Case study 3

Moving HVAC and Build to HDFS

```
[acadgild@localhost ~]$ hadoop fs -cat build.csv
18/11/06 13:03:37 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
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
```

```
drwxr-xr-x   - acadgild supergroup          0 2018-07-29 14:59 /user
-rw-r--r--   1 acadgild supergroup        370 2018-07-30 07:35 /wordcount1.txt
-rw-r--r--   1 acadgild supergroup     248592 2018-11-06 12:53 HVAC.csv
```

Starting the spark sessions

```
scala> import org.apache.spark.sql.catalyst.encoders.ExpressionEncoder
import org.apache.spark.sql.catalyst.encoders.ExpressionEncoder

scala> import org.apache.spark.sql.Encoders
import org.apache.spark.sql.Encoders

scala> import org.apache.spark.sql.types._
import org.apache.spark.sql.types._

scala> import org.apache.spark.sql.Row
import org.apache.spark.sql.Row

scala> import org.apache.spark._
import org.apache.spark._

scala> import org.apache.spark.rdd.RDD
import org.apache.spark.rdd.RDD

scala> import org.apache.spark.util.IntParam
import org.apache.spark.util.IntParam

scala> import org.apache.spark.SQLContext
<console>:38: error: object SQLContext is not a member of package org.apache.spark
       import org.apache.spark.SQLContext
              ^

scala> import org.apache.spark.sql.functions._
import org.apache.spark.sql.functions._

scala> import org.apache.spark.sql._
import org.apache.spark.sql._

scala> import org.apache.spark.mlib.stat.Statistics
```

```
scala> val sqlContext = new org.apache.spark.sql.SQLContext(sc);
warning: there was one deprecation warning; re-run with -deprecation for details
sqlContext: org.apache.spark.sql.SQLContext = org.apache.spark.sql.SQLContext@60ed0b9d
```

```
scala> val data = sc.textFile("HVAC.csv");
data: org.apache.spark.rdd.RDD[String] = HVAC.csv MapPartitionsRDD[3] at textFile at <console>:24

scala> val header = data.first()
header: String = Date,Time,TargetTemp,ActualTemp,System,SystemAge,BuildingID
```

use the commands -

case class hvac_cls(Date:String,Time:String,TargetTemp:Int,ActualTemp:In

t,System:Int,SystemAge:Int,BuildingId:Int)

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.registerTempTable("HVAC")

val hvac1 = sqlContext.sql("select *,IF((targettemp - actualtemp) > 5, '1', IF((targettemp - actualtemp) < -5, '1', 0)) AS tempchange from HVAC")

hvac1.registerTempTable("HVAC1")

Loading Building .csv

```
scala> val data2 = sc.textFile("build.csv")
data2: org.apache.spark.rdd.RDD[String] = build.csv MapPartitionsRDD[15] at textFile at <console>:45

scala> val header1 = data2.first()
header1: String = BuildingID,BuildingMgr,BuildingAge,HVACproduct,Country

scala> val data3 = data2.filter(row => row != header1)
data3: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[16] at filter at <console>:49
```

use the commands-

case class building(buildid:Int,buildmgr:String,buildAge:Int,hvacproduct:String,Country:String)

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

build.registerTempTable("building")

Objective 3

Joining both the tables;

```
val build1 = sqlContext.sql("select h.*, b.country,
b.hvacproduct from building b join hvac1 h on b.buildid =
h.buildingid")

val test = build1.map(x => (new
Integer(x(7).toString),x(8).toString))

val test1 = test.filter(x=> {if(x._1==1) true else false})

test1.groupBy(“_2”).count.show
```