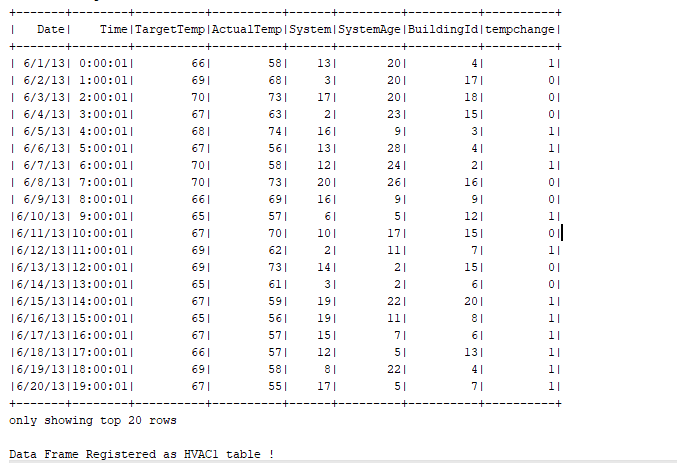
**Code Logic**

**package** SQL  
  
**import** org.apache.spark.sql.SparkSession  
  
**object** SparkSQLUseCase1 {  
  
 **case class** hvac\_cls(Date:String,Time:String,TargetTemp:Int,ActualTemp:Int,System:Int,SystemAge:Int,BuildingId:Int)  
  
 **case class** building(buildid:Int,buildmgr:String,buildAge:Int,hvacproduct:String,Country:String)  
  
 **def** main(args: Array[String]): Unit = {  
  
 *println*(**"hey scala"**)  
  
 **val** spark = SparkSession  
 .*builder*()  
 .master(**"local"**)  
 .appName(**"Spark SQL Use Case 1 "**)  
 .config(**"spark.some.config.option"**, **"some-value"**)  
 .getOrCreate()  
  
 *println*(**"Spark Session Object created"**)  
  
 *//Set the log level as warning* spark.sparkContext.setLogLevel(**"WARN"**)  
  
  
 **val** data = spark.sparkContext.textFile(**"G:\\Bigdata\\Spark\\HVAC.csv"**);  
  
 *println*(**"HVAC Data->>"**+data.count())  
  
 **val** header = data.first()  
  
  
 **val** data1 = data.filter(row => row != header)  
  
 *println*(**"Header removed from the data !"**)  
  
 *//For implicit conversions like converting RDDs and sequences to DataFrames* **import** spark.implicits.\_  
  
 **val** hvac = data1.map(x=>x.split(**","**)).map(x => *hvac\_cls*(x(0),x(1),x(2).toInt,x(3).toInt,x(4).toInt,x(5).toInt,x(6).toInt)).toDF()  
  
 hvac.show()  
  
 *println*(**"HVAC Dataframe created !"**)  
  
 hvac.registerTempTable(**"HVAC"**)  
 *println*(**"Dataframe Registered as table !"**)  
  
  
 **val** hvac1 = spark.sql(**"select \*,IF((targettemp - actualtemp) > 5, '1', IF((targettemp - actualtemp) < -5, '1', 0)) AS tempchange from HVAC"**)  
  
 hvac1.show()  
  
  
 hvac1.registerTempTable(**"HVAC1"**)  
  
 *println*(**"Data Frame Registered as HVAC1 table !"**)  
 hvac.registerTempTable(**"HVAC"**)  
  
  
 *println*(**"Dataframe Registered as table !"**)  
  
  
 **val** data2 = spark.sparkContext.textFile(**"G:\\Bigdata\\Spark\\building.csv"**);  
  
  
 **val** header1 = data2.first()  
  
 **val** data3 = data2.filter(row => row != header1)  
  
  
 *println*(**"Header removed from the building data"**)  
  
 *println*(**"Buildings Data->>"**+data3.count())  
  
 *//Now let us create the building dataframe* **val** build = data3.map(x=> x.split(**","**)).map(x => *building*(x(0).toInt,x(1),x(2).toInt,x(3),x(4))).toDF  
  
 build.show()  
  
  
 build.registerTempTable(**"building"**)  
  
 *println*(**"Buildings data registered as building table"**)  
  
 *//Now join the two tables* **val** build1 = spark.sql(**"select h.\*, b.country, b.hvacproduct from building b join hvac1 h on b.buildid = h.buildingid"**)  
  
 build1.show()  
  
 *//Select temperature and country column from above* **val** tempCountry = build1.map(x => (**new** Integer(x(7).toString),x(8).toString))  
  
 tempCountry.show()  
  
 *//Filter the values* **val** tempCountryOnes = tempCountry.filter(x=> {**if**(x.\_1==1) **true else false**})  
  
 tempCountryOnes.show()  
  
 **val** notSorted = tempCountryOnes.groupBy(**"\_2"**).count  
  
 notSorted.sort(**$"count"**.desc).show()  
 }  
}

**Load HVAC.csv file into temporary table**

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

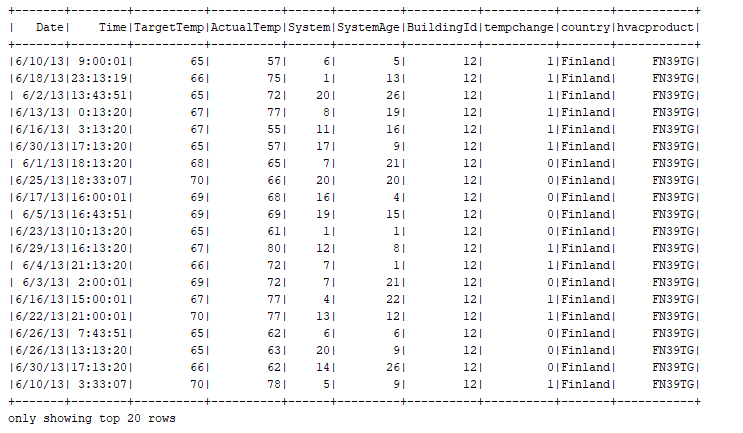
Output



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

**○ Join both the tables.**

*Output*

****

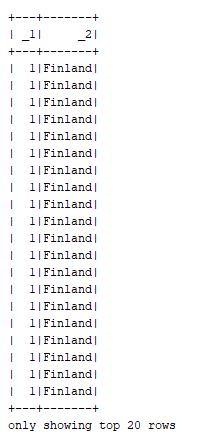
**○ Select temp change and country column**

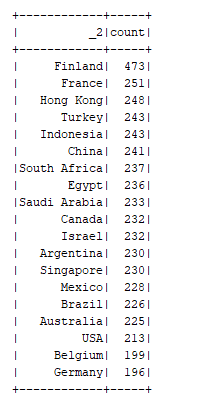
*Output*

****

**○ Filter the rows where tempchange is 1 and count the number of occurrence for each country**

*Output*

****

****