Case_Study_-_III Sensor

DESCRIPTION –

- 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 building Id.

- **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 the buildings.

- Here are the columns that are present in the datasets:
  Building.csv – BuildingID, BuildingMgr, BuildingAge, HVACproduct,Country
  HVAC.csv – Date, Time, TargetTemp, ActualTemp, System, SystemAge, BuildingID

OBJECTIVES –

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

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

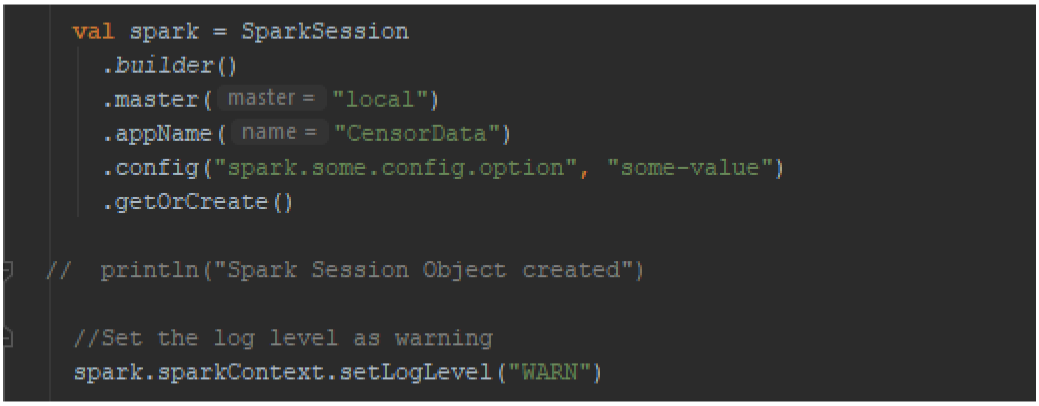
3. Load **building.csv** file into temporary table.

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

- Join both the tables.
- Select **tempchange** and country column
- Filter the rows where **tempchange** is 1 and count the number of occurrence for each country.

SOLUTION –

- Setting up Spark context.



```
  val spark = SparkSession
    .builder()
    .master( master = "local")
    .appName( name = "CensorData")
    .config("spark.some.config.option", "some-value")
    .getOrCreate()

//  println("Spark Session Object created")

  //Set the log level as warning
  spark.sparkContext.setLogLevel("WARN")
```

- Loading data from HVAC.csv, building.csv and removing header details.

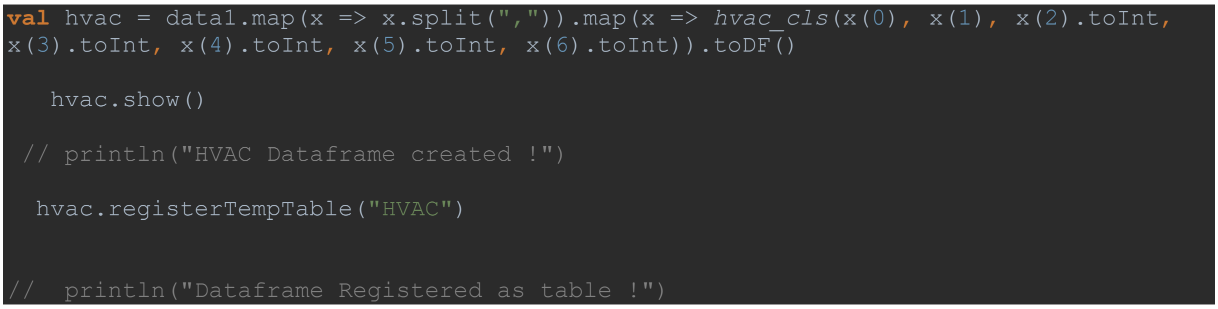
```
val data = spark.sparkContext.textFile(path =
"C:\\Users\\prave\\Desktop\\Spark\\HVAC.csv")

println("HVAC Data->>"+data.count())

val header = data.first()

val data1 = data.filter(row => row != header)

println("Header removed from the data !")
```

```
val data2 =
spark.sparkContext.textFile("C:\\Users\\prave\\Desktop\\Spark\\building.csv");

val header1 = data2.first()

val data3 = data2.filter(row => row != header1)

println("Header removed from the building data")
```

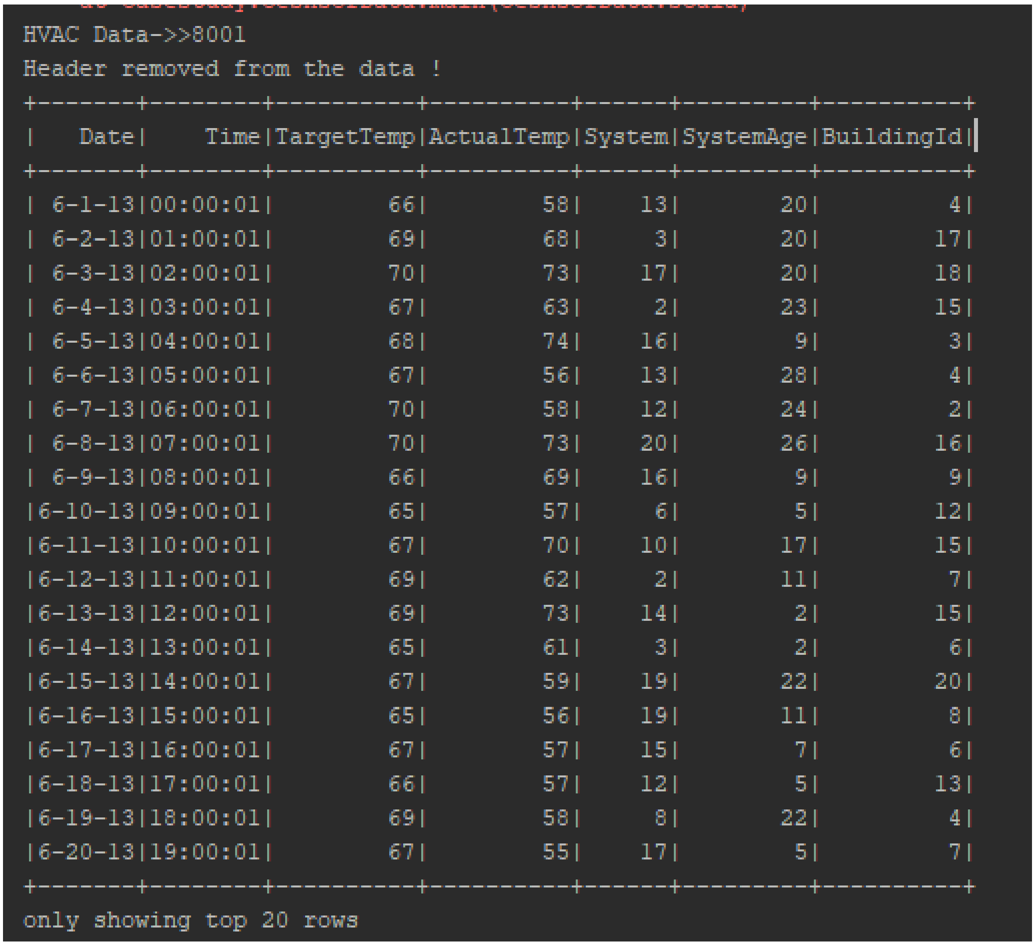
- Case class to create data frame for HVAC.csv and building.csv

```
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)
```

- Create data frame for HVAC.csv and register into a temp table “HVAC”

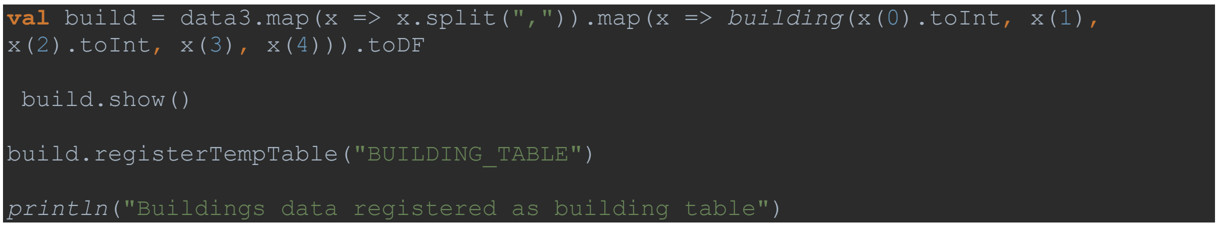


```
val hvac = data1.map(x => x.split(",")).map(x => hvac_cls(x(0), x(1), x(2).toInt,
x(3).toInt, x(4).toInt, x(5).toInt, x(6).toInt)).toDF()

   hvac.show()

 // println("HVAC Dataframe created !")

  hvac.registerTempTable("HVAC")


//  println("Dataframe Registered as table !")
```

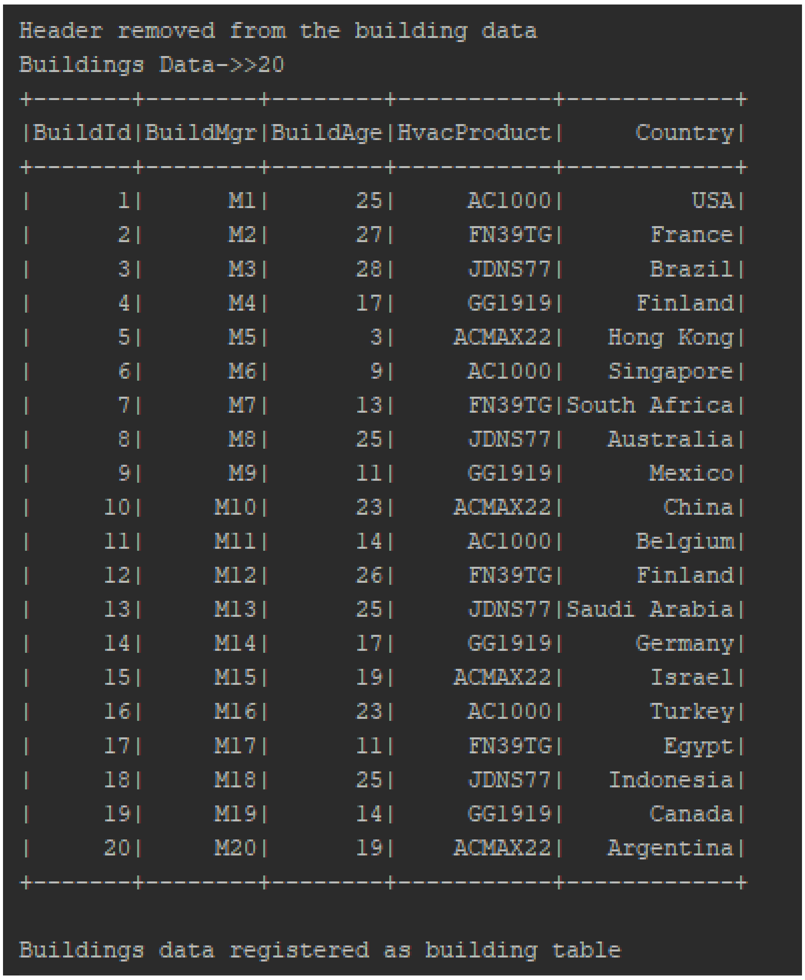


```
HVAC Data->>8001
Header removed from the data !
+-------+--------+----------+----------+------+---------+----------+
|   Date|    Time|TargetTemp|ActualTemp|System|SystemAge|BuildingId|
+-------+--------+----------+----------+------+---------+----------+
| 6-1-13|00:00:01|        66|        58|    13|       20|         4|
| 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|
| 6-6-13|05:00:01|        67|        56|    13|       28|         4|
| 6-7-13|06:00:01|        70|        58|    12|       24|         2|
| 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|
|6-11-13|10:00:01|        67|        70|    10|       17|        15|
|6-12-13|11:00:01|        69|        62|     2|       11|         7|
|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|
|6-16-13|15:00:01|        65|        56|    19|       11|         8|
|6-17-13|16:00:01|        67|        57|    15|        7|         6|
|6-18-13|17:00:01|        66|        57|    12|        5|        13|
|6-19-13|18:00:01|        69|        58|     8|       22|         4|
|6-20-13|19:00:01|        67|        55|    17|        5|         7|
+-------+--------+----------+----------+------+---------+----------+
only showing top 20 rows
```

- Create data frame for building.csv and register into a temp table “BUILDING_TABLE”

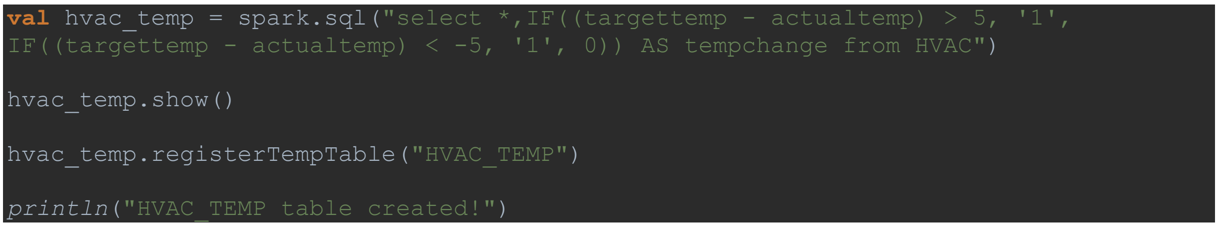


```
val build = data3.map(x => x.split(",")).map(x => building(x(0).toInt, x(1),
x(2).toInt, x(3), x(4))).toDF

 build.show()

build.registerTempTable("BUILDING_TABLE")

println("Buildings data registered as building table")
```

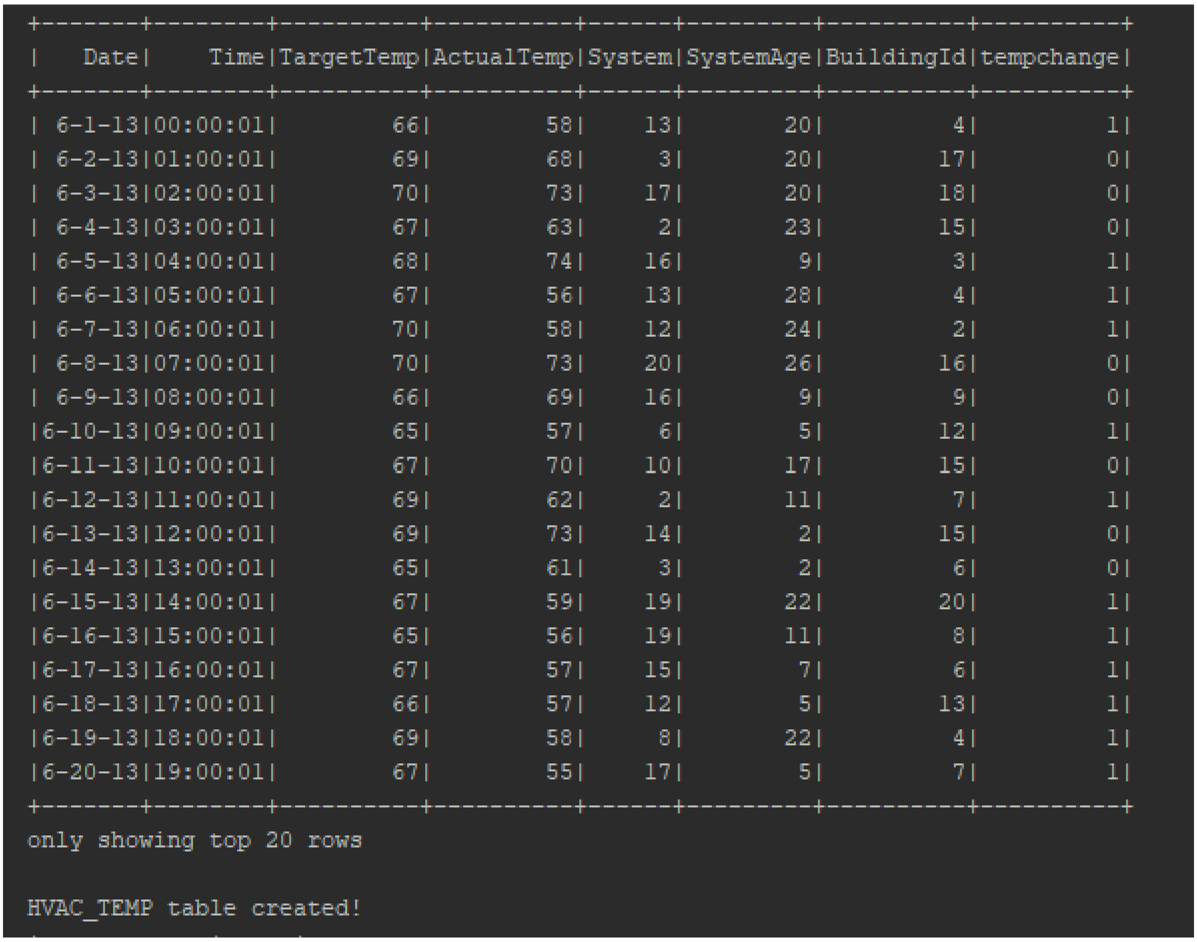


```
Header removed from the building data
Buildings Data->>20
+-------+--------+--------+-----------+------------+
|BuildId|BuildMgr|BuildAge|HvacProduct|     Country|
+-------+--------+--------+-----------+------------+
|      1|      M1|      25|     AC1000|         USA|
|      2|      M2|      27|     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|
+-------+--------+--------+-----------+------------+

Buildings data registered as building table
```

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



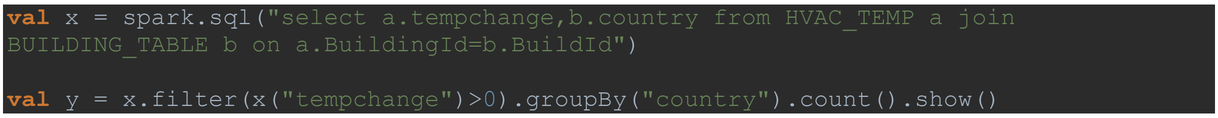
```
val hvac_temp = spark.sql("select *,IF((targettemp - actualtemp) > 5, '1',
IF((targettemp - actualtemp) < -5, '1', 0)) AS tempchange from HVAC")

hvac_temp.show()

hvac_temp.registerTempTable("HVAC_TEMP")

println("HVAC_TEMP table created!")
```



```
+-------+--------+----------+----------+------+---------+----------+----------+
|   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|         0|
| 6-3-13|02:00:01|        70|        73|    17|       20|        18|         0|
| 6-4-13|03:00:01|        67|        63|     2|       23|        15|         0|
| 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|         0|
| 6-9-13|08:00:01|        66|        69|    16|        9|         9|         0|
|6-10-13|09:00:01|        65|        57|     6|        5|        12|         1|
|6-11-13|10:00:01|        67|        70|    10|       17|        15|         0|
|6-12-13|11:00:01|        69|        62|     2|       11|         7|         1|
|6-13-13|12:00:01|        69|        73|    14|        2|        15|         0|
|6-14-13|13:00:01|        65|        61|     3|        2|         6|         0|
|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

HVAC_TEMP table created!
```

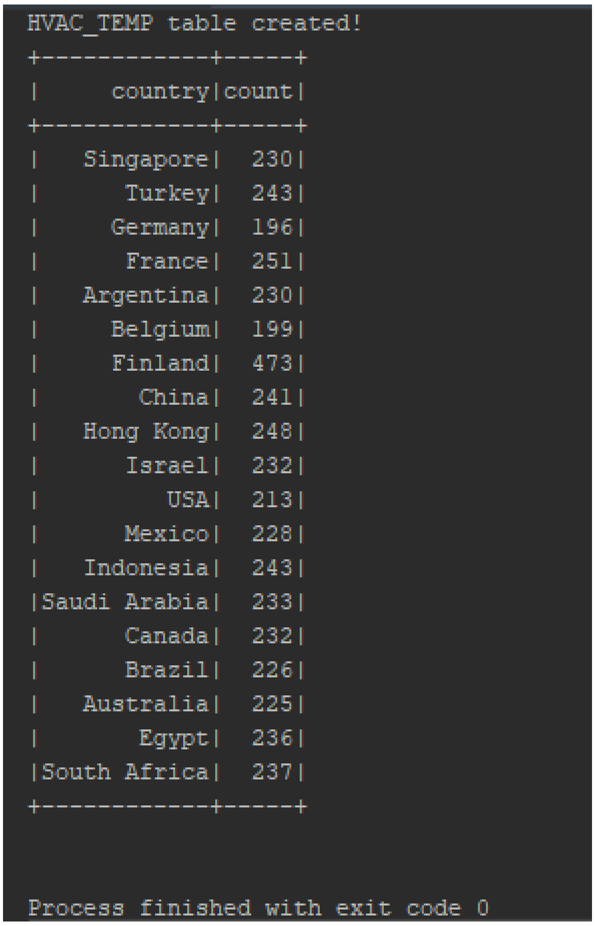
Problem Statement 2: Figure out the number of times, temperature has changed by 5 degrees or more for each country

We have to join HVAC_TEMP and BUILDING_TABLE on BuildingId



```
val x = spark.sql("select a.tempchange,b.country from HVAC_TEMP a join
BUILDING_TABLE b on a.BuildingId=b.BuildId")

val y = x.filter(x("tempchange")>0).groupBy("country").count().show()
```

Then filter by tempchange = “1” and groupBy “country”



```
HVAC_TEMP table created!
+------------+-----+
|     country|count|
+------------+-----+
|   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|
+------------+-----+


Process finished with exit code 0
```