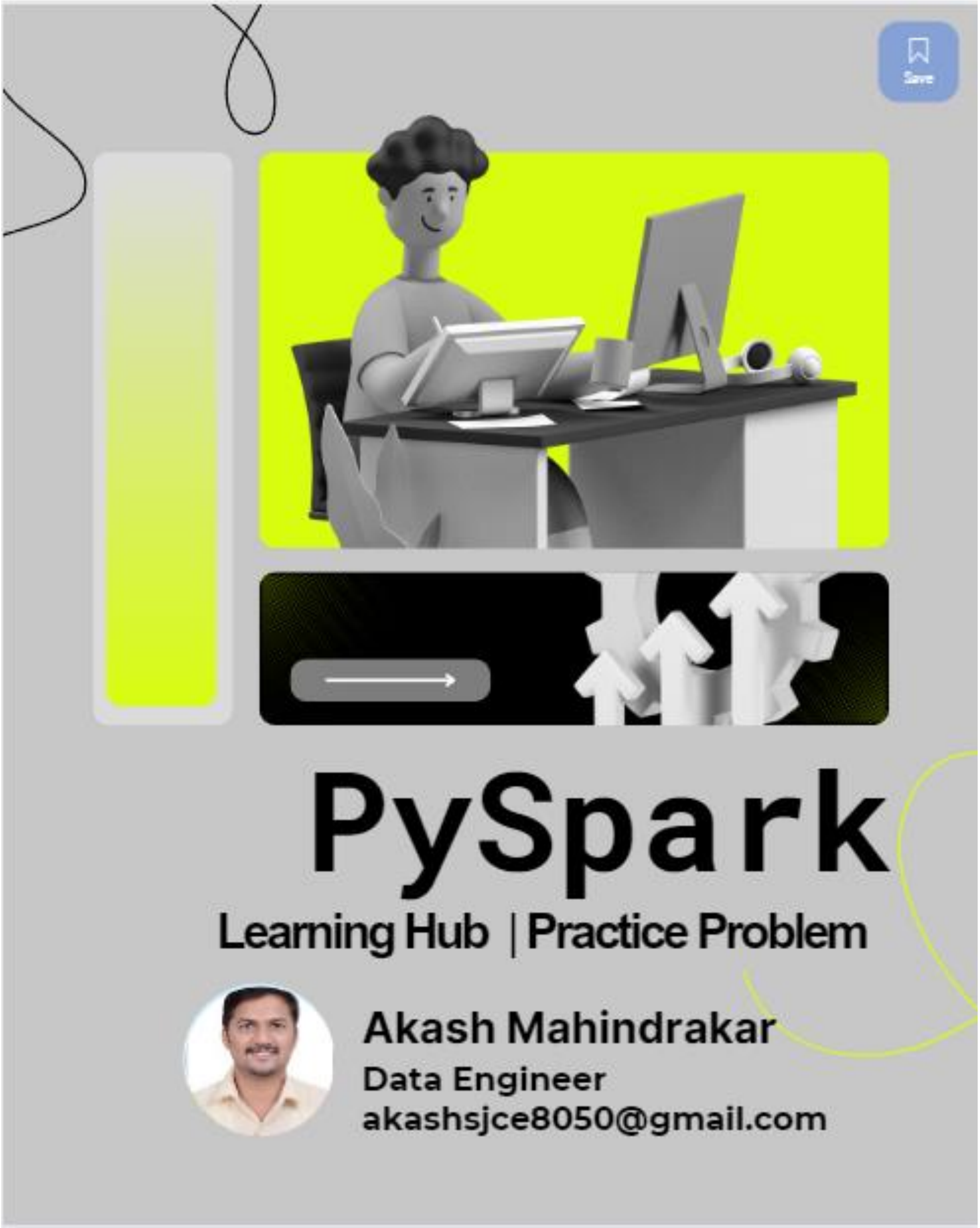
PYSPARK LEARNING HUB : DAY - 7



PySpark

Learning Hub | Practice Problem

Akash Mahindrakar
Data Engineer
akashsjce8050@gmail.com

WWW.LINKEDIN.COM/IN/AKASHMAHINDRAKAR

Step - 1 : Problem Statement

07_Rising Temperature

Write a solution to find all dates' Id with higher temperatures compared to its previous dates (yesterday).

Return the result table in any order.

Difficult Level : EASY

DataFrame:

```
# Define the schema for the "Weather" table
weather_schema = StructType([
        StructField("id", IntegerType(), True),
        StructField("recordDate", StringType(), True),
        StructField("temperature", IntegerType(), True)
])

# Define data for the "Weather" table
weather_data = [
        (1, '2015-01-01', 10),
        (2, '2015-01-02', 25),
        (3, '2015-01-03', 20),
        (4, '2015-01-04', 30)
]
```

Step - 2 : Identifying The Input Data And Expected Output

INPUT

INPUT		
ID	RECORDDATE	TEMPERATURE
1	2015-01-01	10
2	2015-01-02	25
3	2015-01-03	20
4	2015-01-04	30

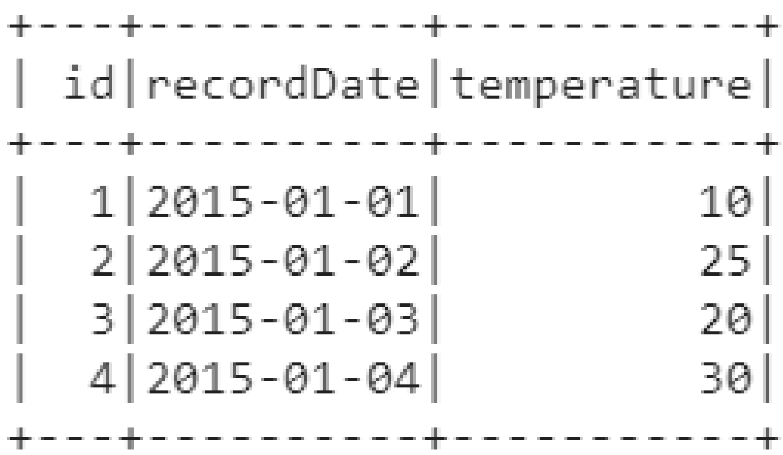
OUTPUT

OUTPUT
ID
2
4

Step - 3 : Writing the pyspark code to solve

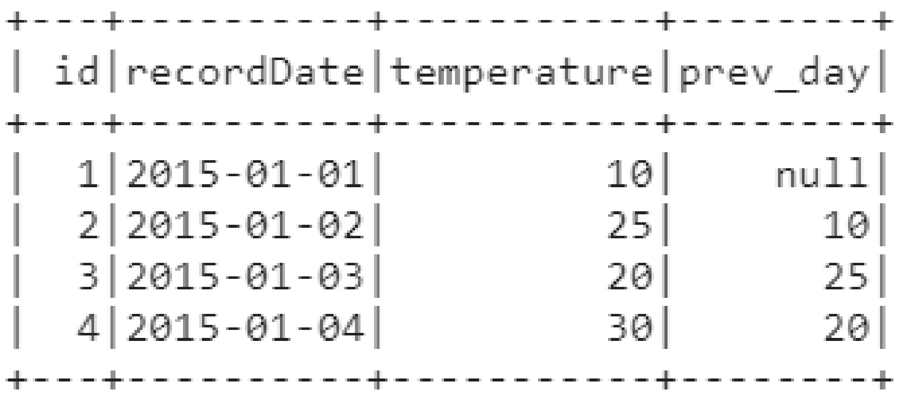
```
# Creating Spark Session
from pyspark.sql import SparkSession,Window
from pyspark.sql.types import
StructType,StructField,IntegerType,StringType
from pyspark.sql.functions import lag, col

#creating spark session
spark = SparkSession. \
builder. \
config('spark.shuffle.useOldFetchProtocol', 'true'). \
config('spark.ui.port','0'). \
config("spark.sql.warehouse.dir", "/user/itv008042/warehouse"). \
enableHiveSupport(). \
master('yarn'). \
getOrCreate()


# Define the schema for the "Weather" table
weather_schema = StructType([
        StructField("id", IntegerType(), True),
        StructField("recordDate", StringType(), True),
        StructField("temperature", IntegerType(), True)
])

# Define data for the "Weather" table
weather_data = [
        (1, '2015-01-01', 10),
        (2, '2015-01-02', 25),
        (3, '2015-01-03', 20),
        (4, '2015-01-04', 30)
]
```

```
# Create a PySpark DataFrame
temp_df=spark.createDataFrame(weather_data,weather_schema)
temp_df.show()
```

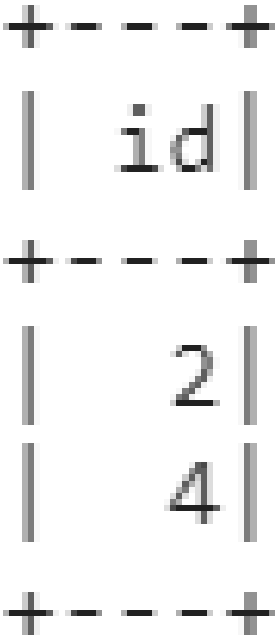


```
+---+----------+-----------+
| id|recordDate|temperature|
+---+----------+-----------+
|  1|2015-01-01|         10|
|  2|2015-01-02|         25|
|  3|2015-01-03|         20|
|  4|2015-01-04|         30|
+---+----------+-----------+
```

```
lag_df=temp_df.withColumn("prev_day",lag(temp_df.temperature).
over(Window.orderBy(temp_df.recordDate)))
lag_df.show()
```



```
+---+----------+-----------+--------+
| id|recordDate|temperature|prev_day|
+---+----------+-----------+--------+
|  1|2015-01-01|         10|    null|
|  2|2015-01-02|         25|      10|
|  3|2015-01-03|         20|      25|
|  4|2015-01-04|         30|      20|
+---+----------+-----------+--------+
lag_df.filter(lag_df["temperature"] > lag_df["prev_day"]
).select("id").show()
```



```
+----+
|  id|
+----+
|   2|
|   4|
+----+
```

Was it helpful?

follow for more!

Akash Mahindrakar
Data Engineer
akashsjce8050@gmail.com

SHARE YOUR THOUGHTS IN COMMENT BELOW