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 -copyFromLocal '/config/workspace/sales_order_data.csv' '/tmp/hive/'
```

```
abc@49b8dc972401:~/workspace$ hdfs dfs -copyFromLocal '/config/workspace/sales_order_data.csv' '/tmp/hive/'
2023-02-28 12:28:19,446 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, re
moteHostTrusted = false
abc@49b8dc972401:~/workspace$ hdfs dfs -ls /tmp/hive/
Found 4 items
drwx-wx-wx   - root supergroup          0 2023-02-28 12:24 /tmp/hive/_resultscache_
drwx------   - abc  supergroup          0 2023-02-28 12:24 /tmp/hive/abc
drwx------   - root supergroup          0 2023-02-28 12:25 /tmp/hive/root
-rw-r--r--   1 abc  supergroup     360233 2023-02-28 12:28 /tmp/hive/sales_order_data.csv
abc@49b8dc972401:~/workspace$
```

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

```
Create table sales_order_data (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");
```

```
      >
      >
      > Create table sales_order_data (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');
OK
Time taken: 1.598 seconds
hive>
```

4. Load data from hdfs path into "sales_order_csv"

```
load data inpath '/tmp/hive/sales_order_data.csv' into table sales_order_data;
```

```
hive> load data inpath '/tmp/hive/sales_order_data.csv' into table sales_order_data;
Loading data to table default.sales_order_data
OK
Time taken: 0.975 seconds
hive> select * from sales_order_data limit 3;
OK
10107   30      95      2       2871    Shipped 1       2       2003    Motorcycles     95      S10_1678        2125557818
NYC     NY      10022   USA     NA      Yu      Kwai    Small
10121   34      81      5       2765    Shipped 2       5       2003    Motorcycles     95      S10_1678        26.47.1555
Reims           51100   France  EMEA    Henriot Paul    Small
10134   41      94      2       3884    Shipped 3       7       2003    Motorcycles     95      S10_1678        +33 1 46 6
2 7555  Paris           75508   France  EMEA    Da Cunha        Daniel  Medium
Time taken: 1.777 seconds, Fetched: 3 row(s)
hive>
```

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` 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 SERDE
  'org.apache.hadoop.hive.ql.io.orc.OrcSerde'
STORED AS INPUTFORMAT
  'org.apache.hadoop.hive.ql.io.orc.OrcInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.orc.OrcOutputFormat'
TBLPROPERTIES (
  'bucketing_version'='2')
```

```
      >
      > 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 SERDE
      >   'org.apache.hadoop.hive.ql.io.orc.OrcSerde'
      > STORED AS INPUTFORMAT
      >   'org.apache.hadoop.hive.ql.io.orc.OrcInputFormat'
      > OUTPUTFORMAT
      >   'org.apache.hadoop.hive.ql.io.orc.OrcOutputFormat'
      > TBLPROPERTIES (
      >   'bucketing_version'='2')
      > ;
OK
Time taken: 0.135 seconds
hive>
```

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

```
insert overwrite table sales_order_orc select * from sales_order_data;
```

```
hive> insert overwrite table sales_order_orc select * from sales_order_data;
Query ID = abc_20230228155914_d7de1a61-cb52-4c97-95c5-a048e87c962b
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1677579642712_0002, Tracking URL = http://0720aabb5fde:8088/proxy/application_1677579642712_0002/
Kill Command = /usr/local/hadoop/bin/mapred job  -kill job_1677579642712_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2023-02-28 15:59:30,958 Stage-1 map = 0%,  reduce = 0%
2023-02-28 15:59:43,590 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 30.97 sec
2023-02-28 15:59:51,928 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 35.61 sec
MapReduce Total cumulative CPU time: 35 seconds 610 msec
Ended Job = job_1677579642712_0002
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://localhost/user/hive/warehouse/sales_order_orc/.hive-staging_hive_2023-02-28_15-59-14_406_3
570009613546042105-1/-ext-10000
Loading data to table default.sales_order_orc
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 35.61 sec   HDFS Read: 399243 HDFS Write: 43264 SUCCESS
Total MapReduce CPU Time Spent: 35 seconds 610 msec
OK
Time taken: 41.952 seconds
hive>
```

Perform below menioned queries on "sales_order_orc" table :

a. Calculatye total sales per year

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

```
hive> select year_id, count(sales) from sales_order_orc group by year_id;
Query ID = abc_20230228160201_0cb9b1e0-83b3-426f-b344-b75c50323471
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1677579642712_0003, Tracking URL = http://0720aabb5fde:8088/proxy/application_1677579642712_0003/
Kill Command = /usr/local/hadoop/bin/mapred job  -kill job_1677579642712_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2023-02-28 16:02:17,118 Stage-1 map = 0%,  reduce = 0%
2023-02-28 16:02:28,594 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 6.76 sec
2023-02-28 16:02:36,952 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 12.02 sec
MapReduce Total cumulative CPU time: 12 seconds 20 msec
Ended Job = job_1677579642712_0003
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 12.02 sec   HDFS Read: 39647 HDFS Write: 152 SUCCESS
Total MapReduce CPU Time Spent: 12 seconds 20 msec
OK
2003    1000
2004    1345
2005    478
Time taken: 37.926 seconds, Fetched: 3 row(s)
hive>
```

b. Find a product for which maximum orders were placed

```
select PRODUCTLINE, sum(QUANTITYORDERED) from sales_order_orc group by
PRODUCTLINE order by 2 desc;
```

c. Calculate the total sales for each quarter

```
select year_id,month_id, qtr_id, sum(sales) from sales_order_orc group by
year_id, month_id, qtr_id order by 4 desc;
```

```
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 7.61 s
Total MapReduce CPU Time Spent: 16 seconds 360 msec
OK
2004    11      4       1088908
2003    11      4       1029698
2003    10      4       568223
2004    10      4       552852
2004    8       3       461437
2005    5       2       457808
2005    3       1       374212
2004    12      4       372747
2005    2       1       358142
2005    1       1       339494
2004    7       3       327106
2004    9       3       320703
2004    1       1       316538
2004    2       1       311378
2004    6       2       286633
2004    5       2       273406
2003    9       3       263937
2003    12      4       261843
2005    4       2       261608
2004    4       2       206114
2004    3       1       205709
2003    4       2       201580
2003    8       3       197780
2003    5       2       192644
2003    7       3       187710
2003    3       1       174483
2003    6       2       168062
2003    2       1       140819
2003    1       1       129735
Time taken: 75.938 seconds, Fetched: 29 row(s)
hive>
```

d. In which quarter sales was minimum

```
Select year_id, month_id, qtr_id, sales from sales_order_orc where sales =
(select min(sales) from sales_order_orc);
```

```
Ended Job = job_1677585148235_0003
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 11.73 sec   HDFS Read: 38174 HDFS Write: 116
Stage-Stage-3: Map: 1   Cumulative CPU: 6.66 sec   HDFS Read: 8527 HDFS Write: 112 SUCCESS
Total MapReduce CPU Time Spent: 18 seconds 390 msec
OK
2005    5       2       482
Time taken: 88.161 seconds, Fetched: 1 row(s)
hive>
```

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

```
Select country, sales from sales_order_orc where sales = (select max(sales)
from sales_order_orc) union all Select country, sales from sales_order_orc
where sales = (select min(sales) from sales_order_orc)
```

f. Calculate quartelry sales for each city

```
select city,year_id,month_id, qtr_id, sum(sales) from sales_order_orc group
by city,year_id, month_id, qtr_id order by 1,2,3,4 desc;
```

```
Total MapReduce CPU Time Spent: 19 seconds 160 msec
OK
Aaarhus 2003    10      4       40316
Aaarhus 2004    10      4       60267
Allentown       2004    7       3       27148
Allentown       2004    8       3       44776
Allentown       2004    11      4       44038
Allentown       2005    5       2       6166
Barcelona       2003    4       2       4219
Barcelona       2003    11      4       44003
Barcelona       2004    11      4       30179
Bergamo 2003    2       1       56172
Bergamo 2003    11      4       40071
Bergamo 2004    11      4       41691
Bergen  2003    11      4       95266
Bergen  2004    9       3       16361
Boras   2003    10      4       48704
Boras   2004    9       3       53933
Boras   2005    3       1       31603
Boston  2003    12      4       63724
Boston  2004    4       2       7128
Boston  2004    6       2       19544
Boston  2004    9       3       15342
Boston  2005    5       2       48310
Brickhaven      2003    6       2       7276
Brickhaven      2003    9       3       34986
Brickhaven      2004    7       3       15109
Brickhaven      2004    8       3       52498
Brickhaven      2004    9       3       12364
Brickhaven      2004    12      4       11527
Brickhaven      2005    1       1       31470
Bridgewater     2003    11      4       26113
Bridgewater     2004    6       2       44126
Bridgewater     2005    5       2       31645
Brisbane        2003    9       3       34094
```

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

```
Select year_id, month_id, max(month_sale) from (Select year_id, month_id,
sum(QUANTITYORDERED) as month_sale  from sales_order_orc Group by year_id,
month_id) a;
```