ASSIGNMENT ON PYSPARK (2)

```
from pyspark.sql import SparkSession
from pyspark.sql import functions as F
from pyspark.sql.window import Window

# Initialize a Spark session
spark = SparkSession.builder \
    .appName("Advanced DataFrame Operations - Different Dataset") \
    .getOrCreate()

# Create two sample DataFrames for Product Sales
data1 = [
    (1, 'Product A', 'Electronics', 1200, '2022-05-10'),
    (2, 'Product B', 'Clothing', 500, '2022-07-15'),
    (3, 'Product C', 'Electronics', 1800, '2021-11-05')
]

data2 = [
    (4, 'Product D', 'Furniture', 3000, '2022-03-25'),
    (5, 'Product E', 'Clothing', 800, '2022-09-12'),
    (6, 'Product F', 'Electronics', 1500, '2021-10-19')
]

# Define schema (columns)
columns = ['ProductID', 'ProductName', 'Category', 'Price', 'SaleDate']

# Create DataFrames
sales_df1 = spark.createDataFrame(data1, columns)
sales_df2 = spark.createDataFrame(data2, columns)

# Task 1: Union of DataFrames (Removing Duplicates)
union_df = sales_df1.union(sales_df2).dropDuplicates()
```

```
union_df.show()

# Task 2: Union of DataFrames (Including Duplicates)
union_all_df = sales_df1.union(sales_df2)
union_all_df.show()

# Task 3: Rank Products by Price Within Their Category
window_spec_rank = Window.partitionBy("Category").orderBy(F.col("Price").desc())
ranked_df = union_all_df.withColumn("Rank", F.rank().over(window_spec_rank))
ranked_df.show()

# Task 4: Calculate Cumulative Price Per Category
window_spec_cum = Window.partitionBy("Category").orderBy(F.col("Price").desc()).rowsBetween(Window.unboundedPreceding, Window.currentRow)
cumulative_df = union_all_df.withColumn("CumulativePrice", F.sum("Price").over(window_spec_cum))
cumulative_df.show()

# Task 5: Convert `SaleDate` from String to Date Type
date_converted_df = union_all_df.withColumn("SaleDate", F.to_date("SaleDate", "yyyy-MM-dd"))
date_converted_df.show()

# Task 6: Calculate the Number of Days Since Each Sale
days_since_sale_df = date_converted_df.withColumn("DaysSinceSale", F.datediff(F.current_date(), "SaleDate"))
days_since_sale_df.show()

# Task 7: Add a Column for the Next Sale Deadline
next_sale_deadline_df = date_converted_df.withColumn("NextSaleDeadline", F.date_add("SaleDate", 30))
next_sale_deadline_df.show()

# Task 8: Calculate Total Revenue and Average Price Per Category
revenue_avg_df = union_all_df.groupBy("Category").agg(
    F.sum("Price").alias("TotalRevenue"),
```

```
    F.avg("Price").alias("AveragePrice")
)
revenue_avg_df.show()
```

 # Task 9: Convert All Product Names to Lowercase

```
lowercase_names_df = union_all_df.withColumn("ProductNameLower", F.lower("ProductName"))
lowercase_names_df.show()
```