**Name: Jeslyn Ho Ka Yan   
Date: 9 Nov 2024**

**Assignment 3, Task 3**

**Terminal 1 (Hadoop)**

bigdata@bigdata-VirtualBox:~$ cd ISIT312/A3

# List files in the current directory to confirm the presence of sales.txt

bigdata@bigdata-VirtualBox:~/ISIT312/A3$ ls

sales.txt solution1.hb solution1.hb~ solution2.hb solution2.hb~ task22024S4.hb

# Upload the sales.txt file from your local file system to HDFS

bigdata@bigdata-VirtualBox:~/ISIT312/A3$ $HADOOP\_HOME/bin/hadoop fs -put sales.txt

# List files in the HDFS root directory to verify the upload

bigdata@bigdata-VirtualBox:~/ISIT312/A3$ $HADOOP\_HOME/bin/hadoop fs -ls

Found 2 items

drwxr-xr-x - bigdata supergroup 0 2017-07-03 01:33 .sparkStaging

-rw-r--r-- 1 bigdata supergroup 180 2024-11-10 21:56 sales.txt

# Display the contents of sales.txt in HDFS to confirm the data is correct

bigdata@bigdata-VirtualBox:~/ISIT312/A3$ $HADOOP\_HOME/bin/hadoop fs -cat sales.txt

bolt 45

washer 3

screw 67

screw 23

nail 5

screw 78

coupler 36

bolt 5

bolt 1

drill 1

drill 1

file 36

file 28

washer 56

washer 7

bolt 10

saw 2

coupler 50

plier 20

2nd Terminal (SPARK)

//Question1=========================================================

// Load the contents of the sales.txt file as an RDD

scala> val salesRDD = spark.read.textFile("file:///home/bigdata/ISIT312/A3/sales.txt").rdd

salesRDD: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[4] at rdd at <console>:23

// Map each line to a tuple containing part name and quantity as an integer

scala> val partQuantities = salesRDD.map(line => {

| val parts = line.split(" ")

| (parts(0), parts(1).toInt)

| })

partQuantities: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[5] at map at <console>:25

// Use reduceByKey to sum quantities per part in the RDD

scala> val totalSalesRDD = partQuantities.reduceByKey(\_ + \_)

totalSalesRDD: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[6] at reduceByKey at <console>:25

// Collect and print each part and its total quantity from the RDD

scala> totalSalesRDD.collect().foreach(println)

(plier,20)

(screw,168)

(nail,5)

(washer,66)

(coupler,86)

(bolt,61)

(saw,2)

(file,64)

(drill,2)

// Question2=========================================================

// Define a case class for Sale with partName and quantity fields

scala> case class Sale(partName: String, quantity: Int)

defined class Sale

// Import implicits to enable the use of .toDS() and other Dataset functions

scala> import spark.implicits.\_

import spark.implicits.\_

// Map RDD lines to Sale case class and convert to Dataset

scala> val salesDS = salesRDD.map(\_.split(" ")).map(attributes => Sale(attributes(0), attributes(1).trim.toInt)).toDS()

salesDS: org.apache.spark.sql.Dataset[Sale] = [partName: string, quantity: int]

// Group by partName and sum quantities in the Dataset

scala> val dsResult = salesDS.groupBy("partName").sum("quantity")

dsResult: org.apache.spark.sql.DataFrame = [partName: string, sum(quantity): bigint]

// Show the resulting grouped and summed Dataset

scala> dsResult.show()

+--------+-------------+

|partName|sum(quantity)|

+--------+-------------+

| saw| 2|

| washer| 66|

| bolt| 61|

| coupler| 86|

| nail| 5|

| file| 64|

| screw| 168|

| drill| 2|

| plier| 20|

+--------+-------------+

//Question3 =========================================================

// Map RDD lines to Sale case class and convert to DataFrame

scala> val salesDF = salesRDD.map(\_.split(" ")).map(attributes => Sale(attributes(0), attributes(1).trim.toInt)).toDF()

salesDF: org.apache.spark.sql.DataFrame = [partName: string, quantity: int]

// Register the DataFrame as a temporary SQL view

scala> salesDF.createOrReplaceTempView("SalesData")

// Use SQL to group by partName and sum quantities from the SalesData view

scala> val sqlDF = spark.sql("SELECT partName, SUM(quantity) as total\_quantity FROM SalesData GROUP BY partName")

sqlDF: org.apache.spark.sql.DataFrame = [partName: string, total\_quantity: bigint]

// Show the resulting grouped and summed DataFrame from SQL query

scala> sqlDF.show()

+--------+--------------+

|partName|total\_quantity|

+--------+--------------+

| saw| 2|

| washer| 66|

| bolt| 61|

| coupler| 86|

| nail| 5|

| file| 64|

| screw| 168|

| drill| 2|

| plier| 20|

+--------+--------------+