Hive_class_Assignment_1

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

2. Store raw data into hdfs location

```
hdfs dfs -mkdir /data
hdfs dfs -put /config/workspace/sales_order_data.csv /data
hdfs dfs -cat /data/sales_order_data.csv                    # (to see content)
```

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

Creating data schema:

```
CREATE TABLE sales_order_csv (
  ORDERNUMBER INT,
  QUANTITYORDERED INT,
  PRICEEACH FLOAT,
  ORDERLINENUMBER INT,
  SALES INT,
  STATUS VARCHAR(15),
  QTR_ID INT,
  MONTH_ID INT,
  YEAR_ID INT,
  PRODUCTLINE VARCHAR(15),
  MSRP INT,
  PRODUCTCODE VARCHAR(15),
  PHONE VARCHAR(15),
  CITY VARCHAR(15),
  STATE VARCHAR(2),
  POSTALCODE VARCHAR(10),
  COUNTRY VARCHAR(10),
  TERRITORY VARCHAR(10),
  CONTACTLASTNAME VARCHAR(15),
  CONTACTFIRSTNAME VARCHAR(15),
  DEALSIZE VARCHAR(15),
  PRIMARY KEY (ORDERNUMBER) DISABLE NOVALIDATE
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
TBLPROPERTIES ("skip.header.line.count"="1")
```

LOCATION '/hdfs/data';   # (optional)

OR

create table sales_order_csv(ORDERNUMBER int,QUANTITYORDERED int,PRICEEACH float,ORDERLINENUMBER int,SALES int,STATUS varchar(15),QTR_ID int, MONTH_ID int,YEAR_ID int,PRODUCTLINE varchar(15),MSRP int,PRODUCTCODE varchar(15),PHONE varchar(15),CITY varchar(15),STATE varchar(2), POSTALCODE varchar(10),COUNTRY varchar(10),TERRITORY varchar(10),CONTACTLASTNAME varchar(15), CONTACTFIRSTNAME varchar(15),DEALSIZE varchar(15), PRIMARY KEY (ORDERNUMBER)) row format delimited field terminated by ‘,’ STORED AS TEXTFILE LOCATION ‘/hdfs/data/sales_order_data.csv';

4. Load data from hdfs path into HIVE table: ”sales_order_csv"

LOAD DATA INPATH '/hdfs/data/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 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
)
STORED AS ORC
TBLPROPERTIES ("skip.header.line.count"="1");;
```

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

```
INSERT INTO TABLE sales_order_orc
SELECT * FROM sales_order_csv;
```

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;
```

b. Find a product for which maximum orders were placed

```
select PRODUCTLINE, a from
(
select PRODUCTLINE, count(*) as a
from sales_order_orc
group by 1
)
where a = (select max(b) from (select PRODUCTLINE, count(*) as b from
sales_order_orc group by 1));
```

c. Calculate the total sales for each quarter

```
select Year_ID,
sum(case when Month_ID > 0 and Month_ID < 4 then sales else 0 end) as
first_quarter,
sum(case when Month_ID > 3 and Month_ID < 7 then sales else 0 end) as
second_quarter,
sum(case when Month_ID > 6 and Month_ID < 10 then sales else 0 end) as
third_quarter,
sum(case when Month_ID > 9 and Month_ID < 13 then sales else 0 end) as
fourth_quarter
```

```
from sales_order_orc
group by Year_ID;
```

d. In which quarter sales was minimum

```
WITH CTE as
(
select Year_ID, sales,
CASE
when Month_ID > 0 and Month_ID < 4 then 1
when Month_ID > 3 and Month_ID < 7 then 2
when Month_ID > 6 and Month_ID < 10 then 3
when Month_ID > 9 and Month_ID < 13 then 4
else 0
END as Quarter_number
from sales_order_orc
group by Year_ID
)
select Year_ID, Quarter_number, sum(sales) as max_sale_in this_quarter from CTE
group by Year_ID, Quarter_number;
```

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

```
with CTE as
(
select year_ID, country, sum(sales) as total_sales
from sales_order_orc
group by year_ID, country
order by year_ID desc
)
select year_ID,
case when total_sales = (select min(total_sales) from CTE group by year_ID,
country) then country else NULL end as country_with_min_sales,
case when total_sales = (select max(total_sales) from CTE group by year_ID,
country) then country else NULL end as country_with_max_sales
from CTE
group by year_ID;
```

f. Calculate quarterly sales for each city

```
select Year_ID, QTR_ID, city, sum(sales) from sales_order_orc
group by Year_ID, QTR_ID, city;
```

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

```
select year_id, month, sum(quantityordered) from sales_order_orc
group by year_id, month
HAVING sum(quantityordered) = (select max(quantityordered) from
sales_order_orc group by year_id, month)
order by year_id desc;
```