Spark EMR Cluster Notebook Commands (Python Dataframes)

Problem Statement: What is the daily product revenue for CLOSED or COMPLETE orders?

Task 1: Read data into Dataframe from DBFS

```
orders = spark.read. \
  csv("/FileStore/tables/retail_db/orders",
     schema="order_id INT, order_date TIMESTAMP,
order_customer_id INT, order_status STRING")

orders.printSchema()
orders.show()
orders.count()
```

Task 2: Read the order item table in to the dataframe

```
orderItems = spark.read. \
  csv("/FileStore/tables/retail_db/order_items",
     schema="'order_item_id INT,
      order_item_order_id INT,
      order_item_product_id INT,
      order_item_quantity INT,
      order_item_subtotal FLOAT,
      order_item_product_price FLOAT"')

orderItems.printSchema()
orderItems.show()
orderItems.count()
```

Task 3: Read the products file in to a Dataframe

```
products = spark.read. \
  csv("/FileStore/tables/retail_db/products",
     schema='''product_id INT,
      product_category_id INT,
      product_name STRING,
      product_description STRING,
      product_price FLOAT,
      product_image STRING''')

products.printSchema()
products.show()
products.count()
```

Task 4: Get Distinct Order statuses

```
orders.select("order_status").distinct.show
```

Task 5:Get the count for each order status

```
orderStatusCount = orders.groupBy("order_status").count
//Databricks cloud specific for visualization
display(orderStatusCount)
```

Click the chart icon to do visualization

Task 6: Filter only COMPLETE or CLOSED orders

```
val ordersCompleted = orders.filter("order_status in
('COMPLETE','CLOSED')")
ordersCompleted.show
ordersCompleted.count
```

Task 7: Join the products , order_items and orders tables

```
from pyspark.sql.functions import col

joinResults = ordersCompleted.join(orderItems, col("order_id") ==
orderItems["order_item_order_id"]). \
  join(products, col("product_id") == col("order_item_product_id")). \
  select("order_date", "product_name", "order_item_subtotal")

joinResults.show(truncate=False)
joinResults.count()
```

Task 8: Calculate the daily product revenue

```
from pyspark.sql.functions import sum, round

dailyProductRevenue = joinResults. \
  groupBy("order_date", "product_name"). \
  agg(round(sum("order_item_subtotal"), 2).alias("revenue"))

dailyProductRevenue.show(truncate=False)
```

Task 9: Sort the data

```
dailyProductRevenueSorted = dailyProductRevenue. \
  orderBy("order_date", col("revenue").desc())

dailyProductRevenueSorted.show(truncate=False)
```

Task 10: Write the results in to the file system

```
//To reduce the no of partitions after sfuffle from 200 to 2
spark.conf.set("spark.sql.shuffle.partitions",2)
```

```
dailyProductRevenueSorted.write.mode("overwrite").csv("/FileStore/tables/retail_db/daily_product_revenue")
```

Task 11: Review the file created from DBFS commands

```
%fs ls /FileStore/tables/retail_db/daily_product_revenue
```