Use Case: Working with Sensor Data

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

Solution:

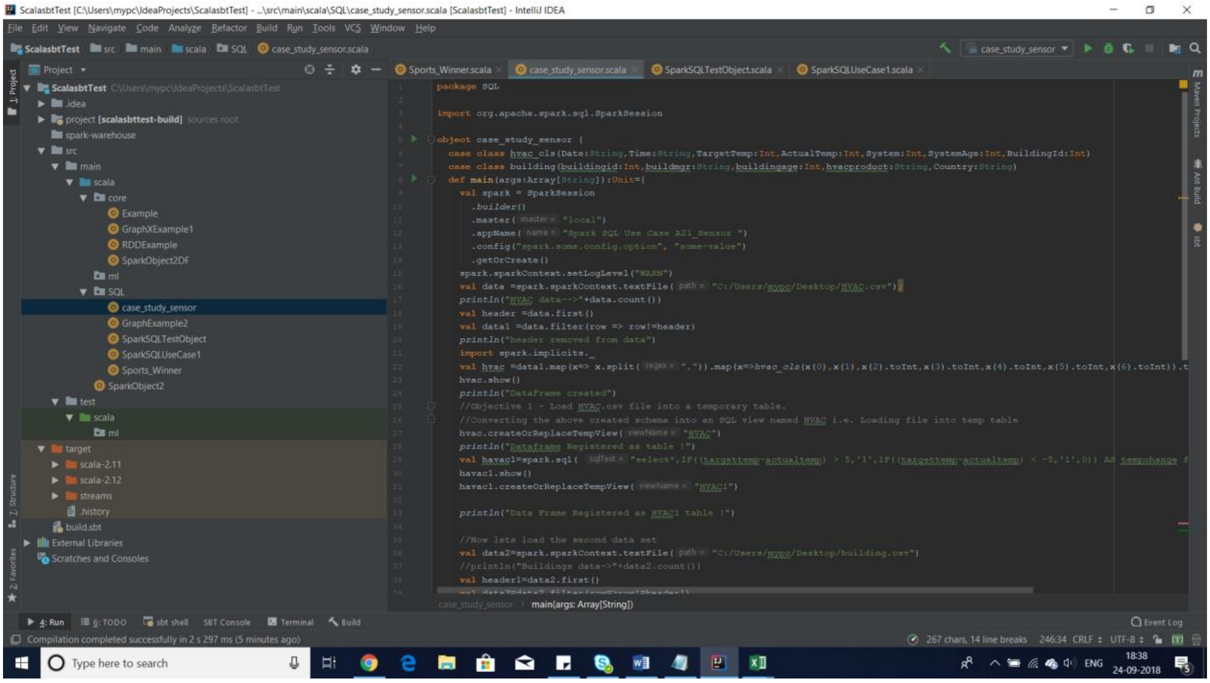
```
package SQL

import org.apache.spark.sql.SparkSession

object case_study_sensor {

////Create a case class hvac_cls and case class building to be used globally inside the main method.
//Inferring the Schema Using Reflection.Automatically converting an RDD containing case classes to a DataFrame.
//The case class defines the schema of the table. The names of the arguments to the case class are read using reflection //and become the names of the columns.

  case class building(buildingid:Int,buildmgr:String,buildingage:Int,hvacproduct:String,Country:String)
case class hvac_cls(Date:String,Time:String,TargetTemp:Int,ActualTemp:Int,System:Int,SystemAge:Int,BuildingId:Int)
  def main(args:Array[String]):Unit={
//Let us create a spark session object
    val spark = SparkSession
      .builder()
      .master("local")
      .appName("Spark SQL Use Case A21_Sensor ")
      .config("spark.some.config.option", "some-value")
```

```
     .getOrCreate()
//Set the log level as warning
    spark.sparkContext.setLogLevel("WARN")
// Reading a file for the hvac_cls schema created above
    val data =spark.sparkContext.textFile("C:/Users/mypc/Desktop/HVAC.csv");
    println("HVAC data-->"+data.count())
// Storing the first line that is header of the file into another variable
    val header =data.first()
// Selecting all other lines into data1 excluding the header.
    val data1 =data.filter(row => row!=header)
    println("header removed from 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("DataFrame created")
    //Objective 1 - Load HVAC.csv file into a temporary table.
    //Converting the above created schema into an SQL view named HVAC i.e. Loading file into temp table
    hvac.createOrReplaceTempView("HVAC")
    println("Dataframe Registered as table !")


         //Add a new column  a new column, tempchange - set to 1, if there is a change of
         greater than +/-5 between actual and target temperature
         //Added a new column based upon a condition
```

```
   val havac1=spark.sql( "select*,IF((targettemp-actualtemp) > 5,'1',IF((targettemp-actualtemp) < -
5,'1',0)) AS tempchange from HVAC")

   havac1.show()

   havac1.createOrReplaceTempView("HVAC1")


   println("Data Frame Registered as HVAC1 table !")




                    //loading the second data
              // Reading a file for the building schema created above




val data2=spark.sparkContext.textFile("C:/Users/mypc/Desktop/building.csv")

   //println("Buildings data->"+data2.count())


// Storing the first line that is header of the file into another variable

   val header1=data2.first()

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

// Selecting all other lines into data1 excluding the header.


   println("header removed from building data")

   println("Buildings data->"+data3.count())

//Now let us create the building dataframe

//converting data3 into dataframe splitting on comma character(,)

   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()

//Converting the above created schema into an SQL view named building i.e. Loading
building.csv as a temporary table

   build.createOrReplaceTempView("building")
```

```
  println("Buildings data registered as building table")
//Now join the two tables building and hvac1 on buildingid
  val build1 = spark.sql("select h.*, b.country, b.buildingage,b.buildmgr,b.hvacproduct from building b join hvac1 h on b.buildingid = h.buildingid")


  build1.show()
//Selecting tempchange and country column from build1 DataFrame created above
  val tempCountry= build1.map(x =>(new Integer(x(7).toString),x(8).toString))
  tempCountry.show()
//filter the values where tempchange is 1
val tempCountryOnes =tempCountry.filter(x=>{(x._1==1)true else false })
tempCountryOnes.show()


//creating a schema with column name as “Count”, ”Country”
val colname = Seq(“Count”, ”Country”)
//Schema of tempCountryOnes is (_1,_2),converting it into ("Count","Country")   in temp_country_count by specifying above two colum as column names
val temp_country_count  = temCountryOnes.toDF(colname:_*)
val final_df = temp_country_count.groupBy("Country").count
final_df.orderBy($"count".desc).show

  }
        }
```

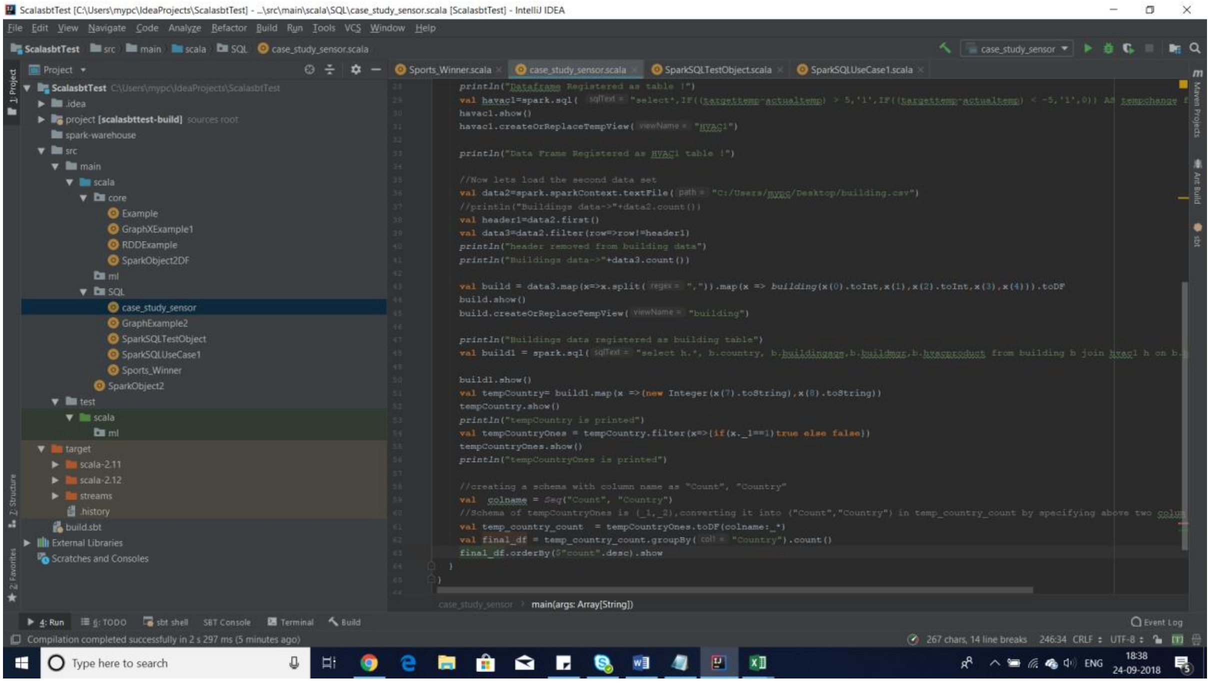
Screenshots:



ScalasbtTest [C:\Users\mypc\IdeaProjects\ScalasbtTest] - ...\src\main\scala\SQL\case_study_sensor.scala [ScalasbtTest] - IntelliJ IDEA

```scala
package SQL

import org.apache.spark.sql.SparkSession

object case_study_sensor {
  case class hvac_cls(Date:String,Time:String,TargetTemp:Int,ActualTemp:Int,System:Int,SystemAge:Int,BuildingId:Int)
  case class building(buildingid:Int,buildmgr:String,buildingage:Int,hvacproduct:String,Country:String)
  def main(args:Array[String]):Unit={
    val spark = SparkSession
      .builder()
      .master("local")
      .appName("Spark SQL Use Case A21_Sensor ")
      .config("spark.some.config.option", "some-value")
      .getOrCreate()
    spark.sparkContext.setLogLevel("WARN")
    val data =spark.sparkContext.textFile("C:/Users/mypc/Desktop/HVAC.csv")
    println("HVAC data-->"+data.count())
    val header =data.first()
    val data1 =data.filter(row => row!=header)
    println("header removed from data")
    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)).t
    hvac.show()
    println("DataFrame created")
    //Objective 1 - Load HVAC.csv file into a temporary table.
    //Converting the above created schema into an SQL view named HVAC i.e. Loading file into temp table
    hvac.createOrReplaceTempView("HVAC")
    println("Dataframe Registered as table !")
    val havac1=spark.sql("select*,IF((targettemp-actualtemp) > 5,'1',IF((targettemp-actualtemp) < -5,'1',0)) AS tempchange f
    havac1.show()
    havac1.createOrReplaceTempView("HVAC1")

    println("Data Frame Registered as HVAC1 table !")

    //Now lets load the second data set
    val data2=spark.sparkContext.textFile("C:/Users/mypc/Desktop/building.csv")
    //println("Buildings data->"+data2.count())
    val header1=data2.first()
```

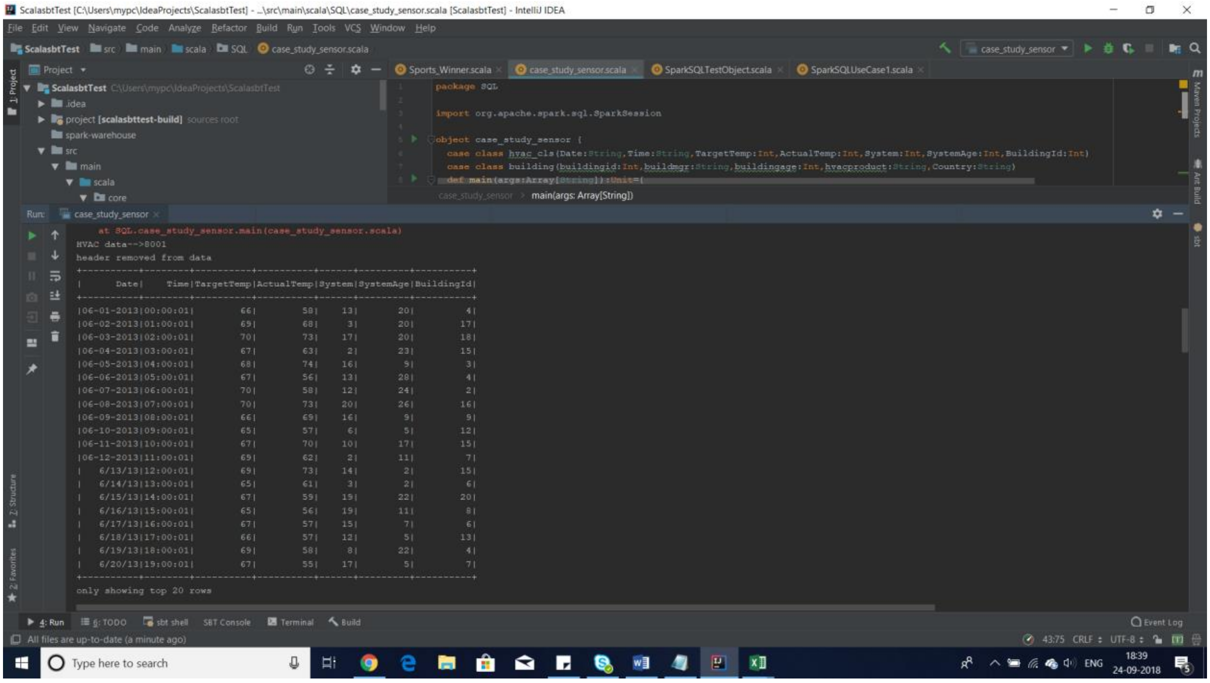
Compilation completed successfully in 2 s 297 ms (5 minutes ago)



ScalasbtTest [C:\Users\mypc\IdeaProjects\ScalasbtTest] - ...\src\main\scala\SQL\case_study_sensor.scala [ScalasbtTest] - IntelliJ IDEA

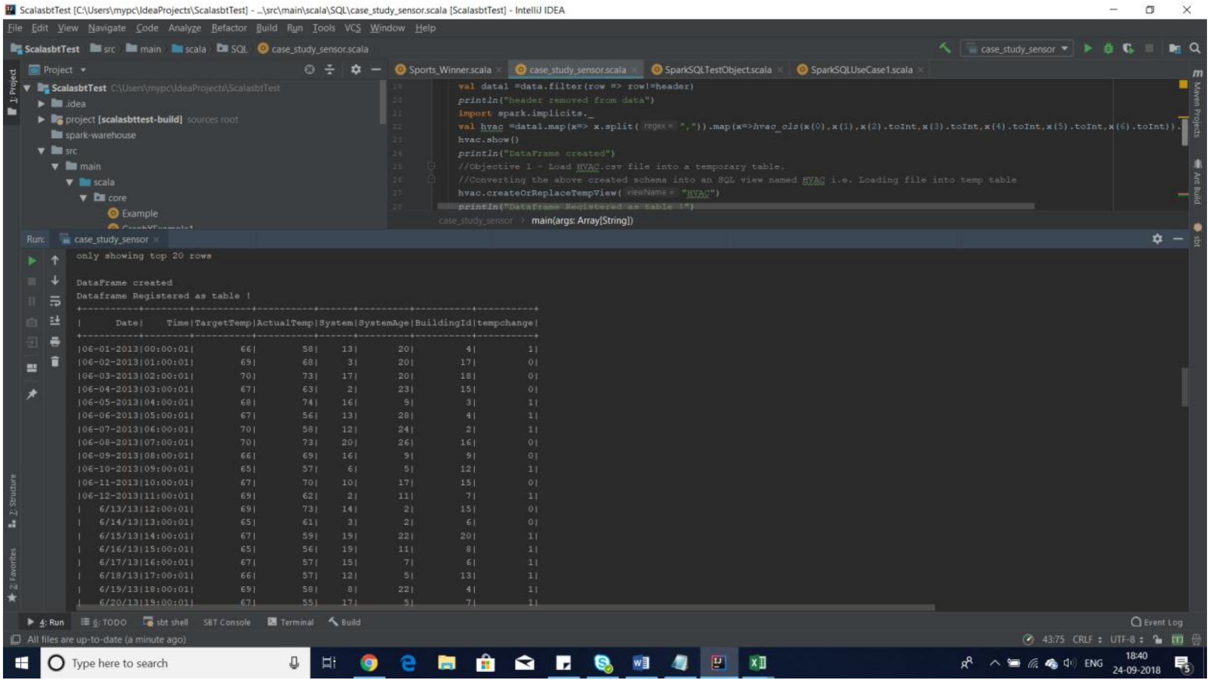
```scala
    println("Dataframe Registered as table !")
    val havac1=spark.sql("select*,IF((targettemp-actualtemp) > 5,'1',IF((targettemp-actualtemp) < -5,'1',0)) AS tempchange f
    havac1.show()
    havac1.createOrReplaceTempView("HVAC1")

    println("Data Frame Registered as HVAC1 table !")

    //Now lets load the second data set
    val data2=spark.sparkContext.textFile("C:/Users/mypc/Desktop/building.csv")
    //println("Buildings data->"+data2.count())
    val header1=data2.first()
    val data3=data2.filter(row=>row!=header1)
    println("header removed from building data")
    println("Buildings data->"+data3.count())

    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.createOrReplaceTempView("building")

    println("Buildings data registered as building table")
    val build1 = spark.sql("select h.*, b.country, b.buildingage,b.buildmgr,b.hvacproduct from building b join hvac1 h on b.

    build1.show()
    val tempCountry= build1.map(x =>(new Integer(x(7).toString),x(8).toString))
    tempCountry.show()
    println("tempCountry is printed")
    val tempCountryOnes = tempCountry.filter(x=>{if(x._1==1)true else false})
    tempCountryOnes.show()
    println("tempCountryOnes is printed")

    //creating a schema with column name as "Count", "Country"
    val  colname = Seq("Count", "Country")
    //Schema of tempCountryOnes is (_1,_2),converting it into ("Count","Country") in temp_country_count by specifying above two colu
    val temp_country_count  = tempCountryOnes.toDF(colname:_*)
    val final_df = temp_country_count.groupBy("Country").count()
    final_df.orderBy($"count".desc).show
  }
}
```

Compilation completed successfully in 2 s 297 ms (5 minutes ago)


ScalasbtTest [C:\Users\mypc\IdeaProjects\ScalasbtTest] - ...\src\main\scala\SQL\case_study_sensor.scala [ScalasbtTest] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
ScalasbtTest src main scala SQL case_study_sensor.scala
case_study_sensor
Project
ScalasbtTest C:\Users\mypc\IdeaProjects\ScalasbtTest
.idea
project [scalasbttest-build] sources root
spark-warehouse
src
main
scala
core
Sports_Winner.scala
case_study_sensor.scala
SparkSQLTestObject.scala
SparkSQLUseCase1.scala
case_study_sensor > main(args: Array[String])
Run: case_study_sensor
4: Run
6: TODO
sbt shell
SBT Console
Terminal
Build
Event Log
All files are up-to-date (a minute ago)
43:75 CRLF UTF-8
Type here to search
ENG
18:39
24-09-2018

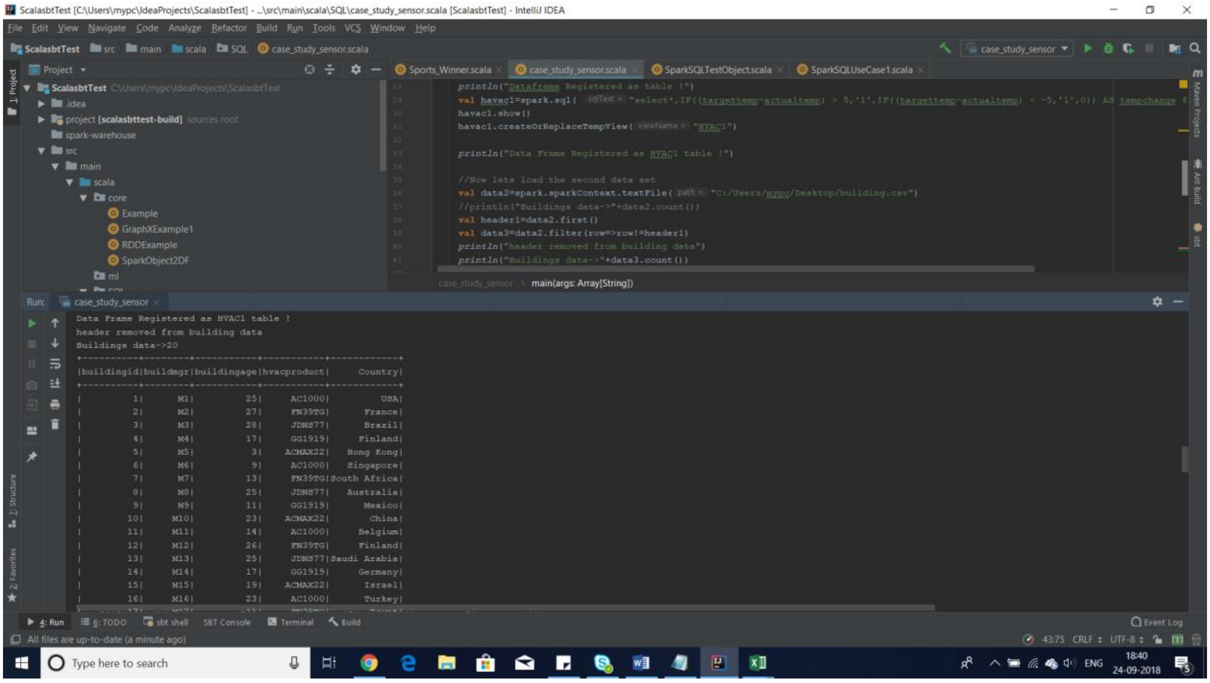
```scala
package SQL

import org.apache.spark.sql.SparkSession

object case_study_sensor {
  case class hvac_cls(Date:String,Time:String,TargetTemp:Int,ActualTemp:Int,System:Int,SystemAge:Int,BuildingId:Int)
  case class building(buildingid:Int,buildmgr:String,buildingage:Int,hvacproduct:String,Country:String)
  def main(args:Array[String]):Unit={
```

```
    at SQL.case_study_sensor.main(case_study_sensor.scala)
HVAC data-->8001
header removed from data
+----------+--------+----------+----------+------+---------+----------+
|      Date|    Time|TargetTemp|ActualTemp|System|SystemAge|BuildingId|
+----------+--------+----------+----------+------+---------+----------+
|06-01-2013|00:00:01|        66|        58|    13|       20|         4|
|06-02-2013|01:00:01|        69|        68|     3|       20|        17|
|06-03-2013|02:00:01|        70|        73|    17|       20|        18|
|06-04-2013|03:00:01|        67|        63|     2|       23|        15|
|06-05-2013|04:00:01|        68|        74|    16|        9|         3|
|06-06-2013|05:00:01|        67|        56|    13|       28|         4|
|06-07-2013|06:00:01|        70|        58|    12|       24|         2|
|06-08-2013|07:00:01|        70|        73|    20|       26|        16|
|06-09-2013|08:00:01|        66|        69|    16|        9|         9|
|06-10-2013|09:00:01|        65|        57|     6|        5|        12|
|06-11-2013|10:00:01|        67|        70|    10|       17|        15|
|06-12-2013|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
```


ScalasbtTest [C:\Users\mypc\IdeaProjects\ScalasbtTest] - ...\src\main\scala\SQL\case_study_sensor.scala [ScalasbtTest] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
ScalasbtTest src main scala SQL case_study_sensor.scala
case_study_sensor
Project
ScalasbtTest C:\Users\mypc\IdeaProjects\ScalasbtTest
.idea
project [scalasbttest-build] sources root
spark-warehouse
src
main
scala
core
Example
Sports_Winner.scala
case_study_sensor.scala
SparkSQLTestObject.scala
SparkSQLUseCase1.scala
case_study_sensor > main(args: Array[String])
Run: case_study_sensor
4: Run
6: TODO
sbt shell
SBT Console
Terminal
Build
Event Log
All files are up-to-date (a minute ago)
43:75 CRLF UTF-8
Type here to search
ENG
18:40
24-09-2018

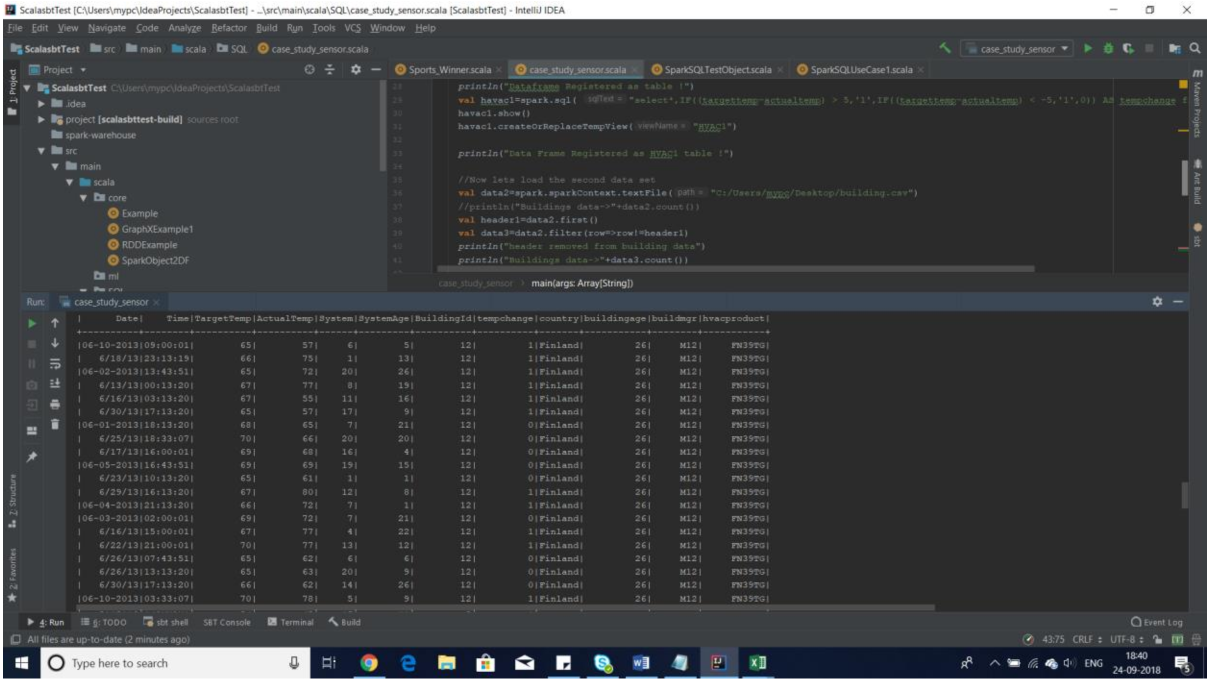
```scala
    val data1 =data.filter(row => row!=header)
    println("header removed from data")
    import spark.implicits._
    val hvac =data1.map(x=> x.split( regex = ",")).map(x=>hvac_cls(x(0),x(1),x(2).toInt,x(3).toInt,x(4).toInt,x(5).toInt,x(6).toInt)).
    hvac.show()
    println("DataFrame created")
    //Objective 1 - Load HVAC.csv file into a temporary table.
    //Converting the above created schema into an SQL view named HVAC i.e. Loading file into temp table
    hvac.createOrReplaceTempView( viewName = "HVAC")
    println("Dataframe Registered as table !")
```

```
only showing top 20 rows

DataFrame created
Dataframe Registered as table !
+----------+--------+----------+----------+------+---------+----------+----------+
|      Date|    Time|TargetTemp|ActualTemp|System|SystemAge|BuildingId|tempchange|
+----------+--------+----------+----------+------+---------+----------+----------+
|06-01-2013|00:00:01|        66|        58|    13|       20|         4|         1|
|06-02-2013|01:00:01|        69|        68|     3|       20|        17|         0|
|06-03-2013|02:00:01|        70|        73|    17|       20|        18|         0|
|06-04-2013|03:00:01|        67|        63|     2|       23|        15|         0|
|06-05-2013|04:00:01|        68|        74|    16|        9|         3|         1|
|06-06-2013|05:00:01|        67|        56|    13|       28|         4|         1|
|06-07-2013|06:00:01|        70|        58|    12|       24|         2|         1|
|06-08-2013|07:00:01|        70|        73|    20|       26|        16|         0|
|06-09-2013|08:00:01|        66|        69|    16|        9|         9|         0|
|06-10-2013|09:00:01|        65|        57|     6|        5|        12|         1|
|06-11-2013|10:00:01|        67|        70|    10|       17|        15|         0|
|06-12-2013|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|
```



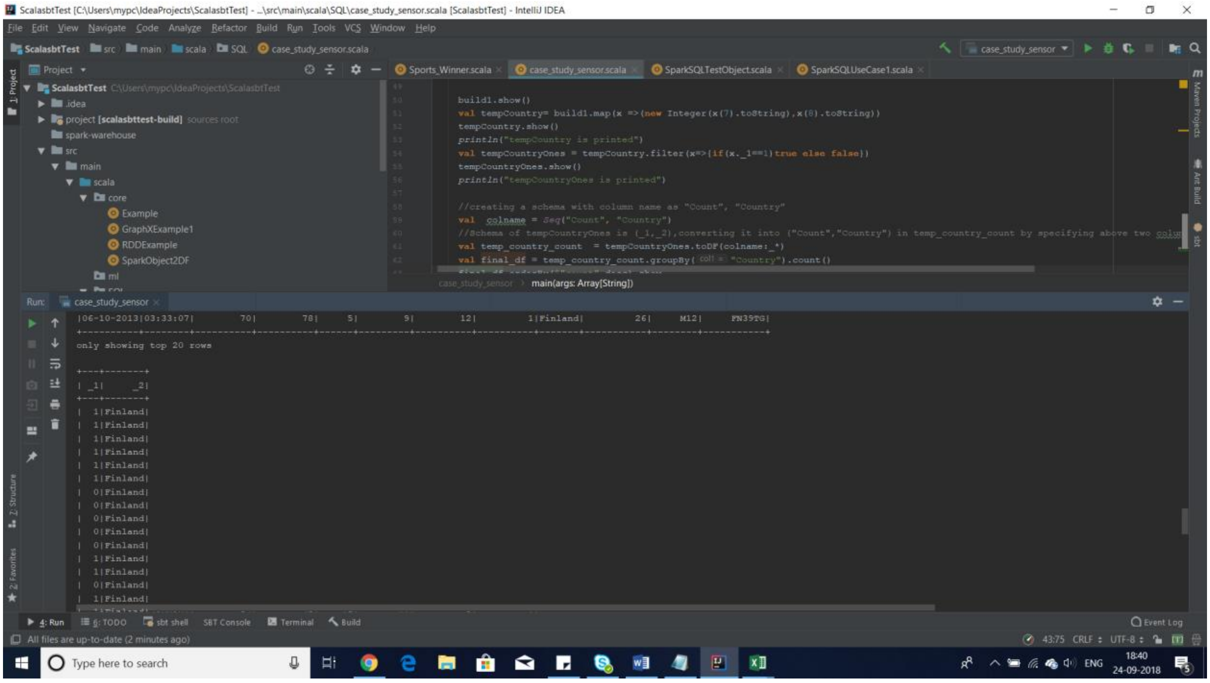
```
println("Dataframe Registered as table !")
val havac1=spark.sql( sqlText = "select*,IF((targettemp-actualtemp) > 5,'1',IF((targettemp-actualtemp) < -5,'1',0)) AS tempchange f
havac1.show()
havac1.createOrReplaceTempView( viewName = "HVAC1")

println("Data Frame Registered as HVAC1 table !")

//Now lets load the second data set
val data2=spark.sparkContext.textFile( path = "C:/Users/mypc/Desktop/building.csv")
//println("Buildings data->"+data2.count())
val header1=data2.first()
val data3=data2.filter(row=>row!=header1)
println("header removed from building data")
println("Buildings data->"+data3.count())
```

```
Data Frame Registered as HVAC1 table !
header removed from building data
Buildings data->20
+----------+--------+-----------+-----------+------------+
|buildingid|buildmgr|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|
```



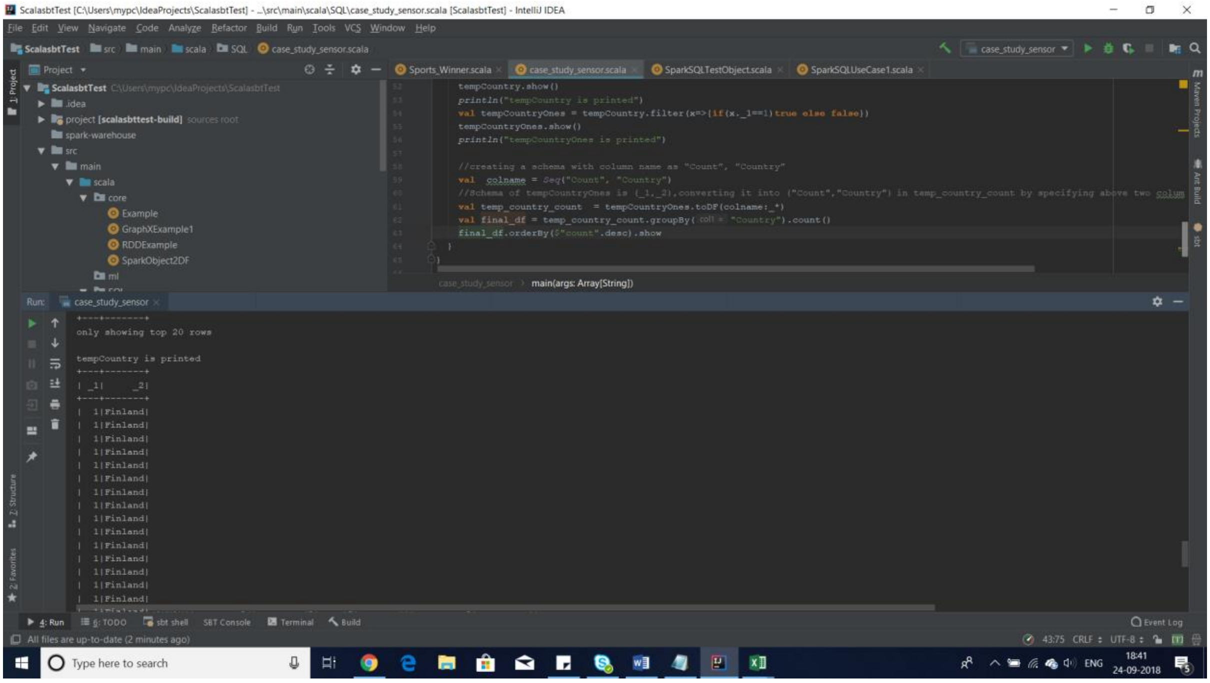
```
println("Dataframe Registered as table !")
val havac1=spark.sql( sqlText = "select*,IF((targettemp-actualtemp) > 5,'1',IF((targettemp-actualtemp) < -5,'1',0)) AS tempchange f
havac1.show()
havac1.createOrReplaceTempView( viewName = "HVAC1")

println("Data Frame Registered as HVAC1 table !")

//Now lets load the second data set
val data2=spark.sparkContext.textFile( path = "C:/Users/mypc/Desktop/building.csv")
//println("Buildings data->"+data2.count())
val header1=data2.first()
val data3=data2.filter(row=>row!=header1)
println("header removed from building data")
println("Buildings data->"+data3.count())
```

```
|      Date|    Time|TargetTemp|ActualTemp|System|SystemAge|BuildingId|tempchange|country|buildingage|buildmgr|hvacproduct|
+----------+--------+----------+----------+------+---------+----------+----------+-------+-----------+--------+-----------+
|06-10-2013|09:00:01|        65|        57|     6|        5|        12|         1|Finland|         26|     M12|     FN39TG|
|   6/18/13|23:13:19|        66|        75|     1|       13|        12|         1|Finland|         26|     M12|     FN39TG|
|06-02-2013|13:43:51|        65|        72|    20|       26|        12|         1|Finland|         26|     M12|     FN39TG|
|   6/13/13|00:13:20|        67|        77|     8|       19|        12|         1|Finland|         26|     M12|     FN39TG|
|   6/16/13|03:13:20|        67|        55|    11|       16|        12|         1|Finland|         26|     M12|     FN39TG|
|   6/30/13|17:13:20|        65|        57|    17|        9|        12|         1|Finland|         26|     M12|     FN39TG|
|06-01-2013|18:13:20|        68|        65|     7|       21|        12|         0|Finland|         26|     M12|     FN39TG|
|   6/25/13|18:33:07|        70|        66|    20|       20|        12|         0|Finland|         26|     M12|     FN39TG|
|   6/17/13|16:00:01|        69|        68|    16|        4|        12|         0|Finland|         26|     M12|     FN39TG|
|06-05-2013|16:43:51|        69|        69|    19|       15|        12|         0|Finland|         26|     M12|     FN39TG|
|   6/23/13|10:13:20|        65|        61|     1|        1|        12|         0|Finland|         26|     M12|     FN39TG|
|   6/29/13|16:13:20|        67|        80|    12|        8|        12|         1|Finland|         26|     M12|     FN39TG|
|06-04-2013|21:13:20|        66|        72|     7|        1|        12|         1|Finland|         26|     M12|     FN39TG|
|06-03-2013|02:00:01|        69|        72|     7|       21|        12|         0|Finland|         26|     M12|     FN39TG|
|   6/16/13|15:00:01|        67|        77|     4|       22|        12|         1|Finland|         26|     M12|     FN39TG|
|   6/22/13|21:00:01|        70|        77|    13|       12|        12|         1|Finland|         26|     M12|     FN39TG|
|   6/26/13|07:43:51|        65|        62|     6|        6|        12|         0|Finland|         26|     M12|     FN39TG|
|   6/26/13|13:13:20|        65|        63|    20|        9|        12|         0|Finland|         26|     M12|     FN39TG|
|   6/30/13|17:13:20|        66|        62|    14|       26|        12|         0|Finland|         26|     M12|     FN39TG|
|06-10-2013|03:33:07|        70|        78|     5|        9|        12|         1|Finland|         26|     M12|     FN39TG|
```


ScalasbtTest [C:\Users\mypc\IdeaProjects\ScalasbtTest] - ...\src\main\scala\SQL\case_study_sensor.scala [ScalasbtTest] - IntelliJ IDEA
Sports_Winner.scala
case_study_sensor.scala
SparkSQLTestObject.scala
SparkSQLUseCase1.scala
Run: case_study_sensor
All files are up-to-date (2 minutes ago)

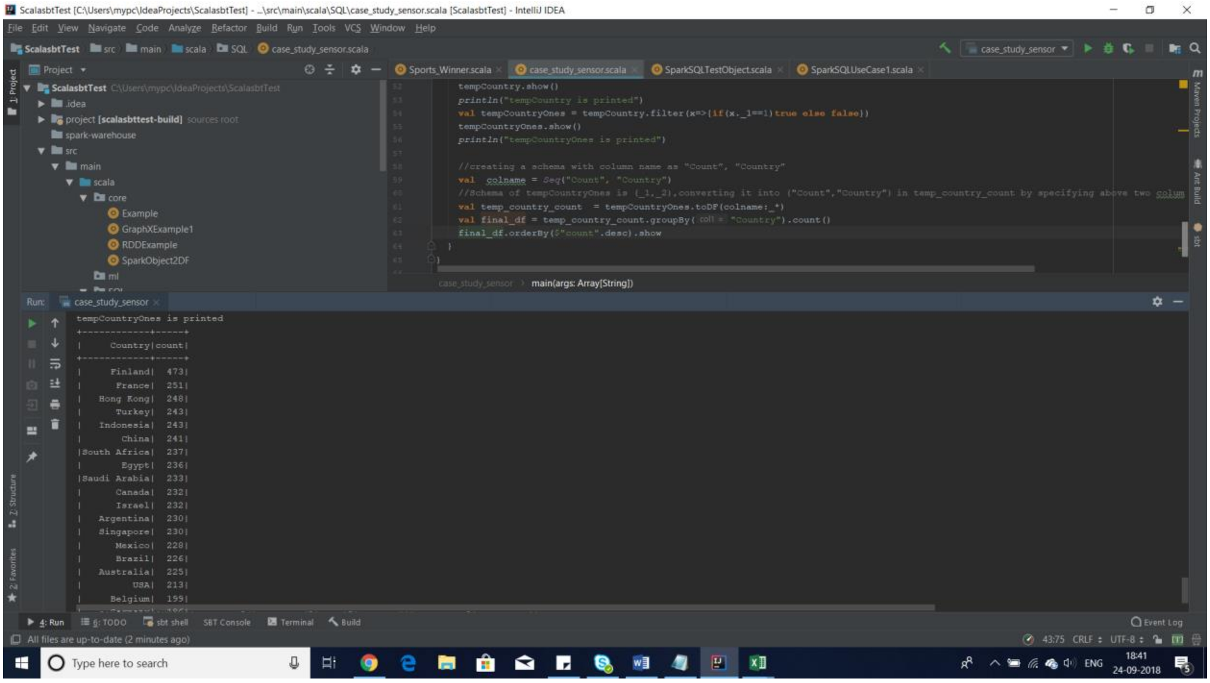
```scala
    buildl.show()
    val tempCountry= buildl.map(x =>(new Integer(x(7).toString),x(8).toString))
    tempCountry.show()
    println("tempCountry is printed")
    val tempCountryOnes = tempCountry.filter(x=>{if(x._1==1)true else false})
    tempCountryOnes.show()
    println("tempCountryOnes is printed")

    //creating a schema with column name as "Count", "Country"
    val  colname = Seq("Count", "Country")
    //Schema of tempCountryOnes is (_1,_2),converting it into ("Count","Country") in temp_country_count by specifying above two colu
    val temp_country_count  = tempCountryOnes.toDF(colname:_*)
    val final_df = temp_country_count.groupBy( col1 = "Country").count()
```

```
|06-10-2013|03:33:07|       70|       78|    5|       9|       12|         1|Finland|         26|      M12|     FN39TG|
+----------+--------+---------+---------+-----+--------+---------+---------+-------+-----------+--------+-----------+
only showing top 20 rows

+---+-------+
| _1|     _2|
+---+-------+
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  0|Finland|
|  0|Finland|
|  0|Finland|
|  0|Finland|
|  0|Finland|
|  1|Finland|
|  1|Finland|
|  0|Finland|
|  1|Finland|
```


ScalasbtTest [C:\Users\mypc\IdeaProjects\ScalasbtTest] - ...\src\main\scala\SQL\case_study_sensor.scala [ScalasbtTest] - IntelliJ IDEA
Sports_Winner.scala
case_study_sensor.scala
SparkSQLTestObject.scala
SparkSQLUseCase1.scala
Run: case_study_sensor
All files are up-to-date (2 minutes ago)

```scala
    tempCountry.show()
    println("tempCountry is printed")
    val tempCountryOnes = tempCountry.filter(x=>{if(x._1==1)true else false})
    tempCountryOnes.show()
    println("tempCountryOnes is printed")

    //creating a schema with column name as "Count", "Country"
    val  colname = Seq("Count", "Country")
    //Schema of tempCountryOnes is (_1,_2),converting it into ("Count","Country") in temp_country_count by specifying above two colum
    val temp_country_count  = tempCountryOnes.toDF(colname:_*)
    val final_df = temp_country_count.groupBy( col1 = "Country").count()
    final_df.orderBy($"count".desc).show
  }
}
```

```
+---+-------+
only showing top 20 rows

tempCountry is printed
+---+-------+
| _1|     _2|
+---+-------+
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
```

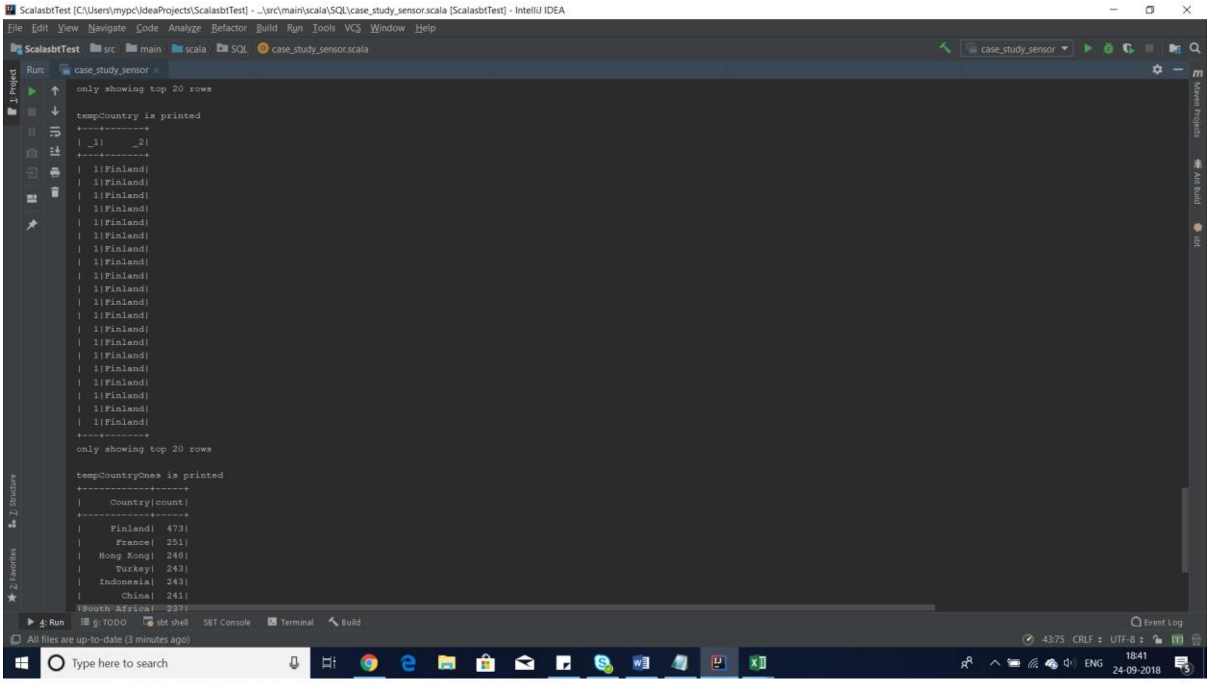


ScalasbtTest [C:\Users\mypc\IdeaProjects\ScalasbtTest] - ...\src\main\scala\SQL\case_study_sensor.scala [ScalasbtTest] - IntelliJ IDEA

```
        tempCountry.show()
        println("tempCountry is printed")
        val tempCountryOnes = tempCountry.filter(x=>{if(x._1==1)true else false})
        tempCountryOnes.show()
        println("tempCountryOnes is printed")

        //creating a schema with column name as "Count", "Country"
        val  colname = Seq("Count", "Country")
        //Schema of tempCountryOnes is (_1,_2),converting it into ("Count","Country") in temp_country_count by specifying above two colum
        val temp_country_count  = tempCountryOnes.toDF(colname:_*)
        val final_df = temp_country_count.groupBy( col1 = "Country").count()
        final_df.orderBy($"count".desc).show
    }
}
```

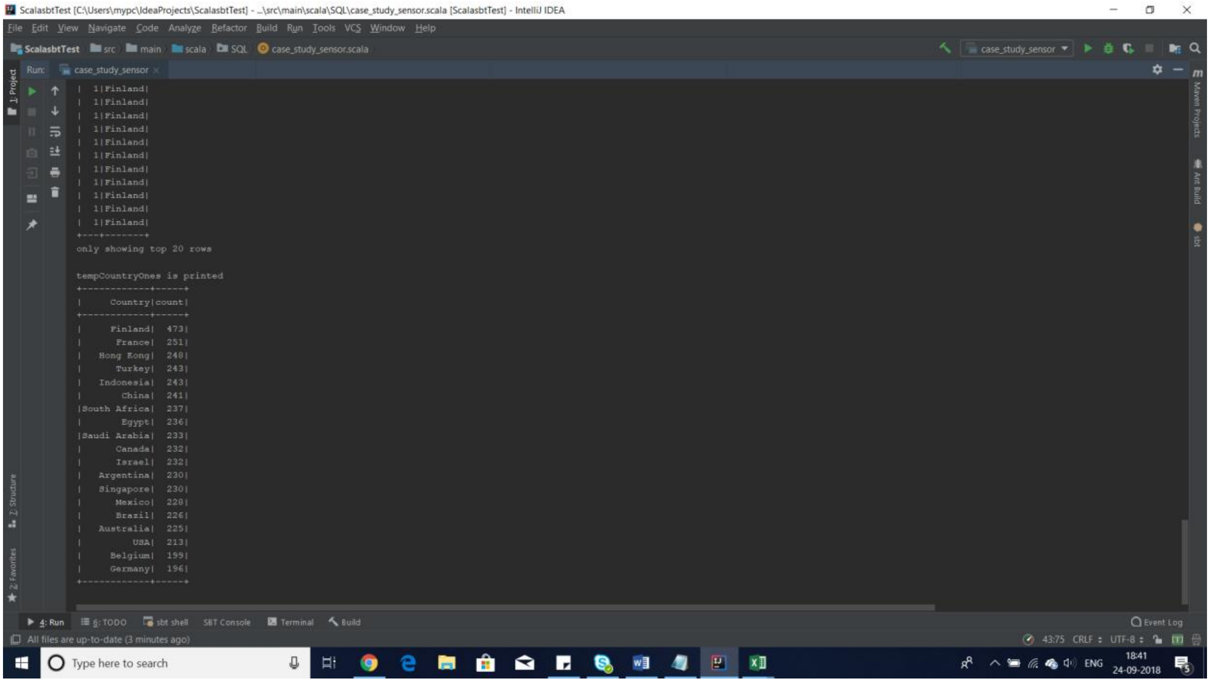
```
tempCountryOnes is printed
+------------+-----+
|     Country|count|
+------------+-----+
|     Finland|  473|
|      France|  251|
|   Hong Kong|  248|
|      Turkey|  243|
|   Indonesia|  243|
|       China|  241|
|South Africa|  237|
|       Egypt|  236|
|Saudi Arabia|  233|
|      Canada|  232|
|      Israel|  232|
|   Argentina|  230|
|   Singapore|  230|
|      Mexico|  228|
|      Brazil|  226|
|   Australia|  225|
|         USA|  213|
|     Belgium|  199|
```

All files are up-to-date (2 minutes ago)



ScalasbtTest [C:\Users\mypc\IdeaProjects\ScalasbtTest] - ...\src\main\scala\SQL\case_study_sensor.scala [ScalasbtTest] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
ScalasbtTest src main scala SQL case_study_sensor.scala
Run: case_study_sensor
4: Run 6: TODO sbt shell SBT Console Terminal Build
Event Log
All files are up-to-date (3 minutes ago)
Type here to search

```
only showing top 20 rows

tempCountry is printed
+---+-------+
| _1|     _2|
+---+-------+
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
+---+-------+
only showing top 20 rows

tempCountryOnes is printed
+------------+-----+
|     Country|count|
+------------+-----+
|     Finland|  473|
|      France|  251|
|   Hong Kong|  248|
|      Turkey|  243|
|   Indonesia|  243|
|       China|  241|
|South Africa|  237|
```



ScalasbtTest [C:\Users\mypc\IdeaProjects\ScalasbtTest] - ...\src\main\scala\SQL\case_study_sensor.scala [ScalasbtTest] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
ScalasbtTest src main scala SQL case_study_sensor.scala
Run: case_study_sensor
4: Run 6: TODO sbt shell SBT Console Terminal Build
Event Log
All files are up-to-date (3 minutes ago)
Type here to search

```
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
|  1|Finland|
+---+-------+
only showing top 20 rows

tempCountryOnes is printed
+------------+-----+
|     Country|count|
+------------+-----+
|     Finland|  473|
|      France|  251|
|   Hong Kong|  248|
|      Turkey|  243|
|   Indonesia|  243|
|       China|  241|
|South Africa|  237|
|       Egypt|  236|
|Saudi Arabia|  233|
|      Canada|  232|
|      Israel|  232|
|   Argentina|  230|
|   Singapore|  230|
|      Mexico|  228|
|      Brazil|  226|
|   Australia|  225|
|         USA|  213|
|     Belgium|  199|
|     Germany|  196|
+------------+-----+
```