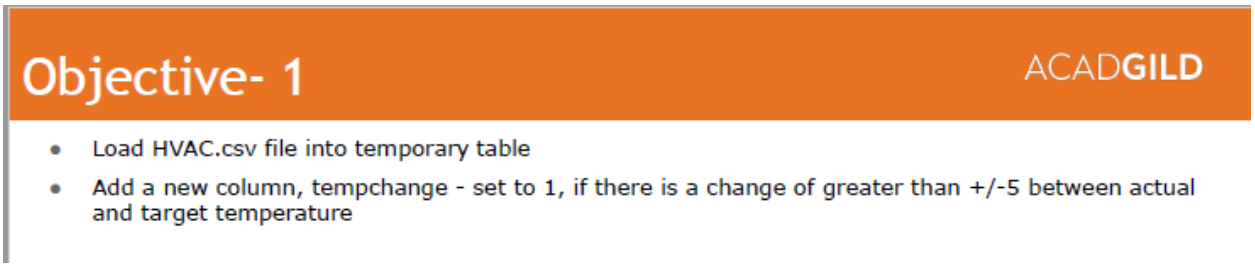
ACADGILD – SPARK SQL II

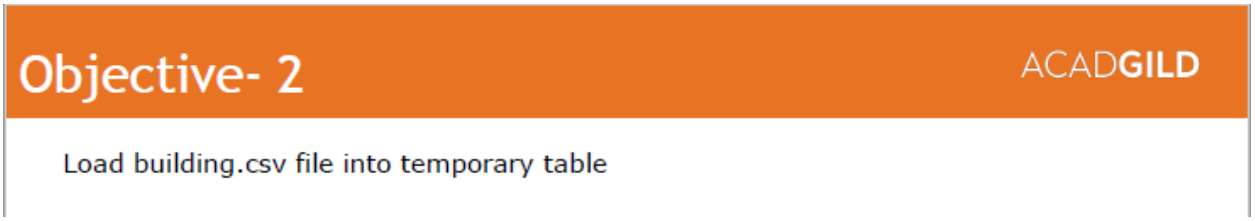
CASE STUDY – Working with Sensor Data

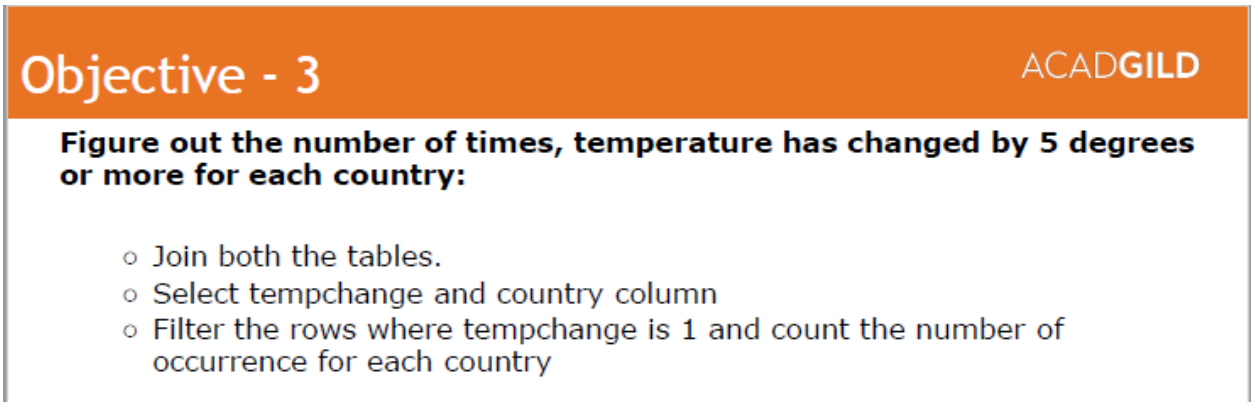
Author: Saravanan Ponnaiah

Date: 04-Feb-2019

Task:







Input Files:

Buildings.csv

HVAC.csv

https://drive.google.com/drive/folders/1npD2CQrLK44Yg1jxlLeF7EEpE9BzVNtV?usp=drive_open

Program:

TemperatureCaseStudy.scala | Task1.scala

```scala
package com.spark.assignment21

import org.apache.spark.SparkContext
import org.apache.spark.SparkContext._
import org.apache.log4j._
import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.types._
import org.apache.spark.sql.types.{StructType, StructField, StringType}
import org.apache.spark.sql.Row
import org.apache.spark.sql.functions._

object TemperatureCaseStudy {
  def main(args : Array[String]) : Unit = {
    Logger.getLogger("org").setLevel(Level.ERROR)

    // Creating a spark context object to run in local machine
    val spark = SparkSession.builder.appName("TemperatureController").master("local").getOrCreate()
    import spark.implicits._

    // Create a dataframe and load the building csv file
    val dfBuildings = spark.sqlContext.read.format("csv")
      .option("header", "true")
      .option("inferSchema", "true")
      .load("E:/Acadgild/Data/building.csv")

      // Create a dataframe and load the hvac csv file
    val dfHvac = spark.sqlContext.read.format("csv")
      .option("header", "true")
      .option("inferSchema", "true")
      .load("E:/Acadgild/Data/HVAC.csv")
```

```scala
    dfBuildings.printSchema()
    dfHvac.printSchema()

    // UDF to determine whether there is a significant variance in temperature
    val isTemperatureChange = (target : Int, actual : Int) => {
      if ((target - actual) > 5 || (target - actual) < -5)
        1
      else
        0
    }

    // Register UDF isTemperatureChange
    spark.udf.register("isTempChange", isTemperatureChange)

    // Add new column TempChange and flag as 1 or 0 based on the temperature variance condition
    val dfNewHvac = dfHvac.withColumn("TempChange", when( ($"TargetTemp" - $"ActualTemp") > 5 || ($"TargetTemp" - $"ActualTemp") < -5, 1  ).otherw

    // Create temporary tables for dataframes
    dfBuildings.createOrReplaceTempView("buildings")
    dfNewHvac.createOrReplaceTempView("hvac")

    val result = spark.sql("SELECT b.Country, COUNT(h.TempChange) AS OccurenceCount FROM buildings AS b INNER JOIN hvac AS h ON b.BuildingID =
    result.show()

    spark.stop()
  }
}
```

- Read both buildings.csv and HVAC.csv files and loaded in to dataframes dfBuildings and dfHvac
- Created a UDF to determine whether there is a significant variance in temperature & registered the UDF
- Create new dataframe and load data from dfHvac with additional column as “TempChange” that flags it as 1 or 0 based on the target and actual temperatures
- Created temp tables for both the dataframes – dfBuildings and dfNewHvac
- Created and executed Spark SQL by joining both temp tables and grouped number of times, the temperature has changed for each country

Execution:

```
+------------+--------------+
|     Country|OccurenceCount|
+------------+--------------+
|   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|
+------------+--------------+
```

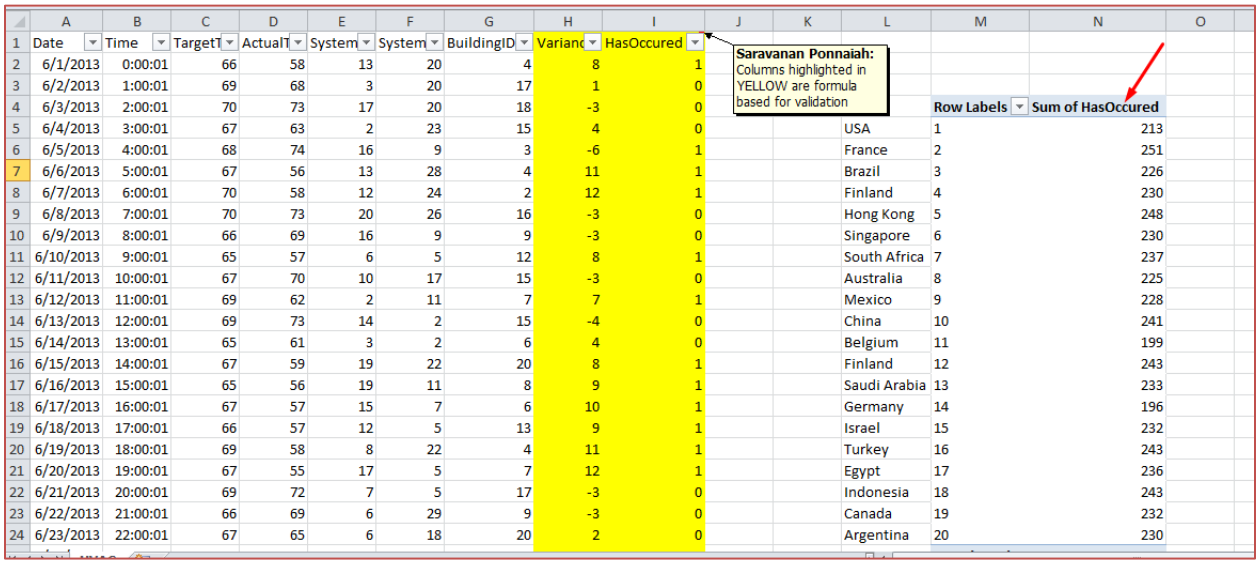
Validation:

For the result validation, we have looked in to the actual csv data file and ensured the result is accurate.

For example, as per buildings.csv file, the building id of USA is 1.

	A	B	C	D	E	F
1	BuildingID	BuildingM	BuildingA	HVACprod	Country	
2	1	M1	25	AC1000	USA	
3	2	M2	27	FN39TG	France	
4	3	M3	28	JDNS77	Brazil	
5	4	M4	17	GG1919	Finland	
6	5	M5	3	ACMAX22	Hong Kong	
7	6	M6	9	AC1000	Singapore	
8	7	M7	13	FN39TG	South Africa	
9	8	M8	25	JDNS77	Australia	
10	9	M9	11	GG1919	Mexico	

In HVAC.csv file, we have added additional columns with formula to get the variance of target and actual temperatures and TempChange column that shows 1 or 0 based on the variance value.



Date	Time	TargetT	ActualT	System	System	BuildingID	Variance	HasOccured
6/1/2013	0:00:01	66	58	13	20	4	8	1
6/2/2013	1:00:01	69	68	3	20	17	1	0
6/3/2013	2:00:01	70	73	17	20	18	-3	0
6/4/2013	3:00:01	67	63	2	23	15	4	0
6/5/2013	4:00:01	68	74	16	9	3	-6	1
6/6/2013	5:00:01	67	56	13	28	4	11	1
6/7/2013	6:00:01	70	58	12	24	2	12	1
6/8/2013	7:00:01	70	73	20	26	16	-3	0
6/9/2013	8:00:01	66	69	16	9	9	-3	0
6/10/2013	9:00:01	65	57	6	5	12	8	1
6/11/2013	10:00:01	67	70	10	17	15	-3	0
6/12/2013	11:00:01	69	62	2	11	7	7	1
6/13/2013	12:00:01	69	73	14	2	15	-4	0
6/14/2013	13:00:01	65	61	3	2	6	4	0
6/15/2013	14:00:01	67	59	19	22	20	8	1
6/16/2013	15:00:01	65	56	19	11	8	9	1
6/17/2013	16:00:01	67	57	15	7	6	10	1
6/18/2013	17:00:01	66	57	12	5	13	9	1
6/19/2013	18:00:01	69	58	8	22	4	11	1
6/20/2013	19:00:01	67	55	17	5	7	12	1
6/21/2013	20:00:01	69	72	7	5	17	-3	0
6/22/2013	21:00:01	66	69	6	29	9	-3	0
6/23/2013	22:00:01	67	65	6	18	20	2	0

Saravanan Ponnaiah:
Columns highlighted in YELLOW are formula based for validation

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

We have compared the above pivot table values with the spark program result (screenshot above) and confirmed both are same.