HIVE ASSIGNMENT

1. Download vechile sales data -> https://github.com/shashank-mishra219/Hive-Class/blob/main/sales_order_data.csv

   ```
   Ls /home/cloudera/Reethesh_hive_challenge/sales_order_data.csv
   ```

2. Store raw data into hdfs location

   ```
   Hadoop fs –put /home/cloudera/Reethesh_hive_challenge/sales_order_data.csv
   /tmp/Reethesh_hive_challenge_hdfs
   ```

3. Create a internal hive table "sales_order_csv" which will store csv data sales_order_csv ..

   make sure to skip header row while creating table

   ```
   Hive> create table Sales_order_csv (Order_No int,Quantity_order int,Price_Each float,Status
   string,Qtr_Id int,Month_Id int,Year_id int,Product_line string,MSRP int,Product_code
   string,Phone_No string,City string,State string,Postal_code string,Country string,Territory
   string,Contact_Last_name string,Contact_First_name,Dealsize string) row format delimited
   fields terminated by  ‘, ‘ tblproperties(“skip.header.line.count” =”1”);
   ```

4. Load data from hdfs path into "sales_order_csv"

   ```
   >Load data inpath  ‘/tmp/Reethesh_hive_challenge_hdfs/sales_order_data.csv’ into table
   sales_order_csv
   ```

5. Create an internal hive table which will store data in ORC format "sales_order_orc"

   ```
   Hive> create table Sales_order_orc (Order_No int,Quantity_order int,Price_Each float,Status
   string,Qtr_Id int,Month_Id int,Year_id int,Product_line string,MSRP int,Product_code
   string,Phone_No string,City string,State string,Postal_code string,Country string,Territory
   string,Contact_Last_name string,Contact_First_name,Dealsize string) stored as ORC;
   ```

6. Load data from "sales_order_csv" into "sales_order_orc"

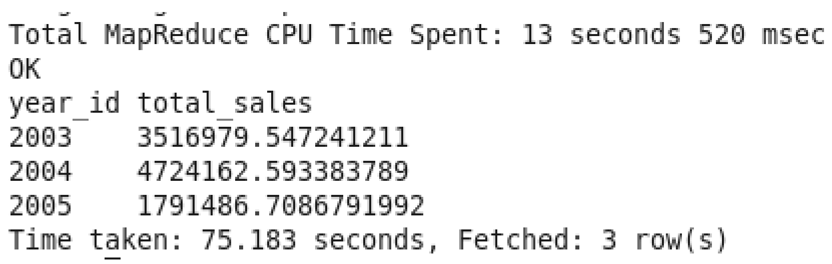
   ```
   >from sales_order_csv insert overwrite table sales_order_orc select*;
   ```

HIVE ASSIGNMENT

Perform below menioned queries on "sales_order_orc" table :

a. Calculate total sales per year

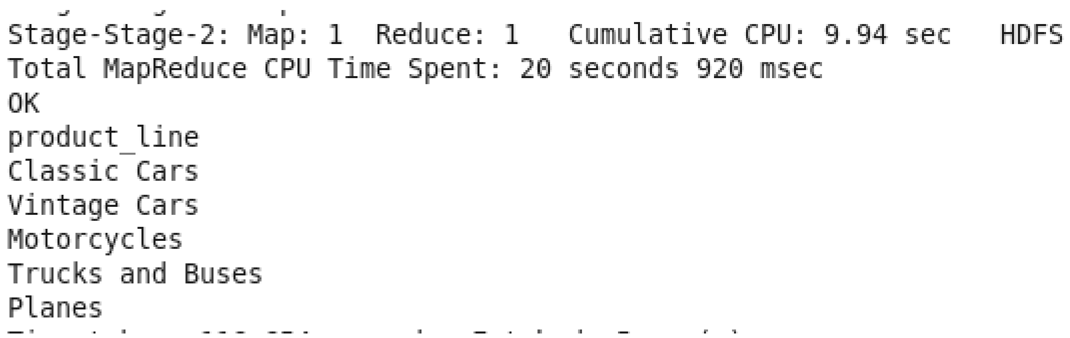
> select year_id,sum(sales) as total_sales from sales_order_orc group by year_id;



```
Total MapReduce CPU Time Spent: 13 seconds 520 msec
OK
year_id total_sales
2003    3516979.547241211
2004    4724162.593383789
2005    1791486.7086791992
Time taken: 75.183 seconds, Fetched: 3 row(s)
```

b. Find a product for which maximum orders were placed

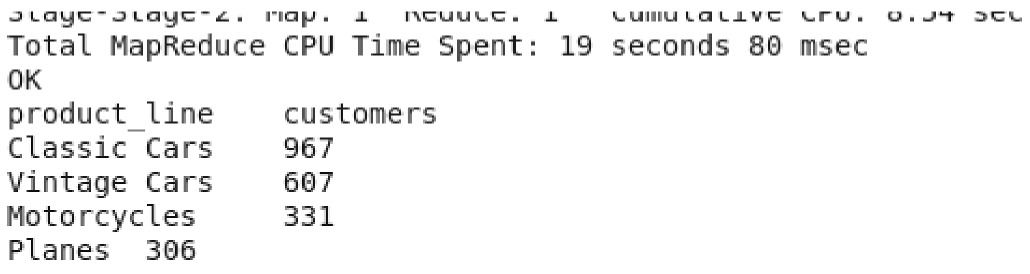
>select product_line from (select product_line,sum(Quantity_order) max from sales_order_csv group by Product_line order by max desc 5)a;



```
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 9.94 sec   HDFS
Total MapReduce CPU Time Spent: 20 seconds 920 msec
OK
product_line
Classic Cars
Vintage Cars
Motorcycles
Trucks and Buses
Planes
```

Count of sales-

>select productline, count(sales) as customers from sales_order_orc group by product_line order by customers desc;

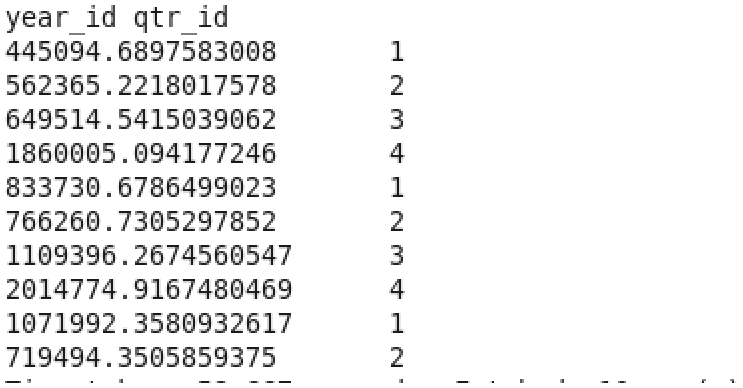


```
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: [illegible] sec
Total MapReduce CPU Time Spent: 19 seconds 80 msec
OK
product_line    customers
Classic Cars    967
Vintage Cars    607
Motorcycles     331
Planes  306
```

c. Calculate the total sales for each quarter

Year by qtr sales

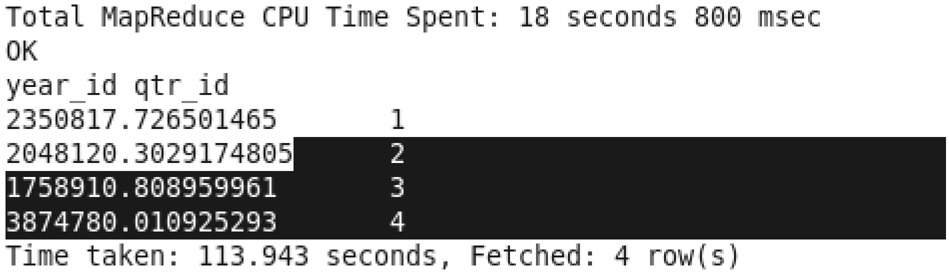
> select sum(sales), year_id,qtr_id from sales_order_csv group by year_id,qtr-id;



```
year_id qtr_id
445094.6897583008       1
562365.2218017578       2
649514.5415039062       3
1860005.094177246       4
833730.6786499023       1
766260.7305297852       2
1109396.2674560547      3
2014774.9167480469      4
1071992.3580932617      1
719494.3505859375       2
```

Total sales of Each 4 qtr including all years

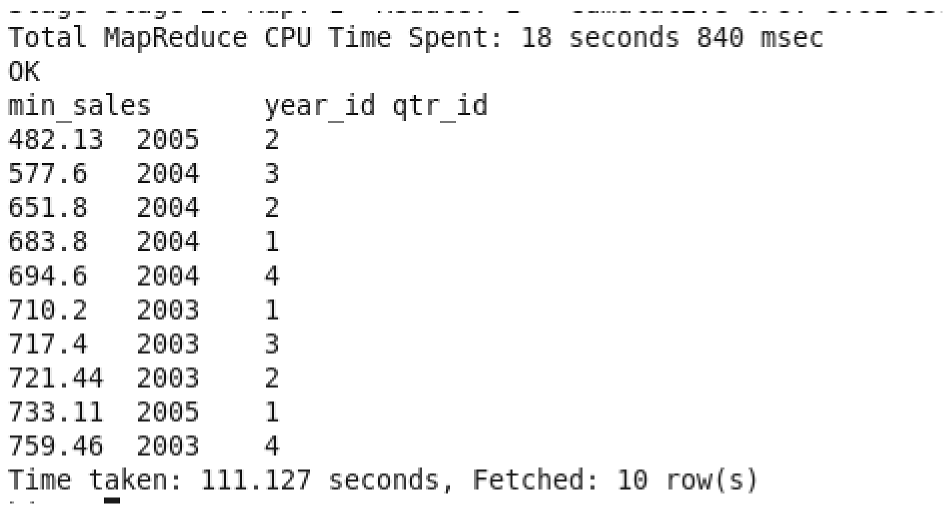
> select sum(sales), year_id,qtr_id from sales_order_csv group by qtr-id order by qtr_id;



```
Total MapReduce CPU Time Spent: 18 seconds 800 msec
OK
year_id qtr_id
2350817.726501465       1
2048120.3029174805      2
1758910.808959961       3
3874780.010925293       4
Time taken: 113.943 seconds, Fetched: 4 row(s)
```

d. In which quarter sales was minimum

>select min(sales) as min_sales,year_id,qtr_id from sales_order_orc group by year_id,qtr_id order by min_sales;

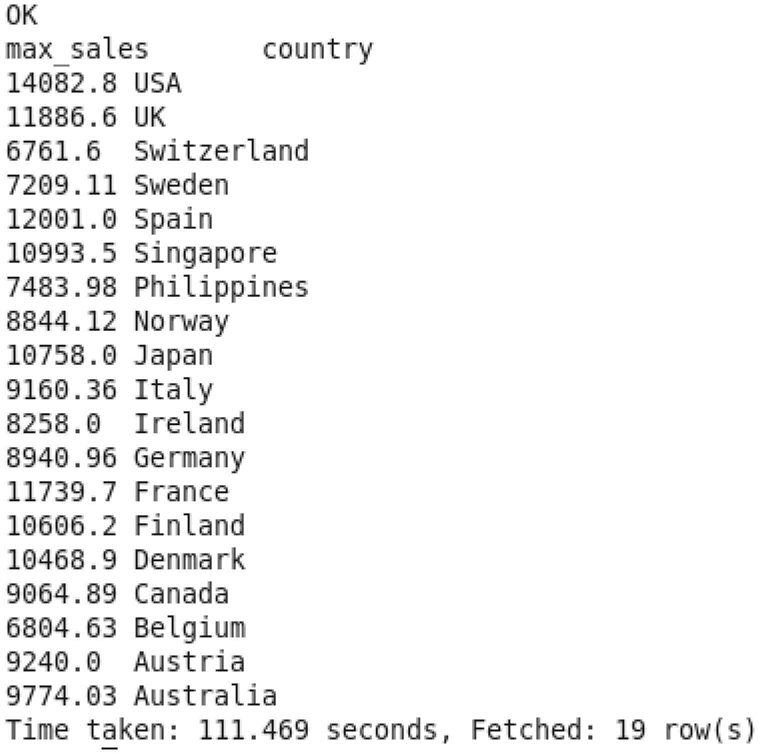


```
Total MapReduce CPU Time Spent: 18 seconds 840 msec
OK
min_sales       year_id qtr_id
482.13  2005    2
577.6   2004    3
651.8   2004    2
683.8   2004    1
694.6   2004    4
710.2   2003    1
717.4   2003    3
721.44  2003    2
733.11  2005    1
759.46  2003    4
Time taken: 111.127 seconds, Fetched: 10 row(s)
```

e. In which country sales was maximum and in which country sales was minimum

   Max sales-country wise

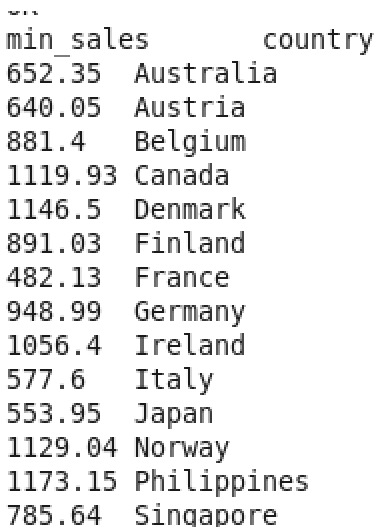
   >Select max(sales) as max_sales,country from sales_order_orc group by country order by country desc;



```
OK
max_sales       country
14082.8 USA
11886.6 UK
6761.6  Switzerland
7209.11 Sweden
12001.0 Spain
10993.5 Singapore
7483.98 Philippines
8844.12 Norway
10758.0 Japan
9160.36 Italy
8258.0  Ireland
8940.96 Germany
11739.7 France
10606.2 Finland
10468.9 Denmark
9064.89 Canada
6804.63 Belgium
9240.0  Austria
9774.03 Australia
Time taken: 111.469 seconds, Fetched: 19 row(s)
```

   Min sales

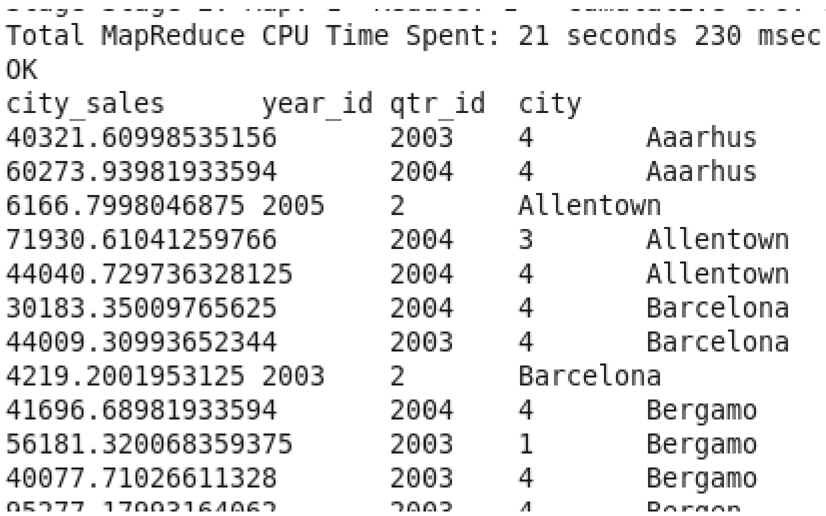
   >Select min(sales) as min_sales,country from sales_order_orc group by country order by country;



```
OK
min_sales       country
652.35  Australia
640.05  Austria
881.4   Belgium
1119.93 Canada
1146.5  Denmark
891.03  Finland
482.13  France
948.99  Germany
1056.4  Ireland
577.6   Italy
553.95  Japan
1129.04 Norway
1173.15 Philippines
785.64  Singapore
```

f. Calculate quartelry sales for each city

Select sum(sales) as city_sales, year_id,qtr_id,city from sales_order_orc group by year_id,qtr_id,city order by city ;

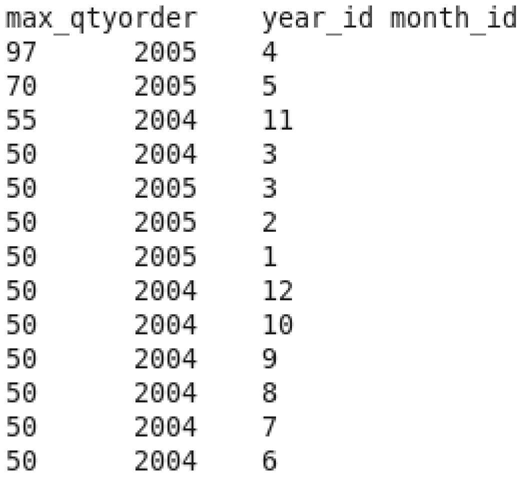


```
Total MapReduce CPU Time Spent: 21 seconds 230 msec
OK
city_sales      year_id qtr_id  city
40321.60998535156       2003    4       Aaarhus
60273.93981933594       2004    4       Aaarhus
6166.7998046875 2005    2       Allentown
71930.61041259766       2004    3       Allentown
44040.729736328125      2004    4       Allentown
30183.35009765625       2004    4       Barcelona
44009.30993652344       2003    4       Barcelona
4219.2001953125 2003    2       Barcelona
41696.68981933594       2004    4       Bergamo
56181.320068359375      2003    1       Bergamo
40077.71026611328       2003    4       Bergamo
95277.17993164062       2003    4       Bergen
```

g. Find a month for each year in which maximum number of quantities were sold

>Select max(quantity_order as max_order,year_id,month_id from sales_order_orc group by year_id,month_id order by max_order desc;

HIVE ASSIGNMENT



```
max_qtyorder    year_id month_id
97      2005    4
70      2005    5
55      2004    11
50      2004    3
50      2005    3
50      2005    2
50      2005    1
50      2004    12
50      2004    10
50      2004    9
50      2004    8
50      2004    7
50      2004    6
```