APRIL 12, 2020

PROJECT REPORT

PART 1

CHINMAY SUNIL KARANDIKAR
BIG DATA MIS 6346.502
THE UNIVERSITY OF TEXAS AT DALLAS

Table of contents

Setting up the project and loading data

```
create database amazon_review;

drop table amazon_review.amazon_reviews_parquet;


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;


create view temporary

as

select * from amazon_review.amazon_reviews_parquet where review_id in (select review_id from
(select customer_id, product_id,review_id,count(*)

from amazon_review.amazon_reviews_parquet

group by customer_id, product_id,review_id

having count(*)=1) as t) and product_category in
('Wireless','Automotive','Music','Digital_Music_Purchase','Sports','Toys','Digital_Video_Games','Video_G
ames');
```

Creating table to filter reviews that are reviewed multiple times by same customer for same product.

```
create table amazon_review.filtered_reviews
AS
select z.* from(
select *,row_number() over(partition by customer_id,product_id) as row1 from temporary)z where
row1=1;
```

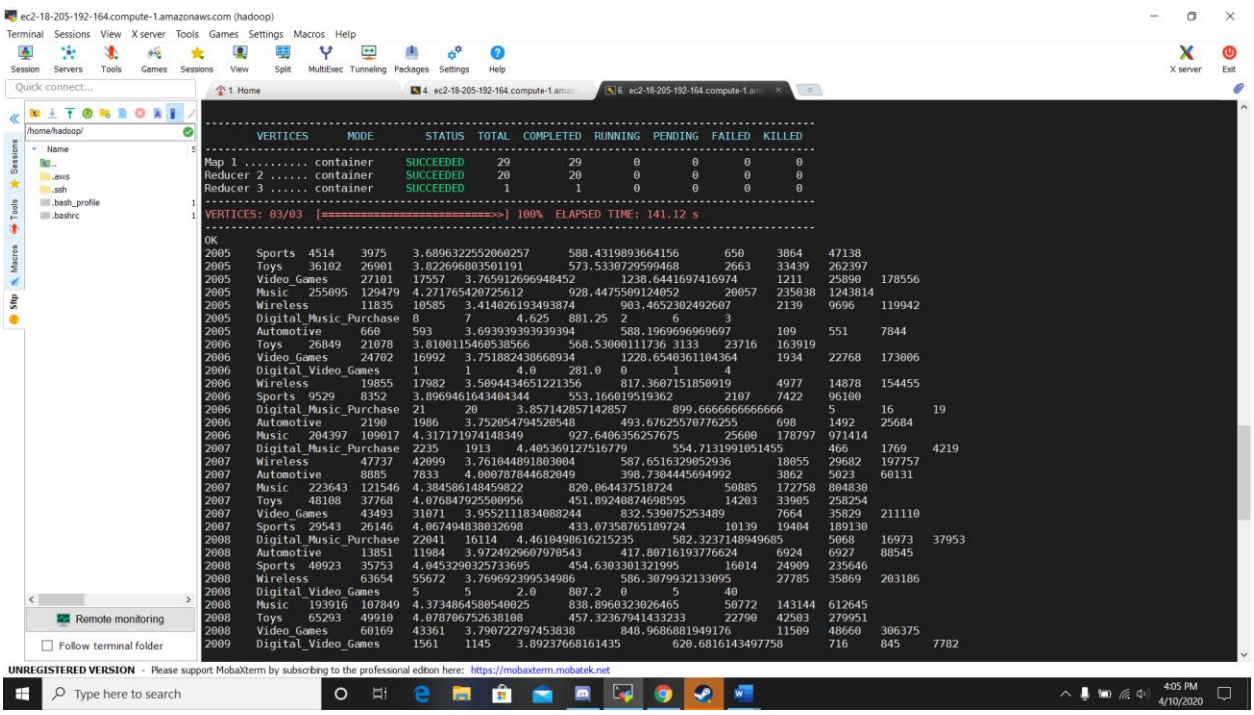
Question 1

Explore the dataset and provide basic exploratory analysis over time and per product category

Query

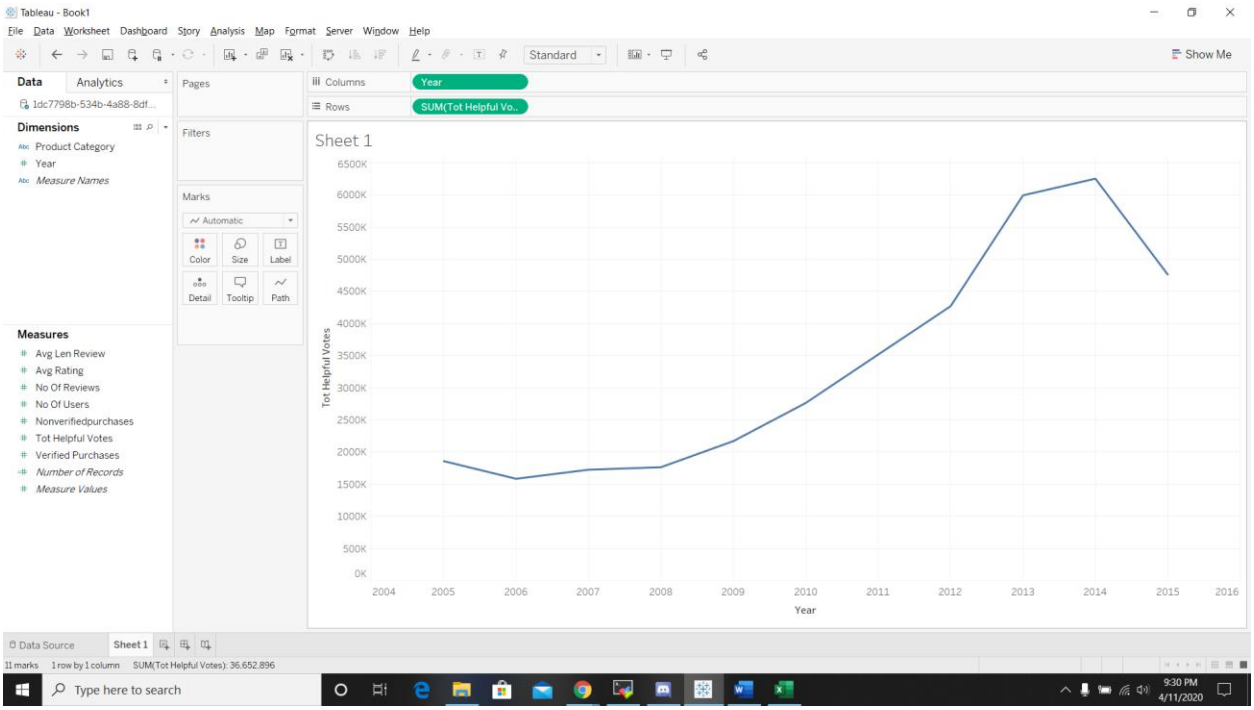
Select year, product_category, count(review_id) as NoOfReviews, count(Distinct(customer_id)) as NoOfUsers, avg(star_rating) as AvgRating ,avg(length(review_body)) as AvgLenReview, sum(case when verified_purchase='Y' then 1 else 0 end) as VerifiedPurchases, sum(case when verified_purchase='N' then 1 else 0 end) as Nonverifiedpurchases, sum(helpful_votes) as TotHelpfulVotes from amazon_review.filtered_reviews where year>=2005 group by year,product_category order by year;

Output



Vizualization

I have taken a line graph for the total helpful votes' vs year. We can clearly see that the number of total helpful votes kept on increasing until the year 2014 put since then dropped significantly.



Question 2

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

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

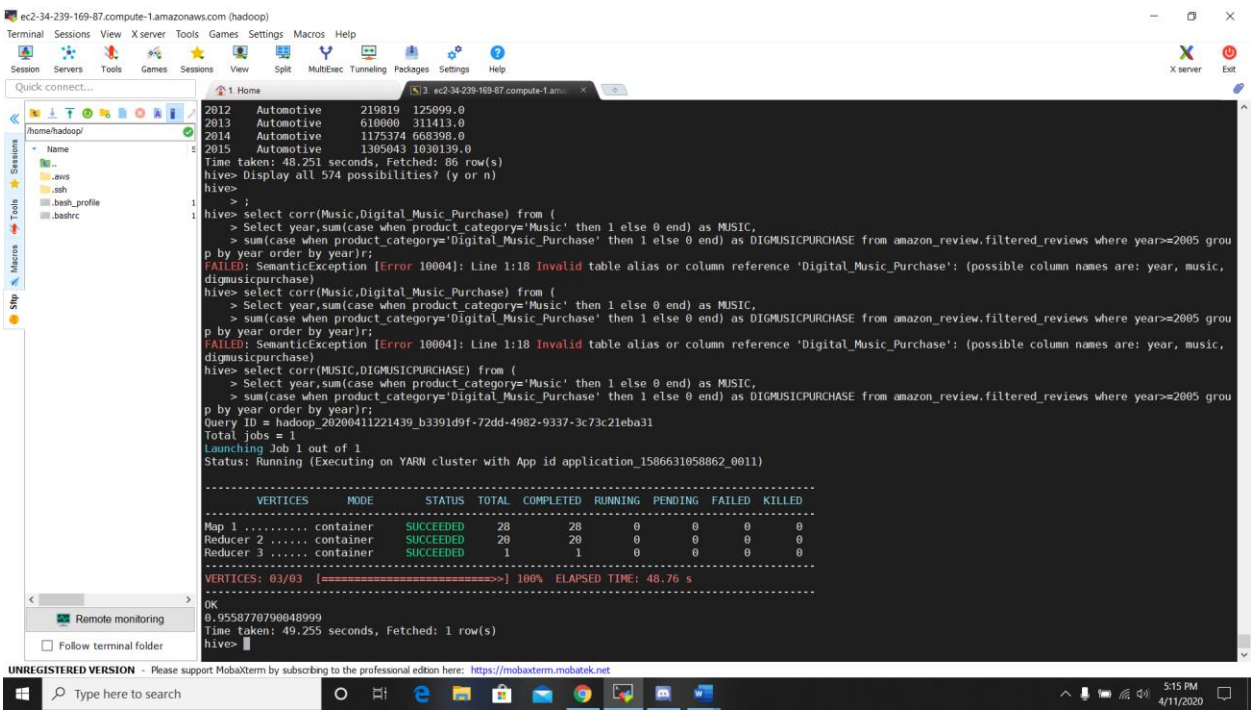
A.

Correlation for Music/Digital_Music_Purchase

Query

select corr(MUSIC,DIGMUSICPURCHASE) from (
Select year,sum(case when product_category='Music' then 1 else 0 end) as MUSIC,
sum(case when product_category='Digital_Music_Purchase' then 1 else 0 end) as DIGMUSICPURCHASE
from amazon_review.filtered_reviews where year>=2005 group by year order by year)r;

Output

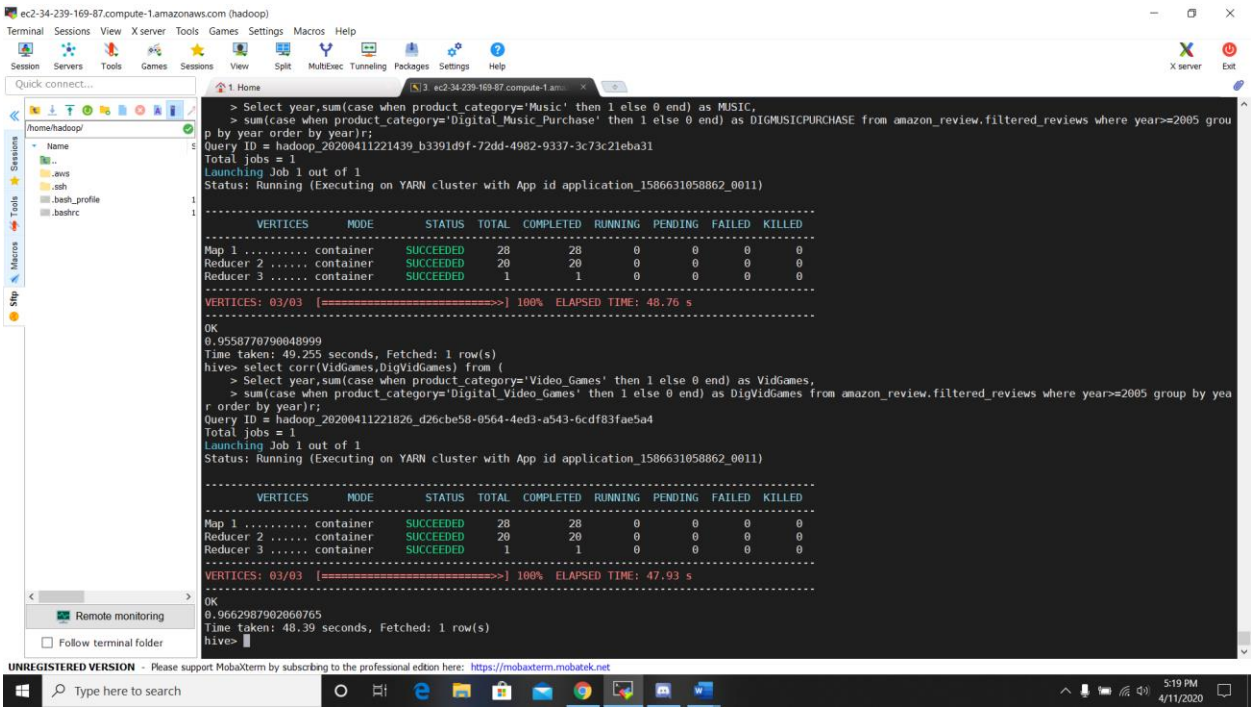


Correlation between Digital_Video_Games/Video_Games based on count of reviews.

Query

select corr(VidGames,DigVidGames) from (
Select year,sum(case when product_category='Video_Games' then 1 else 0 end) as VidGames,
sum(case when product_category='Digital_Video_Games' then 1 else 0 end) as DigVidGames from amazon_review.filtered_reviews where year>=2005 group by year order by year)r;

Output



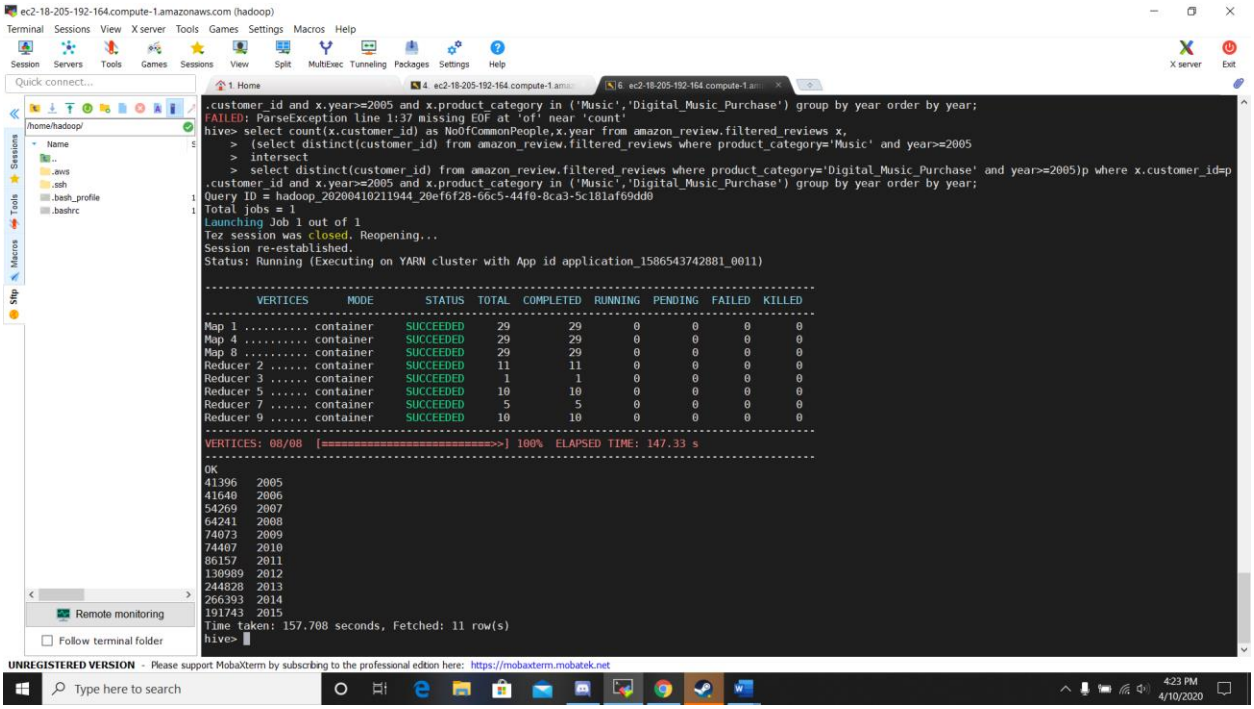
2. Are there same users reviewing in both categories?

A. Providing detailed analysis over time for the Music/Digital_Music_Purchase category.

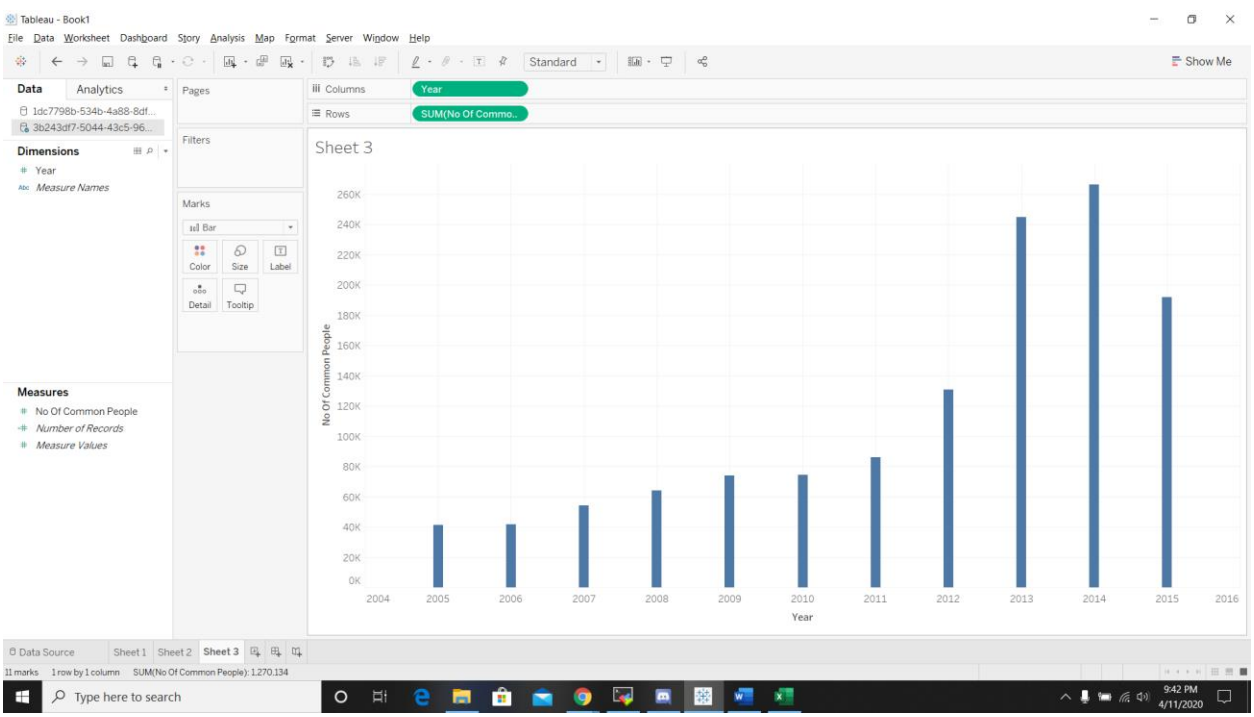
Query

select count(x.customer_id) as NoOfCommonPeople,x.year from amazon_review.filtered_reviews x,
(select distinct(customer_id) from amazon_review.filtered_reviews where product_category='Music' and year>=2005
intersect
select distinct(customer_id) from amazon_review.filtered_reviews where product_category='Digital_Music_Purchase' and year>=2005)p where x.customer_id=p.customer_id and x.year>=2005 and x.product_category in ('Music','Digital_Music_Purchase') group by year order by year;

Output



Visualization



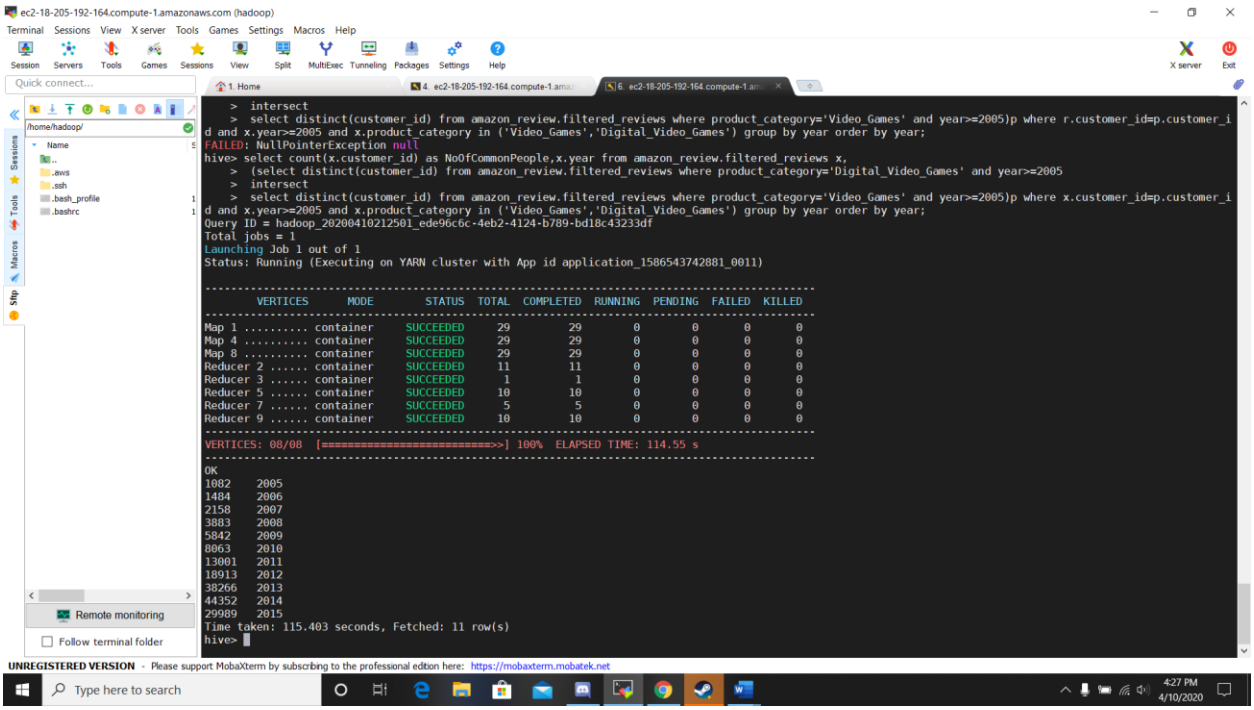
The number of common people dropped after the year 2014 significantly.

Providing detailed analysis over time for the Digital_Video_Games/Video_Games category.

Query

select count(x.customer_id) as NoOfCommonPeople,x.year from amazon_review.filtered_reviews x,
(select distinct(customer_id) from amazon_review.filtered_reviews where
product_category='Digital_Video_Games' and year>=2005
intersect
select distinct(customer_id) from amazon_review.filtered_reviews where
product_category='Video_Games' and year>=2005)p where x.customer_id=p.customer_id and
x.year>=2005 and x.product_category in ('Video_Games','Digital_Video_Games') group by year order by
year;

Output



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

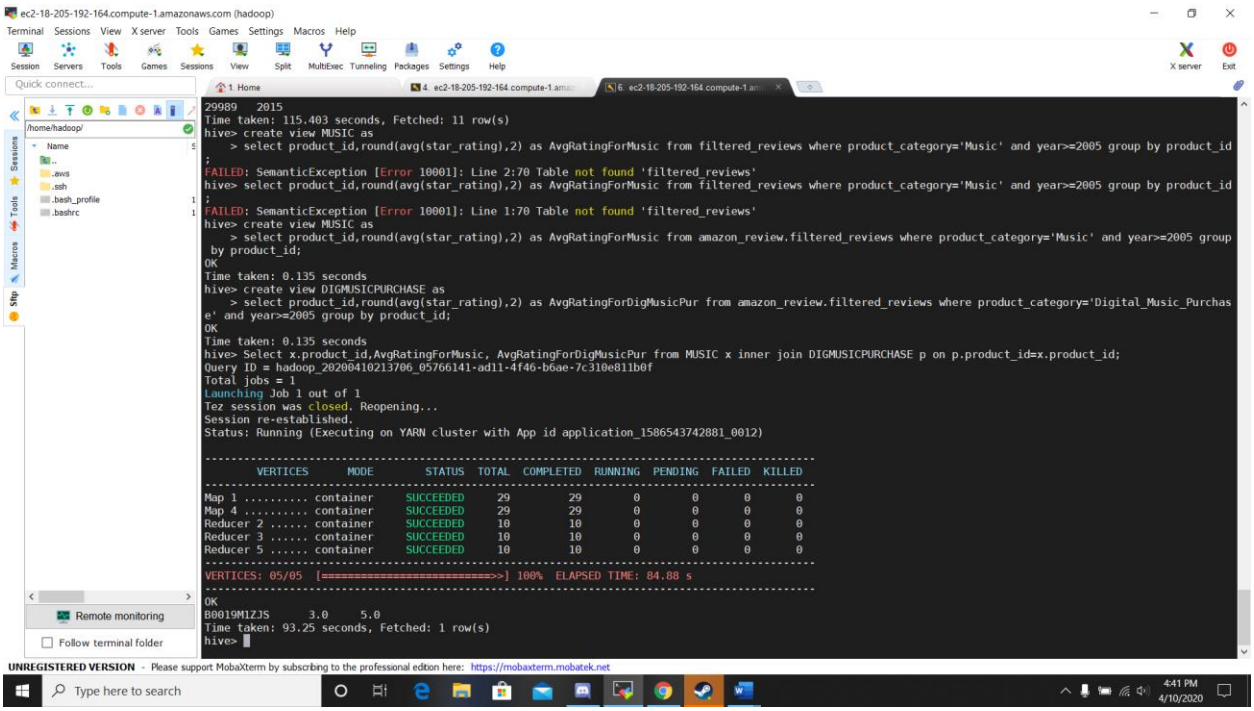
create view MUSIC as
select product_id,round(avg(star_rating),2) as AvgRatingForMusic from amazon_review.filtered_reviews where product_category='Music' and year>=2005 group by product_id;

create view DIGMUSICPURCHASE as
select product_id,round(avg(star_rating),2) as AvgRatingForDigMusicPur from amazon_review.filtered_reviews where product_category='Digital_Music_Purchase' and year>=2005 group by product_id;

Query

Select x.product_id,AvgRatingForMusic, AvgRatingForDigMusicPur from MUSIC x inner join DIGMUSICPURCHASE p on p.product_id=x.product_id;

Output



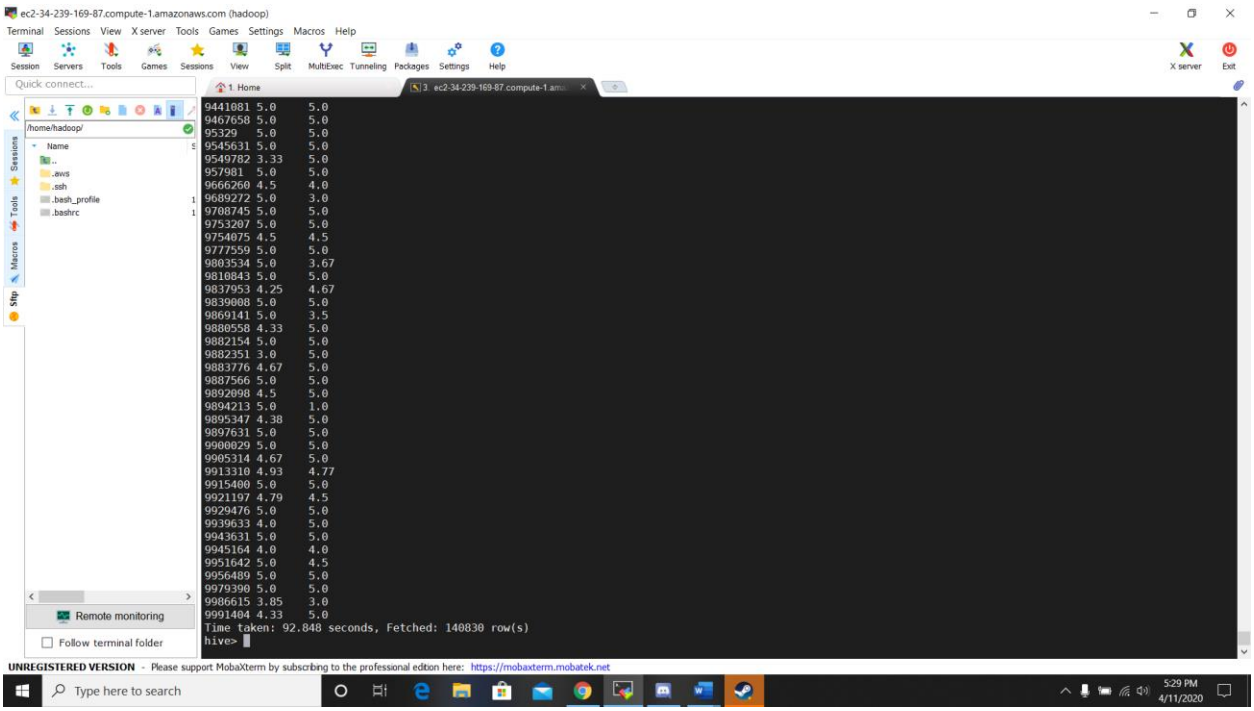
4. You should cover additional questions and not limit yourself to the above questions

List of customers who have given reviews for products in both Music and Digital_Music_Purchase category and their ratings in both categories.

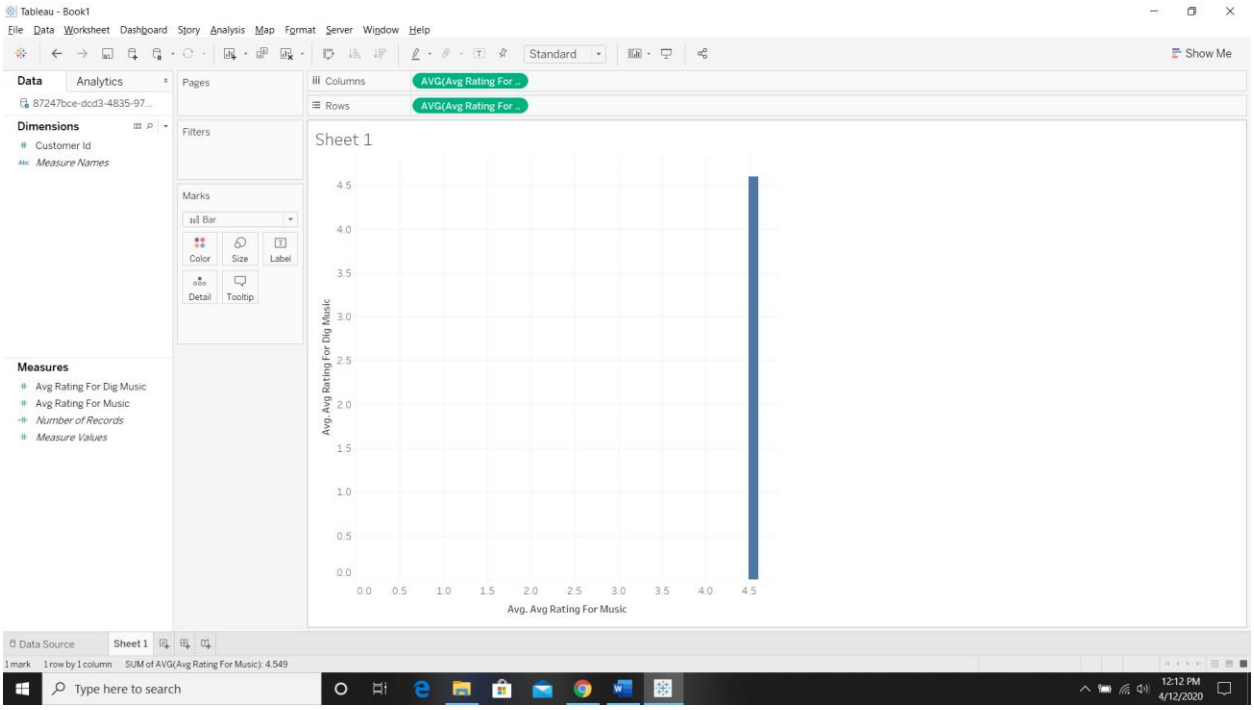
Query-

```
create view music as
 select customer_id,product_category,round(avg(star_rating),2) as AvgRatingForMusic from
amazon_review.filtered_reviews where product_category='Music' and year>=2005 group by
customer_id,product_category;

 create view DigMusic as
 select customer_id,product_category,round(avg(star_rating),2) as AvgRatingForDigMusic from
amazon_review.filtered_reviews where product_category='Digital_Music_Purchase' and year>=2005
group by customer_id,product_category;

select r.customer_id, AvgRatingForMusic, AvgRatingForDigMusic from DigMusic r inner join music u on
u.customer_id=r.customer_id;
```

Output



Visualization



People are fond of both digital music and music categories equally.

Question 3

You should demonstrate your ability to use Hive advanced functions:

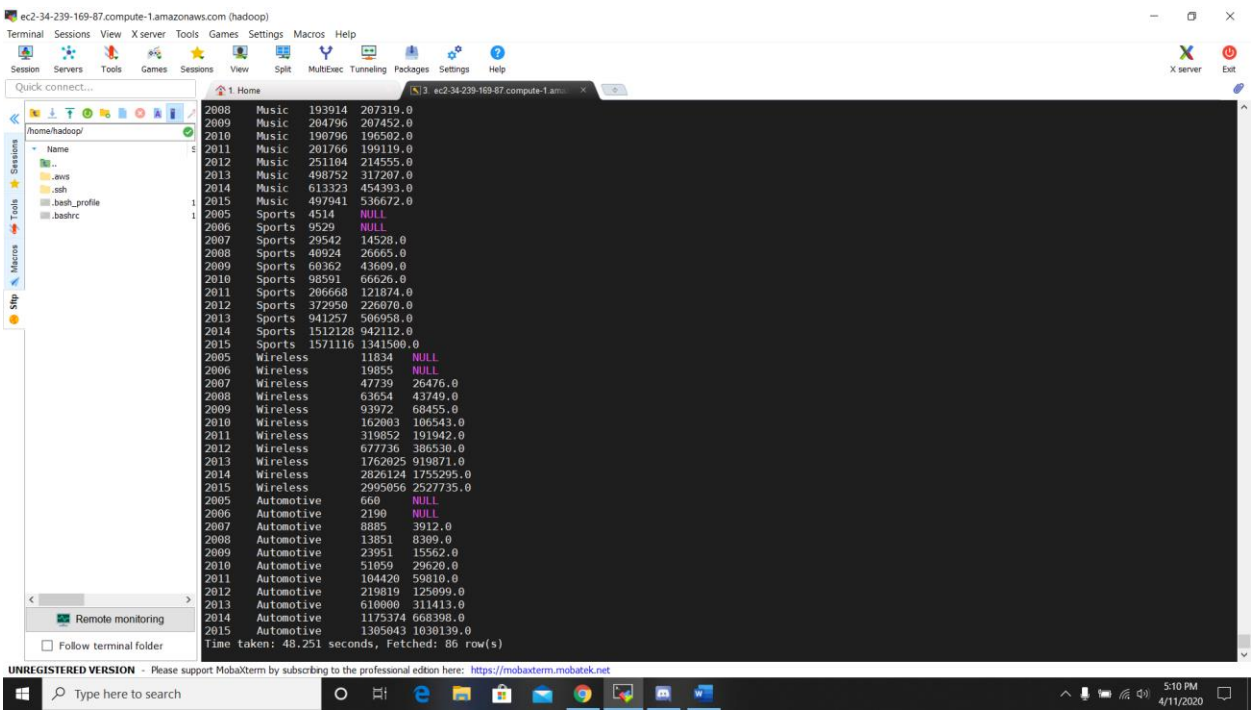
1. Window functions: moving average, rank, aggregation functions using relevant ordering and partitioning

Calculating three year Moving average based on number of reviews per product category over time.

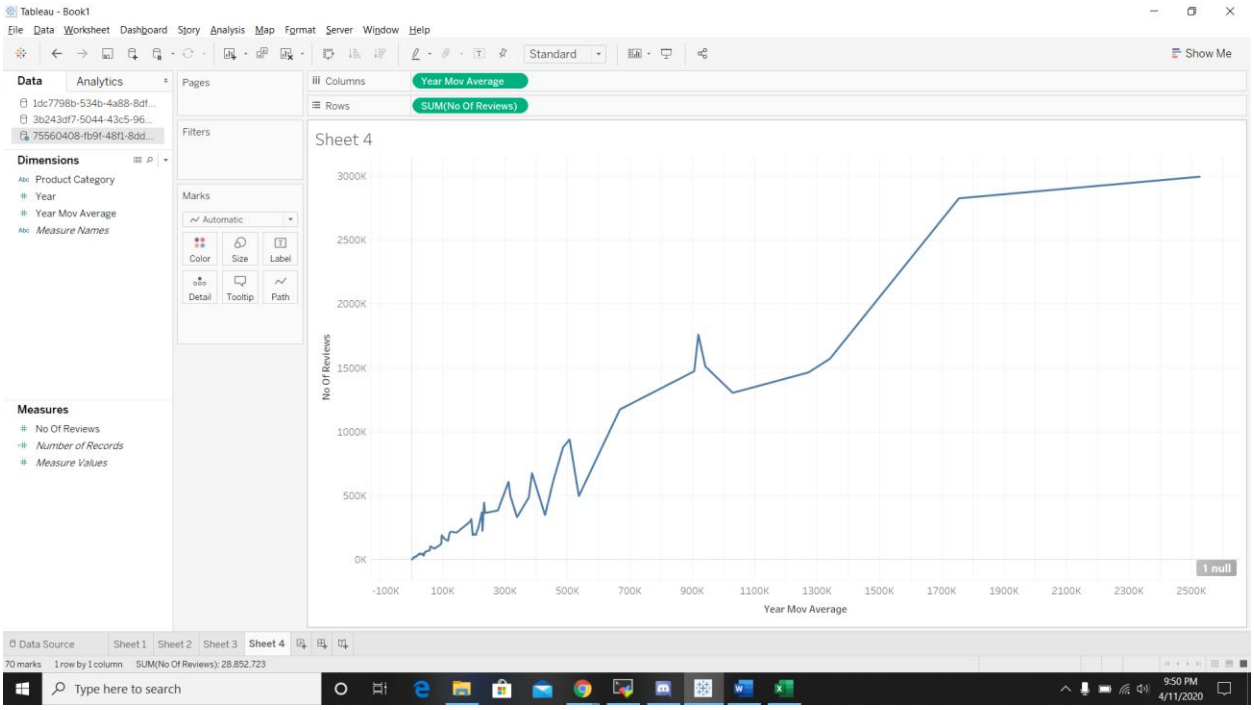
Query

```
select year,product_category,NoOfReviews,(case when row_number() over (Partition by
product_category order by year) > 2
then round(AVG(NoOfReviews) OVER (PARTITION BY product_category order by year ROWS 2
PRECEDING))
end) as 3YearMovAverage from
(Select year,product_category,count(review_id) as NoOfReviews,count(Distinct(customer_id)) as
NumberOfUser,avg(star_rating) as average_review_stars,avg(length(review_body)) as AvgLenOfReview
from amazon_review.filtered_reviews group by year,product_category order by product_category,year)
as x where year>=2005;
```

Output



Visualization



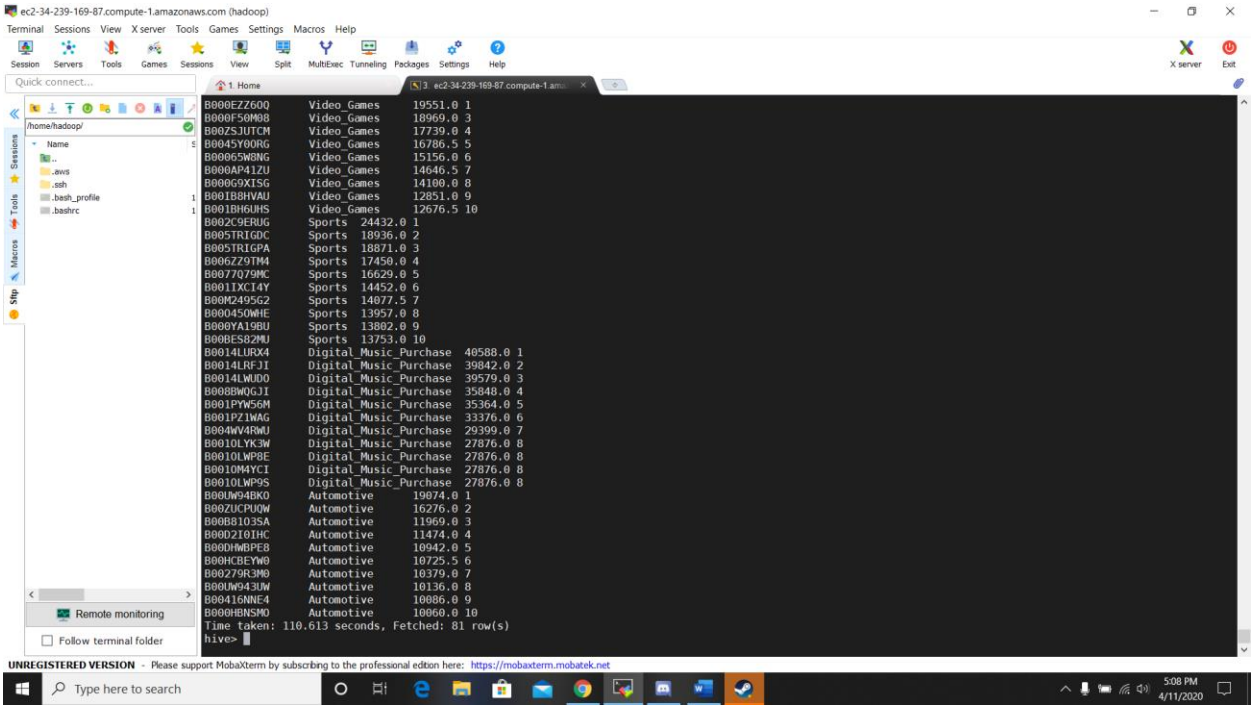
The number of reviews is increasing over the years.

Ranking top 10 products in each category based on average length of reviews.

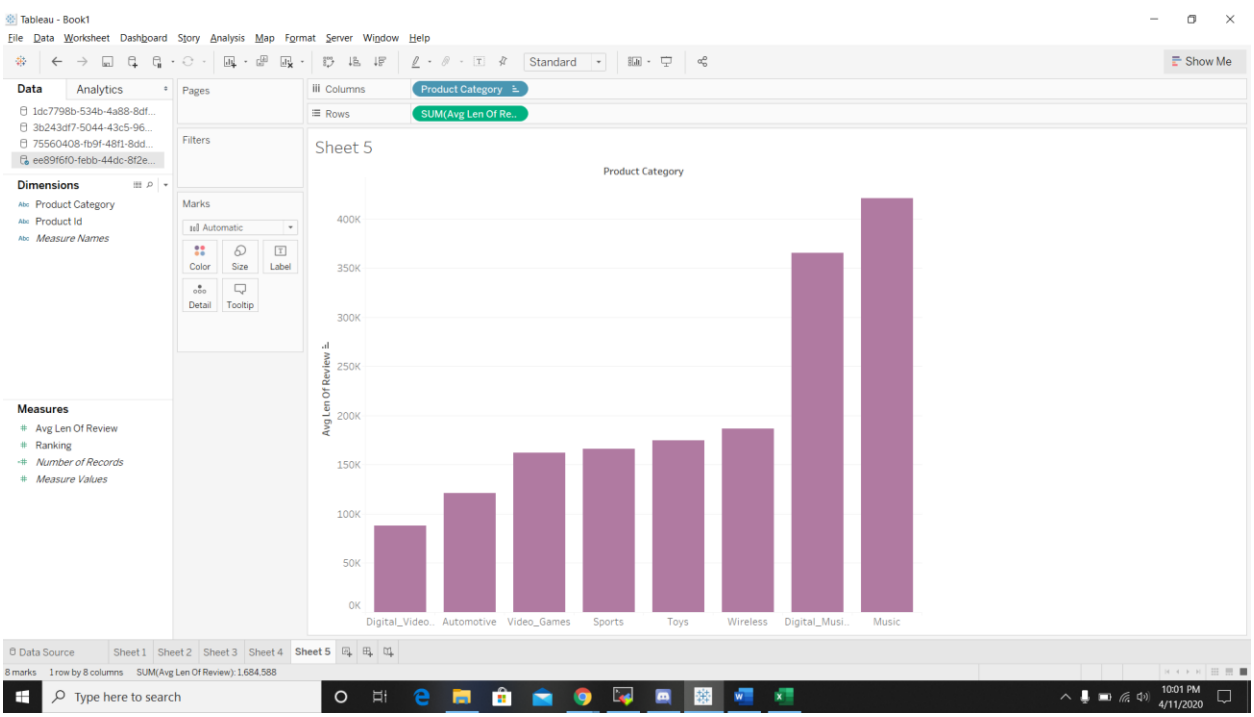
Query

```
select product_id,product_category, AvgLenOfReview,ranking from(
select product_id,product_category, AvgLenOfReview,rank() over (Partition by product_category order by AvgLenOfReview desc) as ranking from
(Select product_id,product_category,count(Distinct(customer_id)) as NoOfUsers,avg(star_rating) as AvgReviewRating,avg(length(review_body)) as AvgLenOfReview
from amazon_review.filtered_reviews group by product_category,product_id)as x)as z where ranking<=10;
```

Output



Visualization



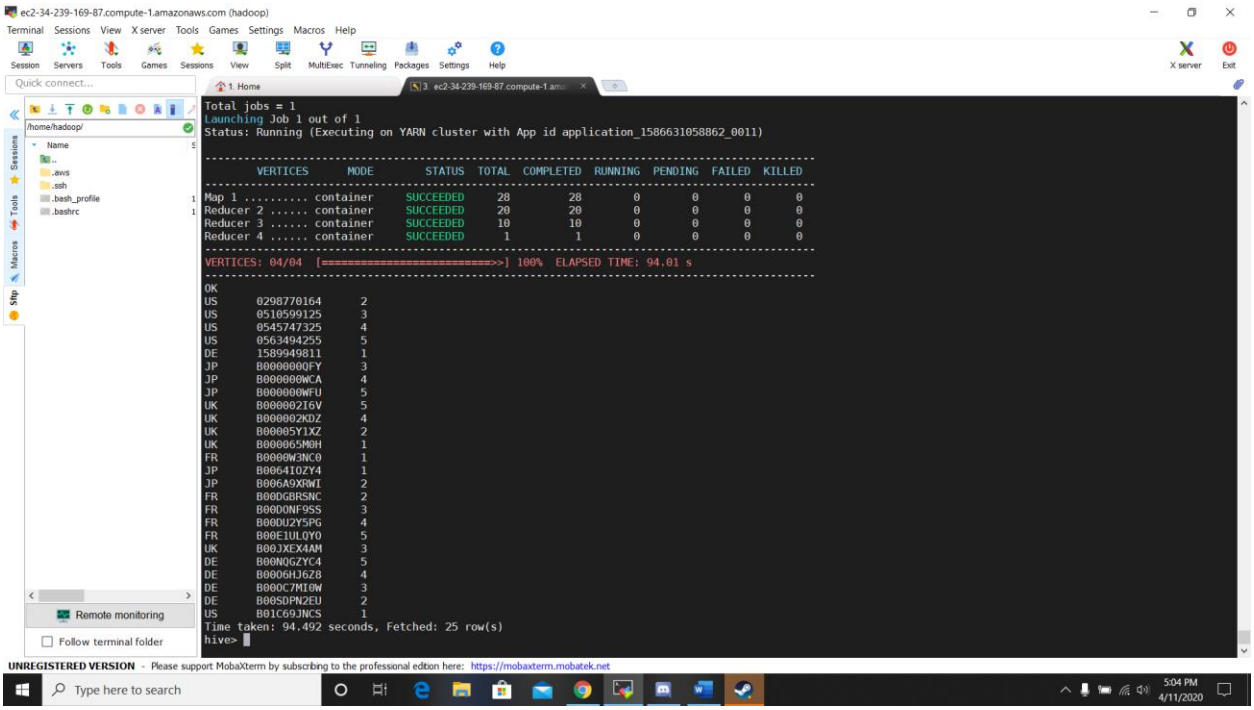
The category music has received the lengthiest reviews.

Using aggregate function average to find out top 5 products in each marketplace based on average star rating

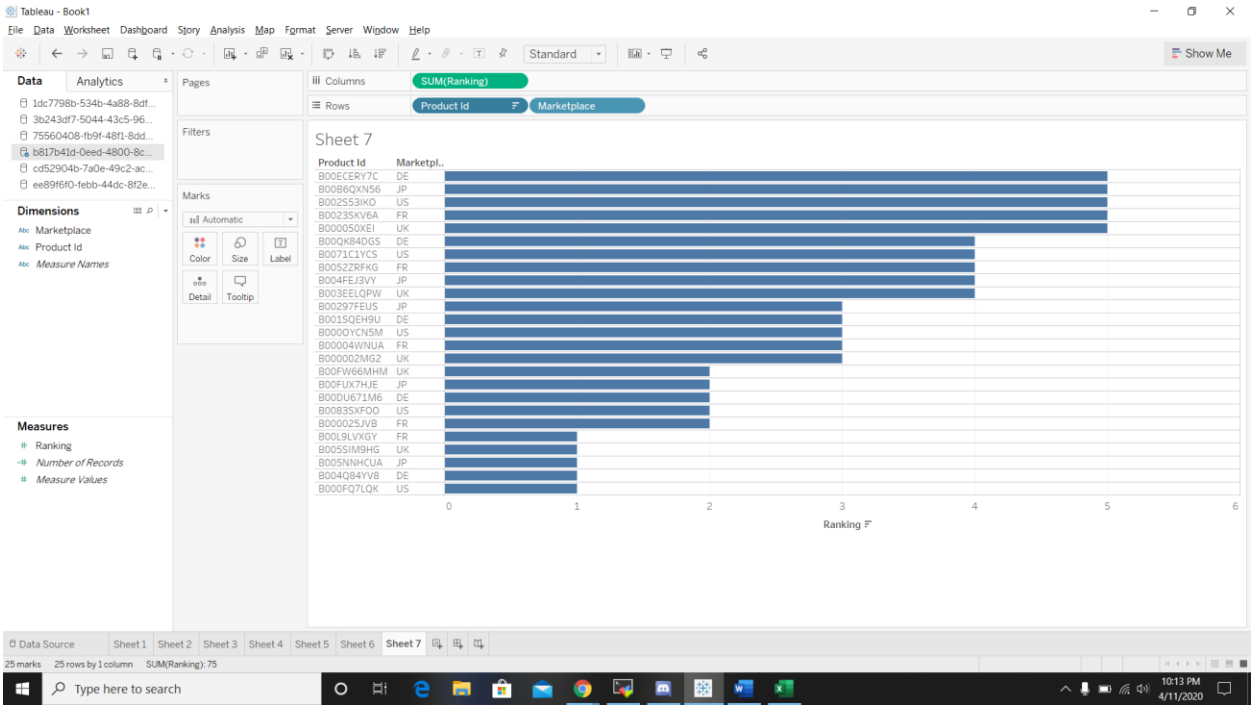
Query

```
SELECT v.marketplace,v.product_id,
      v.Ranking
FROM
   (SELECT z.product_id,
      z.marketplace,z.AvgRating,
      Row_number()
      OVER (partition by z.marketplace
      ORDER BY  z.AvgRating desc) AS Ranking
   FROM
      (SELECT product_id,
      marketplace,
      avg(star_rating) AS AvgRating
      FROM amazon_review.filtered_reviews
      WHERE year>= 2005
      GROUP BY  product_id,marketplace)as z)as v
      WHERE v.Ranking <=5 order by v.product_id;
```

Output



Visualization



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

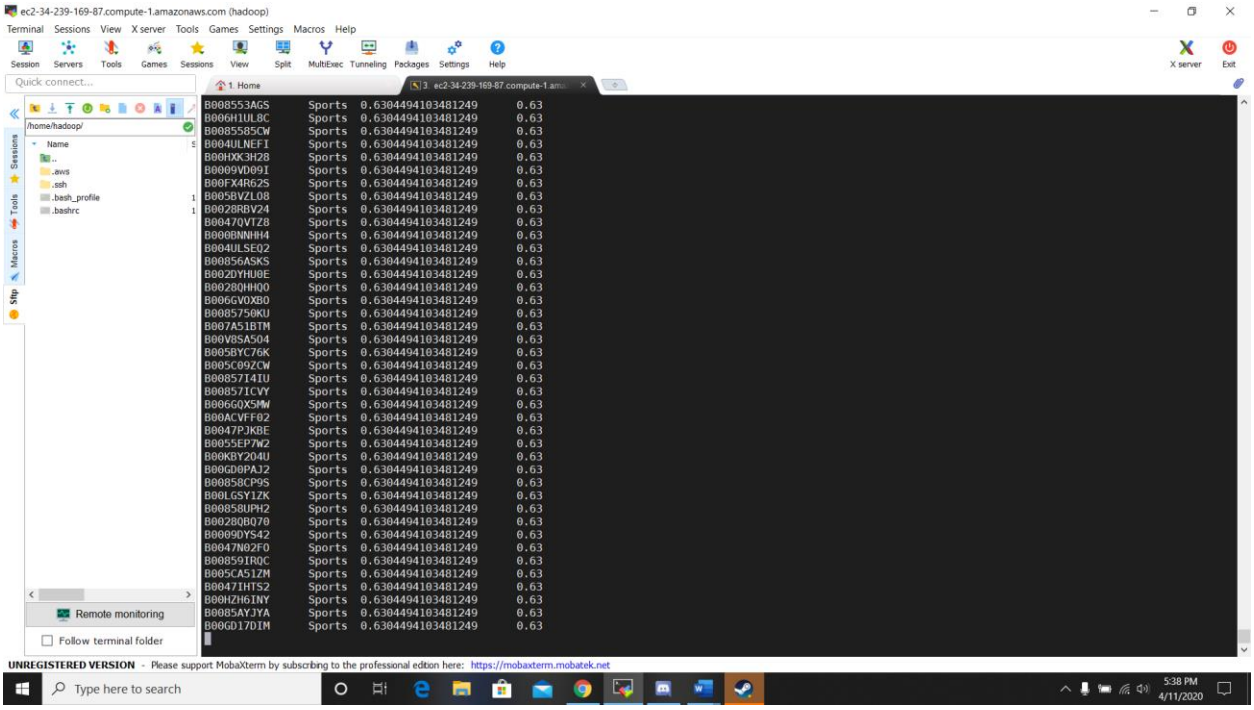
Percentile

Products having highest Percentile of star ratings given by customers:

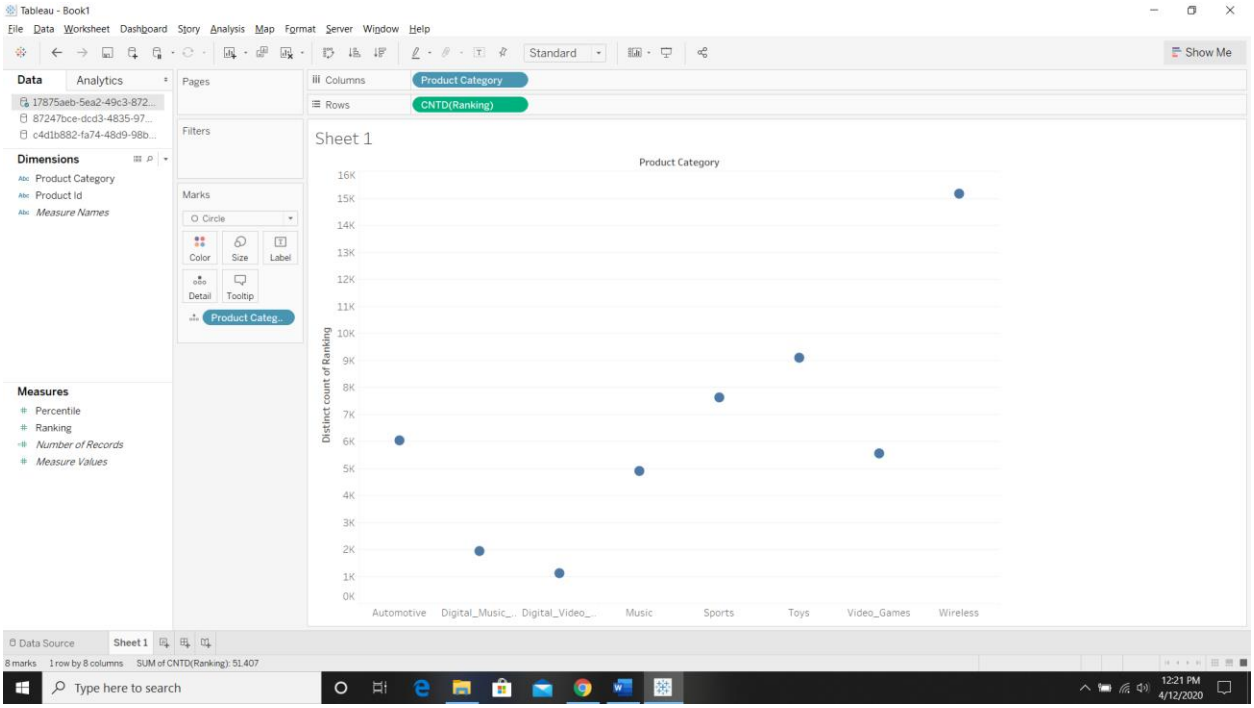
Query

```
SELECT a.product_id, a.product_category,a.Ranking,round(a.Ranking,2) as Percentile from
(SELECT b.product_id,b.product_category,PERCENT_RANK()OVER (partition by b.product_category
ORDER BY b.AvgRating desc) AS Ranking
FROM
(SELECT product_id,product_category,avg(star_rating) AS AvgRating
FROM amazon_review.filtered_reviews
WHERE year>= 2005
GROUP BY product_id,product_category)as b)as a order by a.Ranking desc;
```

Output



Visualization



Among all the categories, the product category wireless has the maximum number of reviews and users.

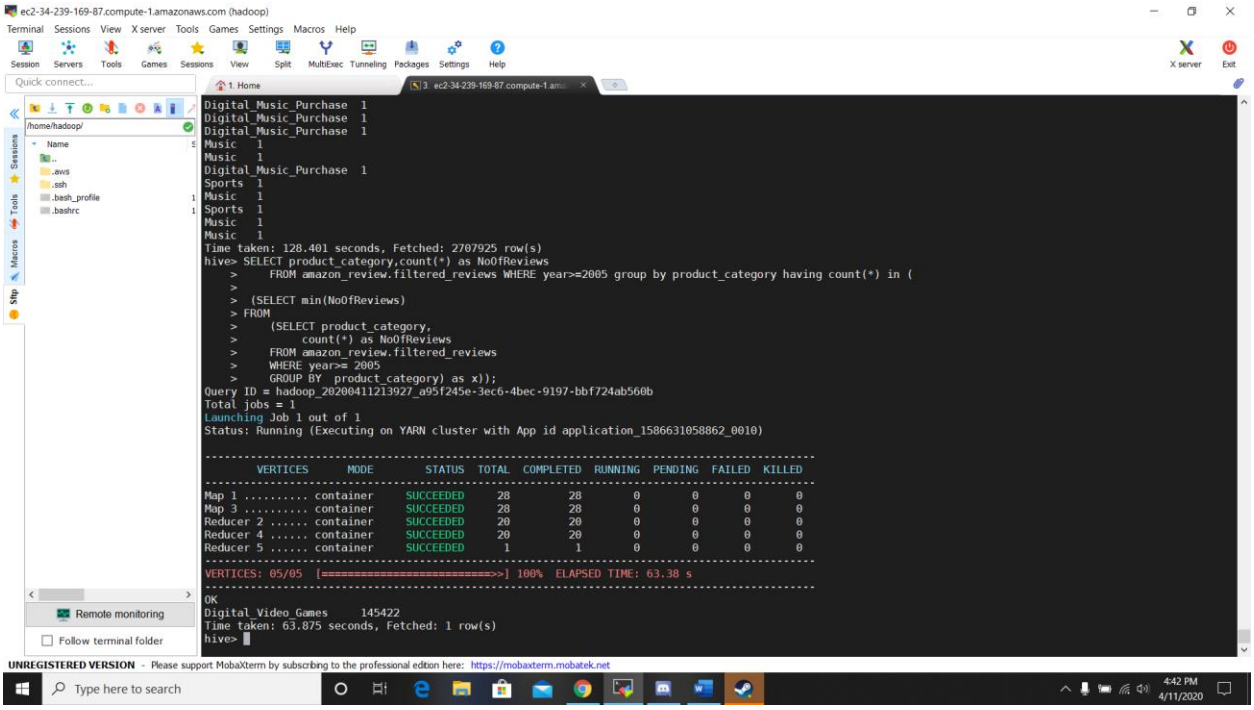
Min

Query

Product category which has got minimum number of reviews.

```
SELECT product_category,count(*) as NoOfReviews
   FROM amazon_review.filtered_reviews WHERE year>=2005 group by product_category having
count(*) in (

 (SELECT min(NoOfReviews)
FROM
   (SELECT product_category,
       count(*) as NoOfReviews
   FROM amazon_review.filtered_reviews
   WHERE year>= 2005
   GROUP BY  product_category) as x));
```

Output



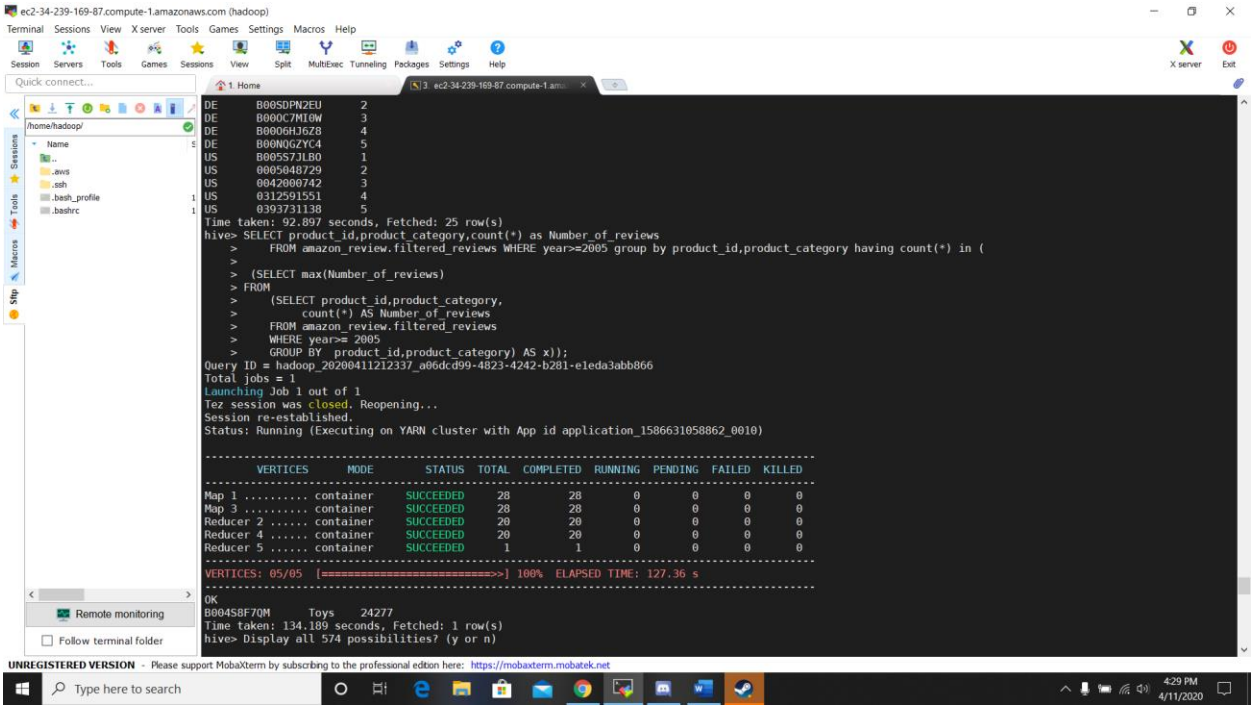
Max

Top product which has got maximum number of reviews.

Query

```
SELECT product_id,product_category,count(*) as NoOfReviews
   FROM amazon_review.filtered_reviews WHERE year>=2005 group by product_id,product_category
having count(*) in (

 (SELECT max(NoOfReviews)
FROM
   (SELECT product_id,product_category,
       count(*) as NoOfReviews
   FROM amazon_review.filtered_reviews
   WHERE year>= 2005
   GROUP BY  product_id,product_category) as x));
```

Output

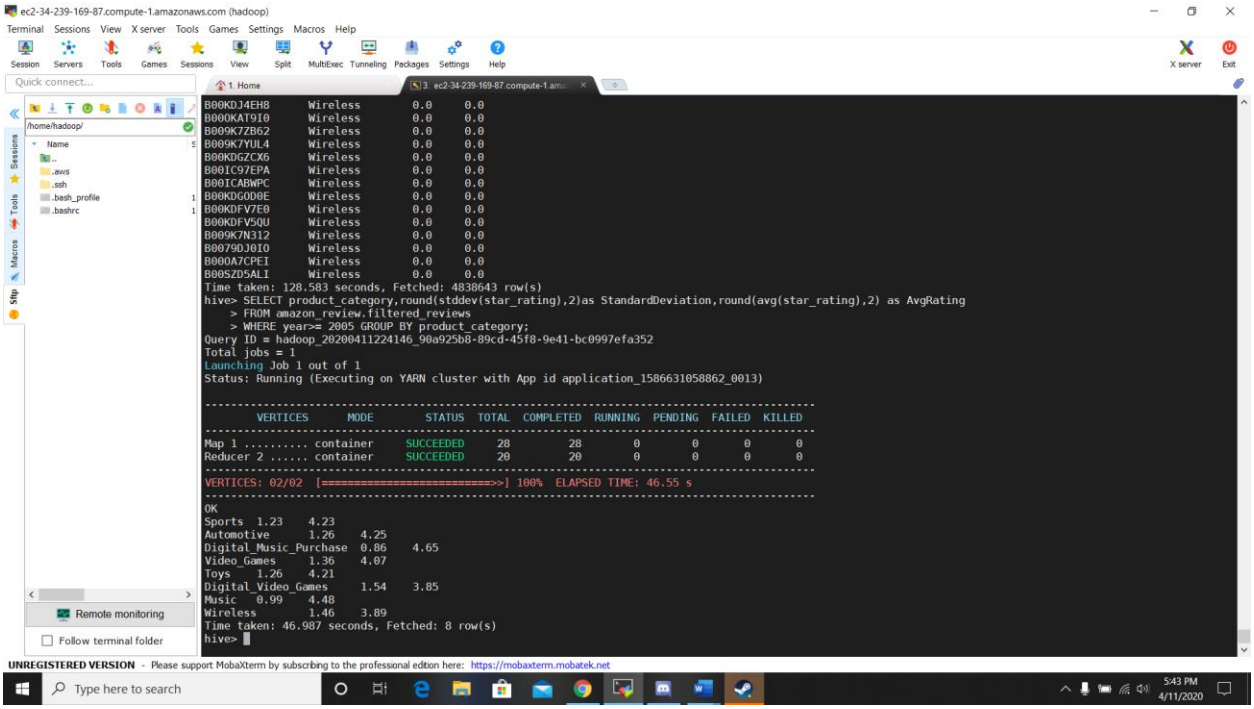


Standard deviation

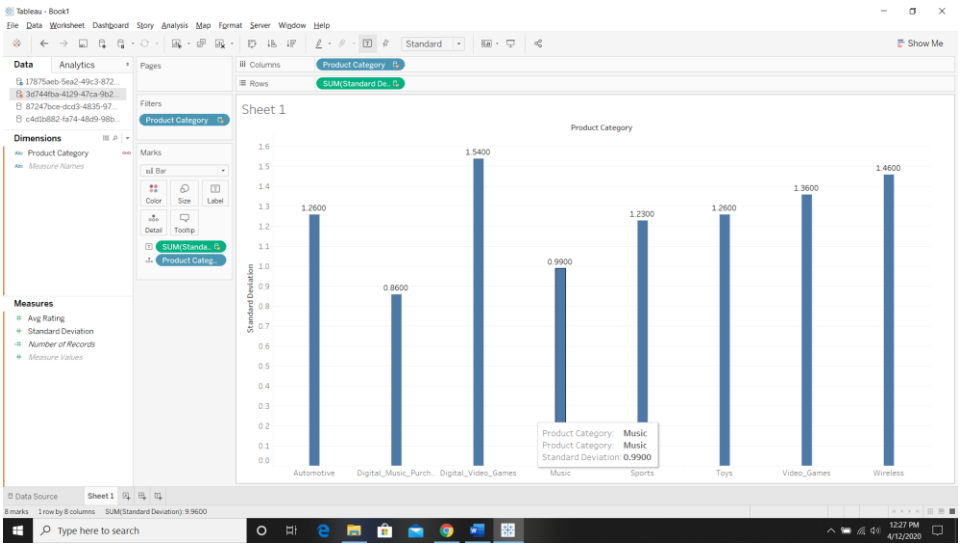
Calculating Standard Deviation to analyze normal distribution of star rating of product categories.

SELECT product_category,round(stddev(star_rating),2)as StandardDeviation,round(avg(star_rating),2) as AvgRating
FROM amazon_review.filtered_reviews
WHERE year>= 2005 GROUP BY product_category;

Output



Visualization



The standard deviation for digital video games category is the highest which means the points of data are spread-out from the mean significantly.

References

https://www.w3schools.com/

https://www.pluralsight.com/courses/aws-athena-get-started?aid=701j0000001heIoAAI&promo=&oid=7014Q0000022aAOQAY&utm_source=non_branded&utm_medium=digital_paid_search_google&utm_campaign=US_Dynamic&utm_content=&gclid=Cj0KCQjw-Mr0BRDyARIsAKEFbefZsQ7ZU8ISJP85zZBCMAmla0xhAxMLSjTxR4MUagHvCcMCBk9A3ugaApqpEALw_wcB

https://www.practicefusion.com/ehr-training/

https://www.kaggle.com/learn/advanced-sql

http://www.sqlcourse2.com/