**SOLUTIONS**

**Problem1:**

**EXPLANATION:**

1. Defining schemas along with Types for 3 dataframes(customerSchema,orderSchema,storeSchema)
2. Creating DataFrames from csv files, those files are read from EdgeNode File System along with predefined schemas.
3. Creating DataFrame by joining orderDF and to get store names based in store id.
4. Then performing cleaning operations like extracting Year and Month from order date field.
5. After this step, I solved using two approach,one using SparkSQL, and next one using SparkDataframe and there I used window functions since using window functions we can perform many aggregation operations on spark dataframe.
6. For SparkSQL, simply registering the cleanDF as in memory table, then using SQL I am performing aggregation operation, then changing the field names as per our requirement and finally saving the dataframe with one partition.
7. For DataFrame approach, I am defining the window specification, then performing window aggregation operation and extracting the result as per our requirement and finally saving the dataframe with one partition.

**Code Snippet:**

*from pyspark import SparkContext,SparkConf*

*from pyspark.sql import SparkSession*

*from pyspark.sql.types import \**

*from pyspark.sql.functions import udf,col,year,month,date\_format,to\_date,sum as sparksum,count,row\_number*

*from pyspark.sql import Window*

*orderSchema = StructType([StructField('id',IntegerType(),True),StructField('total',FloatType(),True),StructField('customer\_id',StringType(),True),StructField('store\_id',StringType(),True),StructField('order\_date',DateType(),True)])*

*storeSchema = StructType([StructField('id',IntegerType(),True),StructField('name',StringType(),True),StructField('address',StringType(),True)])*

*customerSchema = StructType([StructField('id',IntegerType(),True),StructField('first\_name',StringType(),True),StructField('last\_name',StringType(),True),StructField('email',StringType(),True)])*

*orderDF = spark.read.format('csv').option('header','true').schema(orderSchema).option('dateFormat','yyyy-MM-dd').option('quote','"').option('delimiter',',').load('file:///efs/home/ps900191/pyspark\_prjct/DataBricks/Data2/orders\_test.csv',multiLine=True)*

*storeDF = spark.read.format('csv').option('header','true').schema(storeSchema).option('quote','"').option('delimiter',',').load('file:///efs/home/ps900191/pyspark\_prjct/DataBricks/Data2/store\_test.csv',multiLine=True)*

*customerDF = spark.read.format('csv').option('header','true').schema(customerSchema).option('quote','"').option('delimiter',',').load('file:///efs/home/ps900191/pyspark\_prjct/DataBricks/Data2/customer\_test.csv',multiLine=True)*

*joinedDF = orderDF.join(storeDF,[orderDF['store\_id']==storeDF['id']],how='inner').select(orderDF['id'],orderDF['total'],orderDF['order\_Date'],storeDF['name'])*

*cleanDF = joinedDF.withColumn('Year',year(joinedDF['order\_date'])).withColumn('Month',date\_format(to\_date(orderDF['order\_date'],'yyyy-MM-dd'),'MMMMMMM'))*

**Approach1 🡪 By Spark SQL**

*cleanDF.registerTempTable('cleanTable')*

*finalColNames = ['Year','Month','Store Name','Number of Orders','Total Revenue']*

*finalDF = spark.sql('select Year,Month,name,count(id),sum(total) from cleanTable group by Year,Month,name order by name').toDF(\*finalColNames)*

*finalDF.coalesce(1).write.format('csv').mode("overwrite").option("header","true").option("Sep",",").save('file:///efs/home/ps900191/pyspark\_prjct/DataBricks/Data2/output/FINALDATASET1')*

**Approach2 🡪 Spark Dataframe(window functions)**

*windowSpec = Window.partitionBy('Year','Month','name').orderBy('name')*

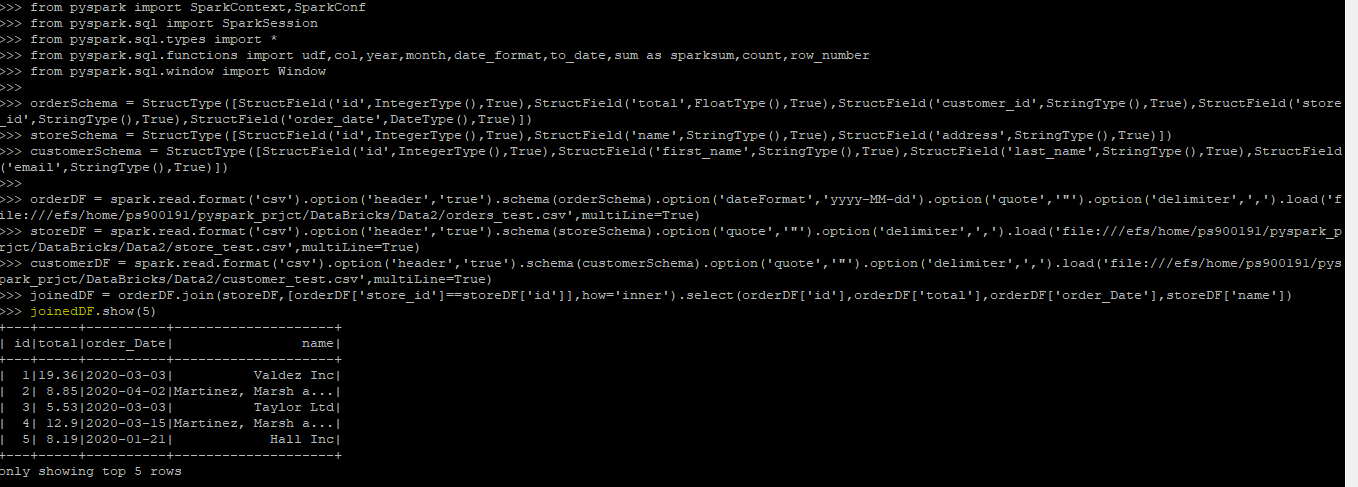
*finalDF = cleanDF.withColumn('Number of Orders',count(col('id')).over(windowSpec)).withColumn('Total Revenue',sparksum(col('total')).over(windowSpec)).withColumn('rows',row\_number().over(windowSpec)).where(col('rows')==1).withColumnRenamed('name','Store Name').select('Year','Month','Store Name','Number of Orders','Total Revenue')*

*finalDF.coalesce(1).write.format('csv').mode("overwrite").option("header","true").option("Sep",",").save('file:///efs/home/ps900191/pyspark\_prjct/DataBricks/Data2/output/FINALDATASET1')*

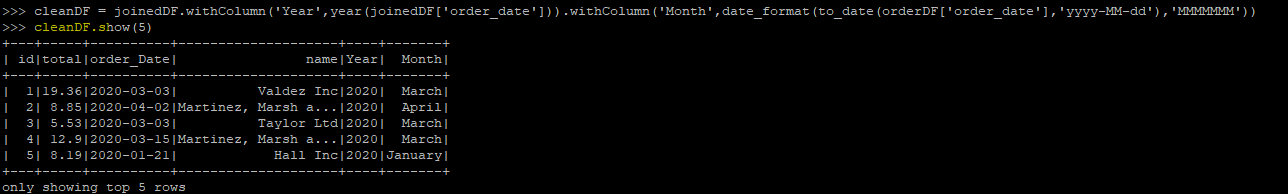
**TEST SCREENSHOTS:**

Here I have added some of the screenshots while performing testing.

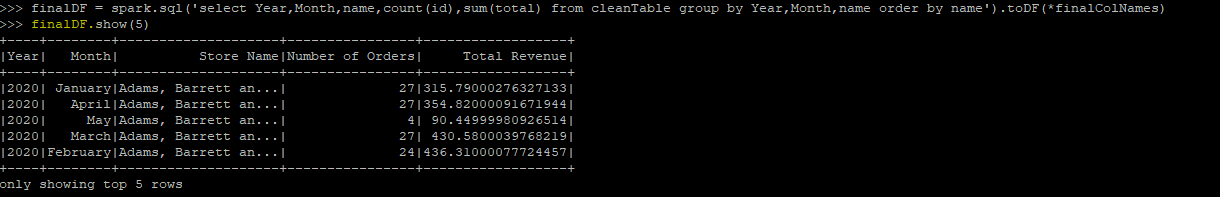
Please find them below

**JoinedDF:** 

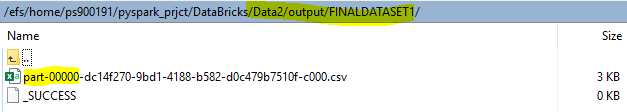
**CleanDF:**



**FinalDF:**



**FianlDF showing in edgenode path:**



**PROBLEM 2:**

**Explanation:**

1. We need Customer and Order datasets for this problem, and its created as explained in problem 1 along with schema.
2. Joining the customer and order dataframes, to get the orders done by the customer.
3. Here also I performed aggregation operation on both SparkSQL as will SparkDataFrame window operations.
4. For SparkSQL, registering the joinedDF as spark in memory temporary table, then writing sparksql on top of the temporary table after that changing the field names as per our requirement and finally saving the dataframe with one partition.
5. For DataFrame approach, I am defining the window specification, then performing window aggregation operation and extracting the result as per our requirement and finally saving the dataframe with one partition.

**Code Snippet:**

*from pyspark import SparkContext,SparkConf*

*from pyspark.sql import SparkSession*

*from pyspark.sql.types import \**

*from pyspark.sql.functions import udf,col,year,month,date\_format,to\_date,sum as sparksum,count,row\_number*

*from pyspark.sql import Window*

*orderSchema = StructType([StructField('id',IntegerType(),True),StructField('total',FloatType(),True),StructField('customer\_id',StringType(),True),StructField('store\_id',StringType(),True),StructField('order\_date',DateType(),True)])*

*storeSchema = StructType([StructField('id',IntegerType(),True),StructField('name',StringType(),True),StructField('address',StringType(),True)])*

*customerSchema = StructType([StructField('id',IntegerType(),True),StructField('first\_name',StringType(),True),StructField('last\_name',StringType(),True),StructField('email',StringType(),True)])*

*orderDF = spark.read.format('csv').option('header','true').schema(orderSchema).option('dateFormat','yyyy-MM-dd').option('quote','"').option('delimiter',',').load('file:///efs/home/ps900191/pyspark\_prjct/DataBricks/Data2/orders\_test.csv',multiLine=True)*

*customerDF = spark.read.format('csv').option('header','true').schema(customerSchema).option('quote','"').option('delimiter',',').load('file:///efs/home/ps900191/pyspark\_prjct/DataBricks/Data2/customer\_test.csv',multiLine=True)*

*joinedDF1 = customerDF.join(orderDF,[customerDF['id']==orderDF['customer\_id']],how='inner').select(customerDF['id'],customerDF['first\_name'],customerDF['last\_name'],customerDF['email'],orderDF['id'].alias('ord\_id'))*

***Approach1🡪SparkSQL***

*joinedDF1.registerTempTable('joinTable')*

*finalColNames = ['First Name','Last Name','Email','Orders Placed by user']*

*finalDF1 = spark.sql('select first\_name,last\_name,email,count(ord\_id) from joinTable group by first\_name,last\_name,email having count(ord\_id)<10').toDF(\*finalColNames)*

*finalDF1.coalesce(1).write.format('csv').mode("overwrite").option("header","true").option("Sep",",").save('file:///efs/home/ps900191/pyspark\_prjct/DataBricks/Data2/output/FINALDATASET2')*

***Approach2 🡪DataFrame window functions***

*windowSpec = Window.partitionBy('id').orderBy('id')*

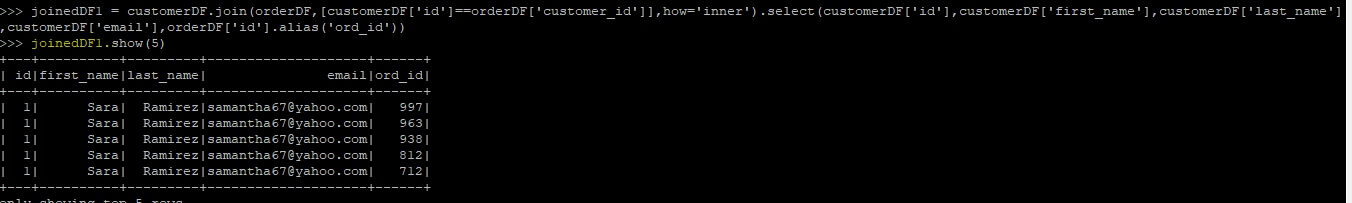
*finalColNames = ['First Name','Last Name','Email','Orders Placed by user']*

*finalDF1 = joinedDF1.withColumn('Orders Placed by User',count('ord\_id').over(windowSpec)).where(col('Orders Placed by User')<10).withColumn('rows',row\_number().over(windowSpec)).where(col('rows')==1).select('first\_name','last\_name','email','Orders Placed by user').toDF(\*finalColNames)*

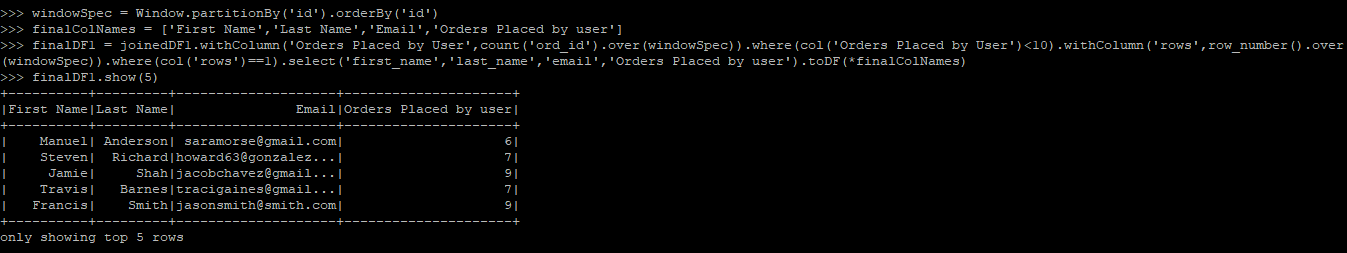
*finalDF1.coalesce(1).write.format('csv').mode("overwrite").option("header","true").option("Sep",",").save('file:///efs/home/ps900191/pyspark\_prjct/DataBricks/Data2/output/FINALDATASET2')*

**Test Screenshots:**

**JoinedDF1:**



**FinalDF1:**



**FinalDF showing in edgenodepath:**

