**Hive Class 4 Assignment**

**(11-09-2022)**

1. Downloaded vechile sales data

=> https://github.com/shashank-mishra219/Hive-Class/blob/main/sales\_order\_data.csv

2. Store raw data into hdfs location

=> /tmp/hive\_class/sales\_order\_data.csv

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

=> Database name:- hive\_class\_assignment

|  |
| --- |
| create table sales\_order\_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"); |
|  |  |

4. Load data from hdfs path into "sales\_order\_csv"

=> load data local inpath 'file:///tmp/hive\_class/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"

|  |
| --- |
| create table sales\_order\_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; |

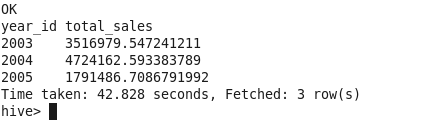
6. Load data from "sales\_order\_csv" into "sales\_order\_orc"

=> from sales\_order\_csv insert overwrite table sales\_order\_orc select \*;

**Perform below mentioned queries on "sales\_order\_orc" table :**

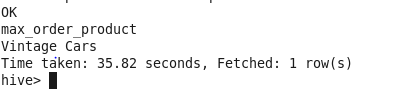
**a. Calculate total sales per year**

=> Select year\_id, sum(sales) as total\_sales from sales\_order\_orc group by year\_id;



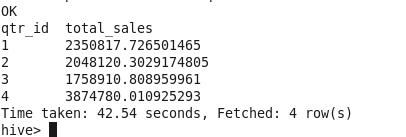
**b. Find a product for which maximum orders were placed**

=> Select max(productline) as max\_order\_product from sales\_order\_orc where status='Shipped';



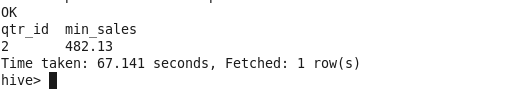
**c. Calculate the total sales for each quarter**

=> Select qtr\_id, sum(sales) as total\_sales from sales\_order\_orc group by qtr\_id;



**d. In which quarter sales was minimum**

=> Select qtr\_id, min(sales) as min\_sales from sales\_order\_orc group by qtr\_id order by min\_sales asc limit 1;

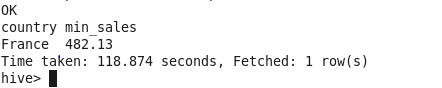


**e. In which country sales was maximum and in which country sales was minimum**

=> Select country, max(sales) as max\_sales from sales\_order\_orc group by country order by max\_sales desc limit 1;

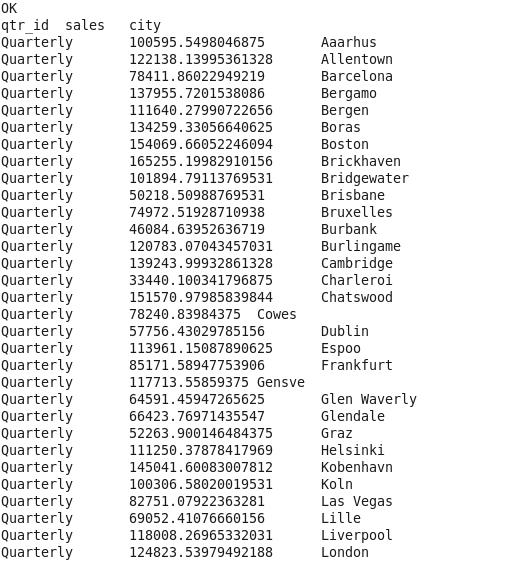


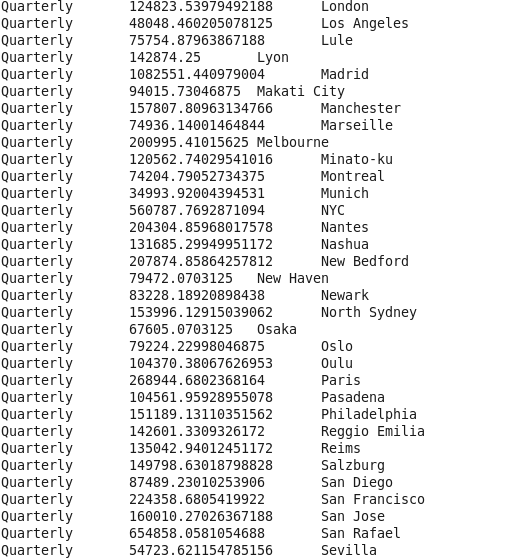
=> Select country, min(sales) as min\_sales from sales\_order\_orc group by country order by min\_sales asc limit 1;

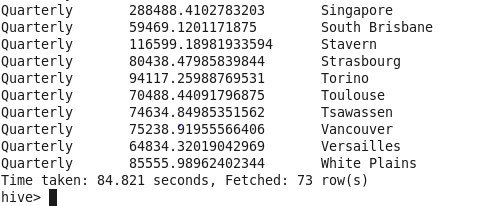


**f. Calculate quarterly sales for each city**

=> Select 'Quarterly' as qtr\_id, sum(sales) as sales, city from sales\_order\_orc group by city;







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

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

=> Select month\_id as MONTH, year\_id, max(quantityordered) as max\_sold\_quantities from sales\_order\_orc where status='Shipped' group by year\_id, month\_id;

