

## **PySpark Project**

```
from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("Sales_Analysis").getOrCreate()

rdd=spark.sparkContext.textFile("hdfs://localhost:9000/user/training/Sales/Sales_Records.csv")
rdd2 = rdd.collect()
rdd3 = spark.sparkContext.parallelize(rdd2[1:])
rdd4 = rdd3.repartition(3)

rdd5 = rdd4.map(lambda x : x.split(','))

rdd5.cache()
print(rdd5.is_cached)
rdd5.getNumPartitions()
```

### **1. Display the number of countries present in the data.**

```
rdd6 = rdd5.map(lambda x : (x[1],1))

rdd7 = rdd6.reduceByKey(lambda x,y : x + y)

rdd7.map(lambda x : x[0]).count()

rdd7.saveAsTextFile('hdfs://localhost:9000/user/training/spark/project/question1')
```

### **2. Display the number of units sold in each region.**

```
rdd6 = rdd5.map(lambda x : (x[0],x[8]))
```

```
rdd7 = rdd6.reduceByKey(lambda x,y : int(x)+int(y))
```

### **3. Display the 10 most recent sales.**

```
def convert_date(var):  
    return datetime.strptime(var,'%m/%d/%Y').date()
```

```
rdd6 = rdd5.map(lambda x : (x[0],x[1],x[2],x[3],x[4],convert_date(x[5]))).take(10)
```

### **4. Display the products with atleast 2 occurrences of 'a'**

```
new_rdd = rdd5
```

```
def function(var):  
    var1 = list(var)  
    count = 0  
    for i in var1:  
        if 'a' == i:  
            count += 1  
    return count
```

```
rdd6 = new_rdd.filter(lambda x : function(x[2]) > 1)  
rdd6.take(10)
```

### **5.Display country in each region with highest units sold. (Using spark)**

```
def function(var):  
    dict = {}  
    for i in var:  
        if i in dict.keys():
```

```
dict[i] += 1
```

```
else:
```

```
dict[i] = 1
```

```
rdd6 = rdd5.map(lambda x : ((x[0],x[1]),x[8]))
```

```
rdd7 = rdd6.reduceByKey(lambda x,y : int(x)+int(y))
```

```
rdd8 = rdd7.map(lambda x : (x[0][0],(x[0][1],x[1]))).reduceByKey(lambda a,b : max(a,b ,key=lambda  
x : x[1])).collect()
```

## **6. Display the unit price and unit cost of each item in ascending order.**

```
rdd6 = rdd5.map(lambda x : [x[10], x[11]]).sortBy(lambda x : x[1], ascending = True)
```

```
rdd6.saveAsTextFile('hdfs://localhost:9000/user/training/spark/project/question6')
```

## **7. Display the number of sales yearwise. (Using pyspark)**

```
def convert_date(var):
```

```
    variable = var.split('/')

```

```
    return variable[-1]
```

```
rdd6 = rdd5.map(lambda x : (convert_date(x[7])))
```

```
rdd7 = rdd6.map(lambda x : (x , 1)).reduceByKey(lambda x,y : x+y)
```

```
rdd7.saveAsTextFile('hdfs://localhost:9000/user/training/spark/project/question7')
```

## 8. Display the number of orders for each item.

```
rdd6 = rdd5.map(lambda x : x[2]).map(lambda x : (x, 1)).reduceByKey(lambda x,y : x+ y)
```

```
rdd6.saveAsTextFile('hdfs://localhost:9000/user/training/spark/project/question8')
```

```
miles@MILE-BL-4824-LAP: $ hadoop fs -ls hdfs://localhost:9000/user/training/spark/project
Found 1 items
drwxr-xr-x  - miles supergroup          0 2023-03-17 14:51 hdfs://localhost:9000/user/training/spark/project/question8
miles@MILE-BL-4824-LAP: $ hadoop fs -ls hdfs://localhost:9000/user/training/spark/project/question8
Found 4 items
-rw-r--r--  3 miles supergroup          0 2023-03-17 14:51 hdfs://localhost:9000/user/training/spark/project/question8/_SUCCESS
-rw-r--r--  3 miles supergroup         89 2023-03-17 14:51 hdfs://localhost:9000/user/training/spark/project/question8/part-00000
-rw-r--r--  3 miles supergroup         55 2023-03-17 14:51 hdfs://localhost:9000/user/training/spark/project/question8/part-00001
-rw-r--r--  3 miles supergroup         79 2023-03-17 14:51 hdfs://localhost:9000/user/training/spark/project/question8/part-00002
miles@MILE-BL-4824-LAP: $ hadoop fs -ls hdfs://localhost:9000/user/training/spark/project/question8/part-00000
-rw-r--r--  3 miles supergroup         89 2023-03-17 14:51 hdfs://localhost:9000/user/training/spark/project/question8/part-00000
miles@MILE-BL-4824-LAP: $ hadoop fs -cat hdfs://localhost:9000/user/training/spark/project/question8/part-00000
('Baby Food', 445)
('Snacks', 398)
('Clothes', 386)
('Personal Care', 415)
('Meat', 399)
miles@MILE-BL-4824-LAP: $ hadoop fs -cat hdfs://localhost:9000/user/training/spark/project/question8/part-00001
('Fruits', 447)
('Household', 424)
('Vegetables', 410)
miles@MILE-BL-4824-LAP: $ hadoop fs -cat hdfs://localhost:9000/user/training/spark/project/question8/part-00002
('Beverages', 447)
('Cereal', 385)
('Office Supplies', 420)
('Cosmetics', 424)
```