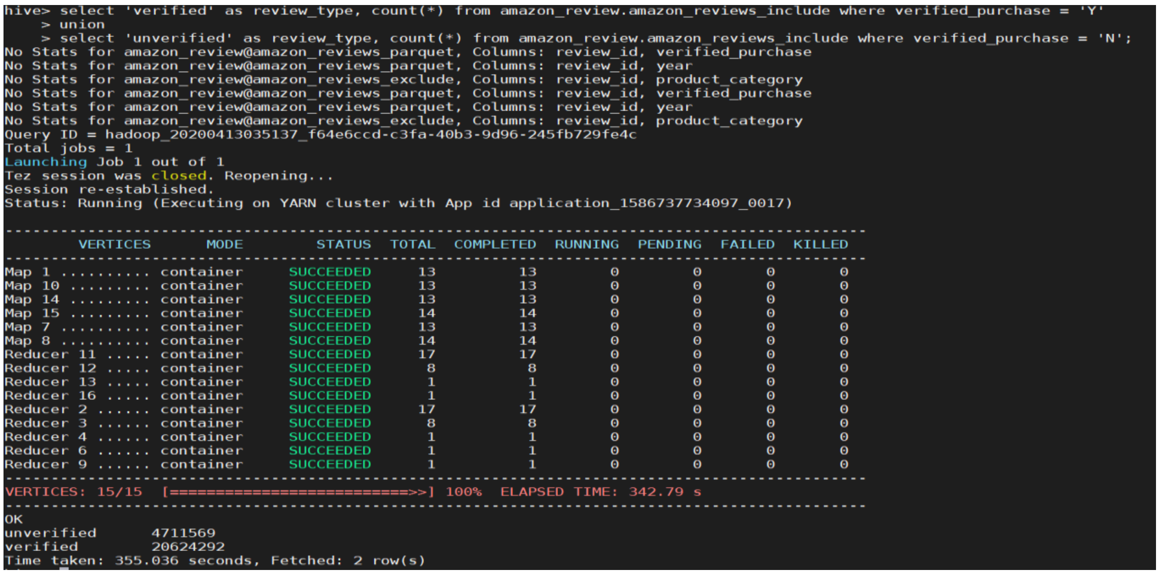
Query:

select 'verified' as review_type, count() from amazon_reviews_include where verified_purchase = 'Y'

union

select 'unverified' as review_type, count() from amazon_reviews_include where verified_purchase = 'N'

Output:



```
hive> select 'verified' as review_type, count(*) from amazon_review.amazon_reviews_include where verified_purchase = 'Y'
    > union
    > select 'unverified' as review_type, count(*) from amazon_review.amazon_reviews_include where verified_purchase = 'N';
No Stats for amazon_review@amazon_reviews_parquet, Columns: review_id, verified_purchase
No Stats for amazon_review@amazon_reviews_parquet, Columns: review_id, year
No Stats for amazon_review@amazon_reviews_exclude, Columns: review_id, product_category
No Stats for amazon_review@amazon_reviews_parquet, Columns: review_id, verified_purchase
No Stats for amazon_review@amazon_reviews_parquet, Columns: review_id, year
No Stats for amazon_review@amazon_reviews_exclude, Columns: review_id, product_category
Query ID = hadoop_20200413035137_f64e6ccd-c3fa-40b3-9d96-245fb729fe4c
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_1586737734097_0017)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     13         13        0        0       0       0
Map 10 ......... container     SUCCEEDED     13         13        0        0       0       0
Map 14 ......... container     SUCCEEDED     13         13        0        0       0       0
Map 15 ......... container     SUCCEEDED     14         14        0        0       0       0
Map 7 .......... container     SUCCEEDED     13         13        0        0       0       0
Map 8 .......... container     SUCCEEDED     14         14        0        0       0       0
Reducer 11 ..... container     SUCCEEDED     17         17        0        0       0       0
Reducer 12 ..... container     SUCCEEDED      8          8        0        0       0       0
Reducer 13 ..... container     SUCCEEDED      1          1        0        0       0       0
Reducer 16 ..... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED     17         17        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      8          8        0        0       0       0
Reducer 4 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 6 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 9 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 15/15  [==========================>>] 100%  ELAPSED TIME: 342.79 s
----------------------------------------------------------------------------------------------
OK
unverified      4711569
verified        20624292
Time taken: 355.036 seconds, Fetched: 2 row(s)
```

-- product_category

To calculate the number of product categories reviewed in a given year.

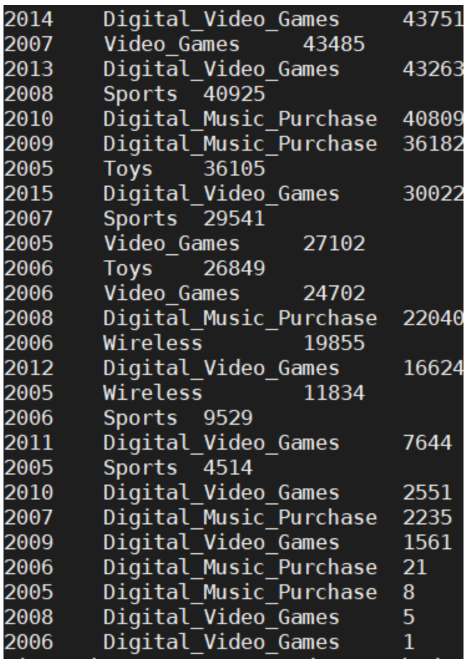
Query:

select year, product_category, count(*) as record_count from amazon_reviews_include

group by year, product_category

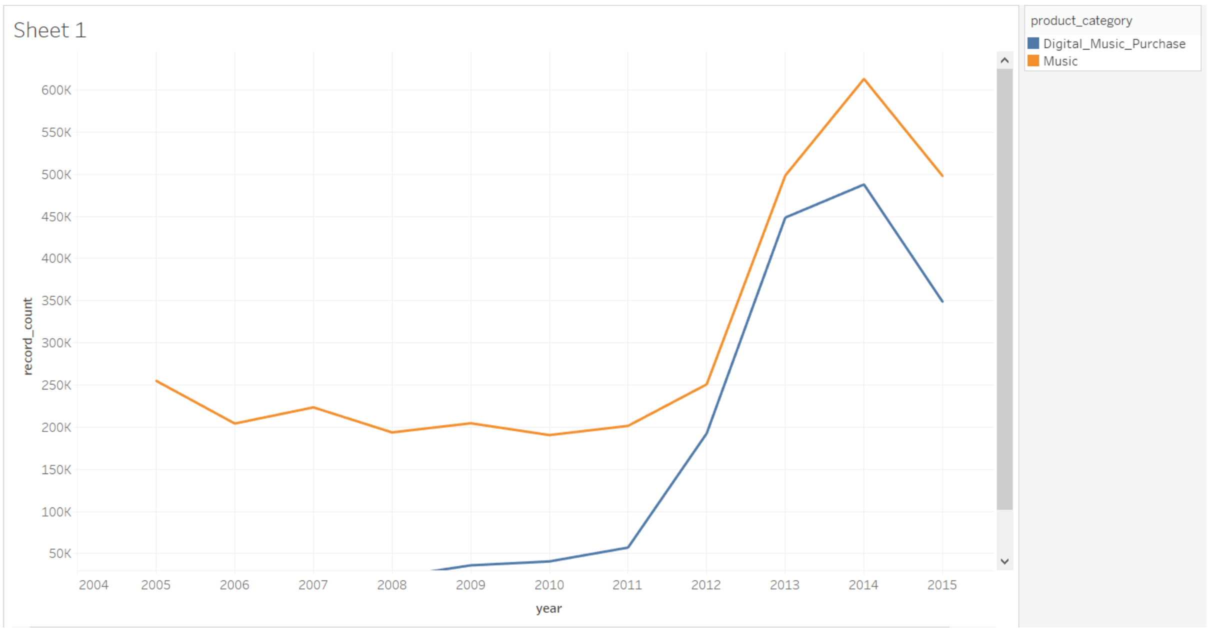
order by record_count desc;

Output:



```
2014    Digital_Video_Games     43751
2007    Video_Games     43485
2013    Digital_Video_Games     43263
2008    Sports  40925
2010    Digital_Music_Purchase  40809
2009    Digital_Music_Purchase  36182
2005    Toys    36105
2015    Digital_Video_Games     30022
2007    Sports  29541
2005    Video_Games     27102
2006    Toys    26849
2006    Video_Games     24702
2008    Digital_Music_Purchase  22040
2006    Wireless        19855
2012    Digital_Video_Games     16624
2005    Wireless        11834
2006    Sports  9529
2011    Digital_Video_Games     7644
2005    Sports  4514
2010    Digital_Video_Games     2551
2007    Digital_Music_Purchase  2235
2009    Digital_Video_Games     1561
2006    Digital_Music_Purchase  21
2005    Digital_Music_Purchase  8
2008    Digital_Video_Games     5
2006    Digital_Video_Games     1
```

Only displaying the last 20 rows



2. **Provide detailed analysis of Music/Digital_Music_Purchase and Digital_Video_Games/Video_Games over time.**

Query:

-- Music/Digital_Music_Purchase over time.

We created two separate tables for product categories ‘Music’ and ‘Digital Music Purchase’ as it becomes easier to compare the two categories and answer questions based on it.

Query:

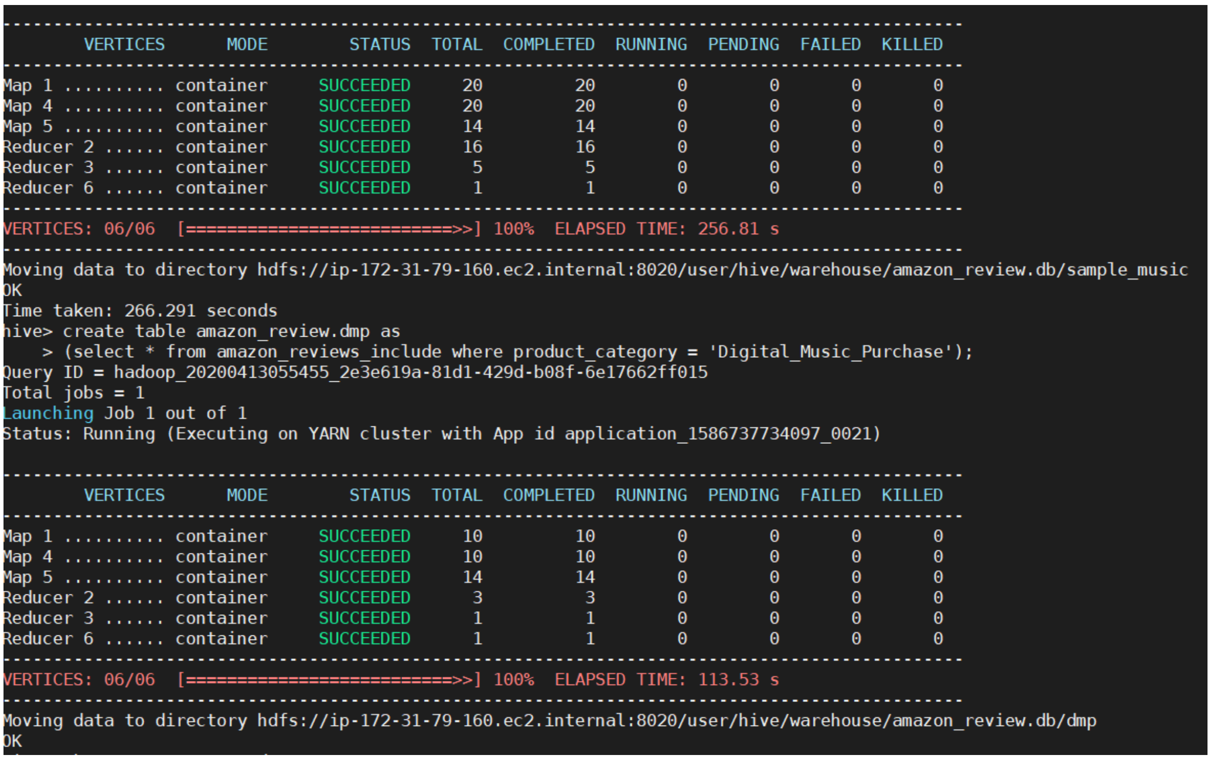
`create table amazon_review.sample_music as `

`(select * from amazon_reviews_include where product_category = 'Music') with data;`

`create table amazon_review.dmp as`

`(select * from amazon_reviews_include where product_category = 'Digital_Music_Purchase') with data;`

Output:



```
----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     20         20        0        0       0       0
Map 4 .......... container     SUCCEEDED     20         20        0        0       0       0
Map 5 .......... container     SUCCEEDED     14         14        0        0       0       0
Reducer 2 ...... container     SUCCEEDED     16         16        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      5          5        0        0       0       0
Reducer 6 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 06/06  [==========================>>] 100%  ELAPSED TIME: 256.81 s
----------------------------------------------------------------------------------------------
Moving data to directory hdfs://ip-172-31-79-160.ec2.internal:8020/user/hive/warehouse/amazon_review.db/sample_music
OK
Time taken: 266.291 seconds
hive> create table amazon_review.dmp as
    > (select * from amazon_reviews_include where product_category = 'Digital_Music_Purchase');
Query ID = hadoop_20200413055455_2e3e619a-81d1-429d-b08f-6e17662ff015
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1586737734097_0021)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     10         10        0        0       0       0
Map 4 .......... container     SUCCEEDED     10         10        0        0       0       0
Map 5 .......... container     SUCCEEDED     14         14        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      3          3        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 6 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 06/06  [==========================>>] 100%  ELAPSED TIME: 113.53 s
----------------------------------------------------------------------------------------------
Moving data to directory hdfs://ip-172-31-79-160.ec2.internal:8020/user/hive/warehouse/amazon_review.db/dmp
OK
```

1. Do you see correlation (maybe negative) between the categories over time?

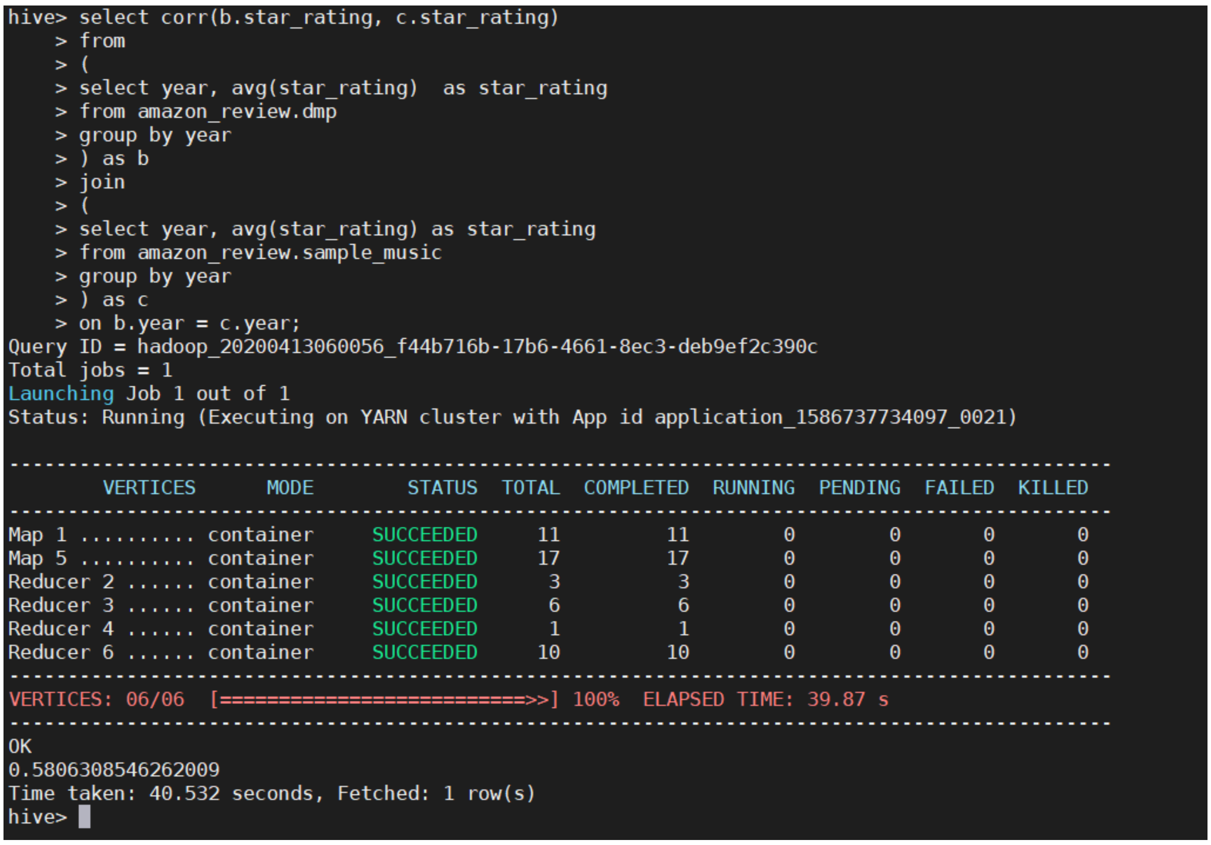
We attempt to correlate the star ratings of both product categories and use corr() function to deteremine the correlation coefficient.

Query:

```
select corr(b.star_rating, c.star_rating)
from(
select year, avg(star_rating)  as star_rating
from amazon_review.dmp
group by year
) as b
join
(select year, avg(star_rating) as star_rating
from amazon_review.sample_music
group by year
) as c
on b.year = c.year;
```

Output:

0.5806308546262009



```
hive> select corr(b.star_rating, c.star_rating)
    > from
    > (
    > select year, avg(star_rating)  as star_rating
    > from amazon_review.dmp
    > group by year
    > ) as b
    > join
    > (
    > select year, avg(star_rating) as star_rating
    > from amazon_review.sample_music
    > group by year
    > ) as c
    > on b.year = c.year;
Query ID = hadoop_20200413060056_f44b716b-17b6-4661-8ec3-deb9ef2c390c
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1586737734097_0021)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     11         11        0        0       0       0
Map 5 .......... container     SUCCEEDED     17         17        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      3          3        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      6          6        0        0       0       0
Reducer 4 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 6 ...... container     SUCCEEDED     10         10        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 06/06  [==========================>>] 100%  ELAPSED TIME: 39.87 s
----------------------------------------------------------------------------------------------
OK
0.5806308546262009
Time taken: 40.532 seconds, Fetched: 1 row(s)
hive>
```

Based on the results, we can say that they are positively correlated.

2. Are there same users reviewing in both categories?

We create a new table called ‘final’ which contains all the records with product category as Music or Digital Music Purchase and which contains the same customer ids in both tables. We attempt to get the number of users who reviewed products for both the categories.

Query:

`create table amazon_review.final as`

`(select a.*`

`from sample_music a`

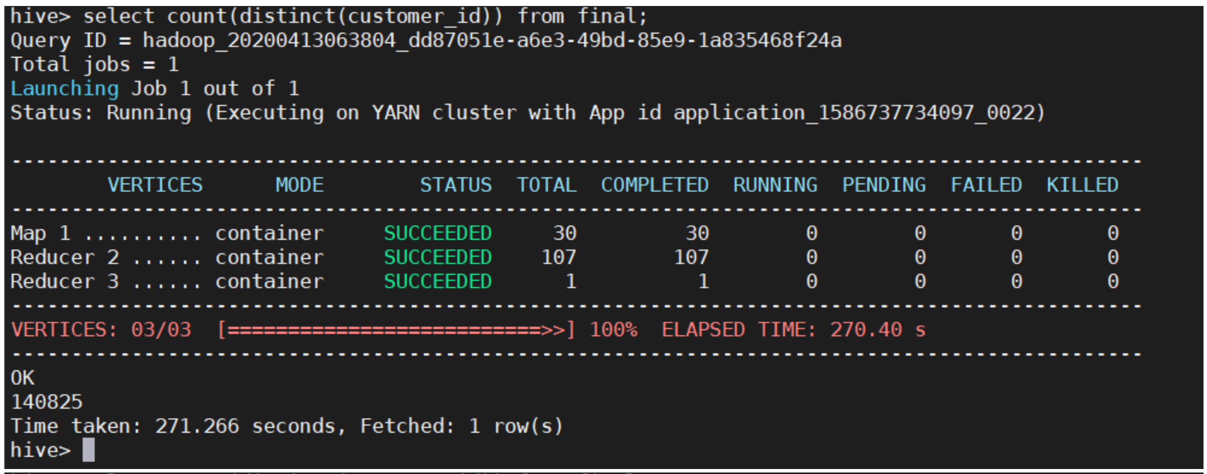
`join dmp b`

`on a.customer_id = b.customer_id) ;`

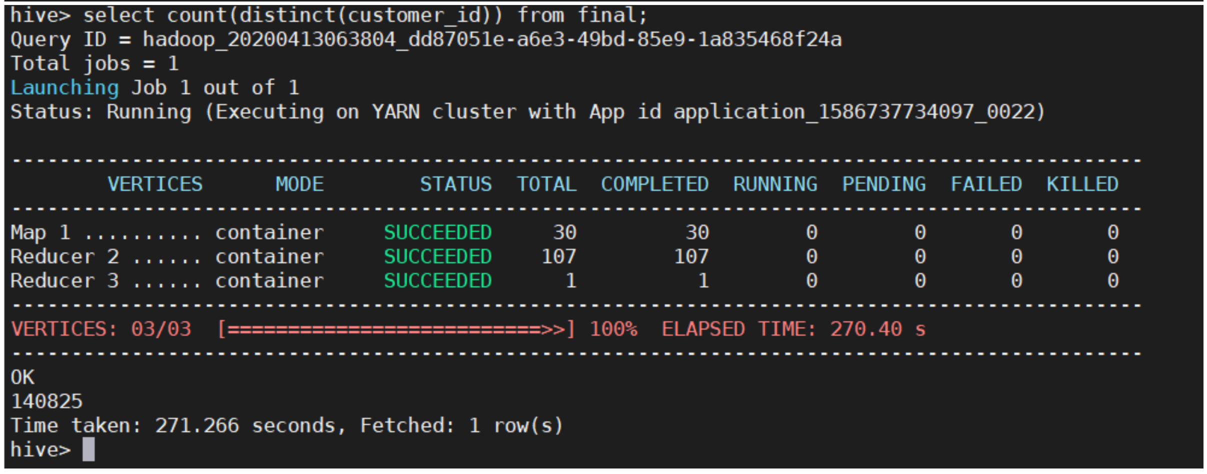
Select count(distinct(customer_id)) from final; -- 140825

Output:

There are 140825 distinct users who have reviewed products from both categories.



```
hive> select count(distinct(customer_id)) from final;
Query ID = hadoop_20200413063804_dd87051e-a6e3-49bd-85e9-1a835468f24a
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1586737734097_0022)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     30         30        0        0       0       0
Reducer 2 ...... container     SUCCEEDED    107        107        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 270.40 s
----------------------------------------------------------------------------------------------
OK
140825
Time taken: 271.266 seconds, Fetched: 1 row(s)
hive>
```



```
hive> select count(distinct(customer_id)) from final;
Query ID = hadoop_20200413063804_dd87051e-a6e3-49bd-85e9-1a835468f24a
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1586737734097_0022)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     30         30        0        0       0       0
Reducer 2 ...... container     SUCCEEDED    107        107        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 270.40 s
----------------------------------------------------------------------------------------------
OK
140825
Time taken: 271.266 seconds, Fetched: 1 row(s)
hive>
```

3. Can you identify similar items in both categories? Do they get same rating?

To identify similar items for both categories, we compare their product ids and check if we have anything in common. We then display their star ratings and see that the star ratings for these products are different.

Query:

`select product_id from dmp`

`Intersect`

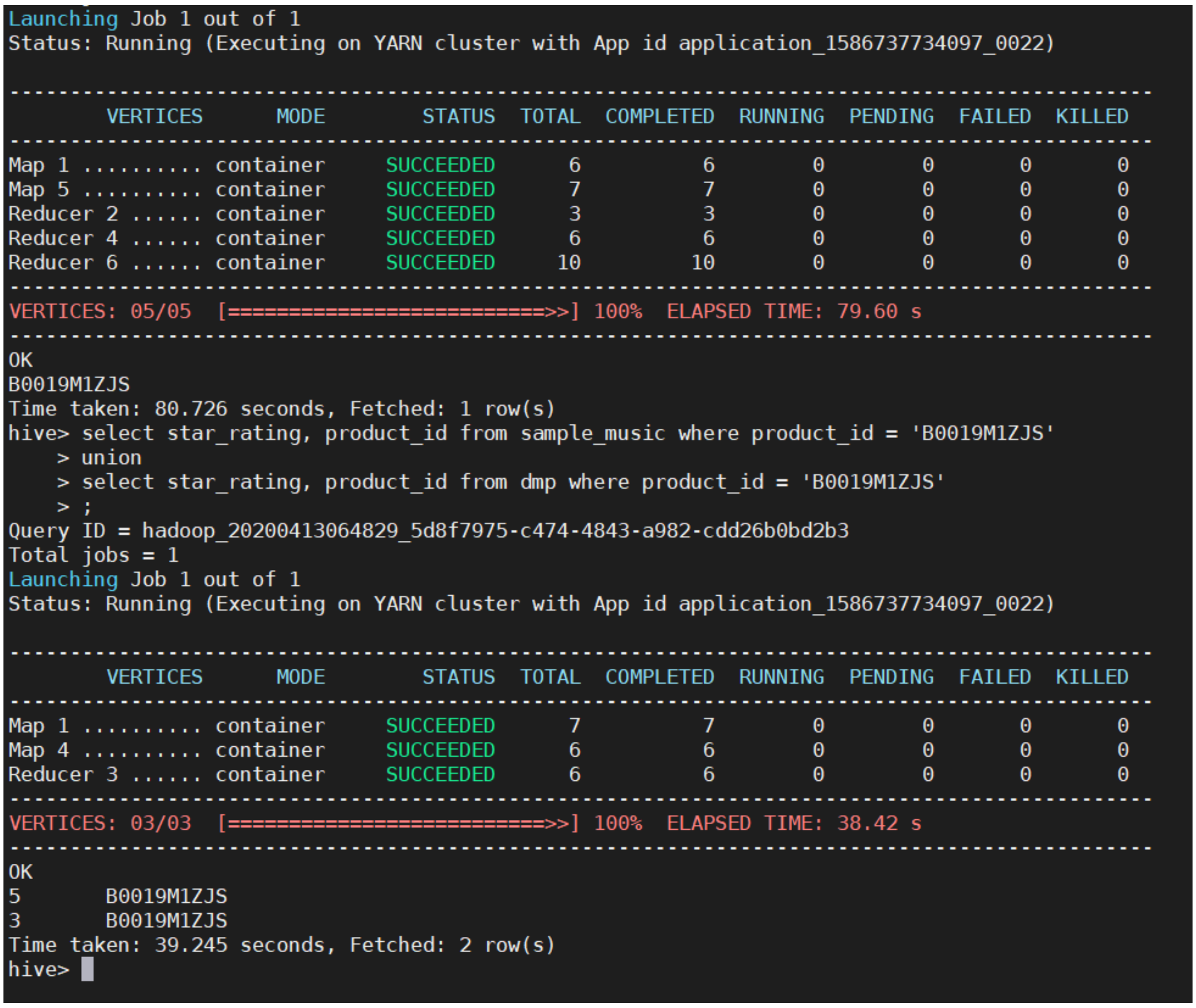
`select product_id from sample_music`

`select star_rating, product_id from sample_music where product_id = 'B0019M1ZJS'`

`union`

`select star_rating, product_id from dmp where product_id = 'B0019M1ZJS'`

Output:


```
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1586737734097_0022)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      6          6        0        0       0       0
Map 5 .......... container     SUCCEEDED      7          7        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      3          3        0        0       0       0
Reducer 4 ...... container     SUCCEEDED      6          6        0        0       0       0
Reducer 6 ...... container     SUCCEEDED     10         10        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 05/05  [==========================>>] 100%  ELAPSED TIME: 79.60 s
----------------------------------------------------------------------------------------------
OK
B0019M1ZJS
Time taken: 80.726 seconds, Fetched: 1 row(s)
hive> select star_rating, product_id from sample_music where product_id = 'B0019M1ZJS'
    > union
    > select star_rating, product_id from dmp where product_id = 'B0019M1ZJS'
    > ;
Query ID = hadoop_20200413064829_5d8f7975-c474-4843-a982-cdd26b0bd2b3
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1586737734097_0022)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      7          7        0        0       0       0
Map 4 .......... container     SUCCEEDED      6          6        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      6          6        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 38.42 s
----------------------------------------------------------------------------------------------
OK
5       B0019M1ZJS
3       B0019M1ZJS
Time taken: 39.245 seconds, Fetched: 2 row(s)
hive>
```

4. Comparing average star ratings for both the categories grouped by years

To compare the average star ratings for both the categories in a given year starting from 2005.

Query:

`select year, product_category, avg(star_rating) as AvgStarRating`

`from sample_music`

`group by year, product_category`

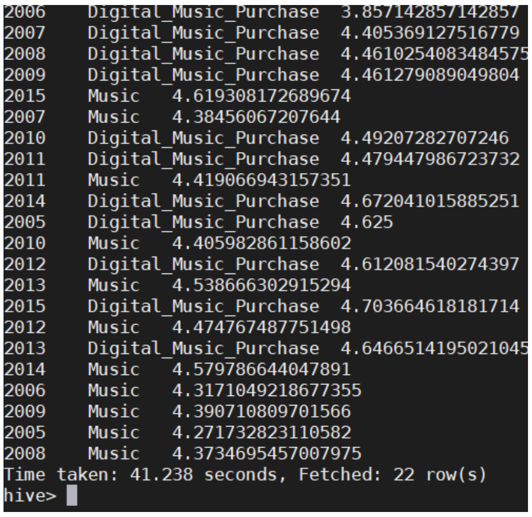
`union`

`select year, product_category, avg(star_rating) as AvgStarRtng`

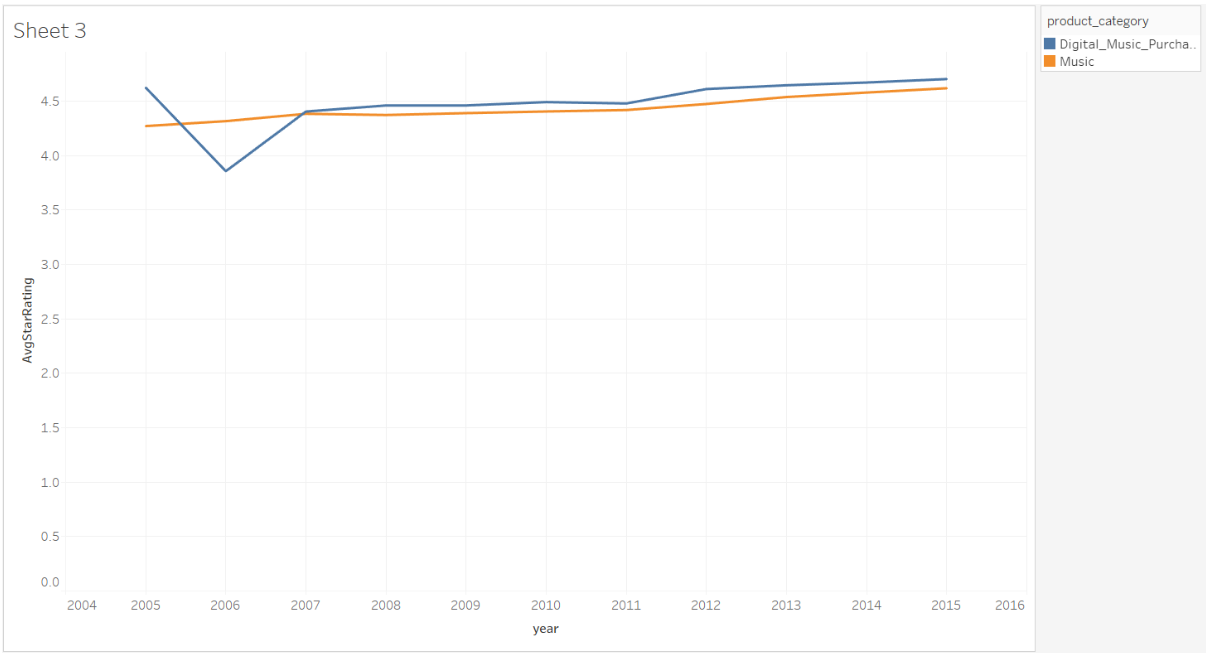
`from dmp`

`group by year, product_category`

Output:



```
2006    Digital_Music_Purchase  3.857142857142857
2007    Digital_Music_Purchase  4.405369127516779
2008    Digital_Music_Purchase  4.4610254083484575
2009    Digital_Music_Purchase  4.461279089049804
2015    Music   4.619308172689674
2007    Music   4.38456067207644
2010    Digital_Music_Purchase  4.49207282707246
2011    Digital_Music_Purchase  4.479447986723732
2011    Music   4.419066943157351
2014    Digital_Music_Purchase  4.672041015885251
2005    Digital_Music_Purchase  4.625
2010    Music   4.405982861158602
2012    Digital_Music_Purchase  4.612081540274397
2013    Music   4.538666302915294
2015    Digital_Music_Purchase  4.703664618181714
2012    Music   4.474767487751498
2013    Digital_Music_Purchase  4.6466514195021045
2014    Music   4.579786644047891
2006    Music   4.3171049218677355
2009    Music   4.390710809701566
2005    Music   4.271732823110582
2008    Music   4.3734695457007975
Time taken: 41.238 seconds, Fetched: 22 row(s)
hive> 
```



5. Comparing number of 5-star ratings with 4-star ratings for reviews for both categories grouped by year

To compare the no of 5-star and 4-star ratings for reviews from both categories in a given year. We then try to plot the results to find a trend in the way the star ratings change.

Query:

`select year, star_rating, product_category, count(*) as count_star_rating`

`from sample_music`

`where star_rating = 5`

`group by year, star_rating, product_category`

`union`

`select year, star_rating, product_category, count(*) as count_star_rating` `from sample_music`

`where star_rating = 4`

`group by year, star_rating, product_category`

`union`

`select year, star_rating, product_category, count(*) as count_star_rating`

`from dmp`

`where star_rating = 5`

`group by year, star_rating, product_category`

`union`

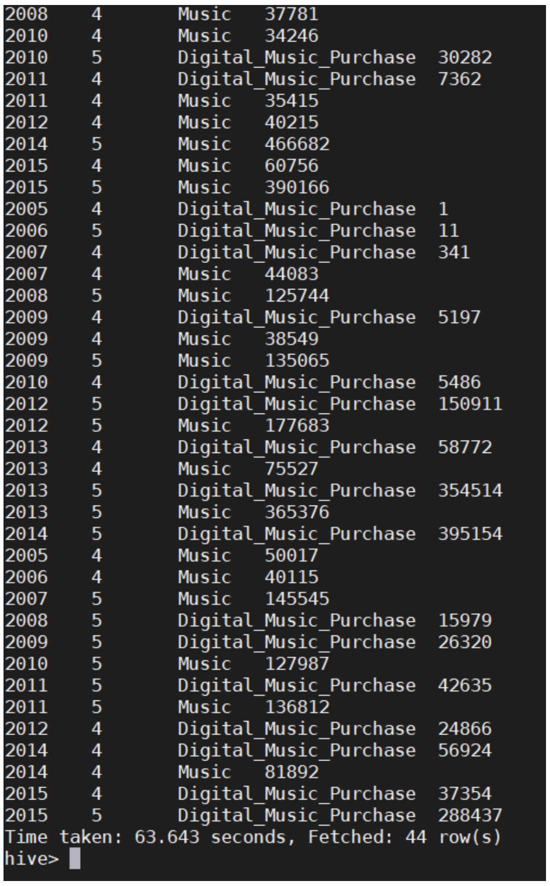
`select year, star_rating, product_category, count(*) as count_star_rating`

`from dmp`

`where star_rating = 4`

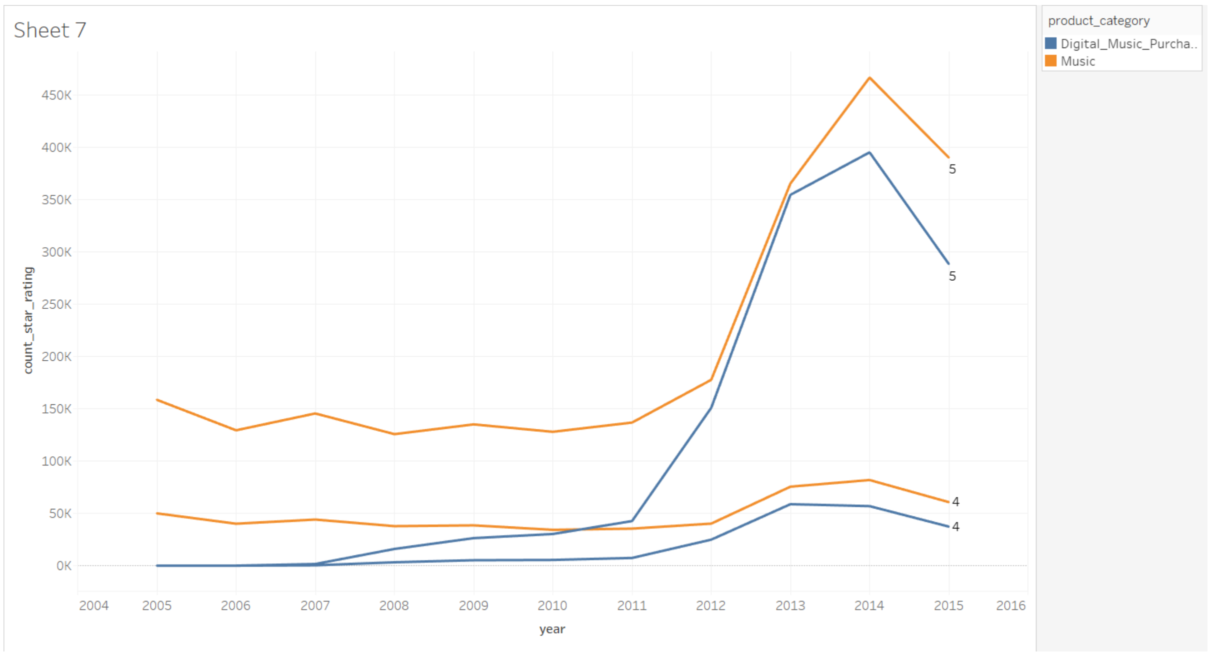
`group by year, star_rating, product_category`

Output:



```
2008    4       Music   37781
2010    4       Music   34246
2010    5       Digital_Music_Purchase  30282
2011    4       Digital_Music_Purchase  7362
2011    4       Music   35415
2012    4       Music   40215
2014    5       Music   466682
2015    4       Music   60756
2015    5       Music   390166
2005    4       Digital_Music_Purchase  1
2006    5       Digital_Music_Purchase  11
2007    4       Digital_Music_Purchase  341
2007    4       Music   44083
2008    5       Music   125744
2009    4       Digital_Music_Purchase  5197
2009    4       Music   38549
2009    5       Music   135065
2010    4       Digital_Music_Purchase  5486
2012    5       Digital_Music_Purchase  150911
2012    5       Music   177683
2013    4       Digital_Music_Purchase  58772
2013    4       Music   75527
2013    5       Digital_Music_Purchase  354514
2013    5       Music   365376
2014    5       Digital_Music_Purchase  395154
2005    4       Music   50017
2006    4       Music   40115
2007    5       Music   145545
2008    5       Digital_Music_Purchase  15979
2009    5       Digital_Music_Purchase  26320
2010    5       Music   127987
2011    5       Digital_Music_Purchase  42635
2011    5       Music   136812
2012    4       Digital_Music_Purchase  24866
2014    4       Digital_Music_Purchase  56924
2014    4       Music   81892
2015    4       Digital_Music_Purchase  37354
2015    5       Digital_Music_Purchase  288437
Time taken: 63.643 seconds, Fetched: 44 row(s)
hive>
```

Displaying only the last 20 lines



-- Digital_Video_Games/Video_Games over time.

We created two separate tables for product categories 'Video_Games and 'Digital_Video_Games' as it becomes easier to compare the two categories and answer questions based on it.

Query:

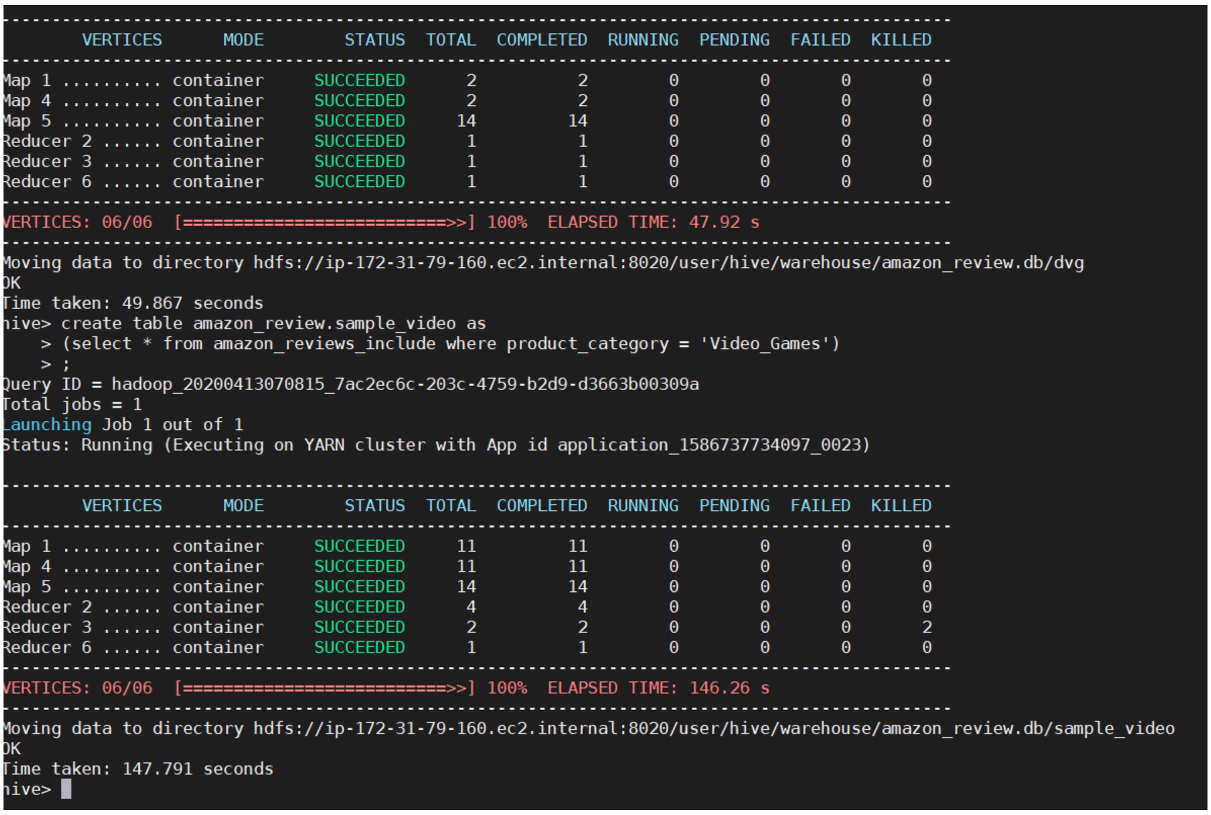
create table amazon_review.dvg as

(select * from amazon_reviews_include where product_category = 'Digital_Video_Games') with data;

create table amazon_review.sample_video as

(select * from amazon_reviews_include where product_category = 'Video_Games') with data;

Output:



```
----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      2          2        0        0       0       0
Map 4 .......... container     SUCCEEDED      2          2        0        0       0       0
Map 5 .......... container     SUCCEEDED     14         14        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 6 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 06/06  [==========================>>] 100%  ELAPSED TIME: 47.92 s
----------------------------------------------------------------------------------------------
Moving data to directory hdfs://ip-172-31-79-160.ec2.internal:8020/user/hive/warehouse/amazon_review.db/dvg
OK
Time taken: 49.867 seconds
hive> create table amazon_review.sample_video as
    > (select * from amazon_reviews_include where product_category = 'Video_Games')
    > ;
Query ID = hadoop_20200413070815_7ac2ec6c-203c-4759-b2d9-d3663b00309a
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1586737734097_0023)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     11         11        0        0       0       0
Map 4 .......... container     SUCCEEDED     11         11        0        0       0       0
Map 5 .......... container     SUCCEEDED     14         14        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      4          4        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      2          2        0        0       0       2
Reducer 6 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 06/06  [==========================>>] 100%  ELAPSED TIME: 146.26 s
----------------------------------------------------------------------------------------------
Moving data to directory hdfs://ip-172-31-79-160.ec2.internal:8020/user/hive/warehouse/amazon_review.db/sample_video
OK
Time taken: 147.791 seconds
hive>
```

1. Do you see correlation (maybe negative) between the categories over time?

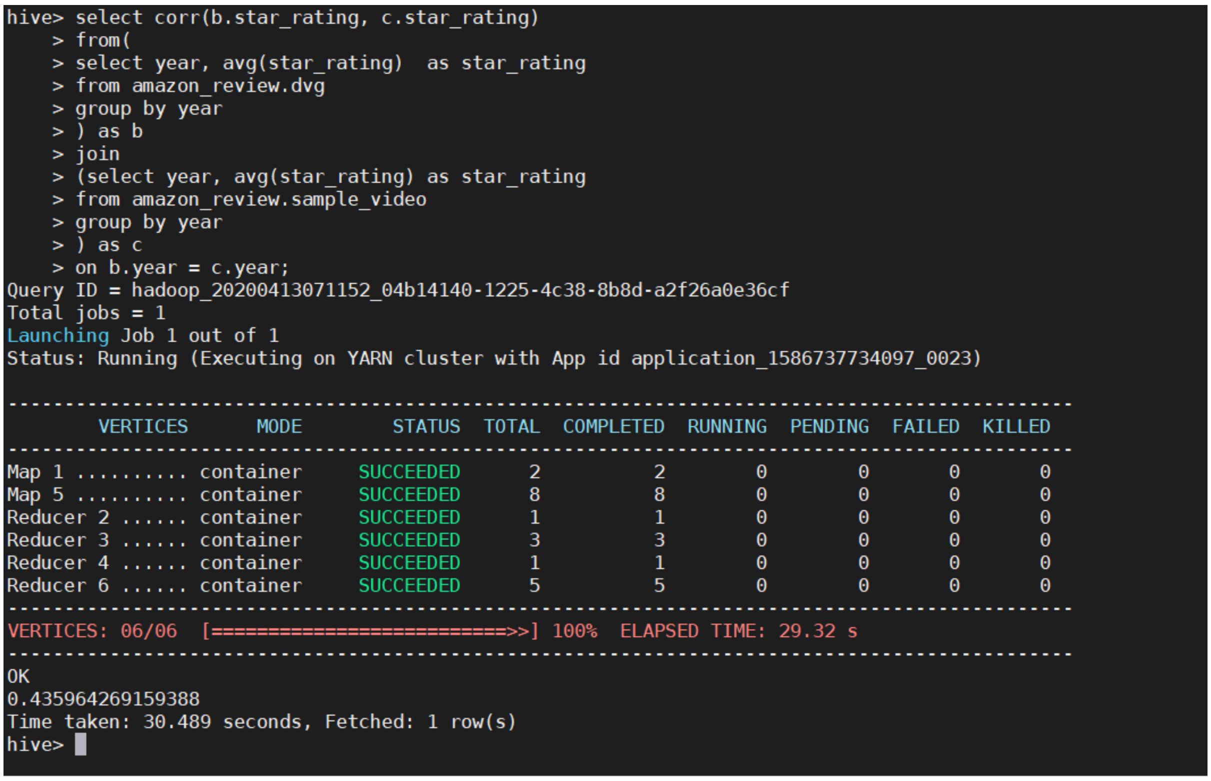
We attempt to correlate the star ratings of both product categories and use corr() function to determine the correlation coefficient.

Query:

```
select corr(b.star_rating, c.star_rating)

from(

select year, avg(star_rating)  as star_rating

from amazon_review.dvg

group by year

) as b

join

(select year, avg(star_rating) as star_rating

from amazon_review.sample_video

group by year

) as c

on b.year = c.year;
```

Output:

0.435964269159388



```
hive> select corr(b.star_rating, c.star_rating)
    > from(
    > select year, avg(star_rating)  as star_rating
    > from amazon_review.dvg
    > group by year
    > ) as b
    > join
    > (select year, avg(star_rating) as star_rating
    > from amazon_review.sample_video
    > group by year
    > ) as c
    > on b.year = c.year;
Query ID = hadoop_20200413071152_04b14140-1225-4c38-8b8d-a2f26a0e36cf
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1586737734097_0023)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      2          2        0        0       0       0
Map 5 .......... container     SUCCEEDED      8          8        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      3          3        0        0       0       0
Reducer 4 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 6 ...... container     SUCCEEDED      5          5        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 06/06  [==========================>>] 100%  ELAPSED TIME: 29.32 s
----------------------------------------------------------------------------------------------
OK
0.435964269159388
Time taken: 30.489 seconds, Fetched: 1 row(s)
hive>
```

After observing the results, we can say that the star ratings for both the categories are positively correlated.

2. Are there same users reviewing in both categories?

We create a new table called ‘final2’ which contains all the records with product category as Video_Games or Digital_Video_Games and which contains the same customer ids in both tables. We attempt to get the number of users who reviewed products for both the categories.

Query:

create table amazon_review.final2 as

(select a.*

from dvg a

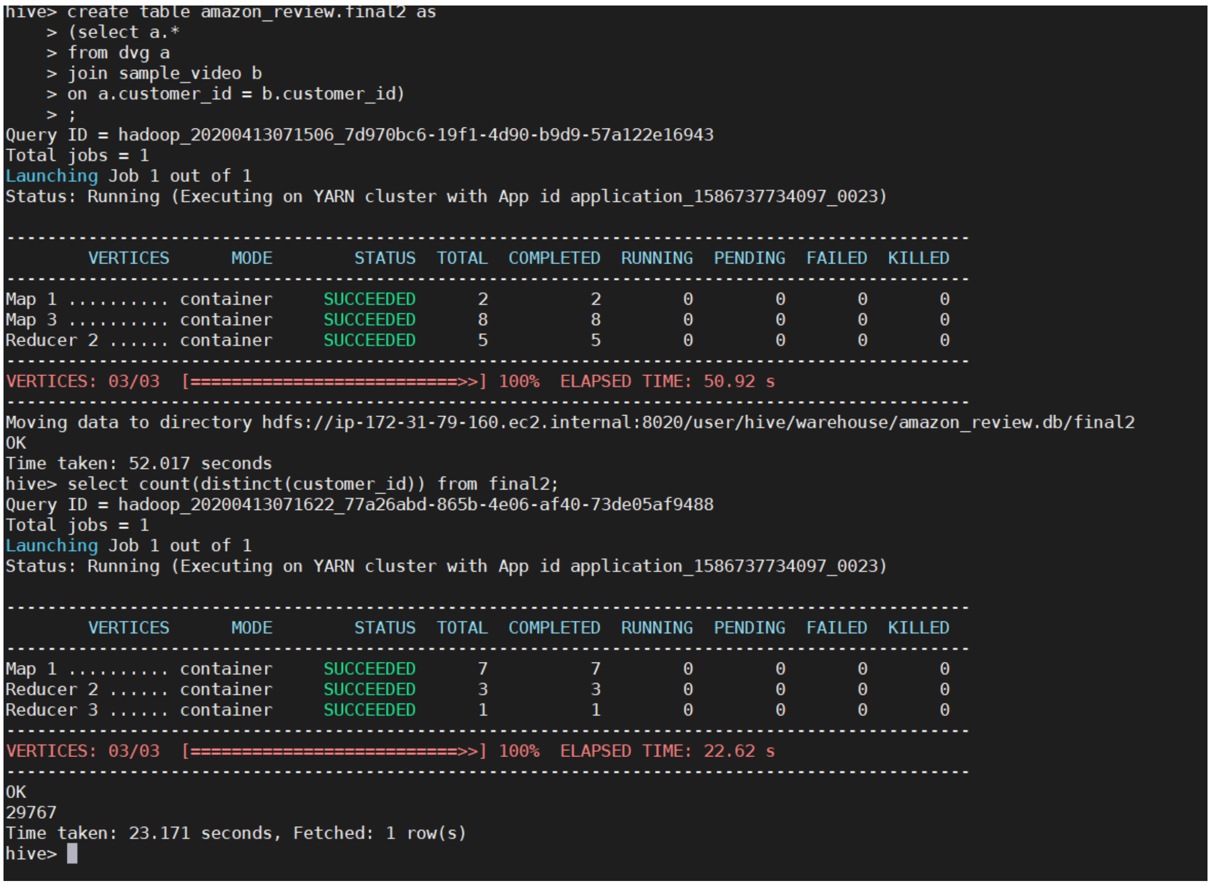
join sample_video b

on a.customer_id = b.customer_id);

select count(distinct(customer_id)) from final2 -- 29767

Output:

Number of users who reviewed the product for both categories are 29767



```
hive> create table amazon_review.final2 as
    > (select a.*
    > from dvg a
    > join sample_video b
    > on a.customer_id = b.customer_id)
    > ;
Query ID = hadoop_20200413071506_7d970bc6-19f1-4d90-b9d9-57a122e16943
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1586737734097_0023)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      2          2        0        0       0       0
Map 3 .......... container     SUCCEEDED      8          8        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      5          5        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 50.92 s
----------------------------------------------------------------------------------------------
Moving data to directory hdfs://ip-172-31-79-160.ec2.internal:8020/user/hive/warehouse/amazon_review.db/final2
OK
Time taken: 52.017 seconds
hive> select count(distinct(customer_id)) from final2;
Query ID = hadoop_20200413071622_77a26abd-865b-4e06-af40-73de05af9488
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1586737734097_0023)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      7          7        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      3          3        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 22.62 s
----------------------------------------------------------------------------------------------
OK
29767
Time taken: 23.171 seconds, Fetched: 1 row(s)
hive>
```

3. Can you identify similar items in both categories? Do they get same rating?

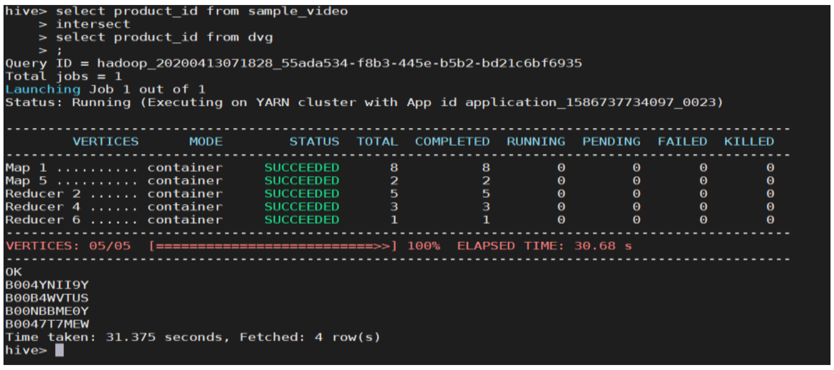
To identify similar items for both categories, we compare their product ids and check if we have anything in common. We then display their star ratings and see that the star ratings for these products are different.

Query:

select product_id from sample_video

intersect

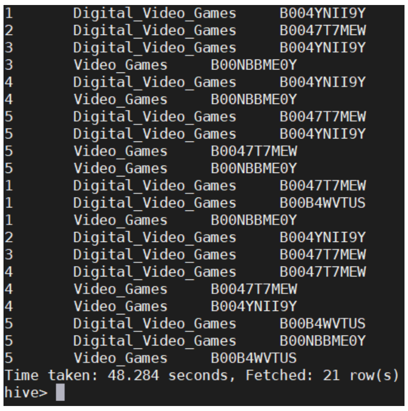
select product_id from dvg



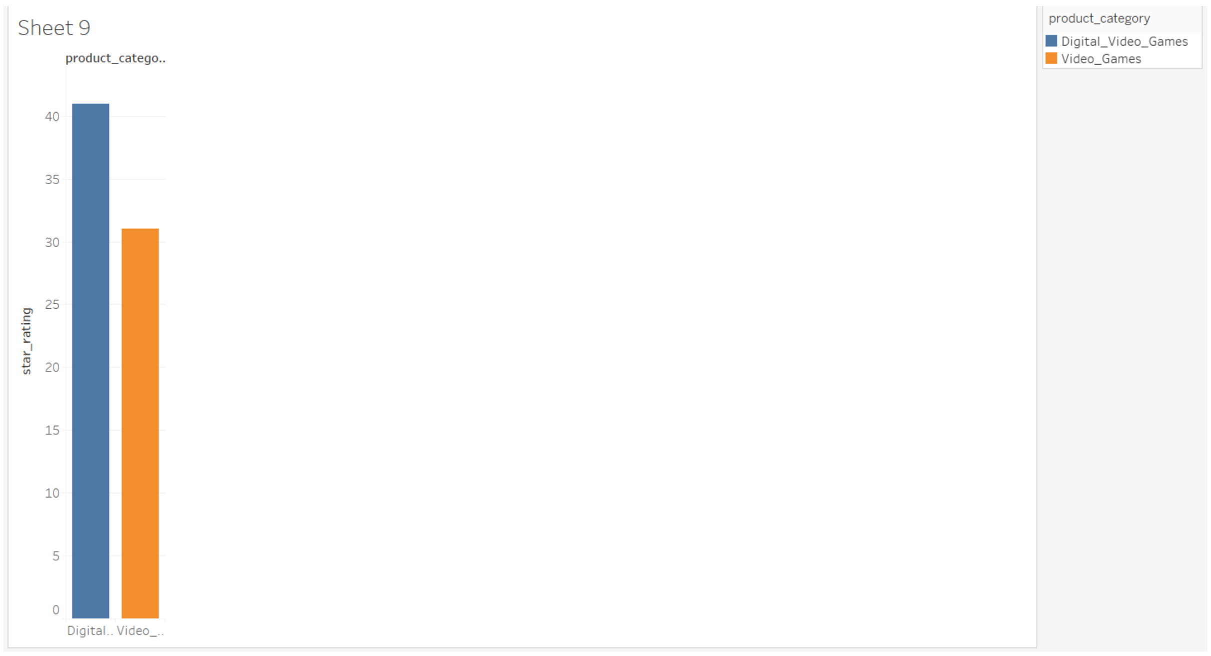
```
hive> select product_id from sample_video
    > intersect
    > select product_id from dvg
    > ;
Query ID = hadoop_20200413071828_55ada534-f8b3-445e-b5b2-bd21c6bf6935
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1586737734097_0023)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      8          8        0        0       0       0
Map 5 .......... container     SUCCEEDED      2          2        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      5          5        0        0       0       0
Reducer 4 ...... container     SUCCEEDED      3          3        0        0       0       0
Reducer 6 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 05/05  [==========================>>] 100%  ELAPSED TIME: 30.68 s
----------------------------------------------------------------------------------------------
OK
B004YNII9Y
B00B4WVTUS
B00NBBME0Y
B0047T7MEW
Time taken: 31.375 seconds, Fetched: 4 row(s)
hive>
```

```
select star_rating, product_category, product_id
from sample_video a
where product_id = 'B00B4WVTUS'
union
select star_rating, product_category, product_id
from dvg
where product_id = 'B00B4WVTUS'
union
select star_rating, product_category, product_id
from sample_video a
where product_id = 'B0047T7MEW'
union
select star_rating, product_category, product_id
from dvg
where product_id = 'B0047T7MEW'
union
select star_rating, product_category, product_id
from sample_video a
where product_id = 'B00NBBME0Y'
union
select star_rating, product_category, product_id
from dvg
where product_id = 'B00NBBME0Y'
union
select star_rating, product_category, product_id
from sample_video a
where product_id = 'B004YNII9Y'
union
select star_rating, product_category, product_id
from dvg
where product_id = 'B004YNII9Y'
```

Output:



```
1       Digital_Video_Games     B004YNII9Y
2       Digital_Video_Games     B0047T7MEW
3       Digital_Video_Games     B004YNII9Y
3       Video_Games     B00NBBME0Y
4       Digital_Video_Games     B004YNII9Y
4       Video_Games     B00NBBME0Y
5       Digital_Video_Games     B0047T7MEW
5       Digital_Video_Games     B004YNII9Y
5       Video_Games     B0047T7MEW
5       Video_Games     B00NBBME0Y
1       Digital_Video_Games     B0047T7MEW
1       Digital_Video_Games     B00B4WVTUS
1       Video_Games     B00NBBME0Y
2       Digital_Video_Games     B004YNII9Y
3       Digital_Video_Games     B0047T7MEW
4       Digital_Video_Games     B0047T7MEW
4       Video_Games     B0047T7MEW
4       Video_Games     B004YNII9Y
5       Digital_Video_Games     B00B4WVTUS
5       Digital_Video_Games     B00NBBME0Y
5       Video_Games     B00B4WVTUS
Time taken: 48.284 seconds, Fetched: 21 row(s)
hive>
```



4. Comparing average star ratings for both the categories grouped by years

To compare the average star ratings for both the categories in a given year starting from 2005.

Query:

select year, product_category, avg(star_rating) as AvgStarRating

from dvg

group by year, product_category

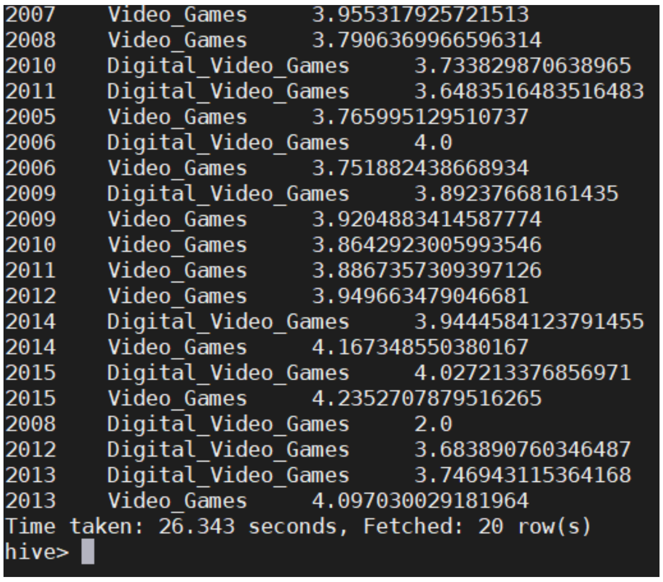
union

select year, product_category, avg(star_rating) as AvgStarRtng

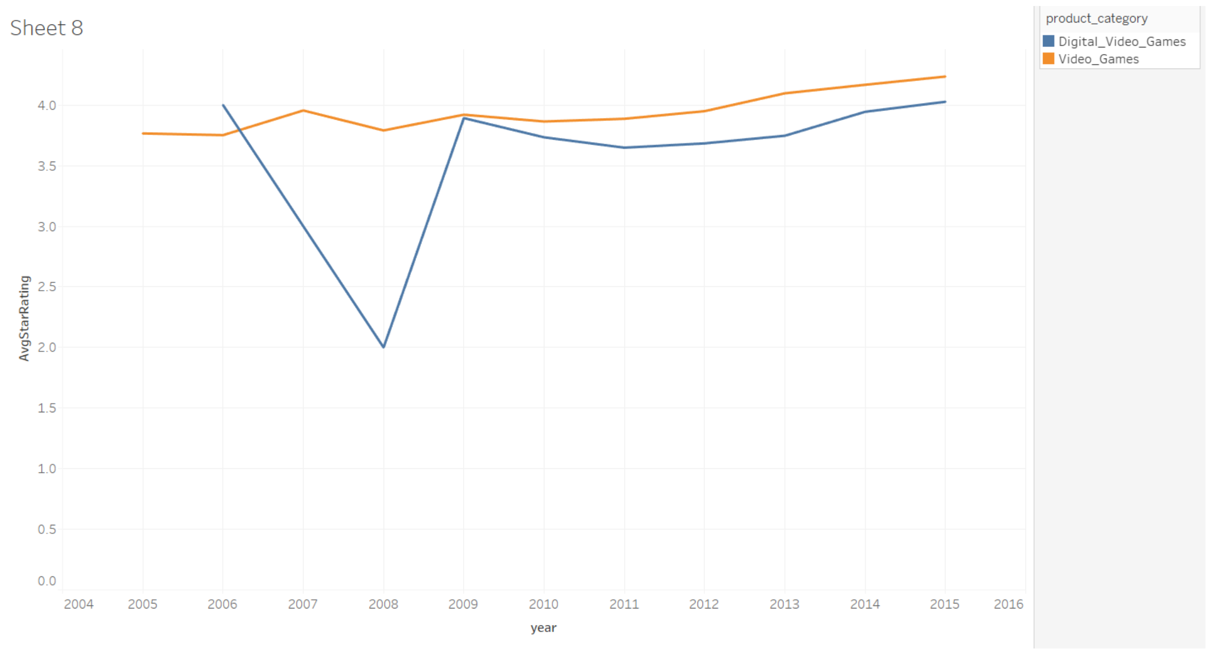
from sample_video

group by year, product_category

Output:



```
2007    Video_Games     3.955317925721513
2008    Video_Games     3.7906369966596314
2010    Digital_Video_Games     3.733829870638965
2011    Digital_Video_Games     3.6483516483516483
2005    Video_Games     3.765995129510737
2006    Digital_Video_Games     4.0
2006    Video_Games     3.751882438668934
2009    Digital_Video_Games     3.89237668161435
2009    Video_Games     3.9204883414587774
2010    Video_Games     3.8642923005993546
2011    Video_Games     3.8867357309397126
2012    Video_Games     3.949663479046681
2014    Digital_Video_Games     3.9444584123791455
2014    Video_Games     4.167348550380167
2015    Digital_Video_Games     4.027213376856971
2015    Video_Games     4.2352707879516265
2008    Digital_Video_Games     2.0
2012    Digital_Video_Games     3.683890760346487
2013    Digital_Video_Games     3.746943115364168
2013    Video_Games     4.097030029181964
Time taken: 26.343 seconds, Fetched: 20 row(s)
hive>
```



5. Comparing number of 5-star ratings with 4-star ratings for reviews for both categories grouped by year

To compare the number of 5-star and 4-star ratings for reviews from both categories in a given year. We then try to plot the results to find a trend in the way the star ratings change.

Query:

select year, star_rating, product_category, count(*) as count_star_rating

from dvg

where star_rating = 5

group by year, star_rating, product_category union

select year, star_rating, product_category, count(*) as count_star_rating

from dvg

where star_rating = 4

group by year, star_rating, product_category

union

select year, star_rating, product_category, count(*) as count_star_rating

from sample_video

where star_rating = 5

group by year, star_rating, product_category

union

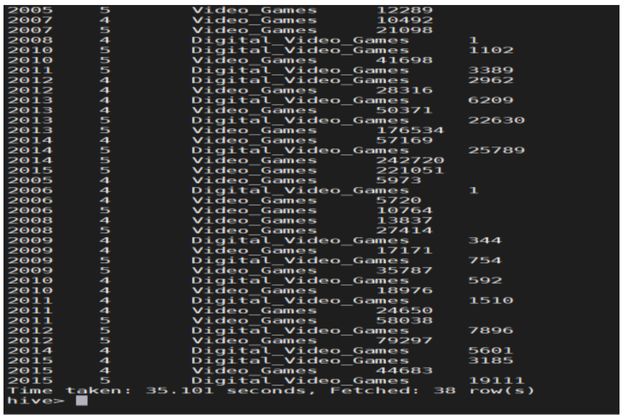
select year, star_rating, product_category, count(*) as count_star_rating

from sample_video

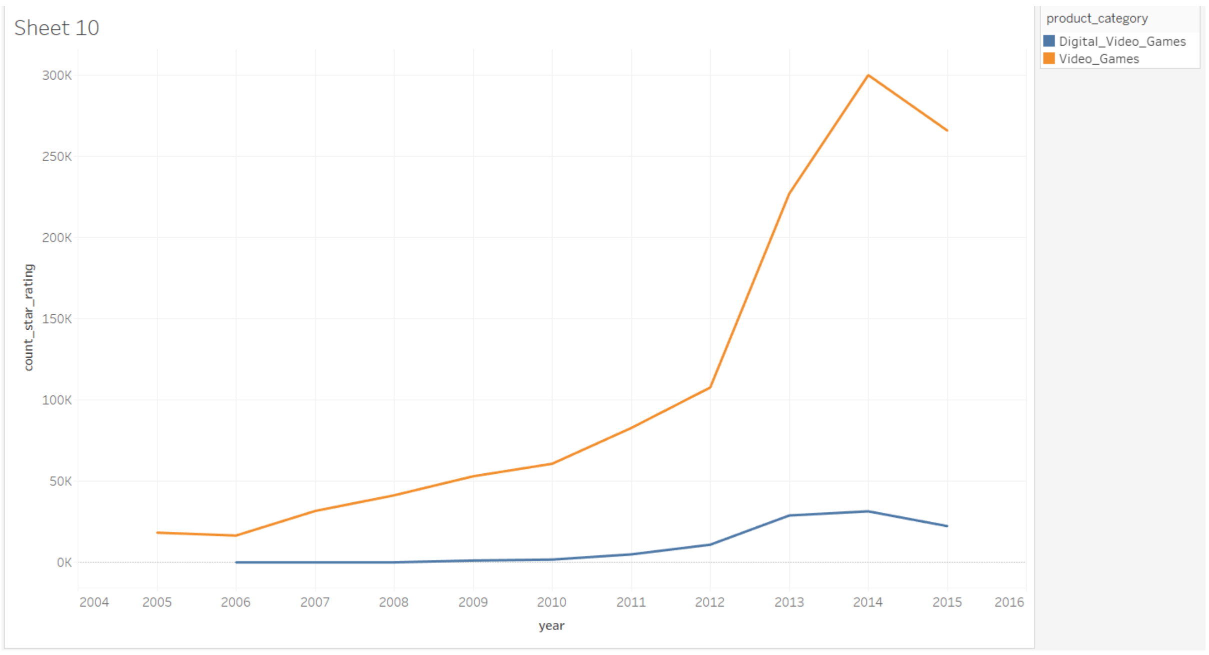
where star_rating = 4

group by year, star_rating, product_category

Output:



```
2005    5       Video_Games     12289
2007    4       Video_Games     10492
2007    5       Video_Games     21098
2008    4       Digital_Video_Games     1
2010    5       Digital_Video_Games     1102
2010    5       Video_Games     41698
2011    5       Digital_Video_Games     3389
2012    4       Digital_Video_Games     2962
2012    4       Video_Games     28316
2013    4       Digital_Video_Games     6209
2013    4       Video_Games     50371
2013    5       Digital_Video_Games     22630
2013    5       Video_Games     176534
2014    4       Video_Games     57169
2014    5       Digital_Video_Games     25789
2014    5       Video_Games     242720
2015    5       Video_Games     221051
2005    4       Video_Games     5973
2006    4       Digital_Video_Games     1
2006    4       Video_Games     5720
2006    5       Video_Games     10764
2008    4       Video_Games     13837
2008    5       Video_Games     27414
2009    4       Digital_Video_Games     344
2009    4       Video_Games     17171
2009    5       Digital_Video_Games     754
2009    5       Video_Games     35787
2010    4       Digital_Video_Games     592
2010    4       Video_Games     18976
2011    4       Digital_Video_Games     1510
2011    4       Video_Games     24650
2011    5       Video_Games     58038
2012    5       Digital_Video_Games     7896
2012    5       Video_Games     79297
2014    4       Digital_Video_Games     5601
2015    4       Digital_Video_Games     3185
2015    4       Video_Games     44683
2015    5       Digital_Video_Games     19111
Time taken: 35.101 seconds, Fetched: 38 row(s)
hive>
```



3. You should demonstrate your ability to use Hive advanced functions:

1. Window functions: moving average, rank, aggregation functions using relevant ordering and partitioning

Query:

-- moving average

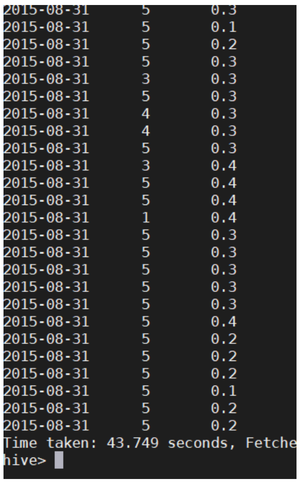
To calculate a 10 day moving average over average star ratings from sample_music table. We order the results by review_date.

Query:

select review_date, star_rating, avg(total_votes) over (order by review_date asc rows 9 PRECEDING) as MA10

from sample_music

Output:



```
2015-08-31      5       0.3
2015-08-31      5       0.1
2015-08-31      5       0.2
2015-08-31      5       0.3
2015-08-31      3       0.3
2015-08-31      5       0.3
2015-08-31      4       0.3
2015-08-31      4       0.3
2015-08-31      5       0.3
2015-08-31      3       0.4
2015-08-31      5       0.4
2015-08-31      5       0.4
2015-08-31      1       0.4
2015-08-31      5       0.3
2015-08-31      5       0.3
2015-08-31      5       0.3
2015-08-31      5       0.3
2015-08-31      5       0.3
2015-08-31      5       0.4
2015-08-31      5       0.2
2015-08-31      5       0.2
2015-08-31      5       0.2
2015-08-31      5       0.1
2015-08-31      5       0.2
2015-08-31      5       0.2
Time taken: 43.749 seconds, Fetche
hive>
```

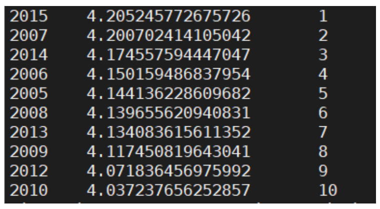
-- rank

To use rank() function to rank the average star ratings grouped by years. We display the first ten ranks.

Query:

```
select a.* from

(select year, avg(star_rating) as AvgStarRtng, RANK() over (order by avg(star_rating) desc) as rank1

 from amazon_reviews_include

 group by year ) as a

 where rank1 <= 10;
```

Output:



```
2015    4.205245772675726       1
2007    4.200702414105042       2
2014    4.174557594447047       3
2006    4.150159486837954       4
2005    4.144136228609682       5
2008    4.139655620940831       6
2013    4.134083615611352       7
2009    4.117450819643041       8
2012    4.071836456975992       9
2010    4.037237656252857       10
```

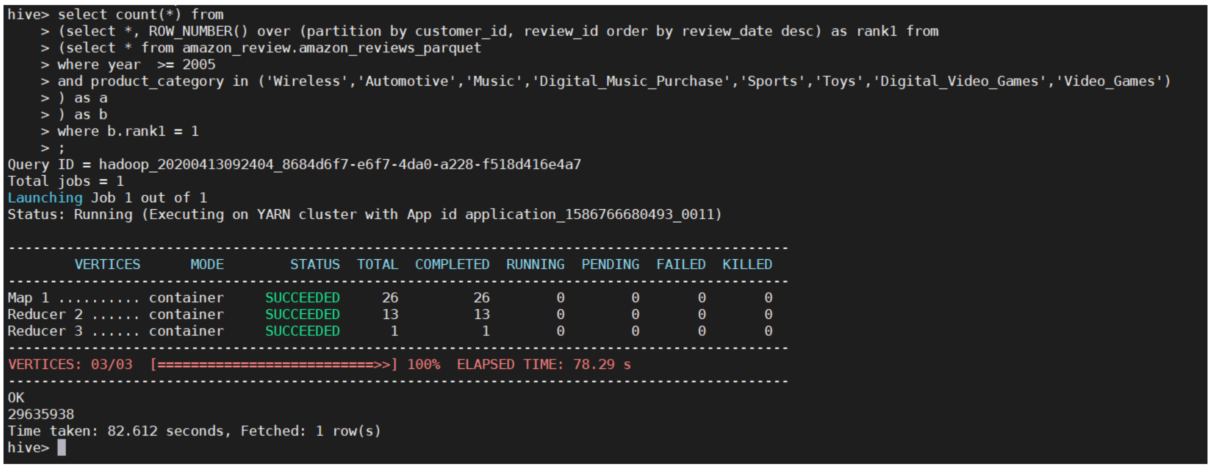
-- aggregate functions and partitioning

Use aggregate functions and partitioning to choose such customers with their most recent review and exclude multiple reviews by that customer. Also, apply basic filters based on our requirement like starting our analysis from 2005 and over a few selected product categories.

Query:

```
select count(*) from

(select *, ROW_NUMBER() over (partition by customer_id, review_id order by review_date desc) as rank1 from

(select * from amazon_review.amazon_reviews_parquet

where year  >= 2005

and product_category in ('Wireless','Automotive','Music','Digital_Music_Purchase','Sports','Toys','Digital_Video_Games','Video_Games')

) as a

) as b

where b.rank1 = 1
```

Output:


```
hive> select count(*) from
    > (select *, ROW_NUMBER() over (partition by customer_id, review_id order by review_date desc) as rank1 from
    > (select * from amazon_review.amazon_reviews_parquet
    > where year  >= 2005
    > and product_category in ('Wireless','Automotive','Music','Digital_Music_Purchase','Sports','Toys','Digital_Video_Games','Video_Games')
    > ) as a
    > ) as b
    > where b.rank1 = 1
    > ;
Query ID = hadoop_20200413092404_8684d6f7-e6f7-4da0-a228-f518d416e4a7
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1586766680493_0011)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     26         26        0        0       0       0
Reducer 2 ...... container     SUCCEEDED     13         13        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 78.29 s
----------------------------------------------------------------------------------------------
OK
29635938
Time taken: 82.612 seconds, Fetched: 1 row(s)
hive>
```

2. Analytical Aggregate functions: percentile, min, max, average, standard deviation, correlation

-- percentile

To calculate the percentile rank of star_rating for all the categories grouped by year.

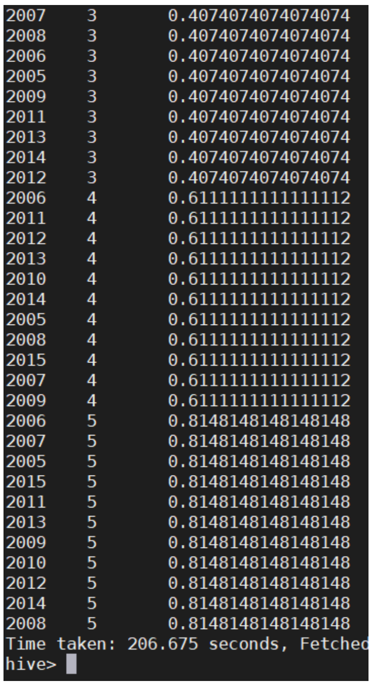
Query:

select year, star_rating, PERCENT_RANK() over (order by star_rating) as percent_rank

from amazon_reviews_include

group by year, star_rating

Output:


```
2007    3       0.4074074074074074
2008    3       0.4074074074074074
2006    3       0.4074074074074074
2005    3       0.4074074074074074
2009    3       0.4074074074074074
2011    3       0.4074074074074074
2013    3       0.4074074074074074
2014    3       0.4074074074074074
2012    3       0.4074074074074074
2006    4       0.6111111111111112
2011    4       0.6111111111111112
2012    4       0.6111111111111112
2013    4       0.6111111111111112
2010    4       0.6111111111111112
2014    4       0.6111111111111112
2005    4       0.6111111111111112
2008    4       0.6111111111111112
2015    4       0.6111111111111112
2007    4       0.6111111111111112
2009    4       0.6111111111111112
2006    5       0.8148148148148148
2007    5       0.8148148148148148
2005    5       0.8148148148148148
2015    5       0.8148148148148148
2011    5       0.8148148148148148
2013    5       0.8148148148148148
2009    5       0.8148148148148148
2010    5       0.8148148148148148
2012    5       0.8148148148148148
2014    5       0.8148148148148148
2008    5       0.8148148148148148
Time taken: 206.675 seconds, Fetched
hive>
```

-- min

To calculate minimum review_date.

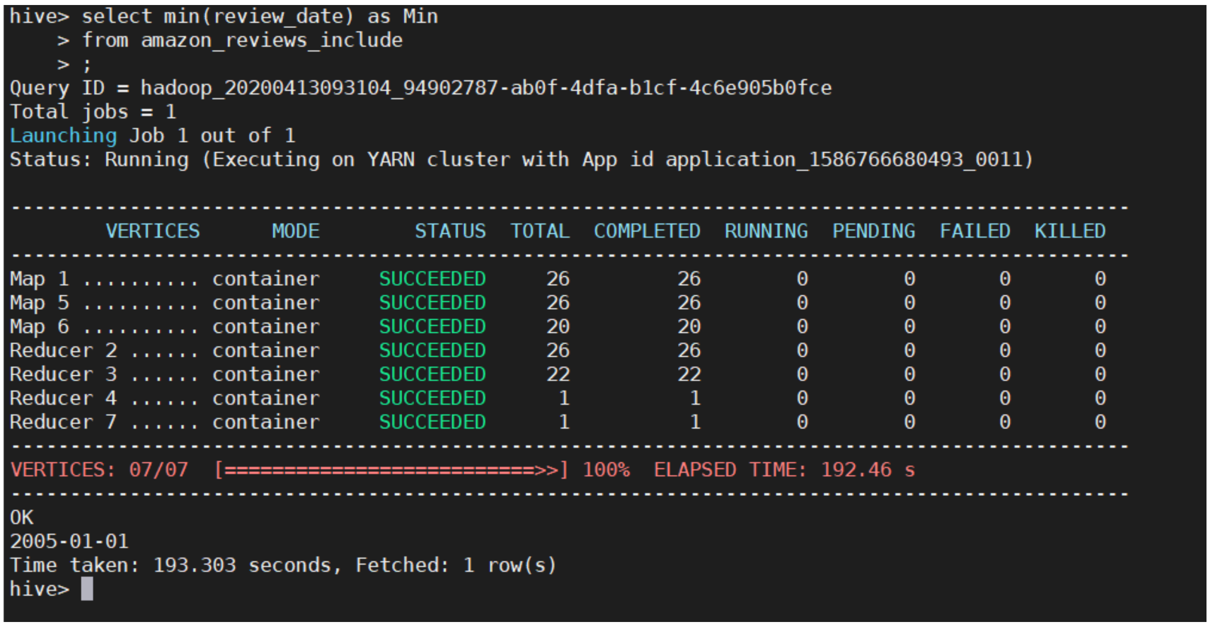
Query:

select min(review_date) as Min

from amazon_reviews_include

Output:

2005-01-01



```
hive> select min(review_date) as Min
    > from amazon_reviews_include
    > ;
Query ID = hadoop_20200413093104_94902787-ab0f-4dfa-b1cf-4c6e905b0fce
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1586766680493_0011)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     26         26        0        0       0       0
Map 5 .......... container     SUCCEEDED     26         26        0        0       0       0
Map 6 .......... container     SUCCEEDED     20         20        0        0       0       0
Reducer 2 ...... container     SUCCEEDED     26         26        0        0       0       0
Reducer 3 ...... container     SUCCEEDED     22         22        0        0       0       0
Reducer 4 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 7 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 07/07  [==========================>>] 100%  ELAPSED TIME: 192.46 s
----------------------------------------------------------------------------------------------
OK
2005-01-01
Time taken: 193.303 seconds, Fetched: 1 row(s)
hive>
```

-- max

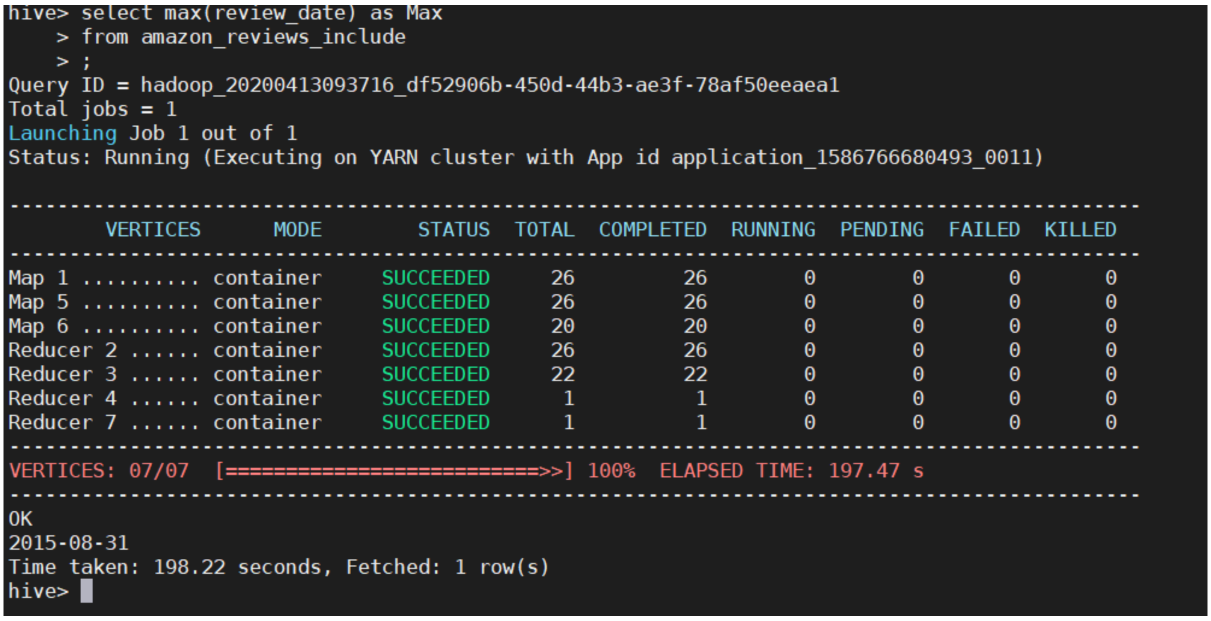
To calculate maximum review_date

Query:

select max(review_date) as Max

from amazon_reviews_include

Output:



```
hive> select max(review_date) as Max
    > from amazon_reviews_include
    > ;
Query ID = hadoop_20200413093716_df52906b-450d-44b3-ae3f-78af50eeaea1
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1586766680493_0011)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     26         26        0        0       0       0
Map 5 .......... container     SUCCEEDED     26         26        0        0       0       0
Map 6 .......... container     SUCCEEDED     20         20        0        0       0       0
Reducer 2 ...... container     SUCCEEDED     26         26        0        0       0       0
Reducer 3 ...... container     SUCCEEDED     22         22        0        0       0       0
Reducer 4 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 7 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 07/07  [==========================>>] 100%  ELAPSED TIME: 197.47 s
----------------------------------------------------------------------------------------------
OK
2015-08-31
Time taken: 198.22 seconds, Fetched: 1 row(s)
hive>
```

-- average

To calculate average star rating grouped by year.

Query:

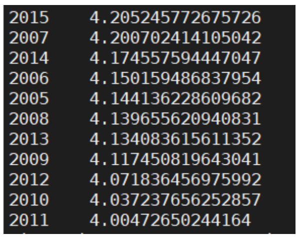
select year, avg(star_rating) as AVG_Rating

from amazon_reviews_include

group by year

order by AVG_Rating desc

Output:



```
2015    4.205245772675726
2007    4.200702414105042
2014    4.174557594447047
2006    4.150159486837954
2005    4.144136228609682
2008    4.139655620940831
2013    4.134083615611352
2009    4.117450819643041
2012    4.071836456975992
2010    4.037237656252857
2011    4.00472650244164
```

-- correlation

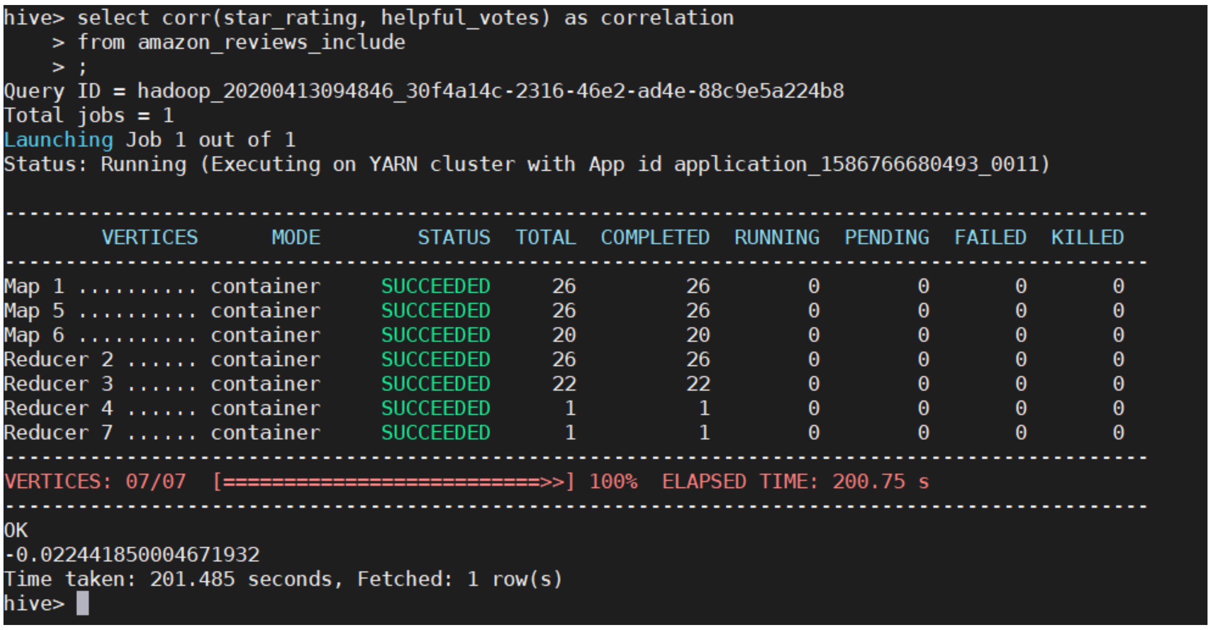
To calculate correlation between star rating and helpful votes.

Query:

select corr(star_rating, helpful_votes) as correlation

from amazon_reviews_include

Output:



```
hive> select corr(star_rating, helpful_votes) as correlation
    > from amazon_reviews_include
    > ;
Query ID = hadoop_20200413094846_30f4a14c-2316-46e2-ad4e-88c9e5a224b8
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1586766680493_0011)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     26         26        0        0       0       0
Map 5 .......... container     SUCCEEDED     26         26        0        0       0       0
Map 6 .......... container     SUCCEEDED     20         20        0        0       0       0
Reducer 2 ...... container     SUCCEEDED     26         26        0        0       0       0
Reducer 3 ...... container     SUCCEEDED     22         22        0        0       0       0
Reducer 4 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 7 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 07/07  [==========================>>] 100%  ELAPSED TIME: 200.75 s
----------------------------------------------------------------------------------------------
OK
-0.022441850004671932
Time taken: 201.485 seconds, Fetched: 1 row(s)
hive>
```

CONCLUSION

We were able to make use of big data technologies to analyse the amazon reviews dataset, answer data exploratory questions, compare product categories and observe trends in metrics over time. We were successfully able to perform exploratory data analysis on AWS Athena and perform queries in Hive over the amazon reviews external table in HDFS. We used numerous aggregation functions over numerical columns like star rating and total votes to help us group it with other metrics and observe different trends according to the product categories. When we were trying to compare the average star ratings for product categories 'Video Games' and 'Digital Video Games', the average dropped drastically for digital video games around 2008 and then resurged but was still less than the almost consistent average star ratings for video games. This just demonstrates one of our findings. To conclude, we were able to perform detailed analysis on the amazon review dataset and we made use of hive queries to obtain results.

REFERENCES

https://piazza.com/class

https://aws.amazon.com/emr/

https://aws.amazon.com/emr/features/hive/

http://www1.udel.edu/evelyn/SQL-Class3/SQL3_Stat.html