1. Download vehicle sales data -> https://github.com/shashank-mishra219/Hive-Class/blob/main/sales\_order\_data.csv

Ans. Downloaded

2. Store raw data into hdfs location

Ans. docker cp sales\_order\_data.csv namenode:sales\_order\_data.csv

docker exec -it namenode bash

hdfs dfs -mkdir -p /hiveass1/sales

hdfs dfs -put sales\_order\_data.csv /hiveass1/sales/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

Ans.

docker exec -it hive-server bash

hive

create database hiveass1;

use hiveass1;

create table sales\_orders\_csv

(

ORDERNUMBER INT,

QUANTITYORDERED INT,

PRICEEACH FLOAT,

ORDERLINENUMBER FLOAT,

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 INT,

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"

Ans. load data inpath '/hiveass1/sales/' into table sales\_orders\_csv;

5. Create an internal hive table which will store data in ORC format "sales\_order\_orc"

Ans.

create table sales\_orders\_orc

(

ORDERNUMBER INT,

QUANTITYORDERED INT,

PRICEEACH FLOAT,

ORDERLINENUMBER FLOAT,

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 INT,

TERRITORY STRING,

CONTACTLASTNAME STRING,

CONTACTFIRSTNAME STRING,

DEALSIZE STRING

)

stored as orc;

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

Ans. from sales\_orders\_csv insert overwrite table sales\_orders\_orc select \*;

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

a. Calculate total sales per year.

select year\_id, sum(sales) from sales\_orders\_orc group by year\_id;

b. Find a product for which maximum orders were placed

select productcode, sum(quantityordered) as sum\_orders from sales\_orders\_orc

group by productcode

order by sum\_orders desc limit 1;

c. Calculate the total sales for each quarter

Ans:

**Overall only quarters**: select qtr\_id, sum(sales) from sales\_orders\_orc group by qtr\_id;

**Year wise quarters:** select year\_id, qtr\_id, sum(sales) from sales\_orders\_orc group by year\_id, qtr\_id;

d. In which quarter sales was minimum

Ans:

**Overall only quarters**: select qtr\_id, sum(sales) as sumsales from sales\_orders\_orc group by qtr\_id order by sumsales asc limit 1;

**Year wise quarters:** select year\_id, qtr\_id, sum(sales) as sumsales from sales\_orders\_orc group by year\_id, qtr\_id order by sumsales asc limit 1;

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

select mt.\* from(select nt.\*, row\_number() over (order by nt.ss desc) rw

from (select territory, sum(sales) ss from sales\_orders\_orc group by territory) nt)mt

where

mt.rw = 1 or

mt.rw =(select count(distinct territory) from sales\_orders\_orc);

f. Calculate quarterly sales for each city

select city,qtr\_id, sum(sales) from sales\_orders\_orc group by city, qtr\_id order by city;

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

select b.\* from (select t.\*, row\_number() over (partition by t.year\_id order by t.ss desc) rw

from (select year\_id, month\_id, sum(quantityordered) ss

from sales\_orders\_orc group by year\_id, month\_id)t)b

where b.rw=1;