Data: > https://github.com/shashank-mishra219/Hive-Class/blob/main/sales_order_data.csv

1)Store raw data into hdfs location.

Solution:

To store raw data in hdfs location we have to use this command.

My data is stored in local system (location of data in my system is on desktop) and I am storing this data in hdfs root directory location.

Syntax: hdfs fs -copyFromLocal localsystemlocation location data to stored

```
Command: hdfs fs -copyFromLocal Desktop/sales_order_csv  /
```

2)Create an internal hive table "sales_order_csv" which will store csv data sales_order_csv ... make sure to skip header row while creating table.

Solution:

To create an internal hive table of sales_order_csv, we have to use the following command.

```
create table sales_order_csv (

ordernumber int, quantityordered int, priceeach int, orderlinenumber int, sales int, status string, qtr_id int, month_id int , year_id int,
productline int , 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 terminated by ‘,’  stored as text;
```

3) Load data from hdfs path into "sales_order_csv".

Solution:

To load data from hdfs path into sales_order_csv we have use the following command:

```
Command: load data inpath ‘/sales_order_csv’ into table sales_order_csv
```

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

Solution:

```
create table sales_order_orc (

ordernumber int, quantityordered int, priceeach int, orderlinenumber int, sales int, status string, qtr_id int, month_id int , year_id int,
productline int , 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 terminated by ‘,’  stored as orc;
```

5) Load data from "sales_order_csv" into "sales_order_orc

Solution: **insert overwrite table sales_order_orc select * from sales_order_csv;**

6) Perform below mentioned queries on "sales_order_orc" table

a. Calculate total sales per year

```
   select year_id, sum(sales) from sales_order_orc group by year_id;
```

b. Find a product for which maximum orders were placed

```
select product_line, sum(quantityordered) as maxordered from sales_order_orc group by product_line
order by maxordered desc limit 1;
```

c. Calculate the total sales for each quarter

```
   select qtr_id, sum(sales) from sales_order_orc group by qtr_id;
```

d. In which quarter sales was minimum

```
   select qtr_id, sum(sales) from sales_order_orc group by qtr_id order by qtr_id asc limit 1;
```

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

```
select country, sum(sales) as tot_sales from sales_order_orc group by country order by tot_sales desc
limit 1;

select country, sum(sales) as tot_sales from sales_order_orc group by country order by tot_sales asc
limit 1;
```

f. Calculate quarterly sales for each city

```
select qtr_id,city sum(sales) as qtr_sales from sales_order_orc group by qtr_id,city order by
qtr_sales;
```

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

```
select month_id, year_id, sum(quantityordered) as qty from sales_order_orc where year_id = 2003

group by year_id order by qty desc limit 1;

select month_id, year_id, sum(quantityordered) as qty from sales_order_orc where year_id = 2004

group by year_id order by qty desc limit 1;

select month_id, year_id, sum(quantityordered) as qty from sales_order_orc where year_id = 2005

group by year_id order by qty desc limit 1 ;
```