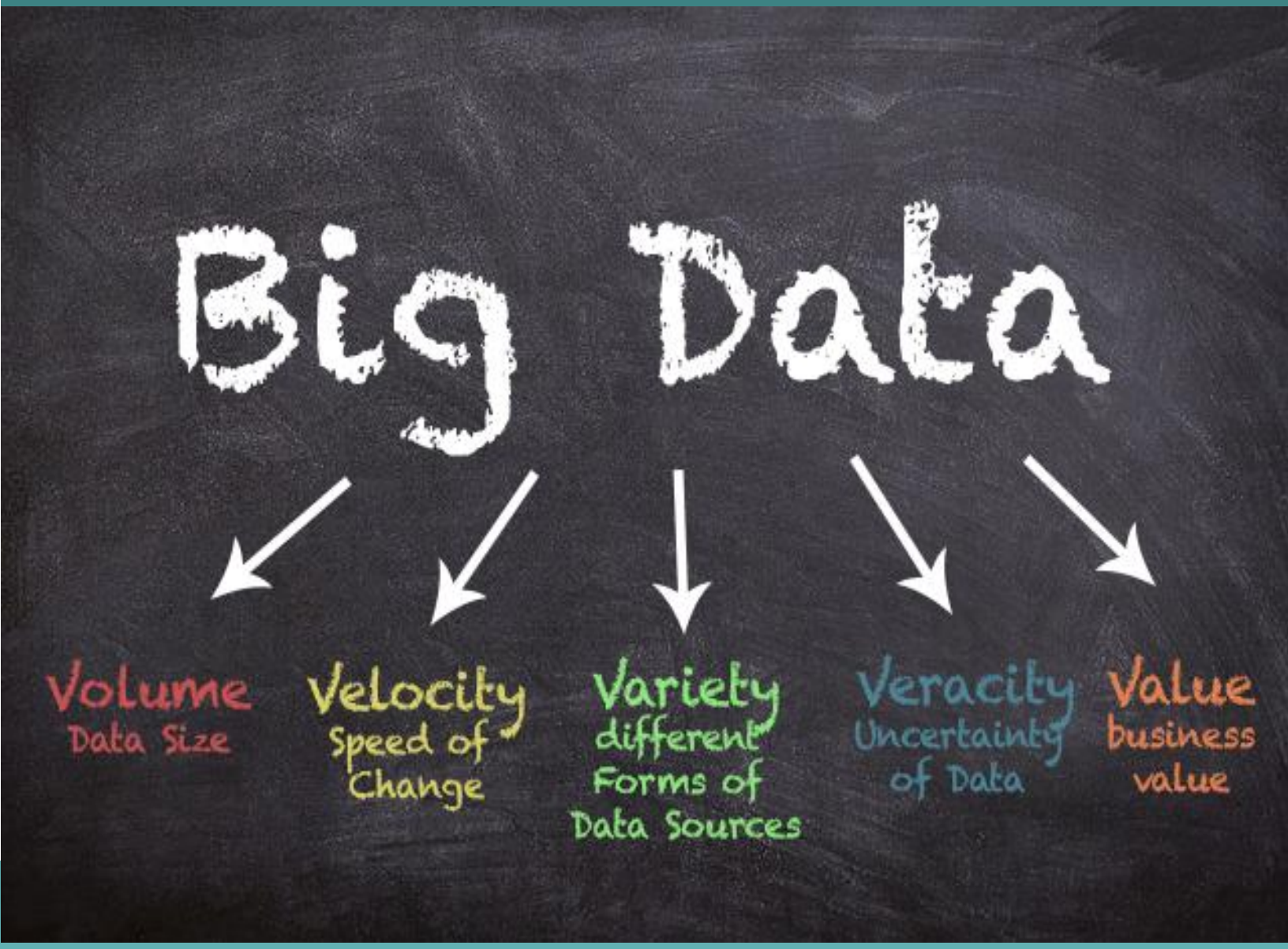
09/11/2018 | By: Ajit K Prasad



Big Data Engineering with Hadoop & Spark

Working with Sensor: Case Study III

Spark

APACHE Spark™

Acadgild

Working with Sensor: Case Study III

This Case Study assignment is aimed at consolidating the concepts that was learnt during the various Scala and Apache Spark, Spark SQL session of the course.

Associated Data Files

Datasets can be downloaded from this ***link***.

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 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.

Dataset Description:

- *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:
   i. Join both the tables.
   ii. Select tempchange and country column
   iii. Filter the rows where tempchange is 1 and count the number of occurrences for each country

Solution:

1. To load ***HVAC.csv*** and ***building.csv*** data to Apache Spark, create manual schemas for the files, which would provide the schema while loading data from both CSV files, as shown below

```
val ManualSchemaHVAC = new StructType(Array(new StructField("Date", StringType, true),
  new StructField("Time", StringType, false),
  new StructField("TargetTemp", LongType, true),
  new StructField("ActualTemp", LongType, false),
  new StructField("System", LongType, false),
  new StructField("SystemAge", LongType, false),
  new StructField("BuildingID", LongType, false)))

val ManualSchemaBuilding = new StructType(Array(new StructField("BuildingID",
LongType, true),
  new StructField("BuildingMgr", StringType, false),
  new StructField("BuildingAge", LongType, true),
  new StructField("HVACproduct", StringType, false),
  new StructField("Country", StringType, false)))
```

Note: StructType is a built-in data type used for Schema definition in Spark SQL, to represent a collection of StructFields that together define a schema or its part.

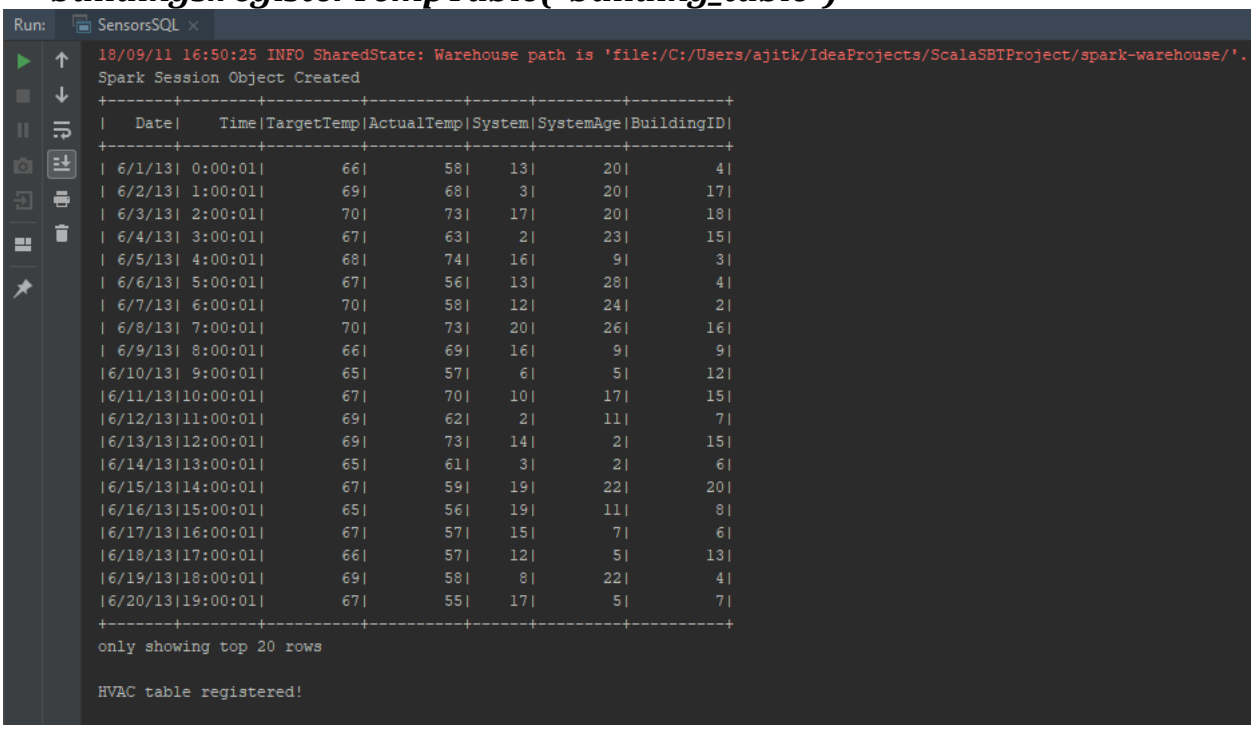
<schema-name> = new
StructType<array_of_columns><Struct_field>(<column_name>, <data_type_of_column>, <nullable _or_not_nullable(true/false)>)

2. Now, the CSV files from local file system to Spark as shown below

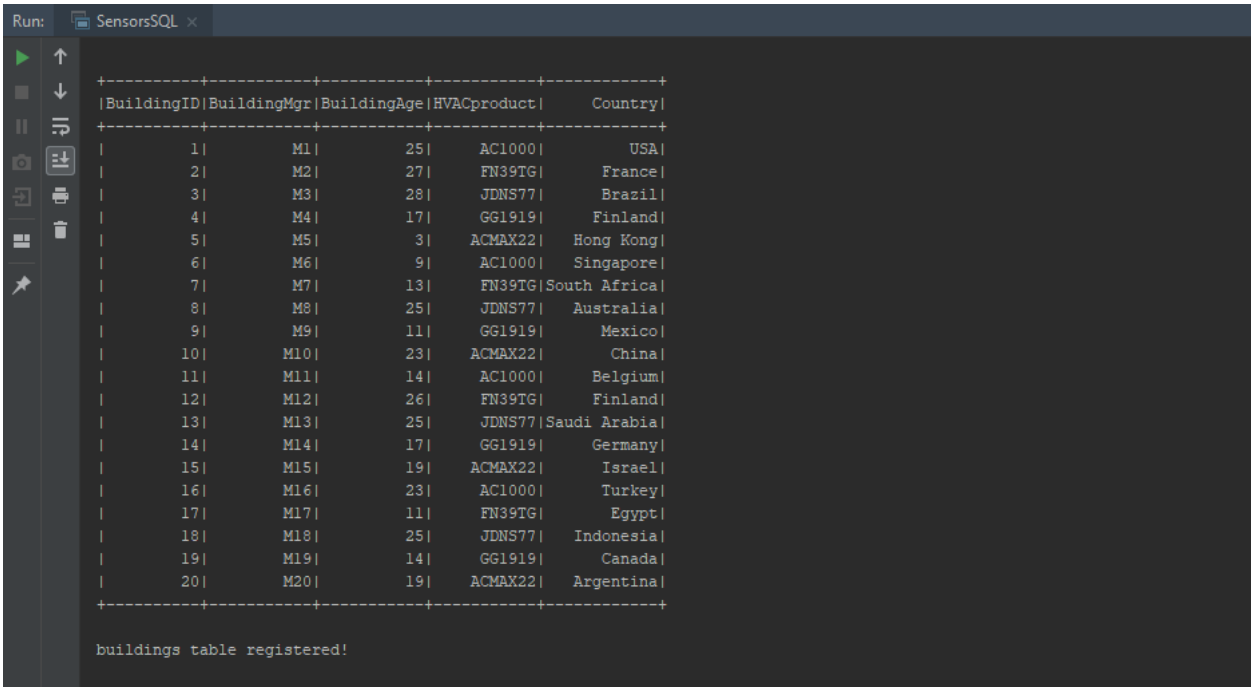
```
//Loading the HVAC.csv data into a Temporary Table
val HVAC = spark.read.format("CSV")
  .option("header", true)
  .schema(ManualSchemaHVAC)
  .load("D:\\AcadGild\\ScalaCaseStudies\\Datasets\\Sensor\\HVAC.csv")
HVAC.show()
HVAC.registerTempTable("HVAC_table")
println("HVAC table registered!\n\n")

//Loading the building.csv data into a Temporary Table
val buildings = spark.read.format("CSV")
  .option("header", true)
  .schema(ManualSchemaBuilding)
  .load("D:\\AcadGild\\ScalaCaseStudies\\Datasets\\Sensor\\building.csv")
buildings.show()
buildings.registerTempTable("building_table")
println("buildings table registered!\n\n")
```

3. The CSV file read format provides various options of which few have been used as follows:
   - Remove the header from the input file
   - Provided the manual schema that we have created in the previous step
   - Provided the path where the CSV file is saved in the local file system

4. Once loading is done, register the tables to TempTable as shown above in the code; ***i.e. HVAC.registerTempTable("HVAC_table"), buildings.registerTempTable("building_table")***



```
Run:  SensorsSQL
18/09/11 16:50:25 INFO SharedState: Warehouse path is 'file:/C:/Users/ajitk/IdeaProjects/ScalaSBTProject/spark-warehouse/'.
Spark Session Object Created
+-------+--------+----------+----------+------+---------+----------+
|   Date|    Time|TargetTemp|ActualTemp|System|SystemAge|BuildingID|
+-------+--------+----------+----------+------+---------+----------+
| 6/1/13| 0:00:01|        66|        58|    13|       20|         4|
| 6/2/13| 1:00:01|        69|        68|     3|       20|        17|
| 6/3/13| 2:00:01|        70|        73|    17|       20|        18|
| 6/4/13| 3:00:01|        67|        63|     2|       23|        15|
| 6/5/13| 4:00:01|        68|        74|    16|        9|         3|
| 6/6/13| 5:00:01|        67|        56|    13|       28|         4|
| 6/7/13| 6:00:01|        70|        58|    12|       24|         2|
| 6/8/13| 7:00:01|        70|        73|    20|       26|        16|
| 6/9/13| 8:00:01|        66|        69|    16|        9|         9|
|6/10/13| 9:00:01|        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

HVAC table registered!
```

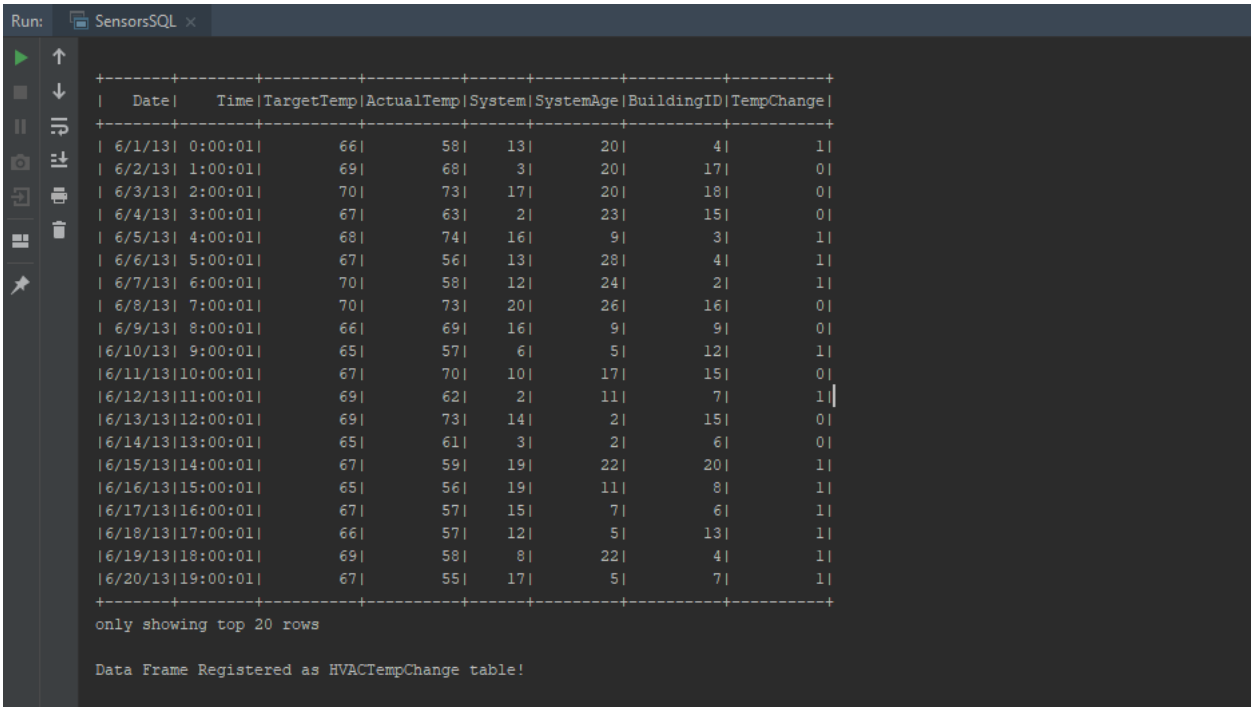


```
Run:  SensorsSQL
+----------+-----------+-----------+-----------+------------+
|BuildingID|BuildingMgr|BuildingAge|HVACproduct|     Country|
+----------+-----------+-----------+-----------+------------+
|         1|         M1|         25|     AC1000|         USA|
|         2|         M2|         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|
+----------+-----------+-----------+-----------+------------+

buildings table registered!
```

5. Now, add a new column, TempChange - set to 1, if there is a change of greater than +/-5 between actual and target temperature

```
//Add a new column, tempChange - set to 1, if there is a change of greater than +/-5
between actual and target temperature
val filterHVAC = spark.sql("select *, IF((TargetTemp-ActualTemp)> 5 ,'1',
IF((TargetTemp-ActualTemp)< -5 ,'1',0)) as TempChange from HVAC_table")
filterHVAC.show()
filterHVAC.registerTempTable("HVACTempChange")
println("Data Frame Registered as HVACTempChange table!\n\n")
```

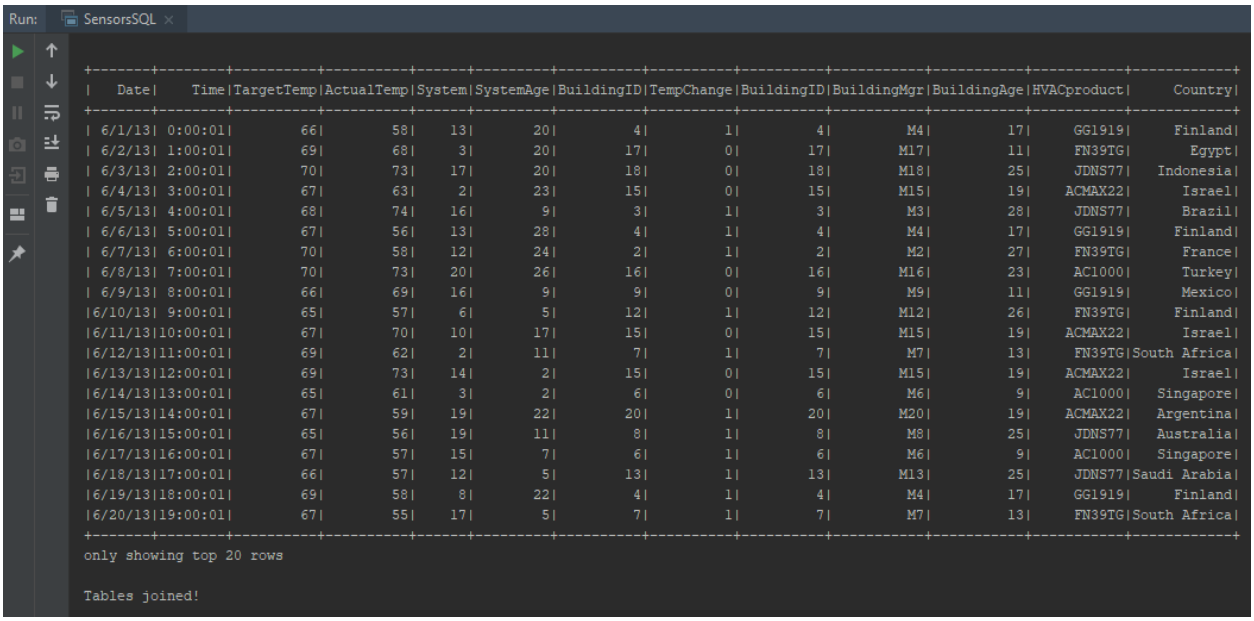
– Here, filtering is based on the difference between the Target temperature and Actual temperature, ***i.e. if the difference is between +/- 5 then give '1' as output if not give output as '0'***.




```
+-------+--------+----------+----------+------+---------+----------+----------+
|   Date|    Time|TargetTemp|ActualTemp|System|SystemAge|BuildingID|TempChange|
+-------+--------+----------+----------+------+---------+----------+----------+
| 6/1/13| 0:00:01|        66|        58|    13|       20|         4|         1|
| 6/2/13| 1:00:01|        69|        68|     3|       20|        17|         0|
| 6/3/13| 2:00:01|        70|        73|    17|       20|        18|         0|
| 6/4/13| 3:00:01|        67|        63|     2|       23|        15|         0|
| 6/5/13| 4:00:01|        68|        74|    16|        9|         3|         1|
| 6/6/13| 5:00:01|        67|        56|    13|       28|         4|         1|
| 6/7/13| 6:00:01|        70|        58|    12|       24|         2|         1|
| 6/8/13| 7:00:01|        70|        73|    20|       26|        16|         0|
| 6/9/13| 8:00:01|        66|        69|    16|        9|         9|         0|
|6/10/13| 9:00:01|        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

Data Frame Registered as HVACTempChange table!
```

6. Now, join both the tables based on BuildingID, i.e. join HVAC and Buildings tables and register the output of join to a temptable called ***"HVACJBUILD"***

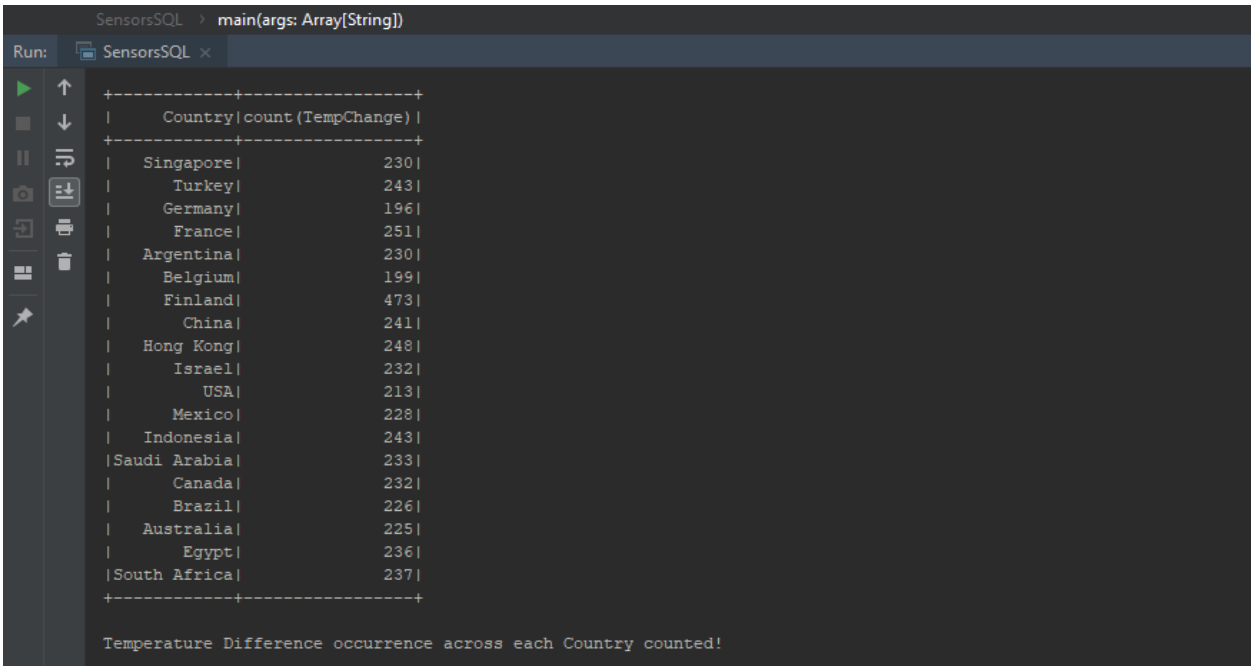
```
//Joining both the tables here
val joinExpression = filterHVAC.col("BuildingID") ===
  buildings.toDF().col("BuildingID")
val HVACJOBUILD = filterHVAC.join(buildings,joinExpression)
HVACJOBUILD.show()
HVACJOBUILD.registerTempTable("HVACJBUILD")
println("Tables joined!\n\n")
```



```
Run:  SensorsSQL

+-------+--------+----------+----------+------+---------+----------+----------+----------+----------+----------+-----------+------------+
|   Date|    Time|TargetTemp|ActualTemp|System|SystemAge|BuildingID|TempChange|BuildingID|BuildingMgr|BuildingAge|HVACproduct|     Country|
+-------+--------+----------+----------+------+---------+----------+----------+----------+----------+----------+-----------+------------+
| 6/1/13| 0:00:01|        66|        58|    13|       20|         4|         1|         4|        M4|        17|     GG1919|     Finland|
| 6/2/13| 1:00:01|        69|        68|     3|       20|        17|         0|        17|       M17|        11|     FN39TG|       Egypt|
| 6/3/13| 2:00:01|        70|        73|    17|       20|        18|         0|        18|       M18|        25|     JDNS77|   Indonesia|
| 6/4/13| 3:00:01|        67|        63|     2|       23|        15|         0|        15|       M15|        19|    ACMAX22|      Israel|
| 6/5/13| 4:00:01|        68|        74|    16|        9|         3|         1|         3|        M3|        28|     JDNS77|      Brazil|
| 6/6/13| 5:00:01|        67|        56|    13|       28|         4|         1|         4|        M4|        17|     GG1919|     Finland|
| 6/7/13| 6:00:01|        70|        58|    12|       24|         2|         1|         2|        M2|        27|     FN39TG|      France|
| 6/8/13| 7:00:01|        70|        73|    20|       26|        16|         0|        16|       M16|        23|     AC1000|      Turkey|
| 6/9/13| 8:00:01|        66|        69|    16|        9|         9|         0|         9|        M9|        11|     GG1919|      Mexico|
|6/10/13| 9:00:01|        65|        57|     6|        5|        12|         1|        12|       M12|        26|     FN39TG|     Finland|
|6/11/13|10:00:01|        67|        70|    10|       17|        15|         0|        15|       M15|        19|    ACMAX22|      Israel|
|6/12/13|11:00:01|        69|        62|     2|       11|         7|         1|         7|        M7|        13|     FN39TG|South Africa|
|6/13/13|12:00:01|        69|        73|    14|        2|        15|         0|        15|       M15|        19|    ACMAX22|      Israel|
|6/14/13|13:00:01|        65|        61|     3|        2|         6|         0|         6|        M6|         9|     AC1000|   Singapore|
|6/15/13|14:00:01|        67|        59|    19|       22|        20|         1|        20|       M20|        19|    ACMAX22|   Argentina|
|6/16/13|15:00:01|        65|        56|    19|       11|         8|         1|         8|        M8|        25|     JDNS77|   Australia|
|6/17/13|16:00:01|        67|        57|    15|        7|         6|         1|         6|        M6|         9|     AC1000|   Singapore|
|6/18/13|17:00:01|        66|        57|    12|        5|        13|         1|        13|       M13|        25|     JDNS77|Saudi Arabia|
|6/19/13|18:00:01|        69|        58|     8|       22|         4|         1|         4|        M4|        17|     GG1919|     Finland|
|6/20/13|19:00:01|        67|        55|    17|        5|         7|         1|         7|        M7|        13|     FN39TG|South Africa|
+-------+--------+----------+----------+------+---------+----------+----------+----------+----------+----------+-----------+------------+
only showing top 20 rows

Tables joined!
```

7. Now let's select TempChange and Country column, then filter the rows where TempChange is 1 and count the number of occurrences for each country

```
//select tempchange and country column, then filter the rows where tempchange is 1 and
count the number of occurrence for each country
val selective = spark.sql("""select TempChange, Country from HVACJBUILD WHERE
TempChange = 1""").toDF()

//Saving the selected fields to variable "selective" and registering the above joined
table to temptable "newSelective"
selective.registerTempTable("newSelective")

//Count the temperature difference across each country
spark.sql("""select Country, count(TempChange) from newSelective Group by
Country""").show()
println("Temperature Difference occurrence across each Country counted!\n\n")
```



```
SensorsSQL › main(args: Array[String])
Run:  SensorsSQL

+------------+-----------------+
|     Country|count(TempChange)|
+------------+-----------------+
|   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|
+------------+-----------------+

Temperature Difference occurrence across each Country counted!
```