# Assignment – 1

(Name: - Trijit Adhikary, Batch: -1)

## 1) Store raw data into hdfs location: -

```
[cloudera@quickstart assignData]$ hdfs dfs -put '/home/cloudera/Desktop/trijit/hive/assignData/sales_order_data_filtered.csv' '/trijit/hive/'
```

## 2) Create an internal hive table “sales_order _csv”

```
hive (assignment)> create table sales_order_csv (
                 > order_id int,quant int,price_each int,sales double,month int, year int,product string,city string,country string)
                 > row format delimited
                 > fields terminated by ','
                 > lines terminated by '\n'
                 > tblproperties("skip.header.line.count" = "1");
OK
Time taken: 1.059 seconds
hive (assignment)>
```

## 3) load data from hdfs path to “sales_order _csv”

```
hive (assignment)> load data inpath '/trijit/hive/sales_order_data_filtered.csv' into table sales_order_csv;
Loading data to table assignment.sales_order_csv
Table assignment.sales_order_csv stats: [numFiles=1, totalSize=159938]
OK
Time taken: 1.54 seconds
hive (assignment)> select * from sales_order_csv limit 5;
OK
10107   30      95      2871.0  2       2003    Motorcycles     NYC     USA
10121   34      81      2765.9  5       2003    Motorcycles     Reims   France
10134   41      94      3884.34 7       2003    Motorcycles     Paris   France
10145   45      83      3746.7  8       2003    Motorcycles     Pasadena        USA
10159   49      100     5205.27 10      2003    Motorcycles     San Francisco   USA
Time taken: 0.875 seconds, Fetched: 5 row(s)
hive (assignment)>
```

## 4) Create an internal hive table “sales_order _orc”

```
hive (assignment)> create table sales_order_orc (
                 > order_id int,quant int,price_each int,sales double,month int, year int,product string,city string,country string)
                 > stored as orc;
OK
Time taken: 0.338 seconds
```

## 5) load data from “sales_order _csv” into “sales_order _orc”

```
hive (assignment)> from sales_order_csv insert overwrite table sales_order_orc select *;
Query ID = cloudera_20220913222020_3e2e8089-6546-443c-817f-bc8c50a91ea2
```

## Queries: -

### a. Calculate total sales per year

```
hive (assignment)> select year, sum(sales) as total_sales from sales_order_orc group by year;
Query ID = cloudera_20220913222727_f4d03ae6-7688-4f77-9311-a59bdc97c3e9
Total jobs = 1
Launching Job 1 out of 1
```

```
OK
year    total_sales
2003    3516979.540000001
2004    4724162.599999997
2005    1791486.71
Time taken: 54.637 seconds,
```

### b. Find a product for which maximum orders were placed

```
hive (assignment)> select product, sum(quant) as total_quant  from sales_order_orc group by product order by total_quant desc limit 1;
Query ID = cloudera_20220913224343_a5b64f4f-6913-48d1-a066-ae64de5cb845
Total jobs = 2
```

```
product total_quant
Classic Cars    33992
Time taken: 87.171 seconds, Fetched: 1 row(s)
hive (assignment)>
```

### c. Calculate the total sales for each quarter

```
hive (assignment)> select concat(year,"-Q",((floor((month-1)/3))+1)) as qt_year,sum(sales) as total_sales from sales_order_orc group by concat(year,"-Q",((floor((month-1)/3
))+1));
```

```
qt_year total_sales
2003-Q1 445094.6900000002
2003-Q2 562365.22
2003-Q3 649514.5399999999
2003-Q4 1860005.0899999987
2004-Q1 833730.6800000005
2004-Q2 766260.7299999996
2004-Q3 1109396.2700000005
2004-Q4 2014774.9199999995
2005-Q1 1071992.3600000003
2005-Q2 719494.3500000001
Time taken: 51.087 seconds, Fetched: 10 row(s)
hive (assignment)>
```

## d. In which quarter sales was minimum

```
create view temp_min_quar as select concat(year,"-Q",((floor((month-1)/3))+1)) as qt_year,sum(sales) as total_sales from sales_order_orc group by concat(year,"-Q",((floor
((month-1)/3))+1));

create view temp_min_quar_ranked as select *, row_number() over(partition by substr(qt_year,1,4) order by total_sales) as rowNum from temp_min_quar;

select qt_year,total_sales from temp_min_quar_ranked where rowNum = 1;
```

```
qt_year total_sales
2003-Q1 445094.6900000002
2004-Q2 766260.7299999996
2005-Q2 719494.3500000001
Time taken: 93.149 seconds, Fetched: 3 row(s)
hive>
```

## e. In which country sales was maximum

```
hive (assignment)> select country, sum(sales) as total_sales from sales_order_orc group by country order by total_sales desc limit 1;
Query ID = cloudera_20220914000606_06aa934f-b0f3-419a-9e70-1140923fa0d8
```

```
country total_sales
USA     3627982.83
Time taken: 111.165 seconds, Fetched: 1 row(s)
hive (assignment)>
```

## f. In which country sales was minimum

```
hive (assignment)> select country, sum(sales) as total_sales from sales_order_orc group by country order by total_sales limit 1;
Query ID = cloudera_20220914001313_f132fa4f-4cdb-4df9-a5c3-7390dad29f62
```

```
country total_sales
Ireland 57756.43
Time taken: 99.727 seconds, Fetched: 1 row(s)
hive (assignment)>
```

## g. Quarterly sales of each city

```
hive (assignment)> select concat(city,"-",year,"-Q",((floor((month-1)/3))+1)) as qt_year,
                 > sum(sales) as total_sales
                 > from sales_order_orc
                 > group by concat(city,"-",year,"-Q",((floor((month-1)/3))+1));
Query ID = cloudera_20220914004343_695db8fb-665e-436d-bd23-6c3166583961
Total jobs = 1
```

```
qt_year total_sales
Aaarhus-2003-Q4 40321.60999999999
Aaarhus-2004-Q4 60273.939999999995
Allentown-2004-Q3       71930.61
Allentown-2004-Q4       44040.73
Allentown-2005-Q2       6166.8
Barcelona-2003-Q2       4219.2
Barcelona-2003-Q4       44009.31
Barcelona-2004-Q4       30183.35
Bergamo-2003-Q1 56181.32
Bergamo-2003-Q4 40077.71000000001
Bergamo-2004-Q4 41696.689999999995
Bergen-2003-Q4  95277.18000000001
Bergen-2004-Q3  16363.1
Boras-2003-Q4   48710.92
Boras-2004-Q3   53941.69
Boras-2005-Q1   31606.72
Boston-2003-Q4  63730.780000000006
Boston-2004-Q2  26677.35
Boston-2004-Q3  15344.640000000001
Boston-2005-Q2  48316.89
Brickhaven-2003-Q2      7277.35
Brickhaven-2003-Q3      34992.399999999994
Brickhaven-2004-Q3      79982.14
Brickhaven-2004-Q4      11528.53
Brickhaven-2005-Q1      31474.78
Bridgewater-2003-Q4     26115.800000000003
Bridgewater-2004-Q2     44130.520000000004
Bridgewater-2005-Q2     31648.469999999998
Brisbane-2003-Q3        34100.03
Brisbane-2004-Q1        16118.48
Bruxelles-2004-Q1       18800.09
Bruxelles-2004-Q3       47760.48
Bruxelles-2005-Q2       8411.95
```

```
Bruxelles-2005-Q2       8411.95
Burbank-2003-Q4 8234.560000000001
Burbank-2004-Q1 37850.079999999994
Burlingame-2003-Q3      42031.83
Burlingame-2003-Q4      62305.469999999994
Burlingame-2004-Q4      2916.2
Burlingame-2005-Q1      13529.57
Cambridge-2004-Q1       21782.699999999997
Cambridge-2004-Q2       14380.92
Cambridge-2004-Q3       48828.72
Cambridge-2004-Q4       54251.66
Charleroi-2003-Q2       1711.26
Charleroi-2003-Q3       1637.2
Charleroi-2004-Q4       13463.48
Charleroi-2005-Q1       16628.16
Chatswood-2003-Q3       28397.260000000002
Chatswood-2004-Q3       41297.14
Chatswood-2004-Q4       37905.149999999994
Chatswood-2005-Q2       43971.43
Cowes-2004-Q1   26906.68
Cowes-2004-Q4   51334.16
Dublin-2004-Q1  38784.47
Dublin-2004-Q3  18971.96
Espoo-2003-Q3   31569.429999999993
Espoo-2004-Q2   31018.230000000003
Espoo-2005-Q1   51373.490000000005
Frankfurt-2003-Q1       11432.34
Frankfurt-2003-Q4       27257.79
Frankfurt-2004-Q1       37266.49
Frankfurt-2004-Q4       9214.970000000001
Gensve-2004-Q1  50432.55
Gensve-2004-Q3  67281.01000000001
Glen Waverly-2003-Q4    37878.55
Glen Waverly-2004-Q3    12334.82
Glen Waverly-2005-Q2    14378.09
```

```
Glen Waverly-2005-Q2    14378.09
Glendale-2003-Q2        20350.95
Glendale-2003-Q3        7600.12
Glendale-2003-Q4        5142.15
Glendale-2004-Q4        29343.35
Glendale-2005-Q1        3987.2
Graz-2003-Q4    43488.73999999999
Graz-2005-Q1    8775.16
Helsinki-2003-Q4        42083.5
Helsinki-2004-Q3        42744.06
Helsinki-2005-Q1        26422.82
Kobenhavn-2003-Q1       58871.11
Kobenhavn-2004-Q2       36079.01
Kobenhavn-2004-Q4       24078.610000000004
Kobenhavn-2005-Q2       26012.870000000003
Koln-2003-Q4    31363.18
Koln-2004-Q4    68943.40000000001
Las Vegas-2003-Q2       33847.619999999995
Las Vegas-2004-Q3       34453.85
Las Vegas-2004-Q4       14449.61
Lille-2003-Q4   48874.280000000006
Lille-2004-Q1   20178.129999999997
Liverpool-2003-Q4       26797.210000000003
Liverpool-2004-Q2       50408.25
Liverpool-2005-Q2       40802.810000000005
London-2003-Q2  32376.289999999997
London-2003-Q4  70230.13
London-2004-Q1  8477.220000000001
London-2004-Q4  13739.900000000001
Los Angeles-2003-Q4     24159.14
Los Angeles-2004-Q1     23889.32
Lule-2003-Q1    9749.0
Lule-2004-Q4    66005.88
Lyon-2003-Q4    41535.10999999999
Lyon-2004-Q1    101339.14000000001
```

```
Lyon-2004-Q1    101339.14000000001
Madrid-2003-Q1  44621.95999999999
Madrid-2003-Q2  100689.02999999997
Madrid-2003-Q3  47727.81999999999
Madrid-2003-Q4  112573.32999999997
Madrid-2004-Q1  105491.34000000003
Madrid-2004-Q2  119656.04
Madrid-2004-Q3  21986.27
Madrid-2004-Q4  203007.48
Madrid-2005-Q1  207555.19
Madrid-2005-Q2  119242.98000000001
Makati City-2003-Q1     55245.020000000004
Makati City-2003-Q4     22841.96
Makati City-2004-Q4     15928.75
Manchester-2003-Q1      51017.91999999999
Manchester-2004-Q4      106789.89
Marseille-2003-Q2       52481.840000000004
Marseille-2004-Q4       20136.859999999997
Marseille-2005-Q1       2317.44
Melbourne-2003-Q2       60135.840000000004
Melbourne-2004-Q1       49637.57
Melbourne-2004-Q4       91222.00000000001
Minato-ku-2004-Q2       25928.750000000004
Minato-ku-2004-Q4       55888.65000000001
Minato-ku-2005-Q1       38191.39
Minato-ku-2005-Q2       553.95
Montreal-2003-Q4        15947.29
Montreal-2004-Q2        24564.53
Montreal-2005-Q2        33692.97
Munich-2004-Q3  34993.92
NYC-2003-Q1     32647.81
NYC-2003-Q2     93239.56000000001
NYC-2003-Q4     89600.84000000001
NYC-2004-Q2     71860.78
NYC-2004-Q3     63027.919999999984
```

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

```
hive> create view temp_sales as select year, month, sum(quant) total_quant from sales_order_orc group by year,month order by total_quant desc;
OK
Time taken: 1.503 seconds
hive> create view sales_ranked as select year,month,total_quant, row_number() over(partition by year order by total_quant desc) as rowNum from temp_sales;
OK
Time taken: 0.846 seconds
hive> select * from sales_ranked where rowNum=1;
```

```
2003    11      10179   1
2004    11      10678   1
2005    5       4357    1
Time taken: 202.437 seconds, Fetched: 3 row(s)
hive>
```