hive_class_3_assignment

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

Sol: Downloaded and transferred file into cloudera VM.

```
        [cloudera@quickstart ~]$ ls /tmp/hive_class/sales_order*
        /tmp/hive_class/sales_order_data.csv
```

2. Store raw data into hdfs location

Sol: Move data from local file system to HDFS.

```
        [cloudera@quickstart ~]$ hdfs dfs -put
        /tmp/hive_class/sales_order_data.csv /tmp/hive_class/
```

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

Sol:

```
        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 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"

Sol: load data inpath '/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"

Sol:

```
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 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 ','
stored as ORC;
```

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

Sol: from sales_order_csv insert overwrite table sales_order_orc select *;

Perform below menioned queries on "sales_order_orc" table :

a. Calculate total sales per year

```
Sol:     year_id total_sales
      1   2003   3516514
      2   2004   4723531
      3   2005   1791264
```

b. Find a product for which maximum orders were placed

Sol: SELECT productcode, count(*) as cnt from sales_order_orc group by productcode ORDER BY cnt DESC limit 1;

```
productcode      cnt
S18_3232          52
```

c. Calculate the total sales for each quarter

Sol: SELECT qtr_id, sum(sales) as totalsales from sales_order_orc GROUP BY qtr_id;

```
qtr_id   totalsales
  1      2350510
  2      2047855
  3      1758673
  4      3874271
```

d. In which quarter sales was minimum

Sol:SELECT qtr_id, sum(sales) as totalsales from sales_order_orc GROUP BY qtr_id ORDER BY totalsales limit 1;

```
      qtr_id   totalsales
      3          1758673
```

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

Sol:

```
      SELECT country, sum(sales) as totalsales from sales_order_orc   GROUP
      BY country ORDER BY totalsales limit 1;
        Minimum: Ireland       57749

      SELECT country, sum(sales) as totalsales from sales_order_orc GROUP BY
      country ORDER BY totalsales desc limit 1;
       Maximum: USA    3627511
```

f. Calculate quartelry sales for each city

Sol:

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

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

Sol:

```
select month_id, year_id, max(quantityordered) as quantity from sales_order_orc group by
year_id, month_id order by year_id;
```

Note: I am not able perform last query properly, I am getting all records of each month and year