**Pyspark Assignment – TASK 3**

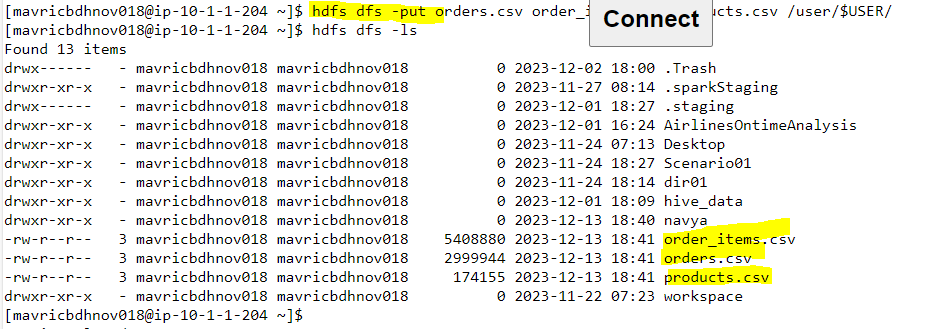
**Order Data analysis (Joins)**

1. Problem Statement: What is the daily product revenue for CLOSED or

COMPLETE orders?

**Step 1:**

**Moving the files into HDFS:**



1. Load the required data in to DF like categories, customer,departments,order\_items,orders and products

*>>> orderitemsDF=spark.read.option("header",True).option("inferschema",True). csv("order\_items.csv")*

*>>> orderitemsDF.createOrReplaceTempView("orderitems")*

*>>> orderitemsDF.printSchema()*

A yellow and black text

Description automatically generated

*>>> ordersDF=spark.read.option("header",True).option("inferschema",True).csv("Order.csv")*

*>>> ordersDF.createOrReplaceTempView("orders")*

*>>> ordersDF.printSchema()*

*>>> productsDF=spark.read.option("header",True).option("inferschema",True).csv("products.csv")*

*>>> productsDF.createOrReplaceTempView("products")*

*>>> productsDF.printSchema()*

A yellow tape with black text

Description automatically generated

1. Get the count for each order status

*>>> spark.sql("SELECT order\_status, count(order\_id) from orders group by order\_status").show()*

A screenshot of a computer

Description automatically generated

1. Filter only COMPLETE or CLOSED orders

*>>> spark.sql("SELECT \* from orders where order\_status in ('CLOSED','COMPLETE')").show()*

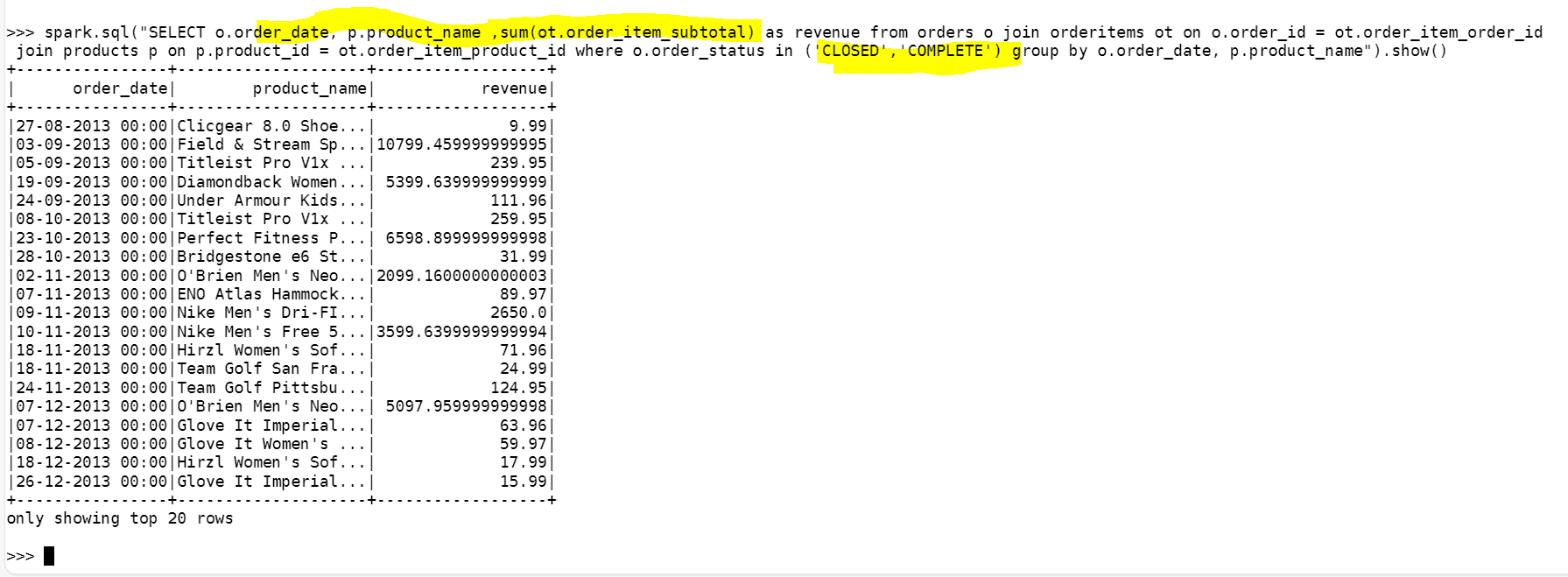
A screenshot of a computer screen

Description automatically generated

1. Join the products , order\_items and orders tables and calculate daily product revenue

*>>> spark.sql("SELECT o.order\_date, p.product\_name ,sum(ot.order\_item\_subtotal) as revenue from orders o join orderitems ot on o.order\_id = ot.order\_item\_order\_id*

*join products p on p.product\_id = ot.order\_item\_product\_id where o.order\_status in ('CLOSED','COMPLETE') group by o.order\_date, p.product\_name").show()*



Getting the count for the daily revenue:

*>>> spark.sql("SELECT count(\*) from (SELECT o.order\_date, p.product\_name ,sum(ot.order\_item\_subtotal) as revenue from orders o join orderitems ot on o.order\_id =*

*ot.order\_item\_order\_id join products p on p.product\_id = ot.order\_item\_product\_id where o.order\_status in ('CLOSED','COMPLETE') group by o.order\_date, p.product\_n*

*ame)a").show()*

A screenshot of a computer

Description automatically generated

1. Write the data in to the table Daily product revenue in Hive

*>>> from pyspark.sql import SparkSession*

*>>> spark=SparkSession.builder.appName("Spark Hive").enableHiveSupport().config("spark.sql.warehouse.dir","/user/hive/warehouse").getOrCreate()*

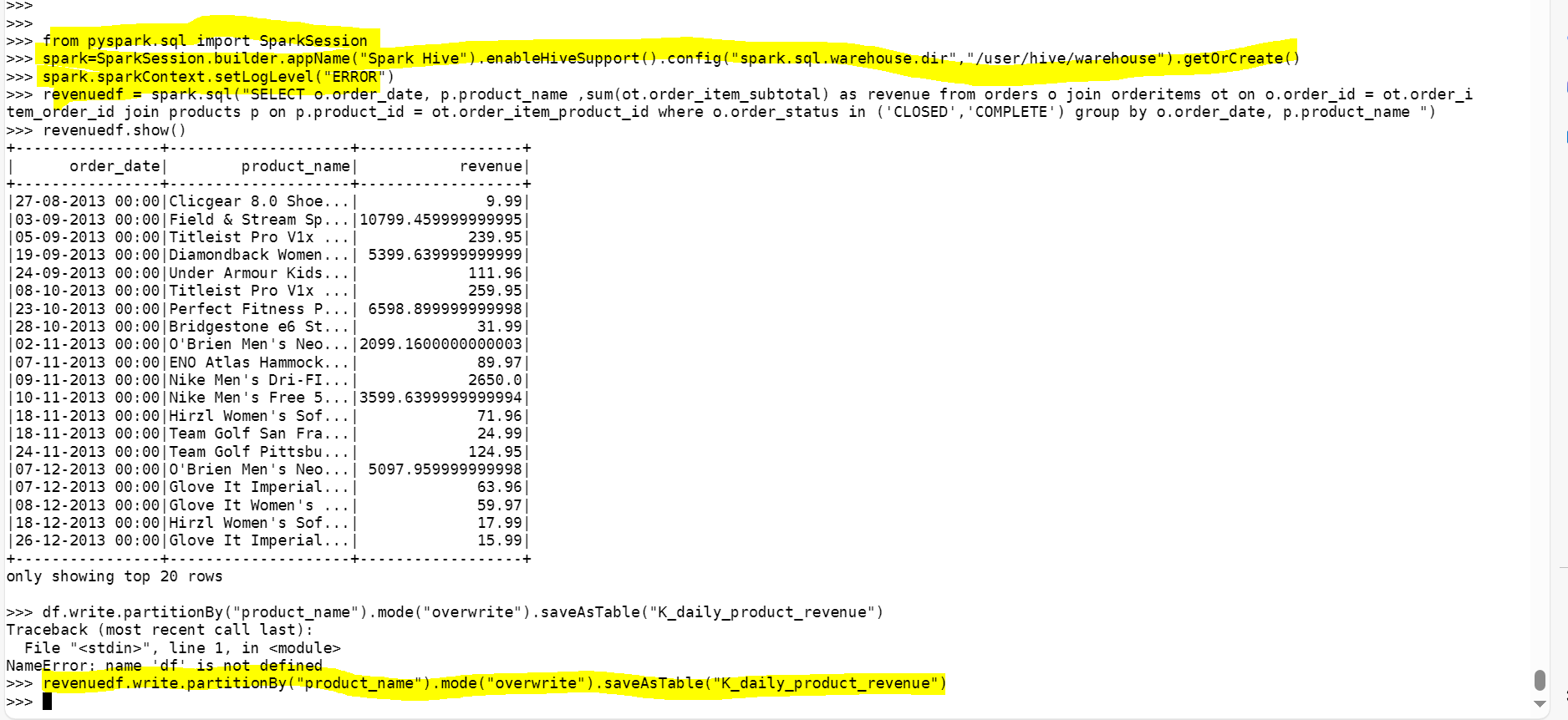
*>>> spark.sparkContext.setLogLevel("ERROR")*

*>>> revenuedf = spark.sql("SELECT o.order\_date, p.product\_name ,sum(ot.order\_item\_subtotal) as revenue from orders o join orderitems ot on o.order\_id = ot.order\_i*

*tem\_order\_id join products p on p.product\_id = ot.order\_item\_product\_id where o.order\_status in ('CLOSED','COMPLETE') group by o.order\_date, p.product\_name ")*

*>>> revenuedf.show()*

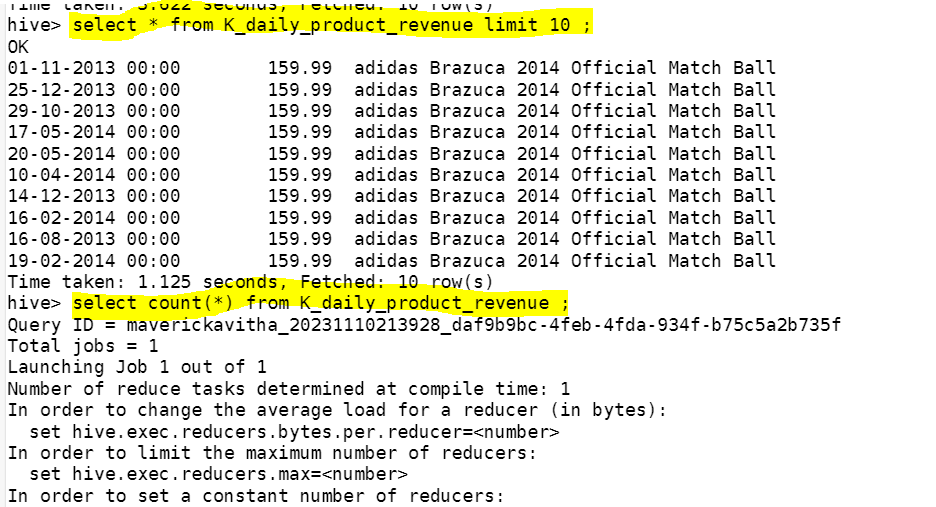
*>>> revenuedf.write.partitionBy("product\_name").mode ("overwrite").saveAsTable("K\_daily\_product\_revenue")*



**Checking data in hive :**

hive> select \* from K\_daily\_product\_revenue limit 10 ;

hive> select count(\*) from K\_daily\_product\_revenue ;



A screenshot of a computer

Description automatically generated

**Count matched for no of records in hive and pyspark SQL :**

A screenshot of a computer

Description automatically generated

Spark count:

*>>> spark.sql("SELECT count(\*) from (SELECT o.order\_date, p.product\_name ,sum(ot.order\_item\_subtotal) as revenue from orders o join orderitems ot on o.order\_id =*

*ot.order\_item\_order\_id join products p on p.product\_id = ot.order\_item\_product\_id where o.order\_status in ('CLOSED','COMPLETE') group by o.order\_date, p.product\_n*

*ame)a").show()*

A screenshot of a computer

Description automatically generated