

HIVE ASSIGNMENT-1

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_data_csv
> (
> ORDERNUMBER int,
> QUANTITYORDERED int,
> PRICEEACH float,
> ORDERLINENUMBER int,
> SALES float,
> STATUS string,
> QTR_ID int,
> MONTH_ID int,
> YEAR_ID int,
> PRODUCTLINE string,
> MSRP int,
> PRODUCTCODE string,
> PHONE string,
> CITY string,
> STATE string,
> POSTALCODE string,
> COUNTRY string,
> TERRITORY string,
> CONTACTLASTNAME string,
> CONTACTFIRSTNAME string,
> DEALSIZE string
> )
> row format delimited
> fields terminated by ','
> tblproperties("skip.header.line.count"="1")
> ;
```

OK

Time taken: 0.79 seconds

hive> █

Load data from hdfs path into "sales_order_csv"

Time taken: 0.019 seconds

hive> load data inpath '/bhavey/sales.csv' into table sales_order_data_csv;

Loading data to table hive_class_b1.sales_order_data_csv

Table hive_class_b1.sales_order_data_csv stats: [numFiles=1, totalSize=357411]

OK

Time taken: 0.447 seconds

hive> █

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

```
> ;
hive> create table sales_order_data_orc
> (
> ORDERNUMBER int,
> QUANTITYORDERED int,
> PRICEEACH float,
> ORDERLINENUMBER int,
> SALES float,
> STATUS string,
> QTR_ID int,
> MONTH_ID int,
> YEAR_ID int,
> PRODUCTLINE string,
> MSRP int,
> PRODUCTCODE string,
> PHONE string,
> CITY string,
> STATE string,
> POSTALCODE string,
> COUNTRY string,
> TERRITORY string,
> CONTACTLASTNAME string,
> CONTACTFIRSTNAME string,
> DEALSIZE string
> )
> stored as orc
> ;
```

OK

Time taken: 0.108 seconds

s

Load data from "sales_order_csv" into "sales_order_orc"

```
Time taken: 0.018 seconds, Fetched: 2 row(s)
hive>
> insert into table sales_order_data_orc select * from sales_order_data_csv;
Query ID = cloudera_20220916073939_1810c515-8796-4b1d-ae3b-ea2c997ad5db
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1663306194010_0001, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1663306194010_0001/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1663306194010_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2022-09-16 07:39:51,603 Stage-1 map = 0%, reduce = 0%
2022-09-16 07:39:55,739 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.3 sec
MapReduce Total cumulative CPU time: 1 seconds 300 msec
Ended Job = job_1663306194010_0001
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to: hdfs://quickstart.cloudera:8020/user/hive/warehouse/hive_class_b1.db/sales_order_data_orc/.hive-staging_hive_2022-09-16_07-39-44_655_248345027
6206075653-1/-ext-10000
Loading data to table hive_class_b1.sales_order_data_orc
Table hive_class_b1.sales_order_data_orc stats: [numFiles=1, numRows=2823, totalSize=37548, rawDataSize=3153291]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Cumulative CPU: 1.3 sec HDFS Read: 364569 HDFS Write: 37645 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 300 msec
OK
Time taken: 12.415 seconds
```

a. Calculate total sales per year

select year_id as YEAR, sum(sales) as TOTAL_SALES from sales_order_data_orc group by year_id;

```
Total MapReduce CPU Time Spent: 1 seconds 640 msec
OK
year      total_sales
2003      3516979.547241211
2004      4724162.593383789
2005      1791486.7086791992
Time taken: 14.214 seconds, Fetched: 3 row(s)
hive>
```

b. Find a product for which maximum orders were placed

Select productcode, sum(Quantityordered) as Quantity_ORDERED from
sales_order_data_orc group by productcode order by Quantity_ORDERED desc limit 1 ;

```
Total MapReduce CPU Time Spent: 3 seconds 630 msec
OK
productcode      quantity_ordered
S18_3232         1774
Time taken: 28.727 seconds, Fetched: 1 row(s)
```

c. Calculate the total sales for each quarter

```
select QTR_ID, sum(sales) as SALES from sales_order_data_orc group by QTR_ID;
```

```
Total MapReduce CPU Time Spent: 1 seconds 650 msec
OK
qtr_id  sales
1       2350817.726501465
2       2048120.3029174805
3       1758910.808959961
4       3874780.010925293
Time taken: 13.678 seconds, Fetched: 4 row(s)
hive> █
```

d. In which quarter sales was minimum

```
select QTR_ID, sum(sales) as SALES from sales_order_data_orc group by QTR_ID order by
Sales limit 1;
```

```
Total MapReduce CPU Time Spent: 3 seconds 90 msec
OK
qtr_id  sales
3       1758910.808959961
Time taken: 29.256 seconds, Fetched: 1 row(s)
hive> █
```

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

```
select country As country_with_max_sale from (Select country,sum(sales) as Total_Sales from sales_order_data_orc group by Country order by Total_Sales desc limit 1) t;
```

```
select country As country_with_min_sale from (Select country,sum(sales) as Total_Sales from sales_order_data_orc group by Country order by Total_Sales limit 1) tab;
```

Total MapReduce CPU Time Spent: 3 seconds 370 msec

OK

country_with_max_sale

USA

Time taken: 28.788 seconds, Fetched: 1 row(s)

Total MapReduce CPU Time Spent: 4 seconds 500 msec

OK

country_with_min_sale

Ireland

Time taken: 32.92 seconds, Fetched: 1 row(s)

f. Calculate quartelry sales for each city

```
select city,QTR_ID, sum(sales) as SALES from sales_order_data_orc group by city,QTR_ID order by city,QTRD_ID;
```

OK

city	qtr_id	sales
Aarhus	4	100595.5498046875
Allentown	2	6166.7998046875
Allentown	3	71930.61041259766
Allentown	4	44040.729736328125
Barcelona	2	4219.2001953125
Barcelona	4	74192.66003417969
Bergamo	1	56181.320068359375
Bergamo	4	81774.40008544922
Bergen	3	16363.099975585938
Bergen	4	95277.17993164062
Boras	1	31606.72021484375
Boras	3	53941.68981933594
Boras	4	48710.92053222656
Boston	2	74994.240234375
Boston	3	15344.640014648438
Boston	4	63730.7802734375
Brickhaven	1	31474.7802734375
Brickhaven	2	7277.35009765625
Brickhaven	3	114974.53967285156
Brickhaven	4	11528.52978515625
Bridgewater	2	75778.99060058594
Bridgewater	4	26115.800537109375
Brisbane	1	16118.479858398438
Brisbane	3	34100.030029296875
Bruxelles	1	18800.089721679688
Bruxelles	2	8411.949829101562
Bruxelles	3	47760.479736328125
Burbank	1	37850.07958984375
Burbank	4	8234.559936523438
Burlingame	1	13529.570190429688
Burlingame	3	42031.83020019531
Burlingame	4	65221.67004394531
Cambridge	1	21782.699951171875
Cambridge	2	14380.920043945312
Cambridge	3	48828.71942138672
Cambridge	4	54251.659912109375
Charleroi	1	16628.16015625
Charleroi	2	1711.260009765625
Charleroi	3	1637.199951171875

```

Philadelphia 4 116503.07043457031
Reggio Emilia 2 41509.94006347656
Reggio Emilia 3 56421.650390625
Reggio Emilia 4 44669.740478515625
Reims 1 52029.07043457031
Reims 2 18971.959716796875
Reims 3 15146.31982421875
Reims 4 48895.59014892578
Salzburg 2 98104.24005126953
Salzburg 3 6693.2802734375
Salzburg 4 45001.10986328125
San Diego 1 87489.23010253906
San Francisco 1 72899.19995117188
San Francisco 4 151459.4805908203
San Jose 2 160010.27026367188
San Rafael 1 267315.2586669922
San Rafael 2 7261.75
San Rafael 3 216297.40063476562
San Rafael 4 163983.64880371094
Sevilla 4 54723.621154785156
Singapore 1 28395.18994140625
Singapore 2 92033.77014160156
Singapore 3 90250.07995605469
Singapore 4 77809.37023925781
South Brisbane 1 21730.029907226562
South Brisbane 3 10640.290161132812
South Brisbane 4 27098.800048828125
Stavern 1 54701.999755859375
Stavern 4 61897.19006347656
Strasbourg 2 80438.47985839844
Torino 3 94117.25988769531
Toulouse 1 15139.1201171875
Toulouse 3 17251.08056640625
Toulouse 4 38098.240234375
Tsawassen 2 31302.500244140625
Tsawassen 3 43332.349609375
Vancouver 4 75238.91955566406
Versailles 1 5759.419921875
Versailles 4 59074.90026855469
White Plains 4 85555.98962402344
Time taken: 14.906 seconds, Fetched: 182 row(s)

```

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

```

select year_id, month_id, sum (QUANTITYORDERED) as QTY_ORDERED from
sales_order_data_orc group by year_id, month_id order by QTY_ORDERED desc;

```

```

year_id month_id qty_ordered
2004 11 10678
2003 11 10179
2003 10 5515
2004 10 5483
2004 8 4564
2005 5 4357
2005 3 3852
2004 12 3804
2005 1 3395
2005 2 3393
2004 1 3245
2004 7 3174
2004 9 3171
2004 2 3061
2004 6 2971
2005 4 2634
2004 5 2618
2003 9 2510
2003 12 2489
2004 4 2077
2003 5 2017
2003 4 1993
2004 3 1978
2003 8 1974
2003 3 1755
2003 7 1725
2003 6 1649
2003 2 1449
2003 1 1357
Time taken: 30.728 seconds, Fetched: 29 row(s)

```

