Vijay's Assignment – Spark SQL 3

For this data analysis, you can download the necessary dataset fromthislink.

In the above link there are two datasets; building.csv contains the details of the top 20 buildings all over the world and HVAC.csv contains the target temperature and the actual temperature along with the buildingId.

HVAC (heating, ventilating/ventilation, and air conditioning) is the technology of indoor and vehicular environmental comfort. Its goal is to provide thermal comfort and acceptable indoor air quality. Through the HVAC sensors, we will get the temperature of thebuildings.

Here are the columns that are present in thedatasets:

Building.csv –BuildingID, BuildingMgr, BuildingAge,HVACproduct,Country

HVAC.csv –Date, Time, TargetTemp, ActualTemp, System, SystemAge,BuildingID

Generic section of the program

```
package vksp1

import org.apache.spark.SparkConf
import org.apache.spark.SparkContext
import org.apache.spark.rdd.RDD.rddToPairRDDFunctions
import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.functions.udf
import org.apache.spark.sql.functions._

import org.apache.spark.sql.types._
object vksprksql3 {

    def main(args: Array[String]) {

    val conf = new SparkConf()
      .setAppName("vjsprksql3")
      .setMaster("local")

    val sc = new SparkContext(conf)

    val spark = SparkSession
      .builder()
      .appName("Spark SQL basic example")
      .config("spark.some.config.option", "some-value")
      .getOrCreate()

    import spark.implicits._
```

Objective 1:

Load HVAC.csv file into temporarytable

●Add a new column, tempchange -set to 1, if there is a change of greater than +/-5 between actual and target temperature

```
val hvacDF = spark.read.format("csv")
    .option("header", "true")
    .option("inferschema", "true")
    .load("C:\\Users\\VIJAYLAKSHMANAN\\spark\\hvac1.csv")


    val udf1 = udf((a:Int, b:Int) => {if ((a-b)>=5||(a-b)<= -5) "1" else " "})

    val hvacDF1 =
hvacDF.withColumn("tempchange",udf1(hvacDF.col("TargetTemp"),hvacDF.col("ActualTem
p")))

    hvacDF1.show()

    hvacDF1.registerTempTable("hvac1")
```

Output:

```
19/02/22 18:20:49 INFO DAGScheduler: Job 2 finished: show at vjsprksql3.scala:40, took 0.075210 s
+-------+--------+----------+----------+------+---------+----------+----------+
|   Date|    Time|TargetTemp|ActualTemp|System|SystemAge|BuildingID|tempchange|
+-------+--------+----------+----------+------+---------+----------+----------+
| 6-1-13|00:00:01|        66|        58|    13|       20|         4|         1|
| 6-2-13|01:00:01|        69|        68|     3|       20|        17|          |
| 6-3-13|02:00:01|        70|        73|    17|       20|        18|          |
| 6-4-13|03:00:01|        67|        63|     2|       23|        15|          |
| 6-5-13|04:00:01|        68|        74|    16|        9|         3|         1|
| 6-6-13|05:00:01|        67|        56|    13|       28|         4|         1|
| 6-7-13|06:00:01|        70|        58|    12|       24|         2|         1|
| 6-8-13|07:00:01|        70|        73|    20|       26|        16|          |
| 6-9-13|08:00:01|        66|        69|    16|        9|         9|          |
|6-10-13|09:00:01|        65|        57|     6|        5|        12|         1|
|6-11-13|10:00:01|        67|        70|    10|       17|        15|          |
|6-12-13|11:00:01|        69|        62|     2|       11|         7|         1|
|6-13-13|12:00:01|        69|        73|    14|        2|        15|          |
|6-14-13|13:00:01|        65|        61|     3|        2|         6|          |
|6-15-13|14:00:01|        67|        59|    19|       22|        20|         1|
|6-16-13|15:00:01|        65|        56|    19|       11|         8|         1|
|6-17-13|16:00:01|        67|        57|    15|        7|         6|         1|
|6-18-13|17:00:01|        66|        57|    12|        5|        13|         1|
|6-19-13|18:00:01|        69|        58|     8|       22|         4|         1|
|6-20-13|19:00:01|        67|        55|    17|        5|         7|         1|
+-------+--------+----------+----------+------+---------+----------+----------+
only showing top 20 rows
```

Obective 2:

Load building.csv file into temporarytable

```
val buildDF = spark.read.format("csv")
    .option("header", "true")
    .option("inferschema", "true")
    .load("C:\\Users\\VIJAYLAKSHMANAN\\spark\\building1.csv")

    buildDF.registerTempTable("building1")

    val query7DF = spark.sql("select * from building1")
    query7DF.show()
```

Output:

```
19/02/22 18:25:03 INFO DAGScheduler: Job 5 finished: show at vjsprksql3.scala:51, took 0.061302 s
+----------+-----------+-----------+-----------+------------+
|BuildingID|BuildingMgr|BuildingAge|HVACproduct|     Country|
+----------+-----------+-----------+-----------+------------+
|         1|         M1|         25|     AC1000|         USA|
|         2|         M2|         27|     FN39TG|      France|
|         3|         M3|         28|     JDNS77|      Brazil|
|         4|         M4|         17|     GG1919|     Finland|
|         5|         M5|          3|    ACMAX22|   Hong Kong|
|         6|         M6|          9|     AC1000|   Singapore|
|         7|         M7|         13|     FN39TG|South Africa|
|         8|         M8|         25|     JDNS77|   Australia|
|         9|         M9|         11|     GG1919|      Mexico|
|        10|        M10|         23|    ACMAX22|       China|
|        11|        M11|         14|     AC1000|     Belgium|
|        12|        M12|         26|     FN39TG|     Finland|
|        13|        M13|         25|     JDNS77|Saudi Arabia|
|        14|        M14|         17|     GG1919|     Germany|
|        15|        M15|         19|    ACMAX22|      Israel|
|        16|        M16|         23|     AC1000|      Turkey|
|        17|        M17|         11|     FN39TG|       Egypt|
|        18|        M18|         25|     JDNS77|   Indonesia|
|        19|        M19|         14|     GG1919|      Canada|
|        20|        M20|         19|    ACMAX22|   Argentina|
+----------+-----------+-----------+-----------+------------+

19/02/22 18:25:03 INFO SparkContext: Invoking stop() from shutdown hook
```

Objective 3

Figure out the number of times, temperature has changed by 5 degrees or more for eachcountry:

- Join both thetables.
- Select tempchange and countrycolumn
- Filter the rows where tempchange is 1 and count the numberofoccurrence for eachcountry

```
val query8DF = spark.sql("select a.country, count(b.*) as Cnt from building1 a,
hvac1 b where a.BuildingID = b.BuildingID and b.tempchange = 1 group by
a.Country")
    query8DF.show()
```

Output

country	Cnt
Singapore	230
Turkey	243
Germany	196
France	251
Argentina	230
Belgium	199
Finland	473
China	241
Hong Kong	248
Israel	232
USA	213
Mexico	228
Indonesia	243
Saudi Arabia	233
Canada	232
Brazil	226
Australia	225
Egypt	236
South Africa	237