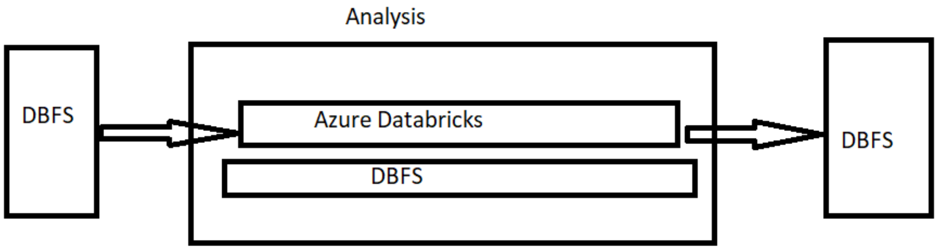
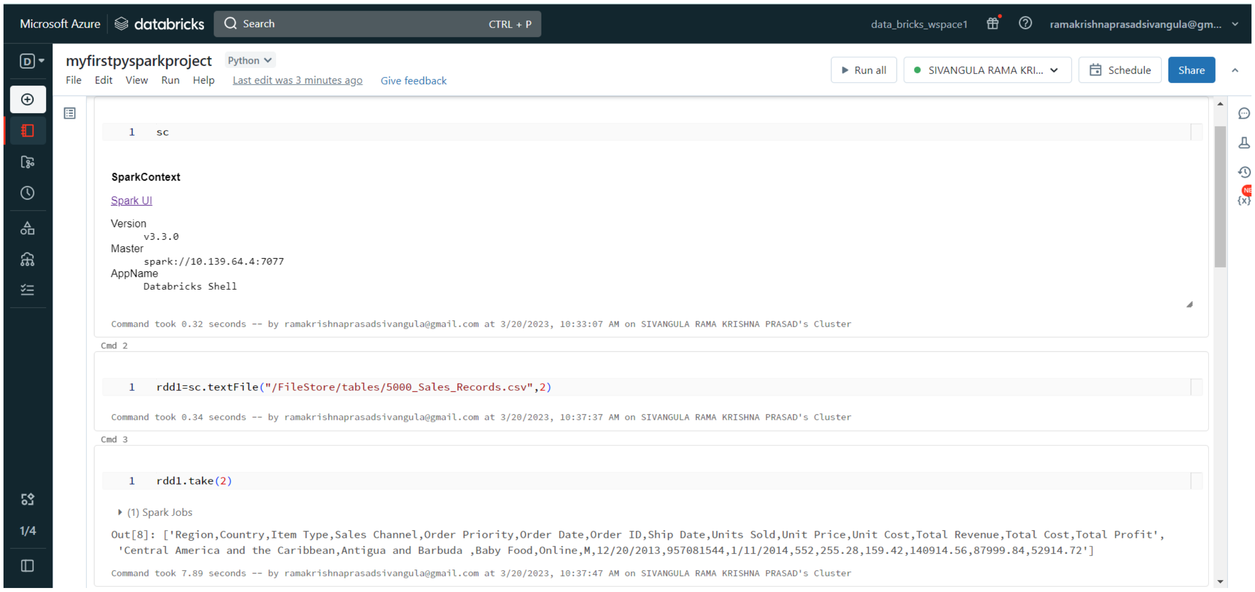
Pyspark Sales Analytics using Databricks



Loading data from DBFS :



```
sc
```

SparkContext

Spark UI

```
Version
     v3.3.0
Master
     spark://10.139.64.4:7077
AppName
     Databricks Shell

Command took 0.32 seconds -- by ramakrishnaprasadsivangula@gmail.com at 3/20/2023, 10:33:07 AM on SIVANGULA RAMA KRISHNA PRASAD's Cluster
```

Cmd 2

```
rdd1=sc.textFile("/FileStore/tables/5000_Sales_Records.csv",2)
```

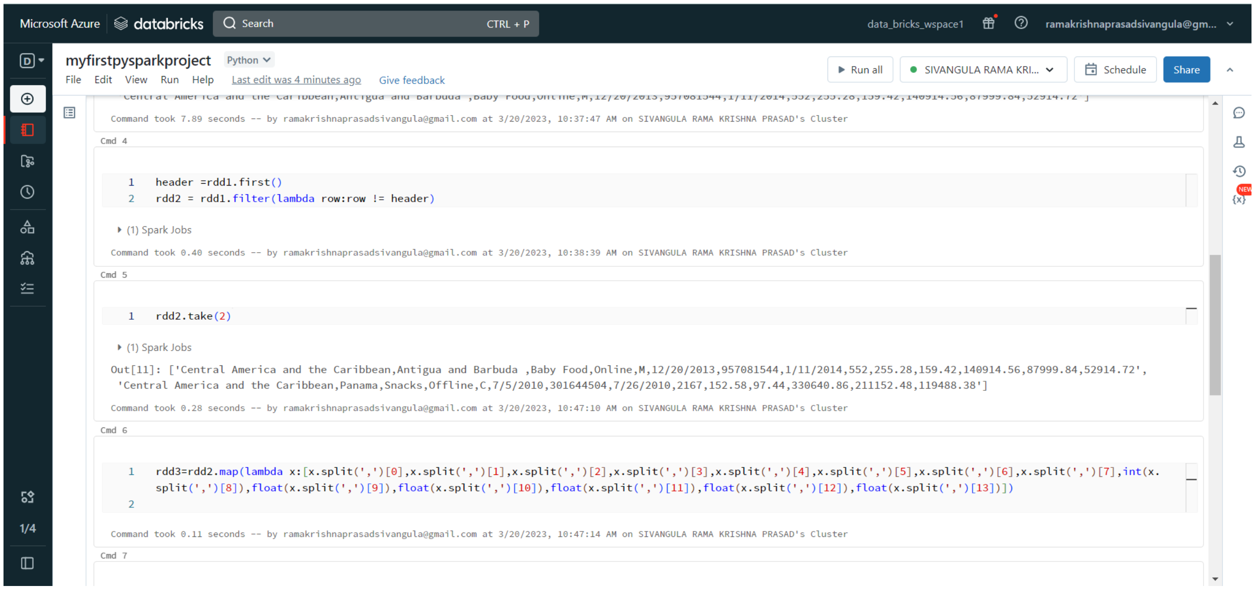
```
Command took 0.34 seconds -- by ramakrishnaprasadsivangula@gmail.com at 3/20/2023, 10:37:37 AM on SIVANGULA RAMA KRISHNA PRASAD's Cluster
```

Cmd 3

```
rdd1.take(2)
```

```
▸ (1) Spark Jobs
Out[8]: ['Region,Country,Item Type,Sales Channel,Order Priority,Order Date,Order ID,Ship Date,Units Sold,Unit Price,Unit Cost,Total Revenue,Total Cost,Total Profit',
 'Central America and the Caribbean,Antigua and Barbuda ,Baby Food,Online,M,12/20/2013,957081544,1/11/2014,552,255.28,159.42,140914.56,87999.84,52914.72']
Command took 7.89 seconds -- by ramakrishnaprasadsivangula@gmail.com at 3/20/2023, 10:37:47 AM on SIVANGULA RAMA KRISHNA PRASAD's Cluster
```

Removing Header and type casting required columns:



```
 'Central America and the Caribbean,Antigua and Barbuda ,Baby Food,Online,M,12/20/2013,957081544,1/11/2014,552,255.28,159.42,140914.56,87999.84,52914.72']
Command took 7.89 seconds -- by ramakrishnaprasadsivangula@gmail.com at 3/20/2023, 10:37:47 AM on SIVANGULA RAMA KRISHNA PRASAD's Cluster
```

Cmd 4

```python
header =rdd1.first()
rdd2 = rdd1.filter(lambda row:row != header)
```

```
(1) Spark Jobs
Command took 0.40 seconds -- by ramakrishnaprasadsivangula@gmail.com at 3/20/2023, 10:38:39 AM on SIVANGULA RAMA KRISHNA PRASAD's Cluster
```

Cmd 5

```python
rdd2.take(2)
```

```
(1) Spark Jobs
Out[11]: ['Central America and the Caribbean,Antigua and Barbuda ,Baby Food,Online,M,12/20/2013,957081544,1/11/2014,552,255.28,159.42,140914.56,87999.84,52914.72',
 'Central America and the Caribbean,Panama,Snacks,Offline,C,7/5/2010,301644504,7/26/2010,2167,152.58,97.44,330640.86,211152.48,119488.38']
Command took 0.28 seconds -- by ramakrishnaprasadsivangula@gmail.com at 3/20/2023, 10:47:10 AM on SIVANGULA RAMA KRISHNA PRASAD's Cluster
```

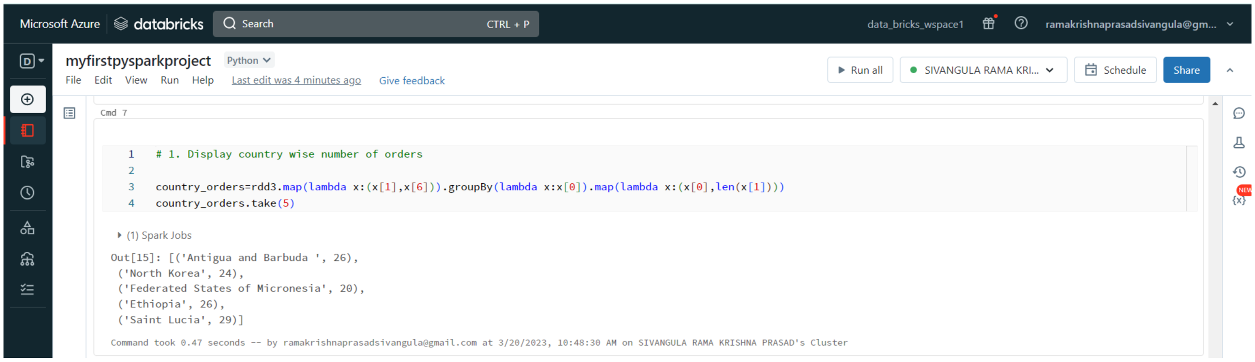
Cmd 6

```python
rdd3=rdd2.map(lambda x:[x.split(',')[0],x.split(',')[1],x.split(',')[2],x.split(',')[3],x.split(',')[4],x.split(',')[5],x.split(',')[6],x.split(',')[7],int(x.split(',')[8]),float(x.split(',')[9]),float(x.split(',')[10]),float(x.split(',')[11]),float(x.split(',')[12]),float(x.split(',')[13])])

```

```
Command took 0.11 seconds -- by ramakrishnaprasadsivangula@gmail.com at 3/20/2023, 10:47:14 AM on SIVANGULA RAMA KRISHNA PRASAD's Cluster
```

Cmd 7

Problem Statement1: Display country wise number of orders

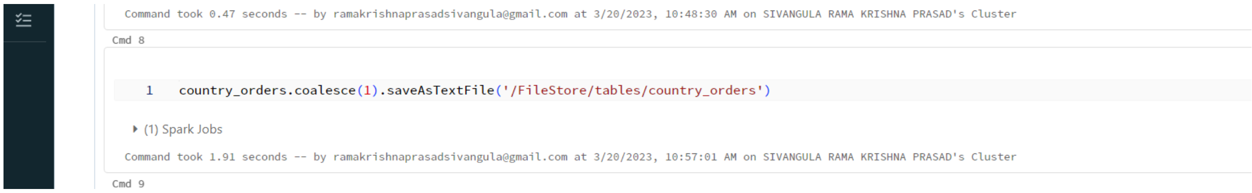


Cmd 7

```python
# 1. Display country wise number of orders

country_orders=rdd3.map(lambda x:(x[1],x[6])).groupBy(lambda x:x[0]).map(lambda x:(x[0],len(x[1])))
country_orders.take(5)
```

```
(1) Spark Jobs
Out[15]: [('Antigua and Barbuda ', 26),
 ('North Korea', 24),
 ('Federated States of Micronesia', 20),
 ('Ethiopia', 26),
 ('Saint Lucia', 29)]
Command took 0.47 seconds -- by ramakrishnaprasadsivangula@gmail.com at 3/20/2023, 10:48:30 AM on SIVANGULA RAMA KRISHNA PRASAD's Cluster
```

Saving the result to DBFS



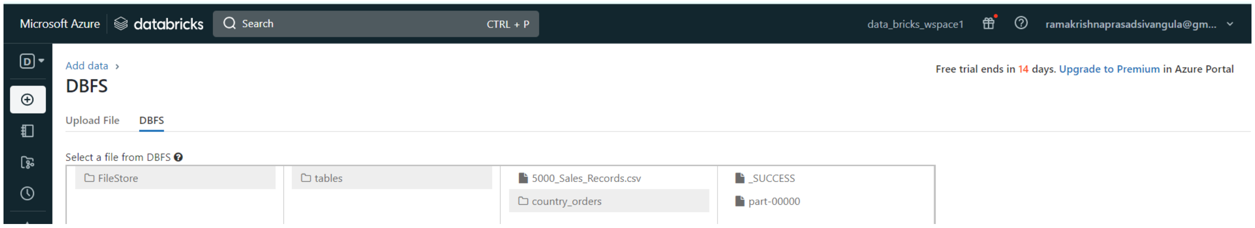
```
Command took 0.47 seconds -- by ramakrishnaprasadsivangula@gmail.com at 3/20/2023, 10:48:30 AM on SIVANGULA RAMA KRISHNA PRASAD's Cluster
```

Cmd 8

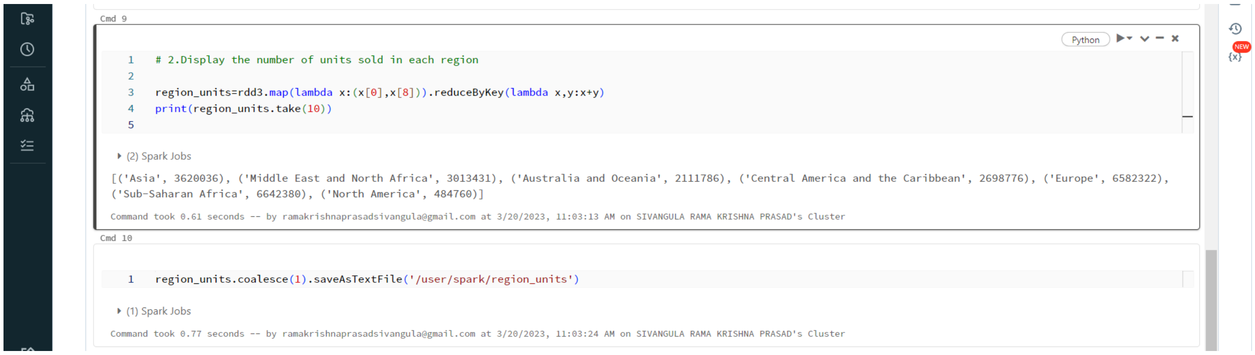
```python
country_orders.coalesce(1).saveAsTextFile('/FileStore/tables/country_orders')
```

```
(1) Spark Jobs
Command took 1.91 seconds -- by ramakrishnaprasadsivangula@gmail.com at 3/20/2023, 10:57:01 AM on SIVANGULA RAMA KRISHNA PRASAD's Cluster
```

Cmd 9



Problem Statement2: Display the number of units sold in each region



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

region_units=rdd3.map(lambda x:(x[0],x[8])).reduceByKey(lambda x,y:x+y)
print(region_units.take(10))

```

(2) Spark Jobs

```
[('Asia', 3620036), ('Middle East and North Africa', 3013431), ('Australia and Oceania', 2111786), ('Central America and the Caribbean', 2698776), ('Europe', 6582322),
('Sub-Saharan Africa', 6642380), ('North America', 484760)]
```

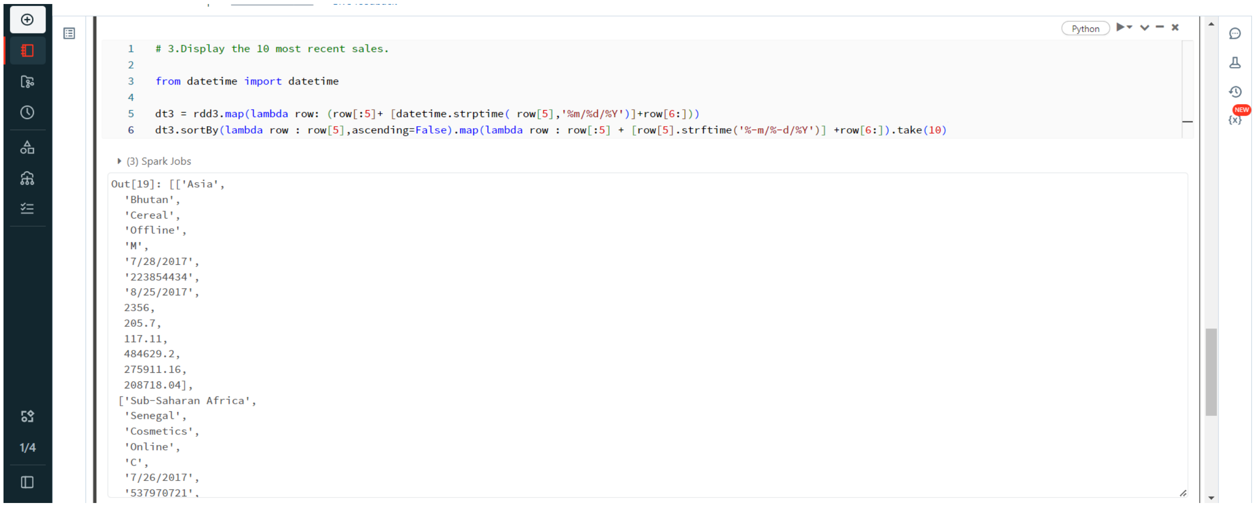
Command took 0.61 seconds -- by ramakrishnaprasadsivangula@gmail.com at 3/20/2023, 11:03:13 AM on SIVANGULA RAMA KRISHNA PRASAD's Cluster

```
region_units.coalesce(1).saveAsTextFile('/user/spark/region_units')
```

(1) Spark Jobs

Command took 0.77 seconds -- by ramakrishnaprasadsivangula@gmail.com at 3/20/2023, 11:03:24 AM on SIVANGULA RAMA KRISHNA PRASAD's Cluster

Problem Statement3:Display the 10 most recent sales.

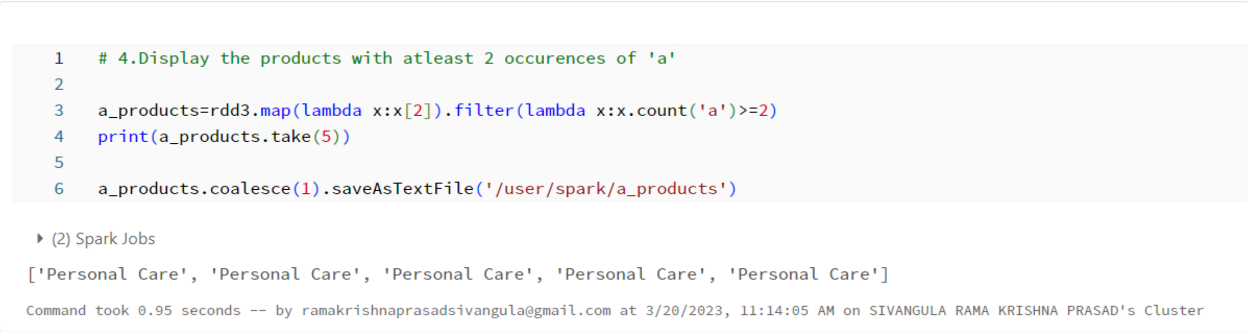


```
# 3.Display the 10 most recent sales.

from datetime import datetime

dt3 = rdd3.map(lambda row: (row[:5]+ [datetime.strptime( row[5],'%m/%d/%Y')]+row[6:]))
dt3.sortBy(lambda row : row[5],ascending=False).map(lambda row : row[:5] + [row[5].strftime('%-m/%-d/%Y')] +row[6:]).take(10)
```

(3) Spark Jobs

```
Out[19]: [['Asia',
  'Bhutan',
  'Cereal',
  'Offline',
  'M',
  '7/28/2017',
  '223854434',
  '8/25/2017',
  2356,
  205.7,
  117.11,
  484629.2,
  275911.16,
  208718.04],
 ['Sub-Saharan Africa',
  'Senegal',
  'Cosmetics',
  'Online',
  'C',
  '7/26/2017',
  '537970721',
```

Problem Statement4: Display the products with atleast 2 occurences of 'a'



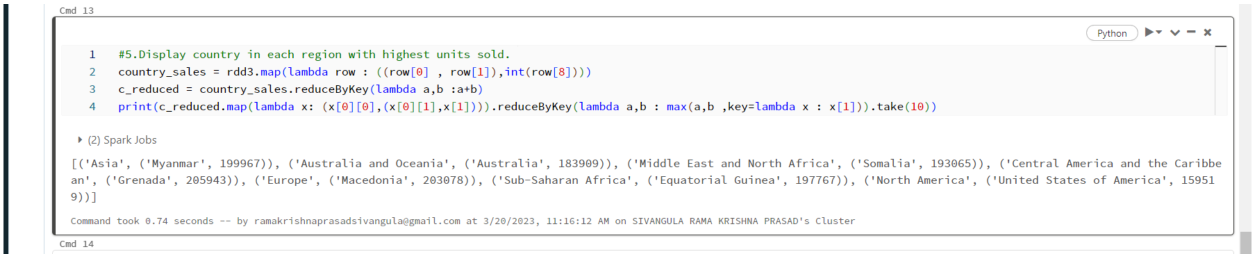
```
# 4.Display the products with atleast 2 occurences of 'a'

a_products=rdd3.map(lambda x:x[2]).filter(lambda x:x.count('a')>=2)
print(a_products.take(5))

a_products.coalesce(1).saveAsTextFile('/user/spark/a_products')
```

(2) Spark Jobs

```
['Personal Care', 'Personal Care', 'Personal Care', 'Personal Care', 'Personal Care']
```

Command took 0.95 seconds -- by ramakrishnaprasadsivangula@gmail.com at 3/20/2023, 11:14:05 AM on SIVANGULA RAMA KRISHNA PRASAD's Cluster

Problem statement5: Display country in each region with highest units sold.



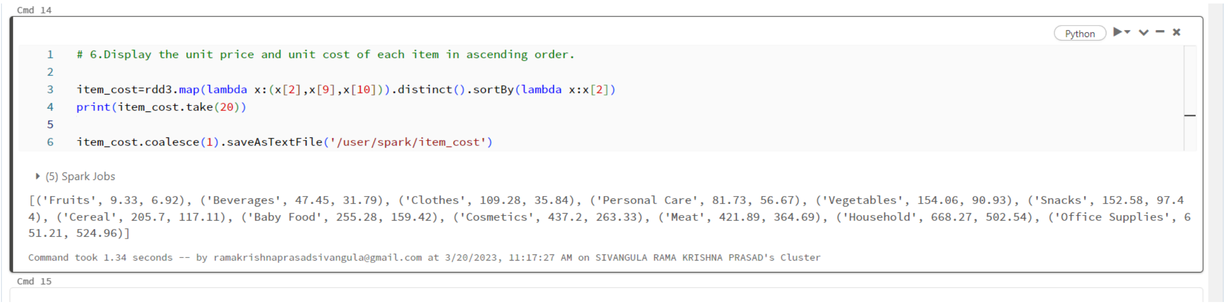
```
#5.Display country in each region with highest units sold.
country_sales = rdd3.map(lambda row : ((row[0] , row[1]),int(row[8])))
c_reduced = country_sales.reduceByKey(lambda a,b :a+b)
print(c_reduced.map(lambda x: (x[0][0],(x[0][1],x[1]))).reduceByKey(lambda a,b : max(a,b ,key=lambda x : x[1])).take(10))
```

▸ (2) Spark Jobs

```
[('Asia', ('Myanmar', 199967)), ('Australia and Oceania', ('Australia', 183909)), ('Middle East and North Africa', ('Somalia', 193065)), ('Central America and the Caribbe
an', ('Grenada', 205943)), ('Europe', ('Macedonia', 203078)), ('Sub-Saharan Africa', ('Equatorial Guinea', 197767)), ('North America', ('United States of America', 15951
9))]
```

Command took 0.74 seconds -- by ramakrishnaprasadsivangula@gmail.com at 3/20/2023, 11:16:12 AM on SIVANGULA RAMA KRISHNA PRASAD's Cluster

Problem Statement6: Display the unit price and unit cost of each item in ascending order.



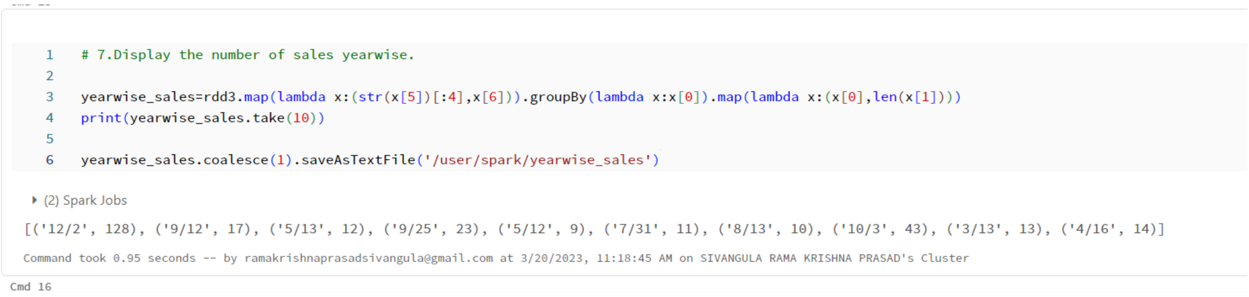
```
# 6.Display the unit price and unit cost of each item in ascending order.

item_cost=rdd3.map(lambda x:(x[2],x[9],x[10])).distinct().sortBy(lambda x:x[2])
print(item_cost.take(20))

item_cost.coalesce(1).saveAsTextFile('/user/spark/item_cost')
```

▸ (5) Spark Jobs

```
[('Fruits', 9.33, 6.92), ('Beverages', 47.45, 31.79), ('Clothes', 109.28, 35.84), ('Personal Care', 81.73, 56.67), ('Vegetables', 154.06, 90.93), ('Snacks', 152.58, 97.4
4), ('Cereal', 205.7, 117.11), ('Baby Food', 255.28, 159.42), ('Cosmetics', 437.2, 263.33), ('Meat', 421.89, 364.69), ('Household', 668.27, 502.54), ('Office Supplies', 6
51.21, 524.96)]
```

Command took 1.34 seconds -- by ramakrishnaprasadsivangula@gmail.com at 3/20/2023, 11:17:27 AM on SIVANGULA RAMA KRISHNA PRASAD's Cluster

Problem Satetement7: Display the number of sales yearwise.



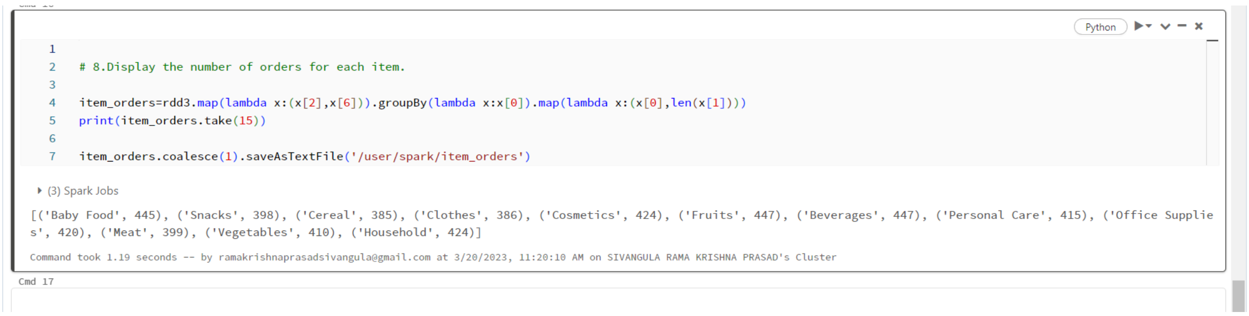
```
# 7.Display the number of sales yearwise.

yearwise_sales=rdd3.map(lambda x:(str(x[5])[:4],x[6])).groupBy(lambda x:x[0]).map(lambda x:(x[0],len(x[1])))
print(yearwise_sales.take(10))

yearwise_sales.coalesce(1).saveAsTextFile('/user/spark/yearwise_sales')
```

▸ (2) Spark Jobs

```
[('12/2', 128), ('9/12', 17), ('5/13', 12), ('9/25', 23), ('5/12', 9), ('7/31', 11), ('8/13', 10), ('10/3', 43), ('3/13', 13), ('4/16', 14)]
```

Command took 0.95 seconds -- by ramakrishnaprasadsivangula@gmail.com at 3/20/2023, 11:18:45 AM on SIVANGULA RAMA KRISHNA PRASAD's Cluster

Problem Statement8: Display the number of orders for each item.



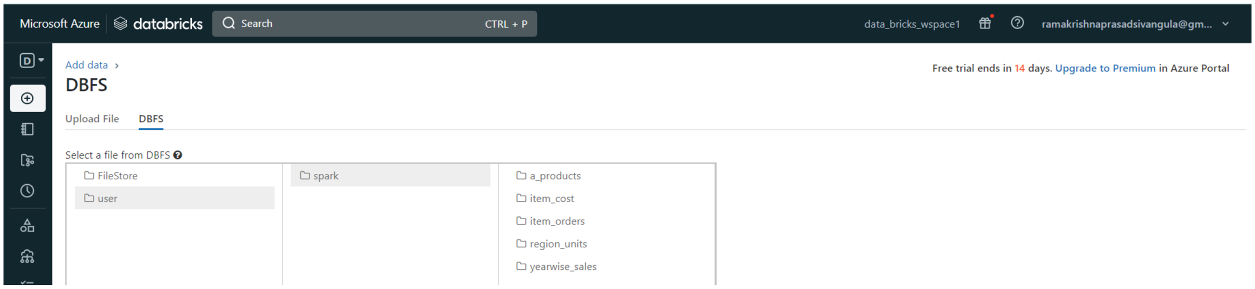
```

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

item_orders=rdd3.map(lambda x:(x[2],x[6])).groupBy(lambda x:x[0]).map(lambda x:(x[0],len(x[1])))
print(item_orders.take(15))

item_orders.coalesce(1).saveAsTextFile('/user/spark/item_orders')
```

▸ (3) Spark Jobs

```
[('Baby Food', 445), ('Snacks', 398), ('Cereal', 385), ('Clothes', 386), ('Cosmetics', 424), ('Fruits', 447), ('Beverages', 447), ('Personal Care', 415), ('Office Supplie
s', 420), ('Meat', 399), ('Vegetables', 410), ('Household', 424)]
```

Command took 1.19 seconds -- by ramakrishnaprasadsivangula@gmail.com at 3/20/2023, 11:20:10 AM on SIVANGULA RAMA KRISHNA PRASAD's Cluster

Outputs stored in DBFS:



Add data ›

DBFS

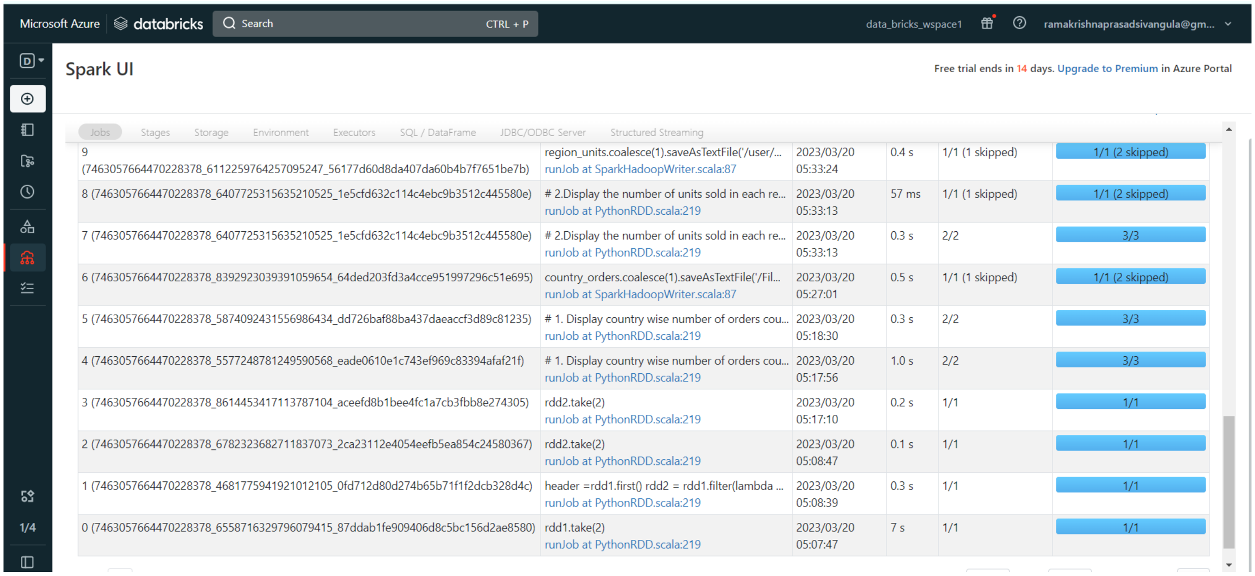
Free trial ends in 14 days. Upgrade to Premium in Azure Portal

Upload File DBFS

Select a file from DBFS

FileStore	spark	a_products
user		item_cost
		item_orders
		region_units
		yearwise_sales

Spark UI:

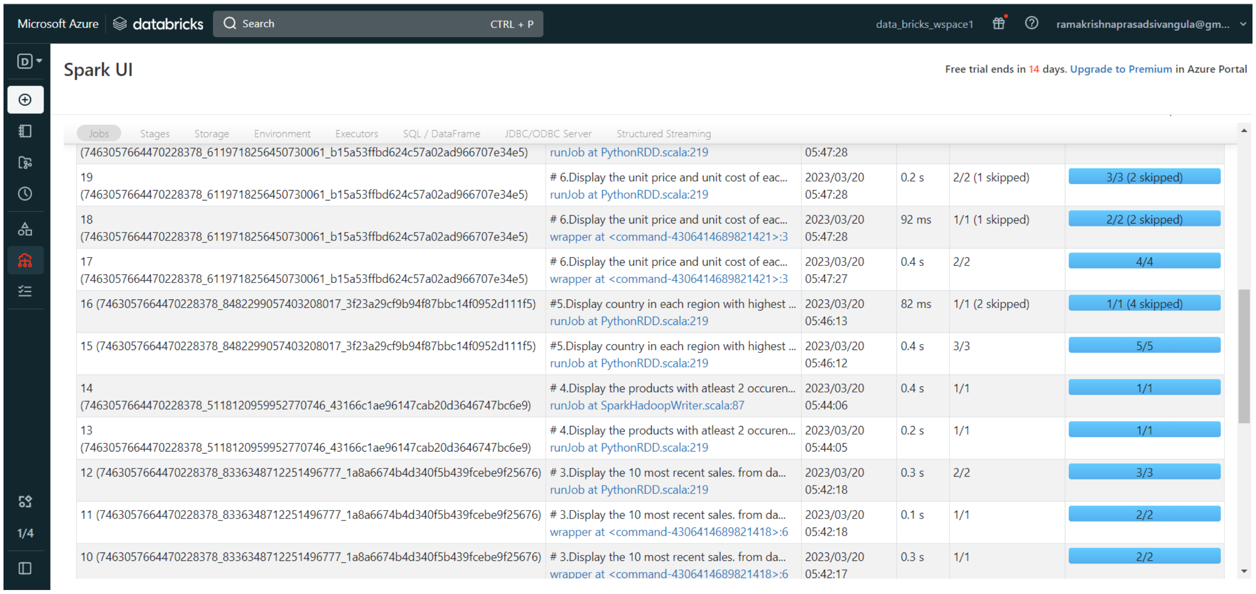


Spark UI

Free trial ends in 14 days. Upgrade to Premium in Azure Portal

Jobs Stages Storage Environment Executors SQL / DataFrame JDBC/ODBC Server Structured Streaming

9 (7463057664470228378_6112259764257095247_56177d60d8da407da60b4b7f7651be7b)	region_units.coalesce(1).saveAsTextFile('/user/... runJob at SparkHadoopWriter.scala:87	2023/03/20 05:33:24	0.4 s	1/1 (1 skipped)	1/1 (2 skipped)
8 (7463057664470228378_6407725315635210525_1e5cfd632c114c4ebc9b3512c445580e)	# 2.Display the number of units sold in each re... runJob at PythonRDD.scala:219	2023/03/20 05:33:13	57 ms	1/1 (1 skipped)	1/1 (2 skipped)
7 (7463057664470228378_6407725315635210525_1e5cfd632c114c4ebc9b3512c445580e)	# 2.Display the number of units sold in each re... runJob at PythonRDD.scala:219	2023/03/20 05:33:13	0.3 s	2/2	3/3
6 (7463057664470228378_8392923039391059654_64ded203fd3a4cce951997296c51e695)	country_orders.coalesce(1).saveAsTextFile('/Fil... runJob at SparkHadoopWriter.scala:87	2023/03/20 05:27:01	0.5 s	1/1 (1 skipped)	1/1 (2 skipped)
5 (7463057664470228378_5874092431556986434_dd726baf88ba437daeaccf3d89c81235)	# 1. Display country wise number of orders cou... runJob at PythonRDD.scala:219	2023/03/20 05:18:30	0.3 s	2/2	3/3
4 (7463057664470228378_5577248781249590568_eade0610e1c743ef969c83394afaf21f)	# 1. Display country wise number of orders cou... runJob at PythonRDD.scala:219	2023/03/20 05:17:56	1.0 s	2/2	3/3
3 (7463057664470228378_8614453417113787104_aceefd8b1bee4fc1a7cb3fbb8e274305)	rdd2.take(2) runJob at PythonRDD.scala:219	2023/03/20 05:17:10	0.2 s	1/1	1/1
2 (7463057664470228378_6782323682711837073_2ca23112e4054eefb5ea854c24580367)	rdd2.take(2) runJob at PythonRDD.scala:219	2023/03/20 05:08:47	0.1 s	1/1	1/1
1 (7463057664470228378_4681775941921012105_0fd712d80d274b65b71f1f2dcb328d4c)	header =rdd1.first() rdd2 = rdd1.filter(lambda ... runJob at PythonRDD.scala:219	2023/03/20 05:08:39	0.3 s	1/1	1/1
0 (7463057664470228378_6558716329796079415_87ddab1fe909406d8c5bc156d2ae8580)	rdd1.take(2) runJob at PythonRDD.scala:219	2023/03/20 05:07:47	7 s	1/1	1/1

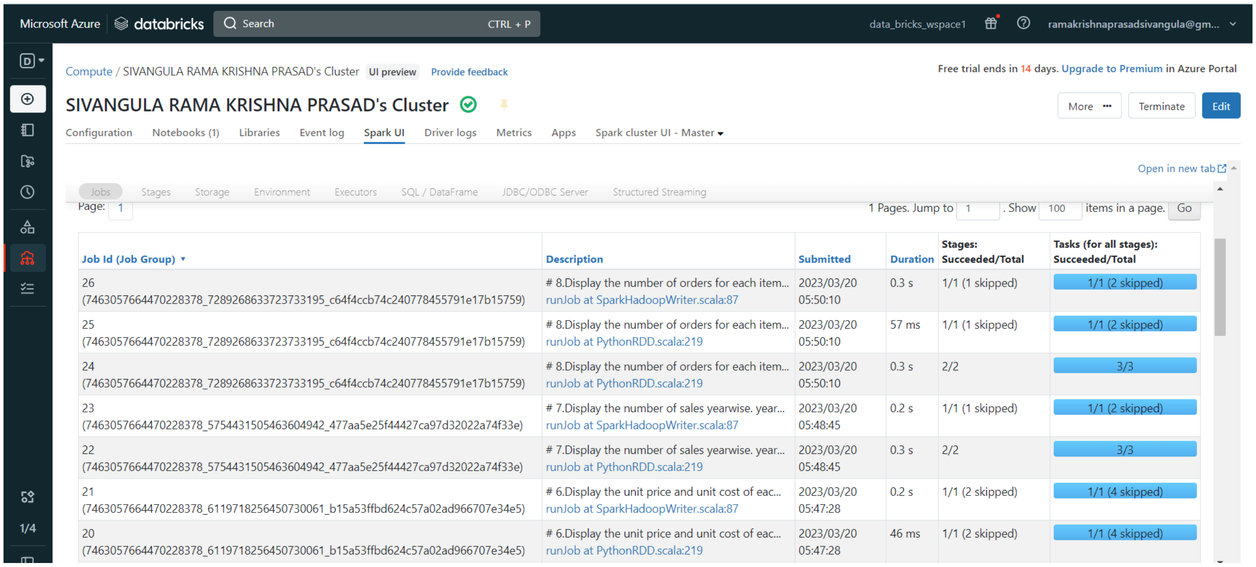


Spark UI

Free trial ends in 14 days. Upgrade to Premium in Azure Portal

Jobs Stages Storage Environment Executors SQL / DataFrame JDBC/ODBC Server Structured Streaming

(7463057664470228378_6119718256450730061_b15a53ffbd624c57a02ad966707e34e5)	runJob at PythonRDD.scala:219	05:47:28			
19 (7463057664470228378_6119718256450730061_b15a53ffbd624c57a02ad966707e34e5)	# 6.Display the unit price and unit cost of eac... runJob at PythonRDD.scala:219	2023/03/20 05:47:28	0.2 s	2/2 (1 skipped)	3/3 (2 skipped)
18 (7463057664470228378_6119718256450730061_b15a53ffbd624c57a02ad966707e34e5)	# 6.Display the unit price and unit cost of eac... wrapper at <command-4306414689821421>:3	2023/03/20 05:47:28	92 ms	1/1 (1 skipped)	2/2 (2 skipped)
17 (7463057664470228378_6119718256450730061_b15a53ffbd624c57a02ad966707e34e5)	# 6.Display the unit price and unit cost of eac... wrapper at <command-4306414689821421>:3	2023/03/20 05:47:27	0.4 s	2/2	4/4
16 (7463057664470228378_8482299057403208017_3f23a29cf9b94f87bbc14f0952d111f5)	#5.Display country in each region with highest ... runJob at PythonRDD.scala:219	2023/03/20 05:46:13	82 ms	1/1 (2 skipped)	1/1 (4 skipped)
15 (7463057664470228378_8482299057403208017_3f23a29cf9b94f87bbc14f0952d111f5)	#5.Display country in each region with highest ... runJob at PythonRDD.scala:219	2023/03/20 05:46:12	0.4 s	3/3	5/5
14 (7463057664470228378_5118120959952770746_43166c1ae96147cab20d3646747bc6e9)	# 4.Display the products with atleast 2 occuren... runJob at SparkHadoopWriter.scala:87	2023/03/20 05:44:06	0.4 s	1/1	1/1
13 (7463057664470228378_5118120959952770746_43166c1ae96147cab20d3646747bc6e9)	# 4.Display the products with atleast 2 occuren... runJob at PythonRDD.scala:219	2023/03/20 05:44:05	0.2 s	1/1	1/1
12 (7463057664470228378_8336348712251496777_1a8a6674b4d340f5b439fcebe9f25676)	# 3.Display the 10 most recent sales. from da... runJob at PythonRDD.scala:219	2023/03/20 05:42:18	0.3 s	2/2	3/3
11 (7463057664470228378_8336348712251496777_1a8a6674b4d340f5b439fcebe9f25676)	# 3.Display the 10 most recent sales. from da... wrapper at <command-4306414689821418>:6	2023/03/20 05:42:18	0.1 s	1/1	2/2
10 (7463057664470228378_8336348712251496777_1a8a6674b4d340f5b439fcebe9f25676)	# 3.Display the 10 most recent sales. from da... wrapper at <command-4306414689821418>:6	2023/03/20 05:42:17	0.3 s	1/1	2/2



Compute / SIVANGULA RAMA KRISHNA PRASAD's Cluster UI preview Provide feedback

Free trial ends in 14 days. Upgrade to Premium in Azure Portal

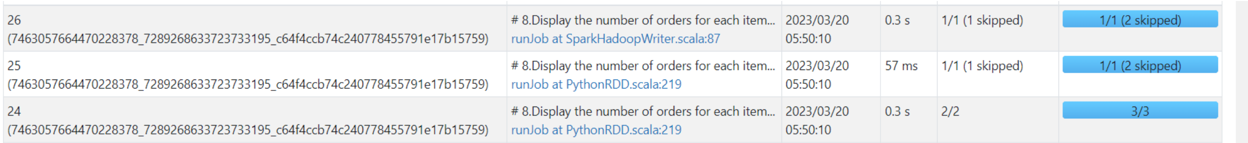
SIVANGULA RAMA KRISHNA PRASAD's Cluster

Page: 1 1 Pages. Jump to 1 . Show 100 items in a page. Go

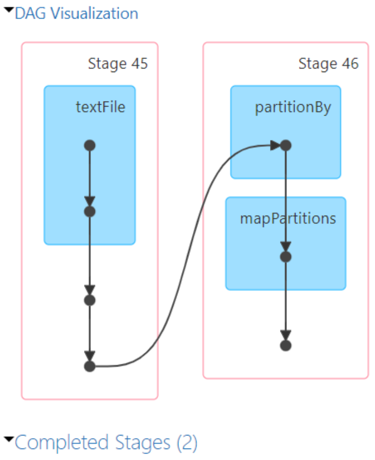
Job Id (Job Group) ▾	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
26 (7463057664470228378_7289268633723733195_c64f4ccb74c240778455791e17b15759)	# 8.Display the number of orders for each item... runJob at SparkHadoopWriter.scala:87	2023/03/20 05:50:10	0.3 s	1/1 (1 skipped)	1/1 (2 skipped)
25 (7463057664470228378_7289268633723733195_c64f4ccb74c240778455791e17b15759)	# 8.Display the number of orders for each item... runJob at PythonRDD.scala:219	2023/03/20 05:50:10	57 ms	1/1 (1 skipped)	1/1 (2 skipped)
24 (7463057664470228378_7289268633723733195_c64f4ccb74c240778455791e17b15759)	# 8.Display the number of orders for each item... runJob at PythonRDD.scala:219	2023/03/20 05:50:10	0.3 s	2/2	3/3
23 (7463057664470228378_5754431505463604942_477aa5e25f44427ca97d32022a74f33e)	# 7.Display the number of sales yearwise. year... runJob at SparkHadoopWriter.scala:87	2023/03/20 05:48:45	0.2 s	1/1 (1 skipped)	1/1 (2 skipped)
22 (7463057664470228378_5754431505463604942_477aa5e25f44427ca97d32022a74f33e)	# 7.Display the number of sales yearwise. year... runJob at PythonRDD.scala:219	2023/03/20 05:48:45	0.3 s	2/2	3/3
21 (7463057664470228378_6119718256450730061_b15a53ffbd624c57a02ad966707e34e5)	# 6.Display the unit price and unit cost of eac... runJob at SparkHadoopWriter.scala:87	2023/03/20 05:47:28	0.2 s	1/1 (2 skipped)	1/1 (4 skipped)
20 (7463057664470228378_6119718256450730061_b15a53ffbd624c57a02ad966707e34e5)	# 6.Display the unit price and unit cost of eac... runJob at PythonRDD.scala:219	2023/03/20 05:47:28	46 ms	1/1 (2 skipped)	1/1 (4 skipped)

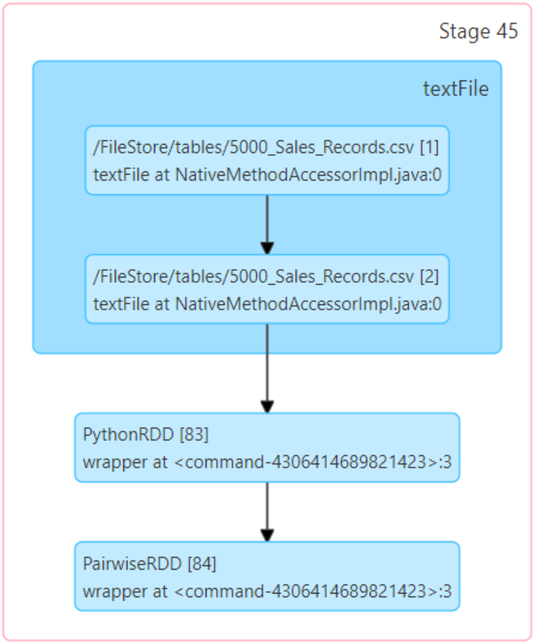
Problem statement8 job execution:

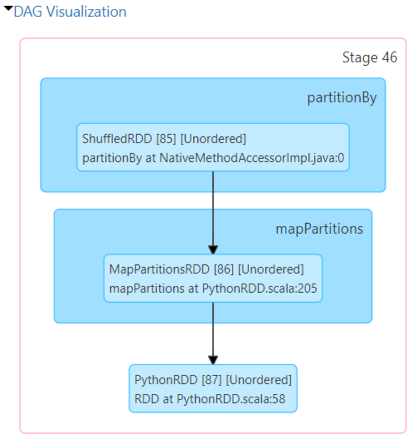
```
# 8.Display the number of orders for each item.

item_orders=rdd3.map(lambda x:(x[2],x[6])).groupBy(lambda x:x[0]).map(lambda
x:(x[0],len(x[1])))

print(item_orders.take(15))

item_orders.coalesce(1).saveAsTextFile('/user/spark/item_orders')
```

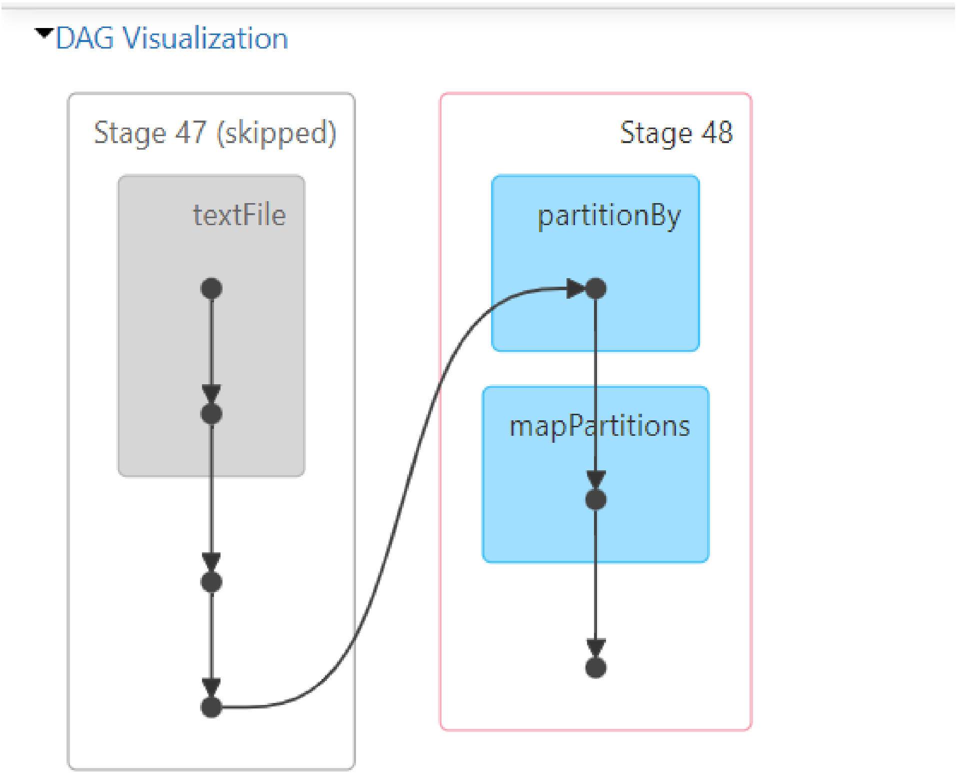


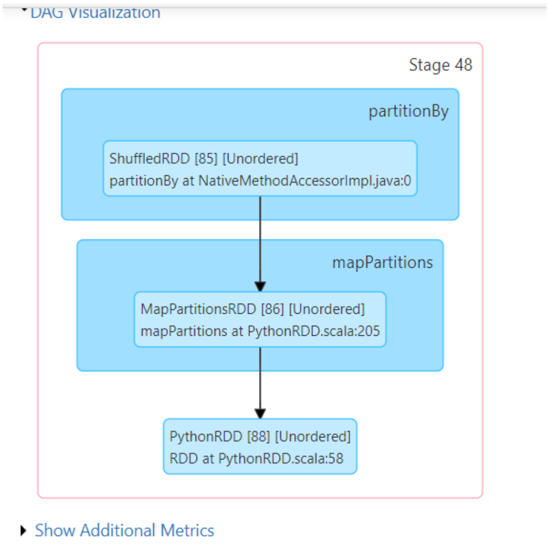
26 (7463057664470228378_7289268633723733195_c64f4ccb74c240778455791e17b15759)	# 8.Display the number of orders for each item... runJob at SparkHadoopWriter.scala:87	2023/03/20 05:50:10	0.3 s	1/1 (1 skipped)	1/1 (2 skipped)
25 (7463057664470228378_7289268633723733195_c64f4ccb74c240778455791e17b15759)	# 8.Display the number of orders for each item... runJob at PythonRDD.scala:219	2023/03/20 05:50:10	57 ms	1/1 (1 skipped)	1/1 (2 skipped)
24 (7463057664470228378_7289268633723733195_c64f4ccb74c240778455791e17b15759)	# 8.Display the number of orders for each item... runJob at PythonRDD.scala:219	2023/03/20 05:50:10	0.3 s	2/2	3/3

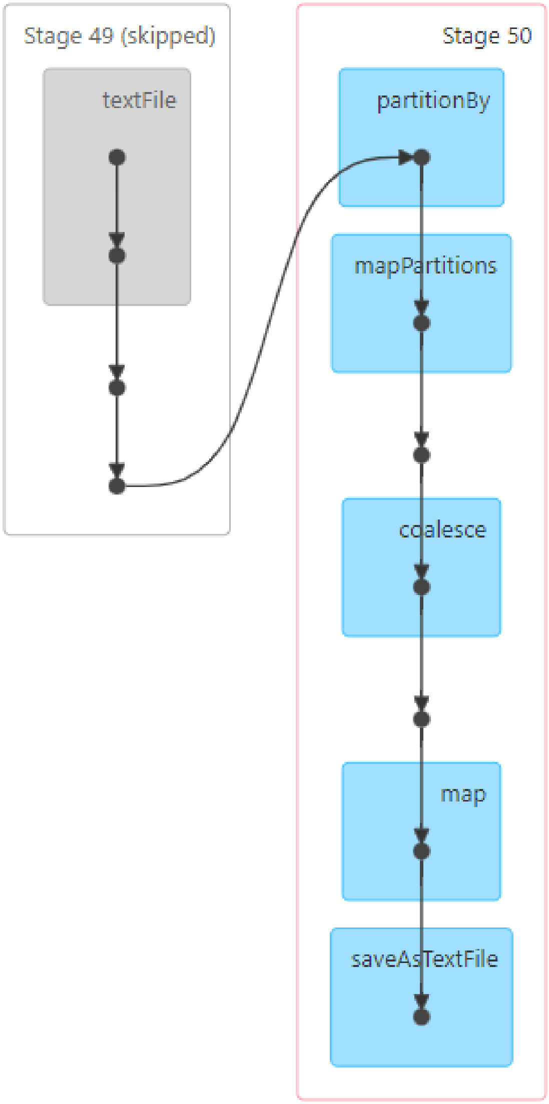

DAG Visualization
Stage 45
textFile
Stage 46
partitionBy
mapPartitions
Completed Stages (2)

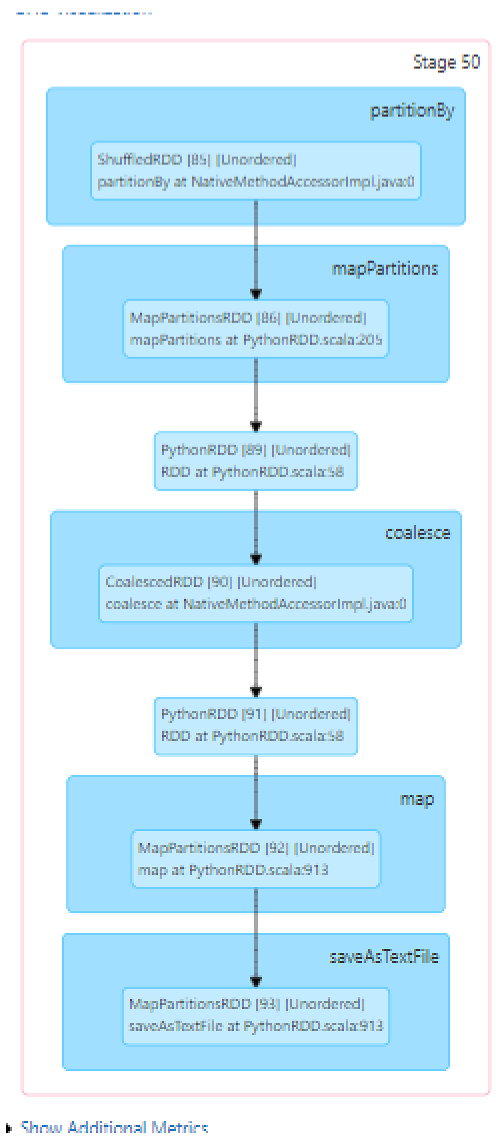

Stage 45
textFile
/FileStore/tables/5000_Sales_Records.csv [1]
textFile at NativeMethodAccessorImpl.java:0
/FileStore/tables/5000_Sales_Records.csv [2]
textFile at NativeMethodAccessorImpl.java:0
PythonRDD [83]
wrapper at <command-4306414689821423>:3
PairwiseRDD [84]
wrapper at <command-4306414689821423>:3


DAG Visualization
Stage 46
partitionBy
ShuffledRDD [85] [Unordered]
partitionBy at NativeMethodAccessorImpl.java:0
mapPartitions
MapPartitionsRDD [86] [Unordered]
mapPartitions at PythonRDD.scala:205
PythonRDD [87] [Unordered]
RDD at PythonRDD.scala:58


DAG Visualization
Stage 47 (skipped)
textFile
Stage 48
partitionBy
mapPartitions


DAG Visualization
Stage 48
partitionBy
ShuffledRDD [85] [Unordered]
partitionBy at NativeMethodAccessorImpl.java:0
mapPartitions
MapPartitionsRDD [86] [Unordered]
mapPartitions at PythonRDD.scala:205
PythonRDD [88] [Unordered]
RDD at PythonRDD.scala:58
Show Additional Metrics


Stage 49 (skipped)
textFile
Stage 50
partitionBy
mapPartitions
coalesce
map
saveAsTextFile


Stage 50
partitionBy
ShuffledRDD [85] [Unordered]
partitionBy at NativeMethodAccessorImpl.java:0
mapPartitions
MapPartitionsRDD [86] [Unordered]
mapPartitions at PythonRDD.scala:205
PythonRDD [89] [Unordered]
RDD at PythonRDD.scala:58
coalesce
CoalescedRDD [90] [Unordered]
coalesce at NativeMethodAccessorImpl.java:0
PythonRDD [91] [Unordered]
RDD at PythonRDD.scala:58
map
MapPartitionsRDD [92] [Unordered]
map at PythonRDD.scala:913
saveAsTextFile
MapPartitionsRDD [93] [Unordered]
saveAsTextFile at PythonRDD.scala:913
Show Additional Metrics