



**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)**

**]**



**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** |



**# Creating Spark Session**

**from pyspark.sql import SparkSession**

**from pyspark.sql.types import StructType,StructField,IntegerType,StringType**

**#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),**

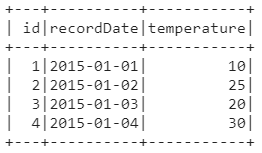
**(4, '2015-01-04', 30)**

**]**

**# Create a PySpark DataFrame**

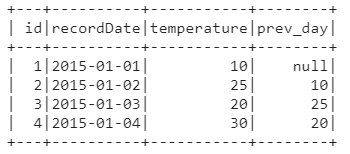
**temp\_df=spark.createDataFrame(weather\_data,weather\_schema)**

**temp\_df.show()**

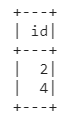
****

**lag\_df=temp\_df.withColumn("prev\_day",lag(temp\_df.temperature).over(Window.orderBy(temp\_df.recordDate)))**

**lag\_df.show()**

****

**lag\_df.filter(lag\_df["temperature"] > lag\_df["prev\_day"] ).select("id").show()**

****

