HiveQL.

CREATE EXTERNAL TABLE 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 bigint,

year int)

PARTITIONED BY (product\_category string)

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

's3://amazon-reviews-pds/parquet/'

;

MSCK REPAIR TABLE amazon\_reviews\_parquet;

Instructions:

Step 1:

Create the table in the hive.

Step 2:

Run the MSCK command to repair the dataset so that there is no problem while doing the analysis.

Step 3 :

Run the queries provided below to get the deep analysis about the amazon review dataset.

**1(not good products).**

hive -e 'select product\_title from amazon\_reviews\_parquet where verified\_purchase="N" AND star\_rating BETWEEN '2' AND '4' AND review\_headline LIKE "%Bad%" AND review\_headline LIKE "%!%" AND review\_headline LIKE "%bad%";' > /home/hadoop/1.tsv

By this query we are analyzing that which products are not good in quality, and reviewed by the non-verified customer with star rating between “2” and “4”, and review content contain words like “bad”, “!” etc. this query can also be used to analyze fake comments by using verified customer entity.

**2(no of products / customer).**

hive -e 'select product\_id,product\_title,COUNT( customer\_id) from amazon\_reviews\_parquet GROUP BY product\_id ORDER BY COUNT(customer\_id) desc;' > /home/hadoop/2.tsv

This query generates the distinct products which shows that how many customer buy that product and gives the sale related to products. This query basically analyze the customer behavior to buy product on the site, which gives the demand of customer per product. And we can use this query to generate more sales by increasing the demand of product and giving discount on that product. This query also sort the product in descending order by counting which product brought more by customers.

**3(ratings Analysis / by year).**

hive -e 'select year,COUNT(CASE WHEN star\_rating=1 THEN star\_rating ELSE NULL END) AS star\_rating\_1, COUNT(CASE WHEN star\_rating=2 THEN star\_rating ELSE NULL END) AS star\_rating\_2, COUNT(CASE WHEN star\_rating=3 THEN star\_rating ELSE NULL END) AS star\_rating\_3, COUNT(CASE WHEN star\_rating=4 THEN star\_rating ELSE NULL END) AS star\_rating\_4, COUNT(CASE WHEN star\_rating=5 THEN star\_rating ELSE NULL END) AS star\_rating\_5 from amazon\_reviews\_parquet GROUP BY year ORDER BY year;' > /home/hadoop/3.tsv

Generally, this query shows the customers rating or reviewing the product year wise and also gives the analysis of the good to bad products per year. This gives that customer are more giving reviews about the product quality which mostly helps the other customers to buy that product or not. This query also shows the customer increasing per year which is good for the company revenue.

**4.(Looking at Reviewer Behavior).**

hive -e 'select customer\_id, count(\*),count(distinct product\_parent) AS category, cast(avg(cast(star\_rating as decimal(5,4))) as decimal (3,2)) AS avg\_rating, cast(avg(cast (helpful\_votes as decimal (18,4))) as decimal (16,2)) AS avg\_help, sum(case when star\_rating = 1 then 1 else 0 end) AS one, sum(case when star\_rating = 2 then 1 else 0 end) AS two, sum(case when star\_rating = 3 then 1 else 0 end) AS three, sum(case when star\_rating = 4 then 1 else 0 end) AS four, sum(case when star\_rating = 5 then 1 else 0 end) AS five, sum(case when vine = "Y" then 1 else 0 end) AS vine\_reviews, sum(case when verified\_purchase = "Y" then 1 else 0 end) AS verified\_purchases from amazon\_reviews\_parquet GROUP BY customer\_id ;' > /home/hadoop/4.tsv

By this query we are analyzing reviewers. This query is looking across all reviewers to see how many reviews they have done, how many product categories those reviews span, the average rating they give, and how many ratings at each level they have given.We also check how helpful their average review is, how many Vine reviews they have, and how many of their purchases are verified

**5.(Ratings Vary With verified\_purchase).**

hive -e 'select verified\_purchase,cast(avg(cast(star\_rating as decimal(5,4))) as decimal (3,2)) avg\_rating,count(\*) from amazon\_reviews\_parquet group by verified\_purchase;' > /home/hadoop/5.tsv

This query basically shows the verified purchases and there average rating by the customers and also give the total count of the verified purchases having “no” or “yes”.

**6.(Ratings Vary With vine\_membership).**

hive -e 'select vine,cast(avg(cast(star\_rating as decimal(5,4))) as decimal (3,2)) avg\_rating,count(\*) from amazon\_reviews\_parquet group by vine;' > /home/hadoop/6.tsv

Basically, vine membership is the most verified/trusted reviewers invited by the amazon itself to give reviews and post opinions on the product. This query shows the count of the vine members and not vine members. It shows the analysis of average rating drops when vine customer reviews because it gives better opinion then the non-vine members.

**7.(Ratings Vary With region)**

hive -e 'select marketplace,cast(avg(cast(star\_rating as decimal(5,4))) as decimal (3,2)) avg\_rating,count(\*) from amazon\_reviews\_parquet group by marketplace;' > /home/hadoop/7.tsv

This query basically shows that how the different products perform in different regions and what is there rating corresponding to it. It also shows that which product is more in demand in which region and also which region is more active on the site.

**8.(Average Ratings OverTime/year)**

hive -e 'select year,cast(avg(cast(star\_rating as decimal(5,4))) as decimal (3,2)) avg\_rating,count(\*) from amazon\_reviews\_parquet group by year;' > /home/hadoop/8.tsv

this query basically shows the average rating per year so that we can get better analysis about the product quality which is showing that product quality is increasing per year and also shows that customer are concern to buy that because of quality increases.

**9.(Average Ratings of product parent/ category)**

hive -e 'select product\_parent,cast(avg(cast(star\_rating as decimal(5,4))) as decimal (3,2)) avg\_rating,count(\*) from amazon\_reviews\_parquet group by product\_parent;' > /home/hadoop/9.tsv

this query shows the average rating by product category or parent, which analyze the quality of product that is which product parent have more good quality on the e-commerce.

**10.(customers/reviewers who repeatedly use the same review headline).**

hive -e 'select customer\_id, review\_headline, count(\*)from amazon\_reviews\_parquet where char\_length(review\_headline) > 10 group by customer\_id,review\_headline having count(\*) > 1;' > /home/hadoop/10.csv

This defines the customers behavior that customer is lazy to write proper review of the product else just use the same content or words when he buy something on the site.

**11.(multiple reviews for the same product, all with the same headline).**

hive -e 'select customer\_id,product\_id,review\_headline, count(\*)from amazon\_reviews\_parquet where char\_length(review\_headline) > 10 group by customer\_id,product\_id,review\_headline having count(\*) > 1; > /home/hadoop/11.tsv

This query shows that there are some products which are having the multiple reviews for it i.e same and also with the same headline.this shows that customer is doing copy and paste and reviewing the same product again and again.