HIVE_CLASS_3_ASSIGNMENT_SOLUTION -NARAYAN ZEERMIRE

ASSIGNMENT BY INUERON.AI

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 (file is present on root directory)

hadoop fs -put 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

```
create table sales_order_csv
(
ORDER_NUMBER int,QUANTITY_ORDERED int,PRICE_EACH int,ORDER_LINE_NUMBER int,
SALES int,STATUS string,QTR_ID int,MONTH_ID int,YEAR_ID int,PRODUCT_LINE string,
MSRP int,PRODUCT_CODE string,PHONE string,CITY string,STATE string,POSTAL_CODE int,
COUNTRY string,TERRITORY string,CONTACT_LAST_NAME string,CONTACT_FIRST_NAME string,
DEAL_SIZE string
)
row format delimited
fields terminated by ',';
```

4. Load data from hdfs path into "sales_order_csv"

```
LOAD DATA INPATH '/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 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 int,

COUNTRY string,TERRITORY string,CONTACTLASTNAME string,CONTACTFIRSTNAME string,

DEALSIZE string

)

stored as ORC;

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. Calculatye total sales per year

select year_ID as year,sum(sales) as Total_Sales from sales_order_csv group by year_ID ;

```
hive> select year_ID as year,sum(sales) as Total_Sales from sales_order_csv  group by year_ID ;
Query ID = hadoop_20230518170458_88c757ac-78b3-4b16-8d77-9c948c128d3d
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1684409950512_0012)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 5.11 s
----------------------------------------------------------------------------------------------
OK
2003    3516514
NULL    NULL
2004    4723531
2005    1791264
Time taken: 8.785 seconds, Fetched: 4 row(s)
```

b. Find a product for which maximum orders were placed

select PRODUCT_LINE from (select PRODUCT_LINE ,sum(QUANTITY_ORDERED) max from sales_order_csv group by PRODUCT_LINE order by max desc limit 1) a ;

```
--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 4.43 s
--------------------------------------------------------------------------------
OK
Classic Cars
Time taken: 10.332 seconds, Fetched: 1 row(s)
```

c. Calculate the total sales for each quarter

select sum(sales) as Total_sales,QTR_ID from sales_order_csv group by QTR_ID order by QTR_ID;

```
--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 4.66 s
--------------------------------------------------------------------------------
OK
NULL    NULL
2350510 1
2047855 2
1758673 3
3874271 4
Time taken: 5.235 seconds, Fetched: 5 row(s)
```

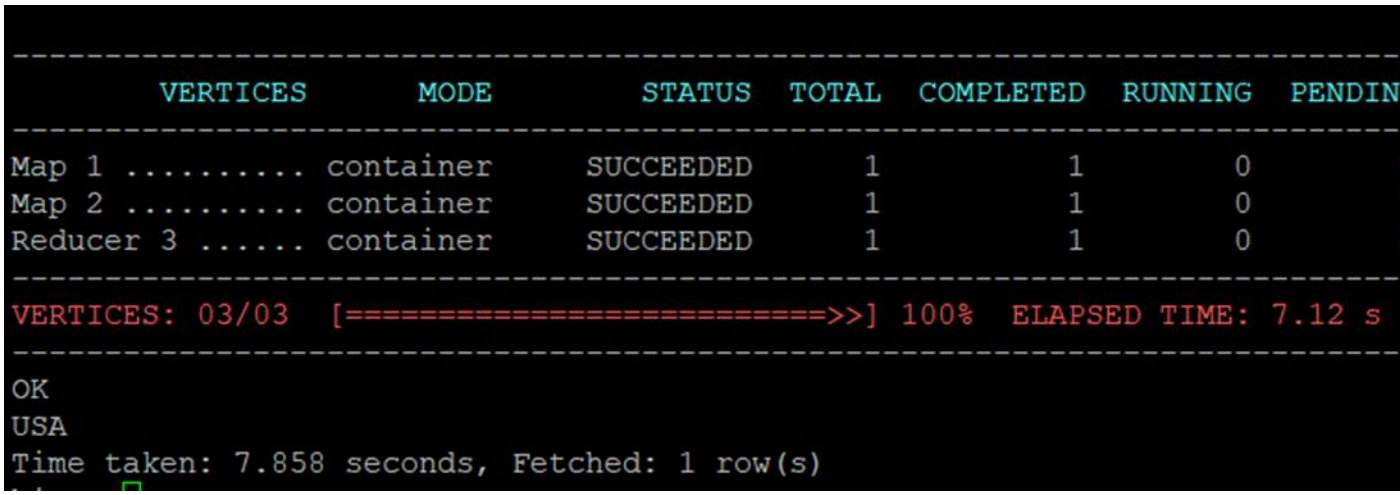
d. In which quarter sales was minimum

> select sum(sales) as sales_order, qtr_id from sales_order_orc group by qtr_id order by sales_order limit 1;

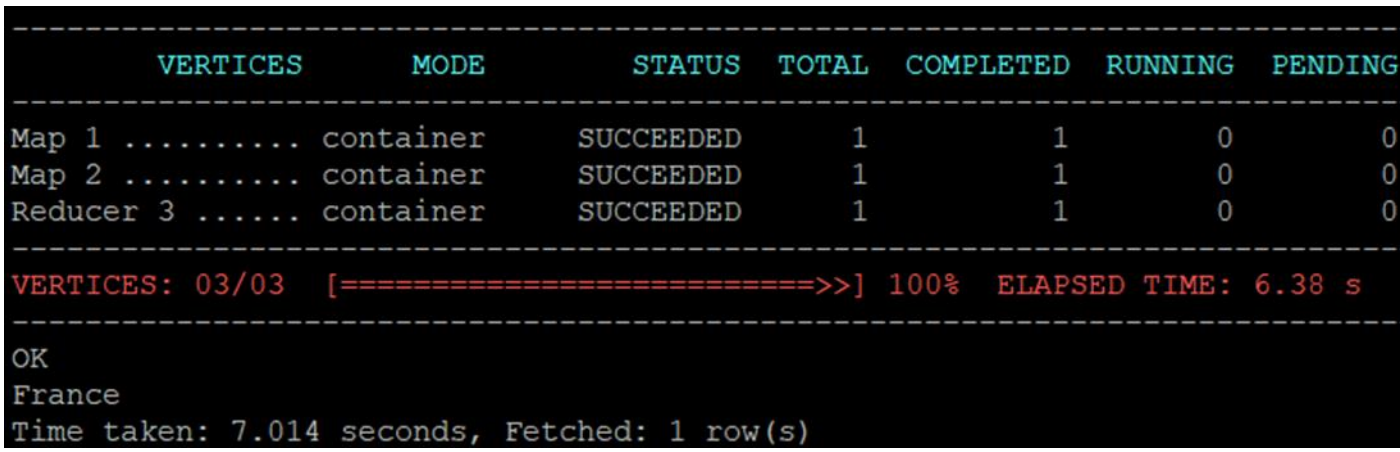
```
MapReduce Total cumulative CPU time: 11 seconds 300 msec
Ended Job = job_1663306393822_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 12.08 sec   HDFS Read: 31640 HDFS Write: 184 SUCCESS
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 11.3 sec   HDFS Read: 5066 HDFS Write: 10 SUCCESS
Total MapReduce CPU Time Spent: 23 seconds 380 msec
OK
1758673 3
Time taken: 186.641 seconds, Fetched: 1 row(s)
hive>
```

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

select s1.COUNTRY as Country_max_sales from sales_order_csv s1 where s1.sales in (select max(s2.sales) from sales_order_csv s2);



select s1.COUNTRY as Country_min_sales from sales_order_csv s1 where s1.sales in (select min(s2.sales) from sales_order_csv s2);



f. Calculate quartelry sales for each city

hive>select quartelry_sales,city from (select sum(s.sales) as quartelry_sales,s.QTR_ID,s.city from sales_order_csv s group by s.QTR_ID,s.city)a;

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

select month_id,year_id,QUANTITY_ORDERED from (select distinct month_id,year_id,QUANTITY_ORDERED ,dense_rank() over(partition by year_id order by QUANTITY_ORDERED desc) as rnk from sales_order_csv)a where a.rnk = 1;

```
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1684409950512_0014)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      2          2        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 5.55 s
--------------------------------------------------------------------------------
OK
NULL    NULL    NULL
2       2003    50
4       2003    50
6       2003    50
9       2003    50
10      2003    50
11      2003    50
11      2004    55
1       2003    50
3       2003    50
5       2003    50
4       2005    97
Time taken: 10.838 seconds, Fetched: 12 row(s)
```